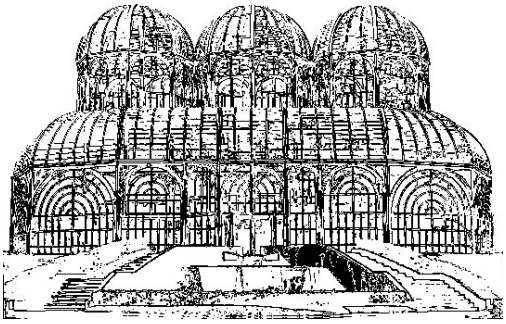
# SBSeg 2012

## XII Simpósio em Segurança da Informação e de Sistemas Computacionais

# ANAIS

**Curitiba PR, 19 a 22 de novembro de 2012**

**SBC - Sociedade Brasileira de Computação**



SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

de 19 a 22 de Novembro de 2012 – Curitiba PR

# ANAIS

**Editora**
Sociedade Brasileira de Computação – SBC

**Organizadores**
Aldri dos Santos
Altair Santin
Carlos Maziero
Michele Nogueira
Ricardo Dahab

**Realização**
Pontifícia Universidade Católica do Paraná – PUCPR
Universidade Federal do Paraná – UFPR
Universidade Tecnológica Federal do Paraná – UTFPR

**Promoção**
Sociedade Brasileira de Computação – SBC

Copyright © 2012 Sociedade Brasileira de Computação
Todos os direitos reservados

Capa e produção gráfica: Carlos Maziero

Dados Internacionais de Catalogação na Publicação

S612 Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (12. : 2012 : Curitiba)
Anais... / XII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais ; organização: Aldri dos Santos et al.— Porto Alegre: Sociedade Brasileira de Computação, 2012.
xvii, 654 p. il. CD-ROM.

ISSN: 2176-0063

1. Ciência da computação. 2. Informática. 3. Segurança da informação. 4. Segurança de sistemas. I. Santos, Aldri dos. II. Título.

CDU 004(063)

Adriano Lopes CRB 9/1429

# Sociedade Brasileira de Computação – SBC

**Presidência**
Paulo Roberto Freire Cunha (UFPE), Presidente
Lisandro Zambenedetti Granville (UFRGS), Vice-Presidente

**Diretorias**
Luciano Paschoal Gaspary (UFRGS), Diretor Administrativo
Luci Pirmez (UFRJ), Diretora de Finanças
Altigran Soares da Silva (UFAM), Diretor de Eventos e Comissões Especiais
Mirella Moura Moro (UFMG), Diretora de Educação
Karin Koogan Breitman (PUC-Rio), Diretora de Publicações
Ana Carolina Salgado (UFPE), Diretora de Planejamento e Programas Especiais
Thais Vasconcelos Batista (UFRN), Diretora de Secretarias Regionais
Edson Norberto Cáceres (UFMS), Diretor de Divulgação e Marketing

**Diretorias Extraordinárias**
Roberto da Silva Bigonha (UFMG), Diretor de Relações Profissionais
Ricardo de Oliveira Anido (UNICAMP), Diretor de Competições Científicas
Raimundo Macêdo (UFBA), Diretor de Cooperação com Sociedades Científicas
Sérgio Vanderlei Cavalcante, Diretor de Articulação de Empresas

**Contato**
Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS

CNPJ: 29.532.264/0001-78
http://www.sbrc.org.br

# Organização do SBSeg 2012

**Coordenação Geral**
Aldri dos Santos, UFPR
Altair Santin, PUCPR
Carlos Maziero, UTFPR

**Comitê de Organização Local**
Arlindo Marcon Jr., PUCPR/IFPR
Elisa Mannes, UFPR
Fernando Henrique Gielow, UFPR
Luiz Albini, UFPR
Maicon Stihler, PUCPR
Marcos Pchek Laureano, IFPR
Robson de Melo, UFPR/UNEMAT

**Coordenação do Comitê de Programa**
Michele Nogueira, UFPR
Ricardo Dahab, UNICAMP

**Coordenação de Minicursos**
Paulo André da S. Gonçalves, UFPE

**Coordenação do Concurso de Teses e Dissertações em Segurança**
Routo Terada, IME-USP

**Coordenação do Workshop de Trabalhos de Iniciação Científica e de Graduação**
Carla Merkle Westphall, UFSC

**Coordenação do Workshop de Gestão de Identidades Digitais**
Michelle Wangham, UNIVALI
Ricardo Custódio, UFSC

**Coordenação de Palestras e Tutoriais**
Paulo Licio de Geus, UNICAMP

**Coordenação do Workshop de Forense Computacional**
Anderson Rocha, UNICAMP
Cinthia Freitas, PUCPR

**Comitê Consultivo**
Marinho Pilla Barcellos, UFRGS
Aldri Luiz dos Santos, UFPR
Anderson Alves Nascimento, UnB
André Luiz Moura dos Santos, UECE
Jeroen van de Graaf, UFMG
Luciano Paschoal Gaspary, UFRGS
Michelle Silva Wangham, UNIVALI
Otto Muniz Bandeira Duarte, UFRJ
Ricardo Dahab, UNICAMP

# Mensagem da Coordenação Geral

O Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg) é um evento científico promovido anualmente pela Sociedade Brasileira de Computação (SBC). A partir de 2005, concomitantemente à criação da Comissão Especial em Segurança da Informação e de Sistemas Computacionais, o SBSeg deixou de ser organizado como um *workshop* e adotou o formato de um simpósio completo. Isso o permitiu atender às demandas crescentes da comunidade brasileira de pesquisadores e profissionais atuantes na área, assumindo a posição de principal fórum no País para a apresentação de pesquisas e atividades relevantes ligadas à segurança da informação e de sistemas.

Desde que se estabeleceu como simpósio em 2005, o evento tem sido realizado anualmente, de forma itinerante. Já sediaram o SBSeg as cidades de Florianópolis, Santos, Rio de Janeiro, Gramado, Campinas, Fortaleza e Brasília. A 12a. edição do simpósio ocorre entre 19 e 22 de novembro de 2012 em Curitiba, organizada pelos programas de pós-graduação em informática da Pontifícia Universidade Católica do Parana (PUCPR), da Universidade Federal do Parana (UFPR) e da Universidade Tecnológica Federal do Paraná (UTFPR).

O simpósio conta com uma ampla gama de atividades. Além das sessões técnicas de apresentação de 28 artigos de pesquisa, serão ministrados 4 minicursos em tópicos recentes na área, duas palestras e dois tutoriais proferidos por especialistas internacionais, além da segunda edição do *Concurso de Teses e Dissertações em Segurança* (CTDSeg), que é realizado a cada dois anos. Também haverá três *workshops*: a sexta edição do *Workshop de Trabalhos de Iniciação Científica e de Graduação* (WTICG), a segunda edição do *Workshop de Gestão de Identidades Digitais* (WGID), e *Workshop de Forense Computacional* (WFC), que acontecerá pela primeira vez, vindo de encontro ao interesse de pesquisadores, peritos, consultores e acadêmicos.

O envolvimento e a colaboração de várias pessoas e entidades foram imprescindíveis para a realização deste simpósio. Agradecemos aos membros do comitê de organização local que voluntariamente ajudaram a proporcionar à comunidade de segurança um evento de excelente qualidade. Somos também gratos à SBC pelo estímulo e apoio prestados ao longo do planejamento e organização do evento. Por fim, o apoio financeiro recebido da CAPES, CNPq, CGI.Br, Intel e Banco do Brasil foi fundamental para a viabilização do evento.

Desejamos a todos os participantes do SBSeg uma semana proveitosa e muito agradável. Aproveitem bem o simpósio e sua estadia em Curitiba!

Aldri dos Santos, UFPR
Altair Santin, PUCPR
Carlos Maziero, UTFPR
Coordenadores Gerais do SBSeg 2012

# Mensagem da Coordenação do Comitê de Programa

Gostaríamos inicialmente de agradecer a confiança em nós depositada pelos coordenadores do SBSeg 2012 ao nos convidar para coordenar o Comitê de Programa. Nossa missão, ao aceitar o convite, foi continuar o trabalho dos nossos antecessores na direção do aperfeiçoamento das sessões técnicas como o principal fórum nacional da área de Segurança da Informação e Sistemas para a discussão de ideias em alto nível pelos vários grupos em atividade no país, na academia e em centros especializados. Para esse fim, fizemos um esforço na expansão da representatividade do comitê de programa, sem descuidar da sua qualidade, mantendo, na medida do possível, um certo grau de internacionalização. Fizemos também um esforço para prover os autores de avaliações dos seus trabalhos que os motivassem a continuar enviando sua produção científica ao SBSeg e dando-lhes uma janela de diálogo com os revisores, ainda que limitada em tempo.

Cremos ter sucedido no objetivo qualitativo dos nossos esforços: o Comitê de Programa, com novos centros e novos talentos, fez um esforço admirável de retornar avaliações cuidadosas, e reconsiderá-las, quando foi o caso, à luz de discussões e refutações dos autores. Nossa tarefa, graças ao espírito crítico e cooperativo dos membros do comitê de programa, foi muito facilitada. O comitê foi composto por representantes de 21 instituições do país, praticamente todas em que existem pesquisadores da área de segurança, e 3 do exterior.

Tivemos 67 submissões de artigos completos. Esse número está acima da média histórica do SBSeg e ligeiramente acima da média dos últimos cinco anos, o que revela um continuado interesse e respeito da comunidade pelo evento. Tivemos também a submissão de 14 artigos curtos, com processo de revisão simplificado, 6 dos quais foram aceitos. Esperamos que, mantida a política de submissão de tais artigos, esse número cresça, já que o seu objetivo é o de promover um fórum de discussões de trabalhos em andamento.

Dos 67 artigos completos submetidos, as áreas com maior número de trabalhos foram Criptografia (22), Detecção e Prevenção de Intrusões (21), Segurança de Redes (18), Controle de Acesso e Autenticação (13), Segurança de Serviços Web e Sistemas Distribuídos (13) e Segurança de Dispositivos Móveis e Sistemas Embarcados (12). Apenas um artigo foi submetido por autores integralmente vinculados a centros do exterior. Foram aceitos 22 artigos completos, isto é, aproximadamente um terço dos submetidos.

Gostaríamos mais uma vez de agradecer o trabalho do Comitê de Programa e de todos os revisores, e o convite e apoio continuado dos coordenadores. Que todos possam estar presentes ao SBSeg 2012 para o coroamento dos resultados do nosso trabalho.

Um bom SBSeg 2012 a todos.

Ricardo Dahab, Unicamp
Michele Nogueira, UFPR
Coordenadores do Comitê de Programa do SBSeg 2012

# Mensagem da Coordenação do CTDSeg

O Concurso de Teses e Dissertações em Segurança da Informação e Sistemas Computacionais (CTDSeg), evento integrante do SBSeg, visa selecionar e premiar as melhores teses de doutorado e as melhores dissertações de mestrado na área concluídas nos últimos dois anos.

Pesquisadores foram convidados a submeter um artigo descrevendo e resumindo os resultados obtidos em suas teses e dissertações concluídas no período de agosto de 2010 a junho de 2012. Nesta edição, doze artigos descrevendo os resultados de dissertações de mestrado ou de teses de doutorado foram submetidos ao CTDSeg 2012.

Os artigos submetidos passaram por uma etapa inicial de avaliação que selecionou as quatro melhores teses de doutorado e as cinco melhores dissertações de mestrado. Cada artigo foi avaliado por pelo menos três membros do comitê de avaliação. Os trabalhos selecionados serão apresentados no dia 19 de novembro de 2012, durante a programação do SBSeg 2012, em Curitiba PR.

A tarefa de coordenar o processo de seleção de trabalhos científicos em um evento não é algo simples de ser realizado. Desta tarefa participaram 7 revisores que, de forma voluntária, avaliaram os artigos submetidos. Gostaria de agradecer o empenho dos membros do comitê de avaliação pela alta qualidade do trabalho realizado. Registro um agradecimento especial a todos os autores que prestigiaram o CTDSeg 2012 ao submeterem trabalhos relatando suas pesquisas. Sem os autores e seu interesse pelo evento não teríamos um evento com trabalhos de tanta qualidade.

Gostaria também de agradecer a todos que colaboraram na organização do CTDSeg 2012, especialmente, à Comissão Especial em Segurança da Informação e de Sistemas Computacionais da SBC que vem incentivando continuamente a realização do evento; aos coordenadores do SBSeg 2012, Profs. Altair Santin, Aldri dos Santos e Carlos Maziero, e toda a equipe do comitê local pelo apoio técnico, logístico e de divulgação dedicado à realização desta edição do CTDSeg. Em nome do Comitê de Avaliação, saúdo a todos os participantes do CTDSeg 2012, com votos de um evento bastante profícuo.

Routo Terada, IME/USP
Coordenador do CTDSeg 2012

# Mensagem da Coordenação do WTICG

O Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG), evento integrante do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), visa incentivar a participação de alunos de graduação e recém-graduados na produção e divulgação de trabalhos científicos sobre temas relacionados às áreas de segurança da informação e de sistemas computacionais. Dessa forma, podemos considerar que o WTICG é um espaço de discussão e troca de conhecimento destinado aos estudantes a respeito das pesquisas em andamento ou já finalizadas. Fortalecer e estimular pesquisa e desenvolvimento de trabalhos na área de segurança, na fase da graduação, é imprescindível para o avanço da comunidade brasileira de segurança.

O Comitê de Programa desta edição foi formado por 15 pesquisadores. O comitê contou também com o apoio de 9 avaliadores externos. Dessa forma, 24 pesquisadores participaram do processo de avaliação dos artigos. Cada artigo recebeu pelo menos 3 avaliações e, ao final do processo de avaliação dos artigos submetidos, tivemos ao todo 52 revisões. Dentre os 18 artigos submetidos, 10 artigos foram selecionados para a publicação e apresentação oral no evento. Todos os artigos selecionados atenderam à restrição dos autores serem estudantes de graduação regularmente matriculados, ou ainda, recém-graduados que tenham concluído a graduação após 30 de junho de 2011. Nesta edição, foram recebidos artigos dos estados do Amazonas, Paraná, Santa Catarina, Minas Gerais, Rio de Janeiro e São Paulo.

A programação do WTICG está dividida em 3 sessões técnicas: a primeira sessão trata sobre privacidade e segurança em dispositivos móveis; na segunda sessão os artigos descrevem pesquisas na área de algoritmos, protocolos e técnicas criptográficas e, finalmente, a terceira sessão reúne artigos sobre segurança de sistemas.

Agradeço aos membros do comitê de programa e aos avaliadores externos por aceitarem o convite para fazer parte deste time que, com dedicação e competência, realizou o processo de avaliação dos artigos. Obrigada aos organizadores e coordenadores gerais do SBSeg 2012 tanto pela oportunidade de coordenar o WTICG como pelo apoio recebido para viabilizar o evento. Gostaria de agradecer especialmente aos autores que submeteram os seus trabalhos e que a cada ano fortalecem e consolidam o WTICG.

Desejo a todos os participantes do VI Workshop de Trabalhos de Iniciação Científica e de Graduação (WTICG) um excelente workshop, esperando que ótimas trocas de experiências e discussões aconteçam em Curitiba!

Carla Merkle Westphall, UFSC
Coordenadora do WTICG 2012

# Mensagem da Coordenação do WGID

O Workshop de Gestão de Identidades (WGID), evento integrante do SBSeg, visa ser um fórum para discussões e apresentações técnicas de trabalhos recentes e/ou em andamento em torno do estado da arte de tecnologias relacionadas à gestão de identidades. Além disso, busca-se também identificar os desafios de pesquisa na área e possibilitar o mapeamento dos grupos de pesquisa.

Em sua segunda edição, o programa do Workshop contemplará duas palestras técnicas de profissionais renomados na área de gestão de identidades. A primeira será do Sr. Renato Martini, presidente do ITI (Instituto Nacional de Tecnologia da Informação) e a segunda palestra do Sr. Paulo Pagliusi, diretor da *Cloud Security Alliance – Chapter Brazil*.

Três artigos convidados serão apresentados na sessão técnica deste ano: "Um Survey sobre Ferramentas para Single Sign-On", "A Framework for Secure Single Sign-On" e "Geração de Certificados Digitais a partir da Autenticação Federada Shibboleth". Gostaríamos de agradecer aos pesquisadores que aceitaram o nosso convite.

Com o objetivo de promover o estudo de temas de gestão de identidade, a RNP lançou em 2012 o primeiro Programa de Gestão de Identidades (PGID). Os resultados obtidos nos projetos que participaram do programa serão apresentados e discutidos durante o WGID 2012.

Encerrando o programa de 2012, teremos um Painel de Discussão sobre o Impacto das Novas Tecnologias de Gestão de Identidades na Comunidade Acadêmica que contará com a participação dos coordenadores do Comitê Técnico de Gestão de Identidades da RNP (CT-GId) e com a equipe de Gerência de Serviços da RNP.

Gostaríamos também de agradecer a todos que colaboraram na organização do WGID, especialmente, ao André Marins (RNP) e a professora Noemi Rodriguez (coordenadora do CT-GId da RNP). Agradecemos ainda o apoio da Rede Nacional de Ensino e Pesquisa (RNP), da Comissão Especial em Segurança da Informação e de Sistemas Computacionais da SBC e dos coordenadores do SBSeg 2012.

Em nome do Comitê de Programa, saudamos a todos os participantes do WGID 2012, com votos de um evento bastante profícuo.

Profa. Michelle S. Wangham, UNIVALI
Prof. Ricardo Felipe Custódio, UFSC
Coordenadores do WGID 2012

# Mensagem da Coordenação do WFC

Em 2012, como uma inovação, o SBSeg abre as portas para a área interdisciplinar de Forense Computacional, despertando o interesse de pesquisadores, peritos, consultores e acadêmicos, tanto da área de Informática quanto da área do Direito concentrando esforços em um evento satélite denominado *Workshop de Forense Computacional (WFC)*.

Tal evento representa a Ciência da Computação e Informática trazendo a técnica, os métodos e a segurança necessária em todos os tipos de trabalhos, operações, investigações e atividades relacionadas com a Forense Computacional. Representa, também, o Direito trazendo os aspectos jurídicos, a legislação e as implicações à sociedade brasileira decorrentes de atos ilícitos.

A área de Forense Computacional ocupa-se da atualidade, a qual vem transferindo para o ambiente virtual (digital), a maioria das relações da sociedade convencional. Assim, problemas sociais do cotidiano rompem fronteiras do mundo real e adentram o mundo digital surgindo diversos e novos problemas técnicos e jurídicos. Problemas tais como a invasão de privacidade, descumprimentos de contratos celebrados pela Internet, disseminação de e-mails caluniosos ou difamatórios, pornografia infantil, *cyberbulling*, compartilhamento ilícito de softwares, músicas e filmes, entre outros problemas que passaram a fazer parte da nova realidade vivida pela sociedade brasileira e mundial.

Assim, o WFC 2012 reúne renomados profissionais para ministrar palestras oriundos das diversas esferas: Polícia Federal, Núcleo de Combate aos Cibercrimes – NUCIBER – Curitiba-PR e Polícia Científica do Paraná – Instituto de Criminalística do Paraná. Estas entidades trazem ao evento seus peritos para apresentar os seguintes temas: Desafios na Forense Computacional, Cibercrimes e a Realidade Brasileira e, ainda, A Atuação do Perito Criminal Especialista em Informática Forense.

Além disto, marcam presença no evento profissionais atuantes na área de Direito Digital, Tecnologia da Informação e Segurança. Destaca-se, portanto, os temas: Estratégias de Investigação e Coleta de Evidências: desafios, oportunidades e tendências, Os Documentos e Assinaturas Eletrônicas e Seus Aspectos Legais e, também, Aspectos Jurídicos da Forense Computacional. Foram ainda selecionados seis artigos para apresentação oral durante o evento. Tais artigos versam sobre: Identificação de Autoria em Cartas Forenses, Análise Forense de Imagens Digitais, Observação de Pessoas em Computação Forense e Redes Sociais.

Agradece-se em especial ao Comitê de Programa que auxiliou na avaliação dos artigos submetidos e, conclui-se, portanto, que a Forense Computacional desempenha papel importante e vem sendo considerada como uma peça fundamental do quebra-cabeça da investigação. Saudamos todos os palestrantes e participantes do WFC com os votos de um excelente Workshop e uma ótima estadia em Curitiba!

Cinthia O. de A. Freitas, PUCPR
Anderson Rocha, UNICAMP
Coordenadores do WFC 2012

# Comitê de Programa e Revisores do SBSeg 2012

Abdelhalim Zaidi, Univ. Evry, France
Aldelir Fernando Luiz, UFSC
Aldri dos Santos, UFPR
Alison Arantes, UFMG
Altair Santin, PUCPR
Anderson Ferreira, UNICAMP
Anderson Lira Queiroz, UFPE
Bernardo David, UnB
Anderson Nascimento, UnB
André Grégio, CTI
Antonio Maio, Kryptus
Arlindo L. Marcon Jr., PUCPR
Bernardo David, UnB
Carla Merkle Westphall, UFSC
Carlos Vasconcelos Junior, UFPE
Carlos Maziero, UTFPR
Carlos Silva, Unicamp, UFMS
Carlos Westphall, UFSC
Célio Neves de Albuquerque, UFF
Cinthia Freitas, PUCPR
Charles Prado, Inmetro
Christian Alonso, UFPR
Cleber Souza, Unicamp
Cristiano Silva, UFSJ
Daniel Mesquita, UFU
Daniel Santana de Freitas, UFSC
Dario Fernandes, Unicamp
Davidson Boccardo, Inmetro
Diego Aranha, UnB
Diogo Mattos, UFRJ
Eduardo Souto, UFAM
Elisa Mannes, UFPR
Emerson Ribeiro de Mello, IFSC
Fabio Piva, Unicamp
Flavia Agostini, Inmetro
Felipe Carbone, UFRGS
Fernando Oliveira Castro, UFMG
Fernando Gielow, UFPR
Gabriel Cavalcante, Unicamp
Hao Chi Wong, Intel, USA
Heitor Motta, UFMG
Henrique Arcoverde, UFPE
Hugo Carvalho, UFRJ
Hylson Netto, IFSC
Jéferson Nobre, UFRGS
Jean Martina, Univ. Cambridge, UK
Jeremy Jean, ENS, France
Jeroen van de Graaf, UFMG
Jim Lau, UFSC
Joaquim Celestino Jœnior, UECE
João Vitor Torres, UFRJ
José De Souza, UFC
José Santanna, UFRGS
José Bringel Filho, UFC
Julio Hernandez, Unicamp
Lau Cheuk Lung, UFSC
Leonardo Fagundes, Unisinos
Leonardo Ribeiro, Inmetro
Lisandro Granville, UFRGS
Lucas Bondan, UFRGS
Lucas Muller, UFRGS
Luciana Rech, UFSC
Luciano Paschoal Gaspary, UFRGS
Luiz Fernando Rust Carmo, UFRJ
Luiz Oliveira, UFPR
Luiz Carlos Albini, UFPR
Lyno Ferraz, UFRJ
Maicon Stihler, PUCPR
Marcelo Caggiani Luizelli, UFRGS
Marcelo Pellenz, PUCPR
Marco Rojas, USP
Marcos Simplicio, USP
Matheus Lehmann, UFRGS
Márcio Carvalho, UFRGS
Michel Abdalla, ENS-CNRS, France
Michelle Wangham, Univali
Miguel Franklin de Castro, UFC
Otto Muniz Bandeira Duarte, UFRJ
Paulo Barreto, USP
Paulo Mafra, UFSC
Paulo Rego, UFC
Paulo Andre Goncalves, UFPE
Pedro Pisa, UFRJ
Pedro Velloso, UFF
Rafael Dowsley, UCSD, USA
Rafael Kunst, UFRGS
Rafael Obelheiro, UDESC
Raphael Machado, Inmetro
Raul Ceretta Nunes, UFSM
Raul Weber, UFRGS
Reinaldo Braga, UFC
Ricardo Custódio, UFSC

Roberto Gallo, KRYPTUS
Robson Gomes de Melo, UFPR
Rodrigo Mansilha, UFRGS
Rodrigo Ruas Oliveira, UFRGS
Rossana Andrade, UFC
Routo Terada, USP
Ruy Barretto de Queiroz, UFPE
Sergio Oliveira, UFSJ
Silvio Danilo, UFPE
Thiago Oliveira, UFSJ
Tiago Nascimento, Inmetro
Vinícius Guimar‹es, IFRGS
Vinícius Thiago, UFC
Vitor Afonso, Unicamp
William Schwartz, UFMG
Wilson Melo Jr, Inmetro

## Avaliadores do CTDSeg 2012

Anderson Nascimento, UnB
Jeroen van de Graaf, UFMG
Julio Hernandez, UNICAMP
Michelle Wangham, UNIVALI
Paulo Licio de Geus, UNICAMP
Ricardo Custódio, UFSC
Ricardo Dahab, UNICAMP

## Revisores do WTICG 2012

Altair Olivo Santin, PUCPR
André Luiz Moura dos Santos, UECE
Antônio Marinho Barcellos, UFRGS
Carla Merkle Westphall, UFSC
Carlos Oberdan Rolim, UFRGS
Daniel Ricardo dos Santos, UFSC
Edmar Santana de Rezende, UNICAMP
Evaldo Martins de Souza, UFMG
José Antônio Moreira Xexéo, IME
Henrique Kawakami, KRIPTUS
Jorge Werner, UFSC
Leonardo Barbosa e Oliveira, UFMG
Marcos Simplício Jr, Poli-USP
Michele Nogueira Lima, UFPR
Michelle Silva Wangham, UNIVALI
Natália Castro Fernandes, UFF
Paulo André da Silva Gonçalves, UFPE
Paulo Fernando da Silva, UFSC
Paulo Messeder Barreto, USP
Rafael Bosse Brinhosa, UFSC
Rafael Rodrigues Obelheiro, UDESC
Raul Ceretta Nunes, UFSM
Shirlei Aparecida de Chaves, UFSC
William de Souza, CASNAV

## Comitê de Programa do WGID 2012

Adilson Eduardo Guelfi, IPT
Aldri dos Santos, UFPR
Altair Santin, PUCPR
Débora Muchaluat, UFF
Emerson Ribeiro de Mello, IFSC
Jeroen van de Graaf, UFMG
Joni Fraga, UFSC
Michelle Wangham, UNIVALI
Michele Nogueira, UFPR
Noemi Rodriguez, PUC-Rio
Ricardo Custódio, UFSC
Roberto Samarone, UFPA
Vinod Rebello, UFF

## Comitê de Programa do WFC 2012

Alceu de Souza Britto Jr, PUCPR
Eduardo Valle, UNICAMP
Helio Pedrini, UNICAMP
Julio Hernandez, UNICAMP
Paulo Miranda, IME/USP
William Schwartz, UFMG

# Sumário

## Trilha principal (artigos completos)

### I. Detecção e Prevenção de Ataques e Vulnerabilidades

1. Análise de métodos de aprendizagem de máquina para detecção automática de spam hosts.
*Renato Moraes Silva (UNICAMP), Tiago Almeida (UFSCar), Akebo Yamakami (UNICAMP)* ..... 2

2. Avaliação da sensibilidade de preditores de suavização exponencial na detecção de ataques de inundação.
*Nícolas Silva, Ronaldo Salles (IME)* ..... 16

3. Detecção de variações de malware metamórfico por meio de normalização de código e identificação de subfluxos.
*Marcelo Cozzolino (Depto Polícia Federal do Brasil), Gilbert Martins (FUCAPI), Eduardo Souto (UFAM), Flavio Deus (UnB)* ..... 30

4. Dynamic detection of address leaks.
*Gabriel Silva Quadros, Rafael Martins, Fernando Quintão Pereira (UFMG)* ..... 44

5. SDA-COG Sistema de detecção de ataques para rede de rádios cognitivos.
*Joffre Filho (UFRJ), Luiz Rust, Raphael Machado (Inmetro), Luci Pirmez (UFRJ)* ..... 58

### II. Criptografia e PKI

6. Aprimoramento de esquema de identificação baseado no problema MQ.
*Fábio Monteiro, Denise Goya, Routo Terada (USP)* ..... 72

7. Chi-square attacks on Block-Cipher based compression functions.
*Daniel Santana de Freitas (UFSC), Jorge Nakahara Jr* ..... 86

8. Impossible-Differential attacks on block-cipher based hash and compression functions using 3D and Whirlpool.
*Daniel Santana de Freitas (UFSC), Jorge Nakahara Jr* ..... 99

9. Segurança do bit menos significativo no RSA e em curvas elípticas.
*Dionathan Nakamura, Routo Terada (USP)* ..... 112

10. Universally composable committed oblivious transfer with a trusted initializer.
*Adriana Pinto, Anderson Nascimento, Bernardo David (UnB), Jeroen Graaf (UFMG)* ..... 126

11. Cleaning up the PKI for long-term signatures.
*Martín Vigil (Tech Univ Darmstadt), Ricardo Custódio (UFSC)* ..... 140

### III. Mitigação e Tolerância a Ataques

12. Mitigando ataques de egoísmo e negação de serviço em nuvens via

agrupamento de aplicações.
*Daniel Marcon, Miguel Neves, Rodrigo Ruas Oliveira, Luciana Buriol, Luciano Paschoal Gaspary, Marinho Barcellos (UFRGS)* ..... 154

13. Um esquema cooperativo para análise da presença de ataques EUP em redes ad hoc de rádio cognitivo.
*Julio Soto, Saulo Queiroz, Michele Nogueira (UFPR)* ..... 168

14. Usando criptografia de limiar para tolerar clientes maliciosos em memória compartilhada dinâmica.
*Eduardo Alchieri (UnB), Alysson Bessani (Univ. Lisboa), Joni da Silva Fraga (UFSC)* ..... 182

**IV. Segurança Baseada em Hardware e Controle de Acesso**

15. A secure relay protocol for door access control.
*Hans Hüttel, Erik Wognsen, Mikkel Normann Follin, Marcus Calverley, Henrik Søndberg Karlsen, Bent Thomsen (Aalborg University)* ..... 196

16. Modified Current Mask Generation (M-CMG): an improved countermeasure against differential power analysis attacks.
*Daniel Mesquita (UFU), Hedi Besrour, Machhout Mohsen, Tourki Rached (Monastir University)* ..... 210

17. Uma arquitetura de segurança para medidores inteligentes – verificação prática de dados de energia multitarifada.
*Sérgio Câmara (UFRJ), Raphael Machado (Inmetro), Luci Pirmez (UFRJ), Luiz Rust (Inmetro)* ..... 221

**V. Virtualização**

18. Redes virtuais seguras: uma nova abordagem de mapeamento para proteger contra ataques de disrupção na rede física.
*Rodrigo Ruas Oliveira, Leonardo Bays, Daniel Marcon, Miguel Neves, Luciana Buriol, Luciano Paschoal Gaspary (UFRGS)* ..... 235

19. Um modelo para mapeamento ótimo de redes virtuais com requisitos de segurança.
*Leonardo Bays, Rodrigo Ruas Oliveira, Luciana Buriol, Marinho Barcellos, Luciano Paschoal Gaspary (UFRGS)* ..... 249

20. Uma arquitetura para auditoria de nível de serviço para computação em nuvem.
*Juliana Bachtold, Altair Santin, Maicon Stihler, Arlindo L. Marcon Jr., Eduardo Viegas (PUCPR)* ..... 263

**VI. Criminalística**

21. Identificação de autoria de documentos eletrônicos.
*Walter Oliveira Jr, Edson Justino (PUCPR), Luiz Oliveira (UFPR)* ..... 277

22. Modelagem de aliciamento de menores em mensagens instantâneas de texto.

*Priscila Santin, Cinthia Freitas, Emerson Paraiso, Altair Santin (PUCPR)* ..... 288

## Trilha principal (artigos curtos)

1. Arquitetura de sistema integrado de defesa cibernética para detecção de botnets.
*Sérgio Cardoso, Ronaldo Salles (IME-RJ)* ..... 303

2. Avaliação do classificador Artmap Fuzzy em redes 802.11 com criptografia pré-RSN (WEP e WPA).
*Nelcileno Araujo (UFMT), Ruy de Oliveira (IFMT), Ailton Shinoda (UNESP), Ed' Wilson Ferreira, Valtemir do Nascimento (IFMT)* ..... 310

3. Identity management requirements in future Internet.
*Jenny Torres (Univ. Paris 6), Ricardo Tombesi Macedo, Michele Nogueira (UFPR), Guy Pujolle (Univ. Paris 6)* ..... 317

4. IPSFlow – uma proposta de IPS distribuído para captura e bloqueio seletivo de tráfego malicioso em redes definidas por software.
*Fábio Nagahama, Fernando Farias (UFPA), Elisangela Aguiar (Univ Amazônia/SERPRO), Eduardo Cerqueira, Antônio Abelém (UFPA), Luciano Gaspary (UFRGS)* ..... 324

5. Uma arquitetura para mitigar ataques DDoS em Serviços Web sob nuvem.
*Cinara Menegazzo (UDESC), Fernando Bernardelli, Fernando Gielow, Nadine Pari, Aldri dos Santos (UFPR)* ..... 331

6. Um modelo de segurança e privacidade para redes sociais móveis aplicadas à área da saúde.
*Jesseildo Gonçalves, Ariel Teles, Francisco José Silva (UFMA)* ..... 338

## CTDSeg – Concurso de Teses e Dissertações em Segurança

### I. Teses de doutorado

1. Implementação eficiente em software de curvas elípticas e emparelhamentos bilineares.
*Diego Aranha (UnB), Julio Hernandez (UNICAMP)* ..... 346

2. Criptografia de chave pública sem certificado.
*Denise Goya, Routo Terada (USP)* ..... 354

3. An orchestration approach for unwanted Internet Traffic Identification.
*Eduardo Feitosa (UFAM), Djamel Sadok (UFPE)* ..... 362

4. Análise de políticas de controle de acesso baseado em papéis com rede de Petri colorida.
*Eduardo Takeo Ueda, Wilson Ruggiero (USP)* ..... 372

**II. Dissertações de mestrado**

5. Segurança do bit menos significativo no RSA e em curvas elípticas.
*Dionathan Nakamura, Routo Terada (USP)* ..... 380

6. Um esquema bio-inspirado para tolerância a má-conduta em sistemas de quórum apoiando serviços em MANETs.
*Elisa Mannes, Michele Nogueira, Aldri dos Santos (UFPR)* ..... 388

7. Assinatura digital Rabin-Williams sem randomização e com prova eficiente de segurança.
*Bernardo Magri, Routo Terada (USP)* ..... 396

8. Protocolos de acordo de chaves baseados em emparelhamentos para dispositivos móveis.
*Cleber Okida, Routo Terada (USP)* ..... 404

9. Mineração de dados para detecção de fraudes em transações eletrônicas.
*José Felipe Jr, Wagner Meira Jr. (UFMG)* ..... 413

## WTICG – Workshop de Trabalhos de Iniciação Científica e de Graduação

1. Uma estratégia para gerenciar o compartilhamento de recursos entre dispositivos móveis.
*Tiago Bono, Luciana Martimiano (UEM)* ..... 422

2. Computação distribuída com preservação de privacidade.
*Luís Pena, Olivia Coelho, Jeroen van de Graaf (UFMG)* ..... 432

3. Senhas descartáveis em dispositivos móveis para ambientes de Telemedicina.
*Thaís Idalino, Dayana Spagnuelo (UFSC)* ..... 442

4. Modelagem e verificação formal de aspectos de tempo real do protocolo Kerberos.
*Bruno Cremonezi, Guilherme Pinto (UFJF)* ..... 452

5. Análise e implementação de um método para prover integridade a sistemas de banco de dados.
*Gabriel Becker, Lucas Perin, Anderson Silvério (UFSC), Marcelo Carlos (Univ. London), Ricardo Custódio (UFSC)* ..... 461

6. Um protocolo criptográfico para controle de medicamentos.
*Bruno Imhof, Eduardo dos Santos, Ricardo Custódio (UFSC)* ..... 471

7. Adaptações em um HSM para homologação na ICP-Brasil.
*Camila Cardozo, Guilherme Welter, Cláudio Dettoni Jr, André Bereza Jr, Ricardo Custódio (UFSC)* ..... 481

8. Detecção de phishing em páginas Web utilizando técnicas de aprendizagem de máquina.
*Francisco Cunha, Eulanda dos Santos, Eduardo Souto (UFAM)* ..... 491

9. Desenvolvimento de roteiros laboratoriais de segurança computacional em ambientes virtualizados.
*Rodrigo Ferreira, Carla Westphall (UFSC)* ..... 501

10. CURUPIRA: uma solução para controle dos pais em navegadores Web.
*Diego Azulay, Hugo Cunha, Eduardo Feitosa (UFAM)* ..... 511

## WGID – Workshop de Gestão de Identidades Digitais

1. Um survey sobre ferramentas para single sign-on.
*Hendri Nogueira, Douglas Santos, Ricardo Custódio (UFSC)* ..... 522

2. A framework for secure single sign-on.
*Bernardo David, Anderson Nascimento, Rafael Tonicelli, Rafael de Sousa Jr (UnB)* ..... 543

3. Geração de certificados digitais a partir da autenticação federada Shibboleth.
*Michelle Wangham (UNIVALI), Emerson de Mello (IFSC), Davi Böger, Joni Fraga (UFSC), Marlon Guérios (Inohaus)* ..... 555

## WFC – Workshop de Forense Computacional

1. Metodologia para análise forense de imagens digitais utilizando sensor pattern noise.
*Alisson Nascimento, Leonardo Torres, Valter Ramos, José Alencar-Neto (RASTRU)* ..... 576

2. Além do óbvio: a análise forense de imagens e a investigação do conteúdo implícito e explícito de fotografias digitais.
*Tiago Carvalho, Ewerton Silva, Filipe Costa, Anselmo Ferreira, Anderson Rocha (UNICAMP)* ..... 584

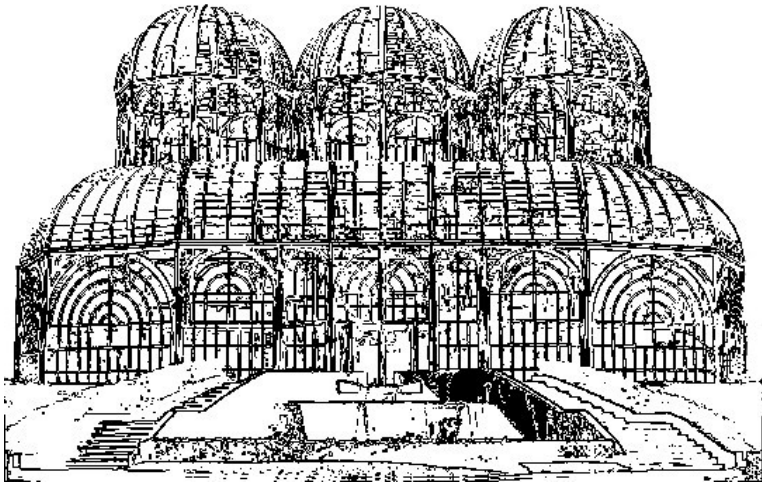
3. Identificação de autoria offline em documentos manuscritos.
*Aline Amaral (CESUMAR), Cinthia Freitas (PUCPR), Flávio Bortolozzi (CESUMAR)* ..... 598

4. Aplicação de observação de pessoas em computação forense.
*William Schwartz (UFMG)* ..... 611

5. Content-based filtering for video sharing social networks.
*Eduardo Valle (UNICAMP), Sandra Avila, Fillipe de Souza (UFMG), Marcelo Coelho (UFMG, EPCAR), Arnaldo Araújo (UFMG)* ..... 625

6. Identificação da origem de imagens com zoom utilizando sensor pattern noise.
*Leonardo Torres (RASTRU, UFAL), Alisson Nascimento, Valter Ramos, José Alencar-Neto (RASTRU), Alejandro Frery (UFAL)* ..... 639

**Índice de autores** ..... 653



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# Trilha principal – Artigos Completos

# Análise de Métodos de Aprendizagem de Máquina para Detecção Automática de *Spam Hosts*

**Renato Moraes Silva**[1], **Tiago A. Almeida**[2], **Akebo Yamakami**[1]

[1]Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP) – Campinas – SP – Brasil

[2]Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar) – Sorocaba – SP – Brasil

{renatoms,akebo}@dt.fee.unicamp.br, talmeida@ufscar.br

***Abstract.*** *Web spamming is one of the main problems that affect the quality of search engines. The number of web pages that use this technique to achieve better positions in search results is growing. The main motivation is the profit achieved with the online advertising market, besides attacks on Internet users through malware that steal information to facilitate bank thefts. Given this scenario, this paper presents an analysis of machine learning techniques employed to detect spam hosts. Experiments performed with a real, public and large dataset, indicate that ensemble of decision trees are promising in the task of spam hosts detection.*

***Resumo.*** *Web spamming é um dos principais problemas que afeta a qualidade das ferramentas de busca. O número de páginas web que usam esta técnica para conseguir melhores posições nos resultados de busca é cada vez maior. A principal motivação são os lucros obtidos com o mercado de publicidade online, além de ataques a usuários da Internet por meio de malwares, que roubam informações para facilitar roubos bancários. Diante disso, esse trabalho apresenta uma análise de técnicas de aprendizagem de máquina aplicadas na detecção de spam hosts. Experimentos realizados com uma base de dados real, pública e de grande porte indicam que as técnicas de agregação de métodos baseados em árvores são promissoras na tarefa de detecção de spam hosts.*

## 1. Introdução

As ferramentas de busca são grandes aliadas dos usuários da Internet para encontrar informações e por isso são responsáveis por uma porcentagem expressiva das visitas recebidas pela maioria dos *web sites*. Logo, para ter sucesso, é importante que o *web site* mantenha um alto *ranking* de relevância nos motores de busca (*pagerank*). Dessa forma, poderá melhorar seu posicionamento nas páginas de resultado dessas ferramentas, quando forem consultados termos relacionados ao seu conteúdo ou serviços oferecidos. Para atingir esse propósito, podem ser usadas diversas estratégias, conhecidas como otimização para motores de busca (SEO – *search engine optimization*) [Ledford 2009].

Existem diversas estratégias éticas de SEO, porém como afirma [Ledford 2009], para aprender as mais bem sucedidas, é preciso muito tempo e dedicação. Todos os elementos do *web site* devem ser projetados para maximizar seu *ranking* nos motores de

 ©2012 SBC — Soc. Bras. de Computação

busca. No entanto, como mencionado no guia de SEO publicado pela empresa Google[1], esse processo deve ser pensado principalmente para os consumidores do *web site* e não para as ferramentas de busca. É importante que o projeto do *web site* ofereça conteúdo relevante, facilidade de navegação e outras características que beneficiem o usuário. O problema é que muitos *sites* preferem investir em técnicas antiéticas de SEO, enganando os motores de busca para ganhar relevância sem merecê-la. Os maiores prejudicados são os usuários que ao fazerem suas consultas recebem respostas inesperadas, irrelevantes e muitas vezes infectadas por conteúdos maliciosos e prejudiciais, como *malwares* ou outras pragas virtuais. Essa técnica é conhecida como *web spamming* ou *spamdexing* [Svore et al. 2007] e, segundo [Gyongyi e Garcia-Molina 2005], tais páginas podem possuir tanto conteúdo *spam* quanto *spam links*. O primeiro consiste em criar páginas com milhares de palavras-chaves irrelevantes e o segundo consiste em adicionar *links* que apontam para as páginas que pretende-se promover.

Um dos motivos que fomenta a criação e o crescimento do volume de *web spam* é o incentivo econômico a essas práticas. Geralmente, os *spammers* criam páginas com conteúdo irrelevante, que conseguem boas posições no *ranking* dos motores de busca por meio de técnicas de *spamming*, e colocam anúncios para produtos de *sites* de compras ou para outras modalidades de *sites*. Assim, ele recebe algum dinheiro por cada clique dado pelos usuários [Gyongyi e Garcia-Molina 2005].

Estudos recentes apontam que o volume de *web spam* vem aumentando consideravelmente nos últimos anos. Segundo [Ntoulas et al. 2006], cerca de 13,8% da páginas de língua inglesa, 25% das francesas e 22% das germânicas são *spam*. Em outro estudo, [John et al. 2011] observaram que 36% dos resultados dos motores de busca Google e Bing contém URLs maliciosas. Um relatório produzido pela empresa Websense[2] mostra que 22,4% dos resultados de busca sobre entretenimento levam a *links* maliciosos. Além disso, segundo um relatório publicado pela empresa McAfee[3], 49% dos termos de busca mais populares retornam algum *site* malicioso entre os 100 primeiros resultados da busca. A mesma pesquisa aponta que 1,2% das consultas retornam *links* de *sites* maliciosos entre os 100 primeiros resultados.

Diante desse cenário, este trabalho apresenta uma análise de desempenho de diversas técnicas bem conhecidas de aprendizado de máquina que podem ser aplicadas para auxiliar no combate desse problema. O objetivo é encontrar métodos promissores que podem ser explorados e empregados para auxiliar na detecção automática de *spam hosts*. Resultados preliminares obtidos com redes neurais artificiais usando atributos extraídos do conteúdo das páginas *web* foram publicados em [Silva et al. 2012a, Silva et al. 2012b]. Contudo, neste artigo são oferecidas as seguintes contribuições:

- experimentos realizados com diversos outros métodos, tais como métodos de agregação de múltiplos classificadores, métodos baseados em árvores, métodos baseados em *cluster* e máquinas de vetores de suporte.
- experimentos realizados com diferentes vetores de características (atributos) extraídos de páginas *web* disponíveis em uma base de dados grande, real e pública;

[1] *Search Engine Optimization Starter Guide*. Consultar: `http://www.google.co.jp/intl/en/webmasters/docs/search-engine-optimization-starter-guide.pdf`

[2] *Websense 2010 Threat Report*. Consultar: `http://www.websense.com/assets/reports/report-websense-2010-threat-report-en.pdf`

[3] *McAfee Threats Report: First Quarter 2011*. Consultar: `http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2011.pdf`

 ©2012 SBC — Soc. Bras. de Computação

- experimentos que ilustram os impactos na acurácia dos métodos devido ao balanceamento dos dados de treinamento;
- comparação dos resultados obtidos nesse trabalho com os resultados obtidos por métodos propostos por outros autores da literatura;

Este artigo está estruturado da seguinte forma: na Seção 2 são brevemente descritos os trabalhos correlatos disponíveis na literatura. Na Seção 3 são apresentados os conceitos básicos sobre os métodos classificadores avaliados neste trabalho. Na Seção 4 são descritas as configurações adotadas nos experimentos e a base de dados empregada. Os resultados experimentais são apresentados na Seção 5. Por fim, conclusões e direções para trabalhos futuros são descritos na Seção 6.

## 2. Trabalhos correlatos

O problema de *web spamming* é relativamente recente e, portanto, existem poucos trabalhos que oferecem avanços significativos no sentido de resolvê-lo. Em um dos trabalhos da literatura de *web spam*, [Gyongyi e Garcia-Molina 2005] descrevem uma variedade de técnicas empregadas pelos *spammers* para disseminar *web spam*. Dentre elas, as mais comuns são: inserção de palavras chaves populares no título das páginas ou nas *meta tags*, métodos que tornam o texto invisível para o usuário, que exploram *links* e que redirecionam o usuário para outras páginas.

Em outro trabalho [Castillo et al. 2007] propõem um conjunto de características baseadas no conteúdo das páginas *web* para separar os *spam hosts* dos *ham hosts* e apresentam um sistema de detecção de *web spam* que combina características baseadas nos *links* e no conteúdo. Além disso, eles usam a topologia do grafo *web* através da exploração da dependência de *links* entre as páginas *web*. Em contrapartida, em uma abordagem diferente da adotada no trabalho anterior, [Svore et al. 2007] usam o método de SVM na detecção de *web spam* através de características baseadas no conteúdo e que consideram o tempo de *rank*.

[Shengen et al. 2011] propõem novas características para a identificação de *web spam* por meio de programação genética usando características baseadas em *links* e apresenta os efeitos do número de indivíduos, do número de gerações e da profundidade da árvore binária nos resultados do algoritmo. [Rungsawang et al. 2011] apresentam um algoritmo de otimização por colônia de formigas para a detecção de *spam hosts* que explora tanto características baseadas no conteúdo quanto características baseadas nos *links*.

[Jayanthi e Sasikala 2012] apresentam um algoritmo para detecção de *web spam* chamado `WESPACT` que usa algoritmos genéticos para classificar atributos baseados em *links* e em conteúdo. Seguindo a mesma linha do trabalho anterior, com o uso de algoritmos de computação natural, [Taweesiriwate et al. 2012] propõem o uso do algoritmo de otimização por colônia de formigas para a detecção de web spam, porém seguem a estratégia de *TrustRank* [Gyongyi et al. 2004] a fim de gerar regras de classificação para a detecção de *web spam*. [Largillier e Peyronnet 2012] usam uma abordagem diferente para o problema e apresentam diversos métodos de agrupamento de nós para a diminuição dos efeitos do *web spamming* no algoritmo *PageRank* (algoritmo de *ranking* dos motores de busca). Por outro lado, [Liu et al. 2012] propõem algumas características baseadas no comportamento dos usuários para separar as páginas *spam* das páginas legítimas (*ham*).

## 3. Classificadores

Apesar da existência de trabalhos cujo intuito é filtrar *web spam*, ainda não há um consenso se a aplicação de técnicas de aprendizado de máquina é realmente eficaz na detecção

 ©2012 SBC — Soc. Bras. de Computação

automática de *spam hosts*. Tendo isso em vista, foram avaliados neste trabalho diversos métodos de classificação bem conhecidos.

Nessa seção, são apresentados os métodos classificadores avaliados nesse trabalho: redes neurais artificiais (RNAs) perceptron de múltiplas camadas (MLP – *multilayer perceptron*), máquina de vetores de suporte (SVM – *support vector machines*), métodos baseados em árvores (C4.5 e floresta aleatória), IBK, *boosting* adaptativo (AdaBoost – *adaptive boosting*), *bagging* e LogitBoost. A escolha de tais métodos reside no fato de terem sido avaliados e listados como as melhores técnicas de mineração de dados atualmente disponíveis [Wu et al. 2008].

As RNAs não fazem parte da lista proposta por [Wu et al. 2008], mas foram escolhidas para serem avaliadas devido a sua alta capacidade de generalização. A técnica de floresta aleatória também foi escolhida, mesmo não estando na lista, pois ela faz uma combinação de árvores de decisão. Logo, como o método C4.5 é um algoritmo de árvore de decisão e está na lista dos melhores métodos, acredita-se que a combinação de árvores de decisão também possa gerar bons resultados. Por fim, também foram feitos experimentos com o método OneR pois ele tem um baixo custo computacional e é um dos algoritmos de aprendizagem de máquina mais simples. Logo, o seu desempenho pode ser usado para analisar o desempenho obtido por algoritmos mais complexos.

### 3.1. Rede neural artificial perceptron de múltiplas camadas

Uma RNA perceptron de múltiplas camadas (MLP – *multilayer perceptron*) é uma rede do tipo *perceptron* que possui um conjunto de unidades sensoriais que formam a camada de entrada, uma ou mais camadas intermediárias de neurônios computacionais e uma camada de saída [Haykin 1998]. Por padrão, o seu treinamento é supervisionado e usa o algoritmo *backpropagation* (retropropagação do erro), que tem a função de encontrar as derivadas da função de erro com relação aos pesos e bias da rede. Esse algoritmo pode ser resumido em duas etapas: *forward* e *backward* [Bishop 1995].

Na etapa *forward*, o sinal é propagado pela rede, camada a camada, da seguinte forma: $u_j^l(n) = \sum_{i=0}^{m^{l-1}} w_{ji}^l(n) y_i^{l-1}(n)$, sendo $l = 0, 1, 2, ..., L$ o índice das camadas da rede. Quando $l = 0$, ele representa a camada de entrada e quando $l = L$ representa a camada de saída. Já, $y_i^{l-1}(n)$ é a função de saída do neurônio $i$ na camada anterior $l-1$, $w_{ji}^l(n)$ é o peso sináptico do neurônio $j$ na camada $l$ e $m^l$ é a quantidade de neurônios na camada $l$. Para $i = 0$, $y_0^{l-1}(n) = +1$ e $w_{j0}^l(n)$ representa o bias aplicado ao neurônio $j$ da camada $l$. A saída do neurônio é dada por: $y_j^l(n) = \varphi_j(u_j^l(n))$, onde $\varphi_j$ é a função de ativação do neurônio $j$. O erro pode ser calculado por: $e_j^l(n) = y_j^l(n) - d(n)$, sendo que $d(n)$ é a saída desejada para o padrão de entrada $x(n)$.

Na etapa *backward*, inicia-se a derivação do algoritmo *backpropagation*, a partir da camada de saída, onde tem-se: $\delta_j^L(n) = \varphi_j'(u_j^L(n)) e_j^L(n)$, sendo $\varphi_j'$ a derivada da função de ativação. Para $l = L, L-1, ..., 2$, calcula-se: $\delta_j^{l-1}(n) = \varphi_j'(u_j^{l-1}(n)) \sum_{i=1}^{m^l} w_{ji}^l(n) * \delta_j^l(n)$, para $j = 0, 1, ..., m^l - 1$.

Para maiores detalhes sobre as MLPs, consulte [Bishop 1995, Haykin 1998].

 ©2012 SBC — Soc. Bras. de Computação

### Algoritmo de Levenberg-Marquardt

O algoritmo de Levenberg-Marquardt é um método de otimização e aceleração da convergência do algoritmo *backpropagation*. Ele é considerado um método de segunda ordem, assim como os métodos do gradiente conjugado e do método quase-Newton, pois utiliza informações sobre a derivada segunda da função de erro [Bishop 1995].

## 3.2. Máquinas de vetores de suporte

Máquinas de vetores de suporte (SVM – *support vector machines*) [Cortes e Vapnik 1995] é um método de aprendizagem de máquina que pode ser usado para problemas de classificação e regressão e outras tarefas de aprendizagem [Haykin 1998, Chang e Lin 2011]. Elas foram conceitualmente implementadas seguindo a ideia de que vetores de entrada são não-linearmente mapeados para um espaço de atributos de alta dimensão. Nesse espaço, é construída uma superfície de decisão que permite distinguir as classes dos exemplos de entrada.

Para conseguir separar dados linearmente ou não-linearmente separáveis, um dos principais elementos usados pelo método SVM é uma função de *kernel*. Através dela, o SVM constrói uma superfície de decisão que é não-linear no espaço de entrada, mas é linear no espaço de atributos [Haykin 1998]. As principais funções de *kernel* que podem ser utilizadas no SVM são [Haykin 1998, Hsu et al. 2003]: linear, radial basis function (RBF), polinomial e sigmoidal.

Para a escolha dos parâmetros do SVM, a técnica recomendada em [Hsu et al. 2003] é a *grid search* (busca em grade). Então, considerando o SVM com *kernel* RBF, em que deve-se definir o parâmetro de regularização C e o parâmetro $\gamma$, eles propõem testar as seguintes sequências exponenciais: $C = 2^{-5}, 2^{-4}, 2^{-3}..., 2^{15}$ e $\gamma = 2^{-15}, 2^{-14}..., 2^{3}$.

## 3.3. C4.5

O C4.5 [Quinlan 1993] é um dos mais clássicos algoritmos de árvores de decisão e trabalha tanto com atributos categóricos quanto contínuos. Além disso, ele permite o uso de atributos desconhecidos, desde que sejam representados por "?". O C4.5 usa um método de dividir e conquistar para aumentar a capacidade de predição das árvores de decisão. Dessa forma, um problema é dividido em vários sub-problemas, criando-se sub-árvores no caminho entre a raiz e as folhas da árvore de decisão.

## 3.4. Floresta aleatória

Uma floresta aleatória (*random forest*) [Breiman 2001] é uma combinação de árvores de decisão, em que cada árvore depende dos valores de vetores aleatórios amostrados de forma independente e distribuídos igualmente para todas as árvores na floresta. Nesse método, depois que um determinado número de árvores são geradas, cada uma lança um voto para uma classe do problema, considerando um vetor de entrada. Então, a classe mais votada será escolhida na predição do classificador.

## 3.5. IBK

O algoritmo IBK é um algoritmo de aprendizagem baseado em instâncias (IBL – *instance-based learning*) [Aha et al. 1991]. Esse tipo de algoritmo é derivado do método de classificação $k$-vizinhos mais próximos (KNN – $k$-*nearest neighbor*). Porém, este último

 ©2012 SBC — Soc. Bras. de Computação

é um algoritmo não-incremental e tem como objetivo principal manter uma consistência perfeita com o conjunto inicial de treinamento. Já o algoritmo do tipo IBL é incremental e tem como objetivo maximizar a acurácia sobre novas instâncias do problema [Aha et al. 1991].

Assim como no método KNN, no método IBK existe uma função de similaridade que obtém um valor numérico obtido pelo cálculo da distância euclidiana. Então a classificação gerada para um padrão $i$ será influenciada pelo resultado da classificação dos seus $k$-vizinhos mais próximos, pois padrões similares devem ter classificações similares [Aha et al. 1991, Witten e Frank 2005].

### 3.6. *Boosting* adaptativo

O método de *boosting* adaptativo (AdaBoost – *adaptive boosting*) [Freund e Schapire 1996] é um algoritmo de *boosting* amplamente utilizado para problemas de classificação. Em geral, assim como qualquer método de *boosting*, ele faz uma combinação de classificadores. Porém, segundo [Freund e Schapire 1996], ele possui algumas propriedades que o tornam mais prático e fácil de ser implementado do que os algoritmos de *boosting* que o antecederam, pois ele não necessita de nenhum conhecimento prévio das predições obtidas por classificadores ruins. Em vez disso ele se adapta as predições ruins e gera uma hipótese majoritária ponderada, em que o peso das predições fornecidas pelos classificadores ruins torna-se uma função de sua predição.

### 3.7. *Bagging*

O *bagging* (abreviação de *bootstrap aggregating*) [Breiman 1996] é um método de geração múltiplas versões de um classificador que são combinadas para a obtenção de um classificador agregado. O processo adotado pelo método de *bagging* é semelhante ao do método de *boosting*, porém de acordo com [Witten e Frank 2005], diferente do que ocorre no segundo método, no *bagging*, os diferentes modelos de classificadores recebem o mesmo peso na geração de uma predição.

No método de *bagging* quando a predição do classificador deve ser numérica, é feita uma média sobre os resultados de todos os modelos de classificadores. Por outro lado, se a predição deve ser uma classe, é feita uma votação e a classe com maior pontuação será escolhida para representar o padrão [Breiman 1996].

### 3.8. LogitBoost

O método LogitBoost [Friedman et al. 1998] é uma versão estatística do método de *boosting* e, segundo [Witten e Frank 2005], possui algumas semelhanças com o método Adaboost, porém ele otimiza a probabilidade de ocorrência de uma classe, enquanto o Adaboost otimiza uma função de custo exponencial. Em [Friedman et al. 1998] esse método é definido como um algoritmo para montagem de modelos aditivos de regressão logística.

### 3.9. OneR

O método OneR ou 1R (abreviação de 1-*rules*) [Holte 1993] pode ser considerado uma árvore de decisão com 1 nível, pois gera um conjunto de regras, uma para cada atributo do conjunto de dados e classifica uma amostra com base em um único atributo. O atributo escolhido é aquele, cuja regra produz o menor erro.

 ©2012 SBC — Soc. Bras. de Computação

### 3.10. Naive Bayes

O método naive Bayes [John e Langley 1995] é um classificador probabilístico simples baseado no teorema de Bayes. Esse método é denominado ingênuo (*naive*) porque ele assume que os atributos contribuem de forma independente para formar probabilidades estimativas da ocorrência de uma determinada classe do problema durante o processo de classificação. Logo, segundo [Witten e Frank 2005] a existência de redundância nos atributos pode distorcer o processo de aprendizagem.

## 4. Base de dados e configurações

Os experimentos foram realizados com a base de dados pública *WEBSPAM-UK2006 collection*[4]. Ela é composta por 77,9 milhões de páginas *web* hospedadas em 11.000 *hosts*. Essa base foi utilizada no *Web Spam Challenge Track I* e *II*[5], que trata-se de uma competição de técnicas de detecção de *web spam*.

Assim como empregado nos campeonatos, nos experimentos realizados foram utilizados 3 conjuntos de 8.487 vetores de características pré-computados. Cada conjunto é composto por 1.978 *hosts* rotulados como *spam* e 6.509 *hosts* rotulados como *ham* (não-spam). Essas informações foram extraídas da coleção de dados e fornecidas pelos organizadores do evento aos times participantes. O primeiro conjunto de vetores de características é composto por 96 características baseadas no conteúdo [Castillo et al. 2007], o segundo é composto por 41 características baseadas nos *links* [Becchetti et al. 2006] e o terceiro é composto por 138 características baseadas nos *links* transformados [Castillo et al. 2007], que são simples combinações ou operações logarítmicas sobre as características baseadas em *links*.

Para avaliar o desempenho de cada classificador foi usada uma validação por subamostragem aleatória, também conhecida como validação cruzada de Monte Carlo [Shao 1993]. Essa método de validação foi escolhido pois dá mais liberdade na seleção dos subconjuntos de treinamento e teste, já que diferente do método de validação k-folds, permite que sejam feitas quantas repetições forem desejadas, usando qualquer porcentagem de dados para treinamento e teste. Então, foram feitas 10 simulações com cada classificador e em cada uma, 80% dos dados foram usados para treinamento e 20% para teste. Eles foram selecionados aleatoriamente e com reposição a cada simulação. Ao final de todas as simulações foi calculada a média e o desvio padrão dos resultados. Para avaliar e comparar os resultados dos classificadores foram utilizadas as seguintes medidas de desempenho amplamente empregadas na literatura:

- *Sensitividade (recall):* proporção de padrões da classe positiva (*spam*) identificada corretamente.
- *Precisão (precision):* porcentagem de padrões classificados como pertencentes a classe positiva e que realmente pertencem a classe positiva.
- *F-medida:* média harmônica entre precisão e sensitividade, dada por:

$$F - medida = 2 \times \left( \frac{precisao \times sensitividade}{precisao + sensitividade} \right).$$

[4] *Yahoo! Research: "Web Spam Collections"*. Disponível em: `http://barcelona.research.yahoo.net/webspam/datasets/`.

[5] *Web Spam Challenge:* `http://webspam.lip6.fr/`

 ©2012 SBC — Soc. Bras. de Computação

### 4.1. Configurações

Para tornar os resultados completamente reprodutíveis, são apresentadas nessa seção as configurações adotadas para cada classificador.

#### 4.1.1. Redes neurais artificiais

Neste trabalho, foram avaliadas as seguintes RNAs: MLP treinada com o algoritmo *backpropagation* e método do gradiente descendente (MLP-GD) e MLP treinada com o método de Levenberg-Marquardt (MLP-LM).

Todas as redes MLP implementadas usam uma única camada intermediária e possuem um neurônio na camada de saída. Além disso, adotou-se uma função de ativação linear para o neurônio da camada de saída e uma função de ativação do tipo tangente hiperbólica para os neurônios da camada intermediária. Dessa forma, os pesos e o bias da rede foram inicializados com valores aleatórios entre $1$ e $-1$ e os dados usados para a classificação foram normalizados para o intervalo $[-1, 1]$, por meio da equação $x = 2 \times \frac{x - x_{min}}{x_{max} - x_{min}} - 1$, sendo $x$ a matriz com todos os vetores de características e $x_{min}$ e $x_{max}$ o menor e o maior valor da matriz $x$, respectivamente. Além disso, em todas as simulações com as RNAs MLP os critérios de parada adotados foram: número de épocas maior que um limiar $\theta$ , erro quadrático médio (EQM) do conjunto de treino menor que um limiar $\gamma$ ou aumento do EQM do conjunto de validação (verificado a cada 10 épocas).

Os parâmetros de cada RNA foram empiricamente calibrados e são apresentados na Tabela 1.

**Tabela 1. Parâmetros das RNAs.**

| Parâmetro | MLP-GD | MLP-LM |
|---|---|---|
| Limite máximo de iterações $\theta$ | 10000 | 500 |
| Limite mínimo do EQM $\gamma$ | 0.001 | 0.001 |
| Passo de aprendizagem $\alpha$ | 0.005 | 0.001 |
| Número de neurônios na camada intermediária | 100 | 50 |

#### 4.1.2. Máquinas de vetores de suporte

O SVM foi implementado utilizando a biblioteca LIBSVM [Chang e Lin 2011] disponível para a ferramenta MATLAB. Foram feitas simulações com as funções de *kernel* linear, RBF, sigmoidal e polinomial. A técnica de *grid search* foi empregada para a definição dos parâmetros. Porém, nas SVMs com *kernel* polinomial e sigmoidal, que possuem um maior número de parâmetros a serem definidos, optou-se por realizar a *grid search* apenas sobre os parâmetros $C$ e $\gamma$, devido ao excessivo custo computacional. Neste caso, adotou-se os valores padrões da LIBSVM para os demais parâmetros.

A *grid search* foi realizada com um conjunto de treino (80% dos dados) e teste (20% dos dados), escolhidos aleatoriamente para cada configuração de classificação. Depois de executado, os melhores parâmetros foram escolhidos e usados para realizar os experimentos com o SVM. Nesses experimentos, todos os tipos de *kernel* foram avaliados. Porém, optou-se por apresentar apenas os resultados obtidos com o *kernel* RBF, uma vez que, este obteve o melhor desempenho. A Tabela 2 apresenta os parâmetros usados nas simulações com o método SVM.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 2. Parâmetros usados no método SVM com *kernel* RBF, obtidos por *grid search*.**

| Tipos de vetores de características | Classes balanceadas? | $C$ | $\gamma$ |
|---|---|---|---|
| Conteúdo | Sim | $2^{14}$ | $2^{3}$ |
| Conteúdo | Não | $2^{15}$ | $2^{3}$ |
| *Links* | Sim | $2^{15}$ | $2^{-1}$ |
| *Links* | Não | $2^{15}$ | $2^{-1}$ |
| *Links* Transformados | Sim | $2^{5}$ | $2^{-5}$ |
| *Links* Transformados | Não | $2^{10}$ | $2^{-9}$ |
| *Links* + conteúdo | Sim | $2^{14}$ | $2^{-4}$ |
| *Links* + conteúdo | Não | $2^{15}$ | $2^{-2}$ |

### 4.1.3. Demais métodos

Os demais classificadores foram implementados usando a ferramenta WEKA [Hall et al. 2009]. Os métodos AdaBoost e *bagging* foram treinados com 100 iterações, sendo que ambos empregam agregação de múltiplas versões do método C4.5. Para os demais métodos foram empregados os parâmetros padrões da ferramenta utilizada.

## 5. Resultados

Nessa seção, são apresentados os resultados da detecção automática de *spam hosts* obtidos pelos métodos de aprendizado de máquina considerados os melhores atualmente disponíveis [Wu et al. 2008]. Para cada método e conjunto de vetores de características, foram feitas simulações usando classes desbalanceadas, conforme originalmente fornecido na competição de *web spam*, sendo 6.509 *hosts* (76,6%) da classe *ham* e 1.978 (23,4%) da classe *spam*. Em seguida, para avaliar se o desbalanceamento dos dados interfere na acurácia dos métodos, também foram realizados experimentos usando o mesmo número de amostras em cada classe. Neste caso, ambas as classes passaram a ter 1.978 representantes cada, aleatoriamente selecionados.

As Tabelas 3 e 4 apresentam os resultados obtidos por cada método de classificação explorando as características baseadas no conteúdo, nos *links*, nos *links* transformados e na combinação de *links* com o conteúdo. Os resultados estão ordenados pelo valor da F-medida. Os valores em negrito indicam os melhores resultados para cada tipo de atributo. Os valores em negrito precedidos pelo símbolo "*" indicam os melhores resultados considerando todas as simulações.

De maneira geral, os métodos avaliados obtiveram bons resultados, pois foram capazes de detectar com alta precisão uma quantidade expressiva de *spam hosts*, independentemente da característica empregada. Além disso, os resultados indicam que entre as técnicas avaliadas, o método de *bagging*, na média, apresentou os melhores resultados. Ele foi capaz de detectar em média 83,5% dos *spam hosts* com uma precisão média de 82,9%, independente do tipo de atributo utilizado. Por outro lado, o classificador SVM, em média, apresentou o pior desempenho. Outro método que merece destaque é o método naive Bayes que diferente dos outros métodos, nos experimentos com classes desbalanceadas, obteve uma ótima taxa de sensitividade, próxima a 100%. Porém esse resultado foi obtido a custa de muitos falsos positivos e por isso, a taxa de precisão foi baixa.

Os resultados também indicam que os classificadores obtiveram melhor desempenho quando foram treinados usando classes balanceadas. Portanto, verifica-se que, em ge-

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 3. Resultados obtidos na detecção de *spam hosts* usando classes desbalanceadas.**

| | Sensitividade | Precisão | F-medida |
|---|---|---|---|
| | Atributos extraídos do conteúdo | | |
| Bagging | 68.7±2.3 | **84.4**±1.9 | **0.757**±0.014 |
| Floresta aleatória | 65.7±4.4 | 83.4±3.7 | 0.734±0.024 |
| MLP-LM | 69.3±4.2 | 77.6±4.6 | 0.731±0.032 |
| AdaBoost | 66.6±3.2 | 78.4±2.3 | 0.720±0.024 |
| IBK | 64.6±1.3 | 72.8±2.0 | 0.685±0.011 |
| C4.5 | 67.7±4.6 | 68.0±0.6 | 0.678±0.024 |
| MLP-GD | 57.0±4.6 | 77.5±2.7 | 0.656±0.039 |
| LogitBoost | 54.2±3.9 | 71.6±3.2 | 0.616±0.023 |
| SVM | 36.0±2.4 | 76.2±2.8 | 0.488±0.023 |
| OneR | 38.3±2.4 | 65.0±2.7 | 0.482±0.023 |
| Naive Bayes | ***97.4**±1.0 | 25.4±1.3 | 0.402±0.015 |
| | Atributos extraídos dos *links* | | |
| Bagging | 79.2±1.6 | **77.2**±2.0 | **0.781**±0.009 |
| Floresta aleatória | 76.5±7.9 | 77.0±3.7 | 0.764±0.027 |
| AdaBoost | 71.8±1.8 | 74.4±2.2 | 0.731±0.017 |
| C4.5 | 73.2±2.1 | 72.3±1.4 | 0.727±0.011 |
| MLP-LM | 70.8±6.6 | 74.1±3.7 | 0.723±0.049 |
| LogitBoost | 71.8±4.0 | 71.4±2.1 | 0.715±0.009 |
| MLP-GD | 61.6±3.1 | 75.3±2.9 | 0.677±0.026 |
| IBK | 67.8±2.8 | 64.2±2.4 | 0.659±0.023 |
| Naive Bayes | **94.6**±1.3 | 46.5±1.2 | 0.624±0.012 |
| SVM | 47.1±2.5 | 62.8±1.9 | 0.538±0.021 |
| OneR | 49.4±3.5 | 54.7±1.9 | 0.518±0.023 |
| | Atributos extraídos dos links transformados | | |
| Bagging | 78.9±1.2 | **77.6**±2.1 | **0.782**±0.014 |
| Floresta aleatória | 76.5±3.1 | 75.8±4.1 | 0.760±0.012 |
| MLP-LM | 74.9±3.9 | 76.1±2.1 | 0.754±0.027 |
| SVM | 76.6±2.3 | 73.9±1.6 | 0.752±0.012 |
| MLP-GD | 75.2±2.2 | 73.9±3.1 | 0.745±0.014 |
| AdaBoost | 72.6±3.0 | 74.9±1.9 | 0.737±0.019 |
| LogitBoost | 70.2±4.9 | 72.3±3.2 | 0.711±0.019 |
| C4.5 | 73.5±1.8 | 68.5±2.2 | 0.709±0.016 |
| IBK | 67.5±2.4 | 67.3±3.5 | 0.673±0.011 |
| OneR | 64.3±2.1 | 63.2±2.8 | 0.637±0.018 |
| Naive Bayes | **96.5**±1.0 | 26.2±0.3 | 0.412±0.004 |
| | Combinação dos atributos extraídos do conteúdo e dos links | | |
| AdaBoost | 80.4±1.9 | ***86.3**±1.2 | ***0.832**±0.014 |
| MLP-LM | 81.7±2.0 | 84.4±1.9 | 0.830±0.008 |
| Bagging | 81.1±1.8 | 84.0±1.7 | 0.825±0.016 |
| Floresta aleatória | 74.4±2.5 | 85.0±2.4 | 0.793±0.006 |
| MLP-GD | 74.2±3.2 | 79.5±2.5 | 0.767±0.016 |
| C4.5 | 75.9±2.3 | 73.8±0.9 | 0.748±0.014 |
| LogitBoost | 70.2±2.6 | 77.7±2.3 | 0.737±0.019 |
| IBK | 68.7±2.3 | 72.5±2.0 | 0.705±0.015 |
| Naive Bayes | **95.3**±1.0 | 45.9±0.8 | 0.620±0.008 |
| SVM | 52.4±2.0 | 69.6±2.5 | 0.598±0.018 |
| OneR | 50.1±2.4 | 54.4±2.2 | 0.521±0.018 |

ral, os métodos de classificação analisados tendem a favorecer a classe com maior número de representantes apresentados na fase de treinamento. Isso pode ser confirmado, pois os valores da sensitividade obtidos pelos métodos usando classes desbalanceadas foram, em geral, mais baixos, o que mostra que os classificadores erraram mais na identificação dos padrões pertencentes à classe *spam*.

Note que, o melhor resultado médio, considerando todas as técnicas, foi obtido no cenário em que foram utilizados os vetores de características baseados em *links* transformados (Tabela 4). Porém, o maior valor de F-medida foi obtido pelo método *AdaBoost*

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 4. Resultados obtidos na detecção de *spam hosts* usando classes balanceadas.**

| | Sensitividade | Precisão | F-medida |
|---|---|---|---|
| | Atributos extraídos do conteúdo | | |
| MLP-LM | **86.5**±2.0 | ***89.1**±2.4 | **0.877**±0.017 |
| Bagging | 83.5±2.3 | 86.1±1.3 | 0.848±0.013 |
| MLP-GD | 82.9±2.0 | 86.1±2.4 | 0.845±0.015 |
| Floresta aleatória | 81.8±3.3 | 83.4±2.3 | 0.825±0.015 |
| AdaBoost | 81.8±2.5 | 83.3±1.6 | 0.825±0.014 |
| IBK | 77.0±2.6 | 81.4±1.5 | 0.791±0.017 |
| C4.5 | 79.2±2.5 | 78.3±1.6 | 0.787±0.012 |
| LogitBoost | 77.0±3.2 | 78.9±1.1 | 0.779±0.015 |
| Naive Bayes | 83.6±17.7 | 62.8±10.2 | 0.694±0.039 |
| SVM | 60.6±2.7 | 76.7±1.5 | 0.677±0.019 |
| OneR | 61.6±1.0 | 68.3±2.0 | 0.648±0.013 |
| | Atributos extraídos dos *links* | | |
| Bagging | 91.7±1.9 | 84.9±1.4 | **0.882**±0.013 |
| Floresta aleatória | 88.8±2.5 | **87.1**±1.6 | 0.879±0.010 |
| MLP-LM | 92.1±3.8 | 82.3±2.6 | 0.868±0.019 |
| AdaBoost | 87.7±1.7 | 83.9±0.8 | 0.858±0.009 |
| MLP-GD | 90.2±2.6 | 80.0±2.9 | 0.848±0.019 |
| LogitBoost | 88.0±3.7 | 80.4±1.0 | 0.840±0.018 |
| C4.5 | 85.4±2.2 | 82.3±1.4 | 0.838±0.011 |
| Naive Bayes | **94.8**±0.9 | 73.7±2.0 | 0.829±0.014 |
| OneR | 88.7±1.0 | 77.5±1.5 | 0.827±0.010 |
| IBK | 80.8±1.4 | 77.1±0.8 | 0.789±0.003 |
| SVM | 77.1±2.2 | 75.8±1.7 | 0.765±0.014 |
| | Atributos extraídos dos links transformados | | |
| Bagging | 91.3±1.6 | **86.2**±1.1 | **0.886**±0.011 |
| AdaBoost | 88.8±0.9 | 85.9±1.7 | 0.873±0.011 |
| MLP-GD | 89.6±2.2 | 85.9±2.0 | 0.877±0.019 |
| Floresta aleatória | 89.5±3.7 | 86.0±1.7 | 0.876±0.016 |
| MLP-LM | 87.5±4.0 | 85.8±3.3 | 0.866±0.027 |
| SVM | 88.5±1.7 | 84.4±1.3 | 0.864±0.007 |
| LogitBoost | 87.6±3.7 | 82.7±1.0 | 0.850±0.021 |
| C4.5 | 85.6±2.4 | 83.1±1.8 | 0.844±0.020 |
| OneR | 86.5±2.0 | 80.5±1.2 | 0.834±0.008 |
| IBK | 81.6±0.7 | 80.2±0.4 | 0.809±0.002 |
| Naive Bayes | ***96.8**±0.8 | 56.6±3.2 | 0.714±0.025 |
| | Combinação dos atributos extraídos do conteúdo e dos *links* | | |
| AdaBoost | 92.6±1.4 | **88.5**±0.9 | ***0.905**±0.005 |
| Bagging | **93.2**±0.8 | 87.8±1.0 | 0.904±0.005 |
| MLP-GD | 92.6±2.4 | 86.7±2.1 | 0.895±0.013 |
| Floresta aleatória | 92.1±2.1 | 86.9±1.2 | 0.894±0.008 |
| MLP-LM | 91.9±3.5 | 85.6±2.7 | 0.886±0.021 |
| LogitBoost | 89.9±1.3 | 84.3±1.2 | 0.870±0.005 |
| C4.5 | 86.9±2.2 | 85.1±0.9 | 0.860±0.009 |
| IBK | 84.7±2.7 | 82.4±1.0 | 0.835±0.012 |
| Naive Bayes | 92.7±6.5 | 75.2±3.5 | 0.828±0.016 |
| OneR | 88.8±1.7 | 77.5±1.7 | 0.827±0.012 |
| SVM | 71.4±1.7 | 78.7±2.3 | 0.749±0.015 |

usando a combinação de características baseadas no conteúdo da página e na relação dos *links*. Portanto, é importante observar que, a escolha dos atributos mais adequados varia de acordo com a técnica de aprendizagem de máquina escolhida.

Para facilitar a avaliação dos métodos analisados, é apresentado na Tabela 5 uma comparação entre os melhores resultados obtidos nesse trabalho e os disponíveis na literatura de *web spamming*. Para oferecer uma comparação justa, os métodos propostos na literatura foram treinados com a mesma base de dados, atributos e configurações usados nos métodos apresentados nesse trabalho. Foram usados exa-

 ©2012 SBC — Soc. Bras. de Computação

tamente os mesmos parâmetros descritos nos trabalhos propostos na literatura ou os parâmetros padrão da ferramenta ou biblioteca utilizada. Em resumo, os métodos de *bagging* de árvores de decisão [Castillo et al. 2007, Ntoulas et al. 2006] e *boosting* de árvores de decisão [Ntoulas et al. 2006] foram implementados usando a ferramenta WEKA. Já, as máquinas de vetores de suporte [Svore et al. 2007] foram implementadas usando a biblioteca LIBSVM na ferramenta MATLAB. Por outro lado, método de programação genética [Shengen et al. 2011] não foi reimplementado porque os autores adotaram a mesma base de dados usada nesse trabalho, logo, para a comparação foram usados os resultados originais fornecidos pelos autores.

**Tabela 5. Comparação entre os melhores resultados obtidos nesse trabalho e os resultados disponíveis na literatura**

| Classifiers | Precisão | Sensitividade | F-medida. |
|---|---|---|---|
| Resultados disponíveis na literatura | | | |
| Bagging [Castillo et al. 2007, Ntoulas et al. 2006] - conteúdo | 81.5 | 68.0 | 0.741 |
| Máquinas de vetores de Suporte [Svore et al. 2007] - conteúdo | 55.5 | **86.5** | 0.677 |
| Boosting [Ntoulas et al. 2006] - conteúdo | 78.2 | 68.2 | 0.728 |
| Programação genética [Shengen et al. 2011] - links | 69.8 | 76.3 | 0.726 |
| Programaçao genética [Shengen et al. 2011] - links transformados | 76.5 | 81.4 | 0.789 |
| Melhores resultados obtidos nesse trabalho | | | |
| *Bagging* - conteúdo+*links* | 84.0 | 81.1 | 0.825 |
| AdaBoost - conteúdo+*links* | **86.3** | 80.4 | **0.832** |
| *Bagging* - conteúdo+*links* (classes balanceadas) | 87.8 | **93.2** | 0.904 |
| AdaBoost - conteúdo+*links* (classes balanceadas) | **88.5** | 92.6 | **0.905** |

É importante observar na tabela 5 que os métodos *bagging* e *AdaBoost* obtiveram desempenho superior aos métodos propostos em outros trabalhos da literatura. Portanto, é conclusivo que técnicas de aprendizado de máquina podem ser empregadas com sucesso para auxiliar o processo de detecção automática de *spam hosts*.

## 6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma análise de desempenho dos métodos de classificação mais empregados na literatura para auxiliar na tarefa de detecção automática de *spam hosts*.

Os resultados dos experimentos mostraram que todas as técnicas avaliadas oferecem bons resultados, independente do tipo de atributo utilizado. Isso demonstra que além de eficientes, os métodos apresentados possuem boa capacidade de generalização.

Dentre os métodos avaliados, as técnicas de agregação de métodos classificadores baseados em árvores, tais como *bagging* e *AdaBoost*, obtiveram os melhores desempenhos, demonstrando serem adequadas para auxiliar na detecção de *spam hosts*.

Com relação aos vetores de características, os melhores resultados foram obtidos quando os métodos classificadores utilizaram a combinação de informações extraídas do conteúdo das páginas e relação de *links*. É importante destacar também que, as técnicas avaliadas demonstraram ser mais eficientes quando treinadas com número igual de representantes em cada classe, pois ficou evidente que a classificação torna-se tendenciosa para a classe com maior número de amostras usada na etapa de treinamento.

Trabalhos futuros compreendem o estudo de formas de adaptar os métodos mais promissores para otimizar seu desempenho, a análise das características para verificar a viabilidade do emprego de técnicas de seleção de atributos, além da proposição de novos tipos de atributos que possam aumentar a capacidade de predição dos algoritmos.

 ©2012 SBC — Soc. Bras. de Computação

## Agradecimentos

Os autores são gratos a Capes, Fapesp e CNPq pelo apoio financeiro.

## Referências

Aha, D. W., Kibler, D., e Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., e Baeza-Yates, R. (2006). Using rank propagation and probabilistic counting for link-based spam detection. In *Proc. of the WebKDD'06*, Philadelphia,USA.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford Press, Oxford.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Castillo, C., Donato, D., e Gionis, A. (2007). Know your neighbors: Web spam detection using the web topology. In *Proc. of the 30th SIGIR*, pages 423–430, Amsterdam, The Netherlands.

Chang, C.-C. e Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2:27:1–27:27.

Cortes, C. e Vapnik, V. N. (1995). Support-vector networks. In *Machine Learning*, pages 273–297.

Freund, Y. e Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. of the 13th ICML*, pages 148–156, Bari, Italy. Morgan Kaufmann.

Friedman, J., Hastie, T., e Tibshirani, R. (1998). Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407.

Gyongyi, Z. e Garcia-Molina, H. (2005). Spam: It's not just for inboxes anymore. *Computer*, 38(10):28–34.

Gyongyi, Z., Garcia-Molina, H., e Pedersen, J. (2004). Combating web spam with trustrank. In *Proc. of the 30th VLDB*, pages 576–587, Toronto, Canada.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., e Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New York, NY, USA, 2th edition.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90.

Hsu, C.-W., Chang, C.-C., e Lin, C.-J. (2003). A practical guide to support vector classification. Technical report, National Taiwan University.

Jayanthi, S. K. e Sasikala, S. (2012). WESPACT: Detection of web spamdexing with decision trees in GA perspective. In *Proc. of the PRIME'12*, pages 381–386.

John, G. H. e Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proc. of the 11th UAI)*, pages 338–345, Montreal, Quebec;, Canada.

 ©2012 SBC — Soc. Bras. de Computação

John, J. P., Yu, F., Xie, Y., Krishnamurthy, A., e Abadi, M. (2011). deSEO: combating search-result poisoning. In *Proc. of the 20th SEC*, pages 20–20, Berkeley, CA, USA.

Largillier, T. e Peyronnet, S. (2012). Webspam demotion: Low complexity node aggregation methods. *Neurocomputing*, 76(1):105–113.

Ledford, J. L. (2009). *Search Engine Optimization Bible*. Wiley Publishing, Indianapolis, Indiana, USA, 2th edition.

Liu, Y., Chen, F., Kong, W., Yu, H., Zhang, M., Ma, S., e Ru, L. (2012). Identifying web spam with the wisdom of the crowds. *ACM Trans. on the Web*, 6(1):2:1–2:30.

Ntoulas, A., Najork, M., Manasse, M., e Fetterly, D. (2006). Detecting spam web pages through content analysis. In *Proc. of the WWW*, pages 83–92, Edinburgh, Scotland.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann, San Mateo, CA, USA, 1th edition.

Rungsawang, A., Taweesiriwate, A., e Manaskasemsak, B. (2011). Spam host detection using ant colony optimization. In *IT Convergence and Services*, volume 107 of *Lecture Notes in Electrical Engineering*, pages 13–21. Springer Netherlands.

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):486–494.

Shengen, L., Xiaofei, N., Peiqi, L., e Lin, W. (2011). Generating new features using genetic programming to detect link spam. In *Proc. of the ICICTA'11*, pages 135–138, Shenzhen, China.

Silva, R. M., Almeida, T. A., e Yamakami, A. (2012a). Artificial neural networks for content-based web spam detection. In *Proc. of the 14th International Conference on Artificial Intelligence (ICAI'12)*, pages 1–7, Las Vegas, NV, USA.

Silva, R. M., Almeida, T. A., e Yamakami, A. (2012b). Redes neurais artificiais para detecção de web spams. In *Anais do 8th Simpósio Brasileiro de Sistemas de Informação (SBSI'12)*, pages 636–641, São Paulo, Brazil.

Svore, K. M., Wu, Q., e Burges, C. J. (2007). Improving web spam classification using rank-time features. In *Proc. of the 3rd AIRWeb*, pages 9–16, Banff, Alberta, Canada.

Taweesiriwate, A., Manaskasemsak, B., e Rungsawang, A. (2012). Web spam detection using link-based ant colony optimization. In *Proc. of the 26th AINA*, pages 868–873.

Witten, I. H. e Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2th edition.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., e Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

 ©2012 SBC — Soc. Bras. de Computação

# Avaliação da Sensibilidade de Preditores de Suavização Exponencial na Detecção de Ataques de Inundação

**Nícolas R. e Silva[1], Ronaldo M. Salles[1]**

[1] Seção de Engenharia de Computação
Instituto Militar de Engenharia (IME) – Rio de Janeiro, RJ – Brazil

nicolasime@gmail.com, salles@ime.eb.br

***Abstract.*** *This paper analyzes the sensitivity of typical exponential smoothing predictors used to detect Distributed Denial of Service (DDoS) flooding attacks. We compare two predictors (EWMA and Holt-Winters) and evaluate their detection accuracy within different settings and scenarios. The performance is investigated in terms of false positive and false negative ratios. We insert attacks on real IP traces (MAWILab) and on real traffic samples from RNP's WAN backbone to perform simulations with different levels of flooding. Simulations show that to optimize the parameters of predictors provide better results.*

***Resumo.*** *Este artigo analisa a sensibilidade de preditores de suavização exponencial usados para detectar ataques distribuídos de negação de serviço (DDoS). Comparou-se a capacidade de detecção de dois preditores (EWMA e Holt-Winters) com diferentes configurações e cenários. Foi verificado o desempenho através das taxas de falsos positivos e falsos negativos gerados. Foi inserido ataques em traces reais do MAWILab e em amostras reais de tráfego de backbone da RNP com o intuito de realizar simulações de ataques com diferentes volumes de inundação. As simulações mostram que a otimização de parâmetros dos preditores trazem melhores resultados.*

## 1. Introdução

Cada vez mais, ataques distribuídos de negação de serviço (DDoS) se tornam mais sofisticados e difíceis de detectar. Por isso, desenvolver métodos para combatê-los é uma tarefa desafiadora. Embora existam estudos desde o ano de 2000, ainda não há uma resposta definitiva para a solução deste problema.

Apesar dos avanços na detecção deste tipo de anomalia, especialmente sobre *backbones* de alta velocidade, muitas das abordagens apresentam um custo computacional elevado, requerem mudanças na infraestrutura ou mesmo apresentam baixa sensibilidade a mudanças de comportamento. Por isso, a demanda por métodos mais eficientes justifica o desenvolvimento de estudos nessa área.

Soluções baseadas na correlação de dados e agregação do tráfego em fluxos, parecem ser a tendência para soluções futuras [Feitosa et al, 2008], possibilitando uma detecção antecipada. Essa antecipação pode ser alcançada através da distribuição de detectores em pontos que, embora não sejam objetivos do ataque, constituem vias por onde passam os fluxos destinados à vítima. A eficiência e viabilidade desse tipo de solução depende de uma abordagem colaborativa, onde o esforço conjunto de instituições, através do compartilhamento de dados, pode proteger um número muito

 ©2012 SBC — Soc. Bras. de Computação

maior de redes e fortalecer o sistema como um todo.

Neste caso, a escolha de métricas mais sensíveis parece ser a mais adequada, já que a concentração esperada de tráfego malicioso é relativamente baixa nesses pontos.

Neste artigo, foi comparado o desempenho de dois estimadores de suavização exponencial bastante empregados, *Exponential Weighted Moving Average* (EWMA) e *Holt-Winters* (HW), aplicados à série temporal formada por medidas de entropia de Shannon, considerando-se diferentes configurações, para a detecção de ataques de inundação. Foi verificado o desempenho através dos índices de falsos positivos (FPOS) e falsos negativos (FNEG) observados em cada cenário, com a inserção de pacotes maliciosos em diferentes níveis de concentração.

Uma série temporal de predição, gerada com o auxílio de estimadores, pode servir como referência para o cálculo das margens de segurança que seriam ultrapassadas após o início do ataque. Normalmente, utiliza-se um fator multiplicativo que determina a amplitude dessas margens. Neste trabalho, verificou-se, também, a influência desse fator multiplicativo, que determina a amplitude dos limites de segurança, na sensibilidade destes estimadores.

No presente trabalho, pode-se destacar as seguintes contribuições:

- proposta de uma metodologia simples para inserção de ataques de inundação, com volume ajustável, para a avaliação de técnicas de detecção baseadas na identificação de anomalias, sem a necessidade de alterar o trace original;
- uma análise da sensibilidade de estimadores de suavização exponencial através da inserção de ataques com diferentes volumes;
- uma análise do desempenho destes estimadores com o emprego de parâmetros otimizados.

O restante do artigo está organizado da seguinte forma: na Seção 2, são discutidos os trabalhos relacionados. Os métodos empregados no corrente trabalho são apresentados na Seção 3. Na Seção 4, é destacada a arquitetura do sistema utilizada. A Seção 5, por sua vez, traz os resultados obtidos através de simulações. Finalmente, na Seção 6, são apresentadas as conclusões.

## 2. Trabalhos Relacionados

De maneira geral, o combate aos ataques DDoS requer a execução de 3 etapas: (i) identificar a ocorrência de um ataque; (ii) rastrear a origem dos pacotes maliciosos; e (iii) acionar contramedidas que contribuam para a mitigação ou eliminação dos danos causados pelo ataque, tais como filtragem e bloqueio de pacotes [Castelúcio et al 2009].

No que diz respeito à detecção, são propostos diferentes mecanismos baseados em *wavelet* [Kaur et al, 2010], entropia [Lakhina et al, 2005], [Lucena e Moura, 2008], tabela de roteamento [Park e Lee, 2000], *defense by ofense* [Walfish et al, 2010], caracterização do tráfego [Feng e Liu, 2009], marcação de pacotes [Law et al, 2002], que podem adotar uma abordagem *single-link* [Lucena e Moura, 2008], [Demir e Khan, 2010] ou *network-wide* [Chen e Hwang, 2006]. Algumas arquiteturas independem de base histórica [Lin e Uddin, 2005], enquanto que outras têm seus parâmetros adaptados de acordo com uma *baseline* para melhor se ajustar a fatores sazonais [Kline et al, 2008].

 ©2012 SBC — Soc. Bras. de Computação

Diversos pesquisadores sugerem que a detecção deste tipo de anomalia seja realizada junto à vítima, e que os alertas, bem como o rastreamento e as contramedidas, sejam realizados no sentido contrário do fluxo, como ocorre nos sistemas COSSACK [Papadopoulos et al, 2003] e DefCOM [Mirkovic e Reiher, 2005]. Neste caso, o intuito dos gerentes de rede é proteger a sua própria rede. Entretanto, mesmo com a detecção, o ataque já pode ter comprometido a vítima de alguma maneira e a execução de qualquer medida torna-se mais difícil devido à inundação de dados. Sendo assim, é altamente desejável que a detecção de ataques DDoS ocorra o mais rápido possível, antes que a inundação torne-se generalizada [Chen et al, 2007].

Para conseguir uma detecção antecipada alguns autores recomendam realizar a inspeção de pacotes nos roteadores de borda de Sistemas Autônomos, já que constituem pontos de concentração de fluxo [Park e Lee, 2000], [Lin e Uddin, 2005].

O percentual de pacotes maliciosos que passa por estas interfaces costuma ser relativamente baixo, de forma que as mudanças de comportamento são mais brandas e, por isso, mais difíceis de detectar. Métricas mais sensíveis, como a entropia de Shannon, aplicadas nas distribuições estatísticas dos endereços IP ou dos tamanhos dos pacotes, têm sido consideradas eficazes na detecção de tráfego anormal [Xiang et al, 2011].

Neste artigo, foram testadas configurações otimizadas de dois estimadores a fim de verificar se há uma melhora significativa na precisão da detecção, particularmente em cenários cujo tráfego de ataques DDoS seja mais ameno.

## 3. Métricas Empregadas para Detecção de Ataques

### 3.1. Entropia

Em Shannon (1948), foi desenvolvida uma teoria da comunicação com o intuito de tornar melhores os projetos de sistemas de telecomunicações. Trata-se de uma medida da informação contida numa mensagem que Shannon chamou de entropia, e pode ser definida como:

$$E_S = -\sum_{i=0}^{N} p_i \log_2(p_i) \tag{1}$$

Onde $N$ é o número de diferentes ocorrências no espaço amostral e $p_i$ é a probabilidade associada a cada ocorrência $i$. O resultado varia entre *zero* e $\log_2 N$, onde *zero* indica concentração máxima na distribuição medida, quando ocorre um único valor de $i$, e $\log_2 N$ indica máxima dispersão na distribuição medida, quando todas as ocorrências têm a mesma probabilidade de ocorrência. No corrente trabalho, esta medida foi aplicada ao conjunto de endereços IP em cada intervalo considerado, a fim de identificar mudanças bruscas de padrão com o passar do tempo.

### 3.2. Estimadores

Uma série temporal de predição, gerada com o auxílio de estimadores, pode servir como referência para o cálculo das margens de segurança que seriam ultrapassadas após o início do ataque. No presente trabalho foram aplicados dois estimadores à série temporal dos valores de entropia referente ao tráfego observado.

*1) Exponentially weighted moving average (EWMA)* - Como o próprio nome diz,

 ©2012 SBC — Soc. Bras. de Computação

trata-se de um método que faz o cálculo da média móvel exponencialmente ponderada, também conhecido como suavização exponencial simples (*simple exponential smoothing*). Pode ser expressa da seguinte maneira:

$$x_{t+1} = \alpha X_t + (1-\alpha) x_t \tag{2}$$

Onde $x_t$ representa a média estimada no instante $t$ e $X_t$ é o valor atual real. O valor de $\alpha$ reflete o peso conferido ao valor mais recente, e assume valores entre 0 e 1.

Não é difícil perceber que, a cada iteração, as estimativas mais antigas perdem a influência no resultado calculado de maneira exponencial, de forma que quanto mais recente o valor, maior o peso creditado a ele. Dessa forma, o valor estimado representa uma média ponderada cujos valores mais recentes têm maior peso. Por conta disso, o traçado da série de estimativa gerada se assemelha ao traçado obtido com os valores reais. Quanto menor o valor de $\alpha$, maior a suavidade no traçado da série.

*2) Holt-Winters (HW)* - Trata-se de um método de suavização exponencial tripla (*triple exponential smoothing*), que costuma ser empregado quando os dados da série apresentam tendência e sazonalidade. Para lidar com essas duas características, são utilizados mais dois parâmetros além daquele empregado na suavização exponencial simples em três equações que formam um conjunto resultante denominado *Holt-Winters* (HW). Existem dois modelos principais de HW, aditivo e multiplicativo, que tratam a sazonalidade de maneiras ligeiramente distintas [Kalekar et al, 2004]. No corrente trabalho, foi utilizado o modelo aditivo, que é calculado a partir das seguintes expressões:

$$a_t = \alpha (X_t - c_{t-m}) + (1-\alpha)(a_{t-1} + b_{t-1}) \tag{3}$$

$$b_t = \beta (a_t - a_{t-1}) + (1-\beta) b_{t-1} \tag{4}$$

$$c_t = \gamma (X_t - a_t) + (1-\gamma) c_{t-m} \tag{5}$$

$$x_{t+1} = a_t + b_t + c_{t+1-m} \tag{6}$$

Onde $x_t$ representa a média estimada no instante t e $X_t$ é o valor atual real. $a_t$ denota a componente residual, $b_t$, a componente de tendência de crescimento, $c_t$, a componente de periodicidade da série e $m$ representa o tamanho do período. Os parâmetros $\alpha$, $\beta$ e $\gamma$ refletem a importância conferida a cada componente.

## 4. Arquitetura do Sistema

A ocorrência de um DDoS causa uma mudança acentuada na distribuição de endereços IP de destino, de forma que a entropia desta informação sofre uma queda abrupta.

Sendo assim, para identificar esta anomalia, a arquitetura utilizada prevê a montagem de séries temporais contendo medidas de entropia obtidas nas distribuições estatísticas dos endereços IP de destino, extraídas a cada intervalo de tempo considerado. A partir dos valores reais de entropia, monta-se a série temporal de predição, gerada com o auxílio dos estimadores.

Após montar essas séries, as margens de segurança podem ser calculadas. Para

 ©2012 SBC — Soc. Bras. de Computação

isso foi utilizada uma expressão de suavização exponencial simples para o erro de estimativa.

$$e_t = \gamma |X_t - x_t| + (1-\gamma) e_{t-m} \tag{7}$$

Onde $e_t$ representa o erro de estimativa no instante $t$. A componente é atualizada a cada erro calculado, levando em conta os erros do período anterior.

De acordo com Brutlag (2000), baseando-se na teoria de distribuição estatística e em algumas suposições, este erro deve ser multiplicado por um valor de escala $\delta$ para poder compor as margens de segurança. Valores normalmente empregados para $\delta$ estão entre 2 e 3, de acordo com Ward et al (1998).

Sabe-se que a largura das margens de segurança influencia no índice de FPOS e FNEG, e que esta largura é diretamente proporcional ao valor atribuído a $\delta$. Sendo assim, verificou-se, também, os resultados obtidos com limites mais largos, que aqui chamamos de AMPLOS, e mais ajustados, que aqui chamamos de RIGOROSOS. No primeiro caso, atribuiu-se o valor 3 a $\delta$, enquanto que no segundo, atribuiu-se o valor 2.

Dessa forma, quando o valor real ultrapassa os limites, um alerta é emitido. Os limites superior e inferior podem ser calculados a partir das seguintes expressões:

$$lim_{Sup} = x_t + \delta . e_{t-m} \tag{8}$$

$$lim_{Inf} = x_t - \delta . e_{t-m} \tag{9}$$

Na Figura 1, pode-se observar dois exemplos de resultados obtidos com a utilização da arquitetura empregada. A linha azul representa os valores de entropia, calculados a partir dos dados extraídos dos traces e ataques inseridos. A linha laranja, por sua vez, representa as estimativas de HW, calculadas a partir dos valores reais, enquanto que as linhas vermelha e ocre indicam os limites inferior e superior, respectivamente.

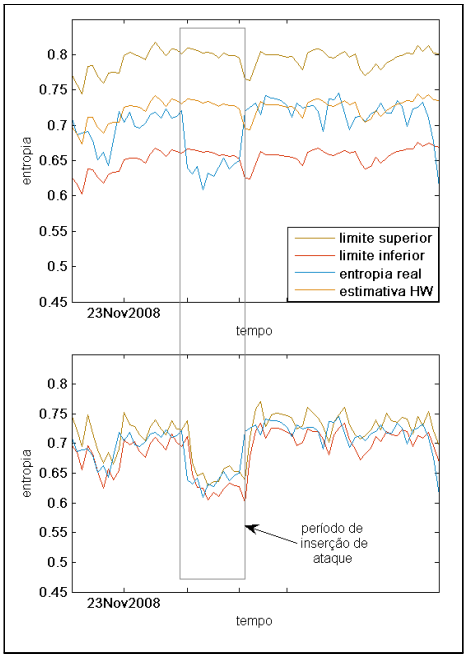



**Figura 1. Exemplos de resultados com estimativas de Holt-Winters**

 ©2012 SBC — Soc. Bras. de Computação

Destaca-se nos gráficos da Figura 1 um período de inserção de ataque que foi identificado em ambos os casos. Quando o valor real ultrapassa o limite inferior, emite-se um alerta, que pode caracterizar a ocorrência de um FPOS, caso não haja um ataque, ou a detecção do DDoS, caso haja.

Pode-se perceber, também, alguns pontos com violações do limite inferior que ocorreram fora do período de ataque, particularmente no segundo gráfico desta Figura, quando foram empregados parâmetros otimizados. Estes eventos caracterizam a ocorrência de FPOS.

# 5. SIMULAÇÕES

Foram realizadas diversas simulações com o intuito de comparar os índices de FPOS e FNEG obtidos com cada configuração de parâmetros. Duas configurações foram empregadas: na primeira, aqui chamada de PADRÃO, foram utilizados os valores dos parâmetros $\alpha$, $\beta$ e $\gamma$ baseados em Brutlag (2000); na segunda, aqui chamada de ÓTIMA, utilizou-se valores que garantiam o menor erro de estimativa sem a presença de ataques. Estes valores foram obtidos empiricamente, como descrito na Seção 5.2.

Para obter os resultados necessários, foi implementado em C++ um programa capaz de realizar todos os cálculos referentes à montagem das séries temporais, desde a agregação dos endereços contidos no mesmo intervalo considerado até a contabilização de FPOS e FNEG.

## 5.1. Bases de dados

Foram utilizadas as informações de traces reais (MAWILab), coletados pelo *WIDE Project* e disponibilizados em: http://www.fukuda-lab.org/mawilab/, e de amostras reais de tráfego oriundas do backbone da RNP, gentilmente cedidas pelo Centro de Engenharia e Operações da RNP, mediante solicitação. Na Tabela 1 são apresentadas algumas características desses traces, que também podem influenciar nos resultados finais.

**Tabela 1. Características dos traces utilizados**

| | | **Base de dados** | |
|---|---|---|---|
| | | **RNP** | **MAWILab** |
| **Período de observação** | **Datas** | De 20Nov08, 0:00h, a 27Nov08, 0:00h | 07Jun10, de 14:00h às 14:15h e 06Dez10, de 14:00h às 14:15h |
| | **Tempo por dia** | 24 horas | 15 minutos |
| | **Total** | 7 dias (ininterruptos) | 30 min (07Jun10 e 06Dez10) |
| **Intervalo** | | 5 minutos | 1 segundo |
| **Capacidade do enlace** | | 2.5 Gbps | 150 Mbps |
| **Amostragem** | | 1:100 | Sem amostragem |
| **Formato** | | NetFlow versão 5 | .dump |
| **Taxa média real** | | 30 Kpps | 48 Kpps |

Devido ao pequeno período de observação disponibilizado nos traces do MAWILab, 15 minutos por dia, foi necessário realizar o estudo com séries temporais de 1 segundo. Além disso, verificou-se uma sazonalidade nos traces do RNP, não presente

 ©2012 SBC — Soc. Bras. de Computação

nos traces do MAWILab. A partir desses traces, foram montadas séries temporais das entropias dos endereços de destino em intervalos fixos de 5 minutos, para RNP, e de 1 segundo, para MAWILab.

### 5.2. Obtenção de Parâmetros Otimizados

Procurou-se pelos parâmetros dos estimadores EWMA e Holt-Winters que fornecessem os menores somatórios de erros de estimativa num período de tráfego livre de ataques. Para fins de estudo, considerou-se que os traces estudados estariam livres de ataques.

A partir desses traces, foram montadas séries temporais das entropias dos endereços de origem e destino para intervalos fixos de 5 minutos, para RNP e de 1 segundo, para MAWILab. Essas séries serviram de base para a montagem das estimativas, que variam de acordo com a escolha dos parâmetros $\alpha$, $\beta$ e $\gamma$. Foram testados valores entre 0 e 1 e com precisão de três casas decimais. O mesmo foi feito para o parâmetro da EWMA. A combinação que forneceu o menor somatório de erros foi eleita como a mais adequada para os traces estudados, de acordo com a Tabela 2.

**Tabela 2. Parâmetros otimizados para estimadores HW e EWMA**

| Trace | ESTIMADOR | | | |
|---|---|---|---|---|
| | Holt-Winters | | | EWMA |
| | α | β | γ | α |
| **RNP** | 0.775 | 0.005 | 0.035 | 0.690 |
| **MAWILab** | 0.630 | 0.010 | 0.058 | 0.653 |

Pode-se perceber na Tabela 2 que os valores de $\beta$ e $\gamma$ são bem menores que os de $\alpha$. Esses valores evidenciam o peso de cada componente de suavização exponencial na estimativa. Sendo assim, as componentes ligadas à sazonalidade e à tendência de crescimento, apresentam menor influência na estimação dos próximos valores.

### 5.3. Inserção de Ataques

Para validar os métodos de detecção foi necessária a inserção de ataques gerados artificialmente. Para tanto, alguns pressupostos foram assumidos:

- Traces originais livres de ataques ou com percentual irrelevante de fluxos maliciosos;
- Volume máximo de tráfego suportado pelos roteadores desprezado;
- Descartes e atrasos de pacotes, devido ao volume de tráfego inserido, não implementados.

As anomalias são identificadas já no início dos ataques, quando há uma mudança de comportamento do tráfego, por isso as considerações acima não invalidam as simulações realizadas, já que parte dos efeitos colaterais causados pelo tráfego adicional, e que foram desconsiderados durante a inserção, ocorre após a detecção. Essa abordagem permite uma implementação mais simples, uma vez que basta acrescentar ao tráfego original as informações dos pacotes maliciosos.

Outros fatores foram considerados para determinar a metodologia de inserção:

- Foco na detecção de ataques DDoS caracterizados por grande volume de tráfego;
- Anomalias presentes na distribuição de probabilidade de endereços de destino.

 ©2012 SBC — Soc. Bras. de Computação

Optou-se pela inserção de ataques com taxa fixa de pacotes, destinados a uma mesma vítima, e número fixo de atacantes. Foram inseridos 15 ataques com duração fixa de 10 unidades de tempo, e espaçados entre si em 50 unidades. Dessa forma, foi possível avaliar a capacidade de detecção em diferentes instantes do período observado.

Para cada trace, foram seguidos os seguintes passos para a inserção:

**Passo 1:** Cálculo do número médio de pacotes por intervalo de tempo para todo o período;

**Passo 2:** Cálculo do percentual a ser acrescentado;

**Passo 3:** Seleção dos intervalos para inserção;

**Passo 4:** Seleção de número de atacantes;

Com o intuito de verificar a sensibilidade de cada métrica, foram realizadas simulações com a inserção de 50, 25, 20, 15, 10 e 5% do volume médio de pacotes transitado no período, de acordo com a Tabela 3.

**Tabela 3. Número de Pacotes acrescentados na inserção de ataque**

| Volume Médio Inserido | Número de pacotes por janela | |
|---|---|---|
| | RNP (5 min) | MAWILab (1 seg) |
| 50% | 42.336 | 23.743 |
| 25% | 21.168 | 11.872 |
| 20% | 16.935 | 9.497 |
| 15% | 12.701 | 7.123 |
| 10% | 8.467 | 4.748 |
| 5% | 4.234 | 2.374 |

Na Tabela 4 são apresentados alguns valores referentes aos ataques 7.7 e 3.4, para fins de comparação com os ataques inseridos neste trabalho. Nota-se que 10% do volume médio do trace do MAWILAb, por exemplo, representa um valor 128 vezes menor que o número de pacotes por segundo (PPS) por vítima observado no ataque 7.7. Percebe-se, também, que o número de PPS por vítima deste ataque é aproximadamente 1.825 vezes maior do que 100% do volume médio do trace da RNP. Desta forma, pode ser inferido que os valores utilizados nas inserções de ataque do corrente trabalho poderiam ser observados em pontos afastados de vítimas de ataques de inundação.

**Tabela 4. Características dos ataques 7.7 e 3.4. Fonte (Ahnlab, 2011)**

| | Ataque 7.7 (07/07/2009) | Ataque 3.4 (04/03/2011) |
|---|---|---|
| Tráfico por zumbi | 103 PPS | 389 PPS |
| Número de zumbis | 115.044 | 116.299 |
| PPS total | 11.849.532 | 45.240.311 |
| Número de vítimas | 23 | 40 |
| PPS por vítima | 515.197 | 1.131.007 |

A inserção foi realizada durante a extração dos dados, de acordo com a Figura 2, sem a necessidade de alterar o trace original.

 ©2012 SBC — Soc. Bras. de Computação

Inicialmente, os endereços IP de destino dos pacotes presentes em cada intervalo são extraídos do trace, de forma a se obter as distribuições de probabilidade para este parâmetro. Em seguida, uma rotina verifica se deve ou não ocorrer a inserção de pacotes, de acordo com os períodos de ataque escolhidos antes da simulação. Caso seja um período de ataque, altera-se o número de pacotes destinados à vítima, adicionando-se o valor listado na Tabela 3. A escolha da vítima que recebe os 15 ataques é realizada na primeira inserção através de sorteio. Após isso, calcula-se a entropia deste parâmetro no intervalo considerado, com ou sem inserção de dados, e atualiza-se a série temporal acrescentando-se o valor calculado.

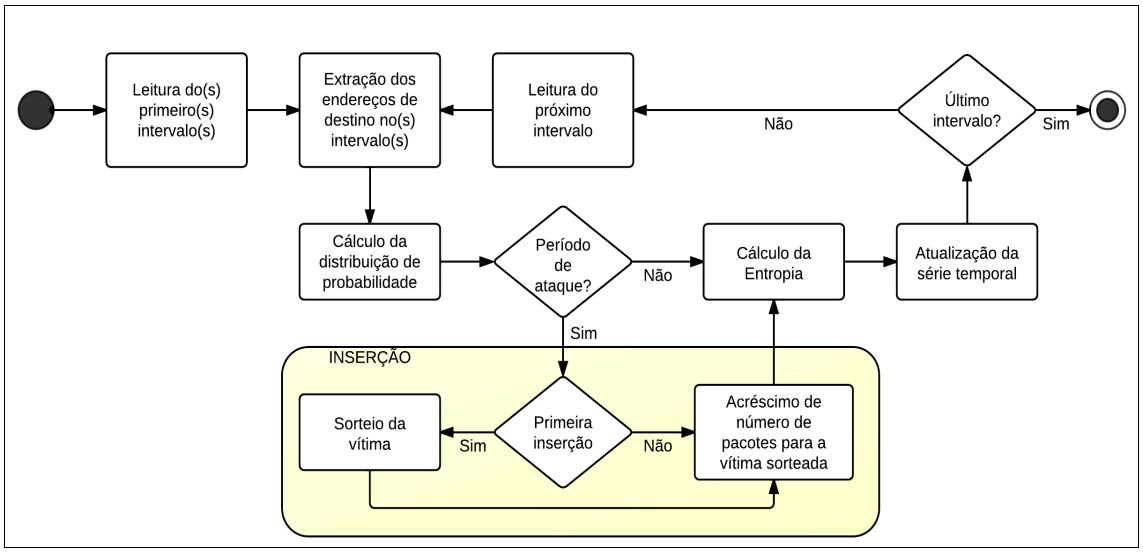



**Figura 2. Algoritmo para inserção de ataques em tempo de execução**

## 5.4. Resultados MAWILab

Observando-se a Figura 3, pode-se verificar que, com a inserção de 15%, 10% e 5% do volume médio transitado, fica mais evidente a capacidade de detecção de cada métrica. Como os ataques são menos expressivos, as mudanças de comportamento são mais brandas e, por isso, mais difíceis de detectar.

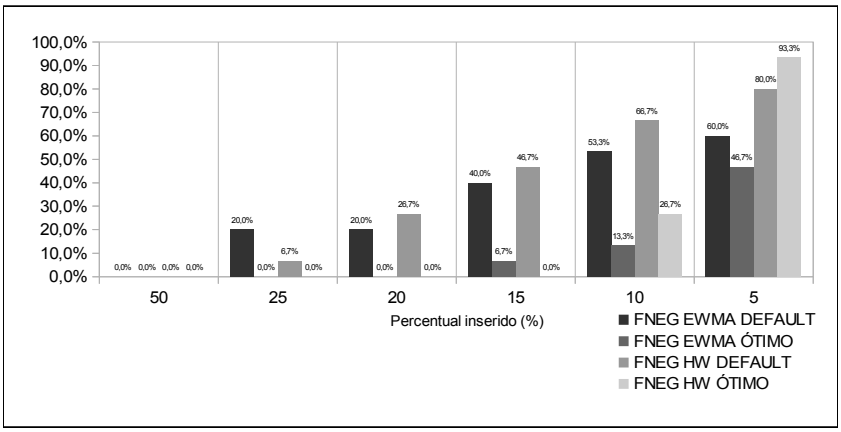



**Figura 3. Índice de FNEG no MAWILab com limites amplos**

Na inserção de 5%, tanto com limites rigorosos (Figura 4) quanto com limites amplos (Figura 3), observou-se o menor índice de FNEG com o emprego do estimador

 ©2012 SBC — Soc. Bras. de Computação

EWMA e parâmetros otimizados, cujos índices de erro alcançaram 20% e 46,7%, respectivamente. Para limites rigorosos, o segundo melhor resultado foi obtido com o estimador HW e parâmetros ótimos. Entretanto, para limites amplos, observou-se para o HW com parâmetros ótimos um crescimento exponencial do número de FNEG na inserção de 5%, quando 93,3% de ataques não são detectados.

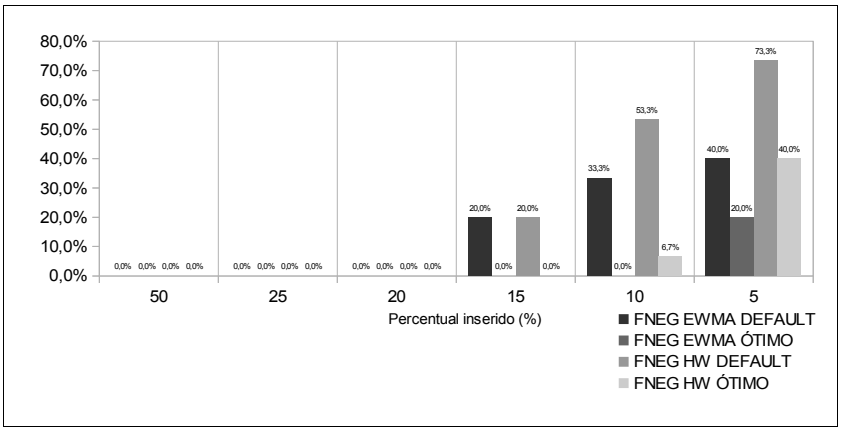



**Figura 4. Índice de FNEG no MAWILab com limites rigorosos**

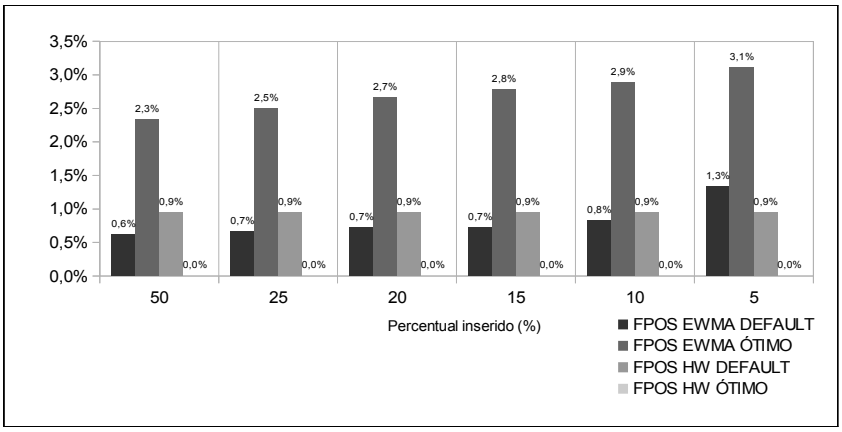



**Figura 5. Índice de FPOS no MAWILab com limites amplos**

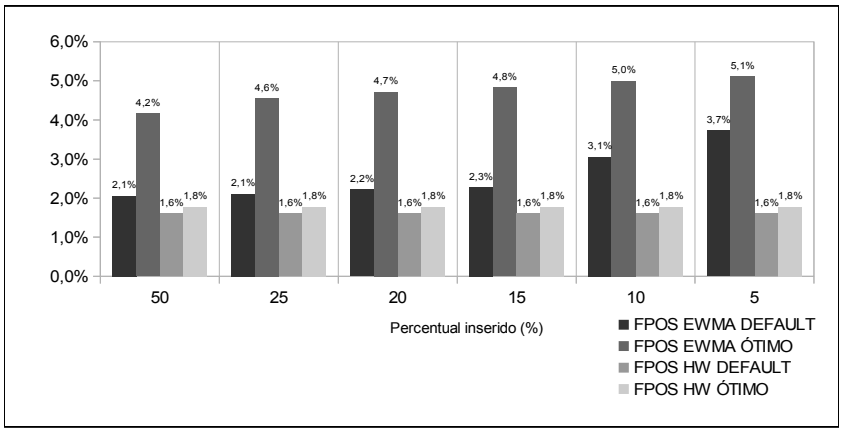



**Figura 6. Índice de FPOS no MAWILab com limites rigorosos**

No que diz respeito ao índice de FPOS, observou-se valores relativamente baixos nas simulações. A variação desse índice, devido à mudança do volume de ataque inserido

 ©2012 SBC — Soc. Bras. de Computação

e ao tipo de limite adotado, também foi relativamente pequena, como pode ser visto na Figura 5 e na Figura 6. Observa-se que o valor máximo obtido foi de 5,11%, quando se utilizou o EWMA com parâmetros otimizados e limites rigorosos. As menores taxas de FPOS foram obtidas com o emprego do HW.

Dessa forma, particularmente para ataques menos volumosos, o estimador que apresentou melhor desempenho na detecção de ataques neste enlace, foi o EWMA com parâmetros otimizados. Quanto aos limites, fica evidente a vantagem de se empregar os rigorosos, dado o pequeno aumento dos índices de FPOS, e a grande diminuição nos índices de FNEG. Esta vantagem foi maior para o HW com parâmetros ÓTIMOS, dado o alto índice de FNEG obtido com o limites amplos.

### 5.5. Resultados RNP

Pode-se perceber pela Figura 7 que, nos traces do RNP, os menores índices de FNEG foram obtidos com o emprego de parâmetros otimizados. Pode-se observar, também, que nas simulações com limites amplos, o número de FNEG cresce mais rapidamente com o EWMA, de forma que, a 10% e 5% de inserção, o HW com parâmetros otimizados apresentou menores índices de erro.

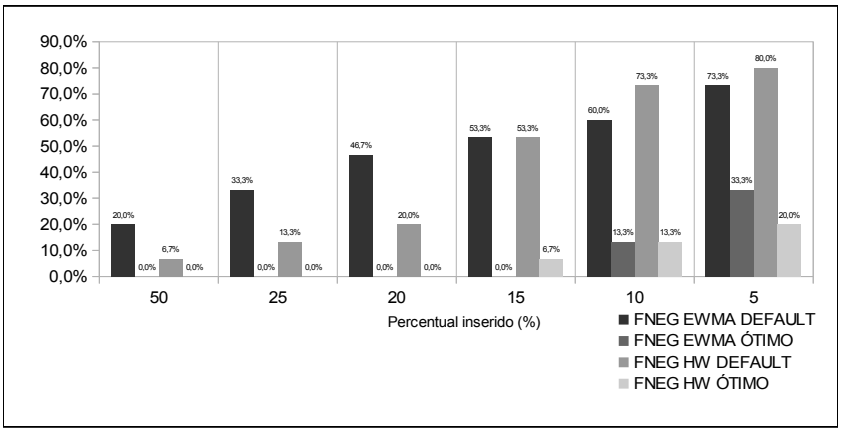



**Figura 7. Índice de FNEG no RNP com limites amplos**

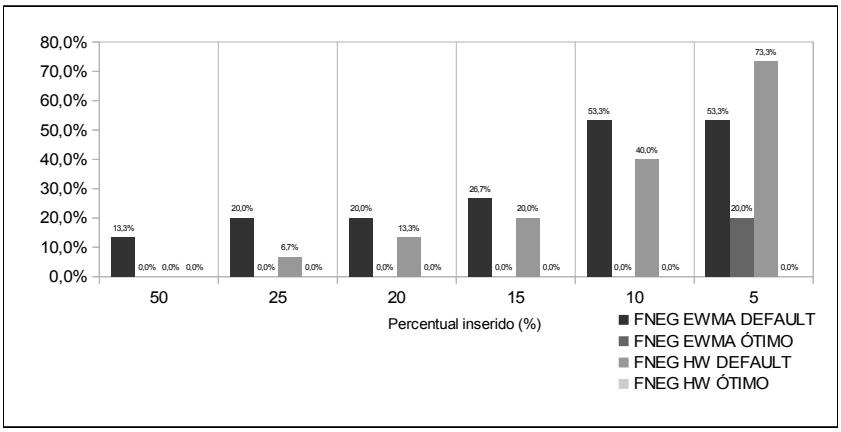



**Figura 8. Índice de FNEG no RNP com limites rigorosos**

 ©2012 SBC — Soc. Bras. de Computação

Além disso, verificou-se que, mesmo a 5% de inserção, o índice de FNEG do HW com parâmetros otimizados foi de 0% (Figura 8). Dessa forma, pode-se dizer que, para esse enlace, o estimador que apresentou o melhor desempenho foi o HW com parâmetros otimizados, tanto para limites amplos como para rigorosos.

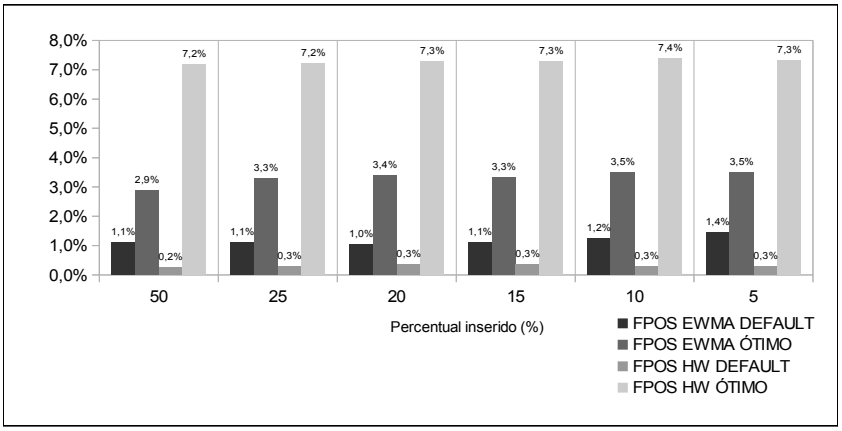



**Figura 9. Índice de FPOS no RNP com limites amplos**

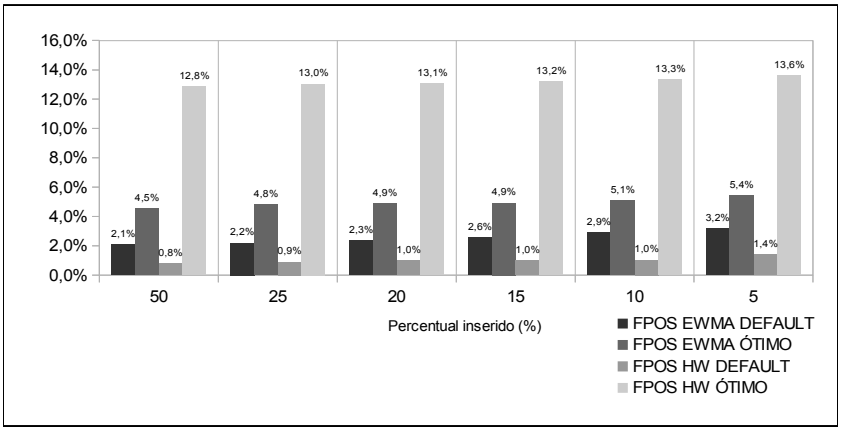



**Figura 10. Índice de FPOS no RNP com limites rigorosos**

Assim como ocorre com o MAWILab, o índice de FPOS muda muito pouco com a variação do volume de ataque inserido. Entretanto, a utilização de limites mais rigorosos quase que duplica o número de FPOS, que, ainda assim, não alcança percentuais muito expressivos. No pior caso, observado com o HW otimizado e limites rigorosos, o índice de FPOS alcançando foi de 13,64%. Com limites mais amplos, esse valor cai para 7,32%. Pode-se observar na Figura 9 e na Figura 10 que o índice de FPOS obtido com o HW equivale a pouco mais que o dobro daquele alcançado com o EWMA, onde os valores não ultrapassam 5,43%, com limites rigorosos, e 3,48%, com limites amplos.

## 6. CONCLUSÃO

Neste artigo, foram avaliados dois métodos de detecção de anomalias de tráfego de rede, baseados na extração de entropias de endereços IP e no uso de estimadores (EWMA e HW). Foram verificados dois tipos de configuração (PADRÃO e ÓTIMA), com o intuito de encontrar qual a combinação mais adequada para a detecção de ataques em pontos afastados da vítima, onde o volume de tráfego malicioso ainda é baixo,

 ©2012 SBC — Soc. Bras. de Computação

considerando-se um cenário colaborativo. Verificou-se, também, o desempenho dessas métricas com o uso de limites AMPLOS e RIGOROSOS.

De maneira geral, pode-se observar que o desempenho dos estimadores diminui a medida em que o volume do ataque inserido é reduzido (de 50% até 5%). Embora não haja alterações significativas nos índices de FPOS, os índices de FNEG aumentam expressivamente a medida em que o volume do ataque diminui.

Diante do exposto, conclui-se que a utilização de parâmetros otimizados e limites mais ajustados aumenta expressivamente a sensibilidade das duas métricas estudadas, trazendo melhores taxas de detecção em ambos os traces. Foi observado um aumento nos índices de FPOS, devido ao maior ajuste dos limites de segurança, embora essa alteração tenha sido relativamente pequena. Por outro lado, os índices de FNEG diminuíram significativamente com parâmetros ÓTIMOS. Por essa razão, a utilização dos estimadores com parâmetros ÓTIMOS em ambientes colaborativos, ou mesmo naqueles onde se deseja identificar volumes menos expressivos de ataque, apresentam uma relação custo-benefício mais vantajosa. Vale salientar que o estimador HW trouxe melhores índices de FNEG apenas para o trace do RNP, devido ao comportamento sazonal identificado neste.

## 7. Referências

Ahnlab, INC. "Analytical report on 3.4 DDoS attack, White Paper", (2011), http://www.ahnlab.com , April.

Feitosa, E. L., Souto E. J. P. e Sadok D. (2008) "Tráfego Internet não Desejado: Conceitos, Caracterização e Soluções", Livro-Texto dos Minicursos do VIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, p. 91-137.

Castelúcio, A. O., Salles, R. M. e Ziviani, A. (2009) "An AS-Level Overlay Network for IP Traceback", IEEE Network, Vol. 23, pp. 36-41.

Kaur, G., Saxena, V. e Gupta, J. P. (2010) "Anomaly Detection in network traffic and role of wavelets", 2nd International Conference on Computer Engineering and Technology (ICCET), v.7, p.46-51.

Lakhina, A., Crovella, M. e Diot, C. (2005) "Mining anomalies using traffic feature distributions", Proceedings of the ACM SIGCOMM'2005, Philadelphia, PA, USA.

Lucena, S. C. e Moura, A. S. (2008) "Detecção de Anomalias Baseada em Análise de Entropia no Tráfego da RNP", XIII Workshop de Gerência e Operação de Redes e Serviços (WGRS), Rio de Janeiro. Anais do XIII Workshop de Gerência e Operação de Redes e Serviços, p. 163-176.

Park, K. e Lee, H. (2000) "A Proactive Approach to Distributed DoS Attack Prevention using Route-Based Packet Filtering", Technical Report CSD-TR-00-017, Purdue University, Dept. of Computer Sciences.

Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D. R. e Shenker S. (2010) "DDoS defense by offense", ACM Transactions on Computer Systems (TOCS), Journal Vol. 28, Issue 1.

 ©2012 SBC — Soc. Bras. de Computação

Feng, J. e Liu, Y. (2009) "The Research of DDoS Attack Detecting Algorithm Based on the Feature of the Traffic", Networking and Mobile Computing 5th International Conference on Wireless Communications (WiCom'09).

Law, K. T., Lui, J. C. S. e Yau, D. K. Y. (2002) "An Effective Methodology to Traceback DDoS Attackers", X IEEE Int'l Symp, MASCOTS'02.

Demir, O. e Khan, B. (2010) "Quantifying Distributed System Stability through Simulation: A Case Study of an Agent-Based System for Flow Reconstruction of DDoS Attacks", IEEE, 2010 ISMS, Liverpool, England, January.

Chen, Y. e Hwang, K. (2006) "Collaborative Change Detection of DDoS Attacks on Community and ISP Networks", IEEE Networks, pp. 401 - 410.

Lin, B. P. e Uddin M. S. (2005) "Synmon Architecture for Source-based SYN-flooding Defense on Network Processor", IEEE, 2005 Asia-Pacific Conference on Communications, Perth, Western Australia.

Kline, J., Nam, S., Barford, P., Plonka, D. e Ron, A. (2008) "Traffic Anomaly Detection at Fine Time Scales with Bayes Nets", The Third International Conference on Internet Monitoring and Protection, ICIMP'08, PP. 37-46.

Papadopoulos, C., Lindell, R., Mehringer, J., Hussain, A. e Govindan, R. (2003) "COSSACK: Coordinated Suppression of Simultaneous Attacks", Proc. Third DARPA Information Survivability Conf. and Exposition (DISCEX-III'03), pp.2-13.

Mirkovic, J. e Reiher, P. (2005) "D-WARD: A Source-End Defense against Flooding DoS Attacks", IEEE Trans. Dependable and Secure Computing, pp. 216-232.

Chen, Y., Hwang, K. e Ku, W. S. (2007) "Collaborative Detection of DDoS Attacks over Multiple Network Domains", IEEE Transactions on Parallel and Distributed Systems, Vol. 18, Issue 12, pp. 1649 - 1662.

Xiang, Y., Li, K., e Zhou, W. (2011) "Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics", IEEE Transactions on Information Forensics and Security, Vol. 6, No. 2.

Shannon, C. E. (1948) "A mathematical theory of communication", Bell System Technical Journal, 27:379-423 and 623-656.

Kalekar, P. S. e Rekhi, K. (2004) "Time series Forecasting using Holt-Winters Exponential Smoothing", School of Information Technology, December 6.

Brutlag, J. D. (2000) "Aberrant Behavior Detection in Time Series for Network Monitoring". Proceedings of the 14th Systems Administration Conference (LISA 2000).

Ward, A., Glynn, P. e Richardson, K. (1998) "Internet Service Performance Failure Detection", ACM SIGMETRICS Performance Evaluation Review, Vol. 26, No. 3.

 ©2012 SBC — Soc. Bras. de Computação

# Detecção de variações de malware metamórfico por meio de normalização de código e identificação de subfluxos

**Marcelo F. Cozzolino[1,2], Gilbert B. Martins[3,4], Eduardo Souto[3], Flávio E. G. Deus[2]**

[1]Departamento de Polícia Federal, Setor Técnico-Científico no Amazonas
Manaus, AM – Brasil

[2] Departamento de Engenharia Elétrica, Universidade de Brasília (UNB)
Brasília, DF – Brasil

[3]Instituto de Computação Universidade Federal do Amazonas (UFAM)
Manaus, AM – Brasil

[4]Fundação Centro de Análise, Pesq. e Inovação Tecnológia (Fucapi)
Manaus, AM – Brasil

cozzolino.mfc@dpf.gov.br, gilbert.martins@fucapi.br,
esouto@icomp.ufam.edu.br, flavioelias@unb.br,

***Abstract.*** *This paper presents a methodology to identify metamorphic malware. The file code is normalized and subdivided in code segments (tokens) delimited by changes in program flow. The combination of each token identifier with the two following ones generates a set of flow identifiers used to measure the similarity to a previously mapped malware code. The results show that proposed methodology is able to accurately identify the presence of metamorphic code.*

**Resumo.** Este artigo apresenta uma metodologia capaz de identificar *malware* metamórficos. O código de um arquivo é submetido a um processo de normalização e subdividido em trechos de códigos (*tokens*) delimitados por mudanças de fluxo do programa. A combinação do identificador de cada *token* com os dois seguintes cria um conjunto de identificadores de fluxo, que são usados para medir a similaridades com um código de *malware* previamente mapeado. Os resultados obtidos mostram que a metodologia proposta é capaz de identificar com precisão a presença de códigos metamórficos.

## 1. Introdução

O termo *malware* (proveniente do inglês *malicious software*) é usado para classificar um software destinado a se infiltrar em um sistema de computador alheio sem consentimento, nem conhecimento, do usuário, com o intuito de causar algum dano ou roubo de informações. Vírus de computador, *worms*, *trojan horses* (cavalos de Troia) e *spywares* são exemplos de *malware* [Skoudis 2004]. Atualmente, a técnica mais comum de detecção destas ameaças, se baseia na procura de uma sequência de bytes (assinatura) conhecida do *malware* em uma base de dados (base de assinaturas).

 ©2012 SBC — Soc. Bras. de Computação

Como resposta a esta abordagem de detecção, alguns desenvolvedores de *malware* empregam técnicas que modificam a estrutura do código automaticamente. Tais técnicas possibilitam a alteração do código do vírus à medida que sua propagação ocorre, tornando esta nova versão do código incompatível com a assinatura associada à versão anterior [Borello e Mé 2008] [Moura e Rebiha 2009].

*Malwares* que empregam estas técnicas podem ser classificados em dois grupos [Notoatmodjo 2010]: Polimórficos e Metamórficos. Os polimórficos utilizam técnicas de criptografia combinadas com uma mudança contínua da chave criptográfica para garantir a diferenciação contínua dos códigos gerados. Os metamórficos não se baseiam em um processo de criptografia, mas fazem uso de técnicas de ofuscação de código mais complexas, criando novos trechos de código que mantém todas as funcionalidades originais intactas. [Rad e Masrom 2010] destacam ainda que a efetividade do metamorfismo se baseia no fato de que a maioria das assinaturas é criada com base na varredura do código, o que gera uma dependência das estruturas sintáticas apresentadas por aquele código. Assim, para evitar a detecção, novas versões do código são criadas mantendo a estrutura semântica original e incluindo alterações sintáticas que os tornam incompatíveis com a assinatura gerada originalmente.

[Bruschi et al. 2007], [Notoatmodjo 2010] e [Kim e Moon 2010] apresentam as técnicas metamórficas mais comuns de ofuscação. A partir de um código original, podem ser aplicadas modificações como: inserção de instruções e variáveis irrelevantes que não interfiram com o conjunto original; renomeação de variáveis ou troca de variáveis entre operações distintas; troca de sequências de instruções por outras equivalentes; e alterações na ordem de execução das instruções, de forma que o resultado final daquela porção de código seja mantido.

Códigos maliciosos podem ser identificados a partir da análise da sequência de instruções. Para tanto, diversas abordagens alternativas têm sido empregadas para este fim. Por exemplo, temos: abordagens baseadas na análise conjuntos de instruções que caracterizam funcionalidades maliciosas [Batista 2008]; técnicas de mineração de dados que podem ser usadas para mapear estruturas de dados comumente encontradas em *malware* [Schultz et al. 2001] ou para identificar a *engine* metamórfica empregada [Andrew et al. 2006]; técnicas de normalização de código tem sido usadas na reversão dos metamorfismos [Christodorescu 2007], na utilização de operações originalmente empregadas em otimização para "limpar" o código [Bruschi et al. 2007], ou mesmo no mapeamento de grafos de dependência [Kim e Moon 2010]. Todas estas alternativas tem como objetivo possibilitar que abordagem de assinaturas possa voltar ser aplicada com sucesso.

A abordagem proposta neste artigo se enquadra na categoria de normalização de código. Primeiramente um *malware*, apresentado em formato binário, tem seu código revertido para o formato *assembler.* Em seguida, este código é normalizado e subdividido em trechos de código (aqui denominados de *tokens*) delimitados por mudanças de fluxo do programa, dividindo o código em blocos que, mesmo reposicionados, tenham a mesma sequência de instruções executadas, preservando sua funcionalidade original.

 ©2012 SBC — Soc. Bras. de Computação

Para cada *token* é calculado um valor identificador de seu conteúdo (denominado de *token id*) através de uma função *hash* que tem como parâmetro de entrada o seu código *assembler*. Para cada *token*, através da análise da sequência de execução do código, são determinados até dois possíveis tokens subseqüentes. Ao conjunto do *hash* deste *token* concatenado com o dos seus dois possíveis subseqüentes chamamos de "identificador de fluxo" (*flow id*). Tal identificador é usado para medir a quantidade de similaridades entre um código de *malware* previamente mapeado e um código suspeito, auxiliando no processo de identificação do *malware.*

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta um conjunto de trabalhos relacionados, apresentando diversas tecnologias utilizadas na identificação de códigos metamórficos; a Seção 3 descreve uma visão geral da metodologia proposta neste trabalho; a Seção 4 apresenta a descrição do processo de preparação da base de referência, que armazena o conjunto de *flow id* utilizado para a identificação do *malware*; a Seção 5 apresenta o detalhamento do procedimento de detecção utilizado para analisar um arquivo sob suspeita; a Seção 6 apresenta um estudo de caso que exemplifica a utilização da metodologia e uma análise dos resultados atingidos; e a Seção 7 apresenta as conclusões deste trabalho.

## 2. Trabalhos Relacionados

Esta seção descreve alguns trabalhos encontrados na literatura que utilizam técnicas de normalização de código no processo de detecção de *malware* metamórficos. Todas as abordagens possuem um conjunto particular de características individuais associadas à forma como cada uma trata o problema de normalização, oferecendo uma visão mais ampla dos desafios associados a esta abordagem.

[Christodorescu et al. 2007] propõe o uso de "normalização" do código no processo de identificação de variantes de *malware* metamórfico em antivírus. Esse trabalho concentra-se em três tipos específicos de mutações metamórficas: *a)* Inserção de código irrelevante – insere instruções inúteis, mas que não interfere com o desenrolar do programa; *b)* Mudança de ordenação – muda a ordem de instruções que não alterem o funcionamento final do programa; e *c)* Compactação – a compactação deixa o código em um estado que ocupa menor espaço para só ser descompactado no momento que for ser executado, o que dificulta a identificação do código conhecido devido à impossibilidade de saber qual algoritmo de compactação foi usado. O processo de normalização descrito inicia revertendo a compactação para obter um código executável, em seguida elimina os desvios (*jumps*), que alteram a ordem das instruções, gerando um código que representa o fluxo "normal" das instruções. Por último, ocorre a eliminação das instruções irrelevantes para assim gerar um novo código que se aproxime ao máximo do código original, que então será submetido à avaliação por processos tradicionais de detecção por assinatura. As principais limitações deste trabalho estão associadas à impossibilidade de garantir que o fluxo original das instruções possa sempre ser completamente reconstruído e que restem somente as instruções realmente relevantes para o funcionamento do *malware* [Christodorescu et al. 2007].

O trabalho de [Bruschi et al. 2007] baseia-se no fato de que a maioria das transformações utilizadas por *malware*, visando disfarçar sua presença, tem como efeito colateral um aumento no tamanho código binário produzido. O código resultante pode

 ©2012 SBC — Soc. Bras. de Computação

ser encarado como um código não otimizado, pois contará com alguns comandos irrelevantes, destinados exclusivamente a mascarar o código original. Desta forma, a normalização deste código, através de técnicas de otimização utilizadas comumente por compiladores, terá como efeito a eliminação deste código "inútil", facilitando o processo de reconhecimento. Esta abordagem se diferencia das outras por adicionar uma etapa de "interpretação de código", onde as instruções *assembler* são simplificadas ainda mais em uma linguagem com poucos operadores semânticos. O código gerado por essa etapa será tratado pelos processos de normalização e de comparação, na busca pela identificação do *malware*. Entretanto, apesar do processo de normalização, aplicado durante a execução de testes, apresentar tempos razoáveis, quando aplicado a programas de tamanho maior demandou uma grande quantidade de tempo e recursos computacionais, o que limita sua efetiva aplicabilidade [Bruschi et al. 2007].

[Kim e Moon 2010] apresentam um mecanismo de detecção de *malware* embutidos em códigos script, escritos em linguagens como *Visual Basic Script* e *JavaScript*. Como estas linguagens são interpretadas, este tipo de *malware* se propaga em forma de código não compilado ou código fonte. Entretanto, mesmo nesta forma de apresentação, estes códigos estão sujeitos à utilização de técnicas de polimorfismo. Para lidar com isto, o código é analisado e transformado num grafo de dependência que representa as relações de dependência entre as linhas de código semântico, com o foco na manipulação das variáveis utilizadas neste programa. Cada vértice do grafo gerado representa uma linha de código e a dependência entre duas linhas é modelada como uma aresta direcional. O grafo é então normalizado, para eliminar vértices ou conjuntos de vértices que não possuam qualquer relação com os outros vértices do grafo principal. Isto, além de diminuir o tamanho do grafo que será analisado, tem o efeito de eliminar código inserido para efeito de ofuscação. A partir daí, o processo de detecção é tratado como o problema de encontrar o maior isomorfismo entre um grafo que modele um código previamente identificado com *malware* e um subgrafo do grafo de dependência que modela o código script contaminado. Como encontrar o máximo isomorfismo de um subgrafo é um problema NP-Difícil, a maior limitação deste trabalho é o custo computacional associado a esta atividade, o que poderia gerar tempos de execução impraticáveis [Kim e Moon 2010].

A metodologia proposta neste trabalho também visa desfazer as mutações metamórficas, contidas em uma biblioteca previamente mapeada, antes de proceder à comparação do código. Entretanto, diferentemente de [Christodorescu et al. 2007], o problema de levantamento do fluxo é tratado neste trabalho possibilitando capturar de forma mais efetiva a lógica de funcionamento do código reconstruído. Além disso, o processo de normalização utilizado neste trabalho é baseado em bibliotecas de metamorfismos pré-conhecidas, apresentando ordem de complexidade inferior aos trabalhos de [Bruschi et al. 2007] e [Kim e Moon 2010].

## 3. Visão geral da Metodologia Proposta

O processo de detecção de um *malware* é baseado na construção de uma base de referência a partir do código normalizado do *malware* buscado, que será usada na comparação com o código também normalizado de um arquivo sob suspeita de contaminação. Na base de referência ficam armazenados os identificadores de trechos de códigos (*token id* e *flow id)* gerados após a normalização e processamento do código de

 ©2012 SBC — Soc. Bras. de Computação

um *malware* metamórfico. Com esta base preparada, o mesmo procedimento de normalização e geração dos identificadores de trechos de códigos é executado em um arquivo suspeito de contaminação. O processo de detecção visa encontrar correspondências de sequências de trechos de códigos entre a base de referência e o conjunto de identificadores de trechos de código gerados a partir do arquivo suspeito, através da comparação de seus *flow id*. Cada correspondência de *flow id* (*flow_id matching)* é então contabilizada para indicação do nível de compatibilidade entre o código do *malware* mapeado e o código sob suspeita. O valor auferido ao término deste do procedimento de comparação é usado para determinar se o arquivo investigado está ou não contaminado. A Figura 1 ilustra uma visão geral da metodologia proposta neste trabalho e as seções seguintes detalham cada etapa.

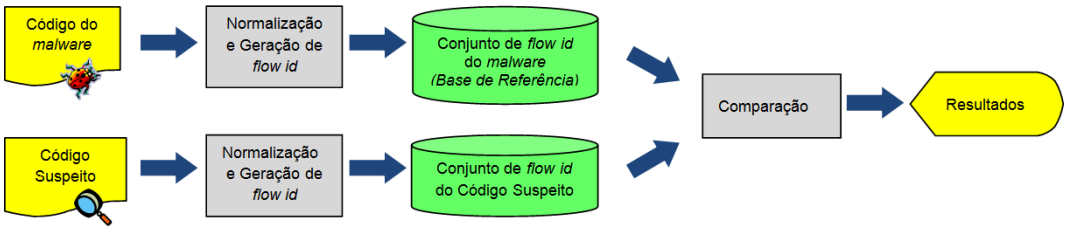



**Figura 1 – Visão geral da metodologia de detecção de *malware* metamórficos**

## 4. Preparação da Base de Referência

Como mencionado anteriormente, a base de referência é basicamente composta por identificadores de trechos de códigos normalizados (*tokens*). O processo de geração desses identificadores é realizado em cinco etapas: *disassembler*, levantamento do fluxo do programa, divisão em *tokens*, normalização e cálculo do *flow id*.

### 4.1. Disassembler

A base das ferramentas de análise de programas maliciosos é o processo de *disassembler*, cuja função é executar a transformação de um código binário, visando à recuperação de um código *assembler* e viabilizando a interpretação das instruções. Este processo pode ser realizado através de *disassemblers* convencionais encontrados no mercado. OllyDbg [OllyDbg 2011] e IDA Pro [Hex-Rays 2011] são exemplos de *disassemblers* compatíveis com o Windows, enquanto que *Bastard Disassembler* [SorceForge 2011] e LIDA [Schallner 2004] são exemplos de *disassemblers* compatíveis com o Linux.

### 4.2. Levantamento do fluxo do programa

Após o processo de *disassembler* do código, inicia-se a etapa que busca levantar a ordem de execução das instruções para construir o fluxo do programa. Para tornar isso possível, são levantadas, para cada instrução, quais as possíveis instruções subsequentes. Cada instrução terá dois endereços de saída, que serão representados pelas variáveis F1 e F2. F1 será sempre o endereço da instrução seguinte pelo posicionamento da memória (execução de código sequencial) e F2 será o endereço da instrução de desvio (em caso de desvio de código). Instruções que permitam mais de dois destinos de saída (retorno de sub-rotina ou desvios indiretos) não terão o fluxo levantado neste trabalho, devido à

 ©2012 SBC — Soc. Bras. de Computação

complexidade desta operação em análise estática. Nestes casos, os valores de F1 e F2 serão igualados a zero, para sinalizar que o fluxo não foi levantado. Nas chamadas de sub-rotinas será considerado que uma sub-rotina terá o comportamento padrão, ou seja, retorna para a instrução seguinte. Neste levantamento não serão considerados os desvios do programa nas chamadas de sub-rotinas.

### 4.3. Divisão em *tokens*

A terceira etapa consiste na divisão do código em partes menores que serão usadas na fase de comparação dos códigos. Essas partes menores têm sua função semântica dentro do código preservada durante as operações de metamorfismo, passando a serem identificadas como *tokens*. A divisão do código em *tokens* deve ser feita de forma a garantir que, mesmo após o processo de metamorfismo, o código alterado permanece dentro do mesmo *token* que sofreu o metamorfismo. Metamorfismos podem além de alterar a forma de apresentação dos *tokens*, alterar a ordem dos *tokens* em que essa ordem não seja importante. Para a divisão dos *tokens* foram utilizados critérios para atender as seguintes condições:

1. Os *tokens* devem ser executados a partir da primeira linha e terminados na última linha. O objetivo é evitar que o código de um *token* comece a ser executado por uma linha que não a primeira deste. Isso faz com que em *tokens* equivalentes, códigos também equivalentes tenham sido executados;
2. A divisão dos *tokens* deve facilitar a identificação do próximo *token* a ser executado logo após a saída do *token* corrente. Caso não seja possível identificar qual o *token* seguinte de forma trivial em análise estática, este fato deve ser sinalizado.

Foram usados três tipos de delimitadores dos *tokens*: *i)* As linhas *assembler* com rótulos (*labels*); *ii)* As instruções de desvio; e *iii)* As instruções de retorno de *subrotina*. Através do fluxo de programa, levantado pela última instrução do *token*, serão identificados *tokens* seguintes, o que será usado posteriormente na etapa de comparação. A Tabela 1 mostra um exemplo de código *assembler* dividido em *tokens*, mostrando também os dois possíveis *tokens* seguintes.

**Tabela 1 – Exemplo da divisão em *tokens* do código *assembler***

| Token | Endereço | Instrução | Parâmetros |
|---|---|---|---|
| 36 | 4011f2h | cmp | eax, 77F00000h |
| 36 | 4011f7h | jnz | short 401221h |
| Próximo token 1 = 37 | | Próximo token 2 = 40 | |
| 37 | 4011f9h | mov | dword ptr [esi], 550004C2h |
| 37 | 4011ffh | mov | dword ptr [esi+4], 0C244C8Bh |
| 37 | 401206h | jmp | short 401221h |
| Próximo token 1 = 40 | | Próximo token 2 = 40 | |
| 38 | 401208h | cmp | eax, 77E00000h |
| 38 | 40120dh | jnz | short 401261h |
| Próximo token 1 = 39 | | Próximo token 2 = 46 | |
| ... | ... | ... | ... |
| 40 | 401221h | mov | edi, eax |

 ©2012 SBC — Soc. Bras. de Computação

### 4.4. Normalização

Na etapa de normalização, os *tokens* passam por um processo que visa tornar semelhantes *tokens* derivados de um mesmo *token* inicial, desfazendo as alterações que foram implementadas pela *engine* metamórfica, de forma a se obter um código mais próximo ao "original". Entretanto, não é possível ter certeza que determinado código é derivado de um metamorfismo. Independente de ter sido originado ou não pela ação da *engine* os códigos serão revertidos. Isso não trará problemas na comparação por que esta reversão também ocorrerá no código do *malware*, antes da comparação. Além disso, existem transformações que podem ser executadas e, após vários metamorfismos sobrepostos, voltar à condição inicial, gerando uma situação "cíclica", não sendo possível definir exatamente qual seria o código inicial. Em casos assim, arbitra-se um código como sendo o inicial em todas as ocorrências, de forma a levar todas as variações metamórficas até um ponto comum.

O processo de normalização usado neste trabalho é baseado em bibliotecas de *engines* metamórficas conhecidas de *malware*. Uma *engine* metamórfica corresponde ao trecho de código do *malware* responsável pela mutação do código. A ideia é entender o processo de metamorfismo empregado pelo *malware* e desfazer uma a uma as mutações. Para garantir a volta ao código inicial, seria necessário desfazer os metamorfismos na ordem inversa em que foram feitos. Contudo, como não é possível identificar a quantidade de vezes e a ordem em que os metamorfismos foram aplicados, não existe um procedimento exato que garanta a reversão dos metamorfismos. Ciente deste problema, este trabalho usa a seguinte ordem de reversão: desfazer primeiro as mutações mais complexas (aquelas que geram mais instruções) e depois as mais simples. Essa escolha de reversão considera que um código resultante de uma mutação mais complexa tem menos chance de ser sido produzido incidentalmente, sem participação da *engine* metamórfica. Como exemplo, a Figura 2 apresenta transformações, que são partes de uma *engine* metamórfica. A reversão do metamorfismo exposto em (a), pode ser inviabilizada se a reversão do metamorfismo exposto em (b) for executado antes.

| **Linha** | **Código com metamorfismo** | **Código Original** |
|---|---|---|
| 1 | push eax | movsb |
| 2 | mov al, [esi] | |
| 3 | add esi, 1 | |
| 4 | mov [edi], al | |
| 5 | add edi, 1 | |
| 6 | pop eax | |
| (a)Metamorfismo complexo | | |
| **Linha** | **Código com metamorfismo** | **Código Original** |
| 1 | mov al, [esi] | lodsb |
| 2 | add esi, 1 | |
| (b)Metamorfismo simples | | |

**Figura 2 – Exemplos de metamorfismos**

O código metamórfico apresentado em (b) corresponde a um subconjunto do código metamórfico apresentado em (a), linhas 2 e 3. Se estas linhas forem substituídas pelo código original em (b) – instrução lodsb, o código original em (a) – instrução movsb, não será gerado pelo processo de reversão.

Durante o processo de reversão todo trecho de código que corresponder ao resultado de um metamorfismo será revertido, não importando se o código foi realmente resultante

 ©2012 SBC — Soc. Bras. de Computação

de um metamorfismo ou não, visto que o objetivo não é obter o código original, mas reconhecer códigos que possam ter a mesma origem independente do estado de metamorfismo.

O algoritmo varre os trechos de códigos correspondentes a um *token* visando identificar todas as instruções ou conjunto de instruções que são candidatas à reversão. Em seguida, esses conjuntos de instruções são processados de acordo com a ordem de complexidade, da maior para a menor, obedecendo à prioridade de reversão explicada anteriormente. Este processo se reinicia até que não reste nenhuma para ser revertida naquele *token*.

### 4.5. Cálculo de *Flow Id*

Após o processo de normalização, inicia-se a geração do conjunto de identificadores de fluxo (*flow id*). Estes elementos são gerados usando os identificadores individuais de cada *token* (*token id*) e de seus dois *tokens* subsequentes, de acordo com o levantamento do fluxo do programa (veja Seção 4.2). Para gerá-los é aplicada uma função *hash* que tem como entrada o conteúdo considerado relevante no código do *token*. Neste cálculo de *hash* são consideradas relevantes às instruções e os parâmetros que não sejam endereços de desvio ou de endereço de chamadas de sub-rotinas. Essa diferença de tratamento nos endereços visa garantir que códigos equivalentes sejam reconhecidos como tal, independente do fato do endereço de desvio dos mesmos ter mudado. Isto acontecerá quase sempre em códigos que sofreram metamorfismo, pois o código sofrerá reposicionamento devido às instruções inseridas, mudando os endereços de destino dos desvios.

## 5. Procedimento de Detecção

O procedimento de detecção consiste em repetir as etapas descritas na Seção 4 para cada um dos arquivos que se pretende avaliar e efetuar a comparação entre os dois conjuntos de *flow id*, como mostrado na Figura 1. O processo de comparação é detalhado nas próximas seções.

### 5.1. Comparação dos Códigos Normalizados

Nesta etapa ocorre a comparação do conjunto de *flow id* dos arquivos suspeitos de contaminação com os do *malware* previamente mapeado. Para cada *token* normalizado do arquivo em análise, será comparado o seu *flow id* com o *flow id* de todos os *tokens* do *malware* buscado. Cada coincidência incrementará um contador de incidência. O valor final desse contador será usado para decidir se o arquivo analisado está ou não contaminado por uma variante metamórfica.

### 5.2. Tratamento de Falsos Positivos

O resultado da comparação dos *flow id* pode ser positivo quando da presença do *malware* buscado ou em casos onde ocorrerem coincidências de trechos de código, conduzindo a falsas identificações do *malware*. Quanto maior for o tamanho do arquivo sob teste, maior será a possibilidade de coincidências acidentais. Podem ocorrer casos onde um programa livre de contaminação atinja uma pontuação de incidência superior a um programa menor que esteja efetivamente contaminado com o *malware*.

 ©2012 SBC — Soc. Bras. de Computação

Para verificar a ocorrência deste tipo de *matching* indevido, deve-se executar novamente todo o procedimento de detecção sem, entretanto, processar a etapa de normalização do código suspeito. Após o término deste novo processamento, os resultados obtidos são comparados com os resultados originais para efeito de validação. Caso não exista diferença na pontuação de *matching*, isto indicará que as pontuações foram por motivo de coincidência acidentais de trechos de código uma vez que, como a etapa de normalização foi omitida, não foi necessário desfazer qualquer metamorfismo para a ocorrência de *matching*, o que prova que o código suspeito está livre de contaminação.

## 6. Estudo de Caso

Nos testes de validação da metodologia, foi escolhido o vírus W32 Evol [Symantec 2007]. A escolha de tal vírus foi devido ao fato deste possuir uma *engine* metamórfica que contempla as técnicas mais comuns empregadas para efeito de ofuscação de código.

Para produção de amostras de teste, a ação da *engine* metamórfica do W32 Evol foi emulada através de um programa em linguagem "C++", construído através da conversão direta do código em *assembler* desta *engine* (encontrado no código fonte do W32 Evol), para seu equivalente em linguagem "C++", que executa o processo de ofuscação de código em qualquer arquivo desejado.

O programa de metamorfismo foi executado produzindo 63 versões do *malware* "W32 Evol.a". Cada execução da *engine* metamórfica sob um arquivo será chamada de "geração". Assim, das 63 versões do *malware*, 32 versões foram feitas executando o programa de metamorfismo uma única vez (amostras de primeira geração), 16 versões foram feitas executando o programa de metamorfismo duas vezes (amostras de segunda geração), 8 versões (amostras de terceira geração), 4 versões (amostras de quarta geração), 2 versões (amostras de quinta geração) e 1 versão (amostra de sexta geração). Os experimentos apresentados nesta Seção mostram que as quantidades de amostras utilizadas são suficientes para apontar as tendências comportamentais do *malware* avaliado. O processo de avaliação é feito comparando o vírus e suas amostras metamórficas geradas com os antivírus comerciais e com a metodologia proposta neste trabalho.

### 6.1. Avaliação comparativa de antivírus comerciais

Para avaliar a identificação dos *malware* foram executados testes na ferramenta *VirusTotal* [Virus Total 2011] , que avalia arquivos suspeitos em 43 antivírus comerciais. Primeiramente foi testado o *malware* "W32 Evol.a" em seu estado original, que foi identificado por 39 dos 43 antivírus.

A Tabela 2 apresenta os resultados dos experimentos realizados no *VirusTotal* com 10 amostras metamórficas geradas, sendo 5 amostras de primeira geração e 5 de segunda geração. São apresentados apenas os antivírus que obtiveram uma identificação positiva, mesmo que sem especificar o *malware* corretamente.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 2 – Resultado do teste das versões metamórficas no VirusTotal**

| Amostra | Geração | Antivirus | Malware Identificado |
|---|---|---|---|
| 1 | 1ª | AntiVir<br>McAfee-GW-Edition<br>PCTools<br>Symantec | HEUR/Malware<br>Heuristic.LooksLike.Win32.Suspicious.J<br>Malware.Evol<br>W32.Evol.Gen |
| 2 | 1ª | AntiVir<br>AVG<br>McAfee-GW-Edition<br>PCTools<br>Symantec | HEUR/Malware<br>Win32/Heur<br>Heuristic.LooksLike.Win32.Suspicious.J<br>Malware.Evol<br>W32.Evol.Gen |
| 3 | 1ª | AntiVir<br>McAfee-GW-Edition | HEUR/Malware<br>Heuristic.LooksLike.Win32.Suspicious.J |
| 4 | 1ª | AntiVir<br>AVG<br>McAfee-GW-Edition<br>PCTools<br>Symantec | HEUR/Malware<br>Win32/Heur<br>Heuristic.LooksLike.Win32.Suspicious.J<br>Malware.Evol<br>W32.Evol.Gen |
| 5 | 1ª | AntiVir<br>McAfee-GW-Edition | HEUR/Malware<br>Heuristic.LooksLike.Win32.Suspicious.J |
| 6 | 2ª | McAfee-GW-Edition | Heuristic.LooksLike.Win32.Suspicious.J |
| 7 | 2ª | Comodo<br>McAfee-GW-Edition | Heur.Packed.Unknown<br>Heuristic.LooksLike.Win32.Suspicious.J |
| 8 | 2ª | McAfee-GW-Edition | Heuristic.LooksLike.Win32.Suspicious.J |
| 9 | 2ª | McAfee-GW-Edition<br>NOD 32 | Heuristic.LooksLike.Win32.Suspicious.J<br>a variant of Win32/Kryptik.AT |
| 10 | 2ª | McAfee-GW-Edition | Heuristic.LooksLike.Win32.Suspicious.J |

Os resultados dos experimentos no *VirusTotal* das variantes metamórficas foram insatisfatórios do ponto de vista de eficiência na detecção. No melhor caso, somente 5 antivírus identificaram simultaneamente os arquivos submetidos, dos quais apenas 2 antivírus (*PCTools* e Symantec) identificaram em alguma amostra como sendo o *malware* Evol W32. Esta baixa taxa de sucesso é compatível com os resultados apresentados por [Kim e Moon 2010].

## 6.2. Avaliação usando a metodologia proposta

Neste cenário de avaliação as 63 amostras metamórficas geradas foram comparadas com o "W32 Evol.a" usando a metodologia deste trabalho. Para cada arquivo foram feitas duas comparações: a primeira sem desfazer os metamorfismos e outra desfazendo. Exemplos dos resultados destas comparações são apresentados na Tabela 3. Além destas amostras foram também comparados 8 programas executáveis que não possuem o *malware* e três versões do *malware* W32 Evol – "W32 Evol a", "W32 Evol b" e "W32 Evol c" – disponíveis no site www.vxheavens.com. A metodologia deste trabalho foi aplicada em cada um destes arquivos.

Este trabalho optou por utilizar o valor absoluto do processo de *matching* (contador de incidência) ao contrário de fazer qualquer tipo de normalização de valores. Desta forma, a diferença na quantidade de pontos de *matching* obtida durante o processamento normal e o de validação (veja seção 5.2) será o elemento que tomaremos como base para identificação da presença ou não de contaminação. A diferença de pontuação entre os arquivos com o *malware* (veja a Tabela 3) e os sem *malware* (Tabela 4) foi tão grande que não foi necessário determinar um limiar mínimo que indique que o arquivo investigado está contaminado.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 3 – Exemplos da pontuação da comparação do "W32 Evol.a" com suas versões metamórficas, separadas por geração de metamorfismo, desfazendo (D) e não desfazendo (ND) o metamorfismo.**

| Número da Amostra | Geração | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1ª (ND) | 1ª (D) | 2ª (ND) | 2ª (D) | 3ª (ND) | 3ª (D) | 4ª (ND) | 4ª (D) | 5ª (ND) | 5ª (D) | 6ª (ND) | 6ª (D) |
| 1 | 40 | 126 | 3 | 65 | 3 | 47 | 2 | 26 | 2 | 10 | 2 | 9 |
| 2 | 36 | 117 | 2 | 63 | 2 | 40 | 2 | 31 | 0 | 15 | | |
| 3 | 40 | 130 | 1 | 60 | 1 | 36 | 0 | 18 | | | | |
| 7 | 37 | 123 | 2 | 60 | 2 | 48 | | | | | | |
| 13 | 42 | 125 | 3 | 66 | | | | | | | | |
| 27 | 40 | 119 | | | | | | | | | | |

No universo de amostras testadas, a menor diferença de pontuação entre o processamento normal e o de validação ocorreu na amostra de sexta geração e foi igual a 7(9-2). Foi observada uma tendência a diminuição desta diferença em cada geração do *malware* possivelmente devido a um aumento de falhas ao desfazer os metamorfismos quando em gerações de maior índice.

A Tabela 4 apresenta os resultados obtidos testando arquivos que não possuem o *malware*, para avaliação de falsos positivos. Na última coluna das Tabelas 4 está o resultado da comparação de cada arquivo com ele mesmo (auto-comparação), para validação dos resultados, certificando que a baixa pontuação na comparação não tenha sido devido à falha da metodologia em identificar os *tokens*. Nestas amostras a diferença de pontuação de matching entre desfazendo e não desfazendo os metamorfismos foi igual a zero em todos os casos.

Para validar o processo de verificação, foi realizado um experimento usando outras versões do "W32 Evol" com o código livre de qualquer tipo metamorfismo. A Tabela 5 apresenta os resultados obtidos. Podemos observar que não fez diferença desfazer ou não os metamorfismos, o que caracteriza que eles não são versões metamórficas do mesmo *malware*, mas sim versões originais (sem metamorfismo) de programas ligeiramente diferentes entre si, o que foi confirmado posteriormente por observação direta dos três códigos.

**Tabela 4 – Pontuação de similaridade do "W32 Evol.a" com arquivos livres de contaminação (sem o *malware*), desfazendo (D) e não desfazendo (ND) o metamorfismo.**

| Arquivo | (ND) | (D) | Tamanho | Auto-comparação |
|---|---|---|---|---|
| cacls.exe | 0 | 0 | 25 KB | 890 |
| dialer.exe | 0 | 0 | 31 KB | 717 |
| fveupdate.exe | 0 | 0 | 13 KB | 185 |
| hh.exe | 0 | 0 | 15 KB | 140 |
| Notepad.exe | 1 | 1 | 148 KB | 1119 |
| Winhelp.exe | 0 | 0 | 251 KB | 500 |
| winhlp32.exe | 0 | 0 | 9 KB | 54 |
| wzsepe32.exe | 0 | 0 | 204 KB | 1388 |

**Tabela 5 – Pontuação de similaridade do "W32 Evol.a" com outras versões do "W32 Evol", desfazendo (D) e não desfazendo (ND) o metamorfismo**

| Arquivo | (ND) | (D) | Tamanho | Auto-comparação |
|---|---|---|---|---|
| W32 Evol a | 297 | 297 | 12 KB | 297 |
| W32 Evol b | 293 | 293 | 12 KB | 298 |
| W32 Evol c | 291 | 291 | 12 KB | 298 |

 ©2012 SBC — Soc. Bras. de Computação

### 6.3. Análise dos resultados

Para melhor entendimento dos resultados obtidos durante o processo de experimentação, e apresentados nas Tabelas 3 e 4, destacam-se as seguintes observações:

1) Cada ponto na comparação pode ocorrer por duas razões: pela presença do *malware* ou, por coincidências acidentais dos *flow id*. Coincidências podem ser provenientes do arquivo analisado usar estruturas semelhantes as existentes no *malware* ou por colisão no algoritmo de *hash* do cálculo do *flow id*;

2) Um *malware* comumente estará em uma de três situações:

   a) Acrescido ao código do programa hospedeiro tornando-o maior. Neste caso, a pontuação do arquivo avaliado será provavelmente igual à soma da pontuação do *malware* isolado mais a pontuação acidental do programa hospedeiro original;

   b) Substituindo uma parte inerte (sem código ou dados) do programa original, para dificultar a detecção do *malware* pelo aumento do tamanho do executável original. Neste caso, o *malware* deve estar sobrepondo uma região onde não há código e, portanto, é muito improvável que haja uma coincidência acidental (que estaria sendo destruído pelo *malware*). Assim a provável pontuação de *matching* do arquivo analisado será também a soma da pontuação do *malware* isolado mais a pontuação acidental do arquivo hospedeiro original (sem o *malware*);

   c) Não possuir programa hospedeiro, estando situado em arquivo só com o *malware*;

3) Não foram feitos testes nas duas primeiras condições (item 2, "a" e "b"), com arquivo hospedeiro infectado, tendo sido feito unicamente com o *malware* isolado, ficando o teste nas outras condições como sugestão para trabalhos futuros;

4) Para avaliação de ocorrências de falsos positivos, a metodologia foi testada em 8 arquivos sem o *malware* (livre de contaminação), todos de tamanho igual ou inferior a 251KB. Neste universo ocorreu em apenas um caso com 1 (um) *matching* por coincidência acidental;

**Tabela 6 – Diferenças máxima e mínima entre a pontuação desfazendo (D) e não desfazendo (ND) o metamorfismo e pontuação máxima e mínima para cada geração do arquivos de *malware*.**

| Geração | Diferença mínima entre (D) e (ND) | Diferença máxima entre (D) e (ND) | Pontuação mínima (D) | Pontuação máxima (D) |
|---|---|---|---|---|
| 1 | 78 | 100 | 116 | 131 |
| 2 | 50 | 66 | 51 | 68 |
| 3 | 35 | 47 | 35 | 49 |
| 4 | 18 | 29 | 18 | 31 |
| 5 | 8 | 15 | 10 | 15 |
| 6 | 7 | 7 | 9 | 9 |

5) Dentre os arquivos com *malware* testados foram levantadas as pontuações máxima e mínima conseguidas dentre as amostras de cada geração desfazendo o metamorfismo, para cada geração do *malware*. Tal critério pode ser usado para a identificação do *malware*, conforme mostra a Tabela 6

6) Como complemento ao critério de identificação do *malware* também foram empregadas as diferenças máxima e mínima desfazendo e não desfazendo o metamorfismo, para cada geração do *malware* (ver Tabela 6).

 ©2012 SBC — Soc. Bras. de Computação

7) Se as amostras apresentadas na Tabela 3 não estivessem sozinhas, e sim anexadas em algum arquivo hospedeiro (casos 2.a e 2.b), a diferença de pontuação permaneceria a mesma. Isto ocorreria porque as pontuações extras decorrente de identificação de estruturas similares (acidentais) ao vírus estariam presentes nos dois momentos do arquivo avaliado (normalizado e não normalizado). Isto seria verdade com os 8 arquivos livre de contaminação com os quais foram feitos os testes (veja tabela 4), mas nem sempre esta diferença será nula, sendo sugerida a avaliação deste comportamento para arquivos maiores em trabalhos futuros.

8) Nos casos testados, não houve interseção entre as regiões de máximo e mínimo nem de pontuação e nem de diferença entre as diferentes gerações, sugerindo ser possível identificar inclusive a geração do *malware* presente no arquivo pela pontuação final, dependendo do intervalo. No entanto, principalmente nas gerações mais elevadas, as amostradas utilizadas foram pequenas para tal afirmação, necessitando de uma avaliação mais ampla para uma conclusão categórica.

9) Analisando os resultados obtidos com os testes em arquivos livres de contaminação, de até 251KB de tamanho, os resultados sugerem que um valor de pontuação entre 1 e 9 ou uma diferença de pontuação (processo de validação, conforme seção 5.2) entre 0 e 7, poderiam ser usados como limiar para identificação deste *malware*. Isto foi verdade para o caso particular deste *malware* com os arquivos usados nos testes. Para determinar um algoritmo que determine os valores deste limiar para um caso genérico serão necessários muito mais testes com amostras muito mais variadas.

## 7. Conclusões

Os resultados obtidos nos experimentos foram bastante promissores, tendo identificado 100% dos arquivos que continham o *malware* e 0% de falsos positivos com os arquivos testados sem *malware*. Durante os experimentos foram identificas que as três versões do *malware* testado não eram versões metamórficas de um mesmo *malware*, mas sim versões ligeiramente diferentes do *malware*, fato confirmado posteriormente através da observação direta dos três códigos, reforçando os indícios de validade da metodologia.

Os resultados obtidos indicam o potencial da metodologia proposta e justificam a continuidade de novas pesquisas. Questões ligadas à cifragem e compactação de código não são tratadas nesta metodologia. Para que a metodologia seja aplicável nestes casos, será necessário usar a metodologia em conjunto com algum outro tratamento específico. Quando não for possível acessar o código decifrado em análise estática essa metodologia poderia ser aplicada de forma dinâmica como, por exemplo, usando o conteúdo da memória. Outra evolução interessante da metodologia é analisar o seu funcionamento quando aplicada a arquivos maiores e sem contaminação, visto que estes podem gerar pontuações de *matching* altas por simples coincidência. Caso ocorram situações de falso positivo, buscar soluções para minimizá-las. Um algoritmo de normalização da pontuação, relevando o tamanho ou número de *tokens*, do arquivo analisado e do *malware* buscado, pode se fazer necessário. A questão da ordem em que são desfeitas as alterações metamórficas pode ser aprimorada inclusive prevendo a possibilidade de testar em várias ordens diferentes, gerando vários *flow id* diferentes para um mesmo *token*.

 ©2012 SBC — Soc. Bras. de Computação

## 8. Agradecimentos

Este trabalho foi parcialmente desenvolvido com o apoio do ao Programa Nacional de Segurança Pública com Cidadania (PRONASCI) e da Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM).

## Referências

Andrew, W., Mathur, R., Chouchane, M. R. e Lakhotia, A. (2006), "Normalizing Metamorphic Malware Using Term Rewriting", Center for Advanced Computer Studies, University of Louisiana at Lafayette.

Batista, E. M. (2008) "ASAT: uma ferramenta para detecção de novos vírus", Universidade Federal de Pernambuco.

Borello, J. e Mé, L. (2008) "Code obfuscation techniques for metamorphic viruses", Journal in Computer Virology, volume 4, número 3.

Bruschi, D. Martignoni, L. Monga, M. (2007) "Code Normalization for Self-Mutating Malware" , IEEE Security & Privacy, volume 5, número 2.

Christodorescu, M. Jha, S. Kinder, J. Katzenbeisser, S. Veith, H. (2007) "Software Transformations to Improve Malware Detection", Journal in Computer Virology, número 3, páginas 253 – 265.

Hex-Rays. (2011) "IDA Pro", http://www.hex-rays.com/products/ida/index.shtml.

Kim, K. e Moon, B. (2010) "Malware Detection based on Dependency Graph using Hybrid Genetic Algorithm", Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation.

Moura, A. V. e Rebiha, R. (2009) "Automated Malware Invariant Generation", International Conference on Forensic Computer Science (ICoFCS).

Notoatmodjo, G. (2010) "Detection of Self-Mutating Computer Viruses", http://www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/gnotoadmojo.pdf, Department of Computer Science, University of Auckland, New Zealand.

OllyDbg. (2011) "OllyDbg", http://www.ollydbg.de.

Rad, B. B. e Masrom, M. (2010) "Metamorphic Virus Variants Classification Using Opcode Frequency Histogram", Latest Trends on Computers, volume 1.

Schultz, M. G. Eleazar, E. Erez, Z. Salvatore, J. S. (2001) "Data mining methods for detection of new malicious executables", Proceedings of the 2001 IEEE Symposium on Security and Privacy, páginas 38–49.

Skoudis, E. (2004) "Malware: Fighting Malicious Code", Prentice-Hall, 2004

SorceForge. (2011) "Bastard", http://sourceforge.net/projects/bastard/.

Symantec. (2007) "W32.Evol", http://www.symantec.com/security_response/writeup.jsp?docid=2000-122010-0045-99.

Schallner , M. (2004) "LIDA", http://lida.sourceforge.net/.

Virus Total. (2011) "Virus Total", http://www.virustotal.com/en/virustotalf.html.

 ©2012 SBC — Soc. Bras. de Computação

# Dynamic Detection of Address Leaks

**Gabriel Silva Quadros, Rafael M. Souza and Fernando Magno Quintão Pereira**

[1]Departamento de Ciência da Computação – UFMG
Av. Antônio Carlos, 6627 – 31.270-010 – Belo Horizonte – MG – Brazil

{gabrielquadros,rafaelms,fernando}@dcc.ufmg.br

***Abstract.*** *An address leak is a software vulnerability that allows an adversary to discover where a program is loaded in memory. Although seemingly harmless, this information gives the adversary the means to circumvent two widespread protection mechanisms: Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP). In this paper we show, via an example, how to explore an address leak to take control of a remote server running on an operating system protected by ASLR and DEP. We then present a code instrumentation framework that hinders address disclosure at runtime. Finally, we use a static analysis to prove that parts of the program do not need to be instrumented; hence, reducing the instrumentation overhead. We claim in this paper that the combination of the static and dynamic analyses provide us with a reliable and practical way to secure software against address leaks.*

***Resumo.*** *Um vazamento de endereço é uma vulnerabilidade de software que permite a um adversário descobrir em que parte da memória estão carregados os diversos módulos que compõem um programa. Embora aparentemente inócua, esse tipo de informação dá ao adversário meios para contornar dois populares mecanismos de proteção usados em sistemas operacionais: a randomização de espaços de endereçamento (ASLR) e a Prevenção de Execução de Dados (DEP). Neste artigo mostraremos, via um exemplo, como explorar vazamentos de endereço para tomar controle de um servidor remoto executando sobre um sistema operacional protegido tanto por ASLR quanto DEP. Mostraremos em seguida um arcabouço para instrumentação de programas que previne vazamentos em tempo de execução. Finalmente, nós usaremos uma análise estática de código que prova que algumas partes do programa não precisam ser instrumentadas para reduzir o custo imposto pela instrumentação. Defendemos assim, neste artigo a tese de que a combinação de análises estáticas e dinâmicas é um recurso efetivo e prático para proteger programas contra o vazamento de endereços.*

## 1. Introduction

Modern operating systems use a protection mechanism called *Address Space Layout Randomization* (ASLR) [Bhatkar et al. 2003, Shacham et al. 2004]. This technique consists in loading the binary modules that form an executable program at different addresses each time the program is executed. This security measure protects the software from well-known attacks, such as *return-to-libc* [Shacham et al. 2004] and *return-oriented-programming* (ROP) [Buchanan et al. 2008, Shacham 2007]. Because it is effective, and

 ©2012 SBC — Soc. Bras. de Computação

easy to implement, ASLR is present in virtually every contemporary operating system. However, this kind of protection is not foolproof.

Shacham *et al.* [Shacham et al. 2004] have shown that address obfuscation methods are susceptible to brute force attacks; nevertheless, address obfuscation slows down the propagation rate of worms that rely on buffer overflow vulnerabilities substantially. However, an adversary can still perform a surgical attack on an ASLR protected program. In the words of the original designers of the technique [Bhatkar et al. 2003, p.115], if "the program has a bug which allows an attacker to read the memory contents", then "the attacker can craft an attack that succeeds deterministically". It is this very type of bug that we try to prevent in this paper.

In this paper we focus on dynamic techniques to prevent address leaks. We have developed an instrumentation framework that automatically converts a program into a software that cannot contain address leaks. This transformation, which we introduce in Section 2, consists in instrumenting every program operation that propagates data. In this way, we know which information might contain address knowledge, and which information might not. If harmful information reaches an output point that an adversary can read, the program stops execution. Thus, by running the instrumented, instead of the original software, the user makes it much harder for an adversary to perform attacks that require address information to succeed.

Non-surprisingly, the instrumentation imposes on the target program a very large runtime overhead: the sanitized program can be as much as 10x slower than the original code. However, we use a static program analysis to reduce this overhead. As we explain in Section 2.1, we only instrument the pieces of code that can deal with potentially harmful data. Innocuous operations, which the static analysis identifies, are not instrumented. We are currently using the static analysis that we had previously designed and implemented to detect address disclosure vulnerabilities [Quadros and Pereira 2011]. The combination of static analysis and dynamic instrumentation lets us produce code secured against address leaks that is, on the average, less than 50% slower than the original code.

We have implemented our instrumentation framework, and the companion static analysis in the LLVM compiler [Lattner and Adve 2004]. The experiments that we describe in Section 3 show encouraging results. We have used our tool to analyze a large number of programs, and in this paper we show results for two well-known benchmark collections: FreeBench and Shootout. Instrumented programs can be 2% to 1,070% slower than the original programs, with an average slowdown of 490%. The static analysis contributes substantially to decrease this overhead. By only instrumenting the operations that the static analysis has not been able to prove safe, we get slowdowns ranging from 0% to 420%, with an average slowdown of 44%.

### 1.1. Address Leak in one Example

We illustrate the address disclosure vulnerability via the echo server seen in Listing 1. The information leak in this example let us perform a stack overflow attack on a 32-bit machine running Ubuntu 11.10, an operating system protected by ASLR and DEP. In this example, the vulnerable program keeps listening for clients at port 4000, and when a client connects, it echoes every data received. DEP hinders a buffer overflow attach in the classic Levy style [Levy 1996], because it forbids the execution of writable memory

 ©2012 SBC — Soc. Bras. de Computação

space. However, we can use a buffer overflow to divert program execution to one of `libc`'s functions. Such functions allows us to fork new processes, to send e-mails and to open socket connections, for instance. In this example we shall open a telnet terminal in the server's machine. This type of attack, usually called return-to-libc, depends on the adversary knowing the address of the target function, e.g., `libc`'s `system` in this example. This information is not easily available in a ASLR protected system, unless the target software contains an address leak.

**Listing 1. An echo server that contains an address leak.**

```
1  void *libc;
2  void process_input(char *inbuf, int len, int clientfd) {
3      char localbuf[40];
4      if (!strcmp(inbuf, "debug\n")) {
5          sprintf(localbuf, "localbuf %p\nsend() %p\n", localbuf,
6                  dlsym(libc, "send"));
7      } else { memcpy(localbuf, inbuf, len); }
8      send(clientfd, localbuf, strlen(localbuf), 0);
9  }
10 int main() {
11     int sockfd, clientfd, c_len, len;
12     char inbuf[5001];
13     struct sockaddr_in myaddr, addr;
14     libc = dlopen("libc.so", RTLD_LAZY);
15     sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
16     myaddr.sin_family = AF_INET;
17     myaddr.sin_addr.s_addr = htonl(INADDR_ANY);
18     myaddr.sin_port = htons(4000);
19     bind(sockfd, (struct sockaddr *)&myaddr, sizeof(myaddr));
20     listen(sockfd, 5);
21     c_len = sizeof(addr);
22     while (1) {
23         clientfd = accept(sockfd, (struct sockaddr *)&addr, &c_len);
24         len = recv(clientfd, inbuf, 5000, 0);
25         inbuf[len] = '\0';
26         process_input(inbuf, len + 1, clientfd);
27         close(clientfd);
28     }
29     close(sockfd); dlclose(libc);
30     return 0;
31 }
```

The information leak in our example occurs at function `process_input`. Whenever the server recognizes the special string "debug" it returns two internal addresses: the base of `localbuf`, which is a stack address, and the address of `send`, a function from `libc`. To build the exploit, we use the address of `send` to calculate the address of `system` and `exit`, two functions present in `libc`. We then use the stack address of `localbuf`'s base pointer, to find the address of `system`'s arguments. A Python script that performs this exploit is shown in Listing 2. This script makes two connections to the echo server. In the first connection it sends the string "debug" to read back the two leaked addresses. In the second connection it sends the malicious data to create a connect-back shell. The malicious data is composed of: 52 A's to fill the stack until before the return pointer; the address of `system`, calculated from the leaked address of `send`;

 ©2012 SBC — Soc. Bras. de Computação

the address of `exit`, also computed from the address of `send`; the address of the string with the command to create the connect-back shell, calculated from `localbuf`'s base pointer; and finally the string containing the command to create the shell. By overwriting the return address of `process_input` with the address of `system` we gain control of the remote machine. By calling `exit` at the end of the exploit we ensure that our client terminates quietly after giving us a shell.

**Listing 2. A Python script that exploits the echo server.**

```
1  import socket
2  import struct
3  c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  c.connect(('localhost', 4000))
5  buf = "debug\n"
6  c.send(buf)
7  buf = c.recv(512)
8  leaked_stack_addr = int(buf[9:buf.find('\n')], 16)
9  leaked_send_addr = int(buf[27:buf.rfind('\n')], 16)
10 c.close()
11 c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12 c.connect(('localhost', 4000))
13 command = ('rm -f backpipe && mknod backpipe p &&'
14         'telnet localhost 8080 0<backpipe | /bin/bash 1>backpipe\x00')
15 command_addr = leaked_stack_addr + 64
16 system_addr = leaked_send_addr - 0x96dd0  # system()
17 system_ret_addr = system_addr - 0xa140 # exit()
18 buf = ('A' * 52 + struct.pack('I', system_addr) +
19         struct.pack('I', system_ret_addr) +
20         struct.pack('I', command_addr) + command)
21 c.send(buf)
22 c.close()
```

## 2. The Proposed Solution

In this section we describe our solution to detect address leaks at runtime. We start by defining a core language containing the constructs from imperative languages that play a role in the address disclosure vulnerability. On top of this language we define an *instrumentation language*. Programs implemented with the instrumentation syntax can track the flow of information at runtime. We then proceed to describe a type system that detects address leaks statically. This type system let us reduce the amount of instrumentation necessary to safe-guard programs against address leaks.

**Angels: the Subject Language.** We define a toy language, which we call *Angels*, to explain our approach to dynamic detection of address leaks. Angels is an assembly-like language, whose syntax is given in Figure 1. This language has six instructions that deal with the computation of data, and three instructions that change the program flow. The six data related instructions represent constructs typical of actual C or C++ programs, as the table in Figure 2 illustrates. We use `adr` to model language constructs that read the address of a variable, namely the ampersand (&) operator and memory allocation functions such as `malloc`, `calloc` or `realloc`. Simple assignments are represented via the instruction `mov`. We represent binary operations via the `add` instruction, which sums up two variables and dumps the result into a third location. Loads to and stores from memory

 ©2012 SBC — Soc. Bras. de Computação

| | | |
|---|---|---|
| (Variables) | ::= | $\{v_1, v_2, \ldots\}$ |
| (Data Producing Instructions) | ::= | |
| – (Get variable's address) | \| | $\texttt{adr}(v_1, v_2)$ |
| – (Assign to variable) | \| | $\texttt{mov}(v_1, v_2)$ |
| – (Binary addition) | \| | $\texttt{add}(v_1, v_2, v_3)$ |
| – (Store into memory) | \| | $\texttt{stm}(v_0, v_1)$ |
| – (Load from memory) | \| | $\texttt{ldm}(v_1, v_0)$ |
| – (Print the variable's value) | \| | $\texttt{out}(v)$ |
| (Control flow Instructions) | ::= | |
| – (Branch if zero) | \| | $\texttt{bzr}(v, l)$ |
| – (Unconditional jump) | \| | $\texttt{jmp}(l)$ |
| – (Halt execution) | \| | $\texttt{end}$ |
| ($\phi$-function - data selector) | \| | $\texttt{phi}(v, \{v_1 : l_1, \ldots, v_k : l_k\})$ |

**Figure 1. The syntax of Angels.**

are modeled by `ldm` and `stm`. Finally, we use `out` to denote any instruction that gives information away to an external user. This last instruction represents not only ordinary printing operations, but also native function interfaces. For instance, a JavaScript program usually relies on a set of native functions to interact with the browser. A malicious user could use this interface to obtain an internal address from the JavaScript interpreter.

We work with programs in the Static Single Assignment (SSA) form [Cytron et al. 1991]. This intermediate representation has the key property that every variable name has only one definition site in the program code. To ensure this invariant, the SSA intermediate representation uses $\phi$-functions, a special notation that does not exist in usual assembly languages. Figure 1 shows the syntax of $\phi$-functions. This representation is not necessary for computational completeness; however, as we will see in Section 2.1, it simplifies the static analysis of programs. Additionally, the SSA format is used by our baseline compiler, LLVM, and many other modern compilers, including gcc. Thus, by adopting this representation we shrink the gap between our abstract formalism and its concrete implementation.

Figure 3 describes the small-steps operational semantics of Angels. We let an abstract machine be a five-element tuple $\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta \rangle$. We represent the program $P$ as a map that associates integer values, which we shall call *labels*, with instructions. We let `pc` be the current *program counter*, and we let `pc`' be the program counter seen by the last instruction processed. We need to keep track of the previous program counter to model $\phi$-functions. These instructions are used like variable multiplexers. For instance, $v = \phi(v_1 : l_1, v_2 : l_2)$ copies $v_1$ to $v$ if control comes from $l_1$, and copies $v_2$ to $v$ if control comes from $l_2$. The previous program counter points out to us the path used to reach the $\phi$-function. $\Sigma$, the memory, is an environment that maps variables to integer values, and $\Theta$ is an output channel. We use the notation $f[a \mapsto b]$ to denote the updating of function $f$; that is, $\lambda x.x = a \ ? \ b : f(x)$. For simplicity we do not distinguish a memory of local variables, usually called *stack*, from the memory of values that out-live the functions that have created them, usually called *heap*. We use a function $\Delta$ to map the names of

 ©2012 SBC — Soc. Bras. de Computação

| | |
|---|---|
| `v1 = &v2` | $\texttt{adr}(v_1, v_2)$ |
| `v1 = (int*)malloc(sizeof(int))` | $\texttt{adr}(v_1, v_2)$ where<br>$v_2$ is a fresh memory location |
| `v1 = *v0` | $\texttt{ldm}(v_1, v_0)$ |
| `*v0 = v1` | $\texttt{stm}(v_0, v_1)$ |
| `*v1 = *v0` | $\texttt{ldm}(v_2, v_0)$, where $v_2$ is fresh<br>$\texttt{stm}(v_1, v_2)$ |
| `v1 = v2 + v3` | $\texttt{add}(v_1, v_2, v_3)$ |
| `*v = v1 + &v2` | $\texttt{adr}(v_3, v_2)$, where $v_3$ is fresh<br>$\texttt{add}(v_4, v_1, v_3)$, where $v_4$ is fresh<br>$\texttt{stm}(v, v_4)$ |
| `f(v1, &v3)`, where `f` is declared<br>as `f(int v2, int* v4);` | $\texttt{mov}(v_2, v_1)$<br>$\texttt{adr}(v_4, v_3)$ |

**Figure 2. Examples of syntax of C mapped to instructions of Angels.**

variables to their integer addresses in $\Sigma$. We represent the output channel as a list $\Theta$. As we see in Rule [OUTSEM], the only instruction that can manipulate this list, by *consing* a value on it, is the `out` instruction. We denote consing by the operator $::$, as in ML and Ocaml.

Angels is a Turing Complete programming language. Given an infinite surplus of variables, we can implement a Turing Machine on it. Figure 4 shows an example of a program written in Angels. The program in Figure 4(a) prints the contents of an array, and then the base address of the array itself. Figure 4(b) shows the equivalent Angels program. We have outlined the label of the first instruction present in each basic block of this example. We have also marked in gray the initial memory used in this example. Angels does not have instructions to load constants into variables; however, we compensate this omission by starting programs with a non-empty memory.

The program in Figure 4 contains an address disclosure vulnerability, a notion that we introduce in Definition 2.1. An Angels program $P$ contains such a vulnerability if an adversary can reconstruct the map $\Delta$ for at least one variable that $P$ uses. Notice that we are not assuming that $P$ terminates. We only require that the program outputs any information that an adversary can read. Our running example contains an address leak vulnerability, because it prints the base address of the array $f$. Thus, by reading the output channel, the adversary would be able to find $\Delta(f)$.

**Definition 2.1** THE ADDRESS DISCLOSURE VULNERABILITY. *An Angels program $P$, such that $\langle P, 0, \lambda x.0, [\,]\rangle \rightarrow \langle \Sigma, \Theta\rangle$, where $\lambda x.0$ is the environment that maps every variable to zero, and $[\,]$ is the empty output channel, contains an address disclosure vulnerability if it is possible for an adversary to discover $\Delta(v)$ for some $v$ used in $P$, given the knowledge of $\Theta$ plus the source code of $P$.*

**The Instrumented language.** In order to secure a program against address disclosures,

 ©2012 SBC — Soc. Bras. de Computação

$$[\textsc{AdrSem}]\quad \frac{\Delta(v_2) = n \qquad \Sigma' = \Sigma[\Delta(v_1) \mapsto n]}{\langle \texttt{adr}(v_1, v_2), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle} \qquad [\textsc{OutSem}]\quad \frac{\Sigma[\Delta(v)] = n \qquad \Theta' = n :: \Theta}{\langle \texttt{out}(v), \Sigma, \Theta\rangle \to \langle \Sigma, \Theta'\rangle}$$

$$[\textsc{MovSem}]\quad \frac{\Sigma[\Delta(v_2)] = n \qquad \Sigma' = \Sigma[\Delta(v_1) \mapsto n]}{\langle \texttt{mov}(v_1, v_2), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle}$$

$$[\textsc{AddSem}]\quad \frac{\Sigma[\Delta(v_2)] = n_2 \qquad \Sigma[\Delta(v_3)] = n_3 \qquad \Sigma' = \Sigma[\Delta(v_1) \mapsto n_2 + n_3]}{\langle \texttt{add}(v_1, v_2, v_3), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle}$$

$$[\textsc{StmSem}]\quad \frac{\Sigma[\Delta(v_0)] = x \qquad \Sigma[\Delta(v_1)] = n \qquad \Sigma' = \Sigma[x \mapsto n]}{\langle \texttt{stmem}(v_0, v_1), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle}$$

$$[\textsc{LdmSem}]\quad \frac{\Sigma[\Delta(v_0)] = x \qquad \sigma[x] = n \qquad \Sigma' = \Sigma[\Delta(v_1) \mapsto n]}{\langle \texttt{ldmem}(v_1, v_0), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle}$$

$$[\textsc{EndSem}]\quad \frac{P[\texttt{pc}] = \texttt{end}}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma, \Theta\rangle}$$

$$[\textsc{JmpSem}]\quad \frac{P[\texttt{pc}] = \texttt{jmp}(l) \qquad \langle P, \texttt{pc}, l, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}$$

$$[\textsc{BzrSem}]\quad \frac{P[\texttt{pc}] = \texttt{bzr}(v, l) \qquad \Sigma[\Delta(v)] \neq 0 \qquad \langle P, \texttt{pc}, \texttt{pc} + 1, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}$$

$$[\textsc{BnzSem}]\quad \frac{P[\texttt{pc}] = \texttt{bzr}(v, l) \qquad \Sigma[\Delta(v)] = 0 \qquad \langle P, \texttt{pc}, l, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle}$$

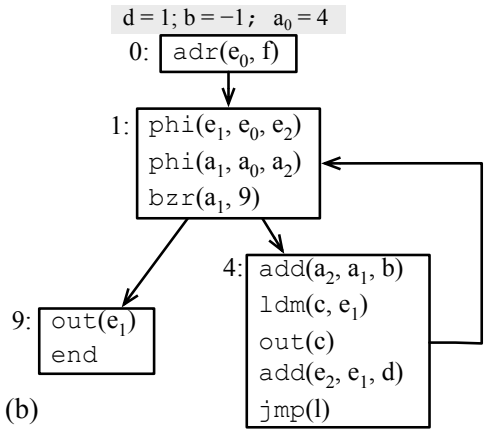
$$[\textsc{PhiSem}]\quad \frac{\begin{array}{c} P[\texttt{pc}] = \texttt{phi}(v, \{v_1 : l_1, \ldots, v_k : l_k\}) \\ \texttt{pc}' = l_i \qquad \langle \texttt{mov}(v, v_i), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle \qquad \langle P, \texttt{pc}, \texttt{pc} + 1, \Sigma', \Theta\rangle \to \langle \Sigma'', \Theta'\rangle \end{array}}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta'\rangle}$$

$$[\textsc{SeqSem}]\quad \frac{P[\texttt{pc}] \notin \{\texttt{bzr}, \texttt{end}, \texttt{phi}, \texttt{jmp}\} \qquad \langle P[\texttt{pc}], \Sigma, \Theta\rangle \to \langle \Sigma', \Theta'\rangle \qquad \langle P, \texttt{pc}, \texttt{pc} + 1, \Sigma', \Theta'\rangle \to \langle \Sigma'', \Theta''\rangle}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta''\rangle}$$

**Figure 3. The Operational Semantics of Angels.**

```
int main() {
    int i = 0, f[4];
    for (i = 0; i < 4; i++) {
        printf("%d\n", f[i]);
    }
    printf("%u\n", f);
}
```

(a)




(b)

**Figure 4. A C program translated to Angels.**

we will instrument it. In other words, we will replace its original sequence of instructions by other instructions, which track the flow of values at runtime. It is possible to instrument an Angels program using only Angels instructions. However, to perform the instrumen-

 ©2012 SBC — Soc. Bras. de Computação

$$[\textsc{AdrIns}]\quad \frac{\langle \texttt{adr}, \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle \qquad \Delta(v_1) = n_1 \qquad \Sigma'' = \Sigma'[n_1 + D \mapsto \mathit{tainted}]}{\langle \texttt{sh_adr}(v_1, v_2), \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta\rangle}$$

$$[\textsc{OutIns}]\quad \frac{\Delta(v) = n_v \qquad \Sigma[n_v + D \mapsto \mathit{clean}] \qquad \langle \texttt{out}(v), \Sigma, \Theta\rangle \to \langle \Sigma, \Theta'\rangle}{\langle \texttt{sh_out}(v), \Sigma, \Theta\rangle \to \langle \Sigma, \Theta'\rangle}$$

$$[\textsc{MovIns}]\quad \frac{\langle \texttt{mov}(v_1, v_2), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle \qquad \Sigma'[\Delta(v_2) + D] = t \qquad \Sigma'' = \Sigma'[\Delta(v_1) + D \mapsto t]}{\langle \texttt{sh_mov}(v_1, v_2), \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta\rangle}$$

$$[\textsc{AddIns}]\quad \frac{\begin{array}{c}\langle \texttt{add}(v_1, v_2, v_3), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle \\ \Sigma'[\Delta(v_2) + D] = t_2 \qquad \Sigma'[\Delta(v_3) + D] = t_3 \qquad t_1 = t_2 \sqcap_{sh} t_3 \qquad \Sigma'' = \Sigma[\Delta(v_1) + D \mapsto t]\end{array}}{\langle \texttt{sh_add}(v_1, v_2, v_3), \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta\rangle}$$

$$[\textsc{StmIns}]\quad \frac{\Sigma[\Delta(v_0)] = x \qquad \Sigma[\Delta(v_1)] = n \qquad \Sigma' = \Sigma[x \mapsto n] \qquad \Sigma'[\Delta(v_1) + D] = t \qquad \Sigma'' = \Sigma'[x + D \mapsto t]}{\langle \texttt{sh_stm}(v_0, v_1), \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta\rangle}$$

$$[\textsc{LdmIns}]\quad \frac{\Sigma[\Delta(v_0)] = x \qquad \Sigma[x] = n \qquad \Sigma' = \Sigma[\Delta(v_1) \mapsto n] \qquad \Sigma'[x + D] = t \qquad \Sigma'' = \Sigma'[\Delta(v_1) + D \mapsto t]}{\langle \texttt{sh_ldm}(v_1, v_0), \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta\rangle}$$

$$[\textsc{PhiIns}]\quad \frac{\begin{array}{c}P[\texttt{pc}] = \texttt{sh_phi}(v, \{v_1 : l_1, \ldots, v_k : l_k\}) \\ \texttt{pc}' = l_i \qquad \langle \texttt{sh_mov}(v, v_i), \Sigma, \Theta\rangle \to \langle \Sigma', \Theta\rangle \qquad \langle P, \texttt{pc}, \texttt{pc} + 1, \Sigma', \Theta\rangle \to \langle \Sigma'', \Theta'\rangle\end{array}}{\langle P, \texttt{pc}', \texttt{pc}, \Sigma, \Theta\rangle \to \langle \Sigma'', \Theta'\rangle}$$

**Figure 5. The Operational Semantics of the instrumented instructions.**

tation in this way we would have to make this presentation unnecessarily complicated. To avoid this complexity we define a second set of instructions, whose semantics is given in Figure 5. The instrumentation framework defines an equivalent shadow instruction for `out` and for each instruction that can update the memory $\Sigma$. To hold the meta-data produced by the instrumentation we create a *shadow memory*. For each variable $v$, stored at $\Sigma[\Delta(v)]$, we create a new location $\Sigma[\Delta(v) + D]$, where $D$ is a displacement that separates each variable from its shadow. In other words, information about shadow and original variables are kept into the same store. The shadow values can be bound to one of two values, *clean* or *tainted*. Instrumented programs are evaluated by the same rules seen in Figure 3, augmented with the six new rules given in Figure 5. It is important to notice that the instrumented programs might terminate prematurely. Rule OUTINS does not progress if we try to print a variable that is shadowed with the tainted value. The meet operator used in the definition of `sh_add`, $\sqcap_{sh}$ is such that $n_1 \sqcap_{sh} n_2 = \mathit{tainted}$ whenever one of them is *tainted*, and is $\mathit{clean}$ otherwise.

Figure 6 defines a relation $\iota$ that converts an ordinary Angels program into an instrumented program. Every instruction that can store data into the memory is instrumented; thus, we call this relation the *full instrumentation framework*. Notice that the instructions that control the execution flow are not instrumented. They do not need to be instrumented because neither of them generates new data in the store.

**Theorem 2.2** *Let $P$ be an angels program, and $P^\iota$ be such that $P \xrightarrow{\iota} P^\iota$. Let $I \in P$ and $I^\iota \in P^\iota$ such that $I \xrightarrow{\iota} I^\iota$. If a variable $v$ is used at an instruction $I \in P$, and $v$ is data dependent on some address, then $\Sigma[\Delta(v) + D]$ contains the value* tainted *when $v$ is used at $I^\iota$.*

**Theorem 2.3** *If $P$ is an angels program, and $P^\iota$ is such that $P \xrightarrow{\iota} P^\iota$, then $\langle P^\iota, 0, 0, \lambda x.0, [\,]\rangle$ does not contain an address disclosure vulnerability.*

 ©2012 SBC — Soc. Bras. de Computação

$$
\begin{array}{rcl@{\qquad}rcl}
\texttt{adr}(v_1, v_2) & \xrightarrow{\iota} & \texttt{sh_adr}(v_1, v_2) & \texttt{stm}(v_0, v_1) & \xrightarrow{\iota} & \texttt{sh_stm}(v_0, v_1) \\
\texttt{out}(v) & \xrightarrow{\iota} & \texttt{sh_out}(v) & \texttt{ldm}(v_0, v_1) & \xrightarrow{\iota} & \texttt{sh_ldm}(v_0, v_1) \\
\texttt{mov}(v_1, v_2) & \xrightarrow{\iota} & \texttt{sh_mov}(v_1, v_2) & \texttt{bzr}(v, l) & \xrightarrow{\iota} & \texttt{bzr}(v, l) \\
\texttt{add}(v_1, v_2, v_3) & \xrightarrow{\iota} & \texttt{sh_add}(v_1, v_2, v_3) & \texttt{jmp}(l) & \xrightarrow{\iota} & \texttt{jmp}(l) \\
\texttt{phi}(v, \{.., v_i : l_i, ..\}) & \xrightarrow{\iota} & \texttt{sh_phi}(v, \{.., v_i : l_i, ..\}) & \texttt{end} & \xrightarrow{\iota} & \texttt{end}
\end{array}
$$

**Figure 6. The full instrumentation of an angels program.**

### 2.1. Using Static Analysis to Reduce the Instrumentation Overhead.

As we will show empirically, a fully instrumented program is considerably slower than the original program. In order to decrease this overhead, we couple the dynamic instrumentation with a static analysis that detects address leaks. We have developed this static analysis in a previous work [Quadros and Pereira 2011]. This static analysis is a form of tainted flow analysis [Schwartz et al. 2010]. Thus, the objective of the static analysis is to find a path from a source of sensitive data, i.e., an `adr` instruction, to a output channel, i.e., an `out` instruction. Paths are formed by *dependency relations*. If a variable $v_1$ is defined by an instruction $i$ that uses a variable $v_2$, then we say that $v_1$ depends on $v_2$. As an example, $v_1$ depends on $v_2$ if the instruction $\texttt{mov}(v_1, v_2)$ is in the program.

We call the graph formed by all the dependencies in the program the *dependence graph*. We say that a variable $v$ is *tainted* if: (i) $v$ transitively depends on some variable $v_1$ created by an `adr` instruction; and (ii) there is some variable $v_2$ used in an `out` instruction that depends on $v$. Our static analysis [Quadros and Pereira 2011] determines the set of program variables that are tainted.

**Partial Instrumentation.** Given the result of our static analysis, we can reduce the amount of instrumented code in the target program that the $\iota$ relation from Figure 6 inserts. With such purpose, we define a new instrumentation framework as the $\gamma$ relation, which is given in Figure 7. Contrary to the algorithm in Figure 6, this time we only change an instruction if (i) it defines a variable that has the *tainted* type; or (ii) it is the output operation, and uses a tainted variable. Theorem 2.4 proves that the partial instrumentation is correct. For the sake of space, we omit from Figure 7 the rules that do not change the target instruction. An instruction might remain unchanged in the instrumented program, either because this instruction has been deemed safe by the static analysis, or because it does not create values in memory.

**Theorem 2.4** *If $P$ is an angels program, and $P^\gamma$ is such that $P \xrightarrow{\gamma} P^\gamma$, then $\langle P^\gamma, 0, 0, \lambda x.0, [\,]\rangle$ does not contain an address leak.*

Figure 8 shows the result of applying the static analysis onto the program seen in Figure 4(b). Figure 8(a) shows the dependence graph that the target program induces. Each variable in this graph has been annotated with the abstract state that our static analysis infers to it. Thus, a variable may have been assigned the type *clean* ($C$) or *tainted* ($T$). In this case, the variables $e_0, e_1$ and $e_2$ have been marked *tainted* ($T$). The partial instrumentation framework changes the instructions that either define these variables, or output them. Figure 8(b) shows the result of this transformation. The figure shows that only four, out of 11 instructions have been instrumented. This partially instrumented program is certainly more efficient than the fully instrumented program.

 ©2012 SBC — Soc. Bras. de Computação

$$\Gamma \vdash v_1 = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{adr}(v_1, v_2), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_adr}(v_1, v_2)$$

$$\Gamma \vdash v = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{out}(v), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_out}(v)$$

$$\Gamma \vdash v_1 = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{mov}(v_1, v_2), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_mov}(v_1, v_2)$$

$$\Gamma \vdash v_1 = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{add}(v_1, v_2, v_2), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_mov}(v_1, v_2, v_3)$$

$$\exists x \in \Pi[v_0], \Gamma \vdash x = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{stm}(v_0, v_1), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_stm}(v_0, v_1)$$

$$\Gamma \vdash v_1 = \textit{tainted} \quad \Rightarrow \quad \langle \texttt{ldm}(v_1, v_0), \Gamma, \Pi \rangle \xrightarrow{\gamma} \texttt{sh_ldm}(v_1, v_0)$$

**Figure 7. The relation $\gamma$ that partially instruments Angels programs.**

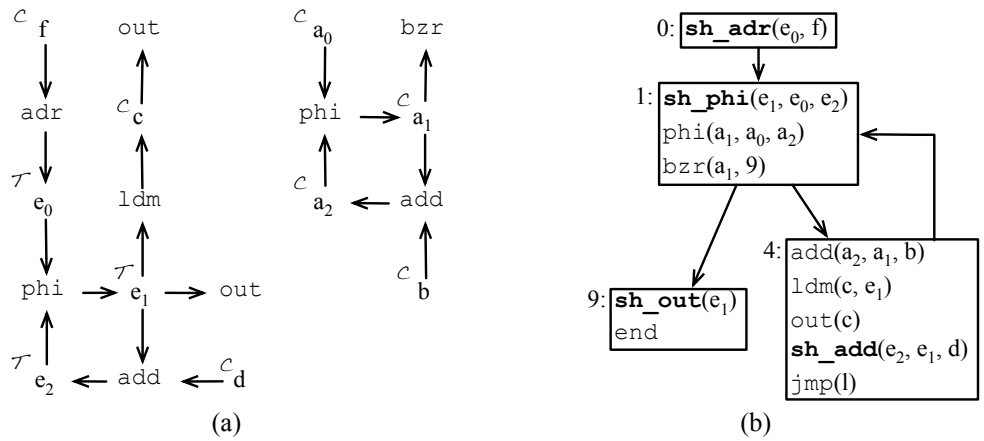



**Figure 8. (a) The result of the static analysis applied on the program seen in Figure 4(a). (b) Partially instrumented program.**

## 3. Experiments

We have implemented our algorithm on top of the LLVM compiler [Lattner and Adve 2004], and have tested it in an Intel Core 2 Duo Processor with a 2.20GHz clock, and 2 GB of main memory on a 667 MHz DDR2 bus. The operating system is Ubuntu 11.04.

**The benchmarks:** We have run our instrumentation framework on programs taken from two public benchmarks. The first, `FreeBench` (FB) [1], is used to measure system's workloads. This benchmark consists of six applications that include cryptography algorithms, neural networks and data compression. The second benchmark suite, shootout (ST) [2], consists of eight programs used in a popular website that compares different programming languages. Some characteristics of these benchmarks are given by Figure 9. In this section we will consider that an address leaks to the outside world if it is used as an ar-

[1] `http://code.google.com/p/freebench/`
[2] `http://shootout.alioth.debian.org/`

 ©2012 SBC — Soc. Bras. de Computação

| Benchmark | Application | LoC | Sinks | Sources | Assembly |
|---|---|---|---|---|---|
| ST | `fasta` | 133 | 0 | 23 | 218 |
| ST | `fannkuch` | 104 | 3 | 27 | 281 |
| ST | `n-body` | 141 | 9 | 49 | 463 |
| ST | `nsieve-bits` | 35 | 5 | 7 | 130 |
| ST | `partialsums` | 67 | 1 | 38 | 212 |
| ST | `puzzle` | 66 | 1 | 12 | 210 |
| ST | `recursive` | 54 | 1 | 6 | 174 |
| ST | `spectral-norm` | 52 | 2 | 20 | 258 |
| FB | `distray` | 447 | 8 | 260 | 1,379 |
| FB | `fourinarow` | 702 | 1 | 109 | 2,124 |
| FB | `mason` | 385 | 11 | 303 | 690 |
| FB | `neural` | 785 | 16 | 156 | 1,425 |
| FB | `pcompress2` | 903 | 14 | 180 | 1,606 |
| FB | `pifft` | 4,185 | 3 | 1,857 | 19,320 |

**Figure 9. Characteristics of the benchmarks. LoC: number of lines of C. Assembly: number of instructions in the binary representation of the original program.**

gument of the standard `printf(char*, ...)` function. Thus, "sinks", in Figure 9 is the total number of occurrences of the `printf` function in the text of the application. Sources are instructions that produce address information, such as the ampersand (&) operator.

**The results of the static analysis:** Figure 10 shows the results of our static analysis. We compare the size of the constraint graph that the static analysis builds for each program with the fraction of that graph that has been marked as tainted. The tainted nodes belong into a path from a source function, e.g., code that generates address information, to a sink function, e.g., code that disclosures address information. As it is possible to see, about one fourth of each constraint graph is marked as tainted. In other words, the program slice that must be instrumented corresponds to about one fourth of the original program.

**Instrumentation overhead:** Figure 11 compares the runtime overhead of full and partial instrumentation. We have fed each program – original, fully instrumented and partially instrumented – with the reference input of each benchmark. As we can see, the full instrumentation imposes a non-trivial overhead on the programs. On average, fully instrumented programs are 4.9x, e.g., $(74.77secs/15.25secs)$ slower than the original programs. The partially instrumented programs are only 1.4x, e.g., $(22.04secs/15.25secs)$ slower. However, there are some cases, such as `pifft` and `puzzles`, in which the partially instrumented programs are considerably slower. These large slowdowns happen because the instrumentation code is inserted inside critical paths in the program code, such as deeply nested loops.

## 4. Related Work

We believe that this paper describes the first attempt to detect dynamically address disclosure vulnerabilities. However, much work has been done to detect dynamically other types of vulnerabilities. For instance, in 2005, James Newsome designed and imple-

 ©2012 SBC — Soc. Bras. de Computação

| Application | CG | TG | %Tainted |
|---|---|---|---|
| fasta | 91 | 23 | 25.27% |
| fannkuch | 118 | 27 | 22.88% |
| n-body | 171 | 49 | 28.65% |
| nsieve-bits | 70 | 7 | 10.00% |
| partialsums | 192 | 38 | 19.79% |
| puzzle | 80 | 25 | 31.25% |
| recursive | 91 | 6 | 6.59% |
| spectral-norm | 121 | 20 | 16.53% |
| distray | 993 | 260 | 26.18% |
| fourinarow | 1242 | 109 | 8.78% |
| mason | 704 | 303 | 43.04% |
| neural | 855 | 156 | 18.25% |
| pcompress2 | 929 | 180 | 19.38% |
| pifft | 7669 | 1901 | 24.79% |

**Figure 10. Data produced by the static analysis. CG: number of nodes in the program's constraint graph. TG: number of tainted nodes in the constraint graph. %Tainted = TG/CG.**

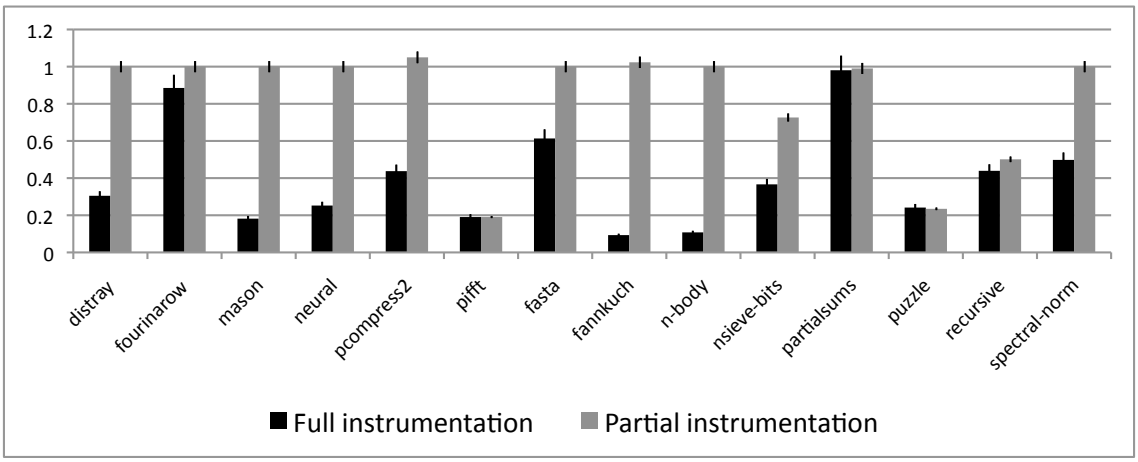



**Figure 11. Runtime of the instrumented program (full or partial) divided by the runtime of the original program. The shorter the bar, the slower is the program.**

mented a dynamic instrumentation framework that, contrary to previous approaches, was able to generate very efficient code [Newsome and Song 2005]. Newsome has been able to achieve this efficiency by applying different optimizations on the instrumented code. One of these optimizations, *fast path*, consists in duplicating the program code, in such a way that non-instrumented code is executed if the program inputs are not tainted. Similarly, Clause *et al.* [Clause et al. 2007] have proposed an instrumentation library that can be customized to track different forms of data. Clause *et al.*'s tool is not as efficient as the framework proposed by Newsome; however, it is much more general. None of these previous work has tracked address leaks. They also did not use static analysis to mitigate the instrumentation overhead.

There are many previous works that have proposed to combine static and dy-

 ©2012 SBC — Soc. Bras. de Computação

namic analysis to secure programs. Huang *et al.*, for instance, have used static analysis to reduce the amount of runtime checks necessary to guard PHP programs against SQL injection attacks [Huang et al. 2004]. Similar approaches have been proposed by Zhang *et al.* [Zhang et al. 2011], Balzarotti *et al.* [Balzarotti et al. 2008] and Keromytis *et al.* [Keromytis et al. 2011]. None of these previous works tracks address leaks.

Finally, there is a large body of literature related to the use of pure static analysis to detect information flow vulnerabilities [Rimsa et al. 2011, Wassermann and Su 2007, Xie and Aiken 2006]. These works deal mostly with information flow in dynamically typed languages, such as PHP. The main problem of concern is to detect tainted flow vulnerabilities. We say that a program contains a tainted flow vulnerability if an adversary can feed it with malicious data, and this data reaches a sensitive function. Contrary to us, none of these previous works deal with address leaks, neither try to instrument the program to secure it dynamically.

## 5. Conclusion

This paper has presented a framework that tracks information leaks at runtime. This framework instruments the source code of the target program, shadowing every program variable and memory location that it uses. These shadow values might be in one of two states: clean or tainted. If the instrumented program tries to output a value shadowed as tainted, then we fire an exception, and the program terminates, providing a log back to the end user. This type of source code instrumentation imposes a heavy burden on the target program. In order to avoid this overhead, we rely on a static analysis that proves that parts of the program do not need to be instrumented. This static analysis has been modeled as the combination of two type inference engines: the first propagates tainted information forwardly, the second propagates information backwards. We have show empirically that this static analysis is effective, being able to reduce the instrumentation overhead from 5 to 1.44 times.

**Future works:** Although we are able to correctly instrument large programs, our framework is still a research artifact. For instance, some partially instrumented programs are still much slower than the original programs. Our next goal is to decrease this overhead. We want to apply program optimizations on the instrumented code, such as Qin's style code coalescing [Qin et al. 2006], a technique that has been developed to improve the runtime of binary instrumentation frameworks.

**Reproducibility:** All the benchmarks used in Section 3 are publicly available at their websites. The LLVM compiler is open source. Our code and further material about this project, including a Prolog interpreter of Angels, is available at `http://code.google.com/p/addr-leaks/`.

## References

Balzarotti, D., Cova, M., Felmetsger, V., Jovanovic, N., Kirda, E., Kruegel, C., and Vigna, G. (2008). Saner: Composing static and dynamic analysis to validate sanitization in web applications. In *SP*, pages 387–401. IEEE Computer Society.

Bhatkar, E., Duvarney, D. C., and Sekar, R. (2003). Address obfuscation: an efficient approach to combat a broad range of memory error exploits. In *USENIX Security*, pages 105–120.

 ©2012 SBC — Soc. Bras. de Computação

Buchanan, E., Roemer, R., Shacham, H., and Savage, S. (2008). When good instructions go bad: generalizing return-oriented programming to RISC. In *CCS*, pages 27–38. ACM.

Clause, J., Li, W., and Orso, A. (2007). Dytan: a generic dynamic taint analysis framework. In *ISSTA*, pages 196–206. ACM.

Cytron, R., Ferrante, J., Rosen, B. K., Wegman, M. N., and Zadeck, F. K. (1991). Efficiently computing static single assignment form and the control dependence graph. *TOPLAS*, 13(4):451–490.

Huang, Y.-W., Yu, F., Hang, C., Tsai, C.-H., Lee, D.-T., and Kuo, S.-Y. (2004). Securing web application code by static analysis and runtime protection. In *WWW*, pages 40–52. ACM.

Keromytis, A. D., Stolfo, S. J., Yang, J., Stavrou, A., Ghosh, A., Engler, D., Dacier, M., Elder, M., and Kienzle, D. (2011). The minestrone architecture combining static and dynamic analysis techniques for software security. In *SYSSEC*, pages 53–56. IEEE Computer Society.

Lattner, C. and Adve, V. S. (2004). LLVM: A compilation framework for lifelong program analysis & transformation. In *CGO*, pages 75–88. IEEE.

Levy, E. (1996). Smashing the stack for fun and profit. *Phrack*, 7(49).

Newsome, J. and Song, D. X. (2005). Dynamic taint analysis for automatic detection, analysis, and signaturegeneration of exploits on commodity software. In *NDSS*. USENIX.

Qin, F., Wang, C., Li, Z., Kim, H.-s., Zhou, Y., and Wu, Y. (2006). LIFT: A low-overhead practical information flow tracking system for detecting security attacks. In *MICRO*, pages 135–148. IEEE.

Quadros, G. S. and Pereira, F. M. Q. (2011). Static detection of address leaks. In *SBSeg*, pages 23–37.

Rimsa, A. A., D'Amorim, M., and Pereira, F. M. Q. (2011). Tainted flow analysis on e-SSA-form programs. In *CC*, pages 124–143. Springer.

Schwartz, E. J., Avgerinos, T., and Brumley, D. (2010). All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). In *S&P*, pages 1–15. IEEE.

Shacham, H. (2007). The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86). In *CCS*, pages 552–561. ACM.

Shacham, H., Page, M., Pfaff, B., Goh, E.-J., Modadugu, N., and Boneh, D. (2004). On the effectiveness of address-space randomization. In *CSS*, pages 298–307. ACM.

Wassermann, G. and Su, Z. (2007). Sound and precise analysis of web applications for injection vulnerabilities. In *PLDI*, pages 32–41. ACM.

Xie, Y. and Aiken, A. (2006). Static detection of security vulnerabilities in scripting languages. In *USENIX-SS*. USENIX Association.

Zhang, R., Huang, S., Qi, Z., and Guan, H. (2011). Combining static and dynamic analysis to discover software vulnerabilities. In *IMIS*, pages 175–181. IEEE Computer Society.

 ©2012 SBC — Soc. Bras. de Computação

# SDA-COG – Sistema de Detecção de Ataques para Rede de Rádios Cognitivos*

**Joffre Gavinho Filho[1], Luiz F. R. C. Carmo[2], Raphael C. S. Machado[2], Luci Pirmez[1]**

[1]NCE/IM Universidade Federal do Rio de Janeiro (UFRJ) – Rio e Janeiro, RJ – Brasil

[2]Instituto Nacional de Metrologia (INMETRO) – Duque de Caxias, RJ - Brasil

{joffreufrj,luci.pirmez}@gmail.comr, {rust,rmachado}@inmetro.gov.br

***Abstract.*** *This paper deals with the development of an Attack Detection System (ADS) for Cognitive Radio Networks (CRN) formed by combining two classic intrusion detection mechanisms: location and reputation. The proposal was evaluated by means of simulations of specific CRNs attacks: Primary User Emulation (PUE) and Sense Spectrum False Feedback (SSFF).*

***Resumo.*** *Este trabalho descreve um Sistema de Detecção de Ataques (SDA) para redes de rádios cognitivos a partir da integração dos métodos de detecção por Localização e por Reputação. O sistema é validado através de simulações que avaliam o desempenho do mesmo face aos ataques específicos de emulação do usuário primário (Primary User Emulation - PUE) e falso diagnóstico do sensoriamento do espectro (Sense Spectrum False Feedback - SSFF).*

## 1. Introdução

A utilização do espectro de freqüência na área de comunicações sem-fio é concedida em seu direito de uso por meio de uma licença (denominada de concessão), fornecida pelos órgãos governamentais responsáveis pela regulamentação e fiscalização das comunicações [Inatel 2009]. O modelo de alocação de bandas do espectro de freqüência no Brasil permite os seguintes tipos de concessões: (i) *Licenciado exclusivo* - direito de uso exclusivo da banda ou canal assegurado por um órgão fiscalizador do espectro por um período pré-determinado, sujeito às limitações da licença; (ii) *Licenciado não exclusivo* - utiliza partes do espectro de frequência onde são concedidas licenças para mais de um usuário, nenhuma entidade tem o controle total desta parte do espectro; e (iii) *Não-licenciado* - utiliza o espectro de frequência sem a necessidade de se obter uma licença e não há um único usuário com direito de uso exclusivo. Medições do espectro de frequência [McHenry 2003] demonstram que a política de alocação estática do espectro é imprópria para o atual cenário de comunicações em redes sem-fio. De acordo com o relatório do FCC (Federal Communications Commission), que é o órgão responsável pelas comunicações nos Estados Unidos [Fcc 2003], a maioria das bandas de espectro atribuídas (bandas licenciadas) não é usada em certos períodos de tempo e/ou em determinadas áreas geográficas, ocasionando o denominado "espaço em branco" (*white space*). Em contrapartida, as bandas de frequências não licenciadas encontram-se saturadas em virtude de sua maciça utilização. Uma forma eficiente para resolver a contradição entre as faixas licenciadas subutilizadas e a disponibilidade limitada em bandas não licenciadas é permitir que os usuários não licenciados (Usuários Secundários - US) acessem dinamicamente as bandas licenciadas, desde que não provoquem a interferência com os proprietários das faixas licenciadas (Usuários Primários - UP).

Nesse contexto, o Rádio Cognitivo (RC) [Mitola III 2009] se apresenta como uma tecnologia promissora que permite o uso dinâmico do espectro de rádio freqüência [Akyildiz 2006a]. Em uma rede formada por RCs, os dispositivos são equipados com rádios

*Artigo financiado com recursos da MCT/FINEP PLATCOG (01.10.0549.00).

 ©2012 SBC — Soc. Bras. de Computação

que possuem flexibilidade no uso do espectro, com capacidade de detecção de bandas disponíveis, reconfiguração da frequência do rádio e de trocas entre as bandas selecionadas [Akyildiz 2006a], [Akyildiz 2006b] e [Thomas 2005]. Com base nas informações de sensoriamento do espectro, os usuários de rádio acessam as bandas licenciadas oportunisticamente quando nenhum usuário primário as estiver utilizando e, necessariamente, devem deixá-las imediatamente ao detectar a atividade do Usuário Primário (UP). Os Rádios Cognitivos podem ser formalmente definidos como dispositivos de comunicação inteligentes e adaptativos capazes de modificar seus parâmetros de transmissão – tais como: frequência de operação, tipo de modulação, potência de transmissão, protocolos de comunicações e outros – baseados em interações com o ambiente em que operam [Mitola 2000].

O Rádio Cognitivo (RC) é uma nova abordagem de acesso ao espectro de radiofreqüência que visa otimizar o uso deste recurso de forma oportunística. Para suportar tais capacidades, o RC possui, basicamente, quatro funcionalidades que gerenciam todas as suas atividades operacionais referentes à utilização do espectro [Akyildiz 2008]: (i) Sensoriamento do Espectro, (ii) Gerenciamento do Espectro, (iii) Mobilidade Espectral e (iv) Compartilhamento do Espectro. O Sensoriamento do Espectro é a funcionalidade responsável por monitorar o espectro e determinar quais são os canais licenciados que estão disponíveis, i.e. quais são os canais que não estão sendo utilizados pelo usuário primário. O Gerenciamento do Espectro de um RC é responsável por selecionar qual o canal licenciado livre mais apropriado para a transmissão de acordo com os seus requisitos de QoS (*Quality of Service*). A Mobilidade Espectral é responsável por desocupar o canal licenciado quando a presença do usuário primário é detectada, i.e. quando o usuário principal começar a utilizar o canal licenciado. Por fim, o Compartilhamento do Espectro é a funcionalidade que permite à rede de rádios cognitivos de utilizar os canais licenciados livres de forma cooperativa e colaborativa entre os rádios da rede.

Entre os vários desafios a serem abordados encontramos o que está relacionado ao provimento de segurança às redes de rádios cognitivos, uma vez que os equipamentos utilizados podem ser vítimas de ações que os impeçam de se comunicar efetivamente.

Entre as novas características operacionais utilizadas pela tecnologia do RC para a efetiva comunicação, encontramos a necessidade do constante monitoramento do meio para que o rádio possa: perceber as frequências licenciadas que não estão em uso; decidir qual freqüência licenciada livre utilizar; e adaptar-se às condições reais de transmissões oportunísticas nas freqüências licenciadas. Todavia, tal característica o torna vulnerável a tipos específicos e novos de ataques não observados nas redes convencionais.

De uma forma geral, o uso inadequado das freqüências licenciadas livres (FLL) pode ser configurado como um tipo de ataque [Leon 2010] às Redes de Rádios Cognitivos (RRC). Tais ações podem degradar de forma parcial ou total o funcionamento da rede cognitiva, ocasionando o não aproveitamento da capacidade oportunística de utilização das faixas licenciadas livres por parte da rede.

Este trabalho visa o desenvolvimento de um sistema de detecção de ataques para a identificação de rádios que estejam promovendo a inadequada utilização das freqüências licenciadas livres em uma RRC. Para isso, buscamos: (i) identificar os principais tipos de ataques aos quais estas redes estão sujeitas, (ii) selecionar os mecanismos de detecção mais adequados a estes tipos de ataques, e (iii) integrá-los de forma a alcançar os melhores índices possíveis de detecção.

Este artigo está organizado da seguinte forma: na Seção 2 são descritos os aspectos de segurança bem como os trabalhos relacionados; na Seção 3 a proposta do Sistema de

 ©2012 SBC — Soc. Bras. de Computação

Detecção de Ataques é apresentada; na Seção 4 são descritos os experimentos realizados; na Seção 5 é feita uma análise dos resultados obtidos; e finalmente, na Seção 6 são tecidas as conclusões finais e as propostas de trabalhos futuros.

## 2. Aspectos de Segurança e Trabalhos Relacionados

Quando se fala de segurança em Rádios Cognitivos, automaticamente são associados dois novos tipos de ataques específicos: a emulação do usuário primário (*Primary User Emmulation* – PUE) e a adulteração das tabelas de Frequencia Licenciada Livre (FLL) (*Spectrum Sense False Feadback.*- SSFF).

***PUE***

No caso do PUE um ou mais Rádios Cognitivos modificam as suas características de transmissão, adaptando-as a forma utilizada pelo usuário primário, e induzem aos outros rádios da Rede de Rádios Cognitivos de que estas frequências livres já estejam em uso. Caso o ataque seja utilizado por pares de rádios para monopolização das freqüências livres, este é denominado de PUE-S (*Primary User Emmulation – Selfish*). Caso a finalidade seja impedir que nenhum rádio utilize as frequências livres, este é definido como PUE-M (*Primary User Emmulation – Malicious*). O mecanismo de localização dos rádios na rede possibilita a determinação do posicionamento geográfico dos rádios. Este mecanismo associado aos níveis de potência de transmissão de cada um dos rádios identifica se o rádio em análise é um usuário principal (torre de transmissão da frequência licenciada), ou um usuário secundário (rádio cognitivo componente da rede). Para a determinação da localização dos transmissores são feitas as seguintes suposições: (i) os transmissores primários são torres com uma posição conhecida e fixa, além de possuir altíssimo poder de transmissão (na escala das centenas de *KiloWatts* - kW); e (ii) o CR é um dispositivo com o poder limitado de transmissão (que varia dos *miliwatts* – mW a alguns watts - W). Diversas abordagens foram apresentadas no contexto de segurança da RRC [Clancy 2008], [Clancy 2009], [Leon 2010], que utilizam como método de detecção de ataque PUE a utilização da localização dos rádios secundários e do usuário primário. Por exemplo, [Shrestha 2010] propõe um método de detecção de PUE, baseado no cálculo Euclidiano do posicionamento dos rádios da rede, bem como da torre de transmissão. A proposta baseia-se no conhecimento prévio do posicionamento e da potência de transmissão de todos os rádios cognitivos da rede, como também do usuário primário. Porém o trabalho define sua hipótese de detecção em um cenário cooperativo, onde os rádios primários transmitem informações de controle, entre elas: posicionamento, potência de transmissão, etc.; contrariando as normas do [Fcc 2009], de que nenhuma alteração na infraestrutura e nas configurações das redes primárias deve ser feita com a finalidade de adaptar-se a nova tecnologia de rádio cognitivo. A proposta de [Park 2008] diferencia-se de sobremaneira das propostas supracitadas porque, além da análise para a decisão de detecção do ataque ser baseada também no comportamento dos rádios da rede, não há a necessidade de qualquer informação fornecida pelo usuário primário.

***SSFF***

Nas redes onde ocorre o monitoramento do espectro de forma colaborativa, a ocupação das freqüências licenciadas livres é baseada nas informações trocadas entre os rádios (tabelas de frequências livres - FLL) sobre quais frequências estão livres ou não. Através da adulteração dessas tabelas, é possível forjar que uma dada freqüência livre esteja ocupada, ou vice-versa, o que configura um ataque do tipo SSFF. Se este procedimento é empregado para que as frequências não sejam compartilhadas, ficando o seu uso exclusivo para um único usuário, denomina-se de ataque SSFF no modo egoísta *(selfish).* Se uma frequência é declarada livre (resp. ocupada) quando estiver ocupada (resp. livre) apenas para impedir a

 ©2012 SBC — Soc. Bras. de Computação

sua utilização, configura-se um ataque SSFF de modo *DoS (denial of service).* A generalização SSFF quanto a qualquer falsificação de informações do sensoriamento do espectro pode ainda ser especificada em três tipos distintos de ataques: (*i*) sempre-livre (SSFF-SL): sempre transmite a informação de que o canal licenciado está livre, (*ii*) sempre-ocupado (SSFF-SO): sempre transmite a informação de que o canal licenciado está ocupado, e (*iii*) sempre-falso (SSFF-SF): onde o rádio malicioso sempre transmite o inverso da realidade do canal, isto é, livre quando ocupado e ocupado quando livre. O ataque SSFF foi mencionado primeiramente em [Misha 2006] e, adicionalmente, por [Ruiliang 2008a] e [Ruiliang 2008b]. Em [Ruiliang 2008b] a detecção dos dados falsificados foi realizada utilizando-se um esquema matemático baseado na relação sequencial de probabilidade, esquema esse com bons resultados. Entretanto este método exige o conhecimento prévio da posição física de todos os elementos constituintes da rede. A proposta de [Zhu 2009] faz uso de um mecanismo de reputação para definir um grau de credibilidade às informações sobre as tabelas de freqüências livres recebidas dos rádios da rede. O uso da reputação atribui um grau de aceitação a um rádio da rede para desempenhar uma tarefa sem que necessariamente o rádio requisitante tenha interagido com o rádio alvo anteriormente. Para tanto, utiliza-se as experiências dos demais rádios da Rede.

Apesar de os mecanismos: de localização [Park 2007] e o de reputação [Zhu 2009] serem completamente independentes, há situações em que a utilização dos dois mecanismos em conjunto (de forma complementar) pode aumentar a eficiência da detecção de ataques. Por exemplo, um ataque de PUE, que normalmente é detectado pelo mecanismo de localização com base na análise das potências de transmissões captadas, pode não ser detectado dependendo do posicionamento geográfico da rede. Quando um rádio está localizado no limite de alcance da transmissão do UP, i.e. onde a potência de recepção captada pelo rádio da transmissão do UP é muito baixa, a análise do mecanismo pode ser inviabilizada, já que a distinção entre uma transmissão de um UP de uma transmissão de outro rádio da rede não é facilmente reconhecida. Porém, com a utilização dos dois mecanismos em conjunto, mesmo que o mecanismo de localização, em virtude do posicionamento geográfico da rede, não consiga inferir sobre algum ataque de PUE, o mecanismo de reputação o possibilitará. Visto que, o uso indevido de uma frequência licenciada livre (FLL) por um atacante PUE induzirá aos rádios ao seu alcance de transmissão que não há canais livres e, por conseqüência, tais rádios disseminarão suas tabelas de freqüências indicando que tais canais estão ocupados pelo Usuário Primário (UP). Porém, todos os rádios que estiverem ao alcance dos rádios atacados e fora do alcance do atacante PUE (e que evidentemente não são por estes influenciados quanto à ocupação da FLL) não detectam transmissões do UP e, por conseqüência, identificam ataques de SSFF daquela região da rede onde o atacante PUE está localizado. A retroalimentação de informações confiáveis de ataques (aferidas e validadas através dos mecanismos de reputação) indica aos rádios atacados que há uma transmissão indevida realizada por um rádio localizado dentro dos seus raios de transmissão.

Logo, esse trabalho propõe um SDA baseado no uso conjunto e adaptado de métodos de localização e de reputação. Tal escolha procura: (i) melhorar a eficiência na detecção de ataques de PUE, mesmo em situações onde o mecanismo de localização não possa inferir sobre o ataque, e (ii) agrupar em um único SDA a detecção de ataques dos tipos PUE e SSFF.

 ©2012 SBC — Soc. Bras. de Computação

# 3. Proposta de SDA

O sistema de detecção de ataques descrito neste trabalho é composto, basicamente, por um mecanismo de localização integrado a um mecanismo de reputação.

## 3.1 Mecanismo de localização

O mecanismo de localização empregado consiste de uma variante da proposta de [Park 2008], adaptada para o ambiente urbano, uma vez que o modelo estatístico original é destinado para áreas rurais. Tal modelo foi substituído pelo proposto em [Hata 1980], para contextualizar o cenário de grandes cidades (área urbana) e com frequências acima de 400 MHz. Cenário esse onde encontrarmos as maiores concentrações de equipamentos que fazem uso das faixas de frequências não-licenciadas, causando, em virtude disso, uma maior escassez espectral, sendo então áreas propícias à utilização de redes de rádios cognitivos. O modelo [Hata 1980] para áreas urbanas é descrito como:

$$\Pr = \frac{Pt * Gr * Gt}{L} \tag{1}$$

$$\begin{aligned} L &= 69{,}55 + 26{,}16\ell og(f_{MHz}) - 13{,}82\ell og(h_{Tef}) - \\ & a(h_{\mathrm{Re}f}) + [44{,}9 - 6{,}55\ell og(h_{Tef})]\ell og(d_{km}) \end{aligned} \tag{2}$$

$$a(h_{\mathrm{Re}f}) = 3{,}2[\ell og(11{,}75 h_{\mathrm{Re}f})]^2 - 4{,}97) \tag{3}$$

Na Equação 1: *Pt* e *Pr* são as potências de transmissão e recepção; *Gt* e *Gr* são os ganhos das antenas transmissoras e receptoras, respectivamente e *L* é a perda no percurso; Na Equação 2: $f_{MHz}$ é a frequência de 150 a 1500 MHz; *d* é a distancia de 1 a 20 km; $h_{Tef}$ é a altura efetiva da antena transmissora de 30 a 200 m; $h_{Ref}$ é a altura efetiva da antena receptora de 1 a 10 m, e, na equação 3, $a(h_{Ref})$ é o fator de correção da altura efetiva da receptora.

Para compor o mecanismo de reputação deste trabalho optou-se pelo modelo clássico de fusão dos dados apresentado em [Zhu 2009]. Este modelo é formado basicamente pelo rádio definido como centro de fusão de dados, isto é, o rádio que recebe as informações de detecção do espectro, e pelos rádios que as transmitem (Figura 1). O centro de fusão pode ser qualquer um dos rádios que necessitem utilizar os canais licenciados livres em um tempo determinado (**Δ*t***). Em nosso exemplo, é representado pelo rádio ***RC0***.

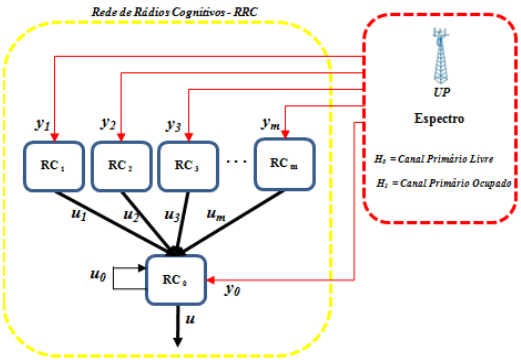



**Figura 1 – Modelo Clássico de Fusão de dados**

Numa rede, cada rádio mantém históricos de avaliações geradas a partir de suas experiências com outros rádios. Uma avaliação é uma nota dada pelo rádio quanto à veracidade da informação que recebeu sobre os canais licenciados livres. Estas avaliações são usualmente conhecidas por "informações de primeira mão", sendo as avaliações

 ©2012 SBC — Soc. Bras. de Computação

recebidas de outros rádios comumente chamadas de "informações de segunda mão" [Zhu 2009].

As informações recebidas pelo centro de fusão de dados (***RC₀***) são definidas como $u_i$, sendo i = 0,1,2,3,...m, dos rádios cognitivos vizinhos (***RCi***), quanto aos canais licenciados livres (**H₀**), e os canais licenciados ocupados pelas transmissões do usuário primário (**H₁**). Sendo $u_i$ =0 ou $u_i$ =1, caso a decisão do ***RCi*** seja **H₀** ou **H₁** respectivamente. Tais informações tem como base a percepção que cada ***RCi*** tem de seus monitoramentos do espectro, e definidas como $y_i$.

Cada $u_i$ recebida por ***RC₀*** do ***RCi*** é carregada no vetor de fusão **V*u*.** Por fim, o centro de fusão, com base nas informações de monitoramento recebidas, bem como a detecção do espectro por ele realizada, extrai uma decisão global, ***u***, onde ***u*** = 1 significa a ocupação do canal licenciado, isto é, **H₁**, e ***u*** = 0 que o canal está livre, **H₀**.

Basicamente o mecanismo faz a análise das informações $u_i$ e a atribuição de pesos ***w*** aos ***RCi***. Após a inicialização do sistema, o crédito de cada rádio ***RCi*** é ajustado para **zero**, sendo que cada ***RCi*** pode acumular créditos por informações $u_i$ corretas.

Sempre que a informação de um determinado rádio for consistente com a decisão global ***u***, isto é, a informação final processada pelo ***RC₀***, seu crédito será aumentado por um; se não diminuído por um. Denotando o crédito para o ***RCi*** por ***Ci***, o sistema de crédito é representado na Equação 4.

$$Ci = \begin{cases} Ci \quad +1, & if \quad u_i = u \\ Ci \quad -1, & if \quad u_i \neq u \end{cases} \tag{4}$$

Com a finalidade de justiça ao pontuar a reputação de um rádio que, porventura, tenha classificado erroneamente a detecção do canal licenciado, o peso, antes de ser atribuído ao ***RCi***, é normalizado pela média dos créditos do rádio ***RCi*** . Denotando-se $w_i$ o peso do ***RCi*** , Equação 5. Onde: $w_i$ é o peso do ***RCi***, ***Ci*** é o seu crédito; ***avg*** (***Ci*** ) denota o crédito médio do ***RCi*** , e ***g*** é uma constante cujo valor é de 5.51 (denominado de Valor do Coeficiente de Normalização VCN) , valor este analisado e calibrado pelo autor [Zhu 2009). O Vetor de Credibilidade *W* é então carregado como o valor de $w_i$ para cada ***RCi*** (Equação 6).

$$w_i = \begin{cases} 0, & if \quad Ci \quad < -g \\ \dfrac{Ci \quad +(-g)}{avg(Ci+g)}, & if \quad Ci > -g \end{cases} \tag{5}$$

$$W = \sum_{i=0} (-1)^{ui+1} wi, \tag{6}$$

$$\begin{cases} W \geq q & \rightarrow aceita\, H_1 \\ W \leq q & \rightarrow aceita\, H_0 \\ -q < W < q & \rightarrow \text{proceder a outra análise} \end{cases}$$

O módulo de teste é inicializado com a determinação do limiar de aceitabilidade ***q.*** Este limiar é utilizado para a tomada de decisão por parte do centro de fusão quanto à ocupação, ou não, do canal licenciado. O valor ***q*** é utilizado como limite superior e ***–q*** utilizado como limite inferior do módulo de teste. Após as análises realizadas, há a convergência da

 ©2012 SBC — Soc. Bras. de Computação

credibilidade para o mínimo das credibilidades daqueles rádios cujas informações de canais livres divergiram da decisão global de cada cluster.

### 3.2 Integração dos mecanismos

Os dois mecanismos são combinados por meio de uma função utilidade [Clemen 2001]. Esta função propõe uma utilidade para cada método, onde é calculado o peso representativo da importância de cada um deles. Para a integração dos mecanismos descritos nas seções anteriores é usada, mais especificamente, a Função Utilidade Aditiva (FUA) [Clemen 2001] em virtude de esta proceder a analise de alternativas com atributos puramente independentes, especificamente o que ocorre quando da combinação dos dois mecanismos aqui utilizados. A metodologia utilizada como base da Função Utilidade Aditiva prevê a atribuição de pesos para cada atributo envolvido na tomada de decisão dentre as alternativas envolvidas entre dois ou mais objetivos [Lopes 2008]. Para tal, utilizamos para atribuição de pesos ponderados a cada atributo da Função Utilidade Aditiva, i.e. ao mecanismo de localização e ao mecanismo de reputação, o conceito de algoritmo genético [Lacerda 1999], pois, este consiste em uma heurística de otimização, cujo objetivo é o de achar a solução que corresponda ao ponto máximo de uma determinada função, enquadrando-se, portanto, ao objetivo deste trabalho (procedimentos detalhados na seção 5).

## 4. Avaliação Experimental

Para a avaliação da proposta foi simulada uma rede sem-fio descentralizada com topologia plana e com nós fixos. Quanto à disposição física da rede, procurou-se reproduzir fielmente a utilizada em [Park 2008] para fim de estudo comparativo. A simulação foi feita em um ambiente MatLab/Simulink.

### 4.1 Objetivos

O objetivo da avaliação experimental teve como foco a análise comparativa dos mecanismos de detecção de forma isolada bem como os seus desempenhos em atuação conjunta, onde, neste caso, houve a calibração tanto dos limiares de aceitabilidade (LA) quanto dos valores dos coeficientes de normalização (VCN). O desempenho da rede, no que diz respeito à ocupação oportunística dos canais licenciados livres em situações normais, bem como sob regime de ataques, constituiu também um dos objetivos da avaliação. Por fim, a avaliação foi utilizada para a atribuição e calibração dos pesos de cada um dos mecanismos na função utilidade por meio do uso do algoritmo genético.

### 4.2 Cenário

A rede foi composta por 300 rádios cognitivos, distribuídos aleatoriamente em uma área quadrada de 2000m; cada rádio possui um raio de alcance de transmissão de 250m, com um raio de interferência de 550m, seguindo o modelo proposto em [Ross 2002] (Figura 2a). Dois usuários primários (torre de TV 1 e Torre de TV 2) são posicionas à 8000m e 5000m das bordas externas da rede (Figura 2b). As torres possuem o alcance de transmissão de 9000m e 7000m, respectivamente. As torres possuem frequência de transmissão de 617 MHz; utilizando 10 canais numerados de 1 a 10.

Foi feita a distribuição dos 300 rádios nas coordenadas cartesianas da rede de simulação de forma randômica. A distribuição utilizada para a aleatoriedade dessa inserção, como também em todos os processos da simulação, é a Distribuição Binomial [Garmeman 1993], podendo com isso cada rádio estar ao alcance: (i) das duas torres de TV; (ii) apenas da torre 1; (iii) apenas da torre 2; ou mesmo, (iv) fora do alcance das duas torres. Nessa primeira fase há então a inicialização da rede onde são formados 300 clusters com cada um dos 300 rádios constituintes, cada rádio como centro de cada cluster.

 ©2012 SBC — Soc. Bras. de Computação

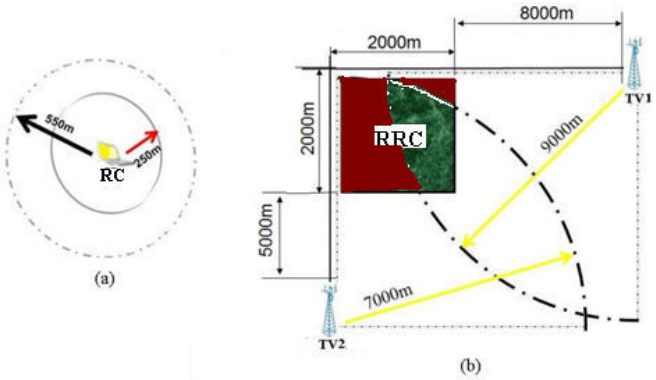



**Figura 2. Cenário Utilizado: (a) Raios de Alcance do rádio (b) Rede**

Cada rádio reconhece seus rádios vizinhos, isto é, os rádios que estão ao alcance do raio de transmissão do centro do cluster (250m); bem como os rádios que estão em seu raio de interferência (550m), Figura 2a.

### 4.3 Primeiro Experimento: Simulação das Transmissões

Nesse experimento as transmissões propriamente ditas, com e sem ataques são simuladas. Os parâmetros para a simulação [Park 2008], são modelados da seguinte forma: ambas as torres de TV possuem a altura de 100m; transmitem a frequência de 617 MHz, utilizando 10 canais de frequência, numerados de 1 a 10, cuja sensibilidade das antenas dos receptores primários é de -37.50 dBm. A potencia de transmissão da Torre de TV 1 é de 84 dBm, e da Torre de TV 2 é de 83dBm. Os rádios cognitivos possuem uma potencia de transmissão de 24 dBm, antenas de 1 m de altura, cuja sensibilidade é de -93dBm; transmitindo a frequência de 617 MHz, utilizando 10 canais de frequência, numerados de 1 a 10.

As transmissões são simuladas a cada 100ms, divididas em *slots* de 10ms cada, durante um período de uma hora. As torres de TV utilizam aleatoriamente 8, 9 ou 10 *slots* de 10 ms por transmissão, podendo ocorrer a transmissão: (i) das duas torres simultaneamente; (ii) de apenas uma delas; ou mesmo, (iii) nenhuma transmissão do usuário primário. Há a escolha randômica de 0 a 150 pares de rádios para a transmissão no espaço de 100ms, divididos em *slots* de 10ms cada. Cada rádio pode transmitir pacotes de controle ( utilizando-se de 1 a 5 *slots* de 10ms) ou pacotes de dados ( de 1 a 10 *slots* de 10ms). Antes de cada transmissão o rádio verifica os canais do espectro de frequência que não estão sendo utilizados: (i) pelas torres de TV; (ii) por outro(s) rádio(s) pertencente(s) ao cluster; ou mesmo, (iii) se não há interferência de alguma transmissão de um cluster vizinho.

São realizadas 30 simulações sem ataque; 30 simulações de cada um dos ataques isoladamente, variando de 1 a 30 rádios atacantes em cada uma delas; e 30 simulações com a combinação de todos os ataques, variando-se de 1 a 30 ataques de cada tipo (Tabela 1).

### 4.4 Segundo Experimento: Simulação da Detecção de Ataques

O mecanismo de Localização é acionado a cada intervalo de 1 segundo, analisando as potências captadas e procedendo a verificação das fontes de transmissão. Enquanto o mecanismo de reputação é disparado em intervalos de 2.5s, onde os graus de credibilidade de cada rádio são processados com base nas tabelas de frequências livres por eles disseminadas, há a simulação das transmissões das torres de TV e dos rádios, por um período de 1 hora.

A cada unidade de tempo equivalente a 10 ms (definidos como *slot*), aleatoriamente é feita a escolha se haverá ocupação (transmissão) de algum(ns) canal(is) nesse *slot*: (i) por parte das Torres de TV, (ii) de uma delas, (iii) ou de nenhuma – onde, nesse caso, configura-se o *white space*. Simultaneamente, são definidos quantos pares de rádios "irão transmitir" nesse

 ©2012 SBC — Soc. Bras. de Computação

*slot*; cada rádio define se irá realizar uma transmissão de dados ou de controle. Na transmissão de controle é feita uma seleção de 1 a 5 pacotes a serem transmitidos ( 1 pacote equivale a 1 *slot* de tempo), enquanto que na transmissão de dados, a seleção varia de 1 a 10 pacotes. O rádio então faz o sensoriamento do meio, verificando se há a ocupação do espectro (canais) por parte: (i) dos Usuários Primários (UPs) (Torres de TV), (ii) de outro(s) rádio(s) vizinho(s), (iii) ou mesmo de alguma transmissão interferente (rádios a 550m transmitindo). Caso contrário, o rádio transmite 1 pacote, e aguarda o próximo *slot* de tempo para a transmissão de outro pacote, caso haja.

São realizadas simulações com e sem ataques. As simulações com ataques consistem, no caso de Ataques de PUE, na inserção de rádios que, em instantes aleatórios, ocupam de forma prioritária a utilização do *slot* vagos. E no caso do SSFF há a inserção de rádios que, em instantes aleatórios, emitem tabelas de frequências adulteradas.

O Mecanismo de Localização é simulado a cada 1s de forma estática, por meio de tabelas de recepção de potencias pré-configuradas. Essas tabelas são inicializadas por meio do cálculo euclidiano das distancias entre cada rádio e seus vizinhos, bem como entre os rádios e as torres de TV. As potências de recepção são então calculadas com uma taxa aleatória de erro de 10% para mais ou para menos do valor nominal encontrado.

Antes da transmissão, o rádio verifica se há ocupação do espectro de frequência licenciada. Caso as características de transmissão captada não se enquadram com a transmissão dos rádios vizinhos (os pacotes não contem os números MAC - endereço físico de 48 bits do rádio, ou, mais especificamente, da interface de rede), o rádio dispara o mecanismo de localização. A potencia do sinal é analisada, quando então é definida a existência ou não de ataque(s) PUE.

**Tabela 1. Dados das Rodadas de Simulações (por *slots* de tempo)**

| Qtde | Tipo de Ataque | White Space | Tx TV | RCs a Transmitir | RCs que Transmitiram | Pacotes para Transmitir | Pacotes Transmitidos |
|---|---|---|---|---|---|---|---|
| **0** | **Sem Atqs** | 1.734.167 | 1.865.033 | 2.691.394 | 5.267 | 12.107.175 | 89.584 |
| **1** | **PUE S** | 1.632.115 | 1.657.182 | 2.032.601 | 5.208 | 11.359.036 | 88.688 |
| **30** | **PUE S** | 1.687.557 | 1.626.369 | 2.695.563 | 1.177 | 12.210.425 | 66.265 |
| **1** | **PUE M** | 1.575.944 | 1.716.623 | 2.128.404 | 5.311 | 12.734.860 | 88.812 |
| **30** | **PUE M** | 1.605.428 | 1.774.777 | 2.413.363 | 226 | 12.131.096 | 66.358 |
| **1** | **SSFF-SL** | 1.655.456 | 1.889.362 | 2.572.904 | 5.018 | 12.226.692 | 87.432 |
| **30** | **SSFF-SL** | 1.784.063 | 1.678.762 | 2.768.152 | 5.199 | 12.733.963 | 65.327 |
| **1** | **SSFF-SO** | 1.551.707 | 1.815.660 | 2.732.220 | 5.347 | 12.116.144 | 85.163 |
| **30** | **SSFF-SO** | 1.764.731 | 1.619.580 | 2.442.436 | 0 | 12.738.830 | 63.631 |
| **1** | **SSFF-SF** | 1.768.732 | 1.801.576 | 2.453.715 | 5.289 | 11.181.238 | 87.771 |
| **30** | **SSFF-SF** | 1.717.718 | 1.885.747 | 2.764.480 | 835 | 12.810.070 | 65.580 |
| **5** | **Todos** | 1.660.131 | 1.670.717 | 2.254.649 | 5.812 | 12.039.828 | 87.904 |
| **150** | **Todos** | 1.648.468 | 1.845.401 | 2.013.124 | 0 | 11.970.240 | 32.888 |

O Mecanismo de Reputação é simulado a cada 2,5s, por meio da análise das tabelas de frequência recebidas dos rádios vizinhos, bem como das tabelas do próprio rádio que faz a análise. Por meio dos cálculos de atribuição de credibilidade, a reputação dos rádios então são a eles atribuídas e selecionadas como suspeitas após ultrapassarem os limiares de aceitabilidade. Os limiares de aceitabilidade (LA), superior e inferior (**q** e – **q**), foram definidos por [Zhu 2009], em 15 e -15, respectivamente. Bem com o valor do coeficiente de aceitabilidade (VCA) ***g*** em 5,51.

Foram realizadas 210 rodadas de simulações: 30 sem ataques; 30 variando-se somente o ataque de PUE Selfish de 1 a 30 pares de rádios atacantes; 30 variando-se somente o ataque

 ©2012 SBC — Soc. Bras. de Computação

de PUE Malicioso de 1 a 30 rádios atacantes; 30 variando-se somente o ataque de SSFF-SL de 1 a 30 rádios atacantes; 30 variando-se somente o ataque de SSFF-SO de 1 a 30 rádios atacantes; 30 variando-se somente o ataque de SSFF-SF de 1 a 30 rádios atacantes, e por fim, 30 variando-se todos os ataques em conjunto de 5 a 150 rádios atacantes.

Na Tabela 1 foram representados os dados referentes as primeiras e as últimas rodadas de cada item acima mencionado. A tabela 1 é formada por 8 colunas, a saber: coluna 01 (Nr de Atacantes), quantidade de atacantes simulados naquele *slot*; coluna 02 indica o tipo de transmissão (sem ataque – Tipo de Ataque); coluna 3 (WS): a quantidade de *slots* de *white spaces*; coluna 4 (TV): a quantidade de slots ocupados pelas Torres de TV; coluna 5 (rádios Tx): quantidade de rádios a transmitir por simulação; coluna 6 (rádios N): quantidade de rádios que transmitiram; coluna 7 (Pc Tx A): número de pacotes a serem transmitidos, e; por fim, coluna 8 (Pc Tx): número de pacotes efetivamente transmitidos.

Podemos observar, por meio dos gráficos representados na Figura 3, a degradação da rede relacionada ao número de rádios que não tiveram oportunidade de transmitir quando os ataques foram inseridos nas simulações. Principalmente quando os ataques são efetuados de forma conjunta, onde, por meio de 15 atacantes, a rede perde totalmente sua capacidade de ocupação oportunística do espectro.

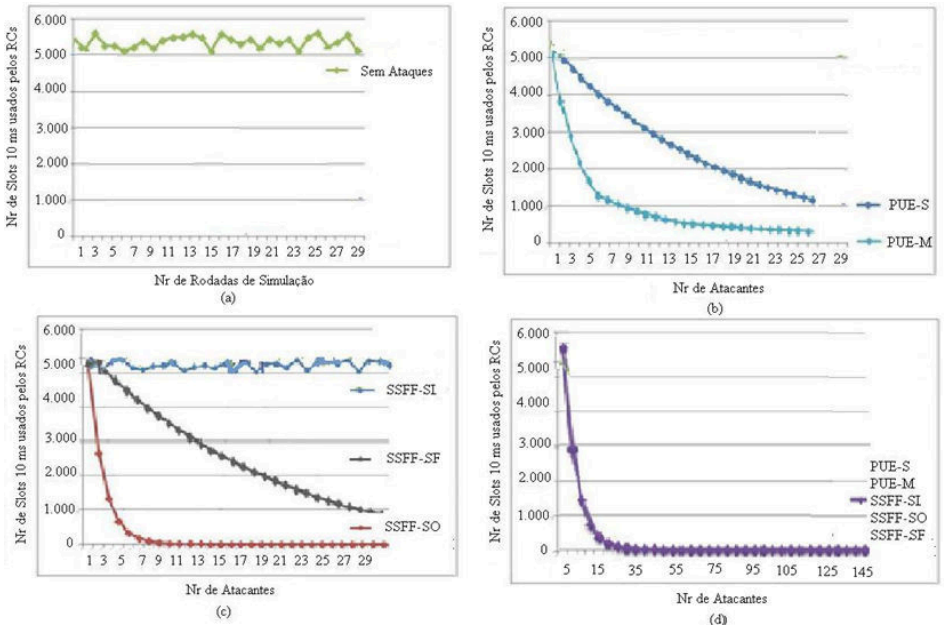



**Figura 3. Degradação da RRC: (a) Sem Ataques. (b) Com Ataques de PUE. (c) Com Ataques SSFF. (d) Combinação de Todos os Ataques**

Simultaneamente à simulação dos diversos tipos de transmissões, os mecanismos de localização e de reputação são disparados. Ao analisarmos cada tipo de ataque, bem como a atuação dos mecanismos de detecção de forma isolada sem qualquer tipo de calibração verificamos que na detecção do ataque de PUE-S tem-se uma taxa média de verdadeiros positivos na faixa de 42,50 %. Quanto ao ataque de PUE-M, a taxa fica em torno de 34,54%. Essa diferença pode ser justificada em virtude da maior agressividade do ataque de PUE-M, quem tem a finalidade de realmente degradar totalmente a operacionalidade da rede. Ao contrário do ataque de PUE-S, que têm por objetivo utilizar os canais livres de forma egoística.

As taxas médias de verdadeiros positivos quando o mecanismo de reputação é usado de forma isolada para a detecção de ataques de SSFF são: 44.51, 42.64 e 33. 22, para os ataques de SSFF-SL, SSFF-SF e SSFF-SO, respectivamente. Novamente foi demonstrado

 ©2012 SBC — Soc. Bras. de Computação

que a agressividade do ataque reflete efetivamente no mecanismo de detecção. Maiores taxas para o ataque de SSFF-SL e menores para o SSFF-SO.

### 4.5. Terceiro Experimento: Variando o LA e o VCN

Neste experimento foram realizadas variações para o limiar de aceitabilidade (LA), bem como o valor da constante de normalização das credibilidades dos rádios (VCN) com intuito de adaptar o mecanismo de análise da reputação para o trabalho em conjunto com o mecanismo de localização. Inicialmente foram fixados o limiar de aceitabilidade e o valor da constante de normalização das credibilidades em, 15 e 5.51, respectivamente, seguindo o proposto no mecanismo original [Zhu 2009]. Em cada rodada de simulação foram realizadas 1.440 avaliações do mecanismo. Sendo realiza avaliações em intervalos regulares de 2.5 segundos. Totalizando nas 30 rodadas de simulações (combinações dos ataques): 43.300 avaliações. A Tabela 2 apresenta os resultados obtidos nesse experimento para o LA;

**Tabela 2. LAs Representativos das Rodadas de Simulações**

| LA | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VP | 48,93 | 49,01 | 50,00 | 50,00 | 50,00 | 50,00 | 49,43 | 49,01 | 48,90 | 48,93 | 47,01 |
| FP | 3,66 | 3,11 | 2,97 | 0,16 | 0,99 | 1,67 | 2,69 | 3,11 | 3,97 | 3,66 | 4,11 |
| FN | 1,07 | 0,99 | 0,00 | 0,00 | 0,00 | 0,00 | 0,57 | 0,99 | 1,10 | 1,07 | 2,99 |
| VN | 46,34 | 46,89 | 47,03 | 49,84 | 49,01 | 48,33 | 47,31 | 46,89 | 46,03 | 46,34 | 45,89 |

A Tabela 3 apresenta os resultados obtidos nesse experimento para o VCN; e A Tabela 4 apresenta os resultados obtidos com a variação do LA e do VCN em conjunto. As métricas para analisar os experimentos são: (i) falsos positivos (FP), que indicam a quantidade de alarmes falsos; (ii) falsos negativos (FN), que indicam uma condição de normalidade quando na verdade está ocorrendo um ataque; (iii) verdadeiros positivos (VP), que indicam que está ocorrendo um ataque durante um ataque; e (iv) verdadeiros negativos (VN), que indicam uma condição de normalidade quando não está ocorrendo nenhum ataque.

**Tabela 3. VCNs das Rodadas de Simulações**

| VCN | 5.46 | 5.47 | 5.48 | 5.49 | 5.50 | 5.51 | 5.52 | 5.53 | 5.54 | 5.55 | 5.56 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VP | 48,93 | 49,01 | 50,00 | 50,00 | 50,00 | 49,00 | 48,43 | 47,86 | 46,7 | 45,73 | 44,02 |
| FP | 7,25 | 5,11 | 2,97 | 0,83 | 0,96 | 1,67 | 2,69 | 3,71 | 4,73 | 5,75 | 6,77 |
| FN | 1,07 | 0,99 | 0,00 | 0,00 | 0,00 | 1,00 | 1,57 | 2,14 | 3,3 | 4,27 | 5,98 |
| VN | 42,75 | 44,89 | 47,03 | 49,17 | 49,04 | 48,33 | 47,31 | 46,29 | 45,27 | 44,25 | 43,23 |

**Tabela 4. LA/VCN Representativos das Rodadas de Simulações**

| LA | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| VCN | 5.46 | 5.47 | 5.48 | 5.49 | 5.50 | 5.51 | 5.52 | 5.53 | 5.54 | 5.55 | 5.56 |
| VP | 44,65 | **45,25** | 45,85 | 46,45 | **47,05** | 46,42 | 45,59 | 45,56 | 44,98 | 44,97 | 44,07 |
| FP | 3,66 | 3,11 | 2,97 | 0,91 | **0,16** | 1,25 | 2,69 | 3,11 | 3,97 | 3,66 | 4,11 |
| FN | 5,35 | 4,75 | 4,15 | 3,55 | **2,95** | 3,58 | 4,41 | 4,44 | 5,02 | 5,03 | 5,93 |
| VN | 46,34 | 46,89 | 47,03 | 49,09 | **49,84** | 48,75 | 47,31 | 46,89 | 46,03 | 46,34 | 45,89 |

 ©2012 SBC — Soc. Bras. de Computação

## 5. Avaliação dos Resultados e Calibração

Observa-se que: na simulação conjunta dos mecanismos, tanto com o aumento, quanto com a diminuição do limiar de aceitabilidade a partir do valor 13, e do valor do coeficiente de normalização, a partir do valor 5.49 ocorre aumento nos valores de FP e FN.

Já na análise em conjuntos do LA e do VCN, com os valores 14 para o LA e o valor 5.50 para o VCN, obtemos os melhores valores de FP e FN. Como nosso estudo de caso prioriza a detecção de ataques com a utilização em conjunto dos dois mecanismos, optamos por trabalhar com o número de limiar de aceitabilidade igual a 14 e do valor do coeficiente de normalização igual a 5.50. Após as calibrações foram realizadas mais 10 rodadas de simulações utilizando-se o LN=14 e o VCN=5.50.

Houve a inserção de todos os ataques em sua configuração máxima, isto é, 30 atacantes de cada tipo. Os resultados obtidos podem ser observados na tabela 5: VP=45.14, VN=48.70, FN=4.86 e FP=1.30. Valores esses quando comparados a [Park 2008] com uma taxa média de VP=44% e a [Zhu 2009] VP=43,18%, demonstram que a combinação dos mecanismos na detecção de intrusos PUE e SSFF tem um melhor desempenho.

**Tabela 5. Comparação entre as métricas de detecção do mecanismos**

| Nr Atacantes: 150 | PARK 2008 | ZHU 2009 | SDA-COG |
|---|---|---|---|
| **VP** | 44.01 | 43.18 | **45.14** |
| **FP** | 5.44 | 4.22 | 1.30 |
| **FN** | 5.99 | 6.82 | 4.86 |
| **VN** | 44.56 | 45.78 | **48.70** |

### 5.1. Calibração dos Pesos dos Mecanismos de Detecção

Após as fases anteriores, uma grande massa de dados foi produzida. Uma parte dessa massa então é utilizada como dados de treinamento para o algoritmo genético (AG) realizar a atribuição de pesos aos mecanismos de detecção. É calculada a Função Detecção *DLRC-i*, do rádio cognitivo *i*. Considerando que: *L-i* representa a análise realizada pelo componente *Localizacao* quanto à identificação de ataque *PUE*, onde *L-i* = 0 (o rádio *i* não é um atacante de *PUE*), ou *L-i* = 1 (o rádio *i* é um atacante de *PUE*). *R-i* representa a análise realizada pelo componente *Reputacao* quanto à identificação de ataque *SSFF*, onde *R-i* = 0 (o rádio *i* não é um atacante de *SSFF*), ou *R-i* = 1 (o rádio *i* é um atacante de *SSFF*). Para determinar a detecção de ataques *DLR-i*, combinam-se os valores *L-i* e *R-i* usando dois coeficientes de calibração $\gamma$ e $\omega$ da seguinte forma:

$$DLR\text{-}i = \gamma L + \omega R \tag{7}$$

Onde: $\gamma + \omega = 1$. Para tal a função DLR*i* ( Equação 7) é ponderada por meio da criação de 5 cromossomos (Equação 8) – o Mecanismo de Localização é representado por dois cromossomos: *L1*: Mecanismo de Detecção de Ataque PUE Selfish, e; *L2*: Mecanismo de Detecção do Ataque PUE Malicioso. Quanto ao Mecanismo de Reputação, este é representado por três cromossomos: *R1*: Mecanismo de Detecção de Ataque SSFF-SL; *R2*: Mecanismo de Detecção de Ataque SSFF-SO, e; *R3*: Mecanismo de Detecção de Ataque SSFF-SF.

$$DLR_i = (\gamma_1 L_1 + \gamma_2 L_2) + (\omega_1 R_1 + \omega_2 R_2 + \omega_3 R_3) \tag{8}$$

Cada cromossomo é composto por 30 genes, formando a primeira geração [Lacerda 1999]. Cada conjunto de genes (cromossomo) forma um vetor de peso. Cada gene, variando de [0-1], representa os pesos de cada mecanismo ($\gamma$ para os mecanismos de Localização, e $\omega$ para os mecanismos de Reputação). São realizados os cruzamentos, seleções e mutações por um número total de 100 gerações nas 30 rodadas de simulações com todos os ataques, Tabela

 ©2012 SBC — Soc. Bras. de Computação

1. Os valores representativos dos pesos que consistem na solução que corresponda ao ponto de máximo da função DLR*i* são (Equação 9):

$$DLR_i = (\mathbf{0{,}199}L_1 + \mathbf{0{,}321}L_2) + (\mathbf{0{,}035}R_1 + \mathbf{0{,}342}R_2 + \mathbf{0{,}103}R_3) \quad (9)$$

Podemos observar que os pesos atribuídos pelo AG condizem com a realidade da degradação da rede (Figura 3), onde o ataque mais agressivo observado é o de SSFF-SO, cujo peso de 0.342 é o de maior valor na função de detecção. Da mesma forma o peso para a detecção do ataque de SSF-SL, 0.035, reflete o ataque de menor expressão, quando comparados aos outros. Aplicando-se o mecanismo com os pesos ponderados pelo AG à massa de treinamento obtemos os valores observados na Tabela 6.

**Tabela 6. Mecanismos Calibrados pelo AG e simulados com os cinco tipos de ataques**

| Nr Atacantes | 5 | 25 | 50 | 75 | 100 | 125 | 150 | Média |
|---|---|---|---|---|---|---|---|---|
| VP | 50,00 | 50,00 | 50,00 | 49,31 | 44,87 | 38,45 | 34,29 | 45,65 |
| FP | 0,00 | 0,00 | 0,00 | 0,69 | 5,13 | 11,55 | 15,71 | 4,35 |
| FN | 0,00 | 0,00 | 0,00 | 0,00 | 1,16 | 3,02 | 5,15 | 1,23 |
| VN | 50,00 | 50,00 | 50,00 | 50,00 | 48,84 | 46,98 | 44,85 | 48,77 |

Finalizando, uma nova massa de dados, diferente dos dados utilizados para o treinamento do AG, é analisada com o mecanismo de detecção, agora com pesos ponderados pela função de detecção (Equação 9), cujos resultados medianos aumentaram a taxa de VP em cerca de 0.5 ponto percentual, isto é: VP=45.63, demonstrando que realmente há a necessidade de atribuição de pesos para a detecção de ataque combinados às redes, Tabela 7.

**Tabela 7. Mecanismos Calibrados pelo AG e simulados com os cinco tipos de ataques**

| Nr | 5 | 25 | 50 | 75 | 100 | 125 | 150 | Média |
|---|---|---|---|---|---|---|---|---|
| VP | 50.00 | 50.00 | 50.00 | 49.44 | 44.91 | 38.54 | 34.33 | 45.63 |
| FP | 0,00 | 0,00 | 0,00 | 0,56 | 5,09 | 11,46 | 15,67 | 4,37 |
| FN | 0,00 | 0,00 | 0,00 | 0,00 | 1,26 | 3,12 | 5,34 | 1,24 |
| VN | 50,00 | 50,00 | 50,00 | 50,00 | 48,74 | 46,88 | 44,66 | 48,76 |

## 6. Conclusão

Este trabalho teve como objetivos apresentar um estudo sobre a detecção de ataques e propor um SDA baseado em técnicas de localização e classificação de reputação para redes de rádios cognitivos. Foi simulada uma rede com a criação dos diversos perfis de comportamento dos rádios cognitivos necessários à análise do desempenho da rede sob condições de ataques. Tal SDA foi programado de forma a atender, então, às demandas e restrições deste tipo de rede. Nos experimentos realizados, observou-se a degradação da capacidade de utilização oportunística de ocupação dos canais licenciados livres por parte dos rádios, proporcionalmente ao aumento do número de atacantes. O SDA mostrou-se eficaz na detecção, tanto de ataques isolados, como também aqueles realizados de forma conjunta. Com o uso do mecanismo de reputação, associado ao de localização, houve uma convergência eficiente da rede como um todo na identificação de ataques sofridos pela rede. Principalmente após a calibração, em uma primeira fase, do mecanismo de reputação, e em seguida, da calibração do SDA com um todo. Em trabalhos futuros serão investigadas outras opções de cálculo do limiar de aceitabilidade, bem como o do valor do coeficiente de normalização, com o intuito de analisar uma melhor forma de detecção do intruso. Pretende-se também realizar outros tipos de testes para a atribuição de pesos a cada mecanismo utilizado, com a finalidade de aumentar a eficiência e a eficácia do SDA.

 ©2012 SBC — Soc. Bras. de Computação

## Referências


Akyildiz, Ian F.; VURAN, Mehmet C.; LEE, Won-Yeol; MOHANTY, Shantidev, (2006a). "NeXt generation/dynamic spectrum access/cognitive radio wireless networks:A survey**".** Computer Networks, n. 50.

Akyildiz Ian F, J. Laskar, and Y. Li, (2006b), "OCRA: OFDM-based cognitive radio networks," Broadband Wireless Networking Laboratory Technical Report, March.

Akyildiz, Ian, F., LEE, WON-YEOL, VURAN, M. C., & MOHANTY S, (2008). "A Survey on Spectrum Management in Cognitive Radio Networks**".** IEEE Communications Magazine,

Clancy,T, Goergen N , (2008). "Security in cognitive radio networks: threats and mitigation". Third International (CrownCom),; 1–8. DOI: 10.1109 / CROWNCOM .2008.4562534.

Clancy, T, A. Khawar, (2009)."Security Threats to Signal Classifiers Using Self-Organizing Maps", Fourth International CrownCom)

Clemen, R. T.; REILLY, T. (2001) Making Hard Decisions: An Introduction to Decision Analisys. Pacific Grove: Duxbury, 2001.

Fcc (2003), Federal Communication Commission Home Page. http://www. fcc.gov -

Garmeman D. e Migon, H.S., (1993). Inferência estatística: uma abordagem integrada. Textos de Métodos Matemáticos do Instituto de Matematica, UFRJ

Hata, M. Empirical formula for propagation loss in land mobile radio services. IEEETrans. On Vehicular Technology, v. VT-29, n. 3, p. 317-325, aug. 1980.

Inatel (2009) Instituto Nacional de Telecomunicações– "Projeto Sendora", Curso de Mestrado em Telecomunicações - http://www.inatel.br/

Lacerda, E.G.M e Carvalho, A.C.P.L. (1999) "Introdução aos algoritmos genéticos", In: Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais. Editado por Galvão, C.O., Valença, M.J.S. Ed. Universidade/UFRGS: Ass Bras de Recursos Hídricos.

Leon, Olga, Juan Hernandez-Serrano, Miguel Soriano (2010). "Securing cognitive radio networks".International Journal of Communication Systems, vol 23,. ISSN/ISBN 1074-

Lopes C. L. (2008) A escolha de um custodiante para uma administradora financeira: Análise multiatributo por medições conjuntas e trocas justas. Mestrado em adm, Ibmec – RJ,

McHenry, M. (2003). Frequency agile spectrum access technologies, FCC Workshop CR.

Mitola, J (2000). "Cognitive radio: An integrated agent architecture for software defined radio". Ph.D. Dissertation, KTH Royal Institute of Technology, Stockholm, Sweden.

Mitola III and G. Q. Maguire, Jr. (2009), "Cognitive Radio: Making Software Radios More Personal," IEEE Personal Communications, Vol. 6, No. 4, pp. 13-18, Aug.

Misha ,S., M Sahai, A., Brodersen, R.W., (2006). Cooperative Sensing among Cognitive Radios. In: IEEE International Conference on Communications, ICC 2006, 1658–1663

Park, J., M. R. Chen, , and J. H. Reed**,** (2008). **"**Defense against primary user emulation attacks in cognitive radio networks**,"** IEEE Journal on Selected Areas in CSICRTA Vol. 26, No. 1,

Ross T., P. Myllymaki, and H. Tirri, (2002). "A statistical modeling approach to location estimation," IEEE Trans. Mobile Computing, Vol. 1, Jan.-March, p 59–69.

Ruiliang, C., Jung-Min, P., Hou, Y.T., Reed, J.H. (2008a): Toward Secure Distributed Spectrum Sensing in Cognitive Radio Networks. IEEE Communications Magazine 46,50–

Ruiliang, C, Jung-Min, P., Kaigui, B. (2008b).: Robust Distributed Spectrum Sensing inCognitive Radio Networks. In: IEEE The 27th, INFOCOM 2008, pp. 1876–1884

Shrestha , J., Sunkara, A. , Thirunavukkarasu, B., (2010). Security in Cognitive Radio. San Jose: San Jose State University.

Thomas R. W., L. A. Da Silva, and A. B. MacKenzie, (2005) "Cognitive Networks," *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005.*,

Zhu, Feng and Seung-Woo Seo, (2009). "'Enhanced Robust Cooperative Spectrum Sensing in Cognitive Radio", JCN, Special Issue on Cognitive Radio: A Path in the Evol of Public


 ©2012 SBC — Soc. Bras. de Computação

# Aprimoramento de Protocolo de Identificação Baseado no Problema $\mathcal{MQ}$

**Fábio S. Monteiro**[1 2], **Denise Goya e Routo Terada**[1]

[1]Departamento de Ciência da Computação – IME – USP

[2]Centro de Coordenação de Estudos da Marinha em São Paulo – Marinha do Brasil

{fsm, dhgoya, rt}@ime.usp.br

***Abstract.*** *The $\mathcal{MQ}$ problem, which consists in solving a system of multivariate quadratic polynomials over finite field, has attracted the attention of researchers for the development of public-key cryptosystems because (1) it's NP-complete, (2) there is no known polynomial-time algorithm for it's solution, even in the quantum computational model, and (3) enable cryptographic primitives of practical interest. In 2011, Sakumoto, Shirai and Hiwatari presented two new zero-knowledge identification protocols based exclusively on the $\mathcal{MQ}$ problem. The 3-pass identification protocol of Sakumoto et al. has knowledge error 2/3. In this paper, we propose an improvement that reduces the knowledge error to 1/2. The result is a protocol that reduces the total communication needed and requires a smaller number of rounds for the same security level.*

***Resumo.*** *O problema $\mathcal{MQ}$, que consiste em resolver um sistema de equações polinomiais multivariáveis quadráticas sobre um corpo finito, tem atraído a atenção de pesquisadores para o desenvolvimento de sistemas criptográficos de chave pública por ser (1) NP-completo, (2) não ter algoritmo conhecido de tempo polinomial para sua solução nem mesmo no modelo computacional quântico e (3) viabilizar primitivas criptográficas de interesse prático. Em 2011, Sakumoto, Shirai e Hiwatari apresentaram dois novos protocolos de identificação de conhecimento-zero baseados exclusivamente no problema $\mathcal{MQ}$. O protocolo em 3 passos de Sakumoto et al. apresenta probabilidade de personificação de 2/3 em uma rodada. No presente artigo é proposto um protocolo aprimorado que reduz essa probabilidade para 1/2. O resultado é um protocolo que diminui a comunicação total necessária e requer um número menor de iterações para o mesmo nível de segurança.*

## 1. Introdução

Protocolos de identificação são peças fundamentais para a construção de mecanismos de autenticação e assinaturas digitais. Quando baseados no modelo de chave pública, sua segurança depende da hipótese de intratabilidade de um determinado problema computacional. O problema de encontrar solução para um sistema de equações polinomiais multivariáveis quadráticas sobre um corpo finito, conhecido por Problema $\mathcal{MQ}$, reúne propriedades de interesse para o desenvolvimento de primitivas criptográficas, visto que é um problema NP-completo [Garey e Johnson 1979], não se conhecendo, até hoje, solução de tempo polinomial, nem mesmo no modelo computacional quântico [Bernstein et al. 2009] e tem possibilitado o desenvolvimento de protocolos com desempenho prático, como em [Petzoldt et al. 2011].

 ©2012 SBC — Soc. Bras. de Computação

Durante a conferência CRYPTO 2011, Sakumoto, Shirai e Hiwatari apresentaram uma nova forma de construir parâmetros públicos e secretos com base no problema $\mathcal{MQ}$, de modo a eliminar a hipótese de dificuldade do problema de isomorfismo de polinômios (IP), presente em vários outros trabalhos [Sakumoto et al. 2011]. Os autores apresentaram dois protocolos de identificação de conhecimento-zero baseados exclusivamente no problema $\mathcal{MQ}$, aplicando uma técnica de divisão do segredo e a forma polar (com propriedade de bilinearidade) de uma função $\mathcal{MQ}$. Para reduzir ao mínimo possível a comunicação, eles utilizaram ainda uma técnica criada originalmente por Jacques Stern [Stern 2006], que consiste em enviar apenas um hash de todos os compromissos no primeiro passo e depois incluir em $Rsp$ aqueles compromissos que não podem ser verificados, para que juntamente com os compromissos reconstituídos pelo verificador seja possível efetuar um novo hash para validar o recebido inicialmente. No artigo de Sakumoto, Shirai e Hiwatari todos os dados atinentes ao tamanho da comunicação do protocolo consideram a utilização desta técnica e por isso iremos assumir que ela será obrigatoriamente utilizada, visto que nosso objetivo principal é, de fato, a diminuição do tamanho da comunicação do protocolo, objetivando atender aplicações onde o consumo de energia é crítico e a complexidade de comunicação requer mais energia do que a complexidade de processamento local, como em redes de sensores sem fio. Chamaremos o protocolo em 3 passos de Sakumoto, Shirai e Hiwatari de $\mathcal{MQID}$-3 e, consequentemente, de $\mathcal{MQID}$-5 o protocolo original em 5 passos.

$\mathcal{MQID}$-3, como o nome sugere, funciona com três trocas de mensagens entre provador e verificador e apresenta probabilidade de personificação de 2/3 em uma rodada. Para alcançar um nível de segurança de $2^{-30}$, comparável a propostas anteriores, esse protocolo precisa ser iterado 52 vezes.

## Contribuições e Organização deste Trabalho

No presente artigo, propomos a construção de uma versão aprimorada do protocolo de identificação $\mathcal{MQID}$-3, apresentado durante a CRYPTO'2011 por Sakumoto, Shirai e Hiwatari [Sakumoto et al. 2011], utilizando uma construção que diminui a probabilidade de personificação em uma rodada de 2/3 para 1/2, elevando o nível de segurança do protocolo e diminuindo o tamanho da comunicação necessária em uma execução. Nossa proposta apresenta uma diminuição de 9,5% no tamanho da comunicação, sempre considerando a utilização da técnica sugerida por Jacques Stern [Stern 2006] para reduzir o número de compromissos trafegados, enviando apenas um hash de todos os compromissos no primeiro passo do protocolo e depois incluindo na resposta do provador apenas aqueles que não podem ser reconstituídos, para que juntamente com os compromissos validados pelo verificador seja possível efetuar um novo hash para comparar com o recebido inicialmente. Nosso protocolo preserva o tamanho dos parâmetros do sistema e também independe da hipótese de dificuldade do problema de isomorfismo de polinômios.

O restante deste documento está organizado da seguinte forma: na Seção 2, estabelecemos algumas notações e terminologia necessárias, bem como revemos algumas definições que nos servirão como base para o desenvolvimento do artigo. Na Seção 3, explicamos o problema $\mathcal{MQ}$, seu uso em criptografia de chave pública e a formulação de Sakumoto *et al.* para os parâmetros do criptossistema. Na Seção 4, revisamos brevemente o protocolo $\mathcal{MQID}$-3 apresentado em [Sakumoto et al. 2011] para, então, propormos

 ©2012 SBC — Soc. Bras. de Computação

modificações na Seção 5, onde o protocolo aprimorado de 3 passos é descrito e analisado. Por fim, na Seção 6, listamos algumas considerações e propomos trabalhos que podem ser derivados deste.

## 2. Fundamentos

Iniciaremos estabelecendo algumas notações e terminologia, bem como revendo definições que nos servirão como base. Durante este texto utilizaremos $\mathbb{F}$ ou $\mathbb{F}_q$ para designar um corpo finito de Galois de ordem $q$. Quando utilizamos $\mathbb{F}^n$ estamos falando do espaço vetorial de dimensão $n$ sobre $\mathbb{F}$. Em nossa notação, $a \in_R \mathcal{D}$ determina que um valor $a$ pertencente ao domínio $\mathcal{D}$ é escolhido aleatoriamente, com igual probabilidade, entre todos os elementos pertencentes a $\mathcal{D}$.

### 2.1. Esquemas de Identificação

Esquemas de identificação são peças fundamentais para a construção de mecanismos de autenticação e assinaturas digitais. Seu objetivo é possibilitar que uma entidade $\mathcal{B}eto$, também chamado de verificador, possa validar com segurança a identidade de uma outra entidade $\mathcal{A}lice$, também chamada de provadora. Obviamente, existem diversas maneiras de se alcançar o objetivo proposto. Podemos, por exemplo, considerar o *login* de um usuário em um sistema operacional como sendo um esquema de identificação baseado em chave secreta, onde o usuário é o provador e o SO é o verificador. Neste exemplo, o verificador simplesmente conhece a chave secreta do provador e faz a checagem validando-o. Em nosso trabalho no entanto, vamos focar apenas em esquemas de identificação baseados no modelo de chave pública, mais especificamente vamos utilizar apenas esquemas de identificação de conhecimento-zero, por isso, a partir de agora, quando nos referirmos a esquema de identificação, estamos falando de um esquema de conhecimento-zero.

Para formalizarmos nossa definição dizemos que, um esquema de identificação $\mathcal{ID}$ é uma tupla de algoritmos composta por Setup, Gen, P e V. Onde Setup recebe um parâmetro de segurança $1^\kappa$ e devolve um parâmetro de sistema *param*. Gen é o algoritmo de geração de chaves que utiliza *param* e retorna um par de chaves sk e pk, onde sk é a chave privada e pk a chave pública. P e V constituem um protocolo interativo de identificação, onde P é executado por $\mathcal{A}lice$ (provadora), utilizando sua chave privada, e V é utilizado por $\mathcal{B}eto$ (verificador), com acesso a chave pública de $\mathcal{A}lice$. A seguir estabelecemos as propriedades requeridas para o nosso esquema de identificação:

**Correção** (*completeness*) $-$ Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele satisfaz a noção de correção se, dados uma provadora honesta $\mathcal{A}lice$ e um verificador honesto $\mathcal{B}eto$, a execução do protocolo resulta no aceite da identidade de $\mathcal{A}lice$ por $\mathcal{B}eto$, ou seja:

$$\Pr[\mathrm{V}(pk_{\mathcal{A}lice}, Cmt\|Ch\|\mathrm{P}(sk_{\mathcal{A}lice}, Cmt\|Ch)) = 1] = 1$$

**Solidez** (*soundness*) $-$ Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele possui solidez se, nenhum falso provador puder convencer um verificador honesto a aceitar sua interação como sendo de uma outra entidade, exceto com uma probabilidade negligenciável. Isto é, se um adversário $\mathcal{C}arlos$ tentar personificar $\mathcal{A}lice$ perante $\mathcal{B}eto$, sua chance de sucesso será desprezível ou, formalmente, menor que $\epsilon$.

$$\Pr[\mathrm{V}(pk_{Alice}, Cmt\|Ch\|\mathrm{P}(sk_{falsa}, Cmt\|Ch)) = 1] < \epsilon$$

 ©2012 SBC — Soc. Bras. de Computação

**Conhecimento-Zero** (*zero-knowledge*) − Seja $\mathcal{ID}$ um esquema de identificação, dizemos que ele é de conhecimento-zero se, caso a afirmação seja verdadeira, nenhum verificador aprende outra coisa senão este fato.

Quando o esquema de identificação é composto de 3 passos, normalmente nomeados como: comprometimento, desafio e resposta, ele é chamado de canônico por Abdalla *et al* [Abdalla et al. 2002] que especifica esta condição como um dos requisitos para possibilitar a geração de um esquema de assinatura digital, seguro no modelo do oráculo aleatório [Bellare e Rogaway 1993], utilizando-se o paradigma da transformação Fiat-Shamir [Fiat e Shamir 1987].

## 2.2. Esquemas de Comprometimento

Para construção do esquema de identificação de conhecimento-zero utilizaremos um esquema de comprometimento (*Commitment Scheme*), que tem por objetivo esconder temporariamente um valor que não pode ser alterado.

Podemos ilustrar o funcionamento de um esquema de comprometimento com um remetente que tranca uma mensagem em uma caixa e a entrega para um destinatário. Podemos ver que a mensagem está escondida, visto que se encontra trancada no interior da caixa, como queríamos desde o princípio. Também percebemos que não é mais possível ao remetente alterar essa mensagem, pois já entregou a caixa ao destinatário. Posteriormente, quando convier ao remetente, ele entrega a chave para o destinatário possibilitando que ele confira a mensagem. Vemos assim que um esquema de comprometimento funciona em duas fases distintas. Na primeira, que chamamos de fase de comprometimento, um remetente irá se "comprometer" com uma mensagem específica, entregando ao destinatário a "caixa trancada". Uma segunda fase, que chamamos de verificação, ocorrerá quando o remetente entregar ao destinatário a "chave" que possibilitará verificar a mensagem original.

Para formalizarmos este conceito, dizemos que um esquema de comprometimento $\mathcal{C}om$ é uma tupla com dois algoritmos chamados Empenha e Abre, onde Empenha recebe uma mensagem $msg \in \mathcal{M}$, sendo $\mathcal{M}$ o espaço de mensagens, e retorna um par c e d, onde c é um valor de compromisso e d é a chave para recuperar $msg$ com a utilização do algoritmo Abre e o próprio c. Abre retorna $msg$ ou então $\perp$, caso c seja inválido. Em um esquema de comprometimento correto deveremos ter que Abre(Empenha($msg$)) $= msg$. Precisamos ainda que este esquema de comprometimento possua as seguintes propriedades:

**Ocultação** (*hiding*) − Seja $\mathcal{C}om$ um esquema de comprometimento, dizemos que ele possui ocultação se é computacionalmente difícil descobrir qualquer informação sobre $msg$ a partir de c.

**Vinculação** (*binding*) − Seja $\mathcal{C}om$ um esquema de comprometimento, dizemos que ele possui vinculação se, efetuado Empenha($msg_1$) $\rightarrow$ c, d, é computacionalmente difícil formular um valor d' tal que Abre(c,d') = $msg_1$ ou Abre(c, d') $\neq \perp$.

Em [Halevi e Micali 1996] pode ser visto um modelo de esquema de comprometimento prático que é mostrado seguro e é construído a partir da utilização de funções de hash.

 ©2012 SBC — Soc. Bras. de Computação

## 3. Problema $\mathcal{MQ}$ e os Criptossistemas de Chave Pública Multivariável

Os sistemas criptográficos de chave pública amplamente utilizados hoje em dia têm sua segurança baseada na suposição da intratabilidade dos problemas de fatoração de inteiros, no caso de sistemas RSA, e do logaritmo discreto, em sistemas ElGamal ou de Curvas Elípticas. O fato de que tais problemas podem ser resolvidos em tempo polinomial com algoritmos quânticos [Shor 1997], tornaria inseguros os sistemas criptográficos atuais na presença de computadores quânticos com a capacidade adequada. Uma proposta para enfrentar este desafio é a utilização do Problema $\mathcal{MQ}$ como base para o desenvolvimento de sistemas criptográficos de chave pública seguros.

Inicialmente iremos apresentar o problema do Sistema de Equações Polinomiais Multivariáveis Simultâneas, seja $\mathcal{P} = (p_1, \ldots, p_m)$ um sistema de $m$ polinômios de grau $d$ em $n$ variáveis sobre um corpo finito $\mathbb{F}$ de ordem $q$, o problema do Sistema de Equações Polinomiais Multivariáveis Simultâneas consiste em encontrar $\mathrm{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ tal que $\mathcal{P}(\mathrm{x}) = \mathrm{y}$, sendo $\mathrm{y} = (y_1, \ldots, y_m) \in \mathbb{F}^m$:

$$\mathcal{P} = \begin{cases} p_1(x_1, ..., x_n) = y_1 \\ p_2(x_1, ..., x_n) = y_2 \\ \vdots \\ p_m(x_1, ..., x_n) = y_m \end{cases}$$

Quando o grau do sistema de equações $\mathcal{P}$ é igual a 2 $(d = 2)$, chamamos então de Problema $\mathcal{MQ}$. Patarin e Goubin demonstraram em [Patarin e Goubin 1997] que o problema $\mathcal{MQ}$ é NP-Completo, não se conhecendo até hoje nenhum algoritmo, nem mesmo quântico, de tempo polinomial que possa resolver este problema [Bernstein et al. 2009], contribuindo assim para a segurança de sistemas criptográficos baseados nesta primitiva. Estes criptossistemas foram de alguma forma originados do trabalho de Matsumoto e Imai [Matsumoto e Imai 1988] e são conhecidos como MPKC [Ding et al. 2006, Wolf e Preneel 2005, Wolf 2005] (acrônimo da nomenclatura em inglês que significa Criptossistema de Chave Pública Multivariável).

Chamamos o Sistema de Equações Polinomiais Multivariáveis Quadráticas $\mathcal{P}$ de Função $\mathcal{MQ}$ e a denotamos por $\mathcal{MQ}(n, m, \mathbb{F})$ para estabelecer suas propriedades. Uma função $\mathcal{MQ}$ é uma função de via-única (*one-way function*) [Patarin e Goubin 1997].

Uma característica dos MPKC conhecidos atualmente é que nenhum tem sua segurança baseada exclusivamente no problema $\mathcal{MQ}$, sendo a grande maioria dependente também do problema de Isomorfismo de Polinômios (IP), o qual podemos dizer de forma simplificada que, consiste em, dados $\mathcal{P}$ e $F$, encontrar duas transformações afins $S$ e $T$ tais que $S \circ F \circ T = S(F(T(x)))) = \mathcal{P}(x)$, sendo $\mathcal{P}$, uma função $\mathcal{MQ}$ utilizada como chave pública de um MPKC e $F$ um mapeamento inversível utilizado juntamente com $S$ e $T$ como chave secreta do mesmo criptossistema. Acredita-se que uma instância aleatória do problema IP seja difícil, porém, quando aplicado às atuais $\mathcal{MQ}$-Trapdoor ainda suscita controvérsia quanto à sua intratabilidade, já tendo inclusive possibilitado a quebra de vários esquemas e protocolos criptográficos anteriormente apresentados, como o esquema de assinatura SFlash [Patarin et al. 2001, Courtois et al. 2003, Dubois et al. 2007, Bouillaguet et al. 2011], que chegou a ser recomendado pelo NESSIE. Esta vulnerabilidade existe pois as atuais $\mathcal{MQ}$-Trapdoor possuem esta estrutura específica de construção que consiste na composição de um mapeamento central $F$ de formato determinado, com

 ©2012 SBC — Soc. Bras. de Computação

duas transformações afins ($S$ e $T$), resultando em uma forma limitada de função $\mathcal{MQ}$, e que, em consequência, possibilita em alguns casos soluções fáceis desta versão do problema IP.

### 3.1. Novo Modelo MPKC de Sakumoto-Shirai-Hiwatari

Recentemente, Sakumoto, Shirai e Hiwatari apresentaram em [Sakumoto et al. 2011] dois novos protocolos de identificação de conhecimento-zero (*zero-knowledge*) os quais chamamos de $\mathcal{MQID}-3$ e $\mathcal{MQID}-5$. Estes protocolos se baseiam em um novo modelo de MPKC, o qual tem sua segurança reduzida ao problema $\mathcal{MQ}$, visto que utiliza uma instância aleatória de função $\mathcal{MQ}$, em vez da forma limitada descrita anteriormente. Neste novo modelo uma chave privada s passa a ser os valores das $n$ variáveis nas $m$ equações que compõe a função $\mathcal{MQ}$, isto é, s $= (x_1, ..., x_n) \in \mathbb{F}^n$, sendo a chave pública então o resultado v $= (y_1, ..., y_m) \in \mathbb{F}^m$ e a função $\mathcal{MQ}$ em si passa a ser uma instância aleatória disponível como parâmetro do sistema para todos os usuários. Por se tratar de uma instância aleatória de função $\mathcal{MQ}$, não se consegue construir uma trapdoor a partir dela.

## 4. Revisão do Esquema de Identificação de Sakumoto-Shirai-Hiwatari

No esquema de identificação de Sakumoto-Shirai-Hiwatari o algoritmo Setup recebe um parâmetro de segurança $1^\kappa$ que determina $n$, $m$ e $\mathbb{F}$ e retorna um parâmetro de sistema $\mathcal{P} \in_R \mathcal{MQ}(n, m, \mathbb{F})$, o qual é uma instância aleatória de função $\mathcal{MQ}$. O algoritmo Gen sorteia uma chave privada aleatória s $\in \mathbb{F}^n$, e executa $\mathcal{P}$(s) =v, onde v será então a chave pública.

O protocolo $\mathcal{MQID}$-3 utiliza uma função bilinear $\mathcal{G}$, que chamamos de forma polar de $\mathcal{P}$, sendo $\mathcal{G}(a, b) = \mathcal{P}(a + b) - \mathcal{P}(a) - \mathcal{P}(b)$, com $a, b \in \mathbb{F}^n$.

Os autores utilizaram uma abordagem de divisão-escolha, onde o segredo é dividido em vários pedaços e o verificador escolhe algum para validar. A ideia básica é que o provador demonstre que possui uma tupla $(r_0, r_1, t_0, t_1, e_0, e_1)$ que satisfaça:

$$\mathcal{G}(t_0, r_1) + e_0 = \mathcal{P}(\mathrm{s}) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1 \tag{1}$$

$$\text{e} \qquad (t_0, e_0) = (r_0 - t_1, \mathcal{P}(r_0) - e_1) \tag{2}$$

A equação (1) acima vem da seguinte maneira, utilizando-se a bilinearidade de $\mathcal{G}$:

$$\begin{aligned}
\mathcal{G}(r_0, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\
\mathcal{G}(t_0, r_1) + \mathcal{G}(t_1, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) \\
\mathcal{G}(t_0, r_1) &= \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) \\
\mathcal{G}(t_0, r_1) &= \mathcal{P}(s) - (e_0 + e_1) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) \\
\mathcal{G}(t_0, r_1) + e_0 &= \mathcal{P}(s) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1
\end{aligned}$$

Para isso a chave secreta $s$ é dividida da forma descrita a seguir. Inicialmente são escolhidos $r_0, t_0 \in_R \mathbb{F}^n$ e $e_0 \in_R \mathbb{F}^m$, e então calculados $r_1 \leftarrow$ s $- r_0$, $t_1 \leftarrow r_0 - t_0$ e $e_1 \leftarrow \mathcal{P}(r_0) - e_0$, ou seja, s$= r_0 + r_1 = r_1 + t_0 + t_1$ e $\mathcal{P}(r_0) = e_0 + e_1$.

A Figura 1 apresenta o protocolo $\mathcal{MQID}$-3 original.

 ©2012 SBC — Soc. Bras. de Computação

Entrada do Provador: $((\mathcal{P}, \mathrm{v}), \mathrm{s})$ | Entrada do Verificador: $(\mathcal{P}, \mathrm{v})$

Escolhe $r_0, t_0 \in_R \mathbb{F}^n$, $e_0 \in_R \mathbb{F}^m$
$r_1 \leftarrow \mathrm{s} - r_0$, $t_1 \leftarrow r_0 - t_0$
$e_1 \leftarrow \mathcal{P}(r_0) - e_0$
$c_0 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$
$c_1 \leftarrow Com(t_0, e_0)$
$c_2 \leftarrow Com(t_1, e_1)$
$hc \leftarrow H(c_0, c_1, c_2)$

$\xrightarrow{hc}$

Escolhe $Ch \in_R \{0, 1, 2\}$

$\xleftarrow{Ch}$

Se $Ch = 0, Rsp \leftarrow (r_0, t_1, e_1, c_0)$
Se $Ch = 1, Rsp \leftarrow (r_1, t_1, e_1, c_1)$
Se $Ch = 2, Rsp \leftarrow (r_1, t_0, e_0, c_2)$

$\xrightarrow{Rsp}$

Se $Ch = 0$, utiliza $Rsp = (r_0, t_1, e_1, c_0)$ para verificar:
$c'_1 \leftarrow Com(r_0 - t_1, \mathcal{P}(r_0) - e_1)$
$c'_2 \leftarrow Com(t_1, e_1)$
$hc \stackrel{?}{=} H(c_0, c'_1, c'_2)$
Se $Ch = 1$, utiliza $Rsp = (r_1, t_1, e_1, c_1)$ para verificar:
$c'_0 \leftarrow Com(r_1, \mathrm{v} - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1)$
$c'_2 \leftarrow Com(t_1, e_1)$
$hc \stackrel{?}{=} H(c'_0, c_1, c'_2)$
Se $Ch = 2$, utiliza $Rsp = (r_1, t_0, e_0, c_2)$ para verificar:
$c'_0 \leftarrow Com(r_0, \mathcal{G}(t_0, r_1) + e_0)$
$c'_1 \leftarrow Com(t_0, e_0)$
$hc \stackrel{?}{=} H(c'_0, c'_1, c_2)$

**Figura 1. Resumo do protocolo de identificação $\mathcal{MQID}$-3 de Sakumoto, Shirai e Hiwatari**

O protocolo $\mathcal{MQID}$-3 possui probabilidade de personificação de 2/3 em uma rodada. Por isso, para termos uma probabilidade total de personificação menor ou igual a $2^{-\lambda}$, é preciso repetir o protocolo um total de pelo menos $\lceil \frac{\lambda}{log_2 3 - 1} \rceil$ vezes, como podemos ver abaixo, onde $r$ é o número total de rodadas necessárias:

$$\left(\frac{2}{3}\right)^r \leq 2^{-\lambda}$$
$$r(log_2 2 - log_2 3) \leq -\lambda \cdot log_2 2$$
$$r(1 - log_2 3) \leq -\lambda$$
$$r(log_2 3 - 1) \geq \lambda$$
$$r \geq \frac{\lambda}{log_2 3 - 1}$$

A comunicação em uma rodada é igual a $2n + 5m + 2$ bits, visto que o provador gera um hash $hc$ dos 3 compromissos criados, o qual possui $2m$ bits, e o envia para o verificador, que responde com um desafio $Ch$ de tamanho 2 bits, do qual decorre o envio

 ©2012 SBC — Soc. Bras. de Computação

de $Rsp$ pelo provador, o qual possui $2n + 3m$ bits. Vemos então que a comunicação total seria de $r \cdot (2n+5m+2)$ bits, ou $52 \cdot (2 \cdot 84+5 \cdot 80+2) = 29640$ bits, com os parâmetros originalmente propostos.

## 5. $\mathcal{MQID}$-3 Aprimorado

Primeiramente vale destacar que também em [Sakumoto et al. 2011] foi introduzido o protocolo $\mathcal{MQID}$-5, o qual já possui probabilidade de personificação 1/2 em uma rodada. Porém, ele não é canônico [Abdalla et al. 2002], o que impossibilitaria a geração a partir dele de um esquema de assinaturas pelo paradigma da transformação Fiat-Shamir [Fiat e Shamir 1987]. Muito recentemente, Alaoui et al [Alaoui et al. 2012] propuseram uma extensão do Forking Lemma de Pointcheval e Stern [Pointcheval e Stern 1996, Pointcheval e Stern 2000] e da transformação Fiat-Shamir para possibilitar a utilização de protocolos com $2n+1$ passos, utilizando exatamente o protocolo $\mathcal{MQID}$-5 de Sakumoto, Shirai e Hiwatari como exemplo, porém essa assinatura seria teoricamente mais lenta que uma baseada na nossa proposta, visto que a solução derivada do protocolo em 5 passos necessita de $2r$ hashes para simular o verificador, enquanto a nossa utilizaria apenas um hash.

Para conseguirmos aprimorar o protocolo de identificação $\mathcal{MQID}$-3, incrementamos a divisão do segredo de forma a possibilitar duas alternativas exclusivas de verificação. Vimos que no protocolo original a chave privada s é dividida em $r_0$ e $r_1$, sendo s$= r_0+r_1$. E que depois é efetuada a divisão de $r_0$ em $t_0$ e $t_1$ e de $\mathcal{P}(r_0)$ em $e_0$ e $e_1$, da mesma forma. Nós mantemos as mesmas divisões originais e acrescentamos a divisão de $r_1$ em $d_0$ e $d_1$ e de $\mathcal{P}(r_1)$ em $u_0$ e $u_1$, efetuando o seguinte: escolhemos $d_0 \in_R \mathbb{F}^n$ e $u_0 \in_R \mathbb{F}^m$, e calculamos $d_1 \leftarrow r_1 - d_0$ e $u_1 \leftarrow \mathcal{P}(r_1) - u_0$. Podemos ver então que a Equação (1) em função de $r_1$ pode ser reescrita em função de $r_0$: $\mathcal{G}(r_0, d_1) + u_1 = \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0$. Assim, propomos que o provador demonstre que possui uma tupla $(r_0, r_1, t_0, t_1, e_0, e_1, d_0, d_1, u_0, u_1)$ que satisfaça (1 e 4) OU (3 e 4):

$$\mathcal{G}(t_0, r_1) + e_0 = \mathcal{P}(s) - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1 \quad (1)$$

$$\mathcal{G}(r_0, d_1) + u_1 = \mathcal{P}(s) - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0 \quad (3)$$

$$(t_0, e_0) = (r_0 - t_1, \mathcal{P}(r_0) - e_1) \text{ E } (d_0, u_0) = (r_1 - d_1, \mathcal{P}(r_1) - u_1) \quad (4)$$

Na Figura 2 detalhamos o protocolo $\mathcal{MQID}$-3 aprimorado com probabilidade de personificação em uma rodada de 1/2.

### 5.1. Análise da Proposta

Vemos que nosso protocolo aprimorado satisfaz a definição de Correção, pois uma provadora honesta $\mathcal{A}lice$, utilizando sua chave secreta s, sempre será aceita por um verificador honesto $\mathcal{B}eto$.

Percebe-se que se um adversário $\mathcal{C}arlos$ executar o protocolo tentando personificar $\mathcal{A}lice$ sem a chave secreta dela, e utilizando um valor s' $\in_R \mathbb{F}^n$ em substituição, ele irá obter sucesso em 1/2 das vezes. Mais precisamente, sempre que $Ch \in \{1, 2\}$, quando o verificador não utiliza a chave pública na checagem e sim os valores enviados pelo próprio provador. É fato que no caso de $\mathcal{C}arlos$ trapacear para conseguir obter sucesso em um dos desafios onde é utilizada a chave pública de $\mathcal{A}lice$, ele inevitavelmente

 ©2012 SBC — Soc. Bras. de Computação

Entrada do Provador: $((\mathcal{P}, \mathrm{v}), \mathrm{s})$ Entrada do Verificador: $(\mathcal{P}, \mathrm{v})$

$c_0 \leftarrow Com(r_0, \mathcal{G}(r_0, d_1) + u_1)$
$c_1 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$
$c_2 \leftarrow Com(t_0, e_0)$
$c_3 \leftarrow Com(t_1, e_1)$
$c_4 \leftarrow Com(d_0, u_0)$
$c_5 \leftarrow Com(d_1, u_1)$
$hc \leftarrow H(c_0, c_1, c_2, c_3, c_4, c_5)$

$\xrightarrow{hc}$

Escolhe $Ch \in_R \{0, 1, 2, 3\}$

$\xleftarrow{Ch}$

Se $Ch = 0, Rsp \leftarrow (r_1, t_1, e_1, d_0, u_0, c_0, c_2)$
Se $Ch = 1, Rsp \leftarrow (r_1, t_0, e_0, d_1, u_1, c_0, c_3)$
Se $Ch = 2, Rsp \leftarrow (r_0, t_0, e_0, d_1, u_1, c_1, c_4)$
Se $Ch = 3, Rsp \leftarrow (r_0, t_1, e_1, d_0, u_0, c_1, c_5)$

$\xrightarrow{Rsp}$

Se $Ch = 0$, utiliza $Rsp = (r_1, t_1, e_1, d_0, u_0, c_0, c_2)$ para verificar:
$c'_1 \leftarrow Com(r_1, \mathrm{v} - \mathcal{P}(r_1) - \mathcal{G}(t_1, r_1) - e_1)$
$c'_3 \leftarrow Com(t_1, e_1)$
$c'_4 \leftarrow Com(d_0, u_0)$
$c'_5 \leftarrow Com(r_1 - d_0, \mathcal{P}(r_1) - u_0)$
$hc \stackrel{?}{=} H(c_0, c'_1, c_2, c'_3, c'_4, c'_5)$

Se $Ch = 1$, utiliza $Rsp = (r_1, t_0, e_0, d_1, u_1, c_0, c_3)$ para verificar:
$c'_1 \leftarrow Com(r_1, \mathcal{G}(t_0, r_1) + e_0)$
$c'_2 \leftarrow Com(t_0, e_0)$
$c'_4 \leftarrow Com(r_1 - d_1, \mathcal{P}(r_1) - u_1)$
$c'_5 \leftarrow Com(d_1, u_1)$
$hc \stackrel{?}{=} H(c_0, c'_1, c'_2, c_3, c'_4, c'_5)$

Se $Ch = 2$, utiliza $Rsp = (r_0, t_0, e_0, d_1, u_1, c_1, c_4)$ para verificar:
$c'_0 \leftarrow Com(r_0, \mathcal{G}(r_0, d_1) + u_1)$
$c'_2 \leftarrow Com(t_0, e_0)$
$c'_3 \leftarrow Com(r_0 - t_0, \mathcal{P}(r_0) - e_0)$
$c'_5 \leftarrow Com(d_1, u_1)$
$hc \stackrel{?}{=} H(c'_0, c_1, c'_2, c'_3, c_4, c'_5)$

Se $Ch = 3$, utiliza $Rsp = (r_0, t_1, e_1, d_0, u_0, c_1, c_5)$ para verificar:
$c'_0 \leftarrow Com(r_0, \mathrm{v} - \mathcal{P}(r_0) - \mathcal{G}(r_0, d_0) - u_0)$
$c'_2 \leftarrow Com(r_0 - t_1, \mathcal{P}(r_0) - e_1)$
$c'_3 \leftarrow Com(t_1, e_1)$
$c'_4 \leftarrow Com(d_0, u_0)$
$hc \stackrel{?}{=} H(c'_0, c_1, c'_2, c'_3, c'_4, c_5)$

**Figura 2. Protocolo de Identificação $\mathcal{MQID}$-3 Aprimorado**

causará a falha em um dos desafios que ele normalmente passaria, ou seja, ele sempre permanecerá com uma chance de personificação igual a 1/2 por rodada. Para ilustrarmos essa

 ©2012 SBC — Soc. Bras. de Computação

situação vamos considerar que $\mathcal{C}arlos$ gere $c_0 \leftarrow Com(r'_0, \text{v} - \mathcal{P}(r'_0) - \mathcal{G}(r'_0, d'_0) - u'_0)$ e $c_1 \leftarrow Com(r'_1, \text{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1)$ utilizando a chave pública da entidade que ele quer personificar. Neste caso, se $Ch \in \{0, 3\}$ $\mathcal{C}arlos$ terá sucesso na validação, pois o verificador $\mathcal{B}eto$ irá efetuar rigorosamente a mesma conta, obtendo o mesmo resultado. Porém, a chance continua sendo 1/2 visto que não será possível satisfazer, neste caso, as checagens de $c_0$ e $c_1$ quando $Ch \in \{1, 2\}$, ocorrendo o mesmo de forma análoga para outras alternativas de trapaça. Por uma questão de espaço não podemos detalhar neste artigo todos os casos onde um adversário tentando forjar a validade dos dados para um determinado desafio acaba, inevitavelmente, por impossibilitar a verificação de um outro, permanecendo sempre com uma probabilidade 1/2 de personificação em uma rodada. Estas demonstrações constarão de um futuro trabalho completo.

Vamos demonstrar que nosso protocolo é de conhecimento-zero, visto que um verificador trapaceiro utilizando um simulador $\mathcal{S}$, sem conhecimento da chave secreta, gera uma transcrição do protocolo estatisticamente indistinguível de uma transcrição gerada pela interação de uma provadora honesta $\mathcal{A}lice$ e um verificador honesto $\mathcal{B}eto$. E ainda, que nosso protocolo possui solidez, uma vez que dadas duas execuções do protocolo com o mesmo início (ou seja, os mesmos valores de compromisso) e continuações diferentes (ou seja, $Ch$ e $Rsp$ diferentes) é possível descobrir o segredo s ou então o esquema de comprometimento está quebrado, não possuindo vinculação, visto que teria gerado valores iguais de compromisso a partir de valores diferentes de $msg$.

**Teorema 1** *O Protocolo $\mathcal{MQID}$-3 aprimorado é de conhecimento-zero estatístico quando o esquema de comprometimento $\mathcal{C}om$ possui ocultação estatística.*

Demonstração: Seja $\mathcal{S}$ um simulador que irá personificar uma entidade qualquer utilizando apenas a chave pública v dessa entidade e possuindo acesso a um oráculo $\mathcal{O}$ que indica por meio de um bit se a próxima verificação será em função de $r_1$ ou de $r_0$, ou seja, se o próximo desafio a ser recebido será $Ch \in \{0, 1\}$ ou $Ch \in \{2, 3\}$, respectivamente. A cada rodada do protocolo, $\mathcal{S}$ consulta $\mathcal{O}$ e recebe dele um bit $Ch^*$ indicando que $Ch^* = 1 \rightarrow Ch \in \{0, 1\}$, $Ch^* = 0 \rightarrow Ch \in \{2, 3\}$. $\mathcal{S}$ então escolhe $\text{s}', r'_0, t'_0, d'_0 \in_R \mathbb{F}^n$, $e'_0, u'_0 \in_R \mathbb{F}^m$ e prepara $r'_1 \leftarrow s' - r'_0$, $t'_1 \leftarrow r'_0 - t'_0$ e $d'_1 \leftarrow r'_1 - d'_0$. Se $Ch^* = 0$ então calcula $e'_1 \leftarrow v - \mathcal{P}(s') + \mathcal{P}(r'_1) - e'_0$ e $u'_1 \leftarrow \mathcal{P}(r'_1) - u'_0$, senão $\mathcal{S}$ faz $e'_1 \leftarrow \mathcal{P}(r'_0) - e'_0$ e $u'_1 \leftarrow v - \mathcal{P}(s') + \mathcal{P}(r'_0) - u'_0$. Finalmente, $\mathcal{S}$ gera os valores de compromisso e os envia para o verificador: $c_0 \leftarrow Com(r'_0, \mathcal{G}(r'_0, d'_1) + u'_1)$, $c_1 \leftarrow Com(r'_1, \mathcal{G}(t'_0, r'_1) + e'_0)$, $c_2 \leftarrow Com(t'_0, e'_0)$, $c_3 \leftarrow Com(t'_1, e'_1)$, $c_4 \leftarrow Com(d'_0, u'_0)$ e $c_5 \leftarrow Com(d'_1, u'_1)$.

Após receber $Ch$, $\mathcal{S}$ responde normalmente com seus valores já calculados, ou seja, se $Ch = 0, Rsp \leftarrow (r'_1, t'_1, e'_1, d'_0, u'_0, c_0, c_2)$, ou se $Ch = 1, Rsp \leftarrow (r'_1, t'_0, e'_0, d'_1, u'_1, c_0, c_3)$, ou se $Ch = 2, Rsp \leftarrow (r'_0, t'_0, e'_0, d'_1, u'_1, c_1, c_4)$, ou se $Ch = 3, Rsp \leftarrow (r'_0, t'_1, e'_1, d'_0, u'_0, c_1, c_5)$.

Verificamos que nos casos em que $\mathcal{O}$ prediz corretamente o desafio, ou seja, em que $Ch^* = 1$ e $Ch \in \{0, 1\}$, ou então, $Ch^* = 0$ e $Ch \in \{2, 3\}$, os valores enviados por $\mathcal{S}$ serão aceitos por qualquer verificador executando normalmente o protocolo. Podemos confirmar o caso concreto quando $Ch^* = 1$ e $Ch = 0$, temos $e'_1 = \text{v} - \mathcal{P}(\text{s}') + \mathcal{P}(r'_1) - e'_0$, logo $c_1$ ao ser verificado será válido pois teremos então $\text{v} - \mathcal{P}(r'_1) - \mathcal{G}(t'_1, r'_1) - e'_1 = \mathcal{G}(t'_0, r'_1) + e_0$. O caso de $Ch^* = 1$ e $Ch = 1$ é mais simples, pois o verificador não utiliza a chave pública v, fazendo a reconstrução dos compromissos com os mesmos valores preparados pelo simulador, o que obviamente redunda no aceite da verificação. Para os

 ©2012 SBC — Soc. Bras. de Computação

casos em que $Ch^* = 0$, e consequentemente $Ch \in \{2, 3\}$, teremos a rigor a mesma situação que acabamos de enumerar, porém com as verificações em função $r_0$.

Vemos assim que um verificador desonesto utilizando $\mathcal{S}$ pode gerar uma transcrição do protocolo que é indistinguível de uma gerada pela interação de uma provadora honesta $\mathcal{A}lice$ e um verificador honesto $\mathcal{B}eto$, comprovando assim que um adversário não adquire conhecimento útil a partir de transcrições do protocolo, desde que o esquema de comprometimento $\mathcal{C}om$ possua ocultação estatística.

**Teorema 2** *O Protocolo $\mathcal{MQID}$-3 aprimorado é argumento de conhecimento com probabilidade de erro 1/2 em uma rodada quando o esquema de comprometimento $\mathcal{C}om$ possui vinculação computacional.*

Demonstração: Sejam $(hc_0, Ch_0, Rsp_0)$, $(hc_1, Ch_1, Rsp_1)$, $(hc_2, Ch_2, Rsp_2)$ e $(hc_3, Ch_3, Rsp_3)$ quatro transcrições da execução de $\mathcal{MQID}$-3 Aprimorado, tais que $Ch_i = i$ e $Dec(\mathcal{P}, \mathrm{v}, hc_i, Ch_i, Rsp_i) = 1$. Sejam $c_0, c_1, c_2, c_3, c_4, c_5$ valores de compromisso gerados de acordo com o estabelecido na definição do protocolo $\mathcal{MQID}$-3 Aprimorado, e ainda, sejam $hc_0 = hc_1 = hc_2 = hc_3 = H(c_0, c_1, c_2, c_3, c_4, c_5)$ e $Rsp_0 = (r_1^{(0)}, t_1^{(0)}, e_1^{(0)}, d_0^{(0)}, u_0^{(0)}, c_0, c_2)$, $Rsp_1 = (r_1^{(1)}, t_0^{(1)}, e_0^{(1)}, d_1^{(1)}, u_1^{(1)}, c_0, c_3)$, $Rsp_2 = (r_0^{(2)}, t_1^{(2)}, e_1^{(2)}, d_0^{(2)}, u_0^{(2)}, c_1, c_4)$ e $Rsp_3 = (r_0^{(3)}, t_0^{(3)}, e_0^{(3)}, d_1^{(3)}, u_1^{(3)}, c_1, c_5)$, temos então que:

$$c_0 = \mathcal{C}om(r_0^{(2)}, \mathcal{G}(r_0^{(2)}, d_1^{(2)}) + u_1^{(2)}) = \mathcal{C}om(r_0^{(3)}, \mathrm{v} - \mathcal{P}(r_0^{(3)}) - \mathcal{G}(r_0^{(3)}, d_0^{(3)}) - u_0^{(3)}) \quad \text{(a)}$$

$$c_1 = \mathcal{C}om(r_1^{(0)}, \mathrm{v} - \mathcal{P}(r_1^{(0)}) - \mathcal{G}(t_1^{(0)}, r_1^{(0)}) - e_1^{(0)}) = \mathcal{C}om(r_1^{(1)}, \mathcal{G}(t_0^{(1)}, r_1^{(1)}) + e_0^{(1)}) \quad \text{(b)}$$

$$c_2 = \mathcal{C}om(t_0^{(1)}, e_0^{(1)}) = \mathcal{C}om(t_0^{(2)}, e_0^{(2)}) = \mathcal{C}om(r_0^{(3)} - t_1^{(3)}, \mathcal{P}(r_0^{(3)}) - e_1^{(3)}) \quad \text{(c)}$$

$$c_3 = \mathcal{C}om(t_1^{(0)}, e_1^{(0)}) = \mathcal{C}om(r_0^{(2)} - t_0^{(2)}, \mathcal{P}(r_0^{(2)}) - e_0^{(2)}) = \mathcal{C}om(t_1^{(3)}, e_1^{(3)}) \quad \text{(d)}$$

$$c_4 = \mathcal{C}om(d_0^{(0)}, u_0^{(0)}) = \mathcal{C}om(r_1^{(1)} - d_1^{(1)}, \mathcal{P}(r_1^{(1)}) - u_1^{(1)}) = \mathcal{C}om(d_0^{(3)}, u_0^{(3)}) \quad \text{(e)}$$

$$c_5 = \mathcal{C}om(r_1^{(0)} - d_0^{(0)}, \mathcal{P}(r_1^{(0)}) - u_0^{(0)}) = \mathcal{C}om(d_1^{(1)}, u_1^{(1)}) = \mathcal{C}om(d_1^{(2)}, u_1^{(2)}) \quad \text{(f)}$$

Das equações (a), (b), (c), (d), (e) e (f) resultam as igualdades a seguir ou então $\mathcal{C}om$ não possui vinculação, visto que teria gerado valores iguais de compromisso a partir de valores diferentes de msg. Desta forma temos: $r_1^{(0)} = r_1^{(1)}$, $r_0^{(2)} = r_0^{(3)}$, $t_0^{(1)} = t_0^{(2)} = r_0^{(3)} - t_1^{(3)}$, $t_1^{(0)} = r_0^{(2)} - t_0^{(2)} = t_1^{(3)}$, $e_0^{(1)} = e_0^{(2)} = \mathcal{P}(r_0^{(3)}) - e_1^{(3)}$, $e_1^{(0)} = \mathcal{P}(r_0^{(2)}) - e_0^{(2)} = e_1^{(3)}$, $d_0^{(0)} = r_1^{(1)} - d_1^{(1)} = d_0^{(3)}$, $r_0^{(0)} - d_0^{(0)} = d_1^{(1)} = d_1^{(2)}$, $u_0^{(0)} = \mathcal{P}(r_1^{(1)}) - u_1^{(1)} = u_0^{(3)}$, $\mathcal{P}(r_1^{(0)}) - u_0^{(0)} = u_1^{(1)} = u_1^{(2)}$.

A partir de (a) e (b) nós temos que $\mathrm{v} = \mathcal{G}(r_0^{(2)}, d_1^{(2)}) + u_1^{(2)} + \mathcal{P}(r_0^{(3)}) + \mathcal{G}(r_0^{(3)}, d_0^{(3)}) + u_0^{(3)} = \mathcal{G}(t_0^{(1)}, r_1^{(1)}) + e_0^{(1)} + \mathcal{P}(r_1^{(0)}) + \mathcal{G}(t_1^{(0)}, r_1^{(0)}) + e_1^{(0)})$.

Das equações e igualdades citadas acima podemos verificar que:

$$t_0^{(1)} + t_1^{(0)} = r_0^{(3)} - t_1^{(3)} + t_1^{(0)} = r_0^{(3)}$$
$$e_0^{(1)} + e_1^{(0)} = \mathcal{P}(r_0^{(3)}) - e_1^{(3)} + e_1^{(0)} = \mathcal{P}(r_0^{(3)})$$
$$d_0^{(3)} + d_1^{(2)} = r_1^{(1)} - d_1^{(1)} + d_1^{(2)} = r_1^{(1)}$$
$$u_0^{(3)} + u_1^{(2)} = \mathcal{P}(r_1^{(1)}) - u_1^{(1)} + u_1^{(2)} = \mathcal{P}(r_1^{(1)})$$
$$\mathcal{G}(r_0^{(2)}, d_1^{(2)}) + \mathcal{G}(r_0^{(3)}, d_0^{(3)}) = \mathcal{G}(r_0^{(3)}, r_1^{(1)})$$
$$\mathcal{G}(t_0^{(1)}, r_1^{(1)}) + \mathcal{G}(t_1^{(0)}, r_1^{(0)}) = \mathcal{G}(r_0^{(3)}, r_1^{(0)})$$

 ©2012 SBC — Soc. Bras. de Computação

Chegamos então a v $= \mathcal{G}(r_0^{(3)}, r_1^{(0)}) + \mathcal{P}(r_0^{(3)}) + \mathcal{P}(r_1^{(0)}) = \mathcal{P}(r_0^{(3)} + r_1^{(0)})$, confirmando assim que poderíamos recuperar a chave secreta s com $r_0^{(3)} + r_1^{(0)}$.

### 5.2. Comparação com o original

Em [Sakumoto et al. 2011], Sakumoto, Shirai e Hiwatari incluíram uma comparação da eficiência de $\mathcal{MQID}$-3 com outros protocolos de identificação em 3 passos baseados nos problemas SD binário, CLE e PP. Não vimos razão para repetir essa comparação aqui, uma vez que nosso foco, como já dito, é a diminuição da comunicação total do protocolo de Sakumoto *et al*. Por isso, vamos restringir esta análise comparativa ao tamanho da comunicação do nosso protocolo aprimorado ante ao originalmente apresentado.

Em comparação ao protocolo $\mathcal{MQID}$-3 original, vamos demonstrar que nossa versão aprimorada alcança uma comunicação 9,5% menor para os parâmetros sugeridos por Sakumoto *et al*, apesar de aumentar o tráfego em uma rodada. Este ganho é resultado da diminuição do número de rodadas para o mesmo nível de segurança, visto que conseguimos uma probabilidade de personificação em uma rodada de 1/2, contra 2/3 do protocolo $\mathcal{MQID}$-3.

Como já vimos na revisão do protocolo original, o $\mathcal{MQID}$-3 precisa de $\lceil \frac{\lambda}{log_2 3 - 1} \rceil$ repetições para alcançar uma probabilidade total de personificação menor que $2^{-\lambda}$, porém nosso protocolo aprimorado, por ter uma probabilidade 1/2 em uma rodada, precisa de apenas $\lambda$ rodadas. Vemos assim que o protocolo original necessita aproximadamente 71% mais rodadas para o mesmo nível de segurança.

No protocolo original, lembrando que obrigatoriamente estamos sempre considerando a utilização da extensão de [Stern 2006], conforme já descrito anteriormente neste texto e também explicitado no último paragrafo do item 3 de [Sakumoto et al. 2011], temos que o provador gera 3 compromissos e envia um hash $hc$ da concatenação dos mesmos para o verificador, que responde com um desafio $Ch$ do qual decorre o envio de $Rsp$ pelo provador. O hash dos compromissos ($hc$) possui o tamanho de $2m$ bits, $Ch$ ocupa 2 bits e $Rsp$ é igual a $2n + 3m$ bits, totalizando assim $2n + 5m + 2$ bits trafegados em uma rodada. Nosso protocolo gera 6 compromissos inicialmente, efetuando o envio de um hash $hc$ da mesma forma já citada, um desafio de mesmo tamanho e nosso $Rsp$ possui $3n + 6m$ bits, perfazendo um total de $3n + 8m + 2$ bits. Considerando agora a comunicação total para uma execução com probabilidade total de personificação menor que $2^{-\lambda}$, chegamos ao total de $\lambda \cdot (3n + 8m + 2)$ para o nosso protocolo aprimorado, enquanto o original totaliza $1,71\lambda \cdot (2n + 5m + 2) = \lambda \cdot (3,42n + 8,55m + 3,42)$. Considerando n=m temos nossa proposta alcançando um tráfego de dados aproximadamente 8% menor. Utilizando os parâmetros propostos pelos autores originais, $n = 84$ e $m = 80$, $\mathcal{MQID}$-3 dispende 570 bits por rodada, enquanto nosso protocolo utiliza 894, porém, para uma segurança igual a $2^{-30}$ o protocolo original utiliza 52 rodadas, enquanto o nosso necessita apenas de 30. Assim, teríamos uma comunicação total de 26820 bits com nosso protocolo aprimorado contra 29640 bits de $\mathcal{MQID}$-3, ou seja, uma redução de 9,5%.

## 6. Considerações Finais

Neste artigo, apresentamos um novo protocolo de identificação no modelo de chave pública baseado no problema $\mathcal{MQ}$, que é um aprimoramento do protocolo $\mathcal{MQID}$-3 descrito em [Sakumoto et al. 2011]. A proposta é fundamentada no emprego da forma polar da

 ©2012 SBC — Soc. Bras. de Computação

função $\mathcal{MQ}$ sugerida por esses autores, porém adota uma divisão diferente da chave secreta. Como resultado, obtivemos um protocolo que apresenta probabilidade 1/2 para personificação em uma rodada, contra 2/3 do original, diminuindo o número total de rodadas necessárias para o mesmo nível de segurança e alcançando uma redução de 9,5% da comunicação necessária em uma execução, considerando os parâmetros de segurança propostos originalmente.

## Agradecimento

Gostaríamos de registrar um agradecimento especial ao professor Paulo S. L. M. Barreto, da Escola Politécnica (USP), pelas valiosas sugestões que muito contribuíram para o aprimoramento deste trabalho.

## Referências

[Abdalla et al. 2002] Abdalla, M., An, J. H., Bellare, M. e Namprempre, C. (2002). From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 418–433, London, UK. Springer-Verlag.

[Alaoui et al. 2012] Alaoui, S. M. E. Y., Dagdelen, O., Véron, P., Galindo, D. e Cayrel, P.-L. (2012). Extended security arguments for signature schemes. In *Proceedings of the 5th international conference on Cryptology in Africa*, AFRICACRYPT'12, pages 19–34, Berlin, Heidelberg. Springer-Verlag.

[Bellare e Rogaway 1993] Bellare, M. e Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, CCS '93, pages 62–73, New York, NY, USA. ACM.

[Bernstein et al. 2009] Bernstein, D. J., Buchmann, J. e Dahmen, E., editors (2009). *Post-Quantum Cryptography*. Springer.

[Bouillaguet et al. 2011] Bouillaguet, C., Faugère, J.-C., Fouque, P.-A. e Perret, L. (2011). Practical cryptanalysis of the identification scheme based on the isomorphism of polynomial with one secret problem. In *Proceedings of the 14th international conference on Practice and theory in public key cryptography conference on Public key cryptography*, PKC'11, pages 473–493, Berlin, Heidelberg. Springer-Verlag.

[Courtois et al. 2003] Courtois, N. T., Goubin, L. e Patarin, J. (2003). SFLASHv3, a fast asymmetric signature scheme. *IACR Cryptology ePrint Archive*, 2003:211.

[Ding et al. 2006] Ding, J., Gower, J. E. e Schmidt, D. (2006). *Multivariate Public Key Cryptosystems*, volume 25 of *Advances in Information Security*. Springer.

[Dubois et al. 2007] Dubois, V., Fouque, P.-A., Shamir, A. e Stern, J. (2007). Practical cryptanalysis of SFLASH. In *Proceedings of Advances in Cryptology–CRYPTO '07*. Springer-Verlag.

[Fiat e Shamir 1987] Fiat, A. e Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings of Advances in Cryptology–CRYPTO '86*, pages 186–194, London, UK, UK. Springer-Verlag.

 ©2012 SBC — Soc. Bras. de Computação

[Garey e Johnson 1979] Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[Halevi e Micali 1996] Halevi, S. e Micali, S. (1996). Practical and provably-secure commitment schemes from collision-free hashing. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 201–215, London, UK, UK. Springer-Verlag.

[Matsumoto e Imai 1988] Matsumoto, T. e Imai, H. (1988). Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Proceedings of Advances in Cryptology-EUROCRYPT '88*, pages 419–453. Springer-Verlag New York, Inc.

[Patarin et al. 2001] Patarin, J., Courtois, N. e Goubin, L. (2001). Flash, a fast multivariate signature algorithm. In *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, CT-RSA 2001, pages 298–307, London, UK, UK. Springer-Verlag.

[Patarin e Goubin 1997] Patarin, J. e Goubin, L. (1997). Trapdoor one-way permutations and multivariate polynomials. In Han, Y., Okamoto, T. e Qing, S., editors, *Information and Communications Security*, volume 1334 of *Lecture Notes in Computer Science*, pages 356–368. Springer Berlin / Heidelberg.

[Petzoldt et al. 2011] Petzoldt, A., Thomae, E., Bulygin, S. e Wolf, C. (2011). Small public keys and fast verification for multivariate quadratic public key systems. In *Proceedings of the 13th international conference on Cryptographic hardware and embedded systems*, CHES'11, pages 475–490, Berlin, Heidelberg. Springer-Verlag.

[Pointcheval e Stern 1996] Pointcheval, D. e Stern, J. (1996). Security proofs for signature schemes. In *Proceedings of the 15th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'96, pages 387–398, Berlin, Heidelberg. Springer-Verlag.

[Pointcheval e Stern 2000] Pointcheval, D. e Stern, J. (2000). Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396.

[Sakumoto et al. 2011] Sakumoto, K., Shirai, T. e Hiwatari, H. (2011). Public-key identification schemes based on multivariate quadratic polynomials. In Rogaway, P., editor, *Advances in Cryptology CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 706–723. Springer Berlin / Heidelberg.

[Shor 1997] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509.

[Stern 2006] Stern, J. (2006). A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768.

[Wolf 2005] Wolf, C. (2005). *Multivariate Quadratic Polynomials in Public Key Cryptography*. PhD thesis, Katholieke Universiteit Leuven.

[Wolf e Preneel 2005] Wolf, C. e Preneel, B. (2005). Taxonomy of public key schemes based on the problem of multivariate quadratic equations. *IACR Cryptology ePrint Archive*, 2005:77.

 ©2012 SBC — Soc. Bras. de Computação

# $\chi^2$ Attacks on Block-Cipher based Compression Functions

**Daniel Santana de Freitas**[1], **Jorge Nakahara Jr**

[1]Depto de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC - Brazil

santana@inf.ufsc.br, jorge_nakahara@yahoo.com.br

***Abstract.*** *In this paper, we report on $\chi^2$ analyses of block-cipher based (cryptographic) compression functions. Our aim is not to find collisions nor (second) preimages, but to detect non-random properties that may distinguish a compression function from an ideal primitive such as a random oracle. We study some well-known single-block modes of operation such as Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP), and double-block modes such as Hirose's, Tandem-DM, Abreast-DM, Parallel-DM and MDC-2. This paper shows how a weakness ($\chi^2$ correlation) in the underlying block cipher can propagate to the compression function via the mode of operation used in hash constructions. To demonstrate our ideas, we instantiated the block cipher underlying these modes with variable-round RC5, RC6 and ERC6 block ciphers.*

Keywords: $\chi^2$ cryptanalysis, block-cipher-based (cryptographic) compression functions, single- and double-block-length modes of operation.

## 1. Introduction

Hash and compression functions are pervasive cryptographic primitives used for privacy and authentication purposes in environments as diverse as computer networks, sensor networks and mobile devices [Kaufman et al. 2002, Stallings 2003]. There are many security properties that compression and hash functions have to satisfy depending on the application environment. In this paper, we apply $\chi^2$ analysis techniques to block-cipher based compression and hash functions in order to detect some nonrandom behavior that demonstrates that they are not ideal primitives. Our aim is not to find collisions nor (second) preimages [Menezes et al. 1997], but statistical correlations between the input and the output of the compression or hash functions. We recall that NIST requested in [NIST 2007b] that the SHA-3 candidates should behave as close as possible to random oracles [NIST 2007a], which is as expected of an ideal cryptographic primitive.

We selected the following block ciphers to instantiate the permutation mapping(s), denoted $E$, inside the compression functions under analysis (Fig. 1):

- the RC6 block cipher [Rivest et al. 1998] has a Feistel Network structure and was a finalist candidate to the AES competition [NIST 2001] aimed at selecting the successor of the DES [NIST 1993] cipher. RC6 actually stands for a family of ciphers, and is more appropriately designated as RC6-$w/r/b$, where $w$ stands for the word size in bits (RC6 is a word-oriented cipher in the sense that it operates on neatly partitioned $w$-bit words), $r$ is the number of rounds and $b$ is the key length in bytes. For the AES competition, $w = 32$ (a text block consists of four $w$-bit

 ©2012 SBC — Soc. Bras. de Computação

words), $r = 20$ and three key sizes: 128 bits ($b = 16$), 192 bits ($b = 24$) and 256 bits ($b = 32$). This version is called simply RC6.

- the RC5 block cipher [Rivest 1994] has a Feistel Network structure, and is the predecessor of RC6. More formally, RC5 stands for a family of ciphers, and is designated RC5-$w/r/b$, with parameters $w$, $r$ and $b$ having the same meaning as in RC6 and value ranges: $w \in \{16, 32, 64\}$, $0 \leq k \leq 255$, $0 \leq r \leq 255$. A text block in RC5 consists of two $w$-bit words. The original parameter values are $w = 32$ (leading to a 64-bit block), $r = 12$ (nowadays, 20 rounds are suggested) and $b = 16$ (or 128-bit key).
- the ERC6 block cipher [Ragab et al. 2001] has a Feistel Network structure and is a large-block generalization of RC6, thus the name Extended RC6. Formally, ERC6 stands for a family of ciphers, and is parameterized as ERC6-$w/r/b$, with the same meaning as for RC6. A text block in ERC6 consists of eight $w$-bit words. Nominal values of these parameters are not available, but in [Ragab et al. 2001] one can find the values $w = 32$, $r = b = 16$ used for performance evaluation in software.

This choice of block ciphers leads to hash digests (or chaining variables) of 64, 128, 256 and 512 bits for the single- and double-block length hash modes. Our attacks do not depend on related-keys nor on the key schedule algorithms. Further details about it, for each key size, can be found in [Rivest et al. 1998, Rivest 1994, Ragab et al. 2001].

Our choice of RC5/RC6/ERC6 was motivated by well-known results of $\chi^2$ attacks on reduced-round versions of each cipher [Isogai et al. 2002, Knudsen 2002, Jr. et al. 2009]. These results indicate that not only a large number of rounds can be distinguished from a random permutation without exhausting the codebook, but also that the $\chi^2$ correlation patterns are *iterative*, which is a fundamental feature for leveraging the attack to a large number of rounds when these ciphers are used as building blocks in compression and hash function constructions.

This paper is organized as follows: Sect. 2 summarizes the contributions of this paper. Sect. 3 briefly explains a $\chi^2$ analysis. Sect. 4 describes attacks on the compression function in DM mode. Sect. 5 describes attacks on compression functions in MMO and MP modes. Sect. 6 presents an attack on a double-block-length mode of operation called Tandem-DM. Sect. 7 presents an attack on the Abreast-DM mode. Sect. 8 presents an attack on the Parallel-DM mode. Sect. 9 presents an attack on the MDC-2 mode. Sect. 10 concludes this paper.

## 2. Contributions

The contributions of this paper include

- a concrete application of $\chi^2$ technique [Knudsen and Meier 2000] to compression functions; in particular we aim at block-cipher based hash function constructions.
- to demonstrate our attacks, we instantiated the block cipher(s) inside the hashing modes of operation with variable-round versions of RC5 [Rivest 1994], RC6 [Rivest et al. 1998] and ERC6 [Ragab et al. 2001], for which there are well-known $\chi^2$ correlations for a large number of rounds. We require the target block ciphers to be susceptible to $\chi^2$ (distinguishing) attacks, so that we can extend this correlation further to the compression function, via the mode of operation, and eventually

 ©2012 SBC — Soc. Bras. de Computação

to the hash function. Note that the fact that up to 44-round ERC6 (a large number of rounds) is susceptible to $\chi^2$ attacks already indicates that these primitives are not appropriate building blocks in compression function constructions.

- further, *we exploit the fact that the correlations patterns are iterative*, that is, they have the same form both at the input and at the output of the block cipher, so that we can bypass the feedforward and feedback of chaining variables commonly found in modes of operation, and thus extend the distinguishing capability of $\chi^2$ technique beyond the block cipher framework.
- we perform only *distinguishing attacks* since there are no secret keys in compression/hash function settings. On one hand, this fact implies that the attack complexities are lower since no keys are recovered/guessed. On the other hand, the fact that the key is not secret means that it is under the control of the adversary. So, in our attacks, we can freely choose the key to be weak concerning the $\chi^2$ attacks, and thus, cover a large number of rounds of the underlying block ciphers. Even though $\chi^2$ attacks can be performed under a known-plaintext setting [Shimoyama et al. 2001, Miyaji et al. 2002], we have to choose some message blocks and chaining variables to satisfy the attack requirements. Thus, all our attacks operate under a chosen-message-or-chaining-variable (CMCV) setting.
- we analyse single-block modes of operation such as Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP) [Menezes et al. 1997]. This is a subset of a large series of modes of operation originally described by Preneel *et al.* in [Preneel et al. 1994]. We also analysed double-block modes such as Hirose's [Hirose 2006], Tandem-DM, Abreast-DM [Lai and Massey 1993], Parallel-DM and MDC-2 [Brachtl et al. 1990], [Meyer and Schilling 1988].
- our attacks use the $\chi^2$ technique to demonstrate non-random properties of the underlying block cipher. We describe how a particular weakness, a $\chi^2$ correlation, can propagate from the block cipher to the compression function via the mode of operation. Therefore, we *do not* aim at traditional attacks such as collisions or (second) preimage, but our findings are relevant in applications where hash functions are expected to behave as random oracles such as (pseudo-)random number generators, which is required by NIST for the ongoing SHA-3 competition [NIST 2007b, NIST 2007a]. While most of the traditional analysis of hash functions use differential cryptanalysis (aimed at finding collisions), our approach uses the $\chi^2$ technique for the analysis of compression functions in order to uncover weaknesses and non-random behavior which prove that these functions are not ideal cryptographic primitives.

## 3. A Brief Overview of $\chi^2$ Correlations

We recall from [Knudsen and Meier 2000] a brief description of the $\chi^2$ goodness-of-fit test to distinguish a random source with unknown probability distribution $p_X$ from a random source with uniform distribution $p_U$. This $\chi^2$ test will be used to detect correlations between specific input and output bits of $r$-round ciphers. Let $X = x_0, x_1, \ldots, x_{n-1}$ be identically and independently distributed random variables taking values in the set $\{a_0, a_1, \ldots, a_{m-1}\}$ with unkown probability distribution. The $\chi^2$ test is used to decide if an observation $x_0, x_1, \ldots, x_{n-1}$ is consistent with the hypothesis $\Pr(x = a_j) = p(j)$ for $0 \leq j < m$, where $p_X = \{p(j)\}$ is a discrete probability distribution on a set of $m$

 ©2012 SBC — Soc. Bras. de Computação

elements. Let $N_{a_j}(X)$ denote the number of times the observation $X$ takes on the value $a_j$. Then, $\sum_j N_{a_j}(X) = n$. The $\chi^2$ statistic is the random variable defined by

$$\chi^2 = \sum_{j=0}^{m-1} \frac{(N_{a_j}(X) - np(j))^2}{np(j)}.$$

For the uniform distribution $p_U$, the $\chi^2$ statistic is just $\frac{m}{n}\sum_j (N_{a_j}(X) - \frac{n}{m})^2$. In a $\chi^2$ test, the observed $\chi^2$ statistic is compared to $\chi^2_{a,m-1}$, the threshold for the $\chi^2$ test with $m-1$ degrees of freedom and with significance level $a$.

In this paper, we exploit $\chi^2$ correlations as a distinguishing tool to detect non-random behavior of block ciphers inside a compression function in modes of operation such as Matyas-Meyer-Oseas (MMO), Davies-Meyer (DM) and Miyaguchi-Preneel (MP) [Menezes et al. 1997]. Further, we look to extend such correlations to the full mode of operation, and eventually to the hash function as well. Since there is no key involved in compression or hash functions, all attacks are of the distinguish-from-random type.

In [Knudsen and Meier 2000], Knudsen and Meier showed $\chi^2$ correlations can be used to distinguish 15-round RC6 from a random permutation (and up to 17 rounds for weak keys). These correlations were derived from 2-round iterative linear relations called Type-I approximations, having the form

$$(A \cdot e_t) \oplus (C \cdot e_s) = (A' \cdot e_u) \oplus (C' \cdot e_v), \tag{1}$$

where $A$ and $C$ are the first and third $w$-bit input words, that is, a plaintext block has the form $A\|B\|C\|D$, while $A'$ and $C'$ are the first and third $w$-bit words two rounds later, that is, the cipher state after two rounds has the form $A'\|B'\|C'\|D'$; $e_t$ denotes a $w$-bit mask with a single bit $1$ in the $t$-th least significant bit position. If $t$, $s$, $u$ and $v$ are nonzero but less than five, there is a nonzero bias which depends on the values of $t, s, u, v$. The idea is to fix each of the $\text{lsb}_5(A)$ and $\text{lsb}_5(C)$ and collect statistics of the ten bits: $\text{lsb}_5(A')$ and $\text{lsb}_5(C')$. That means an iterative distinguisher that can be concatenated every two rounds. Note that the $\chi^2$ analysis sets the least significant $\log_2 w$ bits of $A$ and $C$ to zero. This is a consequent of previous analysis of RC5 in [Borst et al. 1999].

RC6 is well-known for the extensive use of data-dependent rotations. It was mentioned in [Knudsen and Meier 2000] that for some weak keys that generate zero rotation amounts, $\chi^2$ correlations could be detected up to 16-round RC6. Since we control the key input, which means the message blocks $m_i$ or chaining variables $H_{i-1}$ and $H_i$, we can afford to choose them so that the key is weak. Also for this same reason, we can include the pre-whitening layer with subkeys. But, we omit the post-whitening layer since our distinguishers do not cover the full nominal number of rounds. Therefore, following [Knudsen and Meier 2000], we assume that with about $2^{118}$ chosen plaintexts, we can detect $\chi^2$ correlation across 16-round RC6 under weak keys.

In [Miyaji et al. 2002], Miyaji *et al.* described improvements to $\chi^2$ attacks applied to RC5-32 ($w = 32$), for up to 10 rounds. The analysis approach is similar to equation (1). Let a plaintext block be denoted $A\|B$ and the output after $h$ rounds be $A'\|B'$. The attack setting in reduced-round RC5 starts by setting $\text{lsb}_5(B)$ to zero and computing $\chi^2$ correlation in $\text{lsb}_5(B')$. The idea is similar to (1). For an even number of rounds:

$$B \cdot e_t = B' \cdot e_u, \tag{2}$$

 ©2012 SBC — Soc. Bras. de Computação

where $e_t$ denotes a $w$-bit mask with a single bit $1$ in the $t$-th least significant bit position. The best $\chi^2$ distinguishing attack reported in [Miyaji et al. 2002] reaches 10 rounds and requires $2^{57.87}$ chosen plaintexts and equivalent 10-round computations.

In [Shimoyama et al. 2001], Shimoyama *et al.* described $\chi^2$ analysis of RC5-32/10 using $2^{57.87}$ chosen ciphertexts and equivalent number of 10-round computations. This attack has high success rate and uses a similar approach to (2). For weak keys that lead to small or zero rotation values, $\chi^2$ correlations can be detected further up to 12 rounds.

In [Isogai et al. 2002], Isogai *et al.* described attacks on RC5-64, ie. with $w = 64$ up to 16 rounds. This is an indication that increasing the word size helps improve the attack to more rounds since $\chi^2$ correlations can be detected over a larger set of $\log_2 w$ bits of some input and output words of $r$-round RC5-64. Actually, as demonstrated by attacks on RC5, RC6, ERC6 and further on MRC6 [El-Fishawy et al. 2004], in this order, it seems that increasing the number of words in a text block also helps improve the $\chi^2$ attacks to a larger number of rounds.

In [Jr. et al. 2009], Nakahara *et al.* described $\chi^2$ distinguishing attacks on up to 44-round ERC6 using $2^{233}$ chosen plaintexts. There is no mention of attack improvements due to weak keys/subkeys in [Jr. et al. 2009]. So, we apply our analysis assuming the key is fixed but arbitrary. The correlation pattern for ERC6 has a similar form to (1) but a text block has eight words, denoted $A\|B\|C\|D\|E\|F\|G\|H$. For an even number of rounds, and following a Type-I approximation, the correlation pattern for ERC6 is

$$A \cdot e_{t_1} \oplus C \cdot e_{t_2} \oplus E \cdot e_{t_3} \oplus G \cdot e_{t_4} = A' \cdot e_{t_5} \oplus C' \cdot e_{t_6} \oplus E' \cdot e_{t_7} \oplus G' \cdot e_{t_8}. \tag{3}$$

There are many other ciphers that could be attacked, but they are all variants of RC5/RC6, weakened or extended versions such as RC6-64 [Knudsen 2002] and MRC6 in [El-Fishawy et al. 2004]. We limit our analyses to these 3 ciphers: RC5, RC6 and ERC6, since they provide enough freedom of choice of block and key sizes.

## 4. Attacks on the DM mode

The DM mode is depicted in Fig. 1(b). The $i$-th chaining value is computed as

$$H_i = H_{i-1} \oplus E_{m_i}(H_{i-1}), \tag{4}$$

where $H_0 = IV$ is the initial value, $m_i$ is the $i$-th message block and $E_x(y)$ is a block cipher with key $x$ and plaintext $y$. In all settings, we assume the key size to $E$ can be adapted to fit the message or chaining variable length.

Let us consider RC6-32/16/16 as $E$ for an initial attack on a compression function in DM mode since equation (1) is iterative, that is, the very same words are used both at the input and the output. In particular, let $m_i, H_{i-1} \in \mathbb{Z}_2^{128}$ be the input and $H_i \in \mathbb{Z}_2^{128}$ be the output of a compression function in DM mode. The attack proceeds as follows: we fix $m_i$ appropriately as a weak key so that we our attack reaches 16 rounds of $E$. We choose $H_{i-1}$ as input to $E$ according to (1), that is, *with the five least significant bits of its first and third words set to zero* so that a $\chi^2$ correlation can be detected in the same words at $E_{m_i}(H_{i-1})$.

 ©2012 SBC — Soc. Bras. de Computação

Despite the feedforward of $H_{i-1}$ in DM mode in (4), this correlation can be *extended* to $H_i$ because of the way $H_{i-1}$ was constructed, that is, the zero bits in $H_{i-1}$ do not affect the correlation in $H_i$ and the correlation pattern appears at the same place in both $H_i$ and $E_{m_i}(H_{i-1})$. Following [Knudsen and Meier 2000], by appropriately choosing $H_{i-1}$ and keeping $m_i$ fixed, we can detect a biased distribution in $H_i$, and thus, distinguish a compression function in DM mode instantiated by RC6-32/16/16 from a random mapping using $2^{118}$ chosen chaining variable values and equivalent effort. Memory is negligible since we do not need to store the messages, but only the $\chi^2$ statistics. Note that even if the feedforward of $H_{i-1}$ was combined via modular addition instead of xor, the attack would still hold because the least significant five bits of the 1st and 3rd words of $H_{i-1}$ were purposefully set to zero.

Suppose we set $i = 2$ in the previous paragraph. If we vary $m_1$ over all $2^{128}$ possible text blocks, with $H_0$ fixed, then $H_1 = E_{m_1}(H_0) \oplus H_0$ would behave as a random function. According to [Rijmen et al. 1997], varying $m_1$ this way would cover about $\frac{1}{e} \cdot 2^{128} = 2^{126.56}$ of the codebook. Suppose now that we keep $m_2$ fixed as key input to the next compression function instance, then $H_2 = E_{m_2}(H_1) \oplus H_1$. If $m_2$ is the last message block then it should contain padding and the total message length (following MD strengthening). But, we cannot control $H_1$ to have zero bits in the appropriate positions for the $\chi^2$ attack on the underlying block cipher. Thus, our attack is restricted to the compression function only.

Now, suppose we instantiate $E$ with RC5-32/10/8 in DM mode. We apply the iterative pattern in (2). This time, $m_i, H_{i-1} \in \mathbb{Z}_2^{64}$ are the input and $H_i \in \mathbb{Z}_2^{64}$ is the output of a compression function in DM mode. The attack proceeds as follows: we fix $m_i$ as a key so that $E$ behaves as a permutation. We choose $H_{i-1}$ as input to $E$ according to (2), that is, with the five least significant bits of its second word only set to zero so that a $\chi^2$ correlation can be detected in the same words at $E_{m_i}(H_{i-1})$. Despite the feedforward of $H_{i-1}$ in (4), this correlation can be leveraged to $H_i$ because of the way $H_{i-1}$ was constructed, that is, the zero bits in $H_{i-1}$ do not affect the correlation in $H_i$ and the correlation pattern appears at the same place in both $H_i$ and $E_{m_i}(H_{i-1})$. Following [Miyaji et al. 2002], by appropriately choosing $H_{i-1}$ and keeping $m_i$ fixed, we can detect a biased distribution in $H_i$, and thus, distinguish a compression function in DM mode instantiated by RC5-32/10/8 from a random mapping using $2^{57.87}$ chosen messages and equivalent effort.

Repeating the same reasoning for ERC6-32/44/32 in DM mode, and applying the pattern (4). This time $m_i, H_{i-1} \in \mathbb{Z}_2^{256}$ are the input and $H_i \in \mathbb{Z}_2^{256}$ are the output of the compression function. The attack proceeds as follows: we fix $m_i$ as key so that $E$ behaves as a permutation. We choose $H_{i-1}$ as input to $E$ so that the five least significant bits of four words, according to (4), are set to zero so that a $\chi^2$ correlation can be detected in the same words at $E_{m_i}(H_{i-1})$, despite the feedforward of $H_{i-1}$ in (4), as explained in the previous paragraphs. Following [Jr. et al. 2009], by appropriately choosing $H_{i-1}$ and keeping $m_i$ fixed, we can detect a biased distribution in $H_i$, and thus, distinguish a compression function in DM mode instantiated by ERC6-32/44/32 from a random mapping using $2^{233}$ chosen messages and equivalent effort.

A reasoning why so far we could not extend the attack to a hash function in DM mode is the following: suppose a message $M = m_1 \| m_2$ where $m_1$ is variable while $H_0$

 ©2012 SBC — Soc. Bras. de Computação

is fixed. It means that $H_1 = H_0 \oplus E_{m_1}(H_0)$ behaves as a random function. According to [Rijmen et al. 1997], if $m_1$ runs over all $2^{128}$ values, then about $\frac{1}{e} \cdot 2^{128} \approx 2^{126.56}$ distinct values appear in $H_1$. We keep $m_2$ fixed since it might contain (eventual) padding and the length of $M$, due to the Merkle-Damgaard (MD) strengthening. Thus, $E_{m_2}(H_1)$ behaves as a permutation. If we could filter out only those values of $H_1$ which have zero bits in the five least significant positions of its first and third words, then we could detect a $\chi^2$ correlation in $H_2 = H_1 \oplus E_{m_2}(H_1)$, which would be the hash digest. Out of the $2^{126.56}$ inputs, 10 bits would have to be fixed. That means only $2^{116.56}$ would be available, which is not enough. We need $2^{118}$ texts.

## 5. Distinguishing attacks for the MMO and MP modes

In MMO mode, the $i$-th chaining variable is computed as

$$H_i = m_i \oplus E_{H_{i-1}}(m_i), \tag{5}$$

with $H_0 = IV$, as depicted in Fig. 1(a). Let us consider RC6-32/16/16 as $E$ in MMO mode. Let $m_i, H_{i-1} \in \mathbb{Z}_2^{128}$ be the input and $H_i \in \mathbb{Z}_2^{128}$ be the output of the compression function. The attack proceeds as follows: we choose a fixed (weak) value for $H_{i-1}$ so that it is a (weak) key to $E$. We choose $m_i$ as input to $E$ according to (1), that is, with the five least significant bits of its first and third words set to zero so that a $\chi^2$ correlation can be detected in the same words at $E_{H_{i-1}}(m_i)$. Due to the feedforward of $m_i$, this $\chi^2$ correlation can be extended to $H_i$ because of (5). Following [Knudsen and Meier 2000], by appropriately choosing $m_i$ and keeping $H_{i-1}$ fixed, we can detect a biased distribution in $H_i$, and thus, distinguish a compression function MMO mode instantiated by RC6-32/16/16 from a random mapping using $2^{118}$ chosen messages and equivalent effort. Memory is negligible since we do not need to store the messages, but only the $\chi^2$ statistics.

Now, suppose we instantiate $E$ with RC5-32/10/8 in MMO mode. The attack would proceed similarly to the one for RC6-32/16/16, but $m_i, H_{i-1}$, $H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

Analogously, if we instantiate $E$ with ERC6-32/44/32 in MMO mode. Once again, the attack would proceed similarly to the one for RC6, but $m_i, H_{i-1}$, $H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

In MP mode, the $i$-th chaining variable is computed as

$$H_i = m_i \oplus H_{i-1} \oplus E_{H_{i-1}}(m_i), \tag{6}$$

with $H_0 = IV$, as depicted in Fig. 1(c). We consider RC6-32/16/16 instantiating $E$ in MP mode. The attack proceeds analogous to the one in the previous paragraph. The MP and MMO modes are quite similar, except for a feedforward of $H_{i-1}$. We keep $H_{i-1}$ fixed as a (weak) key input. Thus, we can simple remove it from $H_i$ before checking the $\chi^2$ correlation, by just xoring $H_i$ with $H_{i-1}$ in (6). The attack complexity remains the same as in the MMO case.

Now, if we instantiate $E$ with RC5-32/10/8 in MP mode, the attack on the compression function would proceed similarly to the one for RC6-32/16/16, but $m_i, H_{i-1}$, $H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

 ©2012 SBC — Soc. Bras. de Computação

Analogously, we instantiate $E$ with ERC6-32/44/32 in MP mode. Once again, the attack on the compression function would proceed similarly to the one for RC6-32/16/16, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

## 6. Distinguishing attacks on the Tandem-DM mode

In [Lai and Massey 1993], Lai and Massey proposed a double-block length hash mode called Tandem-DM that uses two instances of an $n$-bit block, $2n$-bit key cipher $E$. A compression function instance is depicted in Fig. 1(e). The initial values are $H_0^1$ and $H_0^2$.

The attack proceeds as follows: we instantiate $E$ with RC6-32/16/32, message blocks $m_i \in \mathbb{Z}_2^{128}$ and chaining variables $H_i^1, H_i^2 \in \mathbb{Z}_2^{128}$. We keep $H_{i-1}^1$ and $m_i$ fixed as key input, so that $E_{H_{i-1}^1 \| m_i}(.)$ behaves as a permutation. We vary $H_{i-1}^2$ over $2^{118}$ distinct messages, with the least significant five bits of its first and third words set to zero. We shall detect $\chi^2$ correlation at $E_{H_{i-1}^1 \| m_i}(H_{i-1}^2)$ with high probability. Because of our choice of $H_{i-1}^2$, its feedforward will still allow to detect correlation at $H_i^2 = H_{i-1}^2 \oplus E_{H_{i-1}^1 \| m_i}(H_{i-1}^2)$.

Due to the feedback of $E_{H_{i-1}^1 \| m_i}(H_{i-1}^2)$ as part of the key input to the other $E$ instance, our attack is restricted to the compression function (and only to one $E$ instance). The attack complexity is $2^{118}$ chosen messages and equivalent computation effort.

Now, if we instantiate $E$ with RC5-32/10/16 (we have to adjust the key size to fit the double-sized key input) in Tandem-DM mode. The attack on the compression function proceeds similarly to the one in the previous paragraph, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

Analogously, if we instantiate $E$ with ERC6-32/44/64 in Tandem-DM mode. Once again, the attack on the compression function would proceed similarly to the previous ones, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

## 7. Distinguishing attacks on Abreast-DM mode

In [Lai and Massey 1993], Lai and Massey proposed a double-block length hash mode called Abreast-DM. A compression function instance is depicted in Fig. 1(f).

The attack proceeds as follows: we instantiate $E$ with RC6-32/16/32. We keep $H_{i-1}^1$ and $m_i$ fixed as key inputs, so that $E_{H_{i-1}^1 \| m_i}(.)$ behaves as a permutation. We vary $H_{i-1}^2$ over $2^{118}$ distinct messages, all of which have the pattern (1). We shall detect $\chi^2$ correlation at $E_{H_{i-1}^1 \| m_i}(H_{i-1}^2)$ with high probability. Because of our choice of $H_{i-1}^2$, its feedforward will still allow to detect correlation at $H_i^2$.

Due to the feedforward of $H_{i-1}^2$ as part of the key input to the other $E$ instance, our attack is restricted to the compression function (and only to one $E$ instance). In the original definition of Abreast-DM, the $E$ instance whose input is $H_{i-1}^1$ is negated (bitwise NOT). For our attacks, it does not matter since we do not depend on this $E$ instance. We use only the other $E$ instance[1]. Anyway, if necessary, we could compensate for this bitwise NOT by xoring with '1' bits the specific positions where we look for $\chi^2$

[1]Thus, as long as the $E$ instance we attack contains RC6-32/16/32, the other $E$ instance could be any block cipher whatsoever.

 ©2012 SBC — Soc. Bras. de Computação

correlation. We can afford to do it since we attack the compression function. Thus, for simplicity, we omit the bitwise NOT in Fig. 1(f) and in the attack description.

The final attack complexity is $2^{118}$ chosen messages and equivalent computational effort.

Now, if we instantiate $E$ with RC5-32/10/16 in Abreast-DM mode. The attack on the compression function would proceed similarly to the one in the previous paragraph, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

Analogously, if we instantiate $E$ with ERC6-32/44/64 in Abreast-DM mode. Once again, the attack on the compression function would proceed similarly to the previous ones, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

## 8. Distinguishing attacks on the Parallel-DM mode

The Parallel-DM is a double-block length hash mode of operation designed by Hohl *et al.* in [Hohl et al. 1993]. A compression function instance is depicted in Fig. 1(g).

Let us consider RC6-32/16/32 as $E$ in Parallel-DM mode. We attack the compression function only. Let $m_i^1, m_i^2 \in \mathbb{Z}_2^{128}$ be fixed values, as key inputs to the $E$ instance whose input is $H_{i-1}^1 \in \mathbb{Z}_2^{128}$. The attack proceeds as follows: we choose a weak value for $m_i^1 \oplus m_i^2$ so that it is a (weak) key to $E$, which behaves as a permutation. We choose $H_{i-1}^1 \oplus m_i^1$ as input to $E$ so that the least significant five bits of its first and third words are zero, while the remaining bits vary over $2^{118}$ values. This means that $H_{i-1}^1$ shall be chosen carefully to compensate for the xor with a fixed $m_i^1$ before being input to $E$. This way, a $\chi^2$ correlation can be detected in the same words at $E_{m_i^1 \oplus m_i^2}(m_i^1 \oplus H_{i-1}^1)$. Due to the feedforward of $H_{i-1}^1 \oplus m_i$, this $\chi^2$ correlation can be extended to $H_i^1$ because of the way $H_{i-1}^1$ was constructed. Following [Knudsen and Meier 2000], by appropriately choosing $m_i^1$, $m_i^2$ and $H_{i-1}^1$, we can detect a biased distribution in $H_i^1$, and thus, distinguish a compression function Parallel-DM mode instantiated by RC6-32/16/32 from a random mapping using $2^{118}$ chosen messages and equivalent effort. Memory is negligible since we do not need to store the messages, but only the $\chi^2$ statistics.

Now, if we instantiate $E$ with RC5-32/10/8 in Parallel-DM mode the attack on the compression function would proceed similarly to the one in the previous paragraph, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

Analogously, we instantiate $E$ with ERC6-32/44/32 in Parallel-DM mode. Once again, the attack on the compression function would proceed similarly to the previous ones, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

## 9. Distinguishing attacks on MDC-2 mode

The MDC-2 mode is a double-block length hash mode described by Bracthl *et al.* in [Brachtl et al. 1990] and in [Meyer and Schilling 1988]. It is depicted in Fig. 1(h). According to [Menezes et al. 1997], there is a fixed word swap after the feedforward of $m_i$,

 ©2012 SBC — Soc. Bras. de Computação

before the output is assigned to $H_i^1$ and $H_i^2$, but this swap does not affect our attack. Thus, we omit this swap to simplify the analysis.

Let us consider RC6-32/16/16 as $E$ in MDC-2 mode. We attack the compression function only. Let $H_{i-1}^1, H_{i-1}^2 \in \mathbb{Z}_2^{128}$ be fixed values, as key inputs to the $E$ instances, and $m_i \in \mathbb{Z}_2^{128}$. The attack proceeds as follows: we choose a weak value for $H_{i-1}^1$ and $H_{i-1}^2$ so that they are (weak) keys to the two $E$'s, that behave as permutations. We choose $m_i$ so that the least significant five bits of its first and third words are zero, while the remaining bits vary over $2^{118}$ values. This way, a $\chi^2$ correlation can be detected in the same words at $E_{H_{i-1}^1}(m_i)$ and at $E_{H_{i-1}^2}(m_i)$. Due to the feedforward of $m_i$ in both $E$ instances, this $\chi^2$ correlation can be detected in $H_i^1$ and $H_i^2$ as well, because of the way $m_i$ was constructed. Following [Knudsen and Meier 2000], we can detect a biased distribution in both $H_i^1$ and $H_i^2$ and thus, distinguish a compression function MDC-2 mode instantiated by RC6-32/16/16 from a random mapping using $2^{118}$ chosen messages and equivalent effort. Memory is negligible since we do not need to store the messages, but only the $\chi^2$ statistics.

Now, we instantiate $E$ with RC5-32/10/8 in MDC-2 mode. The attack on the compression function would proceed similarly to the one in the previous paragraph, but $m_i, H_{i-1}, H_i \in \mathbb{Z}_2^{64}$ and the complexity becomes $2^{57.87}$ chosen messages and equivalent effort.

Analogously, if we instantiate $E$ with ERC6-32/44/32 in MDC-2 mode, the attack on the compression function would proceed similarly to the previous ones, but $m_i, H_{i-1}$, $H_i \in \mathbb{Z}_2^{256}$ and the complexity would become $2^{233}$ chosen messages and equivalent effort.

## 10. Conclusions

This paper presented a $\chi^2$ analysis of block-cipher based compression functions. Our attacks included both single-block-length modes such as Matyas-Meyer-Oseas (MMO), Davies-Meyer (DM) and Miyaguchi-Preneel (MP) [Menezes et al. 1997], and double-block-length hash modes such as Tandem-DM, Hirose's, Abreast-DM, Parallel-DM and MDC-2. Attack complexities are listed in Table 1. Memory complexity is negligible, since we do not need to store the messages, and since we perform distinguishing attacks only. There is no point in key-recovery attacks since there is no secret value involved.

We describe how $\chi^2$ correlations can be extended from the underlying block ciphers to the compression function, in several modes of operation, so that these primitives do not behave ideally, but can be distinguished from a random mapping in a chosen-text setting.

In our analyses, we used variable-round versions of ERC6, RC6 and RC5 ciphers to instantiate the block cipher(s) underlying all modes of operation studied. Our choice of block ciphers is based on the fact that these versions are susceptible to $\chi^2$ cryptanalysis. Nonetheless, they serve as a proof-of-concept. Our point is to make clear that weaknesses such as $\chi^2$ correlations can propagate beyond the block cipher to a compression function in some hashing mode of operation such as DM, MMO and MP, allowing one to distinguish them from random mappings. Unlike other attacks, we do not exploit this weakness to find collisions nor (second) preimages. Rather, we search for other properties ($\chi^2$ correlations) that have been overlooked and that may indicate that these primitives are

 ©2012 SBC — Soc. Bras. de Computação

**Table 1. $\chi^2$ attack complexities on compression functions in several modes of operation. Memory cost is negligible.**

| Data (CMCV) | Time | Modes | Block Cipher |
|---|---|---|---|
| $2^{233}$ | $2^{233}$ | DM, MMO, MP, Parallel-DM, MDC-2 | ERC6-32/44/32 |
| $2^{233}$ | $2^{233}$ | Tandem-DM, Abreast-DM | ERC6-32/44/64 |
| $2^{118}$ | $2^{118}$ | DM, MMO, MP, Parallel-DM, MDC-2 | RC6-32/16/16 |
| $2^{118}$ | $2^{118}$ | Tandem-DM, Abreast-DM | RC6-32/16/32 |
| $2^{57.87}$ | $2^{57.87}$ | DM, MMO, MP, Parallel-DM, MDC-2 | RC5-32/10/8 |
| $2^{57.87}$ | $2^{57.87}$ | Tandem-DM, Abreast-DM | RC5-32/10/16 |

CMCV: Chosen Message or Chaining Variable

not ideal candidates for applications such as pseudorandom number generators and PRP's [NIST 2007a]. We believe these properties are relevant and shall not be underestimated.

Our results were possible due to the fact that the correlation patterns for RC5, RC6 and ERC6 are iterative, that is, they have the same form for both the input and the output of the block cipher. This simple fact allowed us to bypass the feedback and feedforward of values commonly used in modes of operation.

So far, our analyses apply to the compression functions only. We leave as an open problem how to extend our $\chi^2$ correlations attacks to the full hash function.

## References

Borst, J., Preneel, B., and Vandewalle, J. (1999). Linear cryptanalysis of RC5 and RC6. In *Fast Software Encryption (FSE)*, LNCS 1636, pages 16–30. Springer.

Brachtl, B., Coppersmith, D., Hyden, M., Jr., S. M., Meyer, C., Oseas, J., Pilpel, S., and Schilling, M. (1990). Data authentication using modification detection codes based on a public one-way encryption function. US Patent 4908861.

El-Fishawy, N., Danaf, T., and Zaid, O. (2004). A modification of RC6 block cipher algorithm for data security (MRC6). In *International Conference on Electrical, Electronic and Computer Engineering*, pages 222–226.

Hirose, S. (2006). Some plausible constructions of double-block length hash functions. In Robshaw, M., editor, *Fast Software Encryption Workshop, FSE 2006*, LNCS 4047, pages 210–225. Springer.

Hohl, W., Lai, X., Meier, T., and Waldvogel, C. (1993). Security of iterated hash functions based on block ciphers. In D.R.Stinson, editor, *Adv. in Cryptology, Crypto 1993*, LNCS 773, pages 379–390. Springer.

Isogai, N., Miyaji, A., and Nonaka, M. (2002). Cryptanalysis of RC5-64 with improved correlation attack. In *SCIS 2002, The 2002 Symposium on Cryptography and Information Security*, pages 657–662. The Institute of Electronics, Information and Communications Engineers.

Jr., J. N., Sekar, G., de Freitas, D. S., Chiann, C., de Souza, R. H., and Preneel, B. (2009). A new approach to $\chi^2$ cryptanalysis of block ciphers. In P.Samarati, editor, *Information Security Conference (ISC)*, LNCS 5735, pages 1–16. Springer.

 ©2012 SBC — Soc. Bras. de Computação

Kaufman, C., Perlman, R., and Speciner, M. (2002). *Network Security: PRIVATE Communication in a PUBLIC World*. Prentice-Hall.

Knudsen, L. (2002). Correlations in RC6 on 256-bit blocks. NESSIE technical report nes/doc/uib/wp5/022/1, Univ. of Bergen.

Knudsen, L. and Meier, W. (2000). Correlations in reduced round variants of RC6. In B.Schneier, editor, *Fast Software Encryption, FSE 2000, 7th International Workshop*, LNCS 1978, pages 94–108. Springer.

Lai, X. and Massey, J. (1993). Hash function based on block ciphers. In *Adv. in Cryptology, Eurocrypt 1992*, LNCS 658, pages 55–70. Springer.

Menezes, A., van Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

Meyer, C. and Schilling, M. (1988). Secure program load with manipulation detection code. In *Proceedings 6th Worldwide Congress on Computer and Communications Security and Protection (SECURICOM 1988)*, pages 111–130.

Miyaji, A., Nonaka, M., and Takii, Y. (2002). Improved correlation attack on RC5. IEICE Trans. on Fundamentals, vol. E85-A, no. 1, 44–57.

NIST (1993). FIPS: Data encryption standard. Federal Information Processing Standards Publication 46-2, supersedes FIPS PUB 46-1.

NIST (2001). FIPS: Advanced encryption standard. Federal Information Processing Standards Publication 197.

NIST (2007a). Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. Federal Register, vol. 72, no. 212, Nov.

NIST (2007b). Cryptographic hash algorithm competition. available at http://csrc.nist.gov/groups/ST/hash/sha-3/index.html.

Preneel, B., Govaerts, R., and Vandewalle, J. (1994). Hash functions based on block ciphers: a synthetic approach. In *Adv. in Cryptology, Crypto 1993*, LNCS 773, pages 368–378. Springer.

Ragab, A., Ismail, N., and Allah, O. (2001). Enhancements and implementation of RC6 block cipher for data security. IEEE TENCOM.

Rijmen, V., Preneel, B., and Win, E. D. (1997). On weaknesses of non-surjective round functions. *Design, Codes and Cryptography*, 12(3):253–266.

Rivest, R. (1994). The RC5 encryption algorithm. In *Proceedings 2nd International Workshop on Fast Software Encryption (FSE)*, LNCS, pages 86–96. Springer.

Rivest, R., Robshaw, M., Sidney, R., and Yin, Y. (1998). The RC6 block cipher. http://www.rsa.com/rsalabs.

Shimoyama, T., Takeuchi, K., and Hayakawa, J. (2001). Correlation attack to the block cipher RC5 and the simplified variants of RC6. Advanced Encryption Standard (AES3) Conference, Jun.

Stallings, W. (2003). *Cryptography and Network Security: Principles and Practice*. Prentice Hall.

 ©2012 SBC — Soc. Bras. de Computação

## A. Modes of Operation Schematics

Fig. 1 illustrates several modes of operation discussed in this paper. $E$ stands for a block cipher; the triangle in the side of the square holding $E$ indicates the key input; $c$ is a constant.

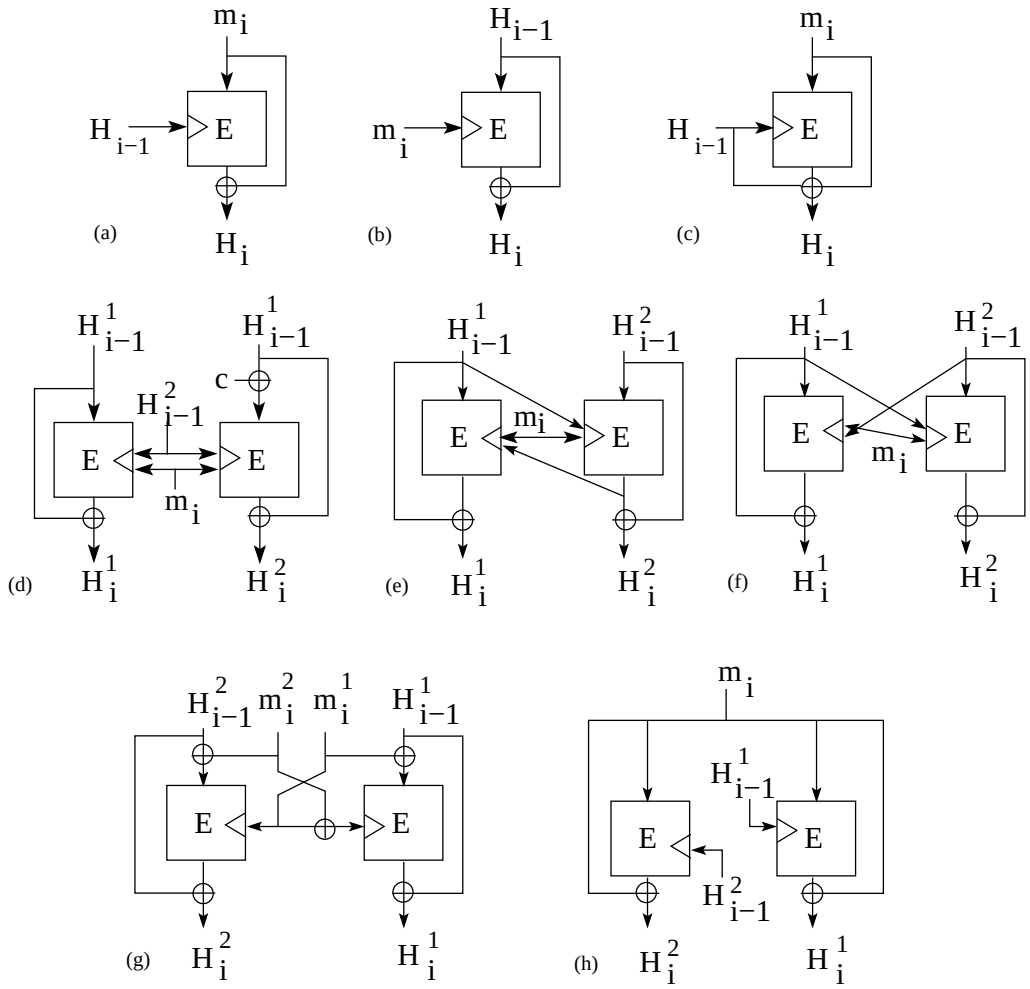



**Figure 1. Modes of operation: (a) Matyas-Meyer-Oseas (MMO); (b) Davies-Meyer (DM); (c) Miyaguchi-Preneel (MP); (d) Hirose's; (e) Tandem-DM; (f) Abreast-DM; (g) Parallel-DM; (h) MDC-2**

 ©2012 SBC — Soc. Bras. de Computação

# Impossible-Differential Attacks on block-cipher based Hash and Compression Functions using 3D and Whirlpool

**Daniel Santana de Freitas**[1], **Jorge Nakahara Jr**

[1]Depto de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC - Brazil

santana@inf.ufsc.br, jorge_nakahara@yahoo.com.br

***Abstract.*** *In this paper, we analyse block-cipher-based hash functions, which means hash functions that use block ciphers as compression functions in a mode of operation, such as Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP), for instance. We use impossible differentials (ID) to distinguish the compression (or hash) function from an ideal primitive (a random oracle) by detecting a nonrandom behavior. We applied an ID analysis to an 8-round variant of the 3D block cipher used in MMO mode, as a compression function of a hypothetical hash function. This attack effectively improves upon the previously known distinguishing ID attacks on reduced-round 3D. We can also attack a hash function using 3D as compression function in DM mode. Finally, we attacked the compression function in Whirlpool with a 5-round $W$ cipher in MP mode with $2^{100}$ time and $2^{64}$ memory.*

Keywords: impossible differentials, block-cipher-based hash functions, modes of operation, Whirlpool hash function.

## 1. Introduction

There are several properties a hash function is expected to satisfy, depending on its application. The most well-known are: collision resistance, preimage resistance and second preimage resistance [17]. But, there are also other relevant properties such as partial preimage resistance, and nonrandom properties [11, 19, 13] that although not leading to collisions nor to (second) preimages, may turn the hash function unsuitable to applications that expect it to behave as a random mapping [22], in some particular (abstract) model. Thus, detectable evidence of nonrandom behavior demonstrates that the hash function is not an ideal primitive, which is expected for a high-security cryptographic function. For instance, in the SHA-3 competition [23], NIST has requested that candidate hash function algorithms behave as close as possible to random oracle. In this paper, we apply the impossible-differential (ID) technique to detect nonrandom properties in the compression or the hash functions.

The impossible differential (ID) approach is a chosen-plaintext technique formerly described in [12]. To construct ID distinguishers, truncated differentials and the miss-in-the-middle technique [5] are typically used. In our case, a truncate differential has two types of difference: zero and nonzero (byte) difference. Moreover, the nonzero difference value is not relevant (so we do not pay in probability for any specific difference pattern). The idea of the miss-in-the-middle approach is to concatenate two truncated differentials, say, $\alpha \xrightarrow{f} \beta$ and $\epsilon \xleftarrow{g} \eta$, both of which hold with certainty, into a single differential

 ©2012 SBC — Soc. Bras. de Computação

$\alpha \overset{g^{-1}\circ f}{\rightarrow} \eta$, where $g^{-1} \circ f$ stands for the functional composition of $f$ and $g^{-1}$ in this order. Nonetheless, $\beta \neq \epsilon$, that is, the differences do not match in the middle of the distinguisher, which leads to a contradiction. The terminology $\alpha \overset{f}{\rightarrow} \beta$ means that the difference $\alpha$ causes difference $\beta$ after the transformation $f$ in the encryption (or forward) direction; $\epsilon \overset{g}{\leftarrow} \eta$ means that the difference $\eta$ causes difference $\epsilon$ after the transformation $g$ in the decryption (or backwards) direction. Note the direction of the arrows. Consequently, $\alpha \overset{g^{-1}\circ f}{\rightarrow} \eta$ holds with probability zero, or analogously, $\alpha \overset{g^{-1}\circ f}{\not\rightarrow} \eta$ holds with certainty. The terminology $\alpha \overset{g^{-1}\circ f}{\not\rightarrow} \eta$ means that $\alpha$ can never cause the difference $\eta$ across the transformations $g^{-1} \circ f$ in the encryption direction. Analogously, the same reasoning applies for the decryption direction. This set of differences $\alpha$, $\beta$, $\epsilon$, $\eta$ characterize an ID distinguisher constructed with the miss-in-the-middle technique. The initial difference $\alpha$ and the final difference $\eta$ will be clearly identified in the ID distinguishers (3) and (9) in this paper.

The ID technique has already been applied to several ciphers, including IDEA and Khufu [5], Rijndael [2, 6], among others.

Table 1 compares our distinguishing attacks and previous ones on reduced-round versions of the 3D cipher. Table 2 summarizes attacks on Whirlpool and its compression function.

This paper is organized as follows: Sect. 2 summarizes the contributions of this paper; Sect. 3 briefly describes the 3D block cipher; Sect. 3.1 describes a distinguishing attack on a compression function using an 8-round version of the 3D cipher in the MMO mode; Sect. 4 presents an attack on the Whirlpool hash function, whose block cipher $W$ in the compression function is reduced to 8 rounds; Sect. 5 concludes this paper.

## 2. Contributions

The contributions of this paper include

- new distinguishing attacks on compression and hash functions that use block ciphers as building blocks in well-known modes of operation such as Matyas-Meyer-Oseas (MMO), Davies-Meyer (DM) and Miyaguchi-Preneel (MP) [17].
- our attacks use for the first time the impossible differential technique to demonstrate properties of the underlying block cipher that may propagate to the entire hash function, depending on the padding, the number of rounds of the block cipher and on the (hashing) mode of operation surrounding the block cipher. In our case, we attack one or two rounds beyond the ID distinguishers for the underlying primitives, showing that there is a further margin of insecurity to take into account when ciphers are used as building blocks.
- we do not aim at traditional attacks such as collisions or (second) preimage, but our findings are relevant in applications where hash functions are expected to behave as random oracles such as (pseudo-)random number generators, and taking into account NIST's requirements for the new SHA-3 algorithms [22, 23].
- our targets are a compression function using the 3D block cipher [20] (in MMO and DM modes), and the $W$ block cipher used in Whirlpool hash function [4] (in MP mode). For 3D, we effectively improve on previous distinguishing attacks on

 ©2012 SBC — Soc. Bras. de Computação

reduced-round 3D (see Table 1), and could attack a hash function in DM mode. In this respect, the DM mode was more vulnerable than the other modes.

## 3. Impossible Differential of 8-round 3D Cipher

The 3D block cipher operates on $512$-bit blocks under a $512$-bit user key, both of which are represented as a $4 \times 4 \times 4$ state of bytes [20]. The three-dimensional cipher state for a $64$-byte data block $A = (a_0, a_1, \ldots, a_{63})$ is denoted as

$$\text{State} = \left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_1 & a_5 & a_9 & a_{13} & a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} \\ a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} & a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} \\ a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} & a_{51} & a_{55} & a_{59} & a_{63} \end{array}\right) \tag{1}$$

that is, bytes are ordered column-wise. The round subkeys follow this same ordering and structure. We denote the first byte of subkey $k_i$ as $k_{i,0}$, the second byte as $k_{i,1}$, and so on. Each set of $16$ bytes in a $4 \times 4$-byte square is called a slice of the state. In total, the 3D cipher iterates 22 rounds. The round transformations in 3D have a clear correspondence with those of the AES [7, 10]. Using the terminology of [20]:

- $\kappa_i$: bit-wise xor with round subkey, equivalent to AddRoundKey in AES;
- $\gamma$: a byte-wise application of the AES S-box, equivalent to SubBytes in AES;
- $\theta_1$, $\theta_2$: equivalent to ShiftRows in AES but applied to each slice of the state alternately; $\theta_1$ in the odd-numbered rounds, $\theta_2$ in the even-numbered rounds;
- $\pi$: matrix multiplication with columns of the state, equivalent to MixColumns in AES.

Each round transformation stands for a fraction of 0.25 (a quarter) of a round. Distinguishers for 3D may sometimes cover fractions of a round, such as 5.25 rounds, for instance (see Table 1). The $i$-th full round of 3D, encrypting a text block $X$, is denoted $\tau_i(X) = \pi \circ \theta_{i \bmod 2+1} \circ \gamma \circ \kappa_i(X) = \pi(\theta_{i \bmod 2+1}(\gamma(\kappa_i(X))))$. The last round does not include $\pi$. The key schedule of 3D follows a similar framework as encryption. We refer to [20] for further details.

### 3.1. A distinguishing attack on the compression function

Suppose an 8-round variant of the 3D cipher, with the first round starting with $\theta_1$, is used as compression function in a hash function in Matyas-Meyer-Oseas (MMO) mode [17]. We show that the impossible-differential technique can be used to distinguish this variant as compression function from a random oracle (RO), that is, to detect a nonrandom behavior of an 8-round 3D variant. Recall that we apply a distinguish-from-random type of attack. This nonrandom property does not imply a collision nor a (second) preimage.

Let $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ denote an $n$-bit block cipher parameterized by a $k$-bit key $K$. Let $H_i$ denote the $i$-th chaining variable during the hash computation, with the initial value IV= $H_0$; let $m_i \in \{0,1\}^n$ denote the $i$-th block of message being hashed; and let $g : \{0,1\}^n \to \{0,1\}^k$. The MMO mode computes

$$H_i = m_i \oplus E_{g(H_{i-1})}(m_i), \tag{2}$$

that is, $g$ adapts the $(i-1)$-th chaining variable of $n$ bits to the key input of $k$ bits. In our case, $E$ is the 3D cipher, $k = n = 512$, and $g$ becomes the identity mapping.

 ©2012 SBC — Soc. Bras. de Computação

The setting of this attack is against the compression function only. $H_0$ and $m_i$ are known, and $H_i = m_i \oplus E_{H_{i-1}}(m_i)$ in the MMO mode. Notice that the feedforward of $m_i$ in (2) can be moved: $E_{H_{i-1}}(m_i) = H_i \oplus m_i$. Thus, for a fixed key $H_{i-1}$, the adversary has access to the input and output of $E$. Using the terminology of Sect. 1, $\alpha = m_i \oplus m_i^*$ and $\eta = E_{H_{i-1}}(m_i) \oplus E_{H_{i-1}}(m_i^*)$ because $H_{i-1}$ is the key and is kept fixed.

The 6-round impossible differential (ID) distinguisher used in this attack is depicted in (3), where '0' denotes a zero byte difference and '$\Delta$' denotes an (arbitrary) nonzero byte difference. Informally, this ID distinguisher means that a single-byte text difference in any of the 64 positions in the message input $m_i$ (input text difference is denoted $\alpha$) cannot lead to a 16-byte zero difference pattern (denoted $\eta$) after 6-round 3D (a very structured difference pattern). This is a slightly modified version of the distinguisher whose construction details can be found in [21]. In summary, there is a contradiction after $\pi$ of the third round: there are only $\Delta$ bytes in each column of the leftmost slice before $\pi$, but only zero byte differences in all four columns after $\pi$, which contradicts the branch number of $\pi$ which is five[1]. In other words, $\Pr(\alpha \rightarrow \eta) = 0$. Notice that there are four impossible output difference patterns, by choosing one of the four patterns of rows with zero byte difference in $\eta$. Note that the output difference $\eta$ contains zero byte differences only in positions (0, 4, 8, 12, 19, 23, 27, 31, 34, 38, 42, 46, 49, 53, 57, 61) of the state (1). Following this construction, let us call $\eta_1$, $\eta_2$ and $\eta_3$ the three other impossible output difference patterns with zero byte differences in positions (1, 5, 9, 13, 16, 20, 24, 28, 35, 39, 43, 47, 50, 54, 58, 62), (2, 6, 10, 14, 17, 21, 25, 29, 32, 36, 40, 44, 51, 55, 59, 63) and (3, 7, 11, 15, 18, 22, 26, 30, 33, 37, 41, 45, 48, 52, 56, 60), respectively.

The attack framework is depicted in (4), where '*'s denote chosen input byte positions before the 6-round distinguisher (3). Thus, $i = 2$ in $\kappa_i$. We use text structures to cover the first two rounds and the 6-round ID distinguisher afterwards, for a total of eight rounds. Note that starting one round before the distinguisher provides only $2^{32}(2^{32} - 1)/2 \approx 2^{63}$ text pairs for the attack, which is not enough, while starting the attack three rounds before the distinguisher would require the full codebook. So, two

[1]The branch number [7, 10] is a measure of the diffusion power of a mapping $F : \mathrm{GF}(2^n)^m \rightarrow \mathrm{GF}(2^n)^m$. In the case of 3D, $n = 8$ and $m = 4$. The branch number of $F$ is $B_F = \min_{x \neq 0}\{HW(x) + HW(F(x))\}$, where HW denotes the Hamming weight. Let $F$ be $\pi$ which consists of a $4 \times 4$ matrix from an MDS code. From the theory of error correcting codes [15], the MDS code guarantees a minimum distance of 5 between two codewords, that is, $B_\pi = 5$.

 ©2012 SBC — Soc. Bras. de Computação

rounds before the distinguisher as in (4) is the best trade-off.

$$\alpha = \left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \xrightarrow{\pi\circ\theta_1\circ\gamma\circ\kappa_i}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$\xrightarrow{\theta_2\circ\gamma\circ\kappa_{i+1}} \left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \end{array}\right)$$

$$\xrightarrow{\theta_1\circ\gamma\circ\kappa_{i+2}} \left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 \\ 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 & 0 & \Delta & 0 & 0 \end{array}\right) \xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{array}\right) \xrightarrow{\theta_2\circ\gamma\circ\kappa_{i+3}}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{array}\right) \not\xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ 0 & 0 & 0 & 0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{array}\right) \xleftarrow{\kappa_{i+4}\circ\gamma^{-1}\circ\theta_1^{-1}\circ\pi^{-1}}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{array}\right)$$

$$\xleftarrow{\kappa_{i+5}\circ\gamma^{-1}\circ\theta_2^{-1}\circ\kappa_{i+6}} \left(\begin{array}{cccc|cccc|cccc|cccc} 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 \\ \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta \\ \Delta & \Delta & \Delta & \Delta & 0 & 0 & 0 & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \end{array}\right) = \eta \quad (3)$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * \end{array}\right) \xrightarrow{\theta_1\circ\gamma\circ\kappa_0}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \\ * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 & * & 0 & 0 & 0 \end{array}\right) \xrightarrow{\pi}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \end{array}\right) \xrightarrow{\theta_2\circ\gamma\circ\kappa_1}$$

$$\left(\begin{array}{cccc|cccc|cccc|cccc} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$\xrightarrow{\pi} \text{6-round ID distinguisher (3)} \quad (4)$$

The attack proceeds as follows: in (4) consider a (text) structure consisting of $(2^8)^{16} =$

 ©2012 SBC — Soc. Bras. de Computação

$2^{128}$ chosen plaintexts $m_i$, in which each byte, denoted by '*' in (4), in the diagonal of each slice of the state of 3D assumes all possible byte values, while the remaining state bytes are constants. Each structure of $2^{128}$ plaintexts contains $2^{128}(2^{128}-1)/2 = 2^{255} - 2^{127}$ text pairs $P_i \oplus P_j$, for $i \neq j$. After one full round, with probability $16 \cdot (2^{-8 \cdot 3})^4 = 2^{-92}$, there will be a single nonzero byte difference per slice of the state, that is, each column of nonzero byte difference, the '*'s, will collapse into a single nonzero byte difference. Notice that there are four possibilities for the single nonzero byte differences to be placed in specific positions for the next round so that they will align into a single column after $\theta_2$. In summary, for the given configuration of '*'s in (4) there are four possible columns where the four '*'s can end up after the first round, and for each column there are four possible places they can be after two rounds. We expect $(2^{255} - 2^{127}) \cdot 2^{-92} = 2^{163} - 2^{35}$ pairs to survive this filtering per structure. After the second round, with probability $4 \cdot 2^{-24} = 2^{-22}$, we expect that the four nonzero byte differences lead to a single nonzero byte difference, which is exactly the input difference to the 6-round ID distinguisher (3). This filtering leaves $(2^{163} - 2^{35}) \cdot 2^{-22} = 2^{141} - 2^{13}$ pairs.

Thus, these $2^{141} - 2^{13}$ text pairs will never lead to a forbidden output difference of (3), since they lead to an impossible differential. The remaining $2^{255} - 2^{127} - 2^{141} + 2^{13}$ pairs may satisfy the output difference of (3), which has sixteen zero byte differences. We expect that a remaining pair can satisfy one of the four output differences ($\eta$, $\eta_1$, $\eta_2$, $\eta_3$) with probability $2^{-128} \times 4 = 2^{-126}$. We expect that $(2^{255} - 2^{127} - 2^{141} + 2^{13}) \cdot 2^{-126} = 2^{129} - 2 - 2^{15} + 2^{-113}$ pairs to survive this filtering. For a random permutation on the same plaintext space as 3D, we expect $(2^{255} - 2^{127}) \cdot 2^{-126} = 2^{129} - 2$ pairs to survive. We exclude from the random permutation set the dual ciphers of 3D, in a similar setting as the duals of the AES [3], which would behave as 3D under the ID distinguisher. Thus, we can distinguish 8-round 3D cipher from a random permutation mapping using $2^{128}$ blocks of memory (for storing the ciphertexts) and effort of $2^{128}$ 8-round computations. Note that unlike in conventional impossible-differential attacks, we use both the text pairs that lead to the ID distinguisher differences and those that do not.

In a block cipher setting, the data complexity is $2^{128}$ chosen plaintexts, but in a compression/hash function setting there is no secret key, or more precisely, the key is known, so the adversary does not depend on the legitimate parties to obtain data. Therefore, in the latter, there is no data complexity, since the adversary knows the key e.g. a chaining variable. Note that the work effort consists in checking the difference in text pairs corresponding simply to the xor of two text blocks. But this xor is much cheaper than a single encryption and is done offline by the adversary. For this attack, in particular, we are interested only in the ciphertext pairs, $C_i \oplus C_j$. Therefore, the $2^{255}$ text pairs do not account for the time complexity. The effort actually come from $2^{128}$ texts encrypted in a structure.

The difference between the number of surviving pairs for reduced-round 3D and a random permutation is just $2^{15}$, which means an advantage of only $2^{15}/2^{129} = 2^{-114}$. Following [1][Sect.6.7], in a distinguish-from-random setting times, to achieve a high success rate attack, the number of queries shall be proportional to $(2^{-114})^{-2} = 2^{228}$. Since we already use $2^{128}$ texts in a structure, this means that the attack shall be repeated $2^{100}$ times which means using $2^{100}$ structures in total. This factor leads to an updated effort of $2^{100} \cdot 2^{128} = 2^{228}$ 8-round 3D computations, but the memory cost remains the same since

 ©2012 SBC — Soc. Bras. de Computação

we do not need to store all structures at the same time. There is no shortage of structures to repeat the attack. Simply choose other constant values for the zero-difference bytes of the state in (3). Since there are 48 bytes with difference zero, one can choose up to $256^{48} = 2^{384}$ constants and thus up to $2^{384}$ structures.

We could describe other 6-round ID distinguishers for 3D, as well as attacks on 8-round 3D versions using these distinguishers simultaneously, just like (3) and (4) without increasing the attack complexities. These additional distinguishers demonstrate that there are large sets of text pairs inside 8-round 3D that never lead to a set of forbidden differences, in comparison to a random permutation. As an example of these distinguishers, note that at the minimum we only need four zero byte differences in $\eta$, for instance, in positions (0, 19, 34, 49) of the state (1). These four byte differences will lead to a single column of zeros after $\pi$ of the 4th round, leading to a contradiction. Note further that there are sixteen such zero-byte difference patterns, leading to sixteen 6-round ID distinguishers for 3D: (4, 23, 38, 53), (8, 27, 42, 57), and so on.

Notice that the text structure in (4) has bytes with nonzero difference over the padding part, assuming the Merkle-Damgård paradigm and a message space with size at least $2^{64}$, which implies bytes $a_{56}$ until $a_{63}$ of the state (1) are reserved for the original message size. Thus, the attack is restricted to the compression function only, and does not apply to the hash function.

The distinguishing attack described in this section improves on the attack described in [21], which was restricted to 6-round 3D, both in terms of number of rounds and attack complexities.

### 3.2. A distinguishing attack on the hash function

Consider now the same 8-round variant of the 3D cipher in Sect. 3.1, but this time the mode of operation is Davies-Meyer (DM) [17] and the target is the hash function using the Merkle-Damgård (MD) padding scheme [8, 18]. In the MD scheme the padding consists of a single '1' bit followed by '0' bits up to $512 - l$ bits, so that the last $l$ bits store the size of the original message to be hashed. For instance, if $l = 128$, then the bytes in positions $a_{48}$ until $a_{63}$ of the state (1) are reserved for the message size.

In the attack setting, $H_0 = IV$ and $m_i$ are known, and the updating rule in the DM mode is

$$H_i = H_{i-1} \oplus E_{m_i}(H_{i-1}). \tag{5}$$

Notice that the feedforward of $H_{i-1}$ can be moved: $E_{m_i}(H_{i-1}) = H_i \oplus H_{i-1}$. Thus, for a fixed key $m_i$, the adversary has access to the input and output of $E$ (that is 8-round 3D).

In this section we show that this hash function can be distinguished from a random oracle (RO) using the impossible-differential technique. The attack is on 2-block messages, $M = m_0 \| m_1$, with $m_0, m_1 \in \{0,1\}^{512}$, but can be extended to larger messages $M$ as well. The attack proceeds as follows:

- we vary the value of the first 512-bit block, $m_0$ to generate text pairs for the attack on the second $E$ instance; in the DM mode, $H_1 = H_0 \oplus E_{m_0}(H_0)$ vary. Note that $E_{m_0}(H_0)$ behaves as a random function for variable $m_0$ and fixed $H_0$. We need

 ©2012 SBC — Soc. Bras. de Computação

$2^{228}$ distinct values of $H_1$ as input to the second $E$ instance, according to the attack described in Sect. 3.1. Following [24], we expect about $2^{224} \cdot e \approx 2^{225.44}$ values of $m_0$ to obtain $2^{224}$ distinct values of $H_1$.

- we keep $m_1$ always fixed in all messages $M$; recall that $m_1$ contains the padding and message size since it is the last message block.
- since $m_1$ is fixed, the second $E$ instance performs a permutation, with input $H_1$, from which we can generate difference pairs for the impossible differential attack in Sect. 3.1.
- we can therefore, detect differences in $H_2$ for the impossible differential, and the attack complexities of Sect. 3.1 hold, but this time the analysis is applied to the hash function, not to the compression function. The $H_1$ value can be computed and separated from $H_2$ to obtain the output of $E_{m_1}(H_1)$ only.

Since we have encryptions of $m_0 \| m_1$ that is, two $E$ computations at a time, the attack effort becomes $2^{226.44}$ 8-round 3D computations.

For larger messages, $M$, say, $t$-block messages, the idea is to vary $m_1$, $m_2$, ..., $m_{t-1}$, but not the last block $m_t$, which contains the padding and message size ($512 \cdot t$ bits), forming the Merkle-Damgård strengthening.

In the impossible-differential attack setting, the DM mode is weaker than the MMO mode, since an adversary can control the key input in DM mode (which is a message block) making the underlying cipher in the compression function to behave as a permutation, which is the setting needed for an impossible-differential attack.

## 4. A distinguishing attack on 5-round *W* in Whirlpool

The Whirlpool hash function was designed by Barreto and Rijmen [4] in 2003, according to the wide-trail design strategy. Whirlpool follows the Merkle-Damgård strengthening (message size is limited to $2^{256}$ bits) and uses a block cipher called $W : \{0,1\}^{512} \times \{0,1\}^{512} \rightarrow \{0,1\}^{512}$ in MP mode. The cipher $W$ is similar to the AES but operates on an $8 \times 8$ state of bytes where bytes are input row-wise, and $W$ iterates ten rounds. An example of a state of $W$ indicating the byte ordering is depicted in (6).

$$\text{State} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & a_{21} & a_{22} & a_{23} \\ a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{30} & a_{31} \\ a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{48} & a_{49} & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ a_{56} & a_{57} & a_{58} & a_{59} & a_{60} & a_{61} & a_{62} & a_{63} \end{pmatrix} \tag{6}$$

Following the terminology of [4], the rounds operations in $W$ are

- $\gamma$: nonlinear layer; byte-wise application of an $8 \times 8$-bit S-box;
- $\pi$: byte permutation, i.e. column $j$ is cyclically shifted down by $j$ positions, where columns are numbered from 0 in left-to-right order;
- $\theta$: diffusion layer; multiplication of each row of the state by an $8 \times 8$ MDS matrix;
- $\sigma_k$: exclusive-or between the state and the $k$-th round subkey (which are predefined constants).

The $k$-th full round of $W$, operating on a text block $X$, consists of $\tau(X) = \sigma_k \circ \theta \circ \pi \circ \gamma(X) = \sigma_k(\theta(\pi(\gamma(X)))$. There is a pre-whitening layer consisting of $\sigma_0$.

 ©2012 SBC — Soc. Bras. de Computação

The ID distinguisher (9) for Whirlpool (in the appendix) covers four rounds, and was constructed with the miss-in-the-middle technique. It consists of two truncated differentials: one in the encryption direction covering three rounds, and the other differential in the decryption direction covering one round. The contradiction (or miss-in-the-middle) happens in the top row of the state after the third round. A row with all eight nonzero byte differences cannot cause a row with only zero byte differences because the MDS matrix used in $\theta$ has branch number nine [7]. Notice that there are $64 \cdot 8$ equivalent 4-round ID distinguishers since the initial nonzero $\Delta$ byte can be in any of the 64 positions of the state, and the output pattern of eight zero byte differences can follow eight possible patterns. The attack framework is depicted in (7) and starts one round before the ID distinguisher (9) covering a total of five full rounds of $W$.

$$\begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\pi \circ \gamma \circ \sigma_i} \begin{pmatrix} * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{\theta} \text{4-round ID distinguisher (9)} \quad (7)$$

The attack setting is the following: Whirlpool uses the MP mode. Let $H_0 = IV$ denote the initial value of the chaining variable; let $m_i$ denote the $i$-th message block (the input to $W$). We aim to find a nonrandom behaviour of the compression function.

The Miyaguchi-Preneel (MP) mode [17] is the following:

$$H_i = m_i \oplus H_{i-1} \oplus E_{H_{i-1}}(m_i), \quad (8)$$

where $E$ is the block cipher $W$ in this case. For Whirlpool, the text and key input of $W$ have the same size. We keep $H_{i-1}$ fixed as key input, so that $W_{H_{i-1}}(m_i)$ behaves as a permutation.

The attack proceeds as follows: consider a (text) structure consisting of $(2^8)^8 = 2^{64}$ chosen messages (plaintexts) $m_i$, in which the eight bytes in the pattern (7) of the state $W$ denoted '*' assume all possible byte values, while the remaining state bytes are constants. Each structure of $2^{64}$ plaintexts contain about $2^{64}(2^{64}-1)/2 = 2^{127} - 2^{63}$ text pairs $m_i^j \oplus m_i^k$, for $k \neq j$. After one full round, with probability $8 \cdot (2^{-8})^7 = 2^{-53}$, there will be a single nonzero byte difference in the first row of the state, that is, the first row will collapse into a single nonzero byte difference in one of the first eight positions in that row. Notice that there are eight possibilities for the single nonzero byte difference $\Delta$ to be placed in specific positions after the first round. We expect $(2^{127} - 2^{63}) \cdot 2^{-53} = 2^{74} - 2^{10}$ pairs to survive this filtering per structure. This single-byte difference is exactly the input difference to the 6-round ID distinguisher (9). The remaining $2^{127} - 2^{63} - 2^{74} + 2^{10}$ pairs may satisfy an output difference of (9) which has eight zero byte differences.

The pattern $\eta$ in (9) has zero byte differences in positions (0, 9, 18, 27, 36, 45, 54, 63), the main diagonal of the state (6). Notice that the eight zero byte differences could follow any of eight possible patterns, each in a sub-diagonal of the state. Thus, in total, there are eight forbidden output patterns, including $\eta$, and we expect that a remaining pair can satisfy one of these output differences with probability $8 \cdot 2^{-64} = 2^{-61}$.

Using a single structure, we expect that $(2^{127} - 2^{63} - 2^{74} + 2^{10}) \cdot 2^{-61} = 2^{66} - 2^2 - 2^{13} + 2^{-51}$ pairs survive for 5-round $W$ as compression function. But, for a random

 ©2012 SBC — Soc. Bras. de Computação

permutation, we expect $(2^{127} - 2^{63}) \cdot 2^{-61} = 2^{66} - 2^2$ pairs to survive. Thus, due to the deficit in the number of surviving pairs for each cipher, we can distinguish a compression function containing 5-round $W$ from a random mapping (oracle) using $2^{64}$ blocks of memory and an effort of $2^{64}$ 5-round $W$ computations.

The difference between the number of surviving pairs for reduced-round $W$ and a random permutation is just $2^{13}$, which means an advantage of $2^{13}/2^{63} = 2^{-50}$. Following [1][Sect.6.7], in a distinguish-from-random setting, to achieve a high success rate attack, the number of queries shall be proportional to $(2^{-50})^{-2} = 2^{100}$. Since we already use $2^{64}$ texts in a structure, this means that the attack shall be repeated $2^{36}$ times which means using $2^{36}$ structures. This factor leads to an updated effort of $2^{36} \cdot 2^{64} = 2^{100}$ 5-round $W$ computations, but the memory cost remains the same since we do not need to store all structures at the same time. There is no shortage of structures to repeat the attack. To find more structures simply choose other constant values for the zero-difference bytes of the state in (7). Since there are 56 bytes with difference zero in (7), one can choose up to $256^{56} = 2^{448}$ constants and thus, up to $2^{448}$ structures.

Notice that the text structure in (7) has (message) bytes with nonzero difference across a sub-diagonal of the state (6). That means these bytes straddle over the padding and message length parts, assuming the Merkle-Damgård paradigm and a message space with size $2^{256}$. Even if the message space were as low as $2^{64}$, the attack would still be restricted to the compression function only.

## 5. Conclusions

This paper presented distinguishing attacks on block-cipher-based compression and hash functions using the impossible-differential technique. In particular, we analysed block-cipher-based hash functions in three classical hash modes: Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP). Our attacks focused on reduced-round versions of the 3D and $W$ block ciphers as compression functions.

Table 1 lists the complexities of ours attacks on a compression function and previous attacks using the 3D cipher. This comparison is for completeness purposes only, since the attack settings are not the same (block cipher and compression/hash function). Our results are not a threat to the full 22-round 3D cipher, but compare quite favorably with the 6-round distinguishing attack in [21]. Note that our results apply in a distinguish-from-random setting, while other attacks operate in key-recovery settings since the latter does not make sense for compression and hash functions. Notice that in a block cipher setting there is a secret key and thus, in a chosen-plaintext attack the adversary needs the legitimate parties to encrypt/decrypt texts for him, because he does not know the key. In a compression function setting the key to the underlying block cipher is not secret, so the adversary does not depend on the legitimate parties to encrypt or decrypt. In this latter case, there is no data complexity, since the adversary can encrypt/decrypt any text (offline).

Table 2 compares the complexities of ours and previous attacks on reduced-round Whirlpool, not just the compression function. It is not straightforward to compare our results with previous attacks since we do not aim at collisions or (second) preimages. Our motivation in looking for nonrandom behavior of the compression and hash function constructions was based in [22, 23], where NIST stated the need for candidate hash functions

 ©2012 SBC — Soc. Bras. de Computação

**Table 1. Attack complexities on reduced-round versions of the 3D cipher.**

| Technique | Time | Data | Memory | #Rounds | Ref. | Setting |
|---|---|---|---|---|---|---|
| Multiset | $2^{19.5}$ | $2^9$ CP | $2^8$ | 4.75 | [20] | block cipher |
| ID | $2^{65.5}$ | $2^{36}$ CP | $2^{32}$ | 5.75 | [20] | block cipher |
| Multiset | $2^{139}$ | $2^{129}$ CP | $2^{128}$ | 5.75 | [20] | block cipher |
| ID | $2^{256}$ | $2^{256}$ CP | $2^{256}$ | 6 | [21] | block cipher |
| Known-key | $2^8$ | $2^8$ CP | negligible | 7.75 | [21] | block cipher |
| ID | $2^{228}$ | — | $2^{128}$ | 8 | Sect. 3.1 | comp. function (MMO mode) |
| ID | $2^{226.44}$ | — | $2^{128}$ | 8 | Sect. 3.2 | hash function (DM mode) |
| Known-key | $2^{56}$ | $2^{56}$ CP | negligible | 9.75 | [21] | block cipher |
| ID | $2^{401}$ | $2^{501}$ CP | $2^{311}$ | 10 | [21] | block cipher |
| Known-key | $2^{200}$ | $2^{200}$ CP | $2^8$ | 15 | [9] | block cipher |

CP: Chosen Plaintext/messages

**Table 2. Attack complexities on reduced-round $W$ cipher of Whirlpool.**

| Technique | Time | Memory | #Rounds | Ref. | Comments |
|---|---|---|---|---|---|
| rebound | $2^{120}$ | $2^{16}$ | 4.5 | [16] | collision |
| ID | $2^{100}$ | $2^{64}$ | 5 | Sect. 4 | non-randomness (comp. function) |
| rebound | $2^{120}$ | $2^{16}$ | 5.5 | [16] | semi-free start collision |
| rebound | $2^{128}$ | $2^{16}$ | 7.5 | [16] | semi-free start near-collision |
| rebound | $2^{184}$ | $2^8$ | 7.5 | [14] | collision |
| rebound | $2^{176}$ | $2^8$ | 9.5 | [14] | near-collision |
| rebound | $2^{188}$ | $2^8$ | 10 | [14] | distinguisher |

to behave as close as possible to ideal functions (random oracles).

In [13], integral distinguishers were found for up to seven rounds of Whirlpool, but they were not applied further to derive attacks.

## References

[1] Baigneres, T.: Quantitative Security of Block Ciphers: Designs and Cryptanalysis Tools. PhD Thesis, Ecole Polytechnique Federale de Lausanne (2008)

[2] Barak, B., Aref, M.R.: Impossible Differential Attack on seven-round AES-128. IET Information Security, (2):2, 28–32 (2008)

[3] Barkan,E., Biham, E.: In How Many Ways can you write Rijndael?. In: Y Zheng (Ed.), Asiacrypt 2002, Springer, LNCS 2501, 160–175 (2002)

[4] Barreto, P.S.L.M., Rijmen, V.: The Whirlpool Hashing Function. available at http://www.larc.usp.br/ pbarreto/WhirlpoolPage.html (2003)

[5] Biham, E., Biryukov, A., Shamir, A.: Miss-in-the-Middle Attacks on IDEA and Khufu. In: Knudsen, L.R. (Ed.), Fast Software Encryption (FSE), Springer, LNCS 1636, 124–138 (1999)

 ©2012 SBC — Soc. Bras. de Computação

[6] Cheon, J.H., Kim, M., Kim, K., Lee, J.Y., Kang,S.: Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. 4th International Conference on Information Security and Cryptology (ICISC), Springer, LNCS 2288, 39–49 (2001)

[7] Daemen,J., Rijmen,V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)

[8] Damgård, I.: A Design Principle for Hash Functions. In Brassard,G. (Ed.), Crypto'89, 9th Annual International Cryptology Conference, 1989. Springer, LNCS 435, 416–427 (1990)

[9] Dong, L., Wu, W., Wu, S., Zou, J.: Known-key distinguishers on round-reduced 3D block cipher. Information Security Applications (WISA), Springer, LNCS 7115, 55–69 (2012)

[10] FIPS197: Advanced Encryption Standard (AES). FIPS PUB 197 Federal Information Processing Standard Publication 197, U.S. Department of Commerce (2001)

[11] Gligoroski, D.: Narrow-pipe SHA-3 candidates differ significantly from ideal random functions defined over big domains. Faculty of Information Technology, Norwegian Univ. of Science and Technology, Trondheim, Norway.

[12] Knudsen, L.R.: DEAL – a 128-bit Block Cipher. Technical Report #151, University of Bergen, Dept. of Informatics, Norway (1998)

[13] Knudsen, L.R.: Nonrandom Properties of reduced-round Whirlpool. NES/DOC/UIB/WP5/016/2, Aug. 15 (2002)

[14] Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schlaffer, M.: Rebound Distinguishers: results on the full Whirlpool Compression Function. Asiacrypt 2009, 15th International Conference, Springer, LNCS 5912, 126–143 (2009)

[15] MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland Mathematical Library (1977)

[16] Mendel, F., Rechberger, C., Schlaffer,M., Thomsen, S.S.: The Rebound Attack: cryptanalysis of reduced Whirlpool and Grøstl. In: Dunkelman, O. (Ed.), Fast Software Encryption (FSE) 2009, Springer, LNCS 5665, 260–276 (2009)

[17] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997)

[18] Merkle, R.C.: One way hash functions and DES. In Brassard,G. (Ed.), Crypto'89, 9th Annual International Cryptology Conference, Springer, LNCS 435, 428–446 (1990)

[19] Mouha, N., Bjorstad, T.E., Preneel, B.: Non-randomness in the Sarmal compression function. The Ecrypt Hash Function website http://ehash.iaik.tugraz.at/wiki/Sarmal

[20] Nakahara Jr, J.: 3D: a Three-Dimensional Block Cipher. In: M.K.Franklin, L.C.K.Hui, D.S.Wong (Eds.), Cryptology and Network Security (CANS), Springer, LNCS 5339, 252–267 (2008)

[21] Nakahara Jr, J.: New Impossible Differential and Known-Key Distinguishers for the 3D Cipher. In: Bao, F., Weng, J. (Eds.), The 7th Information Practice and Experience Conference (ISPEC), Springer, LNCS 6672, 208–221 (2011)

 ©2012 SBC — Soc. Bras. de Computação

[22] NIST: Cryptographic Hash Algorithm Competition. available at http://csrc.nist.gov/groups/ST/hash/sha-3/index.html (2007)

[23] NIST: Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. Federal Register, vol. 72, no. 212, Nov. 2 (2007)

[24] Rijmen, V., Preneel, B., De Win, E.: On weaknesses of non-surjective round functions. Design, Codes and Cryptography (12):3, 253–266 (1997)

## A. Appendix

$$
\alpha = \begin{pmatrix}
\Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\overset{\theta\circ\pi\circ\gamma\circ\sigma_i}{\rightarrow}
\begin{pmatrix}
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\overset{\pi\circ\gamma\circ\sigma_{i+1}}{\rightarrow}
$$

$$
\begin{pmatrix}
\Delta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \Delta & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Delta & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Delta & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \Delta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \Delta & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \Delta & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta
\end{pmatrix}
\overset{\theta}{\rightarrow}
\begin{pmatrix}
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta
\end{pmatrix}
\overset{\theta\circ\pi\circ\gamma\circ\sigma_{i+2}}{\not\rightarrow}
$$

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta
\end{pmatrix}
\overset{\sigma_{i+4}\circ\gamma^{-1}\circ\pi^{-1}\circ\sigma_{i+5}}{\leftarrow}
\begin{pmatrix}
0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & 0 & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & 0 & \Delta & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & 0 & \Delta & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & 0 & \Delta & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & 0 & \Delta & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0 & \Delta \\
\Delta & \Delta & \Delta & \Delta & \Delta & \Delta & \Delta & 0
\end{pmatrix}
= \eta \quad (9)
$$

 ©2012 SBC — Soc. Bras. de Computação

# Segurança do bit menos significativo no RSA e em curvas elípticas

**Dionathan Nakamura**[1], **Routo Terada**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP )
São Paulo, SP – Brazil

{nakamura,rt}@ime.usp.br

***Abstract.*** *The security of the least significant bit of the secret key in the Elliptic Curve Diffie-Hellman (and of the message in the RSA) is related to the security of the whole key (message). In this paper, algorithms are presented to invert these cryptographic systems making use of oracles that predict the LSB. We implement two of them, critical parameters are identified and the original sampling is changed. With the modified sampling we achieve an improvement in the execution times.*

***Resumo.*** *A segurança do bit menos significativo da chave secreta no Diffie-Hellman sobre Curvas Elípticas (e da mensagem no RSA) está relacionada à segurança de toda a chave (mensagem). Neste artigo são apresentados algoritmos que conseguem inverter os criptossistemas citados fazendo uso de oráculos que predizem o LSB. Fazemos a implementação de dois desses algoritmos, identificamos parâmetros críticos e mudamos a amostragem do formato original. Com a modificação na amostragem conseguimos uma melhora nos tempos de execução.*

## 1. Introdução

Algoritmos criptográficos geralmente se baseiam em algum problema computacional sem solução eficiente conhecida, sendo assim considerado um problema difícil — podemos citar o problema do logaritmo discreto (PLD) e o problema da fatoração de inteiros (PFI). As relações entre esses algoritmos e seus problemas são frequentemente alvos de pesquisas, por exemplo, a verificação de equivalência entre os algoritmos e seus respectivos problemas subjacentes.

Dentro da criptografia há casos onde é necessário o uso de números pseudoaleatórios. Muitos estudos também são feitos entorno desses números e um alvo de pesquisa é verificar se um número gerado possa ser considerado criptograficamente seguro.

Algoritmos criptográficos e seus problemas computacionais subjacentes estão relacionados com geradores de números pseudoaleatórios criptograficamente seguros [Blum and Micali 1984]. Podemos citar, por exemplo, o gerador pseudoaleatório de Blum, Blum e Shub [Blum et al. 1986] que é um gerador pseudoaleatório baseado no algoritmo criptográfico Rabin [Rabin 1979]. Ele faz sucessivas encriptações Rabin e utiliza o bit menos significativo (LSB — *Least Significant Bit*) de cada criptograma para compor o número gerado. Um gerador semelhante para o RSA [Rivest et al. 1978] é descrito em [Menezes et al. 1996, Alg.5.35].

 ©2012 SBC — Soc. Bras. de Computação

Pesquisas interessantes têm aparecido relacionando a segurança dos algoritmos criptográficos com os bits que compõem as chaves secretas ou as mensagens encriptadas por esses sistemas [Goldwasser et al. 1982, Ben-Or et al. 1983, Boneh and Venkatesan 1996, Alexi et al. 1988, Fischlin and Schnorr 2000, Boneh and Shparlinski 2001, Jetchev and Venkatesan 2008, Chevalier et al. 2009, Hofheinz and Kiltz 2009, Roh and Hahn 2010], em especial, com o LSB.

Esses estudos analisam o algoritmo RSA e o protocolo Diffie-Hellman [Diffie and Hellman 1976] sobre Curvas Elípticas (DHCE) — enquanto no RSA o interesse está no LSB da mensagem original, no DHCE está no LSB da chave secreta combinada. Nos estudos com RSA, o algoritmo Rabin também é abordado por sua similaridade. Então, verificar a segurança do LSB é o mesmo que verificar a segurança dos números pseudoaleatórios obtidos pelos geradores citados anteriormente.

Dos trabalhos relacionados, destacamos para o RSA o artigo de [Alexi et al. 1988] onde mostrou ser possível inverter o RSA com um oráculo com probabilidade de acerto de $(50+\varepsilon)\%$, sendo $\varepsilon$ positivo e bem pequeno, todavia não negligenciável. Vamos chamar esse trabalho (e o respectivo algoritmo) de ACGS por brevidade.

Para o DHCE, destacamos o artigo de [Boneh and Shparlinski 2001] onde mostrou que o LSB da abscissa do ponto combinado é imprevisível, ou seja, a existência de um oráculo para o LSB implicaria a quebra do DHCE. Vamos chamar esse trabalho (e o respectivo algoritmo) de BS.

Em [Fischlin and Schnorr 2000], os autores mostraram como esses algoritmos podem ter seu tempo de execução demorado, a ponto de ultrapassar o tempo de execução de soluções para problemas difíceis. Aqui surgiu nossa motivação. Os trabalhos são apresentados apenas como artigos científicos, imaginávamos como eles se comportariam depois de implementados. Apesar de apresentarem convergência teórica, queremos verificar se ela se traduz na prática com o uso dos geradores de aleatoriedade que dispomos. Ainda, se o tempo real de computação é viável com recursos computacionais modestos.

Assim, neste trabalho nós implementamos o ACGS para o RSA e em seguida o BS para DHCE. Como parte secundária dos objetivos, queríamos identificar os parâmetros que mais influenciavam no tempo de execução, procurando encontrar onde erros eram superestimados e promover melhorias no modo como os oráculos eram utilizados.

As principais contribuições deste trabalho são as seguintes:

- provê uma implementação da segurança do LSB no algoritmo RSA;
- provê uma implementação da segurança do LSB no protocolo DHCE;
- identificação dos parâmetros relevantes para o tempo de execução;
- conseguimos uma redução no limitante da função de paridade;
- encontramos uma melhoria significativa no número de acessos ao oráculo;
- das melhorias, o ACGS passou a ser mais rápido que o PFI para valores práticos.

**Organização do trabalho** Na Seção 2 apresentamos os conceitos necessários para a compreensão da metodologia desenvolvida. Na Seção 3 são descritos os algoritmos dos trabalhos utilizados e as técnicas utilizadas para a implementação desses. Os resultados obtidos são expostos e discutidos na Seção 4. Por fim, na Seção 5, concluímos a pesquisa e propomos novas direções de trabalhos.

 ©2012 SBC — Soc. Bras. de Computação

## 2. Conceitos preliminares

Consideramos nesse trabalho curvas elípticas $E$ sobre um corpo finito $\mathbb{F}$ de característica prima $\mathrm{char}(\mathbb{F}) = p$, $p \neq 2$, $p \neq 3$ e sendo $p$ grande. Apresentamos uma curva elíptica $E/\mathbb{F}$ na forma da equação simplificada de Weierstrass $E : y^2 = x^3+ax+b$, onde $a, b \in \mathbb{F}$, $\infty$ é o ponto no infinito e o discriminante $\Delta = 4a^3 + 27b^2 \neq 0$.

Sobre a **propriedade multiplicativa** do RSA: considere o criptograma $y \equiv x^e \bmod N$, desejamos obter $z \equiv (cx)^e \bmod N$ para alguma constante $c$. Note ser possível obter o valor de $z$ sem conhecer o valor da mensagem original $x$. Um outro fato importante, a divisão por 2 dentro da aritmética modular é simplesmente o produto com o inverso multiplicativo de 2 módulo $N$. A propriedade multiplicava e os modos de cálculo da divisão por 2 são utilizados no ACGS. Cálculos similares serão feitos em curvas elípticas.

Adotamos notar o módulo $N$ com letra maiúscula e seu tamanho em bits $n = \lfloor \lg N \rfloor + 1$ com letra minúscula. Para facilitar a escrita, seja $x$ um inteiro, denotamos $[x]_N$ como o resto de $x$ módulo N, ou seja, $[x]_N = x \bmod N$.

O que chamamos de oráculo neste trabalho trata-se de um algoritmo que, ao receber como entrada parâmetros públicos de um criptossistema, retorna o LSB da parte secreta desse sistema em tempo polinomial.

Dizemos que um oráculo tem **vantagem** $\varepsilon$ se ele acerta o LSB com uma probabilidade igual ou superior a $\frac{1}{2}+\varepsilon$. Por exemplo, para o DHCE em uma curva $E$, gerador $G$ de ordem $q$ e $a, b$ distribuídos uniformemente em $[1, q-1]$, o algoritmo $\mathcal{A}$ é um oráculo com vantagem $\varepsilon$ em predizer o LSB da coordenada $x$ da função Diffie-Hellman e denotamos $\mathrm{Adv}^{X}_{E,G}(\mathcal{A}) = \left|\mathrm{Pr}_{a,b}[\mathcal{A}(E, G, aG, bG) = \mathrm{LSB}(x)] - \frac{1}{2}]\right| > \varepsilon$.

O máximo divisor comum (MDC, também GCD do inglês *Greatest Common Divisor*) é um algoritmo utilizado aqui para verificar se dois operandos $b$ e $c$ são primos entre si, ou seja, se $\mathrm{MDC}(b, c) = 1$.

Apresentamos uma versão chamada MDC binário baseado em Knuth [Knuth 1981, Sec.4.5.2 Alg.B p.321]. Nela são utilizadas as seguintes propriedades devidas a Stein [Stein 1967]:

- se $|b|$ e $|c|$ são ambos pares, então $\mathrm{MDC}(b, c) = 2 \cdot \mathrm{MDC}(\frac{b}{2}, \frac{c}{2})$;
- se $|b|$ é par e $|c|$ ímpar, então $\mathrm{MDC}(b, c) = \mathrm{MDC}(\frac{b}{2}, c)$;
- se $|b|$ e $|c|$ são ambos ímpares, então $\mathrm{MDC}(b, c) = 2 \cdot \mathrm{MDC}(\frac{b+c}{2}, \frac{b-c}{2})$;
- adicionalmente, no último caso, ou $\frac{b+c}{2}$ ou $\frac{b-c}{2}$ é divisível novamente por 2.

Essas propriedades foram utilizadas para se criar um MDC binário modificado em [Ben-Or et al. 1983] e no Brent Kung GCD [Brent and Kung 1984] utilizado no ACGS.

A Desigualdade de Chebyshev é uma ferramenta estatística que nos permite estabelecer limitantes para uma distribuição de probabilidade que não conhecemos. Sua importância vem do fato de só necessitar o conhecimento da esperança e da variância da distribuição. Ela é utilizada nas provas do ACGS.

Por outro lado, devemos levar em conta que ela pode não definir aproximações muito exatas de probabilidade. Vamos citar dois exemplos desse fato obtidos do livro de Ross [Ross 2006, Example 8.8b]. Em um exemplo para distribuição uniforme, a aproximação foi de $52\%$, enquanto que a probabilidade real era de $20\%$. Já em um segundo exemplo, para distribuição normal, a diferença foi de $25\%$ para $4{,}5\%$.

 ©2012 SBC — Soc. Bras. de Computação

## 3. Métodos e Implementação

Trabalhamos com dois criptossistemas, o RSA e o DHCE. Nesta seção, vamos apresentar métodos que se utilizam de um oráculo para tentar inverter esses criptossistemas. Ainda, daremos detalhes das implementações de tais métodos.

### 3.1. Algoritmo para o RSA

Suponha a existência de um oráculo $\mathcal{O}_N$ com vantagem $\varepsilon$ em descobrir o LSB do RSA com módulo $N$. Onde a vantagem $0 < \varepsilon \leq \frac{1}{2}$ é pequena, porém não negligenciável. Vamos mostrar como o ACGS faz uso de $\mathcal{O}_N$ para inverter o RSA no Algoritmo 3.1.

O ACGS computa dois valores aleatórios $[ax]_N$ e $[bx]_N$, assegura que $[ax]_N$ seja ímpar e depois calcula o $\mathrm{MDC}$ com ajuda do Brent Kung GCD [Brent and Kung 1984], por brevidade chamaremos esse MDC binário de BKGCD.

Ao chegar na Parte 3 do algoritmo, a variável $a$ contém um valor cujo produto com $x$ é o MDC de $[ax]_N$ e $[bx]_N$. É neste momento que esperamos que o $[ax]_N$ e $[bx]_N$ escolhidos sejam coprimos, pois na verificação da Linha 27 teremos $\mathrm{RSA}(ax) = \pm 1$ e assim conseguimos recuperar $x$ através do cálculo do inverso multiplicativo $[\pm a^{-1}]_N$. Caso não sejam coprimos, o algoritmo volta para Linha 2 e sorteia novamente $a$ e $b$ na esperança que esses valores resultem em $[ax]_N$ e $[bx]_N$ coprimos.

Por um famoso teorema de Dirichlet [Knuth 1981, Sec.4.5.2 Theo.D p.324], a probabilidade de dois inteiros aleatórios no intervalo $[-K, K]$ serem relativamente primos converge para $\frac{6}{\pi^2}$ conforme $K$ tende a $\infty$. Isso é algo em torno de $60{,}8\%$. Assim, com duas tentativas em $a$ e $b$ já é esperado encontrar coprimos em $[ax]_N$, $[bx]_N$.

Outra condição que faz o algoritmo retornar à Linha 2 se refere à variável $limiteGCD$, que chamaremos limitante do BKGCD. Esse limitante é para o caso quando a função PAR é falha, assim o algoritmo não ficará executando indefinidamente. A função PAR é uma função de paridade, por exemplo, dado $\mathrm{PAR}(b, y)$, ela retorna a paridade de $[bx]_N$. Ela é apresentada no Algoritmo 3.2.

A função PAR faz uma quantidade $m$ de amostras entorno da paridade de $dx$, denominadas **medidas-dx**. São somados os votos para a resposta, par ou ímpar, nas variáveis $contador_0$ e $contador_1$, respectivamente. O resultado é a maioria dos votos.

Cada voto de paridade é obtido verificando se $\mathrm{LSB}(rx) = \mathrm{LSB}(rx + dx)$. A igualdade se mantém se, e somente se, $dx$ for par, isso porque um número par não altera a paridade de outro número na operação de adição.

O problema aqui é se na adição $rx + dx$ ocorrer uma redução modular (também chamada *overlapping* ou *wraparound*). Como $N$ é ímpar, o voto é invertido. Por isso que desejamos que na escolha de $a$ e $b$, $[ax]_N$ e $[bx]_N$ sejam "pequenos". Aqui, $[ax]_N$ pequeno quer dizer que $\mathrm{abs}_N(ax) < \frac{\varepsilon}{2}N$, onde:

$$\mathrm{abs}_N(x) \stackrel{def}{=} \begin{cases} [x]_N & \text{se } [x]_N < N/2 \\ N - [x]_N & \text{caso contrário} \end{cases}$$

Sendo assim, teremos $[dx]_N$ pequeno e na adição $rx + dx$ a probabilidade de redução modular ocorrer também será pequena ($\frac{\varepsilon}{2}$). Note que conforme a função BKGCD evolui, os valores $[ax]_N$ e $[bx]_N$ intermediários continuam pequenos.

 ©2012 SBC — Soc. Bras. de Computação

**Algoritmo 3.1**: Inversor ACGS para o RSA.

**Entrada**: $y = \mathrm{RSA}(x)$, módulo $N$ e o expoente de encriptação $e$.
**Saída**: a mensagem original $x$.

1 **início**
  `// Parte 1: Aleatorização`
2  Escolha de maneira uniforme e independente $a, b \in \mathbb{Z}_N$;
  `// Parte 2: Brent-Kung GCD de` $[ax]_N$ `e` $[bx]_N$
3  $\alpha \leftarrow n$;
4  $\beta \leftarrow n$;
5  limiteGCD $\leftarrow 0$;
6  **enquanto** $\mathrm{PAR}(a, y) = 0$ **faça**
7  $a \leftarrow [a/2]_N$;
8  $\alpha \leftarrow \alpha - 1$;
9  limiteGCD $\leftarrow$ limiteGCD $+ 1$;
10  **se** limiteGCD $> 6n + 3$ **então** `VolteParaLinha(2);`
11  **repita**
12  **enquanto** $\mathrm{PAR}(b, y) = 0$ **faça**
13  $b \leftarrow [b/2]_N$;
14  $\beta \leftarrow \beta - 1$;
15  limiteGCD $\leftarrow$ limiteGCD $+ 1$;
16  **se** limiteGCD $> 6n + 3$ **então** `VolteParaLinha(2);`
17  **se** $\beta \leq \alpha$ **então**
18  `Troca(a,b);`
19  `Troca(`$\alpha$`,`$\beta$`);`
20  **se** $\mathrm{PAR}(\frac{a+b}{2}, y) = 0$ **então**
21  $b \leftarrow [(a + b)/2]_N$;
22  **senão**
23  $b \leftarrow [(a - b)/2]_N$;
24  limiteGCD $\leftarrow$ limiteGCD $+ 1$;
25  **se** limiteGCD $> 6n + 3$ **então** `VolteParaLinha(2);`
26  **até** $b = 0$;
  `// Parte 3: Inversão`
27  **se** $\mathrm{RSA}(ax) \neq \pm 1$ **então** `VolteParaLinha(2);`
28  $c \leftarrow [\pm a^{-1}]_N$;
29  **retorna** $c$; `/* c é a mensagem original x */`
30 **fim**

 ©2012 SBC — Soc. Bras. de Computação

**Algoritmo 3.2**: Função de paridade PAR.

**Entrada**: $y = \mathrm{RSA}(x)$, o multiplicador $d$.
**Saída**: o $\mathrm{LSB}(\mathrm{abs}_N(dx))$.

1 **início**
2 contador$_0 \leftarrow 0$;
3 contador$_1 \leftarrow 0$;
4 **para** $i \leftarrow 1$ **até** $m$ **faça**
5 escolha de maneira aleatória $r_i \in \mathbb{Z}_N$;
6 **se** $\mathcal{O}_N(\mathrm{RSA}(r_i x)) = \mathcal{O}_N(\mathrm{RSA}(r_i x + dx))$ **então**
7 contador$_0 \leftarrow$ contador$_0 + 1$;
8 **senão**
9 contador$_1 \leftarrow$ contador$_1 + 1$;
10 **se** contador$_0 >$ contador$_1$ **então**
11 **retorna** *0*;
12 **senão** **retorna** *1*;
13 **fim**

Com esse esquema de votos, conseguimos construir uma função de paridade quase perfeita, desde que a quantidade de medidas $m$ seja grande o suficiente. Isso é garantido pela lei dos grandes números.

Na Linha 6 do Algoritmo 3.2 são feitas duas consultas ao oráculos. Isso gera um problema conhecido como **duplicação de erro** [Ben-Or et al. 1983]. Para contorná-lo, o ACGS usa a função PAR de modo que o oráculo só é consultado do lado direito da igualdade ($rx + dx$). Quanto ao lado esquerdo ($rx$), o LSB de $rx$ já é conhecido. Isso é possível dividindo o conjunto dos inteiros módulo N em intervalos e aplicado uma técnica de suposição sobre eles.

Note que das suposições feitas em [Alexi et al. 1988], temos ao todo $2 \cdot 4\varepsilon^{-1} \cdot 2 \cdot 4m\varepsilon^{-1} = 2^6 m\varepsilon^{-2}$ possibilidades. Rodamos o ACGS com cada uma dessas alternativas. Em apenas uma delas estaremos computando corretamente $\mathrm{LSB}(r_i x)$ e chamaremos essa instância de **alternativa correta**. Então, na alternativa correta o ACGS recupera a mensagem original.

Os autores em [Alexi et al. 1988] calcularam utilizando Desigualdade de Chebyshev a probabilidade de erro da função PAR igual a $\Pr(\mathrm{PAR\ errar}) = \frac{4}{m\varepsilon^2}$. Então, é preciso escolher uma amostragem $m$ suficientemente grande para se ter uma função de paridade quase perfeita. Foi escolhida $m \stackrel{def}{=} 64n\varepsilon^{-2}$, assim $\Pr(\mathrm{PAR\ errar}) \leq \frac{1}{16n}$.

Com probabilidade de erro $\frac{1}{16n}$, a função PAR, em média, consegue passar sem erro em todas as requisições da função BKGCD (no máximo $limiteGCD = 6n + 3$).

De acordo com a nossa discussão sobre a Desigualdade de Chebyshev na Seção 2, acreditamos que a probabilidade de erro da função PAR é superestimada aqui. Veremos também, empiricamente, que na média o limitante do BKGCD pode ser menor do que $6n + 3$, pelo menos metade.

O tempo de execução do ACGS corresponde a $\varepsilon^{-2}$ tentativas para se obter $ax$ e

 ©2012 SBC — Soc. Bras. de Computação

$bx$ pequenos, mais duas tentativas para que sejam coprimos $\left(\frac{\pi^2}{6}\right)$, mais a execução das $2^6 m\varepsilon^{-2}$ alternativas, $6n+3$ chamadas à função PAR e $m$ medidas-dx. Escrevendo tudo:

$$\varepsilon^{-2}\cdot 2\cdot 2^6 m\varepsilon^{-2}\cdot(6n+3)\cdot m \approx 3\cdot 2^8\varepsilon^{-4}nm^2 \approx 3\cdot 2^8\varepsilon^{-4}n\left(\frac{64n}{\varepsilon^2}\right)^2 \approx 3\cdot 2^{20}\varepsilon^{-8}n^3 = O(\varepsilon^{-8}n^3)$$

### 3.2. Algoritmo para o DHCE

O algoritmo BS faz uma adaptação do algoritmo ACGS para uso no DHCE. Considere um grupo cíclico de curva prima $E(\mathbb{F}_p)$ com um gerador $G$ de ordem prima $q$ e as chaves privadas de Alice e Beto $a, b \in [1, q-1]$, definimos o segredo da função Diffie-Hellman $DH_{E,G}(aG, bG) = abG$ como a coordenada $x$ do ponto $abG$. Nessa seção, supomos a existência de um oráculo $\mathcal{O}_p$ para $[abG]_x$, tal que $\mathrm{Adv}^X_{E,G}(\mathcal{O}_p) = \varepsilon$.

Para um $\lambda \in \mathbb{F}_p^*$, $\phi_\lambda(E)$ define a curva elíptica $Y^2 = X^3 + A\lambda^4 X + B\lambda^6$, denominada *twisted elliptic curve*. É fácil identificar que esses *twists* são na verdade isomorfismos. Assim, para pontos $Q, R, S, \in E$, com $Q = (x, y)$, temos $(Q)_\lambda = Q_\lambda = (\lambda^2 x, \lambda^3 y) \in \phi_\lambda(E)$ e para $Q + R = S$, temos $Q_\lambda + R_\lambda = S_\lambda$ (propriedade homomórfica). Ainda, com tais *twists*, definimos para uma curva inicial $E_0$, uma família de curvas isomorfas $\phi_\lambda(E_0)_{\lambda\in\mathbb{F}_p^*}$.

Seja $K_{ab}$ o ponto combinado no DHCE sobre a curva $E_0$ com as chaves públicas $PK_a$ e $PK_b$. A coordenada $x$ de $K_{ab}$ é a chave secreta. Da mesma forma como no ACGS, sorteamos $a'$ e $b'$ (notamos $a/b$ linha para não confundir com as chaves privadas $a/b$) esperando que $[a'x]_p$ e $[b'x]_p$ sejam pequenos. Similarmente ao BKGCD, $a'$ e $b'$ geram valores $d$, aqui chamamos $\lambda$, e uma consulta para a paridade é feita com a chamada de $\mathrm{PAR}(\lambda, PK_a, PK_b)$. Porém, a diferença fica por conta da requisição ao LSB. A função PAR é perguntada pelo LSB de $\lambda^2 x$, pois $(K_{ab})_\lambda = (\lambda^2 x, \lambda^3 y)$. Então, a requisição é feita para um ponto que fica em outra curva, diferente de $E_0$.

Os autores em [Boneh and Shparlinski 2001] consideram um oráculo $\mathcal{O}_p$ com vantagem $\varepsilon$ em predizer o LSB na curva $E_0$. Isso quer dizer que $\mathcal{O}_p$ não tem vantagem em todas as curvas na família $\phi_\lambda(E_0)_{\lambda\in\mathbb{F}_p^*}$. Por definição, é considerado que $\mathcal{O}_p$ mantém essa vantagem para pelo menos uma fração $\delta$ das curvas na família de isomorfismo.

O problema é que o algoritmo BS precisa consultar a paridade para vários valores de $\lambda$. Então, é construído um novo oráculo $\mathcal{B}_p$ cuja probabilidade de sucesso dentro ou fora da fração $\delta$ das curvas seja conhecida.

A seguir, resumidamente apresentamos os passos do algoritmo BS:

1. com entrada $\langle E, G, PK_a, PK_b\rangle$, onde $PK_a = aG$ e $PK_b = bG$ e $G$ de ordem prima $q$, queremos calcular o ponto $C = abG$;
2. como $a$ e $b$ são fixos e desconhecidos, aleatorizamos o processo definindo $PK_{ra} = a_r aG$ e $PK_{rb} = b_r bG$, para $a_r, b_r \in [1, q-1]$, e esperamos que os valores $a_r a$ e $b_r b$ levem a um caso onde o oráculo $\mathcal{B}_p$ tenha vantagem não negligenciável;
3. sendo $DH_{E,G}(PK_{ra}, PK_{rb}) = D$, basta calcular $D$ para obtermos $C$, pois $C = c_r \cdot D$, onde $c_r \equiv (a_r b_r)^{-1} \bmod q$;
4. agora, executamos um algoritmo similar ao ACGS com o oráculo $\mathcal{B}_p$;
5. a fim de assegurar que encontremos os valores $a_r a$ e $b_r b$ desejados pode ser necessário repetir todo o processo $\frac{8}{\varepsilon\delta}$ vezes.

 ©2012 SBC — Soc. Bras. de Computação

Ao final do algoritmo BS, teremos uma lista de candidatos $C$ para o ponto $abG$. Porém, ao contrário do ACGS, não temos condições de identificar automaticamente a alternativa correta, o que nos obriga a executar todas as alternativas e apenas ao final tentar identificar a resposta correta. A identificação é possível com um algoritmo devido a Shoup [Shoup 1997, Theo.7].

### 3.3. Biblioteca criptográfica Relic

Para implementar os algoritmos ACGS e BS, utilizamos a linguagem de programação C e a biblioteca criptográfica ***Relic Toolkit*** versão 0.3.0 [Aranha and Gouvêa ].

Utilizamos as curvas SECG_P160 e NIST_P224 da biblioteca para o DHCE. Para o RSA, utilizamos módulos de 1024, 2048 e 5000 bits, além de um módulo teórico de 128 bits para registrar tempos mais demorados. Escolhemos 1024 e 5000 bits devido a [Fischlin and Schnorr 2000] e 2048 bits por ser a recomendação atual do NIST[1]. E por ter equivalência no nível de segurança a 1024 e 2048 bits do RSA, escolhemos 160 e 224 bits para curvas elípticas.

A respeito da implementação dos oráculos, a resposta para a consulta no DHCE é imediata, com a execução de apenas alguns poucos cálculos. Já no RSA, é feita uma decriptação para cada consulta, resultando em um tempo para a resposta bem maior.

O gerador pseudoaleatório construído dentro da Relic é o FIPS 186-2 baseado em SHA1. A Relic já possui ferramentas de cronômetro (*benchmark*) para registrar os tempos de execução, nós utilizamos a versão mais precisa desses cronômetros (HPROC). Para simulação dos algoritmos e coleta dos tempos de execução, utilizamos um computador com processador Intel Core 2 Duo T5450 de 1,66 Ghz com 2 GB de memória RAM.

Os códigos fontes das implementações estão disponíveis no site do Laboratório de Segurança de Dados do IME-USP[2].

## 4. Resultados

Nesta seção apresentamos os resultados obtidos das implementações. Apresentamos os resultados para o RSA com o ACGS e os resultados para o DHCE com o BS.

### 4.1. Resultados para o algoritmo ACGS

O sucesso da execução do algoritmo ACGS dependia que da escolha de $a$ e $b$, resultasse em $ax$ e $bx$ pequenos ($abs_N(ax) < \frac{\varepsilon N}{2}$), e além disso, que fossem primos entre si. Por agilidade e a fim de dar mais regularidade aos tempos apresentados, implementamos o algoritmo de modo a sempre garantir essas condições. Depois, para a inferência de tempos, adicionamos de volta o tempo médio necessário para satisfazê-las. Esse método é consistente, porque que as probabilidades envolvidas são bem conhecidas.

Então, tendo essas condições iniciais satisfeitas e rodando o algoritmo sempre na alternativa correta, calculamos os tempos do algoritmo ACGS. Entretanto, mesmo com esses cortes, rodar e observar o tempo do algoritmo para um RSA de 1024 bits ainda era muito demorado, ainda mais para vantagens pequenas, $\varepsilon < 0{,}05$. Decidimos então coletar os dados de um RSA teórico de 128 bits para calcular esse tempo na alternativa correta.

[1] `http://csrc.nist.gov/groups/ST/toolkit/key_management.html`
[2] `http://lsd.ime.usp.br/`

 ©2012 SBC — Soc. Bras. de Computação

Apresentamos a Figura 1 com os tempos para um RSA de 128 bits em relação ao número $m$ de amostras. Observe que de acordo com o tamanho das amostras, a cada vez que o tamanho da amostra dobra, o tempo consumido de processamento também dobra.

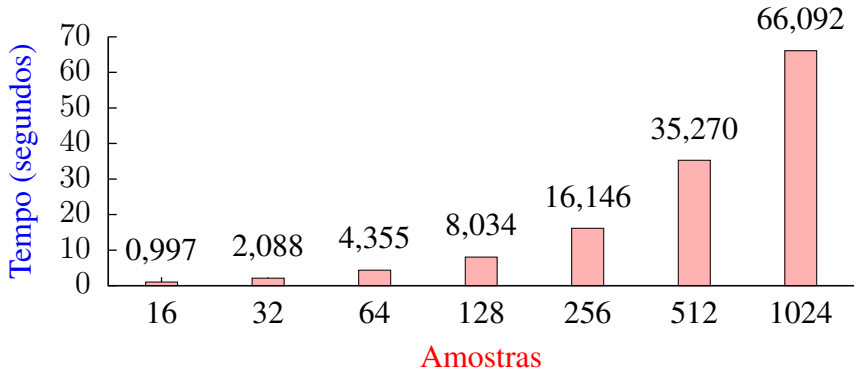



**Figura 1. Tempo consumido em relação ao número de amostras.**

As operações aritméticas realizadas dentro do algoritmo têm tempo variável em relação ao número de bits dos números calculados. O tempo da exponenciação com números de 256 bits é maior do que com números de 128 bits, o tempo da exponenciação com números de 512 bits é maior do que com números de 256 bits, e assim por diante. Na Figura 2 temos o tempo gasto a cada chamada à função PAR em relação ao tamanho dos operandos; a função PAR faz 16 consultas ao oráculo. Note que o tempo para 5000 bits foi apenas indicado por uma flecha para não prejudicar as proporções do gráfico.

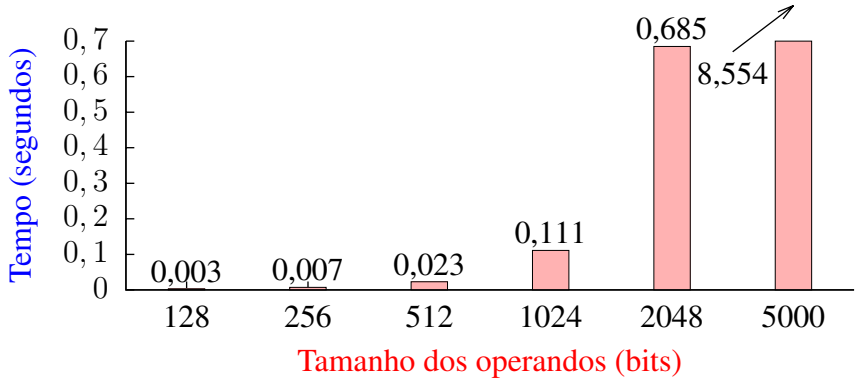



**Figura 2. Tempo da função PAR no ACGS em relação aos operandos ($m = 16$).**

O número $m$ de amostragens é um parâmetro crítico do sistema. No modelo original do ACGS ele é por definição $m \stackrel{def}{=} \frac{64n}{\varepsilon^2}$. Dessa forma, o valor de $m$ depende apenas do tamanho do módulo $N$ em bits e da vantagem $\varepsilon$ do oráculo. Na Tabela 1 apresentamos alguns exemplos de amostragem em relação ao par $n, \varepsilon$.

Procuramos tentar reduzir a amostragem. Percebemos que para valores menores do que a do modelo original, o algoritmo ACGS ainda obtinha sucesso em alguns casos. Após tentativas, obtivemos empiricamente a seguinte definição do parâmetro $m \stackrel{def}{=} \frac{4}{\varepsilon^2}$.

Na Figura 3, apresentamos exemplos para o novo padrão de amostragem. Note que no gráfico incluímos apenas a vantagem, já que no novo padrão o tamanho do módulo

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Amostragem em relação aos bits e à vantagem do oráculo.**

| | 128 bits | 1024 bits | 5000 bits |
|---|---|---|---|
| 0,4 | 51.200 | 409.600 | 2.000.000 |
| 0,3 | 91.023 | 728.178 | 3.555.556 |
| 0,2 | 204.800 | 1.638.400 | 8.000.000 |
| 0,1 | 819.200 | 6.553.600 | 32.000.000 |
| 0,05 | 3.276.800 | 26.214.400 | 128.000.000 |
| 0,01 | 81.920.000 | 655.360.000 | 3.200.000.000 |

não influencia o número de amostras. Faça uma comparação com os valores da Tabela 1.

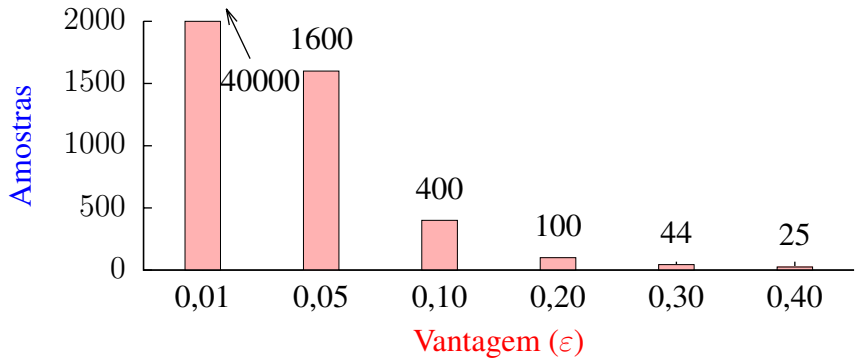



**Figura 3. Número de amostras em relação à vantagem do oráculo.**

Utilizando então essa nova metodologia, executamos simulações do ACGS para um RSA de 128 bits. O algoritmo obteve sucesso em sucessivas execuções até parar na primeira vez que falhou e não conseguiu recuperar a mensagem original. Observamos os seguintes resultados:

- $\varepsilon = 0{,}4$: rodou 66 vezes e parou;
- $\varepsilon = 0{,}3$: rodou 30 vezes e parou;
- $\varepsilon = 0{,}2$: rodou 53 vezes e parou;
- $\varepsilon = 0{,}1$: rodou 44 vezes e parou;
- $\varepsilon = 0{,}05$: rodou 29 vezes e parou;
- $\varepsilon = 0{,}01$: rodou 2 vezes e continuou.

Para o caso de $\varepsilon = 0{,}01$, ele rodou 2 vezes com sucesso, mas tivemos que abortar já que cada teste levou cerca de 40 minutos para ser executado. Para $\varepsilon = 0{,}2$ e $\varepsilon = 0{,}05$, com $m = 100$ e $m = 1600$ respectivamente, tivemos alguns problemas, os testes falhavam muito cedo. Porém, com apenas a adição de uma unidade na amostragem ($m = 101$ e $m = 1601$) tivemos resultados mais consistentes.

Observamos um fato particular ao rodar o ACGS na alternativa correta. Para um RSA de 128 bits, dentro de 200 testes, a função BKGCD fez de 324 a 368 chamadas à função PAR, ou seja, algo entre $2{,}53n$ e $2{,}88n$. Não conseguimos observar uma execução na alternativa correta com mais de $3n$ chamadas à função PAR.

O limitante da função BKGCD é definido por $limiteGCD = 6n + 3$, mas de acordo com nossas observações esse valor poderia ser alterado para $limiteGCD = 3n$.

 ©2012 SBC — Soc. Bras. de Computação

Note que essa redução afeta não apenas a alternativa correta, mas também todas as demais alternativas, oferecendo uma redução pela metade no tempo do ACGS.

Assim, o novo tempo total inclui os tempos da condição inicial ($\varepsilon^{-2} \cdot 2$), o tempo de cada uma das alternativas ($2^6 m\varepsilon^{-2}$), o tempo da função BKGCD ($3n$), o tempo da função PAR ($m$) e o tempo do oráculo (1, consideramos constante): $(\varepsilon^{-2})(2)(2^6 m\varepsilon^{-2})(3n)(m)(1) = 3 \cdot 2^{11}\varepsilon^{-8}n$.

A Tabela 2 traz essa nova estimativa de complexidade junto com a original. Adicionamos também o tempo do algoritmo de solução para o PFI ([Buhler et al. 1993]) como o utilizado em [Fischlin and Schnorr 2000]. Perceba que agora o ACGS é mais rápido que o PFI.

**Tabela 2. Comparação entre as complexidades existentes.**

| | Estimativa | 1024 bits | 2048 bits | 5000 bits |
|---|---|---|---|---|
| PFI | $\exp(1{,}9(\ln N)^{\frac{1}{3}}(\ln\ln N)^{\frac{2}{3}})$ | $6{,}409 \cdot 10^{25}$ | $5{,}817 \cdot 10^{34}$ | $3{,}755 \cdot 10^{50}$ |
| ACGS orig. | $3 \cdot 2^{20}\varepsilon^{-8}n^3$ | $3{,}378 \cdot 10^{31}$ | $2{,}702 \cdot 10^{32}$ | $3{,}932 \cdot 10^{33}$ |
| ACGS novo | $3 \cdot 2^{11}\varepsilon^{-8}n$ | $6{,}292 \cdot 10^{22}$ | $1{,}258 \cdot 10^{23}$ | $3{,}072 \cdot 10^{23}$ |

Agora podemos utilizar os tempos da função PAR na Figura 2 para obter estimativas de tempo. Nessa figura, temos o uso de apenas 16 amostras, mas como vamos utilizar $\varepsilon = 0{,}1$, teremos $m = 400$. Assim, seja **tPAR** o tempo da função PAR na Figura 2, o cálculo proporcional se dá pela fórmula $\frac{400}{16}\text{tPAR} \cdot 3n \cdot 2^6 m\varepsilon^{-2} \cdot \varepsilon^{-2} \cdot 2 = 3{,}84 \cdot 10^{10} \cdot \text{tPAR} \cdot n$.

A Tabela 3 traz os tempos estimados para se tentar inverter o RSA. Temos uma coluna para o tempo na alternativa correta e uma coluna para o tempo total do algoritmo, além do tempo **tPAR** da Figura 2.

**Tabela 3. Tempos estimados para execução do algoritmo ACGS.**

| n (bits) | tPAR (s) | Alt. correta | Tempo total |
|---|---|---|---|
| 1024 | 0,111 | 19,7 dias | $138{,}4 \cdot 10^3$ anos |
| 2048 | 0,685 | 243,5 dias | $1{,}7 \cdot 10^6$ anos |
| 5000 | 8,554 | 20,3 anos | $52{,}1 \cdot 10^6$ anos |

### 4.2. Resultados para o algoritmo BS

Vamos seguir a mesma linha de discussão do RSA: com a nova amostragem definida e com o limitante para o BKGCD igual a $3 \lg p$. Apenas uma execução do algoritmo BS é considerada, assim, não entra aqui as execuções adicionais requeridas pelo algoritmo de Shoup.

Quanto à função BKGCD, não há problemas com o limitante igual a $3 \lg p$. Entretanto, quanto à amostragem, deve-se levar em conta que vamos utilizar dois algoritmos, o oráculo $\mathcal{O}_p$ e o oráculo $\mathcal{B}_p$. O oráculo $\mathcal{B}_p$ é regido pelo parâmetro $\delta$ que define a proporção de curvas onde o oráculo $\mathcal{O}_p$ possui vantagem $\varepsilon$. O caso ideal é quando $\delta = 1$, o que define o oráculo $\mathcal{O}_p$ como aplicável a todas as curvas da família.

No caso onde $\delta = 1$ temos um algoritmo BS semelhante ao ACGS. Com base nesse valor podemos obter os tempos da função PAR para operandos de 160 bits e 224 bits em um DHCE conforme a Figura 4.

 ©2012 SBC — Soc. Bras. de Computação

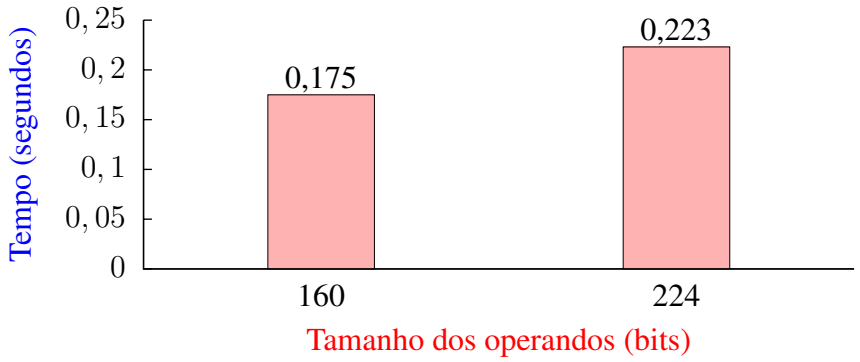



**Figura 4. Tempo da função PAR no DHCE em relação aos operandos ($m = 40000$).**

Escolhemos a amostragem $m \stackrel{def}{=} \frac{8}{\varepsilon^2}$ e executamos simulações do BS para um DHCE de 160 bits. O algoritmo obteve sucesso em sucessivas execuções até errar ou ser abortado pela extensão do tempo. Temos os seguintes resultados desses testes:

- $\varepsilon = 0{,}4$: rodou 368 vezes e parou;
- $\varepsilon = 0{,}3$: rodou 247 vezes e parou;
- $\varepsilon = 0{,}2$: rodou 183 vezes e parou;
- $\varepsilon = 0{,}1$: rodou 862 vezes e continuou;
- $\varepsilon = 0{,}05$: rodou 35 vezes e continuou;
- $\varepsilon = 0{,}01$: rodou 1 vez e continuou.

Agora vamos inferir os tempos para $\varepsilon = 0{,}1$ e $m = 800$, com os tempos **tPAR** da Figura 4. Temos o tempo total $\frac{800}{40000}\text{tPAR} \cdot 3 \lg p \cdot 2^6 m \varepsilon^{-2} \cdot \varepsilon^{-2} \cdot 2 = 6{,}144 \cdot 10^7 \cdot \text{tPAR} \cdot \lg p$. Dessa equação montamos a Tabela 4 com os tempos estimados para o BS.

**Tabela 4. Tempos estimados para execução do algoritmo BS.**

| $\lg p$ (bits) | tPAR (s) | Alt. correta | Tempo total |
|---|---|---|---|
| 160 | 0,175 | 5,6 minutos | 54,6 anos |
| 224 | 0,223 | 9,9 minutos | 97,3 anos |

Agora analisaremos o pior caso, onde $\delta < 1$ e próximo de zero. Fixamos $\delta = 0{,}1$, de modo que em apenas $10\%$ das curvas o oráculo $\mathcal{O}_p$ tem vantagem dentro da família de isomorfismo. Então usamos o oráculo $\mathcal{B}_p$ que implica em uma nova amostragem $m2 \stackrel{def}{=} \frac{8}{\varepsilon\delta^2}$ relacionada com os acessos ao oráculo $\mathcal{O}_p$ pelo oráculo $\mathcal{B}_p$.

Essa nova amostragem é obtida utilizando a mesma metodologia anterior. Diretamente, apresentamos a Tabela 5 com os tempos estimados para se tentar descobrir a chave combinada pelo DHCE com uso do oráculo $\mathcal{B}_p$.

**Tabela 5. Tempos estimados para execução do algoritmo BS com o oráculo $\mathcal{B}_p$.**

| $\lg p$ (bits) | Alt. correta | Tempo total |
|---|---|---|
| 160 | 15,6 anos | $80 \cdot 10^6$ anos |
| 224 | 44,7 anos | $229 \cdot 10^6$ anos |

 ©2012 SBC — Soc. Bras. de Computação

## 5. Conclusões

Com base nos algoritmos propostos, conseguimos construir implementações dos artigos com êxito na recuperação dos segredos. Alcançamos sucesso com um nível de convergência bem maior que o esperado. Nas implementações, fizemos o uso dos números pseudoaleatórios gerados pela Relic.

Nossos testes conseguiram, de maneira experimental, ajustar os parâmetros críticos do sistema para diminuir significativamente o tempo de execução. E com os tempos diminuídos, os métodos ainda alcançavam taxas de sucesso consideradas aceitáveis. Isto é garantido pela lei dos grandes números.

O resultados provêem medidas de quão rápido é inverter os criptossistemas citados a partir de um mínimo comprometimento do LSB. Em particular, no caso do ACGS com RSA-1024 tivemos um avanço significativo no tempo de inversão a ponto de ficar abaixo do tempo de solução para o PFI, anteriormente não alcançado.

**Sugestões para trabalhos futuros.** Em [Alexi et al. 1988] existe uma versão do ACGS para o algoritmo Rabin que poderia ser também implementada para base de comparação. Uma ideia para prova de conceito de nossos métodos é criar uma implementação falha (revelando o LSB através de um ataque de canal secundário) do RSA e então adaptá-la como oráculo para o ACGS. Um trabalho como esse é capaz de dar dimensões práticas da aplicabilidade do método.

## Referências

[Alexi et al. 1988] Alexi, W., Chor, B., Goldreich, O., and Schnorr, C.-P. (1988). RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194–209.

[Aranha and Gouvêa ] Aranha, D. F. and Gouvêa, C. P. L. RELIC is an Efficient LIbrary for Cryptography. `http://code.google.com/p/relic-toolkit/`.

[Ben-Or et al. 1983] Ben-Or, M., Chor, B., and Shamir, A. (1983). On the cryptographic security of single RSA bits. In *ACM Symposium on Theory of Computing (STOC '83)*, pages 421–430, Baltimore, USA. ACM Press.

[Blum et al. 1986] Blum, Blum, and Shub (1986). A simple unpredictable pseudo-random number generator. *SICOMP: SIAM Journal on Computing*, 15.

[Blum and Micali 1984] Blum, M. and Micali, S. (1984). How to generate cryptographically strong sequence of pseudo-random bits. *SIAM Journal Computing*, 13:850–864.

[Boneh and Shparlinski 2001] Boneh, D. and Shparlinski, I. (2001). On the unpredictability of bits of the elliptic curve Diffie-Hellman scheme. In Kilian, J., editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 201–212. Springer.

[Boneh and Venkatesan 1996] Boneh, D. and Venkatesan, R. (1996). Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *CRYPTO '96*, volume 1109 of *LNCS*, pages 129–142. IACR, Springer, Berlin.

[Brent and Kung 1984] Brent and Kung (1984). Systolic VLSI arrays for polynomial GCD computation. *IEEE Transactions on Computers*, 33.

 ©2012 SBC — Soc. Bras. de Computação

[Buhler et al. 1993] Buhler, J. P., Lenstra, H. W., and Pomerance, C. (1993). Factoring integers with the number field sieve. In *The development of the number field sieve*, volume 1554 of *LNM*, pages 50–94. Springer-Verlag, Berlin.

[Chevalier et al. 2009] Chevalier, C., Fouque, P.-A., Pointcheval, D., and Zimmer, S. (2009). Optimal randomness extraction from a Diffie-Hellman element. In Joux, A., editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 572–589. Springer-Verlag.

[Diffie and Hellman 1976] Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

[Fischlin and Schnorr 2000] Fischlin, R. and Schnorr, C.-P. (2000). Stronger security proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244.

[Goldwasser et al. 1982] Goldwasser, S., Micali, S., and Tong, P. (1982). Why and how to establish a private code on a public network (extended abstract). In *FOCS*, pages 134–144, Chicago, Illinois. IEEE.

[Hofheinz and Kiltz 2009] Hofheinz, D. and Kiltz, E. (2009). Practical chosen ciphertext secure encryption from factoring. In Joux, A., editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer.

[Jetchev and Venkatesan 2008] Jetchev, D. and Venkatesan, R. (2008). Bits security of the elliptic curve Diffie-Hellman secret keys. In Wagner, D., editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 75–92. Springer.

[Knuth 1981] Knuth, D. E. (1981). *The Art of Computer Programming*, volume 2, Seminumerical Algorithms. Addison-Wesley, Reading, MA, 2 edition.

[Menezes et al. 1996] Menezes, A. J., Vanstone, S. A., and Oorschot, P. C. V. (1996). *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, USA.

[Rabin 1979] Rabin, M. (1979). Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science.

[Rivest et al. 1978] Rivest, R. L., Shamir, A., and Adleman, L. M. (1978). A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126.

[Roh and Hahn 2010] Roh, D. and Hahn, S. G. (2010). On the bit security of the weak Diffie–Hellman problem. *Information Processing Letters*, 110:799–802.

[Ross 2006] Ross, S. M. (2006). *A First Course in Probability*. Prentice Hall, New Jersey, 7 edition.

[Shoup 1997] Shoup, V. (1997). Lower bounds for discrete logarithms and related problems. In Fumy, W., editor, *EUROCRYPT '97*, volume 1233 of *LNCS*, pages 256–266, Berlin Germany. Springer-Verlag.

[Stein 1967] Stein, J. (1967). Computational problems associated with Racah algebra. *Journal of Computational Physics*, 1(3):397 – 405.

 ©2012 SBC — Soc. Bras. de Computação

# Universally Composable Committed Oblivious Transfer With A Trusted Initializer

**Adriana C. B. Pinto [1], Bernardo Machado David [1] Jeroen van de Graaf[2], Anderson C. A. Nascimento[1]**

[1]Departmento de Engenharia Elétrica – Universidade de Brasília (UNB)
Campus Universitário Darcy Ribeiro – 70910-900 – Brasília – DF – Brasil

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos 6627 – 31270-901 – Belo Horizonte (MG) – Brasil

adrianacbp@unb.br, bernardo.david@aluno.unb.br

jvdg@dcc.ufmg.br, andclay@ene.unb.br

***Abstract.*** *Committed Oblivious Transfer (COT) is a two-party primitive that combines one-out-of-two oblivious transfer with bit commitment. In the beginning of COT, a sender is committed to bits $b_0$,$b_1$ and a receiver to a choice bit $c$. In the end, the receiver is committed to $b_c$ without learning anything about $b_{1-c}$, while the sender learns nothing about $c$. This primitive implies secure multi-party computation assuming that a broadcast channel is available. In this paper, we introduce the first universally composable unconditionally secure committed oblivious transfer protocol based on a Trusted Initializer (TI), which pre-distributes data to the parties. Our protocol builds on simple bit commitment and oblivious transfer protocols, using XOR commitments to prove simple relations in zero-knowledge. Besides providing very high security guarantees, our protocols are significantly simpler and more efficient than previous results, since they rely on pre-computed operations distributed by the TI.*

**Keywords.** Committed Oblivious Transfer, Commodity based cryptography, Unconditional Security, Universal Composability.

## 1. Introduction

Committed Oblivious Transfer (COT) was introduced by Crépeau, van de Graaf and Tapp in [Crépeau et al. 1995] and combines one-out-of-two oblivious transfer and bit commitment. In a nutshell, COT is a variation of Oblivious Transfer where the parties are committed to their inputs and end up committed to their outputs. Basically, this added property enables the parties to verify input correctness, since they can't change their inputs after the commitments are made. It was shown that this primitive implies efficient secure multi-party computation for dishonest majorities in the presence of a broadcast channel [Crépeau et al. 1995], while plain oblivious transfer is only able to achieve secure two-party computation without extra assumptions [Kilian 1988][Goldreich et al. 1987].

Specifically, a COT is defined as a two-party primitive where the sender's inputs are commitments to bits $b_0$,$b_1$ and the receiver's input is a commitment to a choice bit $c$.

 ©2012 SBC — Soc. Bras. de Computação

In the end of the protocol, the receiver is committed to $b_c$ but doesn't learn $b_{1-c}$, while the sender learns nothing about $c$ . Security for the sender consists in the receiver learning nothing about $b_{1-c}$ and security for the receiver consists in the sender learning nothing about $c$.

The original COT protocol proposed in [Crépeau et al. 1995] was proven secure in the stand-alone model, meaning that its security is only guaranteed in a very simple setting where only one copy of the protocol is executed each time. While this provides enough guarantees for simple environments, it is desirable to obtain a COT protocol with arbitrarily composable security, which guarantees that the protocol remains secure even if multiple copies of other protocols and itself are run concurrently (*e.g.*, as in the Internet). Moreover, such security guarantees allow protocols and primitives to be securely used as building blocks for more complex applications. In this paper, we adopt the Universal Composability (UC) framework introduced by Canetti *et al.* in [Canetti 2001], which is one of the main approaches for proving the security of protocols under arbitrary composition.

### 1.1. Committed Oblivious Transfer Applications

As mentioned before, COT was originally shown to imply unconditionally secure multiparty computation with dishonest majorities in the stand-alone model [Crépeau et al. 1995]. Nevertheless, several multiparty computation protocols based on COT have been proposed since the primitive was first introduced. We present a brief list of the main COT applications bellow:

- Universally composable unconditionally secure two-party and multi-party computation with dishonest majorities [Estren 2004] based on [Crépeau et al. 1995].
- Active-secure universally composable two-party and multi-party computation with dishonest majorities based on joint gate evaluation [Garay et al. 2004].
- Efficient active-secure universally composable two-party computation based on Yao's garbled circuits [Jarecki and Shmatikov 2007].
- Efficient active-secure universally composable two-party and multi-party computation with dishonest majorities based on Multi-sender k-out-of-n OT [Lindell et al. 2011], which is implied by COT.

In light of the broad range of applications, we remark that COT is an important building block for efficient secure two-party and multi-party computation. Both classical and recent results show that this primitive is closely related to secure computation with dishonest majorities achieving nice efficiency. However, it is important to construct COT protocols that are as efficient as possible, since the efficiency of such secure computation protocols is directly related to their building blocks.

### 1.2. Related Works

Building on the notions introduced in the seminal work of Crépeau, van de Graaf and Tapp [Crépeau et al. 1995], different constructions of COT with diverse security guarantees have been proposed. These constructions are mainly based on two approaches: using black-box access to primitives [Estren 2004] and building on specific computational assumptions [Garay et al. 2004, Jarecki and Shmatikov 2007, Cramer and Damgård 1997, Cachin and Camenisch 2000]. While the protocols in [Garay et al. 2004, Estren 2004,

 ©2012 SBC — Soc. Bras. de Computação

Jarecki and Shmatikov 2007] offer universally composable security, the protocols in [Cramer and Damgård 1997, Cachin and Camenisch 2000] have only been proved in the stand-alone model.

Cramer and Damgård introduced a COT protocol based on the Q'th Residuosity assumptions with stand-alone security. Cachin and Camenisch proposed another stand-alone secure Verifiable Oblivious Transfer (VOT) protocol based on the Decisional Diffie Hellman (DDH) assumption. VOT is slightly different from COT in that the receiver learns $b_C$ instead of a commitment to $b_c$, while both parties are committed to their inputs. Both protocols achieve nice efficiency but still require a number of modular exponentiations, which is a costly operation.

Estren extended the results of [Crépeau et al. 1995] to the universal composability framework in [Estren 2004]. This work is basically a UC secure version of [Crépeau et al. 1995], relying on black-box access to universally composable oblivious transfer, bit commitment with XOR and authenticated channel functionalities. It requires no extra computational assumptions and builds on coding and privacy amplification techniques. Even though the generic construction is efficient itself, its concrete efficiency depends on the specific constructions of the primitives used as its building blocks. Considering that the most efficient protocol for UC secure oblivious transfer [Peikert et al. 2008] known in current literature still requires several modular exponentiations, the resulting protocol is computationally costly.

Protocols for universally composable committed oblivious transfer based on specific computational assumptions were introduced in [Jarecki and Shmatikov 2007, Garay et al. 2004]. Both of these protocols use the common reference string model as a setup assumption. While the construction in [Jarecki and Shmatikov 2007] is based on Decisional Composite Residuosity (DCR), the result on [Garay et al. 2004] relies on multiple assumptions, including DCR, strong RSA and Decisional Diffie Hellman. The protocol in [Jarecki and Shmatikov 2007] achieves string COT, while [Garay et al. 2004] achieves bit COT. However, both protocols rely on costly operations.

### 1.3. Our contributions

In this paper, we introduce a universally composable committed oblivious protocol based on a Trusted Initializer (TI). The TI is an entity that predistributes values to the parties during a setup phase before the protocol is executed. In contrast to previous protocols, our construction is round-optimal, only requires simple addition operations and achieves *unconditional security* (*i.e.* the protocol does not rely on any computational assumption). The main features of our protocol are summarized as follows:

- **Unconditional security:** the protocol remains secure even against computationally unbounded adversaries.
- **Universal Composability:** the protocol is proven UC secure.
- **Computational Efficiency:** we do not require any expensive operations (*e.g.* modular exponentiation or complex underlying protocols (*e.g.* zero-knowledge proofs).
- **Round Optimal:** Our protocol requires only two rounds, which is optimal.

Trusted initializers were introduced as part of the Commodity Based Cryptography model by Beaver [Beaver 1997, Beaver 1998], which is inspired by client-server

 ©2012 SBC — Soc. Bras. de Computação

architectures. The commodity based cryptography model makes perfect sense in the context of universal composability, since it was shown that it is necessary to rely on setup assumptions to obtain universally composable bit commitment, oblivious transfer and secure multi-party computation in general [Canetti and Fischlin 2001]. As an added advantage, the TI in our scenario can also be used to cheaply predistribute a broadcast channel, which is necessary for obtaining secure multi-party computation based on COT.

To the best of our knowledge this is the first universally composable COT protocol to be constructed on the commodity based cryptography model. Building on this model allowed us to obtain much better efficiency and security by trading off complex building blocks and assumptions for data predistribution and precomputation. In fact, we obtain the strongest possible security guarantees, namely unconditional security with universal composability. Moreover, our protocol is very simple in comparison to other protocols that require zero-knowledge proofs, verifiable encryption and universally composable building blocks.

### 1.4. Organization

The remainder of this paper is organized as follows. In Section 2, we establish notation and discuss preliminary notions and tools that will be used throughout the paper. In Section 3, we provide a brief discussion of the Universal Composability framework and define the ideal functionalities that will be used in this paper. In Section 4, we present the committed oblivious transfer protocol. In Section 5, we present the formal security proof for the protocol. In Section 6, we conclude with closing remarks and directions for future research.

## 2. Preliminaries

In this section, we establish notation and discuss the main tools and notions that we will use throughout the paper.

### 2.1. Notation

We will denote by $x \in_R D$ an uniformly random choice of an element $x$ over a domain $D$. All logarithms are to the base 2. If $X$ and $Y$ are families of distributions indexed by a security parameter $\lambda$, we use $X \overset{s}{\approx} Y$ to mean the distributions $X$ and $Y$ are *statistically close*, *i.e.*, for all polynomials $p$ and sufficiently large $\lambda$, we have $\sum_x |Pr[X = x] - Pr[Y = x]| < 1$. Two sequences $X_n, n \in \mathbb{N}$ and $Y_n, n \in \mathbb{N}$ of random variables are said to be *computationally indistinguishable*, denoted by $X \overset{c}{\approx} Y$, if for every non-uniform Probabilistic Polynomial Time *PPT* distinguisher D there exists a negligible function $\mathsf{negl.}(\cdot)$ such that for every $n \in \mathbb{N}$, $\mid Pr[D(X_n) = 1] - Pr[D(Y_n) = 1] \mid < \mathsf{negl.}(n)$.

Additionally we will denote by $[b]$ a commitment to a bit $b$ and by $|b|$ the information necessary to reveal this commitment; by $\mathfrak{C}$ the co-domain set of the commitments made with a bit commitment scheme $\mathcal{C}$; by $|\mathfrak{C}|$ the set of the information necessary to open a commitment in $\mathfrak{C}$. We will use a prime (') in all the variables on the simulated environment.

 ©2012 SBC — Soc. Bras. de Computação

### 2.2. Commodity Based Cryptography

In this section, we briefly discuss the Commodity Based Cryptography model introduced by Beaver [Beaver 1997, Beaver 1998]. This model was inspired by the client-server distributed computation model, where a powerful server performs complex tasks on behalf of a client. It is an efficient alternative for obtaining secure multi-party computation.

In this model, a Trusted Initializer precomputes certain operations that are then used by individual parties to execute a given protocol. The parties access such operations by requesting correlated predistributed values (the commodities) to the TI before they start executing the protocol itself. It is possible to obtain very efficient protocols in this model, since most of the required complex operations can be delegated to the TI and then predistributed to the parties. The Trusted Initializer is always assumed to be honest. Notably, this model has been used to construct commitments [Rivest 1999, Blundo et al. 2002, Hanaoka et al. 2003] and oblivious transfer [Beaver 1997].

Notice that the TI has no access to the parties' secret inputs nor does it receive any information from the parties. The only communication required between the TI and the parties occurs during a setup phase, when the TI predistributes information. If the parties are isolated from the TI during protocol execution it helps reduce the trust put in the TI. Moreover, data obtained from different TIs can be combined in order to thwart attacks from a single corrupted TI in more complex situations.

### 2.3. Bit Commitment with XOR

A bit commitment is a two-party primitive in which the sender's input is a bit $b$ and the receiver does not have any input. There are two phases (*commit* and *unveil*), and in the end of the protocol, the receiver accepts the bit revealed bit or not. In the commit phase, the sender (or *committer*) makes some computation such that the output appears random from the receiver's point of view. It means that in this phase, the receiver learns nothing about $b$ from the output $[b]$, which is a commitment to $b$. In the unveil phase, the sender sends the information necessary to decommit, $|b|$, (by making the inverse computation, or making the same computation and comparing the results) and the receiver checks if $b$ is consistent with the commitment previously received.

For a bit commitment scheme to be considered secure, it needs to be *private, binding* and *correct*. A bit commitment scheme is private if the receiver is unable to learn anything about the sender's committed bit before the unveil phase. A bit commitment scheme is binding if the sender is unable to unveil a bit $\bar{b} \neq b$ without being caught with high probability. Finally, a bit commitment scheme is correct if it never fails for honest parties and the receiver learns the value $b$ that the sender intended to commit to after the unveil phase.

For the sake of simplicity, lets refer to the sender as Alice and to the receiver as Bob. Our protocol requires a bit commitment scheme for which it is possible to verify XOR relationships between commitments without revealing their contents. More precisely, we want to verify the relation $b_1 = b_2 \oplus b_3$ given only the commitments $[b_1], [b_2]$ and the public bit $b_3$, without opening the commitments. Starting from any regular bit commitment scheme, we can use the *BCX* protocol for this purpose.

 ©2012 SBC — Soc. Bras. de Computação

BCX was first described in [Kilian 1992] (partly attributed to Rudich and Bennett). The protocol works as follows. For Alice to commit to a bit $b$ with a security parameter $n$ and a bit commitment scheme $\mathcal{C}$, she chooses $b_L \in_R \{0,1\}^n$. For each bit $1 \leq i \leq n$, she sets a new string $b_R \in \{0,1\}^n$, such that $b_{iR} = b_{iL} \oplus b$, where $b_{iR}$ is the $i$-th bit of $b_R$ and $b_{iL}$ is the $i$-th bit of $b_L$. Finally, she uses $\mathcal{C}$ to commit to all bits in $b_R$ and in $b_L$. Hence, a commitment to a bit $b$ in BCX consists in $2n$ commitments using $\mathcal{C}$. Then, the co-domain of commitments using BCX is $\mathfrak{C} = (\mathcal{C}(x)_{x\in\{0,1\}})^{2n}$, where $\mathcal{C}(x)_{x\in\{0,1\}}$ is the co-domain of commitments using $\mathcal{C}$. To unveil, Alice should unveil all $2n$ bits. Bob accepts it if $b_{iL} \oplus b_{iR} = b, \forall 1 \leq i \leq n$. So, if the information necessary to open a commitment using $\mathcal{C}$ is $|\mathcal{C}(x)_{x\in\{0,1\}}|$, the information necessary to open a commitment in BCX is $(|\mathcal{C}(x)_{x\in\{0,1\}}|)^{2n}$.

Given the commitments $[a]$,$[c]$ and the public value $x$, Alice proves to Bob that $a \oplus c = x$ by the following procedure. Bob chooses and sends to Alice random permutations $\pi_a, \pi_b$ to shuffle the bit positions of $a_{iL}$,$a_{iR}$, $c_{iL}$, and $c_{iR}$. Alice applies the permutations, sending to Bob $L_i = a_{iL} \oplus c_{iL}, R_i = a_{iR} \oplus c_{iR}, \forall 1 \leq i \leq n$. Bob randomly chooses and sends to Alice $V \in \{L,R\}^n$. For each $i$, if $V_i = L$, then Alice opens $[a_{iL}]$ and $[c_{iL}]$; else opens $[a_{iR}]$ and $[c_{iR}]$. Finally, Bob accepts the proof if $\forall 1 \leq i \leq n$, $a_{iL} \oplus c_{iL} = L_i$, if $V_i = L$; or $a_{iR} \oplus c_{iR} = R_i$ if $V_i = R$.

This simple XOR proof can be used to prove if two commitments are related to the same bit ( if the proof that $[b_1] = [b_2] \oplus 0$ is accepted, then $b_1 = b_2$ ), or not ( if the proof that $[b_1] = [b_2] \oplus 1$ is accepted, then $b_1 \neq b_2$). However, if more than one proof is generated for the same commitment, Bob may learn $b$ before the unveil phase (*e.g.* by learning $b_{iL}$ in the first proof and $b_{iR}$ in the second proof).

In order to solve this issue, if a commitment $[b]$ is meant to be used for a XOR proof, Alice creates three more pairs $b_L^1 \in_R \{0,1\}^n$, $b_R^1 \in \{0,1\}^n$, $b_L^2 \in_R \{0,1\}^n$, $b_R^2 \in \{0,1\}^n$, and $b_L^3 \in_R \{0,1\}^n$, $b_R^3 \in \{0,1\}^n$, where each $b_{iL}^j$, and $b_{iR}^j$, $1 \leq i \leq n, 1 \leq j \leq 3$ are commitments with $b_{iL}^j \oplus b_{iR}^j = b$. Alice concatenates $b_L^1, b_L^2, b_L^3$ and $b_R^1, b_R^2, b_R^3$ obtaining two strings of $3n$ bits each. Bob sends a random permutation $\pi$ that shuffles the $3n$ position bits. After applying the random permutation to $b_L$ and $b_R$, Alice divides $b_L$ in three $n$ bits strings, calling them respectively $b_L^1$, $b_L^2$ and $b_L^3$. She does the same to $b_R$, obtaining again $b_R^1, b_R^2, b_R^3$. Let's say that $b_1 = b_{iL}^1 \oplus b_{iR}^1$,$b_2 = b_{iL}^2 \oplus b_{iR}^2$, $b_3 = b_{iL}^3 \oplus b_{iR}^3$ for all $1 \leq i \leq n$. For Alice to prove to Bob that $b_1 = b_2 = b_3 = b$, she uses the same previous proceeding to prove that $b \oplus b_1 = 0$. So, if the test is successful, $b_L, b_R, b_L^1, b_R^1$ cannot be used any more, but Bob believes that $b_2 = b_3 = b$. At this point, Alice can discard $b_L, b_R$, use $b_L^2, b_R^2$ in the XOR proof. After finishing the proof, she discards $b_L^2, b_R^2$, sets $b_L = b_L^3$, sets $b_R = b_R^3$, and sends the $2n$ bits of $b_L^3, b_R^3$ to Bob.

## 3. Universal Composability

In this section, we present a brief discussion of the Universal Composability framework. We refer the readers to [Canetti 2001, Canetti and Fischlin 2001] for further details.

In the UC framework we consider a set of parties interacting with an Adversary $\mathcal{A}$, and a *environment* $\mathcal{Z}$. The environment is responsible for providing the inputs for the parties and $\mathcal{A}$, and receiving their outputs. All these entities are modeled as Interactive Turing Machines.

 ©2012 SBC — Soc. Bras. de Computação

The whole point of the UC framework is that $\mathcal{Z}$ represents all activity external to the current execution of the protocol. So, in order to prove the security of a specific protocol implementation, we must define an ideal version of the protocol and an ideal adversary $\mathcal{S}$. Then, we have to show that no $\mathcal{Z}$ can distinguish between an execution of the specific protocol implementation with the parties and $\mathcal{A}$, and an execution of the ideal version with the parties and $\mathcal{S}$.

The ideal version of the protocol is called the ideal functionality $\mathcal{F}$ and it does exactly what the protocol should do in a black box manner. In other words, given the inputs, the ideal functionality follows the primitive specification and returns the output as specified. However, the functionality must deal with the actions of corrupted parties, such as invalid inputs and deviations from the protocol.

Some interesting points about $\mathcal{F}$ are: the communication between the parties and $\mathcal{F}$ are made by writing on their input and output tapes; $\mathcal{S}$ has no access to the contents of messages sent between the parties and $\mathcal{F}$ unless one or both parties are corrupted; $\mathcal{Z}$ cannot see the messages sent between the parties and $\mathcal{F}$ (and cannot see the messages sent between the parties in the real protocol execution).

The ideal adversary $\mathcal{S}$ should be constructed to act in the same way as the adversary $\mathcal{A}$ in the interaction with the real protocol. It means that every attack that $\mathcal{A}$ can do in the real protocol must be simulated by $\mathcal{S}$ in the ideal execution of the protocol. In other words, all the attacks can be possibly aimed at the real protocol are those reflected in the ideal functionality. Hence, if the ideal functionality specifies a secure primitive, any attack against a protocol that securely implements such functionality will not disrupt the functionality's security guarantees. A point that should be clarified here is that $\mathcal{S}$ does not deliver any message between the parties. It just simulates the messages of the honest parties (if any) to the corrupted party and delivers messages from parties to the ideal functionality.

It is said that a real protocol securely realizes an ideal functionality (i.e., the protocol implementation is secure under the UC framework) if for every adversary $\mathcal{A}$ in the real protocol there exists an ideal adversary $\mathcal{S}$ such that a $ZZZ$ cannot distinguish an execution of the specific protocol implementation with the parties and $\mathcal{A}$ from an execution of the ideal version with the parties and $\mathcal{S}$. Formally, we have: $\forall\, \mathcal{A},\ \exists\, \mathcal{S},\ s.t.\ \forall \mathcal{Z} : \mathrm{REAL}_{\Pi,\mathcal{A},\mathcal{Z}} \overset{c}{\approx} \mathrm{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$, where the probability distribution is taken over all random tapes of the parties; REAL is the output of the environment $\mathcal{Z}$ when it interacts with $\mathcal{A}$ and the other parties running the real protocol, and, similarly, IDEAL is the output of the environment $\mathcal{Z}$ when it interacts with $\mathcal{S}$ and the other parties in the ideal execution of the protocol. Without loss of generality, let us define REAL and IDEAL as single bits.

In our work, we will show that the parties' random tapes are fed with the same probability distribution in the real execution and in the ideal one. Thus, we can actually show that the real execution is *perfectly* indistinguishable from the ideal simulation.

### 3.1. Ideal Functionalities

The Trusted Initializer functionality is defined as follows.

 ©2012 SBC — Soc. Bras. de Computação

IDEAL FUNCTIONALITY $\mathcal{F}_{TI}$

- When first activated, choose $r_0, r_1, d \in_R \{0,1\}$, compute the commitments $[r_0], [r_1], [d], [r_d]$ (where $r_d \in \{r_0, r_1\}$), set $|r_0|, |r_1|, |d|, |r_d|$ as the information to open the respectively commitments, and distribute $(r_0, r_1, |r_0|, |r_1|, [d], [r_d])$ to Alice and $(d, r_d, |d|, |r_d|, [r_0], [r_1])$ to Bob .

The following COT ideal functionality $\mathcal{F}_{COT}$ is based on functionalities presented in [Estren 2004, Jarecki and Shmatikov 2007]. Each commitment has a unique identifier *cid* and each $\mathcal{F}_{COT}$ instance has a unique identifier *sid*. The parties $P_R, P_S$ refer to Sender and Receiver, respectively, and $P_i$ may refer to any of them. $\mathcal{S}$ is the ideal world adversary.

**Ideal Functionality $\mathcal{F}_{COT}$**

- Upon receiving a (COMMIT, $P_i$,*sid*, *cid*, $b$) message from $P_i$:
  - If *cid* has not been used for any previous commitment, then store $b$ and broadcast a (RECEIPT, $P_i$,*sid*, *cid*, $[b]$), where $[b]$ is the commitment of bit $b$ with identifier *cid*.
  - Otherwise, do nothing.
- Upon receiving $msg$ = (STARTCOT, $P_S$, $P_R$, *sid*, $cid_0, cid_1, cid_n, cid_c$) from $P_R$:
  - If $cid_0$ and $cid_1$ refer to existing valid unopened commitments by $P_S$ to the respective bits $b_0$ and $b_1$, $cid_c$ refers to an existing unopened commitment by $P_R$ to bit $c$, and $cid_n$ does not refer to any previous commitment, then record $msg$ and forward it to $P_S$.
  - If any of the conditions defined in the immediately previous item fails, broadcast message (COTFAILED, $P_S, P_R$,*sid*).
- Upon receiving (COMPLETECOT, $P_S$, $P_R$, *sid*, $cid_0, cid_1, cid_n, cid_c$) from $P_S$:
  - $\mathcal{F}_{COT}$ verifies that a message (STARTCOT, $P_S$, $P_R$, *sid*, $cid_0, cid_1, cid_n, cid_c$) has been recorded. In such a case, $\mathcal{F}_{COT}$ generates a commitment to $b_c$ under $cid_n$ and sends a message (TRANSFERCOT, $P_S$, $P_R$, *sid*, $cid_0, cid_1, cid_n, cid_c, b_c$).
  - If no such record exists, $\mathcal{F}_{COT}$ sends message (COTFAILED, $P_S, P_R$,*sid*).

## 4. The Protocol

In this section we describe our COT protocol. Our protocol is constructed in the commodity based model, where a Trusted Initializer predistributes a number of correlated values to Alice and Bob, allowing them to perform COT. Alice inputs commitments $[b_0], [b_1]$ to bits $b_0, b_1$ for which she knows the opening information $|b_0|, |b_1|$, whereas Bob knows $[b_0], [b_1]$. Likewise, Bob inputs a commitment $[c]$ to a bit $c$ for which he knows the opening information $|c|$, whereas Alice knows $[c]$. In the end of the protocol, Alice learns nothing, while Bob receives $b_c$ and the opening information $|b_c|$ but learns nothing about $b_{1-c}$.

Intuitively, this protocol builds on the techniques of [Crépeau et al. 1995], combining a plain oblivious transfer protocol and a bit commitment scheme with XOR.

 ©2012 SBC — Soc. Bras. de Computação

We obtain the commitment scheme with XOR from any regular bit commitment (*e.g.* [Rivest 1999]) through the BCX construction [Kilian 1992]. The basic idea is to have both parties run the OT protocol with committed inputs instead of plain bits. The OT protocol is slightly modified so that, in the end of the protocol, the receiver learns both the commitment corresponding to $b_c$ and the necessary opening information (*i.e.* learning $b_c$ itself). The receiver then uses the XOR property of the commitment scheme to check that the transferred commitment is indeed one of Alice's initial commitments (using the procedures described in Section 2.3).

The basic oblivious transfer protocol used in our construction is predistributed by the TI as Alice's values $r_0, r_1 \in_R \{0,1\}$ and Bob's values $d \in_R \{0,1\}$, $r_d \in \{r_0, r_1\}$. In order to run the OT protocol, Bob computes $e = c \oplus d$ and sends it to Alice in Step 1. Alice computes $f_0 = b_0 \oplus r_e$, $f_1 = b_1 \oplus r_{1-e}$ and sends these values to Bob in Step 2. Finally, Bob recovers bit $b_c$ by computing $b_c = f_c \oplus r_d$ in Step 3. Notice that $f_c = b_c \oplus r_d$ and $f_{1-c} = b_{1-c} \oplus r_{1-d}$. Hence, Bob can't recover $b_{1-c}$ because he does not know $r_{1-d}$, while Alice can't recover $c$ because she does not know $d$.

| | Alice | Bob |
|---|---|---|
| TI | $r_0, r_1 \in_R \{0,1\}$<br><br>$[d], [r_d], \lvert r_0\rvert, \lvert r_1\rvert$ | $d \in_R \{0,1\}$<br>$r_d \in \{r_0, r_1\}$<br>$[r_0], [r_1], \lvert d\rvert, \lvert r_d\rvert$ |
| Data sent by TI | $\mu_A := (r_0, r_1, \lvert r_0\rvert,$<br>$\lvert r_1\rvert, [d], [r_d])$ | $\mu_B := (d, r_d, \lvert d\rvert,$<br>$\lvert r_d\rvert, [r_0], [r_1])$ |
| Input | $b_0, b_1 \in \{0,1\}$<br>$\lvert b_0\rvert, \lvert b_1\rvert, [c]$ | $c \in \{0,1\}$<br>$\lvert c\rvert, [b_0], [b_1]$ |
| Step 1 | | $e = c \oplus d$<br>Send $\mu_1 := (e)$ to Alice |
| Step 2 | If $e \notin \{0,1\}$, then abort<br>Otherwise, do:<br>$f_0 := b_0 \oplus r_e$<br>$f_1 := b_1 \oplus r_{1-e}$<br>Compute:<br>$[f_0], \lvert f_0\rvert$ and $[f_1], \lvert f_1\rvert$<br>Send $\mu_2 := ([f_0], [f_1],$<br>$\lvert f_0\rvert, \lvert f_1\rvert, f_0, f_1)$ to Bob | |
| Step 3 | | Check if $f_0 \neq [b_0] \oplus [r_e]$<br>or $f_1 \neq [b_1] \oplus [r_{1-e}]$, then abort<br>Otherwise, open $f_0, f_1$,<br>Set $b_c = f_c \oplus r_d$ and<br>if $(f_0, f_1) = (0,0)$ then $[b_c] := [r_d]$<br>if $(f_0, f_1) = (1,1)$ then $[b_c] := \neg[r_d]$<br>if $(f_0, f_1) = (0,1)$ then $[b_c] := [c] + [r_d]$<br>if $(f_0, f_1) = (1,0)$ then $[b_c] := \neg[c] + [r_d]$ |
| Output | $\perp$ | $v := [b_c], b_c$ |

The COT protocol combines the aforementioned OT protocol with BCX commitments by having the TI predistribute commitments $[d], [r_d]$ to Alice and commitments $[r_0], [r_1]$ to Bob, besides the regular OT predistributed values. Moreover, as this is a COT

 ©2012 SBC — Soc. Bras. de Computação

protocol, Alice is required to send Bob the commitments $[b_0], [b_1]$ and Bob is required to send Alice the commitment $[c]$. The OT protocol is run with its normal inputs, except that Alice also sends commitments $[f_0], [f_1]$ and the respective opening information in Step 2. Upon receiving those commitments Bob uses the XOR property to check that they correspond to $f_0 = [b_0] \oplus [r_e]$ and $f_1 = [b_1] \oplus [r_{1-e}]$. If at least one of these relations hold, he continues by determining the commitment $[b_c]$ by checking the relations over $(f_0, f_1)$ specified in Step 3 of the COT protocol.

Notice that Bob can open $[b_c]$, since he has the information to open $[c]$ and $[r_d]$. In Step 3, Bob checks whether Alice set the masks $f_0, f_1$ appropriately (i.e. $f_0, f_1$ are not random numbers), in order to verify that he correctly received one of Alice's input bits. However, we have to verify that $f_0 = b_0 \oplus r_e$ and $f_1 = b_1 \oplus r_{1-e}$ without revealing any of the bits $b_0, b_1, r_0, r_1$. This is the reason why we need a bit commitment with XOR, meaning that it has the property of evaluating XOR between commitments without having to open them. If the commitment scheme is homomorphic, we don't require this XOR property, since a sum of commitments is also a commitment (*i.e.*, $[b_0] + [r_e] = [b_0 + r_e]$) and Alice cannot deviate from the protocol. A extended version of this protocol using a homomorphic bit commitment scheme will be included in the full version.

### 4.1. Correctness

First we analyze correctness in the case that both parties are honest.

*Alice learns nothing about $c$.* Bob sends information about $c$ only in Step 1. In this step, Bob sends $e = c \oplus d$, and provided that Alice cannot extract $d$ from the commitments $[d]$ and $[r_d]$, $c \oplus d$ behaves as a one time pad encryption of $c$, since Alice does not know $d$. Hence, Alice can only guess the value of $c$.

*Bob learns nothing about $b_{\bar{c}}$.* Notice that if $c = 0$, then $e = d$ and $f_1 = b_1 \oplus r_{1-d}$. Conversely, if $c = 1$ then $e = 1 - d$ and $f_0 = b_0 \oplus r_{1-d}$. Also notice that Alice computes $f_{1-c} = b_{1-c} + r_{1-d}$ and Bob doesn't know $r_{1-d}$. Hence, he is not able to compute $b_{1-c}$ from $f_{1-c}$.

## 5. Universally Composable Security Analysis

In this section we construct the simulators and prove that our protocol is universally composable.

Remember that the players in a real world protocol execution interact with the environment $\mathcal{Z}$, with the TI, and with an adversary $\mathcal{A}$, which will be called the real-life adversary. On the other hand, the players in a execution of the ideal process interact with the environment $\mathcal{Z}$ and TI as well, and with an ideal adversary $\mathcal{S}$ (the simulator).

In order to prove the UC security, we have to construct an ideal adversary $\mathcal{S}$ such that the environment cannot guess whether it is interacting with $\mathcal{A}$ in the real-life execution or with $\mathcal{S}$ in the ideal simulation with more than a negligible probability. So, we will construct $\mathcal{S}$ by showing the actions that she takes in the ideal process for each situation: (Alice corrupted $\wedge$ Bob honest) $\vee$ (Alice honest $\wedge$ Bob corrupted) $\vee$ (Alice corrupted $\wedge$ Bob corrupted) $\vee$ (Alice honest $\wedge$ Bob honest).

 ©2012 SBC — Soc. Bras. de Computação

### 5.1. Corrupted Alice and Honest Bob

**Simulation.** In this case only Alice is corrupted, while Bob acts as an honest party. The simulator $\mathcal{S}$ runs an internal copy of the real-life adversary $\mathcal{A}$ called $A'$ and every input value that $\mathcal{S}$ receives from $\mathcal{Z}$ is forwarded to $A'$ without any alteration. The interactions between $A'$ and $\mathcal{S}$ correspond to those of Alice in the real protocol execution with $\mathcal{Z}$, TI, and Bob. $\mathcal{S}$ operates as follows:

1. **Setup Phase:**
   - Simulating $\mathcal{F}_{TI}$: $\mathcal{S}$ chooses $r'_0, r'_1 \in_R \{0,1\}$, $[d'], [r'_{d'}] \in_R \mathfrak{C}$, $|r'_0|, |r'_1| \in_R |\mathfrak{C}|$, and sends $\mu'_A = (r'_0, r'_1, |r'_0|, |r'_1|, [d'], [r'_{d'}])$ to $A'$ when it requests data from $\mathcal{F}_{TI}$.
2. **Protocol**:
   (a) Simulating message $\mu'_1$: $\mathcal{S}$ chooses random bits $d, c \in_R \{0,1\}$ and $r_d \in_R \{r'_0, r'_1\}$, computes $e = c \oplus d$ and sends $\mu'_1 = e$ to corrupted Alice.
   (b) Extracting $b_0, b_1$: Upon receiving message $\mu_2 := ([f'_0], [f'_1], |f'_0|, |f'_1|, f'_0, f'_1)$ from corrupted Alice, $\mathcal{S}$ computes $b'_0 = f'_0 \oplus r_e$ and $b'_1 = f'_1 \oplus r_{1-e}$ (notice that $\mathcal{S}$ is able to do so, since it knows $r_0, r_1$). $\mathcal{S}$ sends the messages (COMMIT, $P_S$,*sid*, $cid_0$, $b'_0$), and (COMMIT, $P_S$,*sid*, $cid_1$, $b'_1$) to $\mathcal{F}_{COT}$.
   (c) Finishing the protocol: Upon receiving a message (STARTCOT, $P_S$, $P_R$, $sid, cid_0, cid_1, cid_n, cid_c$) from $\mathcal{F}_{COT}$, $\mathcal{S}$ sends a message (COMPLETECOT, $P_S$, $P_R$, $sid, cid_0, cid_1, cid_n, cid_c$) to $\mathcal{F}_{COT}$.

**Indistinguishability.** In order to prove that $\forall\mathcal{A}\exists\mathcal{S}\forall\mathcal{Z} : \text{REAL}_{\Pi,\mathcal{A},\mathcal{Z}} \equiv \text{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$, we have to show that the view of the environment $\mathcal{Z}$ in the execution of the real-life protocol in the presence of the real-life adversary $\mathcal{A}$ is indistinguishable from the view of $\mathcal{Z}$ in the execution of the ideal protocol in the presence of the ideal adversary $\mathcal{S}$.

Notice that the values $r'_0, r'_1 \in_R \{0,1\}$, $[d'], [r'_{d'}] \in_R \mathfrak{C}$, $|r'_0|, |r'_1| \in_R |\mathfrak{C}|$ simulated by $\mathcal{S}$ in the setup phase have the same distribution as in the real protocol. Hence the pre-distributed values are perfectly indistinguishable from values in the real world execution.

Apart from that, the only difference in the simulated protocol is that $\mathcal{S}$ chooses a random value for $c$. However, since $d$ is also chosen randomly, $e = c \oplus d$ acts as an one time pad encryption of $c$. Thus, there's no difference in the distribution of the values sent to corrupted Alice in the simulation.

Notice that $\mathcal{S}$ obtains corrupted Alice's inputs $b'_0, b'_1$ since it knows $r'_0, r'_1$, which $\mathcal{S}$ chose itself in the setup phase.

### 5.2. Honest Alice and Corrupted Bob

**Simulation.**

As in the previous case, $\mathcal{S}$ runs an internal copy of the real-life adversary $\mathcal{A}$ called $B'$, and every input value that $\mathcal{S}$ receives from $\mathcal{Z}$ is forwarded to $B'$ without any alteration. The interactions between $B'$ ansd$\mathcal{S}$ are those of Bob in the real protocol execution with $\mathcal{Z}$, the TI and Alice. $\mathcal{S}$ works as follows:

1. **Setup Phase:**

 ©2012 SBC — Soc. Bras. de Computação

- Simulating $\mathcal{F}_{TI}$: $\mathcal{S}$ chooses $d', r'_{d'} \in_R \{0,1\}$, $|d'|, |r'_{d'}| \in_R |\mathfrak{C}|$, $[r'_0], [r'_1] \in_R \mathfrak{C}$, and sends $\mu'_B = (d', r'_{d'}, |d'|, |r'_{d'}|, [r'_0], [r'_1])$ to $B'$.

2. **Protocol**:
   - Extracting $c$: Upon receiving message $\mu'_1 = (e)$, $\mathcal{S}$ computes $c = e \oplus d'$ (notice that this is possible since $\mathcal{S}$ knows $d'$). $\mathcal{S}$ sends message (COMMIT, $P_i$,*sid*, $cid_c$, $c$) to $\mathcal{F}_{COT}$.
   - Obtaining $b_c$: $\mathcal{S}$ waits for two messages messages (RECEIPT, $P_S$,*sid*, $cid_0$, $[b'_0]$) and (RECEIPT, $P_S$,*sid*, $cid_1$, $[b'_1]$) from $\mathcal{F}_{COT}$ and then sends a message (STARTCOT, $P_S$, $P_R$, $sid, cid_0, cid_1, cid_n, cid_c$) to $\mathcal{F}_{COT}$.
   - Simulating $\mu'_2$ and Finishing the protocol: Upon receiving a message (TRANSFERCOT, $P_S$, $P_R$, $sid, cid_0, cid_1, cid_n, cid_c, b'_c$), $\mathcal{S}$ selects random values $r'_0, r'_1 \in_R \{0,1\}$, $[d'], [r'_{d'}] \in_R \mathfrak{C}$, $|r'_0|, |r'_1| \in_R |\mathfrak{C}|$ and a random message $b'_{1-c}$. It computes $f'_0, f'_1, [f'_0], [f'_1], |f'_0|, |f'_1|$ as in the real protocol using messages $b'_c, b'_{1-c}$ (where $b'_c$ was obtained from $\mathcal{F}_{COT}$.

**Indistinguishability.** As in the previous case, all the predistributed values $d', r'_{d'} \in_R \{0,1\}$, $|d'|, |r'_{d'}| \in_R |\mathfrak{C}|$, $[r'_0], [r'_1] \in_R \mathfrak{C}$ simulated by $\mathcal{S}$ are distributed exactly as in the real world protocol. Hence, the only difference lies in message $b'_{1-c}$, that is randomly selected.

It is easy to notice that $\mathcal{S}$ can obtain $c$ by computing $c = e \oplus d$, since it generated $d$ and knows its value.

However, since the only information that corrupted Bob obtains about $b'_{1-c}$ is $f_{1-c}, |f_{1-c}|, [f_{1-c}]$, corrupted Bob cannot distinguish $b'_{1-c}$ in the simulation from the message in the real execution, since $f_{1-c} = b_{1-c} \oplus r_{1-d}$ and corrupted Bob doesn't know $r_{1-d}$, which is randomly selected.

Notice that the distribution of all values exchanged between $\mathcal{S}$ corrupted Bob and the environment is exactly the same as in the real world execution. Thus, the simulation is perfectly indistinguishable from the real world protocol execution.

### 5.3. Both parties are honest or corrupted

When both parties are corrupted or honest, $\mathcal{S}$ simply forwards all messages between parties and from parties to $\mathcal{F}_{COT}$.

## 6. Conclusion

We introduce the first unconditionally secure and universally composable committed oblivious transfer protocol in the commodity based cryptography model. Our protocol achieves the highest security guarantee known in current literature while conserving communication efficiency, since it is round-optimal. Moreover, being based on predistributed values and precomputation by a Trusted Initializer, our protocol is extremely efficient, not requiring any costly operations such as modular exponentiations. We remark that this COT protocol can also be used as a verifiable oblivious transfer protocol with minor modifications. As a future work, we suggest to obtain a modified version of this protocol based on homomorphic commitments, which could achieve even better efficiency. Another interesting related line of work is applying the commodity based cryptography model to perform secure two-party computation on arithmetic circuits (*i.e.* in $\mathbb{Z}_p$) without relying on COT.

 ©2012 SBC — Soc. Bras. de Computação

**Acknowledgement**

The work of the third author was partially supported by a FAPEMIG grant, number APQ-02719-10.

## References

Beaver, D. (1997). Commodity-based cryptography (extended abstract). In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 446–455, New York, NY, USA. ACM.

Beaver, D. (1998). Server-assisted cryptography. In *Proceedings of the 1998 workshop on New security paradigms*, NSPW '98, pages 92–106, New York, NY, USA. ACM.

Blundo, C., Masucci, B., Stinson, D. R., and Wei, R. (2002). Constructions and bounds for unconditionally secure non-interactive commitment schemes. *Des. Codes Cryptography*, 26(1-3):97–110.

Cachin, C. and Camenisch, J. (2000). Optimistic fair secure computation. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 93–111, London, UK, UK. Springer-Verlag.

Canetti, R. (2001). Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, FOCS '01, pages 136–, Washington, DC, USA. IEEE Computer Society.

Canetti, R. and Fischlin, M. (2001). Universally composable commitments. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 19–40, London, UK. Springer-Verlag.

Cramer, R. and Damgård, I. (1997). Linear zero-knowledge - a note on efficient zero-knowledge proofs and arguments. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 436–445, New York, NY, USA. ACM.

Crépeau, C., van de Graaf, J., and Tapp, A. (1995). Committed oblivious transfer and private multi-party computation. In Coppersmith, D., editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 110–123. Springer.

Estren, G. (2004). Universally composable committed oblivious transfer and multi-party computation assuming only basic black-box primitives. Master's thesis, McGill University.

Garay, J. A., Mackenzie, P., and Yang, K. (2004). Efficient and universally composable committed oblivious transfer and applications. In *Proceedings of the First Theory of Cryptography Conference (2004*, pages 297–316. Springer-Verlag.

Goldreich, O., Micali, S., and Wigderson, A. (1987). How to play any mental game. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229, New York, NY, USA. ACM.

Hanaoka, G., Imai, H., M'uller-Quade, J., Nascimento, A., and Otsuka, A. (2003). Unconditionally secure homomorphic pre-distributed bit commitment. *ICS 2003*.

 ©2012 SBC — Soc. Bras. de Computação

Jarecki, S. and Shmatikov, V. (2007). Efficient two-party secure computation on committed inputs. In *Proceedings of the 26th annual international conference on Advances in Cryptology*, EUROCRYPT '07, pages 97–114, Berlin, Heidelberg. Springer-Verlag.

Kilian, J. (1988). Founding crytpography on oblivious transfer. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31, New York, NY, USA. ACM.

Kilian, J. (1992). A note on efficient zero-knowledge proofs and arguments. *Proceeding STOC '92 Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing*, pages 723 – 732.

Lindell, Y., Oxman, E., and Pinkas, B. (2011). The ips compiler: Optimizations, variants and concrete efficiency. *Advances in Cryptology–CRYPTO 2011*, pages 259–276.

Peikert, C., Vaikuntanathan, V., and Waters, B. (2008). A framework for efficient and composable oblivious transfer. In Wagner, D., editor, *Advances in Cryptology ? CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer Berlin / Heidelberg.

Rivest, R. L. (1999). Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer. Preprint available at http://people.csail.mit.edu/rivest/Rivest- commitment.pdf.

 ©2012 SBC — Soc. Bras. de Computação

# Cleaning up the PKI for Long-Term Signatures

**Martín A. G. Vigil**[1], **Ricardo Felipe Custódio**[2]

[1] Kryptographie und Computeralgebra
Technische Universität Darmstadt
Hochschulstraße 10
64289 Darmstadt, Germany

[2]Laboratório de Segurança em Computação (LabSEC)
Universidade Federal de Santa Catarina (UFSC)
PO Box 476
88040-900 Florianópolis, Brazil

vigil@cdc.informatik.tu-darmstadt.de, custodio@inf.ufsc.br

***Abstract.*** *In this paper we present a new approach for the conventional X.509 Public Key Infrastructures (PKI). Our goal is to reduce the effort to handle signatures in the long term. The novelty is that a Root CA reissues subordinate certificates of final users, but adjusting validity periods to exclude the periods after a revocation. The Root CA also authenticates timestamps. The result is the cleaned PKI, which is simpler than the conventional PKI because: a) there is no revocation; b) there is no intermediary Certification Authority; c) signatures are trustworthy as long as the used cryptographic algorithms remain secure. As benefits, we reduce the need of timestamps and consequently the demand for storage space and processing time to use signed documents.*

## 1. Introduction

Digital certificates based on the X.509 Public Key Infrastructure (PKI) and digital signatures are the most prominent means to ensure integrity, authenticity and proof-of-existence for digital documents. However, this solution presents several issues (see [Gutmann 2002] for an overview). One of the hardly criticized drawbacks of PKI is revocation. For instance, a certificate can be revoked *before* it expires, e.g. because of key compromise. But, what if we could foresee when a certificate would be revoked? Then, we could set certificates in advance to expire *before* they are revoked. By doing so, we eliminate the need for revocation, thereby *simplifying* PKI. Base on this idea, we present our proposal.

**Contribution** We present a new proposal for simplifying the conventional X.509 PKI by removing revocation and intermediary Certification Authorities (CA). Our idea consists of reissuing all final certificates in a PKI, but assigning new validity period that *excludes* revocation. This procedure is performed by the Root CA of a PKI, yielding a *cleaned PKI*, which has neither revocation nor intermediary CAs. The Root CA also *authenticates* all the timestamps issued by the subordinate Timestamp Authorities. A digital signature in the cleaned PKI is *trustworthy* as long as the used cryptographic algorithms remain secure. The proposal benefits archivers (e.g. digital libraries), whose computational effort to preserve trustworthy signatures is reduced. Yet, the users of signed documents are benefited too, because the processing time to evaluate signatures is reduced.

 ©2012 SBC — Soc. Bras. de Computação

**Organization** In Section 2, we show that preserving and verifying digital signatures in the long term are hard tasks. In this context, we state the problems that our proposal addresses. We provide the related work in Section 3. We present a top-level description of our proposal, formal definitions and an illustrated example in Section 4. Section 5 describes timestamps, detailing the problems stated in Section 2 and also providing a solution. Section 6 presents the procedures and players in our scheme. Section 7 presents the benefits of the cleaned PKI. Finally, we present our considerations and future work (Section 8).

## 2. Problems of Signatures in the Long Term

Public Key Infrastructure (PKI) [Kohnfelder 1978] is a well-known means to ensure authenticity, integrity and proof-of-existence for digital documents. Authenticity and integrity can be guaranteed by digital signatures. The verification of digital signature requires trustworthy public keys, which are provided by a PKI. Proof-of-existence is guaranteed by timestamps, whose verification may require trustworthy public keys too.

The X.509 [ITU-T 2005] is the most prominent model of PKI, in which entities' public keys are organized in a hierarchy. On the highest level there is the Root CA, bellow which there are other Certification Authorities (CA), Timestamp Authorities (TSA) and final users. The X.509 technically describes public key certificates, and revocation status (e.g. Certificate Revocation List). From here on, we refer to X.509 as the *conventional PKI*.

A trustworthy signature requires mathematical and semantic correctness. The mathematical correctness is guaranteed if the signature is valid under the signer's public key. The semantic correctness is ensured if the binding between the signer's public key and his credentials, i.e. his public key certificate, *was* valid *when* the signature was created. Since the time when a signature was created is hard to securely guarantee by digital means, the time reference of the binding validity is shifted to the present. Thus, the semantic correctness is guaranteed if: a) the signer's certificate and corresponding superordinate CAs' certificates have not expired and have not been revoked *so far*; and b) the used cryptographic algorithms are *currently* secure. The first condition is verified with the help of the so-called *certification path validation* algorithm [Cooper et al. 2008]. The verification of the second condition requires the knowledge of the current status of cryptographic algorithms. Such information has been provided by governmental agencies, e.g. [National Institute of Standards and Technology 2007]

The semantic correctness is easily verified for short-term signatures, when the signing and verification times are close. This is because the signer's and superordinate CAs' certificates have probably not expired yet, revocation status is up to date and available, and the cryptographic algorithms are still secure. Nevertheless, this situation changes in the long term. Therefore, proof-of-existence is necessary to ensure that all those requirements were fulfilled at a time in the past (preferably as close as possible to the signing time). Signed timestamps [Adams et al. 2001a] are the most used means for proof-of-existence in the conventional PKI. Thus, certificates, their revocation status and timestamps are necessary to evaluate the semantic correctness of a signature in the long term. From here on, we refer to these objects as the *protection data* of a signed document.

Since timestamps are signed, the semantic correctness of their signatures should

 ©2012 SBC — Soc. Bras. de Computação

be also verified. Again, proof-of-existence is necessary to allow such verification in the long term. As consequence, new timestamps are required, being the preservation of trustworthiness signatures an end-less process. In this context, we point out the following problems.

**Problem 1.** *The effort to preserve signed documents is not optimal*. Given a set of documents that were signed by different signers, preserving this set usually requires the archiver a non-optimal effort. The reason is that the archiver has to monitor the trustworthiness of each document's signature, applying new timestamps on it *before* the signer's and timestamper's certificates expire (for a detailed explanation, see Section 5). Depending on the number of different certificates and corresponding expiration dates, the archiver is likely to schedule timestamping procedures on several dates, instead of accessing the archive *as few as possible*. Therefore, this approach is neither optimal nor scalable [Huhnlein et al. 2009].

**Problem 2.** *The amount of protection data to store and verify is not optimal in the long term.* The regular use of timestamps on a signed document increases the protection data. The reason is that a timestamp is a signed document too, therefore protection data for it is necessary. Thus, the older the signed document is, the more timestamps are needed, the more protection data exists, the more storage space is used. Moreover, the more protection data exists, the more signatures a verifier has to evaluate to infer the trustworthiness of a signed document.

## 3. Related Work

In literature there are many proposals to improve the conventional PKI. We present them in regard to the problems 1 and 2.

*Evidence Record Syntax* (ERS) [Gondrom et al. 2007] is a solution that addresses Problem 1. Given a set of signed documents, it is timestamped by a single timestamp as follows. Each signed document is hashed. The digests are leaves of a Merkle Tree, whose root is timestamped by a Timestamp Authority (TSA). A copy of the timestamp is stored with each signed document. The archiver has to monitor only the lifetime of the TSA's certificate and the security of the used cryptographic algorithms. Although ERS addresses Problem 1, Problem 2 remains because the demand of timestamps is not reduced.

The amount of protection data is proportional to the length of a certification path [Martinez-Pel et al. 2008]. In that sense, *Nested Certificates* [Levi et al. 2004] address Problem 2 by reducing certification paths. Given a certification path $C$ of length $n$, a *shortcut* is created between two certificates $c_i, c_j \in C$, such that $0 \leq i < j + 1 < n$. The shortcut is a new certificate issued by the subject of $c_j$ and it asserts that the certificates $c_i, c_{i+1} ... c_{j-1}$ are trustworthy. Although attractive, this approach creates additional trusted entity. That is, using a nested certificate requires the trust in the shortcut issuer.

Considering that Certificate Revocation List (CRL) is the usual data structure to implement revocation in the conventional PKI, protection data can be reduced by using schemes that are more efficient than CRL in regard to Problem 2. A straightforward approach is to eliminate the need for revocation by using *Short-Term Certificates* [Rivest 1998]. That is, certificates that are trustworthy for short periods (e.g. a day),

 ©2012 SBC — Soc. Bras. de Computação

so that their probability of being revoked is almost zero. Nevertheless, those certificates are not practical for a CA because once the CA's certificate expires, all subordinate certificates have to be reissued. Moreover, *Online Certificate Status Protocol* (OCSP) [Myers et al. 1999] and *Novomodo* [Micali 2002] provide revocation data that is smaller and easier to evaluate than CRL. These solutions just mitigate Problem 2, because there is still the accumulation of timestamps and certificates. Yet, Problem 1 remains.

Solutions as [Moecke et al. 2010, Vigil et al. 2009, Adams et al. 2001b] use the help of an additional trusted entity. It verifies the semantic and mathematical correctness of a document's signature. If the verification succeeds, the trusted entity asserts that the signature is trustworthy. The assertion is signed by the trusted entity. The original protection data of the signed document is replaced by only the signed assertion and the trusted party's certificate. This replacement allows to address Problem 2. Problem 1 remains. Yet, the addition of a trusted entity creates a further point of failure in the PKI.

## 4. Cleaning the PKI up

In a conventional PKI, a Certification Authority (CA) asserts that the binding between a public key and an entity is trustworthy. This assertion is conveyed by a certificate. The CA defines a validity period for the certificate, after which the certificate expires and is no longer trustworthy. However, the certificate can be revoked at *any time* during the validity period. This possibility leads to the problems 1 and 2. To address them, we propose to *clean up the PKI*, as describe hereinafter. For each description, we provide a formal definition in Section 4.1. Moreover, we give an illustrated example of our proposal in Section 4.2

Given a conventional PKI (cf. Definition 1), which provides trustworthy certificates for the use of signed documents (cf. Definition 4), we clean the PKI up, yielding a *cleaned PKI*. To do so, the Root CA of the conventional PKI reissues each final certificate, but defining a *new* validity period during which the certificate *was indeed trustworthy*. The outcome of this process (cf. Definition 3) is a *cleaned PKI* (cf. Definition 2).

Our proposal addresses Problem 2 in the following way. Since the Root CA reissues final certificates of the conventional PKI, the intermediary CAs' certificates are not present in the cleaned PKI. Yet, the Root CA redefines the validity period of each final certificate in the cleaned PKI, excluding the period *after* the revocation of the certificate in the conventional PKI. For example, if a certificate was valid for $365$ days and was revoked on the $50^{th}$ day, then the Root CA reissues the certificate using a validity period of $49$ days. Since intermediary CAs' certificates and revocation data are no longer necessary after cleaning the PKI up, the size of *protection data* in a signed document is reduced, thereby addressing Problem 2.

We address Problem 1 by reissuing the certificate of a Timestamps Authority (TSA) as a final certificate, but applying the idea of *Off-line Timestamps* [Ansper et al. 2001]. That is, given a TSA and the timestamps that it has ever issued, the Root CA binds the timestamps with the reissued TSA's certificate. This binding *authenticates* the trustworthy timestamps, while preventing forged timestamps, which were created *after* the TSA expired, from being accepted as trustworthy. As a consequence, an archiver only retimestamps a signed document *before* the used cryptographic algorithm become insecure. This reduces the effort of the archiver, addressing Problem 1. For a

 ©2012 SBC — Soc. Bras. de Computação

detailed explanation, we refer the reader to Section 5.

### 4.1. Definitions

In the following, we formally define the concepts involved in the cleaned PKI. Note that we use *tuples* to define certificates, signed documents and protection data. Moreover, given a tuple $Z$ of $n$ elements, we denote the $k^{th}$ element as $\pi_n(Z)$, such that $1 \leq k \leq n$.

**Definition 1.** A *conventional PKI* is a directed graph $G = (V, A)$. The set $V$ comprises the public keys $p_i$, being $i \geq 0$ and $p_0$ assigned to the Root CA. The set $A$ comprises certificates $c_{j,k} = (p_j, p_k)$, such that $p_j, p_k \in V$ and the subject of $p_j$ creates (i.e. signs) $c_{j,k}$ for the subject of $p_k$. The graph $G$ is almost a tree, since there is a single cycle $c_{0,0}$ (a self-signed certificate for the Root CA). Each certificate $c_{j,k} \in A$ is qualified by a time interval $v_{j,k} = [i_{j,k}, e_{j,k}]$, being $i_{j,k}$ and $e_{j,k}$ the issuance and expiration dates for $c_{j,k}$ respectively, such that $i_{j,k} < e_{j,k}$. Moreover, given a certificate and a corresponding sibling, namely $c_{j,k}, c_{k,l} \in A$, then $v_{j,k} \cap v_{k,l} = v_{k,l}$. Furthermore, $c_{j,k} \in A$ *can* be revoked on a date $r_{j,k}$. If $c_{j,k}$ is revoked, then $r_{j,k} \in v_{j,k}$ and the corresponding descendants certificates $c_{t,u} \in A$ become *invalid* from $r_{j,k}$ on. Otherwise, $r_{j,k} > e_{j,k}$.

**Definition 2.** A *cleaned PKI* is a *conventional PKI* $G' = (V', A')$ (cf. Definition 1), such that $\forall c_{j,k} \in A'$, $j = 0, k \geq 0$. That is, all certificates are issued by the Root CA. Moreover, a certificate $c_{0,k} \in A'$ *cannot* be revoked anytime, i.e. it is always valid in $v_{0,k}$. Therefore, we eliminate the revocation dates $r_{j,k}$ from $G'$.

**Definition 3.** A *clean-up function* is a function $f_c : G \mapsto G'$, such that $G = (V, A)$ is a conventional PKI (cf. Definition 1) and $G' = (V', A')$ is a cleaned PKI (cf. Definition 2). The function $f_c$ receives as input $G$ and a date $u$, such that $u \in v_{0,0}$. The function $f_c$ works as presented in Algorithm 1. Line 11 prevents a forged certificate, i.e. a certificate created *after* the revocation of any of its ancestors. In line 12, a signature scheme is implicit in the creation of $c_{0,k}$, being more details given in Section 6.2. Note that $A' \cap A = c_{0,0}$ and $V' = V - \{p_i\}$, such that $p_i \in V$ and corresponds to an intermediary CA.

**Definition 4.** A *signed document* is a tuple $S_{d,i} = (d, \sigma_{d,i}, P_i)$, being $d$ a data object whose signature $\sigma_{d,i}$ is verifiable under public key $p_i \in V$, such that $G = (V, A)$ is a graph that represents a PKI (cf. definitions 1 and 2). The protection data is a tuple $P_i = (c_{k,i}, I_i, R_i, T_i)$ that comprises the certificate $c_{k,i} \in A$ for $p_i$, the set $I_i \subset A$ of certificates that are ancestors of $c_{k,i}$, the set $R_i$ of revocation status for $c_{k,i}$ and each certificate in $I_i$, a set $T_i$ of signed timestamps $S_{n,p}$, such that $T_p = \emptyset$. That is, a timestamp is a signed document that does not encapsulate further timestamps.

### 4.2. Example

Figure 1 depicts an example in which a conventional PKI $G$ (Figure 1(a)) is cleaned up, yielding the cleaned PKI $G'$ (Figure 1(b)). In $G$ there are a Root CA's certificate ($c_{0,0}$), intermediaries CAs' certificates ($c_{0,1}$ and $c_{0,2}$) and the final certificates ($c_{1,3}$,$c_{1,4}$,$c_{1,5}$,$c_{2,6}$,$c_{2,7}$ and $c_{2,8}$). The certificates $c_{0,1}$ and $c_{2,7}$ are revoked (see $\square$symbol) on dates $r_{0,1}$ and $r_{2,7}$ respectively.

On a date $u$, $G$ is cleaned up, yielding $G'$. Note that in $G'$ there is *neither* intermediary CA *nor* revocation. Also, see that a final certificate in $G'$ may have a shorter validity

 ©2012 SBC — Soc. Bras. de Computação

**Input**: A conventional PKI $G = (V, A)$, a set $T$ of digests, and a date $u$
**Output**: A cleaned PKI $G' = (V', A')$

```
1  V' ← {p_0}, such that p_0 ∈ V
2  A' ← {c_{0,0}}, such that c_{0,0} ∈ A
3  G' ← (V', A')
4  foreach leaf c_{j,k} ∈ A, such that i_{j,k} < u do
5      R ← {c}
6      R ← R ∪ {min(e_{j,k}, r_{j,k})}
7      foreach ancestor c_{l,m} of c_{j,k} do
8          R ← R ∪ {min(e_{l,m}, r_{l,m})}
9      end
10     e_{0,k} ← min(R)
11     if i_{j,k} < e_{0,k} then
12         c_{0,k} ← (p_0, p_k), such that p_0, p_k ∈ V
13         v_{0,k} ← [i_{j,k}, e_{0,k})
14         if p_k belongs to a TSA then
15             associate c_{0,k} to λ_k ∈ T
16         end
17         A' ← A' ∪ {c_{0,k}}
18     end
19 end
20 return G'
```

**Algorithm 1:** The clean-up function $f_c$.

period than the corresponding final certificate in $G$, depending on a revocation in $G$. That is, if a certificate in $G$ is revoked, it or its descendants is/are reissued in $G'$, but using a shorter validity period that excludes the period after the revocation. For instance, $c_{0,1}$ is revoked on date $r_{0,1}$ (Figure 1(a)), therefore $c_{0,3}, c_{0,4}, c_{0,5}$ (Figure 1(b)) have shorter validity periods than $c_{1,3}, c_{1,4}, c_{1,5}$ (Figure 1(a)). Yet, note that the certificates of intermediary CAs (namely $c_{0,1}$ and $c_{0,2}$) are not reissued, therefore they are not present in Figure 1(b).

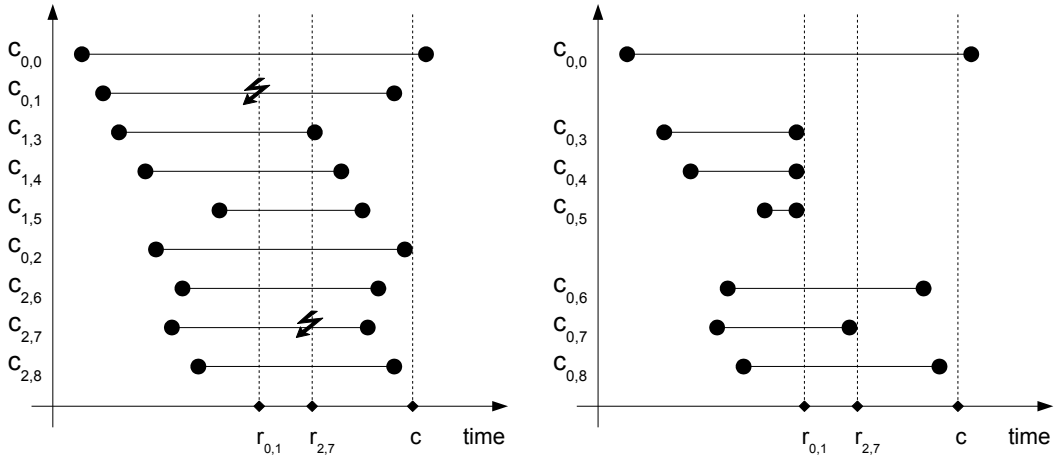
## 5. Timestamps

Timestamps are the most prominent means to guarantee proof-of-existence for data objects in a PKI. That is, timestamps allow us to infer a date on which a data object existed. This facility is necessary to sort data objects in chronological order. For instance, given the compromise of a private key $k_p$ on date $r_p$ and a signature $\sigma_p$ done by $k_p$, a timestamp on $\sigma_p$ on a date $t_{\sigma_p}$ allows us to check if $\sigma_p$ is trustworthy, i.e. if $\sigma_p$ was created *before* the compromise of $k_p$. In other words, if $t_{\sigma_p} < r_p$.

The use of timestamps requires the so-called Timestamp Authority (TSA), which is a trusted entity that provides *signed* timestamps with a *trustworthy* date value. That is, given a data object, the TSA provides a timestamp that binds the data object to the *current* value of the TSA's clock, which is protected and synchronized with a legal source of time.

The compromise of the private key of a TSA invalidates all its issued timestamps.

 ©2012 SBC — Soc. Bras. de Computação




(a) A conventional PKI $G$ in which certificates $c_1$ and $c_7$ are revoked at $r_{0,1}$ and $r_{2,7}$ respectively.

(b) A cleaned PKI $G'$, yielded by cleaning $G$ up.

**Figure 1. Comparison between conventional and cleaned PKIs.**

The reason is that an adversary is able to forge timestamps on behalf of the compromised TSA using *any* date value. As a countermeasure, the CA that issued the certificate of the compromised TSA notices all verifiers that the TSA is no longer trustworthy. That is, the TSA is revoked.

A CA is liable for the revocation of the certificates that the CA issues, but *until* these certificates expire. From this fact Problem 1 raises. For instance, an adversary may compromise the private key of a TSA *after* it goes out of business. In this case, the adversary is able to forge timestamps and verifiers cannot notice the forgery because the TSA's certificate is expired and no revocation status is available. To prevent such a forgery, we *timestamp* timestamps, a process that is end-less for the archiver of signed documents.

A solution to avoid Problem 1 is the destruction of the TSA's private key after the TSA finishes its operations. Nevertheless, proving that a data object was *properly* erased is still an open problem. Therefore, we *assume* that the key is properly erased. Given the number of possible TSAs in a PKI and respective security policies, this trust assumption *may* be prohibitive. Thus, we propose to reduce our assumption to the private key of the Root CA, which authenticates all timestamps issued *before* the PKI is cleaned up. Our proposal uses the idea of *Off-line Timestamps* [Ansper et al. 2001] and is detailed in the following.

Let $S_{n,p}$ be timestamps (cf. Definition 4) issued by a TSA whose certificate $c_{k,p} \in A$, such that $G = (V, A)$ is a conventional PKI (cf. Definition 1) and $n \geq 0$. Before a date $u$, when $G$ is cleaned up, the TSA builds a Merkle Tree [Merkle 1989] whose leaves are the digest of each $S_{n,p}$ and $\lambda_p$ is the root. Afterwards, the Root CA executes the clean up function $f_c$ (cf. Definition 3) in order to create the cleaned PKI $G' = (V', A')$ (cf. Definition 2). While issuing the certificate $c_{0,p} \in A'$ for the TSA (cf. line 15, Algorithm 1), the Root CA binds $\lambda_p$ to $c_{0,p}$. Since $c_{0,p}$ is dated by a validity period $v_{0,p} = [i_p, e_p)$, $\lambda_p$ and consequently all $S_{n,p}$ are timestamped. That is, we infer that

 ©2012 SBC — Soc. Bras. de Computação

$S_{n,p}$ existed *before* the date $i_p$ if we can reconstruct $\lambda_p$ using the digest of $S_{n,p}$ and the corresponding *authentication path* to $\lambda_p$.

Given a forged timestamp $S_{n+1,p}$ created on a date $t_f > i_p$, an adversary is not able to prove that $S_{n+1,p}$ is trustworthy. That is, the adversary is not able to find out a digest for $S_{n+1,p}$ and authentication path that allow him to reconstruct $\lambda_p$ as long as the cryptographic algorithms used to create $\lambda_p$ and $c_{0,p}$ are secure.

Note that we extended the trustworthiness of a timestamp beyond the expiration of the corresponding TSA's certificate, being now the lifetime of the used cryptographic algorithms the constraint. Thus, the archiver needs to monitor only the security of the cryptographic algorithms instead of the expiration of certificates. Since the number of the used cryptographic algorithms is *likely* to be smaller than the number of certificates in an archive, the effort of the archiver is reduced, addressing Problem 1. Furthermore, the lifetime of cryptographic algorithms *usually* lasts longer than the lifetime of final certificates (e.g. in ICP-Brasil). Thus, Problem 2 is addressed, since less timestamps are necessary, requiring then the storage and verification of less protection data.

## 6. Procedures

The procedures described in this section show how the definitions given in sections 4 and 5 are applied in a practical sense, involving the following players. The Root CA gathers certificates, revocation and timestamps (cf. Section 6.1). Next, the Root CA reissues final certificates and authenticates timestamps (Section 6.2), yielding a cleaned PKI. Given the certificates in a cleaned PKI, archivers replace certificates and revocation data within signed documents (cf. Section 6.3). A final user that uses archived and signed documents evaluates their trustworthiness by verifying their signatures (cf. Section 6.4).

### 6.1. Preparing to Clean up

The preparation for cleaning a conventional PKI up is performed by the Root CA and consists of two steps: a) gather all issued certificates in the PKI and the current revocation for each certificate; b) collect a representation of the issued timestamps in the conventional PKI; and c) verify the mathematical correctness of the signatures on the gathered certificates and revocation status. The outcome of executing this protocol is a graph $G = (V, A)$ representing all public keys and certificate in the conventional PKI (cf. Definition 1). In the following, we detail each step.

Gathering the issued certificates and revocation status requires visiting the repository of each Certification Authority in the PKI, starting from the Root CA. This is a well-known task that can be automated, therefore we just assume it is already implemented in a PKI and it returns a graph $G = (V, A)$ and the corresponding revocation status for each gathered certificate.

Collecting a representation of the issued timestamps in the conventional PKI represented by $G = (V, A)$ consists of the following. From each TSA whose certificate $c_{k,i} \in A$, the Merkle Tree's root $\lambda_i$ that represents the timestamps the TSA has ever issued (cf. Section 5) is queried. We assume this task is also implemented in the PKI and returns a set $T$ of Merkle Trees' roots, such that $\lambda_i \in T$

Since a certificate is a signed object, the mathematical correctness of the certificate's signature is necessary, but not sufficient, to ensure the certificate authenticity.

 ©2012 SBC — Soc. Bras. de Computação

Therefore, for each certificate $c_{j,k} \in A$, the signature on $c_{j,k}$ should be valid under the public key $p_j \in V$. A revocation status is usually signed, depending on how its authenticity is ensured. Considering that the revocation status $r_{j,k}$ for certificate $c_{j,k}$ is signed, then the signature on $r_{j,k}$ should be valid under the public key $p_j$.

Note that the semantic correctness for each $c_{j,k} \in A$ is not evaluated here, since the deployment of the clean-up function $f_c$ (cf. Definition 3) includes this task. That is, $f_c$ verifies validity periods and revocation status.

By finishing the above described steps, this procedure outcomes a graph $G = (V, A)$ whose certificates' mathematically correctness is ensured. This result is used in next procedure, which cleans $G$ up (cf. Section 6.2).

### 6.2. Reissuing Final Certificates

Given a graph $G = (V, A)$ representing the certificates in a conventional PKI gathered and partially verified in the preparation procedure and the set $T$ of Merkle Trees' roots that resumes the timestamps issued in the conventional PKI (cf. Section 6.1), the Root CA executes the reissuing procedure. This procedure outcomes a cleaned PKI $G' = (V', A')$ (cf. Definition 2) and consists of executing the clean-up function $f_c$ (cf. Definition 3), using $G$ and an arbitrary date $u$ as inputs. In the following, we detail $u$ and $f_c$.

Recall that $u$ defines the time reference for cleaning a PKI up, being relevant for revocation status and limiting the validity period of a reissued certificate. The choice of different values for $u$ and the corresponding implication are discussed in Section 8.

After $u$ is chosen, $f_c$ is executed by the Root CA. Note that the Root CA can be overloaded, depending on the number of final certificates in $G$ that will be reissued (cf. Algorithm 1). Therefore, we propose the use of Merkle Trees to optimize the signature of final certificates.

Instead of signing each certificate $c_{0,k} \in A'$, such that $G' = (V', A')$ is a cleaned PKI and $c_{j,k} \in A$ is a leaf in a conventional PKI $G = (V, A)$, the Root CA creates $c_{0,k}$ without a signature. If $p_k \in V$ is the public key of a TSA, $\lambda_k \in T$ is associated to $c_{0,k}$. After $f_c$ returns $G'$, a Merkle Tree is built, being $\lambda_{A'}$ the root and the leaves the digest of each certificate $c_{0,k} \in A'$. Then, a single signature $\sigma_{0,\lambda_{A'}}$ on $\lambda_{A'}$ is done by the Root CA, using its private key. Next, for each $c_{0,k}$ an authentication path to $\lambda_{A'}$ is calculated and attached to $c_{0,k}$ along with $\lambda_{A'}$.

Being all certificates in $G'$ signed, archivers can use them to replace the certificates of $G$ in a signed document. This procedure is described in Section 6.3.

### 6.3. Replacing Protection Data

The replacement of protection data is performed by an archiver for each signed document. This procedure consists of generating new and reduced protection data, which replaces the existing protection data in a signed document. In the following, we detail this procedure.

Given a conventional PKI $G = (V, A)$ (cf. Definition 1), consider that an archive comprises signed documents $S_{d,i} = (d, \sigma_{d,i}, P_i)$ (cf. Definition 4), being $P_i = (c_{k,i}, I_i, R_i, T_i)$ the protection data and $c_{k,i} \in A$ a signer's certificate. Moreover, there may exist signed timestamps $S_{S_{d,i},p} = (S_{d,i}, \sigma_{S_{d,i},p}, P_p)$ (cf. Definition 4) on $S_{d,i}$, being

 ©2012 SBC — Soc. Bras. de Computação

**Input**: A protection data $P_i = (c_{k,i}, I_i, R_i, T_i)$, a conventional $G = (V, A)$ and cleaned $G' = (V', A')$ PKIs
**Output**: A cleaned protection data $P'_i = (c_{0,i}, I'_i, R'_i, T'_i)$

1 $T'_i \leftarrow \emptyset$
2 **if** $T_i \neq \emptyset$ **then**
3 $\quad S_{S_{d,i},p} \leftarrow \pi_1(T_i)$
4 $\quad H_{\lambda_p, S_{S_{d,i},p}} \leftarrow$ get auth. path from the digest of $S_{S_{d,i},p}$ to $\lambda_p$
5 $\quad S_{S_{d,i},p} \leftarrow S_{S_{d,i},p} || H_{\lambda_p, S_{S_{d,i},p}}$
6 $\quad P_p \leftarrow \pi_4(S_{S_{d,i},p})$
7 $\quad P'_p \leftarrow$ execute Algorithm 2 using inputs $P_p$, $G$, and $G'$
8 $\quad \pi_4(S_{S_{d,i},p}) \leftarrow P'_p$
9 $\quad T'_i \leftarrow T'_i \cup \{S_{S_{d,i},p}\}$
10 **end**
11 $P'_i \leftarrow (c_{0,i}, \emptyset, \emptyset, T'_i)$, such that $c_{0,i} \in A'$
12 **return** $P'_i$

**Algorithm 2:** Creating a cleaned protection data.

$S_{S_{d,i},p} \in T_i$, $P_p = (c_{j,p}, I_p, R_p, T_p)$ the protection data and $c_{j,p} \in A$ a TSA's certificate. Also, $c_{k,i}$ and $c_{j,p}$ are final certificates, i.e. $k \neq i$, $j \neq p$ and $k, i, j, p > 0$.

After $G$ is cleaned up, a cleaned PKI $G' = (V', A')$ (cf. Definition 2) is obtained. Then, the archiver executes Algorithm 2, using the protection data $P_i$, $G$, and $G'$. The algorithm returns new protection data $P'_i$. Then, the archiver replaces $P_i$ by $P'_i$ in the signed document $S_{d,i}$.

Note that Algorithm 2 removes timestamps in $T_i$, but keeps the first timestamp. The reason is that multiple timestamps address certificate expiration, which is not an issue in the cleaned PKI (cf. Section 5). Therefore, the first timestamp is able to provide the signed document $S_{d,i}$ with proof-of-existence until the used cryptographic algorithms remain secure. We pointed out that Algorithm 2 does not consider a timestamp issued by a TSA in another conventional PKI $G'' \neq G$, however the algorithm can be easily adapted to do so. This is necessary in the long term, after the cryptographic algorithms are no longer secure in $G'$ and a further timestamp under a new PKI, which uses up to date cryptographic algorithms, is required.

The Root CA signs the Merkle Tree's root $\lambda_p$, which represents the authenticated timestamps $S_{S_{d,i},p}$ (cf. Section 6.2). To evaluate the trustworthiness of $S_{S_{d,i},p}$, verifiers need the authentication path $H_{\lambda_p, S_{S_{d,i},p}}$ from the digest of $S_{S_{d,i},p}$ to $\lambda_p$. To do so, there are lines 4 and 5 in Algorithm 2.

Protection data $P'_i$ requires less storage space than $P_i$, because $P'_i$ comprises neither intermediary CAs' certificates nor revocation information. That is, $I_i, R_i = \emptyset$. Moreover, we restrict the set $T'_i$ to a single timestamp, which also has a reduced protection data $P'_p$. By these reductions, we address problems 1 and 2.

The replacement of $P_i$ by $P'_i$ preserves the signer's and TSA's public keys. Also, the Root CA is the same in both PKIs. As a consequence, this procedure does not prevent

 ©2012 SBC — Soc. Bras. de Computação

the verification of signatures, as described in Section 6.4.

### 6.4. Verifying a Signed Document

The verification of a signed document is performed by a final user who wants to evaluate whether the content in the signed document is trustworthy. This procedure consists of evaluating the signatures in the signed document and corresponding timestamps, by checking the mathematical and semantic correctness of each signature (cf. Section 2). In the following, we detail this procedure.

Consider a signed document $S_{d,i} = (d, \sigma_{d,i}, P'_i)$ (cf. Definition 4), being $d$ the content whose trustworthiness is evaluated, $\sigma_{d,i}$ the signature on $d$, $P'_i = (c_{0,i}, \emptyset, \emptyset, T'_i)$ the *reduced* protection data, $c_{0,i} \in A'$ the signer's certificate, such that $G' = (V', A')$ is a cleaned PKI (cf. Definition 2), $T'_i = \{S_{S_{d,i},p}\}$ a set, $S_{S_{d,i},p} = (S_{d,i}, \sigma_{S_{d,i},p}, P'_p)$ (cf. Definition 4) a timestamp on $S_{d,i}$, $P'_p = (c_{0,p}, \emptyset, \emptyset, \emptyset)$ the *reduced* protection data and $c_{0,p} \in A'$ a TSA's certificate. Also, $c_{0,i}$ and $c_{0,p}$ are final certificates, i.e. $i \neq p$ and $i, p > 0$. There are the Merkle Tree's root $\lambda_{A'}$ associated to $c_{0,i}$ and $c_{0,p}$, and the authentication paths $H_{\lambda_{A'},i}$ and $H_{\lambda_{A'},p}$ associated to $c_{0,i}$ and $c_{0,p}$ respectively (cf. Section 6.2). Also, there are the Merkle Tree's root $\lambda_p$ associated to $c_{0,p}$ and an authentication path $H_{\lambda_p,S_{S_{d,i},p}}$ from the digest of $S_{S_{d,i},p}$ to $\lambda_p$ attached to $S_{S_{d,i},p}$ (cf. Section 6.3).

To verify $d$, the signature $\sigma_{d,i}$ is evaluated in terms of mathematical and semantic correctness. The mathematical verification consists of checking whether $\sigma_{d,i}$ is valid under $p_i$, being $p_i$ available in $c_{0,i}$. The semantic verification checks whether $c_{0,i}$ was valid on the date $t_p$ stated by the timestamp $S_{S_{d,i},p}$. This is ensured if the following conditions hold:

a) the signature on $c_{0,i}$ is valid under $p_0$;
b) the signature on $c_{0,p}$ is valid under $p_0$;
c) $\sigma_{S_{d,i},p}$ is valid under $p_{0,p}$;
d) $v_{0,i} \cap v_{0,0} = v_{0,i}$;
e) $v_{0,p} \cap v_{0,0} = v_{0,p}$;
f) $S_{S_{d,i},p}$ is *authenticated* by the Root CA in $c_{0,p}$; and
g) the timestamp date $t_p \in v_{0,i}$.

Note that in conditions a) to c) we verify the mathematical correctness of the signature on the certificates $c_{0,i}$ and $c_{0,p}$ and the timestamp $S_{S_{d,i},p}$, ensuring that these objects are not corrupted. Recall that a) and b) use the Merkle Tree $\lambda_{A'}$ and the authentication paths $H_{\lambda_{A'},c_{0,i}}$ and $H_{\lambda_{A'},c_{0,p}}$. Conditions d) and e) guarantee that the validity periods of $c_{0,i}$ and $c_{0,p}$ do not go beyond the lifetime of Root CA's certificate $c_{0,0}$. Conditions f) and g) prevent forgeries if the signer's or TSA's private key is compromised after the corresponding certificate is expired. Recall, that f) uses the Merkle Tree's root $\lambda_p$ the authentication path $H_{\lambda_p,S_{S_{d,i},p}}$.

## 7. Benefits

By switching from the conventional PKI to the cleaned PKI, we address problems 1 and 2. As a consequence, we simplify the preservation and use of signed documents, thereby benefiting the archivers and final users respectively.

 ©2012 SBC — Soc. Bras. de Computação

While in the conventional PKI the semantic correctness of a signature depends on certificates' validity periods, in the cleaned PKI only the security of cryptographic algorithms is important. The lifetime of a final certificate is *usually* shorter than cryptographic algorithms' lifetimes. Taking the Brazilian National PKI (a.k.a ICP-Brasil) as an example, a final certificate expires in 3 years [ITI 2008b] and the recommended signature suite SHA-256 with RSA-2048 is expected to be secure for 16 years [ITI 2008a] from now on. Since semantic correctness lasts longer in the cleaned PKI, less timestamps are used than in the conventional PKI in the long term. This benefits the archiver because he needs to request fewer timestamps. Yet, it is easier for the archiver to monitor some cryptographic algorithms' lifetimes than several certificates' lifetimes.

The fewer timestamps, the smaller protection data. This benefits the archiver by reducing the costs with storage space. The smaller data protection, the fewer signatures have to be evaluated (e.g. on timestamps, certificates and revocation data). Therefore, final users are benefited because less processing time is necessary to verify the trustworthiness of a signed document. Yet, the absence of revocation eliminates the need to download fresh revocation data each time a signed document is verified. As a consequence, final users can perform *off-line* verifications, i.e. without network access.

Unlike some related works (e.g. [Levi et al. 2004, Vigil et al. 2009, Adams et al. 2001b]), our proposal do not add an extra *trusted* entity, which would strengthen the trust assumption of a conventional PKI. The cleaned PKI has only two failure points, namely the Root CA and the cryptographic algorithms. That is, unless both remain secure (e.g. the Root CA is not compromised and the cryptographic algorithms are not suddenly broken), the cleaned PKI is secure.

## 8. Considerations & Future Work

All in all, we propose a new approach that *cleans up* the conventional PKI to facilitate the use of long-term signatures. We suggest removing intermediary CAs and revocation. To do so, the Root CA reissues final certificates defining validity periods that exclude revocation. The Root CA authenticates all trustworthy timestamps ever issued by any subordinate TSA. Reissued certificates remain trustworthy as long as the used cryptographic algorithms are secure. In the cleaned PKI, archivers use fewer timestamps than in the conventional PKI. Timestamps are only necessary to renew cryptographic algorithms. As a benefit, we reduce: a) the effort to preserve signed documents (Problem 1); and b) the storage space and processing time to handle signed documents (Problem 2). Yet, the cleaned PKI adds no further *trusted entity*.

The overall procedure of cleaning up a PKI is expensive in terms of processing time. The reasons are: a) for each final certificate, the mathematical and semantic correctness are verified; and b) a Merkle Tree, whose leaves are the digest of each final certificate, is created. The execution of both tasks can be delegated to multiple computers while the Root CA is off-line. We point out that the Root CA's private key is used once to sign the Merkle Tree's root. The Root CA does not sign each final certificate.

The replacement of protection data is prevented in the current signature schemes, e.g. XAdES [ESI 2008]. The reason is that these schemes force the signer the sign the purposed data object along with his certificate. Although the certificate can be replaced later, the mathematical correctness of the signature will be compromised.

 ©2012 SBC — Soc. Bras. de Computação

As future work, we suggest to work out the issue of replacing validation data. For instance, signature schemes could bind the signer's meta-data instead of the certificate to the signature, thereby allowing replacements. Another possibility is to explore the use of chameleon hash functions to replace certificates based on collisions [Le et al. 2008].

## Acknowledgements

The authors thank Daniel Cabarcas and the anonymous reviewers for the comments that helped to improve this manuscript. Also, the authors thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior for supporting this work.

## References

[Adams et al. 2001a] Adams, C., Cain, P., Pinkas, D., and Zuccherato, R. (2001a). Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161 (Proposed Standard).

[Adams et al. 2001b] Adams, C., Sylvester, P., Zolotarev, M., and Zuccherato, R. (2001b). Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols. RFC 3029 (Experimental).

[Ansper et al. 2001] Ansper, A., Buldas, A., Roos, M., and Willemson, J. (2001). Efficient long-term validation of digital signatures. In *Public Key Cryptography*, pages 402–415. Springer.

[Cooper et al. 2008] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard).

[ESI 2008] ESI (2008). Electronic Signatures and Infrastructures (ESI); Profiles of XML Advanced Electronic Signatures based on TS 101 903 (XAdES). Technical report, European Telecommunications Standards Institute.

[Gondrom et al. 2007] Gondrom, T., Brandner, R., and Pordesch, U. (2007). Evidence Record Syntax (ERS).

[Gutmann 2002] Gutmann, P. (2002). Pki: it's not dead, just resting. *Computer*, 35(8):41–49.

[Huhnlein et al. 2009] Huhnlein, D., Korte, U., Langer, L., and Wiesmaier, A. (2009). A comprehensive reference architecture for trustworthy long-term archiving of sensitive data. In *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, pages 1–5. IEEE.

[ITI 2008a] ITI (2008a). *Declaração de Práticas de Certificação da Autoridade Certificadora Raiz da ICP-Brasil*. Instituto Nacional de Tecnologia da Informação, Brasília, v.4.0 edition. DOC-ICP-01.

[ITI 2008b] ITI (2008b). *Requisitos Mínimos para as Políticas de Certificado na ICP-Brasil*. Instituto Nacional de Tecnologia da Informação, Brasília, v.3.0 edition. DOC-ICP-04.

[ITU-T 2005] ITU-T (2005). Recommendation X.509 (08/2005) - Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. Technical report, ITU-T.

 ©2012 SBC — Soc. Bras. de Computação

[Kohnfelder 1978] Kohnfelder, L. M. (1978). Towards a practical public-key cryptosystem. Technical report, Massachusetts Institute of Technology.

[Le et al. 2008] Le, D., Bonnecaze, A., and Gabillon, A. (2008). A secure round-based timestamping scheme with absolute timestamps (short paper). *Information Systems Security*, pages 116–123.

[Levi et al. 2004] Levi, A., Caglayan, M., and Koc, C. (2004). Use of nested certificates for efficient, dynamic, and trust preserving public key infrastructure. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):21–59.

[Martinez-Pel et al. 2008] Martinez-Pel, R., Satiz, C., Rico-Novella, F., and Forn, J. (2008). Efficient Certificate Path Validation and Its Application in Mobile Payment Protocols. *2008 Third International Conference on Availability, Reliability and Security*, pages 701–708.

[Merkle 1989] Merkle, R. C. (1989). A Certified Digital Signature. In Brassard, G., editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer.

[Micali 2002] Micali, S. (2002). Scalable certificate validation and simplified pki management. In *1st Annual PKI Research Workshop*, page 15.

[Moecke et al. 2010] Moecke, C. T., Custódio, R. F., Kohler, J. G., and Carlos, M. C. (2010). Uma ICP baseada em certificados digitais autoassinados. In *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 91–104, Fortaleza. SBSEG.

[Myers et al. 1999] Myers, M., Ankney, R., Malpani, A., Galperin, S., and Adams, C. (1999). X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard).

[National Institute of Standards and Technology 2007] National Institute of Standards and Technology (2007). *Recommendation for Key Management – Part 1: General (Revised)*.

[Rivest 1998] Rivest, R. (1998). Can we eliminate certificate revocation lists? In *Financial Cryptography*, pages 178–183. Springer.

[Vigil et al. 2009] Vigil, M. A. G., da Silva, N., Moraes, R., and Custódio, R. F. (2009). Infra-estrutura de Chaves Públicas Otimizada: Uma ICP de Suporte a Assinaturas Eficientes para Documentos Eletrônicos. In *IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, Campinas.

 ©2012 SBC — Soc. Bras. de Computação

# Mitigando Ataques de Egoísmo e Negação de Serviço em Nuvens via Agrupamento de Aplicações

**Daniel Stefani Marcon, Miguel Cardoso Neves, Rodrigo Ruas Oliveira, Luciana Salete Buriol, Luciano Paschoal Gaspary, Marinho Pilla Barcellos**

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{daniel.stefani, mcneves, ruas.oliveira, buriol, paschoal, marinho}@inf.ufrgs.br

***Resumo.*** *Na computação em nuvem, locatários consomem, sob demanda, recursos de hardware e software oferecidos por um provedor remoto. Entretanto, o compartilhamento da rede interna da nuvem por todos os locatários, aliado à falta de isolamento entre fluxos de dados decorrente do uso dos protocolos TCP e UDP, possibilita a ocorrência de ataques de egoísmo e negação de serviço. Os algoritmos de alocação atuais não impedem que a disponibilidade dos recursos de rede seja afetada por ataques. Este artigo propõe uma estratégia para a alocação de aplicações de locatários que visa mitigar o impacto de ataques de egoísmo e negação de serviço na rede interna da nuvem. A ideia chave, inédita na literatura científica, consiste no agrupamento de aplicações em infraestruturas virtuais considerando níveis de confiança mútua entre os locatários. Resultados de avaliações demonstram que a estratégia proposta é capaz de oferecer proteção contra ataques de egoísmo e negação de serviço com pouco ou nenhum custo extra.*

***Abstract.*** *Cloud computing is a model where tenants consume on-demand hardware and software resources from a remote provider. However, the sharing of the internal network by all tenants, combined to the lack of data flow isolation due to the use of TCP and UDP, allows the occurrence of selfish and denial of service attacks. The current allocation algorithms do not prevent the availability of network resources to be affected by such attacks. In this paper, we propose a strategy for the allocation of tenants applications, which aims at mitigating the impact of selfish and denial of service attacks in the cloud internal network. The key, novel idea is to group applications into virtual infrastructures considering the mutual trust between pairs of tenants. Evaluation results show that the proposed strategy is able to offer protection against attacks of selfishness and DoS with little or no extra cost.*

## 1. Introdução

Computação em nuvem é um paradigma que possibilita aos locatários consumirem, sob demanda e de maneira elástica, recursos de hardware e software oferecidos por um provedor remoto. Nesse modelo, os provedores, com a finalidade de reduzir custos operacionais e oferecer escalabilidade de acordo com a demanda, implementam *datacenters* de nuvem como ambientes compartilhados altamente multiplexados, com diferentes aplicações coexistindo sobre os mesmos recursos físicos [Armbrust et al. 2010].

 ©2012 SBC — Soc. Bras. de Computação

Entretanto, não há uma abstração ou mecanismo disponível para capturar os requisitos de rede das interações entre as máquinas virtuais (VMs) alocadas [Guo et al. 2010]. Ademais, mecanismos como o controle de congestionamento do TCP e suas variantes (incluindo TFRC e DCCP), apesar de serem escaláveis e proporcionarem alta utilização dos recursos, não isolam de forma robusta o tráfego de dados de aplicações distintas [Abts and Felderman 2012]. Dessa forma, as VMs de uma mesma aplicação se comunicam em um ambiente (rede da nuvem) não controlado, compartilhado por todos os locatários. Consequentemente, locatários egoístas podem implementar suas aplicações de forma a levar vantagem indevida sobre outros usuários, através da utilização de versões do TCP com controle de congestionamento mais brando, e locatários maliciosos podem lançar ataques de negação de serviço, por exemplo gerando tráfego espúrio dentro da nuvem [Shieh et al. 2011].

Tais ataques prejudicam tanto locatários quanto provedores. Os locatários pagam pela utilização dos recursos contratados mesmo quando estão impossibilitados de utilizá-los devido a um ataque. Já os provedores possuem perda de receita [Greenberg et al. 2009], visto que a falta de disponibilidade da rede, causada por ataque de DoS, reduz o *throughput* da nuvem [Shieh et al. 2011].

Visando alocar recursos com garantias de disponibilidade de rede, as abordagens presentes na literatura focam em consolidação de VMs [Breitgand and Epstein 2012, Meng et al. 2010, Wang et al. 2011] ou no compartilhamento da rede baseado em pesos [Shieh et al. 2011, Lam and Varghese 2010]. Entretanto, elas não impedem ataques de egoísmo e de DoS, visto que a utilização dos recursos da rede de uma aplicação depende das demais aplicações da nuvem.

Nesse contexto, o presente artigo propõe uma estratégia de alocação de recursos para provedores de IaaS (*Infrastructure as a Service*), apresentada como um modelo de otimização, que visa mitigar ataques de egoísmo e de comportamento malicioso na rede interna da nuvem. Diferentemente de trabalhos anteriores, investiga-se uma estratégia baseada no agrupamento de aplicações em infraestruturas virtuais[1] (VIs), que são capazes de oferecer um certo nível de isolamento entre aplicações. São estudadas diferentes estratégias de agrupamento de locatários, considerando o nível de confiança mútuo e os requisitos de rede (largura de banda) das suas aplicações.

O agrupamento visa prover isolamento do tráfego de rede entre aplicações provenientes de locatários mutuamente não confiáveis. Além disso, o agrupamento permite fornecer isolamento entre o tráfego de aplicações com requisitos de largura de banda distintos, minimizando a interferência nociva que pode ser causada ao misturar tráfego com diferentes características [Shieh et al. 2011, Abts and Felderman 2012]. Além da proteção contra os ataques supracitados, o isolamento/agrupamento propiciado pela estratégia proposta pode ser benéfico contra outros tipos de ataques, como ataques contra a privacidade dos dados dos locatários [Ristenpart et al. 2009]. Os mesmos serão explorados em trabalhos futuros. O presente artigo tem as seguintes contribuições:

- discussão e avaliação do impacto de ataques de egoísmo e de negação de serviço no contexto de redes internas de nuvens;
- proposta de uma estratégia inédita para alocação de recursos, baseada no agrupamento de aplicações em VIs considerando níveis de confiança mútua

[1] O termo infraestrutura virtual é utilizado para representar uma rede virtual com as suas máquinas virtuais.

 ©2012 SBC — Soc. Bras. de Computação

entre os locatários, e aplicável a diferentes topologias de datacenters, tais como Fat-Tree [Greenberg et al. 2009] e VL2 [Al-Fares et al. 2008];
- avaliação analítica da estratégia proposta perante diferentes níveis de agrupamento e a comparação da mesma com um esquema padrão sem agrupamento.

O restante desse artigo está organizado da seguinte maneira. A Seção 2 apresenta os principais trabalhos relacionados. A Seção 3 define o modelo de ataque considerado e a Seção 4 define as formulações básicas utilizadas no restante do artigo. Na Seção 5, a estratégia de alocação de recursos é apresentada. Na Seção 6, o modelo de alocação proposto é avaliado e, por fim, as considerações finais são apresentadas na Seção 7.

## 2. Trabalhos Relacionados

As redes internas das nuvens atuais são compartilhadas por todos os locatários, honestos ou não. Apesar disso, não há um mecanismo disponível para capturar e controlar os recursos de rede utilizados pelas VMs alocadas [Grobauer et al. 2011].

Nesse sentido, Liu [Liu 2010] descreve como um atacante pode obter informações sobre a topologia da rede da nuvem, com a finalidade de realizar um ataque de negação de serviço. Já Ristenpart et al. [Ristenpart et al. 2009] conduziram um estudo de caso na Amazon EC2 (*Elastic Compute Cloud*), mostrando diversas vulnerabilidades do ambiente. Os autores relatam que não possuem nenhum dado privilegiado da nuvem, tendo conhecimento apenas das informações divulgadas publicamente pela Amazon. Desse modo, eles assumem que os atacantes são clientes comuns da nuvem. Eles demonstram que a EC2 pode ser mapeada de acordo com as suas zonas de disponibilidade e que é possível verificar a co-residência entre VMs, inclusive listando diversas possibilidades de efetuar tal verificação. O referido trabalho mostra que um locatário egoísta e/ou malicioso, conforme abordado no presente artigo, pode realizar diversos tipos de ataques à nuvem, inclusive de DoS na rede.

Apesar dessas vulnerabilidades, os algoritmos de alocação de recursos empregados atualmente em nuvens utilizam *round-robin* entre servidores ou entre *racks*, considerando somente recursos computacionais (processamento, memória e armazenamento) [Kitsos et al. 2012]. Contudo, torna-se necessário considerar também os recursos de rede, a fim de mitigar ataques de egoísmo e de comportamento malicioso.

O problema de segurança na rede interna da nuvem não é facilmente resolvível. Nesse contexto, a alocação de recursos em nuvens ciente dos recursos de rede representa um grande desafio de pesquisa. As abordagens presentes na literatura focam em consolidação de VMs ou no compartilhamento da rede por meio de pesos. Abordagens baseadas em consolidação de VMs [Meng et al. 2010, Wang et al. 2011, Breitgand and Epstein 2012] propõem que a alocação seja executada por meio da adaptação do problema de otimização *bin packing* [Goel and Indyk 1999]. Contudo, elas não impedem que o tráfego de aplicações egoístas ou maliciosas interfira no tráfego de aplicações honestas. Já Shieh et al. [Shieh et al. 2011] e Lam e Varghese [Lam and Varghese 2010] propõem um esquema de compartilhamento de largura de banda baseado em pesos. Esse esquema, porém, não impede que aplicações maliciosas realizem ataques de egoísmo e de negação de serviço na rede, visto que a largura de banda utilizada por uma aplicação na rede é dependente das outras aplicações da nuvem.

 ©2012 SBC — Soc. Bras. de Computação

Em outra gama de trabalhos, é explorada a alocação de tarefas em grades computacionais respeitando restrições de confiança entre a tarefa e os recursos da grade computacional [Costa and Carmo 2007]. A estratégia apresentada nesse artigo, em uma perspectiva distinta, explora as relações de confiança mútuas entre os locatários da nuvem.

De forma geral, os recursos da rede interna de *datacenters* frequentemente representam o gargalo quando comparados aos recursos computacionais [Greenberg et al. 2009]. Esse gargalo, aliado à falta de mecanismos para o controle do compartilhamento de recursos na rede, tende a facilitar ataques de egoísmo e comportamento malicioso. Nesse sentido, a estratégia apresentada nesse artigo visa mitigar tais ataques, sendo ciente tanto de recursos de rede quanto de recursos computacionais. Adicionalmente, a utilização de VIs diminui a eficácia dos métodos utilizados por Ristenpart et al. [Ristenpart et al. 2009] para a verificação de co-residência, visto que aplicações mutuamente não confiáveis são alocadas em redes virtuais distintas.

## 3. Modelo de Ataque

Similarmente à Ristenpart et al. [Ristenpart et al. 2009], considera-se que um locatário egoísta e/ou malicioso possui os mesmos privilégios dos demais locatários da nuvem. Usuários egoístas utilizam versões mais agressivas do TCP (*i.e.*, versões fora dos padrões, cujo código foi modificado para realizar controle de congestionamento mais brando) com o objetivo de aumentar a vazão de rede das suas VMs. Já locatários maliciosos tem por objetivo realizar um ataque de DoS em um alvo previamente definido. Nesse caso, o locatário emprega os métodos descritos por Ristenpart et al. [Ristenpart et al. 2009] para posicionar as suas VMs próximas ao alvo. Consequentemente, as VMs desse locatário podem enviar uma quantidade suficiente de tráfego para sobrecarregar o destinatário (alvo), um enlace pertencente ao caminho ou um gargalo da rede [Jensen et al. 2009]. No presente trabalho, considera-se somente o ataque de um locatário. Entretanto, ataques de conluio são possíveis, através de múltiplas identidades.

Os ataques de egoísmo e de DoS são efetuados através do aumento do número de fluxos de dados, explorando a falta de isolamento entre fluxos devido ao uso do TCP [Shieh et al. 2011], e pelo envio de grandes fluxos por meio do protocolo UDP. Considerando que a rede interna da nuvem é compartilhada por todos os locatários, o ataque pode afetar um grande número de aplicações.

O ataque precisa ser lançado por um "insider", ou seja, um locatário registrado na nuvem, e que pode portanto ser em tese responsabilizado. Entretanto, não há requisitos estritos para criar contas em nuvens e, mesmo que houvesse, o ataque poderia ser efetuado através de uma conta comprometida [Greenberg et al. 2008]. Além disso, a detecção de ataques não é simples. Tráfego malicioso pode simular um tráfego honesto, dificultando a distinção entre os dois tipos, e os esquemas de ofuscação, utilizados pelos provedores para ocultar a localização dos recursos na nuvem, não são capazes de parar um adversário persistente [Shieh et al. 2011].

## 4. Modelo de Sistema

Com base nas questões de segurança apresentadas anteriormente, é proposta uma estratégia na forma de um modelo de otimização para a alocação de aplicações de

 ©2012 SBC — Soc. Bras. de Computação

locatários que visa mitigar o impacto de ataques de egoísmo e negação de serviço na rede interna da nuvem. A execução desses ataques é facilitada atualmente na nuvem, visto que os provedores não cobram pelo tráfego de dados na rede interna [Guo et al. 2010].

Esta seção define as formulações básicas utilizadas para a modelagem da estratégia. As notações são representadas por meio das seguintes regras: $i$) sobrescritos $s$, $v$ e $r$ indicam entradas relacionadas à rede do substrato físico da nuvem, às infraestruturas virtuais e às requisições dos locatários, respectivamente; $ii$) subscritos são índices provenientes de atributos, variáveis ou elementos de um conjunto. O sistema de notação utilizado é similar ao empregado por Chowdhury et al. [Chowdhury et al. 2009].

### 4.1. Requisições de Aplicações

Cada requisição de aplicação $a \in A^r$, proveniente de um locatário, é definida pela tupla $(M_a^r,\ Band_a^r)$. O número de máquinas virtuais requisitadas é representado por $M_a^r$. Para facilidade de exposição do modelo desenvolvido, os detalhes das VMs são abstraídos e todas elas são consideradas iguais (consomem a mesma quantidade de recursos de CPU, memória e armazenamento). De modo similar a Ballani et al. [Ballani et al. 2011], uma requisição é estendida para especificar, além de recursos computacionais, recursos de rede. A largura de banda disponível para uma VM comunicar-se com outra VM da mesma aplicação é indicada por $Band_a^r$, que é um número real positivo ($Band_a^r \in \mathbb{R}^+$).

Ademais, cada aplicação $a \in A^r$ possui ou não confiança nas outras aplicações da nuvem, de acordo com as relações mútuas entre os locatários das mesmas. A confiança é representada por $T_{a_i,a_j}^r$, que indica se a aplicação $a_i$ confia na aplicação $a_j$. Com o objetivo de simplificar a avaliação (mas sem prejuízo ao modelo), no presente trabalho assume-se que as relações de confiança são diretas, binárias e simétricas. Em outras palavras, um locatário confia ou não em um outro com quem ele interage e, se confia, então é recíproco; presume-se que um esquema padrão como PGP [Zimmermann 1995] pode ser usado para o estabelecimento de confiança mútua.

### 4.2. Infraestruturas Virtuais

O conjunto de infraestruturas virtuais é indicado por $I^v$. Cada elemento $i \in I^v$ é representado por um grafo bidirecional valorado $G_i^v = (S_i^v, M_i^v, E_i^v, Band^v, Oversub^v)$, sendo seus elementos definidos a seguir. O conjunto de dispositivos de rede (*switches*) de $i$ é indicado por $S_i^v$. O conjunto de máquinas virtuais de $i$, tipicamente entre 20 e 40 por *rack* [Greenberg et al. 2009], é indicado por $M_i^v$ e o conjunto de enlaces virtuais por $E_i^v$. Cada enlace $e^v = (w_i, w_j) \in E_i^v$ conecta os nodos $w_i$ e $w_j$ ($w_i, w_j \in S_i^v \cup M_i^v$). Além disso, cada enlace $e^v \in E_i^v$ possui uma largura de banda bidirecional $Band^v(e^v) \in \mathbb{R}^+$ e um fator de sobrecarga $Oversub^v(e^v) \in \mathbb{Z}^+$ empregado pelo provedor aos recursos de rede. Além disso, o subconjunto de *switches Top-of-Rack* (ToR), retornado pela função $ToR(S_i^v)$, é representado por $R_i^v$ ($R_i^v \subset S_i^v$).

### 4.3. Infraestrutura Física

De forma similar a Guo et al. [Guo et al. 2010], o substrato físico da nuvem é modelado considerando servidores, dispositivos de rede e enlaces. Ele é representado por

 ©2012 SBC — Soc. Bras. de Computação

um grafo bidirecional valorado $G^s = (S^s\ , M^s\ , E^s\ , Band^s\ , Slots^s\ , Cost^s\ , Cap^s)$, no qual $S^s$ é conjunto de dispositivos de rede (*switches*), $M^s$ é o conjunto de servidores e $E^s$ é o conjunto de enlaces. Cada servidor $m^s \in M^s$ possui um número de *slots*[2] ($Slots^s(m^s) \in \mathbb{Z}^+$). Cada *switch* $s^s \in S^s$ possui um número máximo de *switches* virtuais que ele pode hospedar ($Cap^s(s^s) \in \mathbb{Z}^+$) e possui um custo ($Cost^s(s^s) \in \mathbb{R}^+$) para hospedar um *switch* virtual. O custo é proporcional à importância do *switch* na rede, ou seja, *switches* localizados em níveis mais próximos do núcleo da rede possuem maior custo para hospedar *switches* virtuais. Cada enlace $e^s = (w_i, w_j) \in E^s \mid w_i, w_j \in S^s \cup M^s$ entre os nodos $w_i$ e $w_j$ é associado a uma capacidade (largura de banda) bidirecional $Band^s(e^s) \in \mathbb{R}^+$.

## 5. Estratégia de Alocação de Recursos

A estratégia apresentada nesse artigo para a alocação de recursos visa mitigar ataques de egoísmo e comportamento malicioso na rede interna da nuvem. Diferentemente de trabalhos anteriores, o objetivo é alcançado por meio do agrupamento de aplicações em infraestruturas virtuais considerando a confiança mútua entre os locatários e a minimização do tráfego de rede gerado pelas interações entre as VMs da mesma aplicação.

Entretanto, existe um desafio fundamental a ser enfrentado: o problema de alocação de recursos em nuvem considerando enlaces de rede com largura de banda limitada é NP-Difícil [Guo et al. 2010]. Por essa razão, dividimos o problema em dois sub-problemas ou etapas menores, propomos uma estratégia de alocação ótima para cada um, e combinamos os resultados. A primeira etapa distribui e mapeia as aplicações em infraestruturas virtuais, enquanto a segunda é responsável por alocar cada infraestrutura virtual no substrato físico da nuvem.

Neste trabalho, o algoritmo para a formação do conjunto de infraestruturas virtuais ($I^v$) é abstraído e foca-se na alocação de aplicações nas VIs segundo critérios de segurança e desempenho e no mapeamento das VIs no substrato físico. Dessa forma, o conjunto de VIs é utilizado como entrada para o modelo de otimização. Apesar disso, sugere-se um um critério para formação de VIs, que poderia ser usado pelo provedor. As VIs podem ser formadas com base em informações obtidas a partir de serviços de monitoramento de recursos da nuvem (por exemplo, o Amazon CloudWatch[3]). O uso de informações pertencentes ao histórico da utilização de recursos sobre um grande período de tempo tende a facilitar e a melhorar a qualidade do processo de alocação [Meng et al. 2010]. Além disso, a topologia de rede de cada VI não é restrita pela topologia da infraestrutura física, visto que a heterogeneidade de topologias pode ser tratada no mapeamento das VIs no substrato físico [Chowdhury et al. 2009].

De modo geral, o processo completo é representado na Figura 1, que indica cada etapa por meio de uma função. A primeira etapa é representada por $\mathcal{F} : A^r \rightarrow I^v$, enquanto a segunda é denotada por $\mathcal{G} : I^v \rightarrow G^s$. A função $\mathcal{H} : A^r \rightarrow G^s$ realiza o mapeamento direto de aplicações na infraestrutura física da nuvem, mostrando como esse processo é efetuado atualmente em nuvens. Dessa forma, pode-se dizer que $\mathcal{H}$ é uma composição de $\mathcal{F}$ e $\mathcal{G}$.

[2]Como assume-se que todas as VMs são iguais, ou seja, consomem a mesma quantidade de recursos dos servidores, os *slots* também são considerados iguais.

[3]http://aws.amazon.com/cloudwatch/

 ©2012 SBC — Soc. Bras. de Computação

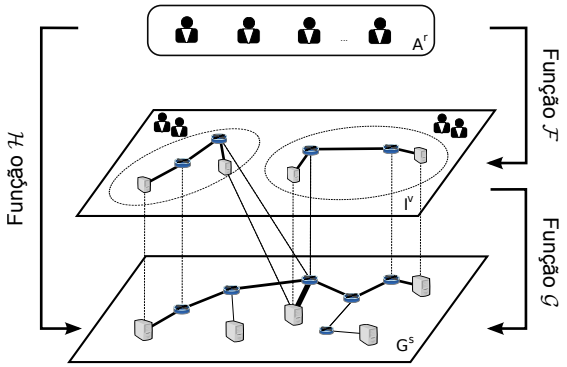



**Figura 1. O processo de alocação de recursos na nuvem.**

Apesar da estratégia ser composta por duas etapas, ambas podem ser executadas simultaneamente, visto que a saída de uma não interfere na entrada da outra. Isso implica que o tempo necessário para a conclusão do processo é o tempo de execução da etapa mais longa. Nesta seção, primeiramente é apresentada a função $\mathcal{F}$ e, após, a função $\mathcal{G}$.

### 5.1. Mapeamento de Aplicações em Infraestruturas Virtuais

A função $\mathcal{F}$ tem por objetivo mapear as aplicações em VIs considerando os níveis mútuos de confiança entre os locatários das mesmas. A função de mapeamento utiliza como dados de entrada as requisições de aplicações na nuvem (Seção 4.1) e o conjunto de infraestruturas virtuais (Seção 4.2).

As principais variáveis relacionadas ao modelo de otimização são:

- $F_{a,i,(w_1,w_2),p} \in \mathbb{R}^+$: indica o tamanho do fluxo total de dados para a aplicação $a \in A^r$ no enlace $(w_1, w_2) \in E_i^v \mid w_1, w_2 \in S_i^v$ para a comunicação entre o par de *racks* $p = (r_1, r_2) \mid r_1, r_2 \in R_i^v$ e $r_1 \neq r_2$ da VI $i \in I^v$. O valor desse fluxo, definido com base no número de VMs da aplicação $a$ nos *racks* $r_1$ e $r_2$, será explicado posteriormente (Equação 2);
- $g_{a,i,r,m} \in \mathbb{B}$: indica se a VM $m \in M_i^v$ do *rack* $r \in R_i^v$ da VI $i \in I^v$ foi alocada para a aplicação $a \in A^r$;
- $G_{a,i}$: indica se a aplicação $a \in A^r$ está na VI $i \in I^v$;
- $h_{a,i,(w_1,w_2),p} \in \mathbb{B}$: indica se a aplicação $a \in A^r$ utiliza o enlace $(w_1, w_2) \in E_i^v \mid w_1, w_2 \in S_i^v$ para a comunicação entre o par de *racks* $p = (r_1, r_2) \mid r_1, r_2 \in R_i^v$ e $r_1 \neq r_2$ da VI $i \in I^v$;
- $H_{i,a_1,a_2}$: indica se as aplicações $a_1, a_2 \in A^r$ estão alocadas na VI $i \in I^v$.

A função objetivo do problema de otimização dessa etapa, apresentada na Equação 1, busca minimizar uma penalização imposta por alocar aplicações provenientes de locatários mutuamente não confiáveis na mesma VI. A penalização considera as características das aplicações (número de VMs e largura de banda) mutuamente não confiáveis. Por exemplo, a aplicação $a_1$ possui 20 VMs, cada uma com largura de banda de 200 Mbps. Caso a aplicação $a_2$ não confie em $a_1$, a função objetivo gera uma penalização de $a_1$ em $a_2$, cujo valor é $20 \times 200 = 4000$. Essa fórmula também endereça as diferenças entre as aplicações dos locatários. Ou seja, há maior risco de um locatário atacar outro, se não houver confiança entre ambos, caso ele requisite uma aplicação composta por 50 VMs e 100 Mbps de largura de banda para cada VM em detrimento de uma aplicação composta por uma única VM com 100 Mbps.

 ©2012 SBC — Soc. Bras. de Computação

$$Z = \text{Min} \sum_{i \in I^v} \sum_{a_1 \in A^r} \sum_{a_2 \in A^r} ((1 - T^r_{a_1,a_2}) * H_{i,a_1,a_2} * M^r_{a_1} * Band^r_{a_1}) \quad (1)$$

As restrições do problema especificam o mapeamento das aplicações nas VIs, de acordo com os recursos computacionais e de rede disponíveis. Na rede, o tráfego de comunicação entre VMs localizadas no mesmo *rack* não possui custo, visto que ele permanece interno ao *rack* e só utiliza os enlaces que ligam as VMs ao *switch* ToR. Entretanto, o tráfego entre VMs de *racks* distintos possui custo, que é definido pela largura de banda consumida e pelos enlaces utilizados.

Desse modo, as VMs de cada aplicação são alocadas em *grupos*, de modo semelhante a Ballani et al. [Ballani et al. 2011]. Um grupo é composto pelo conjunto de VMs da mesma aplicação localizadas no mesmo *rack*. Por isso, visa-se criar o menor número possível de grupos de VMs, ou seja, alocar as VMs da mesma aplicação próximas umas das outras, o que reduz o tráfego de dados na rede e tende a dificultar ataques de egoísmo e DoS.

Com essa abordagem, é necessário garantir que haja largura de banda disponível para a comunicação entre os grupos de VMs. Por exemplo, considerando a existência de dois grupos $G_{a,1}$ e $G_{a,2}$ de VMs da aplicação $a \in A^r$ e que a comunicação ocorra em um caminho virtual $P^v$, todos os enlaces desse caminho devem possuir a largura de banda necessária disponível. Uma única VM da aplicação $a$ não pode enviar ou receber dados a uma taxa maior que a requisitada ($Band^r_a$). Dessa forma, o tráfego entre dois grupos é limitado à largura de banda exigida pelo menor dos grupos: $\min(|G_{a,1}|, |G_{a,2}|) * Band^r_a$. A Figura 2 exemplifica uma situação em que duas aplicações possuem dois grupos de VMs cada uma. Nesse caso, é necessário garantir a largura de banda na rede para a comunicação entre a dupla de grupos de VMs de cada aplicação. A restrição para a comunicação de um grupo com os outros grupos da aplicação $a$ é formulada da seguinte maneira:

$$B_{a,g_i} = \min \left( |g_i| * Band^r_a, \sum_{g \in G^r_a, g \neq g_i} |g| * Band^r_a \right) \qquad \forall g_i \in G^r_a \quad (2)$$

onde $G^r_a$ é o conjunto de grupos da aplicação $a$.

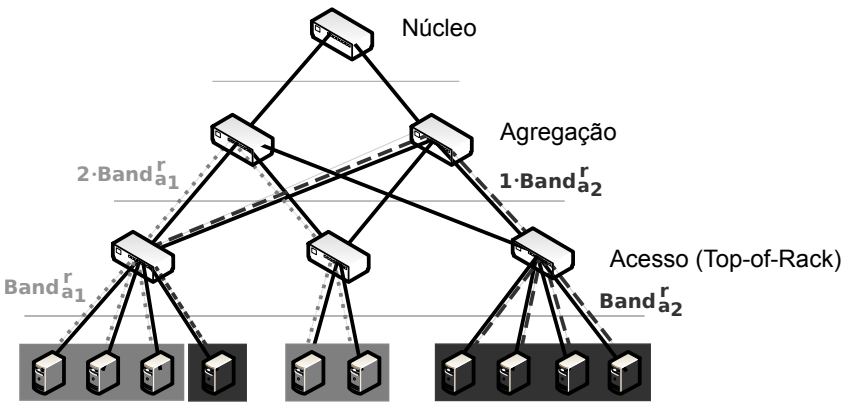



**Figura 2. Comunicação em rede entre grupos de VMs.**

 ©2012 SBC — Soc. Bras. de Computação

Em *datacenters*, o protocolo *Spanning Tree* assegura que o tráfego entre máquinas dentro da mesma camada dois da rede é encaminhado ao longo de uma árvore de expansão. Os dispositivos de rede da nuvem geralmente são interconectados com uma malha de enlaces, cuja utilização é balanceada através do uso de múltiplos caminhos de mesmo custo (*Equal-Cost Multipath* - ECMP) [Abts and Felderman 2012]. Dada a quantidade de multiplexação sobre a malha de enlaces e o número limitado de caminhos, os múltiplos caminhos podem ser considerados como um único caminho agregado para o tratamento de questões de largura de banda.Dessa forma, o modelo matemático proposto considera a utilização de um caminho para a comunicação entre os grupos de VMs da mesma aplicação. As fórmulas para o cálculo dos caminhos são mostradas nas Equações 3 e 4, que determinam o tráfego de saída do *rack* fonte e de entrada no *rack* destino, e nas Equações 5 e 6, utilizadas para garantir que um fluxo possui somente um caminho na rede. Outras restrições do modelo, que buscam garantir que a capacidade dos recursos virtuais não seja excedida, são omitidas devido a questões de espaço.

$$\sum_{\substack{w_2 \in S_i^v \mid \\ (r_1,w_2) \in E_i^v}} F_{a,i,(r_1,w_2),p} = \sum_{\substack{w_1 \in S_i^v \mid \\ (w_1,r_2) \in E_i^v}} F_{a,i,(w_1,r_2),p} = \min\left( \sum_{m \in M_i^v} g_{a,i,r_1,m}, \sum_{m \in M_i^v} g_{a,i,r_2,m} \right) * Band_a^r$$
$$\forall i \in I^v, \quad \forall a \in A^r, \quad \forall r_1, r_2 \in R_i^v \mid r_1 \neq r_2 \text{ and } p = (r_1, r_2) \quad (3)$$

$$\sum_{w_4 \in S_i^v \setminus w_3 \mid (w_3,w_4) \in E_i^v} F_{a,i,(w_3,w_4),p} - \sum_{w_4 \in S_i^v \setminus w_3 \mid (w_4,w_3) \in E_i^v} F_{a,i,(w_4,w_3),p} = 0$$
$$\forall i \in I^v, \quad \forall a \in A^r, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2, \quad \forall w_3 \in S_i^v \setminus \{r_1, r_2\} \quad (4)$$

$$F_{a,i,(w_1,w_2),p} \leq Band^v(w_1, w_2) * h_{a,i,(w_1,w_2),p}$$
$$\forall a \in A^r, \quad \forall i \in I^v, \quad \forall w_1, w_2 \in S_i^v \mid (w_1, w_2) \in E_i^v, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2 \quad (5)$$

$$\sum_{w_2 \in S_i^v \mid (w_1,w_2) \in E_i^v} h_{a,i,(w_1,w_2),p} \leq 1$$
$$\forall a \in A^r, \quad \forall i \in I^v, \quad \forall w_1 \in S_i^v, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2 \quad (6)$$

### 5.2. Mapeamento de Infraestruturas Virtuais no Substrato Físico

A função $\mathcal{G}$ é responsável pelo mapeamento das VIs no substrato físico da nuvem. Ela aborda um problema similar ao problema de mapeamento de redes virtuais (*virtual network embedding* - VNE), o que possibilita lidar com a heterogeneidade das topologias das VIs em relação ao substrato físico [Chowdhury et al. 2009]. No entanto, além da topologia da rede (composta por dispositivos de rede e enlaces), ela considera também a utilização de nodos computacionais (servidores e máquinas virtuais).

O problema de otimização desenvolvido para essa etapa utiliza como entradas o conjunto de infraestruturas virtuais (Seção 4.2) e a infraestrutura física da nuvem (Seção 4.3). Além disso, são utilizados os parâmetros $\alpha_{(w_1,w_2)}$ e $\gamma$, como segue. O fator $\alpha_{(w_1,w_2)}$ é definido de acordo com a importância do enlace ($w_1$, $w_2$) e possui

 ©2012 SBC — Soc. Bras. de Computação

valor máximo 1 ($0 < \alpha_{(w_1,w_2)} \leq 1$). Já o parâmetro $\gamma$ é utilizado para balancear os dois componentes da função objetivo.

As seguintes variáveis são utilizadas no modelo:

- $x_{i,s^v,s^s} \in \mathbb{B}$: indica se o *switch* virtual $s^v \in S_i^v$ da infraestrutura virtual $i \in I^v$ está alocado no *switch* físico $s^s \in S^s$;
- $y_{i,e^v,(w_1,w_2)} \in \mathbb{B}$: indica se o enlace virtual $e^v \in E_i^v$, pertencente à infraestrutura virtual $i \in I^v$, utiliza o enlace físico $(w_1, w_2) \in E^s$;
- $z_{i,m^v,m^s} \in \mathbb{B}$: indica se a VM $m^v \in M_i^v$ da infraestrutura virtual $i \in I^v$ está alocada no servidor $m^s \in M^s$.

A função objetivo (7) busca minimizar os recursos utilizados para alocar o conjunto de infraestruturas virtuais. O primeiro componente está relacionado aos enlaces da rede. Ao dividir-se $\alpha_{(w_1,w_2)}$ pela largura de banda do enlace, garante-se que os enlaces com menor prioridade e maior capacidade são preferidos em detrimento dos enlaces com maior importância. O segundo componente quantifica o custo de alocar *switches* virtuais em *switches* do substrato físico. Apesar da função não considerar requisitos de segurança, planeja-se incluir critérios de resiliência/segurança em trabalhos futuros, a fim de permitir maior isolamento entre as VIs no substrato físico e alocar enlaces virtuais em múltiplos caminhos, mitigando possíveis ataques de DoS.

$$Z = \text{Min} \sum_{(w_1,w_2)\in E^s} \frac{\alpha_{(w_1,w_2)}}{Band^s(w_1,w_2)} * \sum_{i\in I^v} \sum_{e^v \in E_i^v} y_{i,e^v,(w_1,w_2)} * Band^v(e^v) + \gamma * \sum_{s^s \in S^s} \sum_{i\in I^v} \sum_{s^v \in S_i^v} x_{i,s^v,s^s} * Cost^s(s^s) \tag{7}$$

Os valores das variáveis pertencentes à função objetivo são definidos com base nas restrições do modelo. Inicialmente, é necessário assegurar a capacidade dos servidores (Equação 8), dos *switches* físicos (Equação 9) e dos enlaces do substrato (Equação 10) em suportar os respectivos elementos virtuais.

$$\sum_{i\in I^v} \sum_{m^v \in M_i^v} z_{i,m^v,m^s} \leq Slots^s(m^s) \qquad \forall m^s \in M^s \tag{8}$$

$$\sum_{i\in I^v} \sum_{s^v \in S_i^v} x_{i,s^v,s^s} \leq Cap^s(s^s) \qquad \forall s^s \in S^s \tag{9}$$

$$\sum_{i\in I^v} \sum_{e^v \in E_i^v} (y_{i,e^v,(w_1,w_2)} * Band^v(e^v)) \leq Band^s(e^s) \qquad \forall e^s = (w_1,w_2) \in E^s \tag{10}$$

Também deve-se garantir que os enlaces virtuais são mapeados em caminhos válidos do substrato físico (Equação 11). As demais restrições do modelo, responsáveis pelo mapeamento dos recursos virtuais no substrato físico, são omitidas devido a questões de espaço.

 ©2012 SBC — Soc. Bras. de Computação

$$\sum_{w_4 \in S^s} y_{i,e^v,(w_3,w_4)} - \sum_{w_4 \in S^s} y_{i,e^v,(w_4,w_3)} = x_{i,w_1,w_3} - x_{i,w_2,w_3}$$
$$\forall i \in I^v, \quad \forall w_1, w_2 \in S_i^v \mid e^v = (w_1, w_2) \in E_i^v, \quad \forall w_3 \in S^s \quad (11)$$

## 6. Avaliação

Nesta seção, são avaliados dois aspectos da estratégia de alocação de recursos proposta. Primeiramente, a qualidade da solução é quantificada de duas formas: o benefício propiciado, com o aumento do nível de segurança às aplicações, e o custo disso, decorrente das restrições impostas na alocação pelas relações de confiança entre locatários. Em segundo lugar, busca-se verificar o tempo de processamento necessário à alocação ótima de recursos, o que afeta a viabilidade da solução.

### 6.1. Ambiente de Avaliação

Os experimentos foram implementados e executados no IBM ILOG CPLEX *Optimization Studio*[4]. Todos os experimentos foram executados em um computador Intel Core i3-2120 de 3.30 GHz, com 8 GB de memória RAM e sistema operacional GNU/Linux Debian x86_64. Cada experimento é limitado a um tempo máximo de duração de 1 hora e com uma árvore de busca de no máximo 4 GB, devido à complexidade do problema. Dessa forma, os resultados mostrados são os melhores resultados factíveis alcançados pelo CPLEX no período estipulado.

De modo similar a trabalhos relacionados [Ballani et al. 2011, Guo et al. 2010, Shieh et al. 2011], o substrato físico da nuvem foi definido como uma topologia em árvore. O substrato físico é composto por 80 servidores, cada um com 4 *slots*, divididos igualmente em 10 *racks*. Os *racks* são conectados entre si por *switches* das camadas superiores com diversidade de caminho.

A carga de trabalho a ser alocada são conjuntos de 12, 15, 20 ou 25 requisições para instanciar aplicações na nuvem, cada uma pertencendo a um locatário distinto. O número de VMs e a largura de banda especificados em cada requisição são uniformemente distribuídos, respectivamente, nos intervalos [2, 10] e [100,450] Mbps. A confiança mútua entre locatários foi gerada através das relações diretas entre os mesmos em um grafo aleatório, com grau de cada vértice (locatário) seguindo uma distribuição $P(k) \propto \frac{1}{k}$.

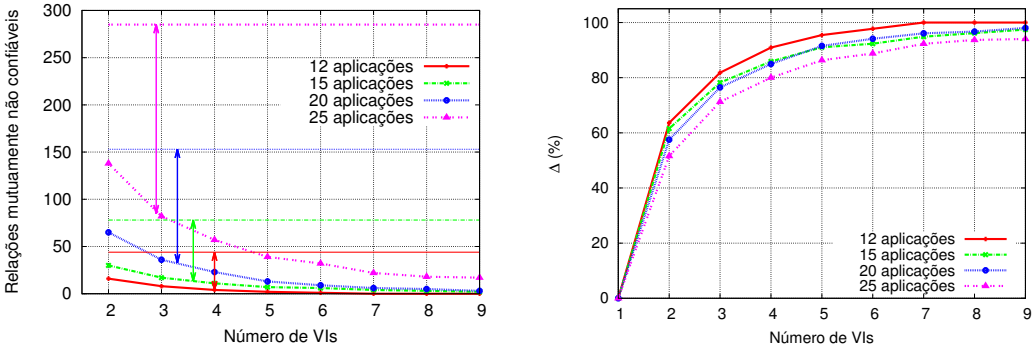
### 6.2. Qualidade da Solução

A qualidade da solução é quantificada de acordo com o critério de confiança mútua entre os locatários das aplicações da nuvem. Ou seja, busca-se avaliar a solução com base na segurança oferecida pelo agrupamento de aplicações de locatários em comparação com o cenário atual, no qual não é efetuado nenhum tipo de agrupamento. Refletindo isso, a métrica empregada é o número de relações entre aplicações de locatários que não possuem uma relação de confiança entre si e que foram alocadas no mesmo agrupamento. Deseja-se que esse valor seja o menor possível, pois representa uma certa exposição da aplicação a ataques.

[4] `http://www-01.ibm.com/software/integration/optimization/cplexoptimization-studio/`

 ©2012 SBC — Soc. Bras. de Computação

A Figura 3(a) mostra a variação do número de relações mutuamente não confiáveis de acordo com a quantidade de VIs oferecidas pelo provedor para os diferentes conjuntos de requisições de aplicações. Com base nesses dados, define-se diferença média do nível de confiança no cenário com agrupamento: $\Delta = 1 - \frac{\Phi}{\Gamma}$, onde $\Phi$ representa o número de relações não confiáveis entre aplicações dentro do agrupamento e $\Gamma$ denota o número de relações não confiáveis sem o agrupamento. O valor de $\Delta$ é apresentado na Figura 3(b), que mostra que o número de aplicações não é o fator principal para o aumento da segurança, mas sim o número de VIs oferecidas pelo provedor. De modo geral, é possível verificar que o ganho de segurança tem um comportamento logarítmico de acordo com o número de VIs.




(a) Relações não confiáveis.

(b) Diferença do nível de segurança com o agrupamento.

**Figura 3. Segurança com o agrupamento de aplicações na nuvem.**

O agrupamento pode ser benéfico também em relação ao consumo de largura de banda e ao isolamento entre o tráfego de aplicações com requisitos de banda distintos, como segue. As VMs pertencentes a uma aplicação são posicionadas, por definição, dentro de uma VI, enquanto os recursos de uma VI tendem a ser alocados próximos uns dos outros na infraestrutura física. Portanto, quanto menor a VI, maior é a chance de VMs comunicantes estarem próximas e, consequentemente, consumirem menos recursos de rede. O isolamento entre o tráfego de aplicações com requisitos de banda distintos, por sua vez, minimiza a interferência nociva que pode ser causada ao misturar tráfego com diferentes características [Abts and Felderman 2012].

Por outro lado, a criação de VIs e o agrupamento de locatários nas mesmas pode ter dois efeitos negativos. Primeiro, a criação e a manutenção das VIs introduz uma sobrecarga de gerência e comunicação em relação a uma rede convencional. Em relação a isso, segundo trabalhos anteriores, essa sobrecarga é proporcionalmente pequena e plenamente justificável considerando os benefícios [Ballani et al. 2011].

Segundo, o agrupamento tende a restringir a alocação de aplicações na nuvem, por causa da fragmentação dos recursos. Ou seja, quanto menor as VIs criadas, maior a fragmentação (interna). Com isso, maior será o número de requisições que não poderão ser atendidas apesar da capacidade agregada disponível. No presente trabalho, não quantificamos esse efeito porque modelamos o problema de otimização considerando recursos suficientes para atender todas as requisições, similarmente à Guo et al. 2010. Pretendemos analisar mais a fundo essa questão, em cenários com carência de recursos, em trabalhos futuros.

 ©2012 SBC — Soc. Bras. de Computação

### 6.3. Tempo de Processamento para Alocação

Outra métrica analisada na avaliação da estratégia de alocação de recursos é o tempo de processamento necessário à alocação de um conjunto de aplicações a um conjunto de VIs, considerando o tratamento *offline* neste artigo. A complexidade da função $\mathcal{F}$ é combinatória de acordo com o número de aplicações, o número de VIs e os seus respectivos elementos virtuais. A complexidade da função $\mathcal{G}$ é combinatória de acordo com as VIs oferecidas e o substrato físico. Por isso, o espaço de busca por soluções de ambos os problemas de otimização é grande, sendo necessária uma grande quantidade de tempo de processamento e memória para encontrar a solução ótima. Apesar disso, soluções factíveis, mas não ótimas, podem ser encontradas com menor dificuldade.

Nesse contexto, a complexidade dos dois problemas serve como grande motivação para o desenvolvimento, em trabalhos futuros, de meta-heurísticas (p.ex., *Simmulated Annealing*) e heurísticas construtivas considerando a chegada dinâmica (*online*) de requisições de aplicações.

## 7. Considerações Finais

As economias de escala estão impulsionando aplicações distribuídas a coexistir em infraestruturas compartilhadas. Entretanto, a falta de mecanismos para controlar o compartilhamento de recursos da rede interna da nuvem possibilita a ocorrência de ataques de egoísmo e DoS, prejudicando tanto provedores quanto locatários. Desse modo, a alocação de aplicações considerando recursos de rede e critérios de segurança representa um grande desafio de pesquisa.

Este artigo apresentou uma estratégia de alocação de recursos que visa mitigar ataques de egoísmo e negação de serviço na rede interna da nuvem. Este é o primeiro trabalho na literatura a explorar o agrupamento de aplicações em infraestruturas virtuais considerando a confiança mútua entre os locatários. Dessa forma, o sistema torna-se mais resiliente contra aplicações que usariam uma fatia desproporcional de recursos, com ou sem o intuito de prejudicar outras aplicações. A estratégia foi avaliada analiticamente perante diferentes níveis de agrupamento e comparada com um esquema padrão, sem agrupamento.

Como trabalhos futuros, pretende-se desenvolver meta-heurísticas e heurísticas construtivas, que considerem alta taxa de chegada e saída de locatários (*churn*) e elasticidade horizontal, para o modelo proposto. Também pretende-se acrescentar questões referentes à segurança na função $\mathcal{G}$ e explorar o isolamento/agrupamento propiciado pela estratégia proposta para mitigar outros tipos de ataques (p.ex., privacidade).

## Referências

Abts, D. and Felderman, B. (2012). A guided tour of data-center networking. *Commun. ACM*, 55(6):44–51.

Al-Fares, M., Loukissas, A., and Vahdat, A. (2008). A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, pages 63–74, New York, NY, USA. ACM.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53:50–58.

 ©2012 SBC — Soc. Bras. de Computação

Ballani, H., Costa, P., Karagiannis, T., and Rowstron, A. (2011). Towards predictable datacenter networks. In *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, SIGCOMM '11, pages 242–253, New York, NY, USA. ACM.

Breitgand, D. and Epstein, A. (2012). Improving Consolidation of Virtual Machines with Risk-aware Bandwidth Oversubscription in Compute Clouds. In *Proceedings of the 31th conference on Information communications*, INFOCOM'12, Piscataway, NJ, USA. IEEE Press.

Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783 –791.

Costa, R. B. and Carmo, L. F. R. C. (2007). Avaliação de Confiança Contextual em Grades Computacionais Multímodo usando Plataformas Seguras. In *VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, SBSeg 2007*, pages 173–185.

Goel, A. and Indyk, P. (1999). Stochastic load balancing and related problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 579 –586.

Greenberg, A., Hamilton, J., Maltz, D. A., and Patel, P. (2008). The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2009). Vl2: a scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 51–62, New York, NY, USA. ACM.

Grobauer, B., Walloschek, T., and Stocker, E. (2011). Understanding Cloud Computing Vulnerabilities. *Security Privacy, IEEE*, 9(2):50–57.

Guo, C., Lu, G., Wang, H. J., Yang, S., Kong, C., Sun, P., Wu, W., and Zhang, Y. (2010). Secondnet: a data center network virtualization architecture with bandwidth guarantees. In *Proceedings of the 6th International Conference*, Co-NEXT '10, pages 15:1–15:12, New York, NY, USA. ACM.

Jensen, M., Schwenk, J., Gruschka, N., and Iacono, L. L. (2009). On Technical Security Issues in Cloud Computing. In *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pages 109–116.

Kitsos, I., Papaioannou, A., Tsikoudis, N., and Magoutis, K. (2012). Adapting data-intensive workloads to generic allocation policies in cloud infrastructures. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 25 –33.

Lam, T. and Varghese, G. (2010). NetShare: Virtualizing Bandwidth within the Cloud. Technical report cs2010-0957, University of California.

Liu, H. (2010). A new form of DOS attack in a cloud and its avoidance mechanism. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, pages 65–76, New York, NY, USA. ACM.

Meng, X., Pappas, V., and Zhang, L. (2010). Improving the scalability of data center networks with traffic-aware virtual machine placement. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 1154–1162, Piscataway, NJ, USA. IEEE Press.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 199–212, New York, NY, USA. ACM.

Shieh, A., Kandula, S., Greenberg, A., Kim, C., and Saha, B. (2011). Sharing the data center network. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 23–23, Berkeley, CA, USA. USENIX Association.

Wang, M., Meng, X., and Zhang, L. (2011). Consolidating virtual machines with dynamic bandwidth demand in data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 71 –75.

Zimmermann, P. R. (1995). *The official PGP user's guide.* MIT Press, Cambridge, MA, USA.

 ©2012 SBC — Soc. Bras. de Computação

# Um Esquema Cooperativo para Análise da Presença de Ataques EUP em Redes Ad Hoc de Rádio Cognitivo

**Julio Soto, Saulo Queiroz*, Michele Nogueira**

[1]NR2 – Núcleo de Redes sem Fio e Redes Avançadas
Universidade Federal do Paraná (UFPR), Brasil

{jchsoto, sjbqueiroz, michele}@inf.ufpr.br

***Abstract.*** *In Cognitive Radio Ad Hoc Networks, malicious Secondary Users can exploit CR capabilities to perform Primary User Emulation Attacks (PUEA). These attacks pretend the characteristics of a Primary User, giving to malicious users the priority of using licensed frequencies over well-behaved unlicensed Secondary Users. To address this problem, studies in the literature have proposed the use of specific criteria to detect PUE attacks. Such solutions, however, do not consider the use of multiple different criteria to infer the presence of PUEA on the network. To fill this gap, we propose INCA, a novel mult**I**ple criteria scheme for the dece**N**tralized and **C**ooperative **A**nalysis of the PUEA presence in CRAHNs. INCA follows two phases. On the first, each SU employs multiple criteria to define a hypothesis about the potential existence of attacks; whereas on the second, these hypotheses are exchanged among neighbors, and each SU employs the Bayes theorem to calculate the final probability of the PUEA presence. Simulation results show the improvement and effectiveness of the multi-criteria approach.*

***Resumo.*** *Nas redes ad hoc de rádio cognitivo, os usuários mal intencionados podem tirar proveito das funcionalidades da tecnologia de rádio cognitivo para realizar ataques de emulação de usuário primário (EUP). Nestes ataques, os usuários mal intencionados imitam as características dos usuários licenciados, visando obter prioridade no uso das bandas de radiofrequências licenciadas e ameaçando o funcionamento da rede. A fim de tratar deste problema, trabalhos na literatura utilizam critérios específicos para detecção de ataques EUP. Tais soluções, contudo, não consideram conjuntamente o uso de múltiplos e diferentes critérios que enriquecem o consenso de inferência sobre a presença de um ataque EUP na rede. Diante deste contexto, este trabalho apresenta INCA, um esquema de múlt**I**plos critérios para a**N**álise **C**ooperativa da presença de **A**taques EUP em redes ad hoc de rádio cognitivo. O esquema INCA é composto por duas fases. Na primeira, cada usuário não licenciado emprega múltiplos critérios para definir uma hipótese individual da presença dos ataques EUP. Na segunda, essas hipóteses são trocadas entre os seus vizinhos e cada usuário não licenciado calcula a probabilidade final da presença de um ataque EUP através do teorema de Bayes. Os resultados de simulação mostram a melhoria e a eficácia da cooperatividade e do uso de múltiplos critérios quando aplicados simultaneamente.*

*autor também filiado à UTFPR (Universidade Tecnológica Federal do Paraná), Brasil.

 ©2012 SBC — Soc. Bras. de Computação

## 1. Introdução

A tecnologia de rádio cognitivo (RC) permite o surgimento das redes ad hoc de rádio cognitivo (do inglês *Cognitive Radio Ad Hoc Network - CRAHN*), nas quais os usuários secundários não licenciados (US) podem aproveitar oportunisticamente as bandas de radiofrequências licenciadas e subutilizadas pelos usuários primários (UPs) (estas bandas também são chamadas de *espaços em branco* ou *ociosos*). As CRAHNs podem ser aplicadas a diferentes situações, como aquelas de emergência, desastres naturais, conflitos bélicos, se beneficiando do uso de espaços em branco existentes, independente da localização do USs [Akyildiz et al. 2009]. Em uma CRAHN, os USs são capazes de comunicar-se entre eles através de múltiplos saltos. Para isto, inicialmente um US realiza um *sensoreamento* do espectro a fim de identificar os espaços em branco, seguido de uma *decisão* sobre qual espaço ocioso ele reutilizará para ter acesso ao espectro. Tal processo não pode causar qualquer interferência ao UP. Dessa forma, mediante a presença de um UP, o US deve eleger uma outra banda e realizar um *salto* de frequência para liberar a faixa atual. Durante todo este processo um US pode sofrer interrupções temporais até que um novo canal disponível seja encontrado [Mitola and Maguire 1999, Akyildiz et al. 2008, Lee and Akyldiz 2011].

As CRAHNs melhoram a eficiência no uso do espectro devido à utilização oportunista dos espaços em branco [Felice et al. 2011]. Por outro lado, tal característica vem sendo explorada como uma *nova* falha de segurança em redes sem fio. Um clássico exemplo nesse sentido é o ataque de emulação de usuário primário (EUP) e suas variantes [Tan et al. 2011, Chen and Park 2006]. Em geral, o ataque EUP é gerado por um US mal intencionado que tenta maximizar seu próprio uso do espectro, denominado US egoísta, ou gerar uma negação do serviço (*DoS* do inglês *Denial of Service*), denominado US malicioso. Um US mal intencionado (atacante) manipula suas configurações de rádio para imitar o comportamento de um UP. Assim, ele prejudica a oportunidade de acesso ao espectro de um US legítimo.

Criar soluções para detectar ou mitigar o efeito de USs mal intencionados constitui um desafio para a segurança dessas redes, devido à dificuldade em proteger a rede contra esses ataques e ao fato deles comprometerem severamente o desempenho dos US legítimos em perdas de oportunidades de acesso ao espectro [Chen and Park 2006, Chen et al. 2008, Jin et al. 2009, Jin et al. 2010]. Nesse sentido, os trabalhos na literatura para detecção ou mitigação desses ataques seguem enfoques (*descentralizadas* ou *centralizadas*) com abordagens *cooperativas* ou *não-cooperativas*. No entanto, esses trabalhos concentram uma grande parte do seu foco na definição do tipo de abordagens que irão utilizar, deixando outro aspecto muito importante de lado: a definição dos **critérios** para decidir sobre a presença de ataques EUP. O uso conjunto de diferentes critérios representam um foco importante devido ao fato de enriquecer as análises com múltiplos dados a fim de chegar a um consenso de inferência sobre a presença ameaçadora de atacantes na rede. Por outro lado, até onde sabemos, nenhum dos trabalhos na literatura aborda a concepção de um esquema de múltiplos critérios para detectar a presença de ataque EUP.

Assim, para tratar deste problema, nós propomos INCA, um esquema de múlt**I**plos critérios para a**N**álise **C**ooperativa da presença de **A**taques EUP em redes ad hoc de rádio cognitivo. O esquema INCA consiste de duas fases: **individual** e **cooperativa**. Na primeira fase, cada US realiza uma coleta de informação do meio, esta é realizada

 ©2012 SBC — Soc. Bras. de Computação

mediante a função de sensoreamento do espectro aplicando a técnica de detecção pela energia. Esta técnica foi escolhida por não precisar de um conhecimento prévio do sinal e pelo seu baixo custo computacional ao comparar com outras técnicas existentes (ex. [Ariananda et al. 2009, Fragkiadakis et al. 2012]). Após o sensoreamento do espectro, cada US utiliza múltiplos critérios para determinar uma probabilidade preliminar da presença de ataque EUP. Na segunda fase, os USs trocam hipóteses preliminares de detecção entre si. Em seguida, cada US aplica o teorema de Bayes sobre as probabilidades preliminares coletadas, a fim de calcular a probabilidade final da presença de ataques EUPs. A aplicação do teorema de Bayes permite a um nó atualizar sua probabilidade acerca da presença de um ataque na rede a partir do conhecimento das probabilidades de ataque advindas dos seus vizinhos. Este cálculo final determina a presença de um ou mais USs mal intencionados atacando a CRAHN. Resultados de simulações mostram que o esquema INCA melhora a análise probabilística da presença de ataque EUP em todos os cenários avaliados, sobretudo quando as duas fases do esquema são executadas. Estes resultados fortalecem o argumento de utilização de múltiplos critérios na detecção de ataques EUP.

Este trabalho está organizado da seguinte forma. Na Seção 2 descrevemos os trabalhos relacionados. Na Seção 3, nós apresentamos uma explicação sobre os ataques EUPs. Na Seção 4, nós detalhamos o esquema INCA como proposta para determinar a presença dos ataques EUP nas CRAHNs. Na Seção 5, avaliamos o esquema diante de diferentes cenários e descrevemos os resultados obtidos. Finalmente, na Seção 6, apresentamos as conclusões do artigo e os trabalhos futuros.

## 2. Trabalhos relacionados

Apesar de ser um tópico de pesquisa recente, trabalhos na literatura vêm empregando diferentes estratégias na tentativa de detectar os ataques EUP nas CRAHN. Neste contexto, observamos que os esquemas de detecção de ataques EUP vêm evoluindo no uso de abordagens centralizadas não-cooperativas a abordagens descentralizadas cooperativas. As primeiras soluções apresentavam modelos centralizados não-cooperativos para a detecção de ataques EUP. Uma primeira abordagem é vista em [Chen and Park 2006], utilizando duas técnicas baseadas na recepção do sinal e na localização dos transmissores. Em [Chen et al. 2008], a técnica LocDef é utilizada, tendo como base a localização do transmissor. Em [Li and Han 2010], o esquema Dogfight, é apresentado e tem como referência a competição entre o atacante e US legítimo utilizando a teoria de jogos. Por outro lado, uma das principais limitações nestes tipos de abordagens é a forte possibilidade de sobrecarregar a estação base central, além de poderem sofrer com altos níveis de latência.

Similarmente, esquemas baseados em abordagens centralizadas cooperativas foram propostos. Em [Min et al. 2011], utiliza-se a USs como sensores para descoberta dos atacantes na rede a partir da transmissão do UP. O esquema denominado IRIS [Chen et al. 2011] tem como base a potência de recepção combinada com pesos de importância de acordo com o estado do canal monitorado pelo US. Por outro lado, [Jin et al. 2010] apresenta um esquema baseado na utilização de uma aproximação log-normal para caracterizar a potência de recepção em cada nó e chegar a um consenso entre as informações coletadas pelos nós em uma estação base central a fim de determinar a probabilidade da presença do ataque EUP. No entanto, estes esquemas podem resolver

 ©2012 SBC — Soc. Bras. de Computação

os problemas das abordagens centralizadas não-cooperativas, mas apresentam outro problema, a alta taxa de falsos negativos e positivos.

Para reduzir as limitações das abordagens centralizadas cooperativas surgiram as abordagens distribuídas cooperativas, as quais seguem diferentes esquemas de detecção ou mitigação de ataques EUP. Em [Yuan et al. 2011], um esquema baseado na propagação de confianças foi proposto (do inglês *belief propagation - BP*). Este esquema calcula a força do sinal recebido por cada nó, e depois troca informação com os vizinhos a fim de determinar a presença do atacante. Em [Z. Jin and Subbalakshmi 2010], um esquema chamado NEAT foi apresentado. Ele utiliza uma aproximação log-normal para caracterizar a potência de recepção e determinar a presença dos atacantes com base na informação da vizinhança do nó monitor. No entanto, todas estas abordagens e esquemas existentes realizam uma análise da presença de ataque EUP na rede considerando apenas um ou dois critérios. Neste trabalho defendemos o uso de um esquema capaz de lidar com múltiplos critérios em abordagens distribuídas e cooperativas a fim de melhorar da taxa de detecção de ataques EUP nas CRAHNs.

## 3. Ataque de emulação de usuário primário

O ataque EUP foi apresentado por [Chen and Park 2006]. Este ataque é gerado por um US mal intencionado que manipula os parâmetros de configuração de seu rádio para fingir ser um UP [Jin et al. 2009, Chen et al. 2008]. Isto é possível pela capacidade e facilidade de configuração dos dispositivos equipados com RC. Um US mal intencionado pode alterar os parâmetros de seu dispositivo de rádio para imitar as características (ex. potência de transmissão, o tipo de modulação, a frequência utilizada, a largura de banda, a taxa de transmissão) de um UP. Como consequência, o atacante pode reduzir o número de espaços em branco disponíveis no espectro, gerando um *DoS*, e prejudicando seriamente o acesso ao espectro dos USs legítimos [Jin et al. 2009].

No ataque EUP, o atacante pode ser classificado em dois tipos: (i) um **usuário malicioso**, que imita o comportamento do UP para afetar o desempenho da rede, e (ii) um **usuário egoísta**, que também imita o comportamento de um UP a fim de obter as vantagens de prioridade de uso do espectro atribuído a ele [Chen and Park 2006]. Os dois tipos de ataque EUP afetam diretamente o processo de compartilhamento do espectro e reduzem drasticamente os recursos disponíveis aos demais USs legítimos [Chen et al. 2008]. Desta forma não fazemos distinção entre os dois tipos de ataques EUP neste trabalho.

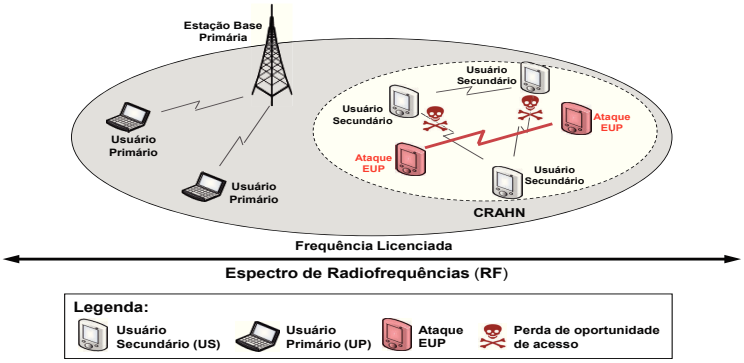



**Figura 1. CRAHN sob ataque EUP**

 ©2012 SBC — Soc. Bras. de Computação

A Figura 1 ilustra um US legítimo utilizando oportunisticamente o espectro ocioso. Em primeiro lugar, o US realiza o sensoreamento do espectro e após acessa as frequências ociosas. Sempre que um UP inicia atividade em sua banda licenciada, o US move-se para outra frequência ociosa do espectro. No entanto, os ataques EUPs podem comprometer diferentes faixas de frequências, levando os espaços em branco a serem ocupados. Desta forma, os US legítimos perdem a oportunidade de acessar o espectro, pois detectam a atividade dos atacantes como sendo atividades de UPs legítimos.

## 4. Esquema INCA

Nesta seção descrevemos o esquema INCA proposto, seguindo suas duas fases, **individual** ou e de **cooperação**, conforme ilustradas na Figura 2. Em síntese, na primeira fase, simplesmente denotada por $A$, cada US considera múltiplos critérios associados a pesos de importância específicos para determinar uma *probabilidade preliminar* $P_{n_i}(A)$ de acontecer um ataque detectável pelo nó $n_i$. Da perspectiva de $n_i$, $P_{n_i}(A)$ é apenas uma estimativa inicial sobre a presença do ataque EUP na rede. Em seguida, a fase de cooperação (denotada por $B$), é caracterizada pela recepção de informações de probabilidades oriundas de nós vizinhos. Dessa forma, cada vizinho $n_j$ compartilhará sua respectiva probabilidade preliminar (denotada por $P_{n_j}(B)$) com o nó $n_i$. Por fim, o esquema INCA emprega o teorema de Bayes para que o nó $n_i$ possa calcular sua probabilidade final $P_{n_i}(A|B)$ a partir de $P_{n_i}(A)$ e das informações de probabilidades recebidas de seus vizinhos.

Nas próximas seções, consideramos a notação sumarizada na Tabela 1 para explicar melhor essas fases. $N_{UP}$ representa o conjunto de usuários primários, e $N_S$ expressa o conjunto de usuários secundários (nós), onde $n_i$ representa um US arbitrário na rede ad hoc de rádio cognitivo. Por sua vez, $N_S = N_{SL} \cup N_{SB}$, em que $N_{SL}$ é o subconjunto de USs legítimos e $N_{SB}$ é o subconjunto de US mal intencionados. $c$ expressa o número de critérios considerados para a análise da presença do ataque EUP e $P_{n_i}(A)$ é a probabilidade preliminar do $i$-ésimo US na rede, isto é $1 \leq n_i \leq |N_{SL}|$.

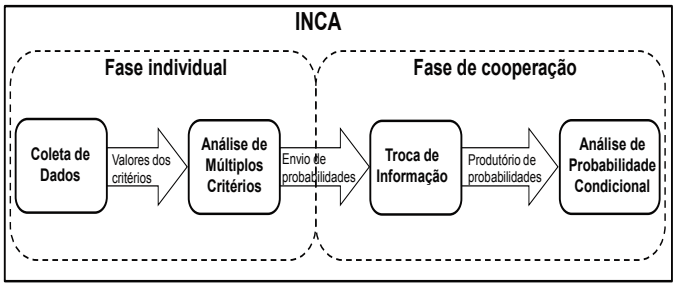



**Figura 2. Esquema INCA**

### 4.1. Fase individual

Na primeira fase do esquema INCA, o $i$-ésimo US realiza uma coleta de dados do meio. Esta coleta é realizada por uma técnica de sensoreamento do espectro definida na tecnologia de RC. Particularmente neste trabalho, utilizamos a técnica de detecção de sinal pela energia [Zhang et al. 2009, Subhedar and Birajdar 2011] devido a sua simplicidade de implementação e baixo custo computacional como demonstrado por

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Notação do esquema INCA**

| Notação | Definição |
|---|---|
| $N_{UP}$ | Conjunto de usuários primários |
| $N_{SL}$ | Conjunto de usuários secundários legítimos |
| $N_{SB}$ | Conjunto de usuários secundários mal intencionados |
| $N_S$ | Conjunto de usuários secundários |
| $n_i$ | $i$-ésimo usuário secundário arbitrário |
| $n_j$ | $j$-ésimo usuário secundário vizinho de $n_i$ |
| $P_{n_i}(A)$ | Probabilidade preliminar calculada pelo nó $n_i$ |
| $P_{n_j}(B)$ | Probabilidade preliminar calculada pelo vizinho $n_j$ |
| $P_{n_i}(A\|B)$ | Probabilidade final calculada pelo nó $n_i$ |
| $P_{n_j}(B\|A)$ | Probabilidade final calculada pelo vizinho $n_j$ |
| $\mathcal{S}$ | Conjunto de critérios |
| $\mathcal{W}$ | Conjunto de pesos de importância para os critérios |
| $\mathcal{MIN}$ e $\mathcal{MAX}$ | Conjunto de valores mínimos e máximos dos critérios |

[Ariananda et al. 2009, Fragkiadakis et al. 2012]. Após a coleta de dados, estes dados são agrupados em um conjunto de critérios estabelecido pelo esquema INCA.

Em seguida, o $i$-ésimo US calcula $P_{n_i}(A)$ sobre os conjuntos $\mathcal{S}$, $\mathcal{W}$, $\mathcal{MIN}$ e $\mathcal{MAX}$. Todo este processo é realizado pelo método de análise de múltiplos critérios NWAUF (do inglês *Normalized Weighted Additive Utility Function*) [Malakooti et al. 2006], pois estudos mostram o baixo custo computacional deste método em comparação a outros para análise de múltiplos critérios, como o AHP (*Analytic Hierarchy Process*) [Saaty 1980] ou ELECTRE [Benayoun et al. 1966], e as suas respectivas variantes [Malakooti et al. 2006, Wang and Triantaphyllou 2008]. Os passos gerais do método são apresentados no Algoritmo 1. $\mathcal{S}$ representa o conjunto de amostras coletadas para cada um dos critérios. Os conjuntos $\mathcal{MIN}$ e $\mathcal{MAX}$ são compostos pelos valores mínimos e máximos para cada um dos $|\mathcal{S}|$ critérios. Estes conjuntos são usados pelo algoritmo NWAUF para normalizar os valores em $\mathcal{S}$ e gerar o conjunto normalizado correspondente $\overline{\mathcal{S}} = \{\overline{s} \mid 0 \leq \overline{s} \leq 1\}$ (linhas 5–6). Finalmente, o algoritmo utiliza o conjunto de pesos $\mathcal{W} = \{w_1, w_2, \ldots, w_c \mid \sum_{z=1}^{c} w_z = 1\}$, em que cada peso de importância é atribuído a um critério em $\overline{\mathcal{S}}$ para calcular $P_{n_i}(A)$ (probablidade preliminar do $i$-ésimo US) (linha 7).

**Algoritmo 1** Análise NWAUF

1: **procedimento** NWAUFANALYSIS($\mathcal{S}, \mathcal{W}, \mathcal{MIN}, \mathcal{MAX}$)
2: $\quad P_{n_i}(A) \leftarrow 0$;
3: $\quad \overline{\mathcal{S}} \leftarrow \emptyset$;
4: $\quad$ **para todo** $z = 1 \rightarrow |\mathcal{S}|$ **faça**
5: $\quad\quad \overline{s_z} \leftarrow \dfrac{\mathcal{S}_z - \mathcal{MIN}_z}{\mathcal{MAX}_z - \mathcal{MIN}_z}$;
6: $\quad\quad \overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \cup \{\overline{s}\}$;
7: $\quad\quad P_{n_i}(A) \leftarrow P_{n_i}(A) + W_z \cdot \overline{s}_z$;
8: $\quad$ **fim para**
9: **fim procedimento**

Cada nó aplica o método NWAUF para calcular uma probabilidade preliminar da presença do ataque EUP através da agregação de dados de múltiplos critérios. Tal resultado é então utilizado para tomar a decisão final na fase de cooperação do INCA. A

 ©2012 SBC — Soc. Bras. de Computação

Figura 3 destaca cada passo da análise NWAUF de múltiplos critérios na primeira fase do INCA.

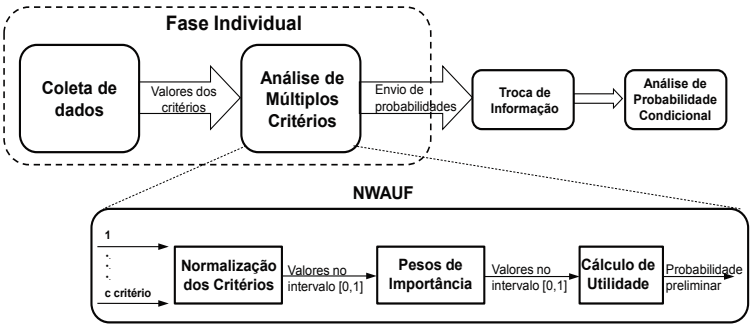



**Figura 3. Fase individual do INCA**

## 4.2. Fase de cooperação

Na fase de cooperação, o $i$-ésimo US não só realiza o compartilhamento de $P_{n_i}(A)$ (probabilidade preliminar da presença de ataque EUP), mas também recebe as probabilidades preliminares de outros nós vizinhos. O esquema INCA considera como vizinhos, os USs que estão no mesmo canal de frequência a um salto de distância de um nó. Esta troca de informação é realizada estabelecendo um canal de transmissão comum entre o US e cada um de seus vizinhos. A informação é colocada em um pacote de controle e enviada aos vizinhos, desta mesma forma os vizinhos ao receber o pacote, eles retornam um pacote de controle contendo suas probabilidades preliminares de detecção.

Depois de receber $k \leq |N_{SL}| - |\{n_i\}|$ probabilidades preliminares de sua vizinhança, o nó $n_i \in N_{SL}$ calcula sua probabilidade final $P_{n_i}(A|B)$ referente à presença de um ataque EUP na rede por meio do teorema de Bayes (Eq. 1). Este teorema permite calcular a probabilidade condicionada $P(A|B)$ de acontecer um dado evento $A$, mediante a ocorrência de um evento $B$. Para isso, o teorema considera uma probabilidade preliminar $P(A)$ de acontecer um evento $A$ sem a influência do evento $B$. Além disso ele também considera a probabilidade preliminar $P(B)$ de acontecer o evento $B$ bem como a probabilidade inversa de $P(A|B)$, isto é, $P(B|A)$.

Neste contexto utilizamos o teorema de Bayes para calcular a probabilidade final $P_{n_i}(A|B)$ do nó $n_i$ a partir dos valores das probabilidades $P_{n_i}(A)$, $P_{n_j}(B)$ e $P_{n_j}(B|A)$. Em particular, o evento $A$ denota o acontecimento de um ataque EUP da perspectiva do nó $n_i$. A probabilidade $P_{n_i}(A)$ é inicialmente estimada por $n_i$ a partir do método NWAUF, como explicado na fase individual do esquema INCA. Do mesmo modo, cada vizinho do nó $n_i$ calcula sua respectiva probabilidade preliminar acerca de um ataque EUP na rede. Da perspectiva do nó $n_i$, o evento $B$ denota o recebimento de estimativas de probabilidade de ataque de seus vizinhos $n_j$, $j = 1, \ldots, k$, em que $k$ é o total de vizinhos do nó $n_i$. Cada vizinho $n_j$ calcula sua própria probabilidade preliminar acerca de um ataque EUP na rede que são denotadas por $P_{n_j}(B)$ da perspectiva do nó $n_i$.

O teorema de Bayes requer que cada probabilidade do tipo $P_{n_j}(B|A)$ (i.e., probabilidade final de vizinhos) seja estimada para o cálculo da *primeira* amostra de probabilidade final $P_{n_i}(A|B)$ de um dado nó $n_i$. Em particular para nosso esquema, este valor

 ©2012 SBC — Soc. Bras. de Computação

foi inicialmente definido em $0.5$. Note (da eq. 1), contudo, que o fato de definirmos um mesmo valor constante para toda probabilidade do tipo $P(B|A)$ implica que a primeira amostra de uma probabilidade do tipo $P(A|B)$ seja inteiramente influenciada apenas por valores oriundos de medições, isto é, probabilidades dos tipos $P(A)$ e $P(B)$. Dessa forma, os valores das probabilidades do tipo $P(B|A)$ serão conhecidos para o cálculo das demais amostras de probabilidades do tipo $P(A|B)$ através da etapa de compartilhamento do esquema INCA.

A Figura 4 destaca os passos para que cada US determine a probabilidade da presença do ataque EUP através da probabilidade condicional definida pelo teorema de Bayes.

$$P_{n_i}(A|B) = \frac{P_{n_i}(A).P_{n_1}(B|A)}{[\sum_{j=1}^{k} P_{n_j}(B).P_{n_j}(B|A)]} \tag{1}$$

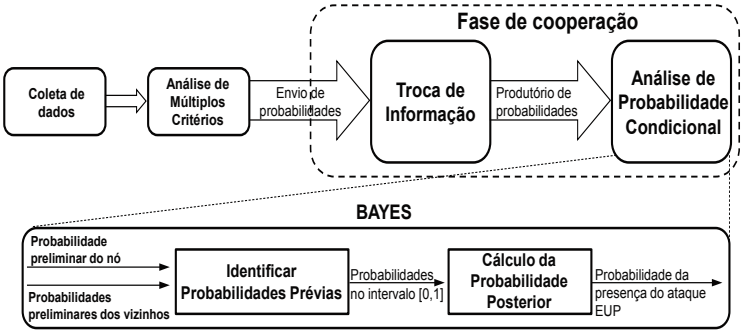



**Figura 4. Fase de cooperação do INCA**

### 4.3. INCA - Estudo de caso

O esquema INCA foi projetado para suportar múltiplos critérios a fim de analisar a presença de ataques EUP. Sem perda de generalidade, nesta subseção, nós selecionamos três critérios a fim de verificar o esquema INCA através de um estudo de caso. Os critérios escolhidos para análise do esquema INCA são a *intensidade do sinal recebido*, *potência de transmissão* e a *distância* com pesos de $50\%$, $25\%$ e $25\%$, respectivamente. Conforme reportamos na seção 5.2, tais pesos representam pelo menos uma configuração que demonstra a melhoria fornecida pela abordagem de múltiplos critérios proposta frente à abordagem representante do estado da arte. Esse resultado constitui o foco chave deste trabalho. Em face disso, o desenvolvimento de algoritmos que exploram a melhor configuração de pesos de forma *on-line* demanda um estudo particular que deixamos para trabalhos futuros.

A primeira fase do esquema INCA requer que os valores das métricas escolhidas possam ser amostrados. No contexto do nosso estudo de caso, a metodologia de amostragem para valores de intensidade do sinal demanda explanação adicional haja visto tal métrica ser intrinsecamente dependente do método de sensoriamento de espectro necessário aos nós com capacidade de rádio cognitivo, conforme discutimos a seguir.

A intensidade do sinal percebida por US legítimo $n_i \in N_{SL}$ a partir de um nó arbitrário $n_j \in N - \{n_i\}$ em um instante $t$ é denotado por $P_R^{n_j \to n_i}(t)$. A potência de transmissão de $n_j$ é denotada por $P_T^{n_j}(t)$ e a distância entre eles é $d_{n_j}^{n_i}(t)$.

 ©2012 SBC — Soc. Bras. de Computação

$$y_{n_i}(t) = \begin{cases} \eta(t), & H_0 \\ P_R^{n_j \to n_i}(t) + \eta(t), & H_1 \end{cases} \tag{2}$$

Levando em conta os modelos relacionados para o critério $P_R^{n_j \to n_i}(t)$, a primeira consideração a ser feita diz respeito a uma transmissão estar ocorrendo no espectro licenciado ou não, isto é, se o nó $n_i$ deve detectar o espectro para diferenciar o ruído da transmissão real. Conforme a Eq. 2 da técnica de detecção de atividade de espectro baseada em energia, a hipótese $H_1$ expressa que um nó arbitrário $n_j$ está usando o espectro licenciado. Neste caso, a potência total $y_{n_i}(t)$ percebida pelo $n_i$ no instante $t$ é $P_R^{n_j \to n_i}(t)$. Adicionalmente, assumimos que o canal é perturbado por um ruído aditivo Gaussiano branco $\eta(t)$. A hipótese $H_0$ ocorre quando não é realizada nenhuma transmissão no espectro e apenas o ruído pode ser detectado, ou seja, $P_R^{n_j \to n_i}(t) = 0$.

Por sua vez, INCA obtém $P_R^{n_j \to n_i}(t)$ a partir de $P_T^{n_j}(t)$ e $d_{n_j}^{n_i}(t)$ por meio dos modelos de propagação de rádio descritos na Eq. 3. Em particular, nós assumimos o modelo de propagação *Free-Space* para o sinal detectado por $n_i$ se $n_j \in N_{UP}$, e o modelo de propagação *two ray ground* para o sinal detectado por $n_i$ a partir de um usuário não primário i.e., US mal intencionado ou um US legítimo ([Z. Jin and Subbalakshmi 2010, Anand et al. 2008]). Finalmente, $G_p^2$ e $G_s^2$ representam os fatores de dissipação log-normais para os respectivos modelos supracitados.

$$P_R^{n_j \to n_i}(t) = \begin{cases} P_T^{n_j}(t)(d_{n_j}^{n_i}(t))^{-2} G_p^2, \text{ se } n_j \in N_P \\ P_T^{n_j}(t)(d_{n_j}^{n_i}(t))^{-4} G_s^2, \text{ se } n_j \in N_S \end{cases} \tag{3}$$

## 5. Análise e avaliação

Nesta seção, são descritos os cenários de simulação usados para avaliar o desempenho do esquema INCA. Inicialmente, são apresentados o ambiente de simulação, os parâmetros e suas justificativas. Em seguida, os resultados e análises.

### 5.1. Cenários de simulação

Nossas simulações estão compostas de $|N_{UP}| = 2$ e $|N_S| = 50$ nós estáticos, onde a quantidade de $|N_{SL}|$ e $|N_{SB}|$ variam para refletir diferentes taxas de atacantes na rede. Sempre que um US pretende transmitir, ele executa uma função de sensoreamento pela energia do sinal para detectar um canal ocioso. Além disso, cada UP acessa a sua banda licenciada seguindo uma distribuição aleatória. Por sua vez, assumimos que os atacantes acessam (i.e. atacam) o canal conforme uma distribuição uniforme.

Na rede, um usuário primário $UP \in N_{UP}$ emprega seu canal licenciado para realizar uma transmissão primária. A potência de transmissão $P_T^{UP}(t)$ de $UP$ pode ser facilmente determinada pelo padrão operacional correspondente a ele [Wild and Ramchandran 2005, Stevenson et al. 2009]. Por sua vez, assumimos que a localização do $UP$ é conhecida (ex. Antena de TV), e que qualquer nó estático $n_i \in N_{SL}$ pode conhecer a distância $d_{UP}^{n_i}(t)$ a partir de $UP$. Assim, com base nesses dados e

 ©2012 SBC — Soc. Bras. de Computação

considerando que as limitações físicas de um atacante (e.g. bateria, antena) forçam-no a transmitir numa potência várias ordens de magnitude menor que a do UP, o modelo de propagação de rádio (Eq. 3) e medições em $n_i$ podem aumentar a capacidade de diferenciação entre uma transmissão primária e uma transmissão mal intencionada e melhorar a probabilidade de detectar a presença do ataque EUP. Nossas simulações foram realizadas no NS-2.31 utilizando o módulo de redes ad hoc de rádio cognitivo desenvolvido por Di Felice et al. [Felice et al. 2011], seguindo os parâmetros descritos na Tabela 2. Os valores dos parâmetros utilizados na simulação foram escolhidos considerando situações realísticas de uma CRAHN, ex. a potência de transmissão de um UP ou de um US. Além disso, configuramos 50 nós estáticos (sem mobilidade) para ver seu comportamento em situações onde a taxa de atacantes muda de forma considerável.

**Tabela 2. Valores dos parâmetros de simulação**

| Parâmetro de simulação | Valor |
|---|---|
| Usuários secundários (USs) | 50 |
| Usuários primários (UPs) | 2 |
| Taxa de ataques EUPs | 10%, 30%, 50% |
| Número de canais | 11 |
| Tempo de simulação | 500 segundos |
| Área de cobertura de transmissão US | 250 m |
| Área de cobertura de transmissão UP | 1000 m |
| Área de cobertura de transmissão ataque EUP | 250 até 1000 m |
| Potência de transmissão US | 24.5 dBm |
| Potência de transmissão UP | 94 dBm (conforme [Stevenson et al. 2009]) |
| Potência de transmissão ataque EUP | 24.5 to 94 dBm |
| Protocolo de roteamento | AODV |
| Área | 1000x1000 $m^2$ |

A principal métrica de avaliação empregada é a **probabilidade da presença do ataque EUP** detectada por um US. Esta determina a probabilidade de sucesso do ataque a partir do momento que este é lançado na rede. Esta métrica é calculada a partir do teorema Bayesiano (Eq. 1) considerando a probabilidade preliminar do nó detector $P_{n_i}(A)$ e o conjunto de probabilidades preliminares calculadas e recebidas por cada um dos vizinhos cooperadores.

Nós também analisamos o impacto do número de vizinhos $k$ em $P_{n_i}(A|B)$. Para isto, nós forçamos a variação de valor $k$ em 3, 6 e 10, para realizar um cálculo da probabilidade da presença do ataque EUP limitando a quantidade de vizinhos cooperadores, para determinar a inferência que as probabilidades vizinhos dos nós causam na probabilidade inicial do nó. Esta variação da métrica principal foi chamada de **taxa de detecção** $T_{k,i}$ do nó $n_i \in N_{SL}$ diante de diferentes números de vizinhos cooperadores $k$. $T_{k,i}$ é dado por $T_{n_i} = \frac{\sum_{j=1}^{k} P_{n_j}(B)}{k}$, em que $P_{n_j}(B)$ é a probabilidade preliminar compartilhada pelo $j$-ésimo vizinho de $n_i$ e $0 \leq k \leq |N_{SL}| - |\{n_i\}|$ é o correspondente total dos nós vizinhos cooperadores.

## 5.2. Resultados

Nesta subseção, nós reportamos os resultados do desempenho do INCA usando múltiplos critérios e comparamos com uma versão usando apenas o critério de potência do sinal recebido, que é o critério amplamente utilizado no estado da arte. Esta versão com um

 ©2012 SBC — Soc. Bras. de Computação

único critério é referenciada como esquema **monocritério** nos gráficos. Nós inicialmente mostramos os resultados considerando INCA sem cooperação, ou seja, a fase individual utilizando *múltiplos critérios* frente à versão monocritério não cooperativa, para depois apresentar os resultados obtidos aplicando as duas fases do INCA. Cabe ressaltar que os resultados obtidos refletem a media de 35 simulações por cada cenário, com um intervalo de confiança de 95%.

A Figura 5 apresenta a comparação da probabilidade da presença do ataque EUP dada pela versão monocritério com as probabilidades dadas pela fase individual do INCA (isto é, a probabilidade preliminar obtida a partir de múltiplos critérios). Pode-se observar que o desempenho do esquema monocritério é levemente comprometido conforme a taxa de atacantes aumenta na rede. Estas probabilidades representam 35.96%, 35.30% e 34.48% para determinar a presença do ataque EUP com taxas de 10%, 30% e 50% de USs maliciosos nos diferentes cenários. Nestes mesmos cenários, o esquema INCA em sua fase individual com análise de múltiplos critérios supera as probabilidades de um único critério em até um 2.36%. Tais melhoras resultam do uso do NWAUF como técnica de agregação dos múltiplos critérios na primeira fase do INCA, sem considerar a fase cooperativa do esquema.

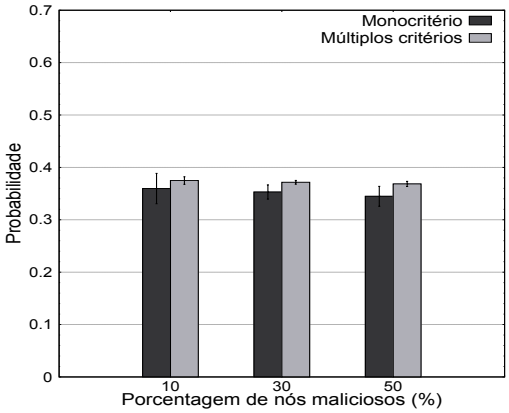



**Figura 5. Fase individual do INCA vs. esquema monocritério não-cooperativo**

Por sua vez, quando a probabilidade da presença do ataque EUP executa a fase de cooperação do INCA, os resultados mostram que ao comparar com o esquema monocritério cooperativo, os resultados são melhores (15.56%), como é observado na Figura 6. Desta vez, a cooperação funciona para enriquecer a análise de detecção e proporcionar uma melhoria de 12.6% em relação ao INCA em sua fase individual. Apesar disto, os benefícios obtidos pelo uso de cooperação podem ser prejudicados se uma abordagem de múltiplos critérios não acontece. Isto é fundamentado pelo fato de que um esquema monocritério cooperativo consegue 0.02% de melhoria em relação ao esquema monocritério não-cooperativo.

Finalmente, a Figura 7 compara a taxa de detecção $T_{k,i}$ sob a variação no número de vizinhos cooperadores $k$ considerando ambos esquemas (INCA e monocritério). Na avaliação cooperativa, nós levamos em conta a quantidade de nós cooperativos e a percentagem de atacantes na CRAHN. O esquema de um único critério apresenta uma variação de $1.33\%$ da probabilidade em cenários com 3, 6 e 10 nós cooperativos. Por sua vez, INCA apresenta uma variação de $0.3\%$ entre os cenários com diferentes quantidades de

 ©2012 SBC — Soc. Bras. de Computação

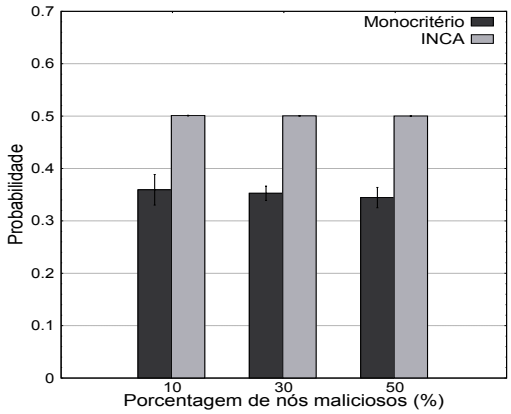



**Figura 6. Esquema INCA vs esquema monocritério cooperativo**

nós cooperativos. Além disso, nós verificamos que a baixa variação na probabilidade é devido ao acesso aleatório do ataque EUP, isto é um US pode não necessariamente realizar um sensoreamento do canal atacado.

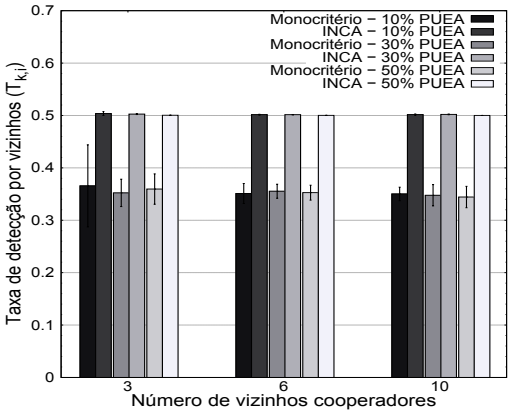



**Figura 7. Taxa de detecção por números de vizinhos**

## 6. Conclusões e trabalhos futuros

Neste trabalho, apresentamos INCA, um novo esquema de múltiplos critérios para a análise cooperativa da presença de ataques EUPs nas CRAHNs. O esquema INCA não só engloba abordagens do estado da arte, como a descentralização e a cooperação, mas também uma abordagem inovadora para levar em conta múltiplos critérios no processo de detecção de ataque EUPs. Nós comparamos os resultados gerados a partir do INCA com uma versão de um esquema que considera um único critério como forma de representar a abordagem considerada no estado da arte. Os resultados obtidos mostram que o esquema INCA tendo como base critérios adicionais, tais como a distância e a potência de transmissão, além da potência de recepção, apresenta um melhor desempenho em relação ao esquema monocritério. Cabe destacar que o esquema $INCA$ não foca seu estudo nos diferentes tipos de critérios utilizados, mas este afere que a utilização de múltiplos critérios em conjunto com uma abordagem cooperativa permite determinar uma melhor detecção da presença do ataque EUP nas redes ad hoc de rádio cognitivo.

 ©2012 SBC — Soc. Bras. de Computação

Assim, nossos resultados indicam as vantagens de criar esquemas de detecção de ataques EUPs que incorporam múltiplos critérios para analisar os ataques de emulação de usuário primário. Uma questão importante inerente à análise de ataque EUP baseada em múltiplos critérios, como o esquema INCA, é a tarefa de atribuição de pesos para representar a importância e diferenciação dos critérios. Para os critérios considerados no estudo de caso deste trabalho, identificamos (por meio de simulações piloto *off-line*) pelo menos uma configuração de pesos que demonstrou a melhoria alcançada pela abordagem de múltiplos critérios proposta frente à abordagem representante do estado da arte. Tal resultado não somente constitui um argumento científico a favor da abordagem proposta como também cria oportunidades para trabalhos futuros. Neste sentido, um trabalho futuro poderia investigar algoritmos de otimização para a definir os pesos de forma *on-line* e variável ao longo do tempo. Adicionalmente, o impacto da mobilidade dos nós poderia ser considerada no processo de cálculo das estimativas de ataque.

## Referências

Akyildiz, I., Lee, W.-Y., Vuran, M., and Mohanty, S. (2008). A survey on spectrum management in cognitive radio networks. *IEEE Comm. Mag.*, 46(4):40 –48.

Akyildiz, I. F., Lee, W.-Y., and Chowdhury, K. R. (2009). Crahns: Cognitive radio ad hoc networks. *Ad Hoc Networks*, 7(5):810–836.

Anand, S., Jin, Z., and Subbalakshmi, K. (2008). An analytical model for primary user emulation attacks in cognitive radio networks. In *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pages 1–6.

Ariananda, D., Lakshmanan, M., and Nikoo, H. (2009). A survey on spectrum sensing techniques for cognitive radio. In *Cognitive Radio and Advanced Spectrum Management, CogART. Second International Workshop on*, pages 74 –79.

Benayoun, R., Roy, B., and Sussman, N. (1966). Manual de reference du programme electre. *Note De Synthese et Formaton, 25*.

Chen, C., Cheng, H., and Yao, Y.-D. (2011). Cooperative spectrum sensing in cognitive radio networks in the presence of the primary user emulation attack. *IEEE Trans. on Wireless Comm.*, 10(7):2135 –2141.

Chen, R. and Park, J.-M. (2006). Ensuring trustworthy spectrum sensing in cognitive radio networks. In *IEEE SDR*, pages 110 –119.

Chen, R., Park, J.-M., and Reed, J. (2008). Defense against primary user emulation attacks in cognitive radio networks. *IEEE Journal on Selected Areas in Comm.*, 26(1):25–37.

Felice, M. D., Chowdhury, K., Kim, W., Kassler, A., and Bononi, L. (2011). End-to-end protocols for cognitive radio ad hoc networks: An evaluation study. *Performance Evaluation*, 68(9):859 – 875.

Fragkiadakis, A., Tragos, E., and Askoxylakis, I. (2012). A survey on security threats and detection techniques in cognitive radio networks. *IEEE Communications Surveys Tutorials*, PP(99):1 – 18.

Jin, Z., Anand, S., and Subbalakshmi, K. (2010). Robust spectrum decision protocol against primary user emulation attacks in dynamic spectrum access networks. In *IEEE GLOBECOM*, pages 1–5.

 ©2012 SBC — Soc. Bras. de Computação

Jin, Z., Anand, S., and Subbalakshmi, K. P. (2009). Detecting primary user emulation attacks in dynamic spectrum access networks. In *IEEE ICC*, pages 2749–2753.

Lee, W.-Y. and Akyldiz, I. F. (2011). A spectrum decision framework for cognitive radio networks. *IEEE Trans. on Mobile Computing*, 10(2):161 –174.

Li, H. and Han, Z. (2010). Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics. *IEEE Trans. on Wireless Comm.*, 9(11):3566 –3577.

Malakooti, B., Thomas, I., Tanguturi, S., Gajurel, S., and Kim, H. (2006). Multiple criteria network routing with simulation results. *Industrial Engineering Research Conference (IERC)*.

Min, A., Kim, K.-H., and Shin, K. (2011). Robust cooperative sensing via state estimation in cognitive radio networks. In *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pages 185 –196.

Mitola, J. and Maguire, G. Q. (1999). Cognitive radio: making software radios more personal. *IEEE Pers. Comm.*, 6(4):13–18.

Saaty, T. L. (1980). The analytic hierarchy process. *McGraw Hill*.

Stevenson, C., Chouinard, G., Lei, Z., Hu, W., Shellhammer, S., and Caldwell, W. (2009). Ieee 802.22: The first cognitive radio wireless regional area network standard. *IEEE Communications Magazine*, 47(1):130 –138.

Subhedar, M. and Birajdar, G. (2011). Spectrum sensing techniques in cognitive radio networks: A survey. *International Journal of NextGeneration Networks*, 3(2):37–51.

Tan, Y., Hong, K., Sengupta, S., and Subbalakshmi, K. P. (2011). Using sybil identities for primary user emulation and byzantine attacks in DSA networks. In *IEEE GLOBECOM*, pages 1–5.

Wang, X. and Triantaphyllou, E. (2008). Ranking irregularities when evaluating alternatives by using some electre methods. *Omega*, 36(1):45 – 63.

Wild, B. and Ramchandran, K. (2005). Detecting primary receivers for cognitive radio applications. In *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pages 124 –130.

Yuan, Z., Niyato, D., Li, H., and Han, Z. (2011). Defense against primary user emulation attacks using belief propagation of location information in cognitive radio networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 599 – 604.

Z. Jin, S. A. and Subbalakshmi, K. P. (2010). Neat: A neighbor assisted spectrum decision protocol for resilience against primary user emulation attacks. Technical report, Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ.

Zhang, W., Mallik, R., and Letaief, K. (2009). Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 8(12):5761 –5766.

 ©2012 SBC — Soc. Bras. de Computação

# Usando Criptografia de Limiar para Tolerar Clientes Maliciosos em Memória Compartilhada Dinâmica*

**Eduardo Adilio Pelinson Alchieri**[1] **Alysson Neves Bessani**[2] **Joni da Silva Fraga**[3]

[1]CIC, Universidade de Brasília, Brasília - Brasil
[2]LaSIGE, Universidade de Lisboa, Lisboa - Portugal
[3]DAS, Universidade Federal de Santa Catarina, Florianópolis - Brasil

***Resumo.*** *Sistemas de quóruns Bizantinos são ferramentas usadas na implementação consistente e confiável de sistemas de armazenamento de dados em presença de falhas arbitrárias. Vários protocolos para implementação destes sistemas foram propostos para ambientes estáticos e, mais recentemente, também surgiram propostas de protocolos para ambientes dinâmicos. Um dos desafios na implementação desses sistemas em ambientes dinâmicos está na reconfiguração do conjunto de servidores devido a entradas e saídas arbitrárias de processos. Este trabalho vai além e apresenta protocolos que toleram a presença de clientes maliciosos em sistemas de quóruns Bizantinos dinâmicos, através do emprego de um mecanismo de criptografia de limiar que fornece a flexibilidade suficiente para operação nestes ambientes. Este mecanismo é utilizado para controlar as ações que clientes maliciosos podem executar contra o sistema. Além disso, todos os protocolos apresentados são para sistemas assíncronos, não necessitando de nenhuma premissa temporal sobre o comportamento do sistema.*

***Abstract.*** *Byzantine quorum systems are an useful tool to implement consistent and available data storage systems in the presence of arbitrary faults. Several protocols were proposed to implement these systems in static environments and, more recent, proposals have also appeared for dynamic environments. One of the challenges to implement these systems in dynamic environments is the reconfiguration of the servers set due to joins and leaves. In this work we go one step further and propose protocols that tolerate malicious clients in dynamic Byzantine quorum systems, by using a threshold cryptography scheme that provides flexibility necessary to operate in these environments. This scheme is used to control actions that malicious clients can execute against the system. Moreover, all proposed protocols are for asynchronous systems.*

## 1. Introdução

Sistemas de quóruns [Gifford 1979] são ferramentas usadas para garantir consistência e disponibilidade de dados, que são armazenados de forma replicada em um conjunto de servidores. Além de serem blocos básicos de construção para protocolos de sincronização (ex.: consenso), o grande atrativo destes sistemas está relacionado com seu poder de escalabilidade e balanceamento de carga, uma vez que as operações não precisam ser executadas por todos os servidores do sistema, mas apenas por um quórum dos mesmos. A consistência de um sistema de quóruns é assegurada pela propriedade de intersecção dos subconjuntos de servidores (quóruns).

Sistemas de quóruns foram inicialmente estudados em ambientes estáticos, onde não é permitida a entrada e saída de servidores durante a execução do sistema [Bazzi and Ding 2004, Malkhi and Reiter 1998]. Esta abordagem não é adequada para sistemas que permanecerão em execução por um longo tempo, uma vez que, dispondo de um quantidade suficiente de tempo,

*Este trabalho recebeu apoio do DPP/UnB através do Edital 10/2012.

 ©2012 SBC — Soc. Bras. de Computação

um adversário pode comprometer um número maior de servidores do que o tolerado e então quebrar as propriedades do sistema. Outra limitação é que estes protocolos não permitem que um administrador, em tempo de execução, adicione máquinas no sistema (para suportar um aumento na carga de processamento) ou troque máquinas antigas. Estes protocolos também não são adequados para sistemas distribuídos modernos, desenvolvidos para redes móveis e auto-organizáveis (e.g., MANETs, P2P), onde, pela sua própria natureza, o conjunto de processos que compõem o sistema sofre modificações durante a execução.

A dinamicidade é de fato um grande desafio no projeto dos sistemas de quóruns e, mais especificamente, no projeto de protocolos para reconfiguração do conjunto de servidores que mantêm a memória compartilhada. Devido a entradas, saídas e falhas de servidores e de clientes, é preciso manter a consistência dos dados e garantir a sua disponibilidade. Esse processo de reconfiguração torna-se ainda mais complexo quando considera-se a possibilidade de componentes maliciosos estarem presentes na computação [Lamport et al. 1982].

Alguns trabalhos foram propostos com o intuito de prover uma memória compartilhada em ambientes dinâmicos e assim implementar um sistema de quóruns dinâmicos capaz de tolerar tanto o *crash* de servidores [Lynch and Shvartsman 2002, Aguilera et al. 2011] quanto o comportamento malicioso dos mesmos [Martin and Alvisi 2004, Rodrigues and Liskov 2004, Alchieri et al. 2012]. Porém, uma característica comum a todas estas propostas é que nenhuma destas fornece um protocolo capaz de tolerar a presença de clientes maliciosos no sistema. De fato, um dos grandes desafios em sistemas de quóruns é desenvolver protocolos eficientes para tolerar comportamento malicioso de clientes, pois os mesmos podem executar uma série de ações maliciosas com o objetivo de ferir as propriedades do sistema, como por exemplo realizar uma escrita incompleta, atualizando apenas alguns servidores. Esta característica é importante na medida em que os sistemas de quóruns foram desenvolvidos para operar em ambientes de larga escala, como a Internet, onde sua natureza de sistema aberto aumenta significativamente a possibilidade de clientes maliciosos estarem presentes no sistema. Se considerarmos um ambiente dinâmico e todas suas incertezas, esta possibilidade é ainda maior.

Neste artigo, usamos como base os protocolos de reconfiguração do QUINCUNX [Alchieri et al. 2012] e propomos protocolos para um sistema de quóruns dinâmico que tolera tanto servidores quanto clientes maliciosos. Para isso, um esquema de criptografia de limiar é utilizado, de forma que os servidores corretos conseguem tanto controlar as ações de clientes maliciosos quanto mascarar o comportamento malicioso de outros servidores. Os protocolos do QUINCUNX foram escolhidos por permitirem a reconfiguração do sistema sem a necessidade de premissas temporais, o que os tornam adequados para sistemas assíncronos[1]. Os procedimentos de reconfiguração envolvem a evolução do sistema de uma configuração (visão) antiga para uma nova configuração (visão) atualizada, onde servidores entram e/ou saem do sistema. Além disso, os protocolos de reconfiguração do QUINCUNX são independentes dos protocolos de leitura e escrita (R/W) usados para acessar o sistema, o que facilita a adequação de outros protocolos de R/W para ambientes dinâmicos, além de aumentar o desempenho destas operações quando estão concorrendo com reconfigurações.

A solução proposta neste trabalho, chamada QUINCUNX-BC (QUINCUNX *for byzantine clients*) preserva estas características, i.e., os protocolos do QUINCUNX-BC não necessitam de nenhuma premissa temporal e mantém o desacoplamento entre os protocolos de reconfiguração e os protocolos de R/W. Os protocolos de R/W do QUINCUNX-BC são uma adaptação para ambientes dinâmicos dos protocolos do PBFT-BC [Alchieri et al. 2009], que utilizam cripto-

[1]Como sabemos, não é necessário premissas de sincronização (tempo) para a manutenção da consistência de uma memória compartilhada em um conjunto estático de servidores [Malkhi and Reiter 1998].

 ©2012 SBC — Soc. Bras. de Computação

grafia de limiar para tolerar processos maliciosos. Neste sentido, um exigência fundamental no QUINCUNX-BC é o emprego de um protocolo de redistribuição de chaves parciais (usadas pelo mecanismo de criptografia de limiar), onde as chaves parciais relacionadas a uma visão mais antiga são redistribuídas entre os servidores de uma visão mais atual. Neste sentido, este trabalho apresenta pelo menos duas contribuições principais:

- Novos protocolos de R/W para sistema de quóruns Bizantinos dinâmicos, que utilizam criptografia de limiar para tolerar tanto clientes quanto servidores maliciosos;
- Um protocolo para redistribuição de chaves parciais entre as visões instaladas no sistema, que não requer primitivas de sincronização.

No QUINCUNX-BC, estes protocolos são utilizados em conjunto com os protocolos de reconfiguração do QUINCUNX resultando em um sistema de quóruns Bizantino dinâmico que tolera tanto clientes quanto servidores maliciosos. Esta característica torna o QUINCUNX-BC interessante para sistemas dinâmicos que, devido às suas características, são mais susceptíveis à presença de componentes maliciosos.

O resto deste artigo é organizado da seguinte forma. A Seção 2 apresenta algumas definições preliminares. Os protocolos para tolerar clientes maliciosos em sistemas de quóruns Bizantinos Dinâmicos são apresentados na Seção 3. A Seção 4 apresenta algumas discussões sobre os protocolos propostos e as conclusões do trabalho são apresentadas na Seção 5.

## 2. Definições Preliminares

Esta seção apresenta os principais conceitos e forma a base teórica para o QUINCUNX-BC.

### 2.1. Modelo de Sistema

Consideramos um sistema distribuído assíncrono completamente conectado composto pelo conjunto universo de processos $U$, que é dividido em dois subconjuntos: um conjunto infinito de servidores $\Pi = \{s_1, s_2, ...\}$; e um conjunto infinito de clientes $C = \{c_1, c_2, ...\}$. Cada processo do sistema (cliente ou servidor) possui um identificador único e está sujeito a faltas Bizantinas [Lamport et al. 1982]. Um processo que apresenta comportamento de falha é dito falho (ou faltoso), de outra forma é dito correto. A chegada dos processos no sistema segue o modelo de chegadas infinita com concorrência desconhecida mas finita [Aguilera 2004]. As comunicações entre processos são realizadas através de canais ponto-a-ponto confiáveis e autenticados.

Nossos protocolos utilizam criptografia de limiar para controlar as ações dos clientes e garantir a integridade dos dados armazenados. Deste modo, consideramos que na inicialização do sistema cada servidor presente na visão inicial recebe a sua chave parcial, usada na elaboração de assinaturas parciais. Servidores corretos nunca revelam estas chaves, que são geradas e distribuídas por um administrador correto que somente é necessário na inicialização do sistema. Após isso, para instalar visões atualizadas no sistema, um protocolo para redistribuição de chaves parciais é utilizado. A chave pública do serviço, usada para verificar as assinaturas geradas por este mecanismo, é armazenada pelos servidores e fica disponível para qualquer processo do sistema. Vale ressaltar que esta chave nunca é modificada, i.e., mesmo que ocorram redistribuições das chaves parciais, a chave pública do serviço não é alterada.

Cada servidor possui um par distinto de chaves (chave pública e privada) para usar um sistema de criptografia assimétrica. Cada chave privada é conhecida apenas pelo seu próprio dono, por outro lado todos os processos conhecem todas as chaves públicas. Denotamos uma mensagem $m$ assinada pelo processo $i$ como $m_i$ e consideramos que apenas mensagens corretamente assinadas são processadas pelos processos corretos.

 ©2012 SBC — Soc. Bras. de Computação

Consideramos a existência de uma função criptográfica de resumo (*hash*) resistente a colisões $h$, de tal modo que qualquer processo é capaz de calcular o resumo $h(v)$ do valor $v$, sendo computacionalmente inviável obter dois valores distintos $v$ e $v'$ tal que $h(v) = h(v')$. Por fim, para evitar ataques de *replay*, *nonces* são adicionados em certas mensagens. Consideramos que os clientes corretos não escolhem *nonces* repetidos, i.e., já utilizados.

### 2.2. Criptografia de Limiar

A principal ferramenta utilizada no QUINCUNX-BC é um esquema de assinatura de limiar (*threshold signature scheme* - TSS) [Desmedt and Frankel 1990, Shoup 2000] através do qual é possível controlar as ações dos clientes e garantir a integridade dos dados armazenados pelos servidores. Em um esquema $(n, k)$-TSS, um distribuidor confiável inicialmente gera $n$ chaves parciais $(SK_1, ..., SK_n)$, $n$ chaves de verificação $(VK_1, ..., VK_n)$, a chave de verificação de grupo $VK$ e a chave pública $PK$ usada para validar assinaturas. O distribuidor envia estas chaves para $n$ partes diferentes, chamados portadores, de modo que cada portador $i$ recebe sua chave parcial $SK_i$ e sua chave de verificação $VK_i$. A chave pública e as chaves de verificação são disponibilizadas para qualquer parte que compõe o sistema.

Após esta configuração inicial, o sistema está apto à gerar assinaturas. Na obtenção de uma assinatura do serviço $A$ para o dado $data$, primeiramente cada portador $i$ gera a sua assinatura parcial $a_i$ para $data$. Posteriormente, um combinador obtém pelo menos $k$ assinaturas parciais válidas $(a_1, ..., a_k)$ e constrói a assinatura $A$ através da combinação destas $k$ assinaturas parciais. Uma propriedade fundamental deste esquema é a impossibilidade de gerar assinaturas válidas com menos de $k$ assinaturas parciais. Este esquema é baseado nas seguintes primitivas:

- $Th_Sign(SK_i, VK_i, VK, data)$: função usada pelo portador $i$ para gerar sua assinatura parcial $a_i$ sobre $data$ e as provas $v_i$ de validade desta assinatura, i.e., $\langle a_i, v_i \rangle$.
- $Th_VerifyS(data, \langle a_i, v_i \rangle, PK, VK, VK_i)$: função usada para verificar se a assinatura parcial $a_i$, apresentada pelo portador $i$, é válida.
- $Th_CombineS(a_1, ..., a_k, PK)$: função usada para combinar $k$ assinaturas parciais válidas e obter a assinatura $A$.
- $verify(data, A, PK)$: função usada para verificar a validade da assinatura $A$ sobre o dado $data$ (verificação normal de assinatura).

O QUINCUNX-BC utiliza o protocolo proposto em [Shoup 2000], onde é provado que tal esquema é seguro no modelo dos oráculos aleatórios [Bellare and Rogaway 1993], não sendo possível forjar assinaturas. Este protocolo representa um esquema de assinaturas de limiar baseado no algoritmo RSA [Rivest et al. 1978], i.e., a combinação das assinaturas parciais gera uma assinatura RSA. Neste modelo a geração e verificação de assinaturas parciais é completamente não interativa, não sendo necessárias trocas de mensagens para executar estas operações.

Para redistribuição de chaves parciais, seguimos as definições apresentadas em [Wong et al. 2002] onde a transição de um esquema $(n, k)$-TSS para um esquema $(n', k')$-TSS funciona da seguinte forma: primeiro, cada um dos $n$ portadores de $(n, k)$-TSS gera $n'$ *shares* de sua chave parcial (um *share* para cada portador do novo esquema) e envia estes *shares* para os $n'$ portadores do novo esquema $(n', k')$-TSS, os quais obtém pelo menos $k$ *shares* válidos e combinam suas novas chaves parciais. Um requisito fundamental para este protocolo funcionar é que, para as novas chaves parciais serem compatíveis, é necessário que os $k$ *shares* combinados por cada portador do esquema $(n', k')$-TSS sejam gerados pelo mesmo conjunto de $k$ portadores de $(n, k)$-TSS [Wong et al. 2002]. Note que as novas chaves parciais derivam das chaves parciais anteriores e são compatíveis com a chave pública $PK$ do serviço. Para redistribuição de chaves parciais entre estes dois esquemas, as seguintes funções são necessárias:

 ©2012 SBC — Soc. Bras. de Computação

- $Th_share(SK_i, j, n', k')$: função usada pelo portador $i$ para gerar o *share* $\widehat{k}_{ij}$ de sua chave parcial $SK_i$. Este *share* é usado por $j$ na obtenção de sua nova chave parcial.
- $Th_generateV(i, n', k')$: função usada pelo portador $i$ para gerar o conjunto de verificadores $e_i$, os quais atestam que os *shares* de sua chave parcial são válidos.
- $Th_verifySK(\widehat{k}_{ji}, e_j)$: função usada pelo portador $i$ para verificar a validade do *share* $\widehat{k}_{ji}$ gerado por $j$. Nesta função, $e_j$ representa os verificadores gerados por $j$.
- $Th_combineSK(\widehat{k}_{1i}, ..., \widehat{k}_{ki}, red_set)$: função usada pelo portador $i$ para obter sua nova chave parcial $SK_i$ (e verificador $VK_i$) a partir da combinação de $k$ *shares* válidos obtidos das chaves parciais da configuração antiga. Os identificadores dos portadores, a partir dos quais os *shares* devem ser usados nesta combinação, estão em $red_set$.

### 2.3. Sistemas de Quóruns Bizantinos

Sistemas de quóruns Bizantinos [Malkhi and Reiter 1998], doravante denominados apenas como sistemas de quóruns, implementam sistemas replicados de armazenamento de dados distribuídos com garantias de consistência e disponibilidade mesmo com a ocorrência de faltas Bizantinas em algumas de suas réplicas. Algoritmos para sistemas de quóruns são reconhecidos por seus bons desempenho e escalabilidade, já que os clientes desse sistema faze m acesso, de fato, a somente um conjunto particular de servidores ao invés de todos os servidores.

Servidores em um sistema de quóruns organizam-se em subconjuntos denominados quóruns. Cada dois quóruns de um sistema mantêm um número suficiente de servidores corretos em comum (garantia de consistência), sendo que existe pelo menos um quórum no sistema formado somente por servidores corretos (garantia de disponibilidade) [Malkhi and Reiter 1998]. Os clientes realizam operações de leitura e escrita em registradores replicados por estes quóruns. Cada registrador detém um par $\langle v, t\rangle$ com um valor $v$ do dado armazenado e uma estampilha de tempo (*timestamp*) $t$ associada. Este *timestamp* é definido pelo cliente quando da sua escrita, sendo que cada cliente $c$ utiliza conjuntos disjuntos de *timestamps*.

Dentre as várias propostas de protocolos de leitura e escrita (R/W) para sistemas de quóruns [Malkhi and Reiter 1998, Liskov and Rodrigues 2006, Alchieri et al. 2009], este trabalho utiliza como base o PBFT-BC (*Proactive Byzantine fault-tolerance for Byzantine clients*) [Alchieri et al. 2009], adicionando mecanismos para suportar a reconfiguração no conjunto de servidores que mantêm o sistema e, com isso, tolerar a presença de clientes maliciosos em ambientes dinâmicos. Os protocolos de R/W do PBFT-BC foram escolhidos por apresentarem resiliência ótima ($n \geq 3f + 1$ servidores, utilizando quóruns de $\lceil \frac{n+f+1}{2} \rceil$ servidores para tolerar até $f$ faltas maliciosas em servidores), tolerarem clientes maliciosos e implementarem um registrador com semântica atômica [Lamport 1978]. Além disso, o PBFT-BC utiliza criptografia de limiar para controlar as ações dos clientes, onde através de um esquema $(n, \lceil \frac{n+f+1}{2} \rceil)$-TSS é necessário que pelo menos um quórum de servidores aprovem determinada ação de um cliente. Como veremos adiante, este mecanismo fornece a flexibilidade necessária para adaptação do protocolo às mudanças que ocorrem na composição do grupo de servidores em um ambiente dinâmico. Por exemplo, os clientes apenas precisam conhecer uma única chave pública do serviço (Seção 2.2), ao invés de todas as chaves públicas dos servidores.

Implementar um sistema de quóruns replicado em apenas $3f + 1$ servidores requer que os dados armazenados sejam auto-verificáveis, pois a intersecção entre os quóruns poderá conter apenas um servidor correto. Assim, os clientes obtêm corretamente os dados armazenados, a partir deste servidor, apenas se tais dados forem auto-verificáveis, pois deste modo é possível verificar a integridade dos mesmos. Neste sentido, a grande diferença do PBFT-BC em relação a outros sistemas que armazenam dados auto-verificáveis está justamente na forma de garan-

 ©2012 SBC — Soc. Bras. de Computação

tir a integridade destes dados, onde o PBFT-BC utiliza um assinatura gerada pelos servidores através do emprego de mecanismos de criptografia de limiar e outros protocolos, como o *f-dissemination quorum system* [Malkhi and Reiter 1998], utilizam assinaturas de clientes que, portanto, não podem ser maliciosos[2]. O BFT-BC [Liskov and Rodrigues 2006] utiliza assinaturas assimétricas de servidores para garantir a integridade dos dados armazenados e portanto, apesar de tolerar clientes maliciosos, não apresenta a flexibilidade necessária para utilização em sistemas distribuídos dinâmicos. A Seção 3.1 apresenta os algoritmos de leitura e escrita do PBFT-BC adaptados para os ambientes dinâmicos.

### 2.4. Dinamicidade do Sistema

Esta seção descreve brevemente os principais aspectos do QUINCUNX [Alchieri et al. 2012], um conjunto de protocolos para reconfiguração de sistemas de quóruns Bizantinos, que são utilizados como base para reconfiguração de nosso sistema. A grande vantagem do QUINCUNX é que seus protocolos de reconfiguração são completamente desacoplados dos protocolos de leitura e escrita, utilizados por clientes para acessar o registrador implementado pelo sistema. Desta forma, qualquer protocolo tradicional de sistemas de quóruns Bizantinos para ambientes estáticos é facilmente adaptado para ambientes dinâmicos através da utilização dos protocolos de reconfiguração do QUINCUNX.

Os protocolos de reconfiguração do QUINCUNX são responsáveis pela atualização da *visão* (*membership*) do sistema. No QUINCUNX, cada visão $v$ é uma tupla $\langle ov, entries, \mathtt{P}\rangle$, onde: $ov$ é a visão cuja atualização gerou $v$; $entries$ é um conjunto de *updates* e define o *membership* de $v$ (o *update* $\langle +, i\rangle$ define que $i \in v$ enquanto o *update* $\langle -, i\rangle$ define que $i \notin v$); e $\mathtt{P}$ é um certificado contendo as provas que garantem tanto a integridade de $v$ como também que $v$ foi instalada no sistema. Sendo $|v|$ o número de servidores presentes em $v$, no máximo $v.f = \lfloor \frac{|v|-1}{3} \rfloor$ servidores de $v$ podem falhar e o sistema utiliza quóruns de tamanho $v.q = \lceil \frac{|v|+v.f+1}{2} \rceil$. Duas visões $v_1$ e $v_2$ são comparadas através de seus campos $entries$. A notação $v_1 \subset v_2$ e $v_1 = v_2$ é utilizada como abreviação de $v_1.entries \subset v_2.entries$ e $v_1.entries = v_2.entries$, respectivamente. Caso $v_1 \subset v_2$, então $v_2$ é *mais atual* do que $v_1$ [Alchieri et al. 2012]. Uma visão é válida caso $\mathtt{P}$ possua as provas de que tal visão foi gerada e instalada no sistema (função $isValidV$ utilizada nos algoritmos da Seção 3).

Na inicialização do sistema, cada servidor $i \in v_0$ recebe a visão inicial $v_0 = \langle \perp, u_0, \emptyset\rangle$, onde $u_0$ representa o *membership* de $v_0$. Após isso, o sistema executa reconfigurações periódicas gerando e instalando novas visões, de forma que é possível determinar uma sequência de visões instaladas a partir de $v_0$ como $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow ....$ Para não acessar dados antigos, os protocolos de leitura e escrita executados pelos clientes sempre devem acessar a visão mais atual instalada no sistema. Um grande atrativo do QUINCUNX é que seus protocolos permitem a reconfiguração do sistema sem a necessidade de premissas temporais a respeito do comportamento do mesmo [Alchieri et al. 2012].

## 3. QUINCUNX-BC: Tolerando Clientes Maliciosos

Esta seção discute como integrar os protocolos do PBFT-BC aos do QUINCUNX, fazendo com que o sistema passe a tolerar também clientes maliciosos. É importante que o sistema apresente esta característica, pois clientes maliciosos podem tentar corromper o sistema através das seguintes ações [Liskov and Rodrigues 2006]: *(i)* escrever valores diferentes associados com o mesmo *timestamp*; *(ii)* executar o protocolo parcialmente, atualizando os valores armazenados

[2]O protocolo *f-masking quorum system* [Malkhi and Reiter 1998] também tolera algumas ações de clientes maliciosos. Porém, por não armazenar dados auto-verificáveis, requer replicação em $4f + 1$ servidores.

 ©2012 SBC — Soc. Bras. de Computação

por apenas algumas réplicas; *(iii)* escolher um *timestamp* muito grande, causando a exaustão do espaço de *timestamps*; e *(iv)* preparar um grande número de escritas e trabalhar junto com um aliado que executa estas operações mesmo após o cliente faltoso ser removido do sistema.

O desafio fundamental no desenvolvimento deste sistema, chamado QUINCUNX-BC (QUINCUNX *for byzantine clients*), está no projeto de um protocolo para redistribuição de chaves parciais que não deve fazer uso de nenhuma primitiva de sincronização para utilização em sistemas completamente assíncronos, além da adaptação dos algoritmos de leitura e escrita do PBFT-BC para ambientes dinâmicos. Além disso, como novos protocolos serão incorporados, novas informações devem fazer parte das visões do sistema. Desta forma, cada visão $v$ do sistema passa a possuir um campo $v.rd_info$ que contém informações sobre o procedimento de redistribuição de chaves, como veremos adiante.

Para adaptar os protocolos de leitura e escrita do PBFT-BC aos protocolos de reconfiguração do QUINCUNX, é necessário que cada cliente verifique a visão corrente do sistema na execução de cada fase do protocolo. Neste sentido, caso uma visão desatualizada esteja sendo utilizada, é necessário reiniciar a fase em execução. A Seção 3.1 apresenta estes protocolos.

Seguindo os limiares adotados no PBFT-BC [Alchieri et al. 2009], para cada visão $v$ do QUINCUNX-BC é definido um esquema de criptografia de limiar $(|v.members|, v.q)$-TSS, utilizado por servidores e clientes que executam operações na visão $v$. Como já comentado, na inicialização do sistema dinâmico cada servidor correto da visão inicial recebe sua chave parcial, que é secreta e nunca será revelada. Estas chaves são geradas e distribuídas por um administrador correto necessário para a inicialização do sistema. Após isso, um protocolo deve ser empregado para redistribuição de chaves parciais entre as visões instaladas no sistema. Este protocolo deve ser executado de modo que os servidores de uma visão anterior na sequência de visões instaladas redistribuam suas chaves parciais para os servidores da visão seguinte e assim por diante. Por exemplo, dada uma sequência de visões instaladas $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ...$, o esquema $(|v_0.members|, v_0.q)$-TSS deve sofrer uma redistribuição para o esquema $(|v_1.members|, v_1.q)$-TSS, que deve ser redistribuído para o esquema $(|v_2.members|, v_2.q)$-TSS, e assim por diante. A Seção 3.2 apresenta este protocolo, que deve ser executado sempre que uma visão mais atual é instalada no sistema.

### 3.1. Leitura e Escrita

Esta seção apresenta os algoritmos de leitura e escrita (R/W) do QUINCUNX-BC. Como a redistribuição de chaves parciais (Seção 3.2) e as reconfigurações [Alchieri et al. 2012] são desacopladas dos protocolos de R/W, estes protocolos são similares as suas versões estáticas [Alchieri et al. 2009]. A principal diferença é que em nosso sistema cada processo (cliente ou servidor) utiliza uma variável $cv$ para armazenar a visão mais atual conhecida.

Para correto funcionamento, os clientes devem escolher *timestamps* de subconjuntos diferentes. Assim, cada cliente concatena seu identificador único com um número de seqüência, i.e., $ts = \langle seq, id \rangle$. *Timestamps* são comparados verificando-se primeiramente o número de seqüência ($seq$) e posteriormente o identificador do cliente ($id$), caso os números de sequência sejam idênticos. *Timestamps* são incrementados através da função $succ(ts, c) = \langle ts.seq + 1, c \rangle$.

Para progredir nas operações, um cliente precisa provar que está agindo corretamente. Isto é realizado através do uso de **certificados**, que contêm os dados indicando a validade da ação que o cliente está tentando executar e uma assinatura de serviço que garante a integridade destes dados. Esta assinatura é gerada a partir de assinaturas parciais de $cv.q$ (um quórum) servidores de $cv$ que aprovaram a ação do cliente. Dois tipos de certificados são utilizados:

 ©2012 SBC — Soc. Bras. de Computação

**Certificado de Preparação**: o cliente utiliza este certificado para provar que preparou uma escrita (um quórum de servidores de $cv$ aprovam a escrita) e os servidores o usam para provar a integridade dos valores armazenados. Um certificado de preparação $cp$ possui três campos: $cp.ts$ – *timestamp* da escrita proposta; $cp.hash$ – *hash* do valor $v$ proposto para ser escrito; $cp.A$ – assinatura do serviço, provando que pelo menos um quórum de servidores de $cv$ aprovam a escrita de $v$ com o *timestamp* $cp.ts$. Para um certificado de preparação ser válido (função $isValidPC$ – algoritmos 1 e 2) é necessário que a assinatura $cp.A$, sobre a tupla $\langle cp.ts, cp.hash\rangle$, seja válida, o que é determinado pela operação $verify(\langle cp.ts, cp.hash\rangle, cp.A, PK)$.

**Certificado de Escrita**: o cliente utiliza este certificado para provar que terminou uma escrita. Um certificado de escrita $ce$ possui dois campos: $ce.ts$ – *timestamp* da escrita realizada; $ce.A$ – assinatura do serviço, provando que o cliente realizou a escrita relacionada com o *timestamp* $ce.ts$ em pelo menos um quórum de servidores de $cv$. Para um certificado de escrita ser válido (função $isValidWC$ – algoritmos 1 e 2) é necessário que a assinatura $ce.A$, sobre $ce.ts$, seja válida, o que é determinado pela operação $verify(ce.ts, ce.A, PK)$.

Este modelo suporta o armazenamento de múltiplos objetos nos servidores, desde que os mesmos tenham identificadores diferentes. Para simplificar a apresentação do protocolo, consideraremos que os servidores armazenam um único objeto. Deste modo, no QUINCUNX-BC cada servidor $i$ utiliza (armazena) as seguintes variáveis: **(1)** $data$ – valor do objeto; **(2)** $P_{cert}$ – certificado de preparação válido para $data$; **(3)** $P_{list}$ – conjunto de tuplas $\langle c, ts, hash\rangle$ contendo o identificador $c$ do cliente, o *timestamp* $ts$ e o *hash* do valor do objeto das escritas propostas; **(4)** $max_{ts}$ – *timestamp* relacionado com a última escrita completa conhecida por $i$; **(5)** $SK_i$ – chave parcial de $i$ para $cv$, usada pelo mecanismo de assinatura de limiar; **(6)** $VK$ e $Vk_j$ para cada $j \in cv$ – chaves de verificação para $cv$, usadas para gerar provas de validade de assinaturas parciais; e **(7)** $PK$ – chave pública do serviço usada para validar certificados. Finalmente, cada cliente $c$ utiliza as seguintes variáveis: **(1)** $W_{cert}$ – certificado de escrita referente à última escrita de $c$; **(2)** $PK$ – chave pública do serviço usada para validar certificados; e **(3)** $VK$ e $Vk_j$ para cada $j \in cv$ – chaves de verificação para $cv$, usadas para validar assinaturas parciais.

Para acessar o sistema, um cliente deve obter a visão atual do mesmo, o que pode ser realizado de duas formas [Alchieri et al. 2012]: (1) recuperar as visões a partir de um local pré-definido; ou (2) realizar uma inundação no sistema requisitando a visão atual. Os algoritmos 1 e 2 apresentam os protocolos de R/W executados por clientes e servidores, respectivamente.

**Escrita.** Após obter a visão atual do sistema e atualizar $cv$, um cliente $c$ que deseja escrever o valor $value$ no registrador primeiramente envia uma mensagem READ_TS para todos os servidores de $cv$ e aguarda por um quórum ($cv.q$) de respostas válidas. Cada servidor $i$ envia uma resposta $\langle$READ_TS_REP$, p, nonceResp, cv_i\rangle$ para $c$ (linha 2, alg. 2), onde $p$ é o $P_{cert}$ armazenado por $i$. Esta resposta é válida caso (linha 4, alg. 1): (1) $p$ é válido; (2) sua autenticidade é comprovada, i.e., os *nonces* da requisição e da resposta são iguais; e (3) a visão atual do servidor ($cv_i$) é válida e igual à visão $cv$ do cliente. Dentre os certificados de preparação recebidos, $c$ seleciona o certificado que contém o maior *timestamp*, chamado $P_{max}$ (linha 10, alg. 1) e define o *timestamp* $ts$ de sua escrita (linha 11, alg. 1).

Na segunda fase, $c$ envia uma mensagem PREPARE para os servidores de $cv$, preparando sua escrita com valor $value$ e *timestamp* $ts$. Esta mensagem contém as seguintes informações: (1) o *timestamp* $ts$ definido para a escrita e $P_{max}$ para provar que $ts$ é um valor válido e $c$ não está executando um ataque de exaustão de *timestamp* [Bazzi and Ding 2004, Liskov and Rodrigues 2006]); (2) o *hash* de $value$ para obter um certificado de preparação; (3) o certificado $W_{cert}$ da última escrita de $c$ ou $null$ caso seja a primeira escrita de $c$; e (4) a visão

 ©2012 SBC — Soc. Bras. de Computação

atual $cv$ de $c$. Então, $c$ aguarda por um quórum de respostas válidas. Cada servidor $i$ envia uma resposta $\langle$PREPARE_REP, $\langle ts_i, hash_i\rangle, \langle a_i, v_i\rangle, cv_i\rangle$ para $c$ (linha 27, alg. 2) com sua assinatura parcial $a_i$, utilizada por $c$ na construção do certificado de preparação. Assim, esta resposta é válida caso (linha 15, alg. 1): (1) $ts_i$ e $hash_i$ são os mesmos valores envidados na mensagem PREPARE; (2) a assinatura parcial de $i$ para $\langle ts_i, hash_i\rangle$ é válida; e (3) a visão atual do servidor ($cv_i$) é válida e igual a visão atual do cliente. O cliente $c$ combina as assinaturas parciais válidas recebidas e obtém uma assinatura de serviço para o par $\langle ts, h(value)\rangle$ (linha 21, alg. 1), construindo o certificado de preparação $P_{new}$ para $ts$ e $h(value)$ (linha 22, alg. 1).

A segunda fase é a mais importante, pois é na mesma que os servidores verificam se: (1) o *timestamp* sendo proposto é correto (linha 15, alg. 2); (2) o cliente esta preparando somente uma escrita (linha 22, alg. 2); (3) o valor proposto é o mesmo de algum (possível) valor anteriormente proposto para o mesmo *timestamp* (line 22, alg. 2); e (4) o cliente completou sua escrita anterior (linhas 16-21, alg. 2). Desta forma, um cliente malicioso $m_c$ não consegue preparar múltiplas escritas que poderão ser executadas por outro processo mesmo após $m_c$ ser removido do sistema, limitando o poder dos clientes maliciosos [Liskov and Rodrigues 2006].

**Algoritmo 1** Algoritmo executado pelo cliente *c*

**procedure** $write(value)$
{**Step 1**}
1: $send\langle$READ_TS, $nonceReq, cv\rangle$ *to each* $i \in cv$
2: **repeat**
3: **wait** $reply$ $\langle$READ_TS_REP, $p, nonceResp, cv_i\rangle$ $from$ $i \in cv$
4: **if** $isValidPC(p) \wedge nonceReq = nonceResp \wedge isValidV(cv_i) \wedge equals(cv, cv_i)$ **then**
5: $RTS_i \leftarrow p$
6: **else if** $isValidV(cv_i) \wedge updateClientView(cv_i)$ **then**
7: $restart\ step\ 1$
8: **end if**
9: **until** $|RTS| \geq cv.q$
10: $P_{max} \leftarrow max(RTS)$
11: $ts \leftarrow succ(P_{max}.ts, c)$
{**Step 2**}
12: $send\langle$PREPARE, $P_{max}, ts, h(value), W_{cert}, cv\rangle$ *to each* $i \in cv$
13: **repeat**
14: **wait** $reply$ $\langle$PREPARE_REP, $\langle ts_i, hash_i\rangle, \langle a_i, v_i\rangle, cv_i\rangle$ $from$ $i \in cv$
15: **if** $(hash_i = h(value)) \wedge isValidV(cv_i) \wedge equals(cv, cv_i) \wedge (ts_i = ts) \wedge Th_verifyS(\langle ts_i, hash_i\rangle, \langle a_i, v_i\rangle, PK, VK, VK_i)$ **then**
16: $PREPARED_i \leftarrow a_i$
17: **else if** $isValidV(cv_i) \wedge updateClientView(cv_i)$ **then**
18: $restart\ step\ 2$
19: **end if**
20: **until** $|PREPARED| \geq cv.q$
21: $A \leftarrow Th_combineS(PREPARED, PK)$
22: $P_{new} \leftarrow \langle ts, h(value), A\rangle$
{**Step 3**}
23: $send\langle$WRITE, $value, P_{new}, cv\rangle$ *to each* $i \in cv$
24: **repeat**
25: **wait** $reply$ $\langle$WRITE_REP, $ts_i, \langle a_i, v_i\rangle, cv_i\rangle$ $from$ $i \in cv$
26: **if** $(ts_i = ts) \wedge isValidV(cv_i) \wedge equals(cv, cv_i) \wedge Th_verifyS(ts_i, \langle a_i, v_i\rangle, PK, VK, VK_i)$ **then**
27: $WROTE_i \leftarrow a_i$
28: **else if** $isValidV(cv_i) \wedge updateClientView(cv_i)$ **then**
29: $restart\ step\ 3$
30: **end if**
31: **until** $|WROTE| \geq cv.q$
32: $A \leftarrow Th_combineS(WROTE, PK)$
33: $W_{cert} \leftarrow \langle ts, A\rangle$

**procedure** $read()$
{**Step 1**}
34: $send\langle$READ, $nonceReq, cv\rangle$ *to each* $i \in cv$
35: **repeat**
36: **wait** $reply$ $\langle$READ_REP, $value, p, nonceResp, cv_i\rangle$ $from$ $i \in cv$
37: **if** $isValidPC(p) \wedge (p.hash = h(value)) \wedge isValidV(cv_i) \wedge nonceReq = nonceResp \wedge equals(cv, cv_i)$ **then**
38: $READ_i \leftarrow \langle p, value\rangle$
39: **else if** $isValidV(cv_i) \wedge updateClientView(cv_i)$ **then**
40: $restart\ step\ 1$
41: **end if**
42: **until** $|READ| \geq cv.q$
43: $R_{max} \leftarrow maxTS(READ)$
{**Step 2 (write-back phase)**}
44: **if** $\exists\langle p, *\rangle \in READ$: $p.ts \neq R_{max}.p.ts$ **then**
45: $send\langle$WRITE, $R_{max}.value, R_{max}.p, cv\rangle$ *to each* $i \in cv$
46: **repeat**
47: **wait** $reply$ $\langle$WRITE_REP, $ts_i, \langle a_i, v_i\rangle, cv_i\rangle$ $from$ $i \in cv$
48: **if** $(ts_i = R_{max}.p.ts) \wedge isValidV(cv_i) \wedge equals(cv, cv_i)$ **then**
49: $WROTE_i \leftarrow a_i$
50: **else if** $isValidV(cv_i) \wedge updateClientView(cv_i)$ **then**
51: $restart\ step\ 2$
52: **end if**
53: **until** $|WROTE| \geq cv.q$
54: **end if**
55: **return** $R_{max}.value$

**procedure** $updateClientView(cv_i)$
56: **if** $((cv \subset cv_i) \vee (cv = cv_i \wedge cv.rd_info \succ cv_i.rd_info))$ **then**
57: $cv \leftarrow cv_i$
58: **for all** $j \in cv.members$ **do**
59: $update\ VK_j\ by\ fetching\ it\ from\ j\ for\ cv.rd_info$
60: **end for**
61: **return** $true$
62: **else**
63: **return** $false$
64: **end if**

**procedure** $equals(v_1, v_2)$
65: **if** $v_1 = v_2 \wedge v_1.rd_info = v_2.rd_info$ **then**
66: **return** $true$
67: **else**
68: **return** $false$
69: **end if**

Na última fase, $c$ envia uma mensagem WRITE para os servidores em $cv$, informando o valor a ser escrito juntamente com o certificado de preparação e aguarda por um quórum de respostas válidas. Cada servidor $i$ envia uma resposta $\langle$WRITE_REP, $ts_i, \langle a_i, v_i\rangle, cv_i\rangle$ para $c$ (linha 10, alg. 2) com sua assinatura parcial $a_i$ que será usada por $c$ na construção do certificado de escrita. Assim, esta resposta é válida caso (linha 26, alg. 1): (1) $ts_i$ é igual ao *timestamp* de escrita; (2) a assinatura parcial de $i$ para $ts_i$ é válida; e (3) a visão do servidor $cv_i$ é válida

 ©2012 SBC — Soc. Bras. de Computação

e igual a visão atual do cliente. Neste caso, $c$ combina as assinaturas parciais válidas recebidas e obtém a assinatura de serviço para $ts$ (linha 32, alg. 1), construindo o certificado de escrita $W_{cert}$ para sua última escrita (linha 33, alg. 1), o qual será usado em sua próxima escrita.

**Leitura.** Para ler o valor armazenado no sistema, um cliente $c$ envia uma mensagem READ para todos os servidores de $cv$ e aguarda por um quórum ($cv.q$) de respostas válidas. Cada servidor $i$ envia uma resposta $\langle$READ_REP$, data, p, nonceRep, cv_i\rangle$ para $c$ (linha 13, alg. 2), onde $p$ é um certificado de preparação válido para o valor $data$ armazenado em $i$. Esta resposta é válida caso (linha 37, alg. 1): (1) $p$ é válido para $data$; (2) a sua autenticidade é comprovada, i.e., os *nonces* da requisição e da resposta são iguais; e (3) a visão atual do servidor $cv_i$ é válida e igual a visão atual do cliente. Dentre todas as respostas recebidas, $c$ seleciona a resposta que possui o certificado de preparação com o maior *timestamp* (linha 43, alg. 1) e retornam o valor $value$ associado com esta resposta (linha 55, alg. 1). Caso nem todos os *timestamps* recebidos sejam iguais, $c$ executa uma fase de *write back* no sistema para o maior *timestamp* (linhas 44-54, alg. 1). Esta fase é semelhante à fase 3 do protocolo de escrita.

**Algoritmo 2** Algoritmo executado pelo servidor *i* para comunicação com clientes

**Upon receipt of** $\langle$READ_TS, $nonce, cv_c\rangle$ $from\ client\ c$
1: **checkServerView**($cv_c$)
2: $send\langle$READ_TS_REP, $P_{cert}, nonce, cv\rangle$ $to\ c$
**Upon receipt of** $\langle$WRITE, $value, P_{new}, cv_c\rangle$ $from\ client\ c$
3: **checkServerView**($cv_c$)
4: **if** $isValidPC(P_{new}) \wedge P_{new}.hash = h(value)$ **then**
5: **if** $P_{new}.ts > P_{cert}.ts$ **then**
6: $data \leftarrow value$
7: $P_{cert} \leftarrow P_{new}$
8: **end if**
9: $\langle a_i, v_i\rangle \leftarrow Th_sign(SK_i, VK_i, VK, P_{new}.ts)$
10: $send\langle$WRITE_REP, $P_{new}.ts, \langle a_i, v_i\rangle, cv\rangle$ $to\ c$
11: **end if**
**Upon receipt of** $\langle$READ, $nonce, cv_c\rangle$ $from\ client\ c$
12: **checkServerView**($cv_c$)
13: $send\langle$READ_REP, $data, P_{cert}, nonce, cv\rangle$ $to\ c$
**Upon receipt of** $\langle$PREPARE, $P_c, ts, hash, W_c, cv_c\rangle$ $from\ client\ c$
14: **checkServerView**($cv_c$)
15: **if** $isValidPC(P_c) \wedge isValidWC(W_c) \wedge ts = succ(P_c.ts, c)$ **then**
16: **if** $W_c \neq \perp$ **then**
17: $max_{ts} \leftarrow max(max_{ts}, W_c.ts)$
18: **for all** $e = \langle *, ts, *\rangle \in P_{list}: ts \leqslant max_{ts}$ **do**
19: $P_{list} \leftarrow P_{list} \setminus e$
20: **end for**
21: **end if**
22: **if** $\nexists\langle c, t, h\rangle \in P_{list}: t \neq ts \vee h \neq hash$ **then**
23: **if** $\langle c, ts, hash\rangle \notin P_{list} \wedge ts > max_{ts}$ **then**
24: $P_{list} \leftarrow P_{list} \cup \langle c, ts, hash\rangle$
25: **end if**
26: $\langle a_i, v_i\rangle \leftarrow Th_sign(SK_i, VK_i, VK, \langle ts, hash\rangle)$
27: $send\langle$PREPARE_REPLY, $\langle ts, hash\rangle, \langle a_i, v_i\rangle, cv\rangle$ $to\ c$
28: **end if**
29: **end if**

**Lidando com o dinamismo.** Para cada resposta recebida dos servidores, o cliente $c$ verifica se a visão do servidor é mais atual do que a sua visão $cv$. Nestes casos, $c$ atualiza a sua variável $cv$ (função $updateClientView$ – linhas 56-64, alg. 1 – o operador $\succ$ usado para comparar os campos $rd_info$ de duas visões é explicado na Seção 3.2) e reinicia a fase que está executando, garantindo que não irá acessar dados obsoletos. Além disso, $c$ envia sua visão atual $cv$ em todas as mensagens e cada servidor verifica se está atrasado em alguma reconfiguração (função $checkServerView$ usada no Algortimo 2). Nestes casos, o servidor deve aguardar até que os procedimentos de reconfiguração instalem esta visão atualizada e então responder para $c$. Usando esta técnica, um cliente apenas recebe respostas de processos corretos com visões iguais a $cv$ ou mais atuais, de forma que respostas com visões antigas podem ser descartadas.

### 3.2. Redistribuição de Chaves Parciais

Esta seção apresenta o protocolo para redistribuição de chaves parciais, onde os servidores de uma visão $v$ ($(|v.members|, v.q)$-TSS) redistribuem suas chaves parciais para os servidores da visão seguinte $w$ ($(|w.members|, w.q)$-TSS) na sequência de visões instaladas. Este protocolo, que pode ser utilizado por qualquer sistema que emprega criptografia de limiar e suporta mudanças no conjunto de processos que compõem o sistema, garante que os servidores da visão $w$ obtém suas chaves parciais a partir de *shares* gerados pelo mesmo conjunto de servidores presentes na visão $v$, o que é um requisito para que as novas chaves parciais sejam compatíveis e continuem gerando assinaturas válidas [Wong et al. 2002].

A ideia principal deste protocolo consiste em fazer com que os servidores de $v$ gerem os *shares* de suas chaves parciais e enviem cada *share* para o seu respectivo receptor de $w$.

 ©2012 SBC — Soc. Bras. de Computação

Após isso, todos os servidores de $w$ combinam os *shares* recebidos de um mesmo conjunto de processos de $v$ e obtêm as suas novas chaves parciais. Como servidores maliciosos de $v$ podem enviar *shares* inválidos, e além disso em um ambiente assíncrono não é possível garantir que as mensagens sejam entregues na mesma ordem nos servidores de $w$, estes servidores devem trocar mensagens até convergirem para a mesma informação a respeito da redistribuição RD_INFO, que é composta por dois conjuntos: RD_INFO.$good_set$ - conjunto de $v.q$ servidores de $v$ a partir dos quais os *shares* devem ser usados na geração das novas chaves parciais; RD_INFO.$bad_set$ - conjunto de servidores maliciosos de $v$ que devem ser desconsiderados.

O Algoritmo 3 apresenta este protocolo e deve ficar sempre ativo para atualizar a RD_INFO de uma visão $w$ assim que novas informações forem recebidas pelos servidores de $w$, causando a obtenção novas chaves parciais. Qualquer alteração na RD_INFO de um servidor é refletida em todos os outros servidores de $w$ (invariante mantido pelo algoritmo), de forma que estes servidores escolhem os *shares* do mesmo conjunto de processos.

**Algoritmo 3** Redistribuindo chaves parciais da visão $v$ para a $w$ (servidor *i*)

**variables:** {Sets and arrays used in the protocol}
RDK, RDV - set of encrypted shares and set of verifiers generated by $i$
GOOD, BAD - set of encrypted good shares and set of invalid shares collected by $i$
SHARES - set of good shares collected by $i$
RD_INFO - information about the redistribution

**procedure** $Partial_Keys_Redist(v, w)$
1: **if** $i \in v$ **then**
2: **for all** $j \in w$ **do**
3: $\widehat{k}_{ij} \leftarrow Th_shareK(SK_i, j, |w.members|, w.q)$; $\mathtt{RDK}^w \leftarrow \mathtt{RDK}^w \cup \langle j, encrypt(\widehat{k}_{ij})\rangle$
4: **end for**
5: $\mathtt{RDV}^w \leftarrow Th_generateV(i, |w.members|, w.q)$;
6: $\forall j \in w, send\langle$REDIST, $\langle \mathtt{RDK}^w, \mathtt{RDV}^w, v, w\rangle_i\rangle$ to $j$
7: **end if**

**Upon receipt of** $\langle$REDIST, $\langle rdk, rdv, v, w\rangle_j\rangle$ from $j \in v \wedge i \in w$
8: **if** $(\mathtt{GOOD}^w_j = \perp) \wedge (\mathtt{BAD}^w_j = \perp)$ **then**
9: $extractSK(\langle rdk, rdv, v, w\rangle_j)$
10: $updateSK(v, w)$
11: **end if**

**procedure** $extractSK(\langle rdk, rdv, v, w\rangle_j)$
12: **if** $(\forall z \in w \rightarrow \exists\langle z, encrypted_z\rangle \in rdk)$ **then**
13: $\widehat{k}_{ji} \leftarrow decrypt(encrypted_i)$
14: **if** $Th_verifySK(\widehat{k}_{ji}, rdv)$ **then**
15: $\mathtt{GOOD}^w_j \leftarrow \langle rdk, rdv, v, w\rangle_j$; $\mathtt{SHARES}^w_j \leftarrow \widehat{k}_{ji}$
16: **else**
17: $\mathtt{BAD}^w_j \leftarrow \langle j, \widehat{k}_{ji}, \langle rdk, rdv, v, w\rangle_j\rangle$
18: **end if**
19: **else**
20: $\mathtt{BAD}^w_j \leftarrow \langle j, \perp, \langle rdk, rdv, v, w\rangle_j\rangle$
21: **end if**

**procedure** $updateSK(v, w)$
22: $\mathtt{RD_INFO}^w.good_set \leftarrow$ *select* $v.q$ *members with messages in* $\mathtt{GOOD}^w$ *with lower ids*
23: **if** $|\mathtt{RD_INFO}^w.good_set| = v.q$ **then**
24: $\mathtt{RD_INFO}^w.bad_set \leftarrow$ *select all members with messages in* $\mathtt{BAD}^w$
25: **if** $\mathtt{RD_INFO}^w$ updated on line 24 or 26 **then**
26: $\langle SK_i, VK_i\rangle \leftarrow Th_combineSK(\mathtt{SHARES}^w, \mathtt{RD_INFO}^w.good_set)$
27: $\forall j \in w, send\langle$REDIST-INFO, $\mathtt{GOOD}^w, \mathtt{BAD}^w, v, w\rangle$ to $j$
28: $\forall j \in w$, *update* $VK_j$ *by fetching it from* $j$ *for* $\mathtt{RD_INFO}^w$
29: **end if**
30: **end if**

**Upon receipt of** $\langle$REDIST-INFO, $good, bad, v, w\rangle$ from $j$
31: **for all** $\langle z, \widehat{k}_{zx}, \langle rdk, rdv, v, w\rangle_z\rangle \in bad$: $(z \in v) \wedge (\mathtt{BAD}^w_z = \perp) \wedge isBad(\widehat{k}_{zx}, \langle rdk, rdv, v, w\rangle_z)$ **do**
32: $\mathtt{BAD}^w_z \leftarrow \langle z, \widehat{k}_{zx}, \langle rdk, rdv, v, w\rangle_z\rangle$; $\mathtt{GOOD}^w_z \leftarrow \perp$; $\mathtt{SHARES}^w_z \leftarrow \perp$
33: **end for**
34: **for all** $\langle rdk, rdv, v, w\rangle_z \in good$: $(z \in v) \wedge (\mathtt{GOOD}^w_z = \perp) \wedge (\mathtt{BAD}^w_z = \perp)$ **do**
35: $extractSK(\langle rdk, rdv, v, w\rangle_z)$
36: **end for**
37: $updateSK(v, w)$

 ©2012 SBC — Soc. Bras. de Computação

**Gerando e redistribuindo chaves parciais.** Para iniciar o processo de redistribuição de chaves parciais da visão $v$ para a visão $w$, todos os servidores de $v$ geram os *shares* de suas chaves parciais (linha 3), juntamente com os verificadores que atestam a validade destes *shares* (linha 5), e enviam estes dados para os servidores de $w$ (linha 6). Para cada portador $j \in w$, um servidor $i \in v$ gera o *share* $\widehat{k}_{ij}$ e o cifra com a chave pública de $j$, de modo que somente $j$ consegue acessá-lo. Todos os verificadores são públicos. Note que, `REDIST` é a única mensagem assinada (assinatura tradicional) nos protocolos do QUINCUNX-BC, sendo possível provar para terceiros que algum servidor enviou *shares* inválidos, como veremos adiante.

As chaves parciais dos servidores de $v$, utilizadas no procedimento de redistribuição, devem ter sido obtidas a partir do mesmo conjunto de *shares* ($\texttt{RD_INFO}^v.good_set$) [Wong et al. 2002]. Então, caso um servidor de $v$ atualizar sua chave parcial (atualizar $\texttt{RD_INFO}^v$), o mesmo deve executar novamente este procedimento e os servidores de $w$ devem obter suas chaves parciais a partir de *shares* obtidos de chaves parciais geradas através dos dados mais atuais em $\texttt{RD_INFO}^v.good_set$ (por simplicidade, isso não aparece no algoritmo).

**Convergindo para o mesmo `RD_INFO`.** Quando um servidor $i \in w$ recebe os *shares* de um servidor $j \in v$, $i$ verifica se ainda não recebeu qualquer informação de $j$ (linha 8), utilizando dois conjuntos: `GOOD` - conjunto de *shares* cifrados válidos; e `BAD` - conjunto de *shares* inválidos. Caso estas informações sejam novas, $i$ decifra seu *share* e caso o mesmo seja válido armazena estes dados em `SHARES` e `GOOD` (linha 15). Do contrário, $j$ é malicioso e estes dados são armazenados em `BAD` (linhas 17 e 20).

A função $updateSK$ é a parte principal do protocolo, pois através desta função sempre que alguma informação mais atual é recebida por algum servidor $i \in w$, $i$ atualiza sua chave parcial (caso necessário). Além disso, sempre que alguma nova informação for recebida e a chave parcial atualizada, $i$ envia uma mensagem REDIST-INFO para os servidores de $w$ contendo todas as informações coletadas, de forma que todos estes servidores possam coletar as mesmas informações sobre o processo de redistribuição das chaves parciais. Para isso, sempre que um servidor receber uma mensagem REDIST-INFO (linhas 31-37), o mesmo atualiza seus conjuntos `GOOD`, `BAD` e `SHARES` de acordo com as informações recebidas e executa a função $updateSK$ para verificar se uma chave parcial nova deve ser obtida.

Um servidor $i \in w$ verifica a informação de que um servidor $z \in v$ gerou um *share* inválido (função $isBad(\widehat{k}_{zx}, \langle rdk, rdv, v, w\rangle_z)$) da seguinte forma: (1) $i$ verifica se $z$ gerou *shares* para todos os servidores de $w$ (como na linha 12); (2) $i$ cifra o *share* $\widehat{k}_{zx}$ (*share* enviado para um servidor $x \in w$) com a chave pública de $x$ e verifica se o resultado é igual ao *share* cifrado enviado de $z$ para $x$ ($encrypted_x$ contido em $rdk$), e caso seja igual, $i$ verifica a validade de $\widehat{k}_{zx}$ através da função $Th_verifySK(\widehat{k}_{zx}, rdv)$. Uma vez que $i$ descobre que $z$ é malicioso, $i$ adiciona $z$ no conjunto `BAD`, de onde nunca é removido.

**Comparando `RD_INFO`.** Para que os processos possam obter assinaturas válidas, os mesmos devem usar assinaturas parciais geradas por chaves compatíveis. Para isso, as mensagens trocadas entre os processos (tanto entre servidores para redistribuição de chaves parciais quanto entre clientes e servidores para execução de operações de leitura e escrita) devem conter a informação `RD_INFO` usada na obtenção da chave parcial mais atual do emissor, de modo que os processos apenas considerem as mensagens acompanhadas de informações mais atuais. Como já discutido, este dado faz parte das visões do sistema (campo $rd_info$) e deve ser obtido um certificado para provar a autenticidade do mesmo, uma vez que são trocados entre os processos. Isto pode ser realizado através da obtenção de uma assinatura de serviço (usando o próprio mecanismo de criptografia de limiar) que deve ser adicionada ao campo `P` das visões (por simplicidade, não

 ©2012 SBC — Soc. Bras. de Computação

apresentamos este passo no algoritmo).

Dadas duas informações sobre a redistribuição $rd_info_1$ e $rd_info_2$, $rd_info_2$ é mais atual do que $rd_info_1$ (representado por $rd_info_1 \succ rd_info_2$ – Algoritmo 1) se $max(rd_info_1.good_set) > max(rd_info_2.good_set) \wedge (\forall j \in rd_info_2.good_set \rightarrow j \notin rd_info_1.bad_set)$: o maior identificador em $rd_info_2.good_set$ é menor do que o maior identificador em $rd_info_1.good_set$, o que implica que $rd_info_2$ é formado por servidores com identificadores menores. A segunda parte da condição verifica se o conjunto $rd_info_2.good_set$ é formado apenas por servidores corretos.

### 3.2.1. *Corretude*

O seguinte teorema prova a propriedade fundamental garantida pelo Algoritmo 3.

**Teorema 1** *Considere a redistribuição de chaves parciais de uma visão $v$ para uma visão $w$ através do Algoritmo 3. Cada servidor $i \in w$ obtêm uma nova chave parcial e este algoritmo mantém o invariante de que as chaves parciais definidas para os servidores de $w$ sempre acabam por serem compatíveis.*

**Prova.** Cada servidor correto de $v$ define os *shares* de sua chave parcial e envia estes dados para os servidores de $w$ (linhas 1-7). Como $v$ possui pelo menos $v.q$ servidores corretos e os canais são confiáveis, o predicado da linha 23 será verdadeiro pelo menos uma vez em cada servidor de $w$, que obterá uma nova chave parcial. Agora temos que provar que os servidores de $w$ sempre acabam definindo o mesmo `RD_INFO`, obtendo suas chaves parciais a partir do mesmo conjunto de servidores de $v$. Sempre que um servidor $i \in w$ alterar sua chave parcial (linha 26) e seu `RD_INFO` (linhas 22 e 24), $i$ envia todas as informações coletadas para os outros servidores de $w$ (linha 27), que também atualizam seus conjuntos `RD_INFO` (linhas 31-37). Desta forma, todos os servidores de $w$ acabam definindo um mesmo `RD_INFO`. Como os servidores de $w$ obtém suas chaves parciais a partir dos *shares* gerados pelo quórum de servidores corretos de $v$ com menores identificadores (linha 22) e acabam definindo o mesmo `RD_INFO`, todos estes servidores acabam obtendo novas chaves parciais compatíveis. Como sempre que algum servidor de $w$ atualiza sua chave parcial e seu `RD_INFO` o mesmo envia estas informações para os outros servidores de $w$, este invariante sempre é mantido no sistema. $\square_{Teorema\ 1}$

## 4. Discussões

**Lidando com clientes maliciosos.** O QUINCUNX-BC limita o número de escritas incompletas de um cliente usando uma lista $P_{list}$ em cada servidor, a qual armazena as escritas preparadas pelos clientes mas ainda não completadas (Algoritmo 2). Considerando que o sistema sofre uma reconfiguração da visão $view_{old}$ para a visão $view_{new}$. Caso um quórum de servidores ($view_{new}.q$) entraram no sistema em $view_{new}$ com suas $P_{list}$ vazias, um cliente malicioso pode preparar uma escrita $w$ em $view_{new}$ sem que tenha completado alguma escrita $w'$ preparada anteriormente em $view_{old}$. Isso significa que o QUINCUNX-BC garante que cada cliente consegue ter no máximo uma escrita preparada mas incompleta por visão, diferentemente do PBFT-BC [Alchieri et al. 2009] que não sofre reconfigurações e, por isso, cada cliente pode ter no máximo uma escrita incompleta durante todo o ciclo de vida do sistema.

**Desempenho.** As otimizações propostas para o protocolo de escrita do PBFT-BC [Alchieri et al. 2009] são facilmente incorporadas ao protocolo de escrita do QUINCUNX-BC, possibilitando que este termine em apenas 2 fases de execução. Além disso, como os protocolos de reconfiguração e de redistribuição de chaves são desacoplados dos protocolos de leitura

 ©2012 SBC — Soc. Bras. de Computação

e escrita, é possível que estas operações sejam executadas concorrentemente, sendo necessário apenas a reinicialização de alguma fase da leitura e/ou da escrita quando uma nova visão for instalada no sistema e/ou novas chaves parciais forem definidas.

## 5. Conclusões

Este artigo apresentou o QUINCUNX-BC, um sistema de quóruns Bizantinos dinâmico capaz de tolerar o comportamento malicioso tanto de servidores quanto de clientes através do emprego de criptografia de limiar, mecanismo que fornece a flexibilidade suficiente para operação em ambientes dinâmicos. Preservando as características dos protocolos utilizados como base, no QUINCUNX-BC as operações de R/W são completamente desacopladas dos procedimentos de reconfiguração, o que permite concorrência entre R/W e reconfigurações. Esta característica tende à aumentar o desempenho do sistema, principalmente durante reconfigurações.

## Referências


Aguilera, M. (2004). A pleasant stroll through the land of infinitely many creatures. *SIGACT News*, 35(2):36–59.

Aguilera, M. K., Keidar, I., Malkhi, D., and Shraer, A. (2011). Dynamic atomic storage without consensus. *JACM*, 58:7:1–7:32.

Alchieri, E. A., Bessani, A. N., Silva Fraga, J., and Greve, F. (2012). Memória compartilhada em sistemas bizantinos dinâmicos. In *Anais do XIII Workshop de Teste e Tolerância a Falhas- WTF 2012*.

Alchieri, E. A. P., Bessani, A. N., Pereira, F., and da Silva Fraga, J. (2009). Sistemas de quóruns bizantinos pró-ativos. *Revista Brasileira de Redes de Computadores e Sistemas Distribuídos*, 2:21–34.

Bazzi, R. A. and Ding, Y. (2004). Non-skipping timestamps for Byzantine data storage systems. In *Proc. of 18th Int. Symposium on Distributed Computing, DISC 2004*, volume 3274 of *LNCS*, pages 405–419.

Bellare, M. and Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of the 1st ACM Conference on Computer and Communications Security*, pages 62–73.

Desmedt, Y. and Frankel, Y. (1990). Threshold criptosystems. In *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology - CRYPTO'89*, pages 307–315. Springer-Verlag.

Gifford, D. (1979). Weighted voting for replicated data. In *Proc. of the 7th ACM Symposium on Operating Systems Principles*, pages 150–162.

Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programing Languages and Systems*, 4(3):382–401.

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565.

Liskov, B. and Rodrigues, R. S. M. (2006). Tolerating byzantine faulty clients in a quorum system. In *The 26th IEEE International Conference on Distributed Computing Systems - ICDCS 2006*.

Lynch, N. and Shvartsman, A. A. (2002). Rambo: A reconfigurable atomic memory service for dynamic networks. In *16th International Symposium on Distributed Computing - DISC*, pages 173–190.

Malkhi, D. and Reiter, M. (1998). Byzantine quorum systems. *Distributed Computing*, 11(4):203–213.

Martin, J.-P. and Alvisi, L. (2004). A framework for dynamic Byzantine storage. In *Proceedings of the International Conference on Dependable Systems and Networks*. IEEE Computer Society.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

Rodrigues, R. and Liskov, B. (2004). Rosebud: A scalable Byzantine-fault-tolerant storage architecture. MIT-LCS-TR 932, MIT Laboratory for Computer Science.

Shoup, V. (2000). Practical threshold signatures. In *Advances in Cryptology: EUROCRYPT 2000, Lecture Notes in Computer Science*, volume 1807, pages 207–222. Springer-Verlag.

Wong, T. M., Wang, C., and Wing, J. M. (2002). Verifiable secret redistribution for archive systems. In *Proceedings of the 1th International IEEE Security in Storage Workshop*.


 ©2012 SBC — Soc. Bras. de Computação

# A Secure Relay Protocol for Door Access Control*

**Erik Ramsgaard Wognsen, Henrik Søndberg Karlsen, Marcus Calverley**
**Mikkel Normann Follin, Bent Thomsen, Hans Hüttel**

[1] Department of Computer Science, Aalborg University
Aalborg, Denmark

{hkarls07,mcalve07,mfolli07}@student.aau.dk
{erw,bt,hans}@cs.aau.dk

***Abstract.*** *Physical keys are easy to use but difficult to manage securely for large institutions. Digital replacements have been created, but dedicated hardware such as smartcards or RFID tags can have the same problems as physical keys. Several commercial products try to solve this by using the users' Bluetooth-enabled mobile devices as keys, but the built-in security of the Bluetooth standard is insufficient. Furthermore, to manage a varying set of users, such systems may require the door locks to be connected to the Internet which may require expensive infrastructure.*
*We present a cryptographic protocol and a prototype implementation that solves these problems by letting door locks communicate with a central server using the Internet connections of the users' mobile devices. The protocol is specified formally in the applied π-calculus and security through secrecy and authenticity is verified using the cryptographic protocol verifier ProVerif. A prototype of the system is implemented for Android smartphones.*

## 1. Introduction

Keys have for centuries been a central part of managing access rights to buildings. Physical keys are easy to use, but difficult to manage, especially for large institutions with a large and varying number of users and even worse when access rights to a building change over time. Digital replacements for physical key systems can solve this problem and they have been around for several decades, but until recently required dedicated hardware to be issued to the user, e.g. smartcards or RFID tags, which to some extent are as cumbersome to manage as their physical key counter parts. With the proliferation of commodity smartphones the idea of using a smartphone as a wireless key has been commercialized by several vendors such as MVC-Data [1], Bluelon ApS [2], Flexipanel Ltd. [3], Steab AB [4], ECKey [5], and SOREX [6]. Several of these commercial systems are based on establishing a Bluetooth connection between door and smartphone and rely on the Bluetooth stack for security. Although this provides "good enough" security between the door and the smartphone for many practical uses, it is possible to compromise the security with some effort and modified consumer hardware [7]. Furthermore, to manage a varying set of users, such systems either require the door locks to be connected to the Internet, or manual entry of each user in each door lock.

In this paper we present a system, developed in collaboration with MVC-Data, that can solve both problems mentioned above. The system is designed so that each door lock

*This work is partially funded by the European Regional Development Fund and Centre for Embedded Software Systems (CISS)

 ©2012 SBC — Soc. Bras. de Computação

requires no knowledge of the individual users, as well as no direct Internet connection. Instead, the door locks use the Internet connection provided by the user's smartphone to set up a secure connection with a central server which holds the information about each user, their access rights and audit logs. For communication between the door lock and the smartphone, the system uses an insecure Bluetooth connection without pairing. Figure 1 illustrates the architecture. Informally, the protocol works as follows: The smartphone app runs a service in the background that keeps a connection to the server and scans for door units. When the smartphone gets within a predefined range of a door unit, it connects to the door unit and asks it to unlock. To find out if it should open, the door unit consults the server through the Internet connection of the smartphone.

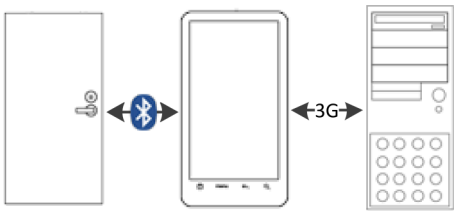



**Figure 1. The architecture: a smartphone creates a link between a door module and a server through Bluetooth and an Internet connection.**

This paper has two main contributions: a formal protocol and verification of relevant security properties of the protocol. The protocol uses a public key infrastructure where a central authentication server is able to authenticate users for individual door locks, based on their private key which they save on their smartphones. It is designed to allow users to securely unlock doors through an insecure Bluetooth connection and authenticate them through an insecure Internet connection. The protocol provides a secure layer using hybrid encryption such that asymmetric encryption is used for authentication and to establish a faster, symmetric encryption for larger message exchanges. The protocol is formalized in the applied $\pi$-calculus [8, 9]. Furthermore, the protocol is verified using ProVerif [10] to be secure against eavesdropping (by strong secrecy), man-in-the-middle attacks, and replay attacks.

To the best of our knowledge, ours is the first formally verified wireless protocol using Bluetooth and 3G communication.

ProVerif has been used to verify the security of several protocols including an electronic voting protocol [11] and implementations of TLS [12]. Additionally, in [13], ProVerif was used to analyze the authentication process of Bluetooth device pairing, where it was able to determine an attack strategy that confirms a previously known vulnerability. A list of these and other publications using ProVerif can be found in [14].

The rest of this paper is organized as follows: in Section 2, an informal description of the protocol is given followed by a classic protocol narration in the form of message sequence charts. In Section 3 the applied $\pi$-calculus is introduced and in Section 4 it is used to formalize the protocol. The applied $\pi$-calculus is used as an intermediate step which helps in the description and verification of properties using ProVerif in Section 5. Section 6 describes the implementation and illustrates how the ProVerif model relates to the prototype implementation. Section 7 concludes.

 ©2012 SBC — Soc. Bras. de Computação

## 2. The Protocol Narration

The protocol narration describes each step in the protocol in a manageable way, but it leaves out many details such as checks that must be done at each step to know when to abort communication because of an attack. These details are instead included in the applied π-calculus model. Table 1 describes the notation we use in the protocol narration.

**Table 1. Notation for the protocol narration.**

| | |
|---|---|
| $M, S, D$ | Identities of the mobile unit, server and door unit |
| $K_X, K_X^{-1}$ | The public and private key of principal $X$ |
| $\{x\}_K, \{x\}_{K^{-1}}$ | Public and private encryption (signature) of message $x$ |
| $[x]$ | A cryptographic hash of message $x$ |
| $N_x$ | A nonce named $x$ |
| $EK_{XY}, AK_{XY}$ | Encryption and authentication keys for symmetrically encrypted communication from $X$ to $Y$ |
| $K_{XY}$ | Key material for deriving $EK_{XY}$, $AK_{XY}$, $EK_{YX}$, and $AK_{YX}$ |
| $[x]^h_{AK}$ | MAC of message $x$ using authentication key $AK$ |
| $\{x\}^s_{EK}$ | Symmetric encryption of message $x$ with encryption key $EK$ |
| $x$ | An arbitrary value |
| "foo" | A string literal |
| $m_x = \ldots$ | A subexpression introduced for clarity |

Before interacting with any doors, the mobile unit $M$ connects to the server $S$ and is authenticated by the server using asymmetric encryption. Figure 2 shows the messages exchanged during this first phase. The data that is encrypted asymmetrically is key material that will be used to encrypt all following messages symmetrically. Nonces are sent in every step to avoid replay attacks and each nonce is returned to prove the receiver was able to decrypt the nonce. After the symmetric keys are exchanged and the principals trust each other, the channel is ready to send arbitrary messages as shown by the message $x$ and the ellipsis. The mobile unit can now look for a door unit to connect to.

When the connection to the server $S$ is established and the mobile unit $M$ wants to connect to a door unit $D$, it queries $S$ for the public key of $D$. It then uses asymmetric encryption to share key material with $D$ for the symmetrically encrypted communication with nonces being sent and returned. Figure 3 shows this process. As before, the mobile unit verifies the identity of its communication partner, but since the server will later decide if the mobile unit is allowed access, the door does not need to verify the identity of the mobile unit.

Figure 4 shows what happens when the user wants to unlock the door after having established connections to the server and the door. It is the most complicated figure since communication is relayed through the mobile unit which takes the output from one secure channel and sends it over the other while the messages themselves are the encrypted

 ©2012 SBC — Soc. Bras. de Computação

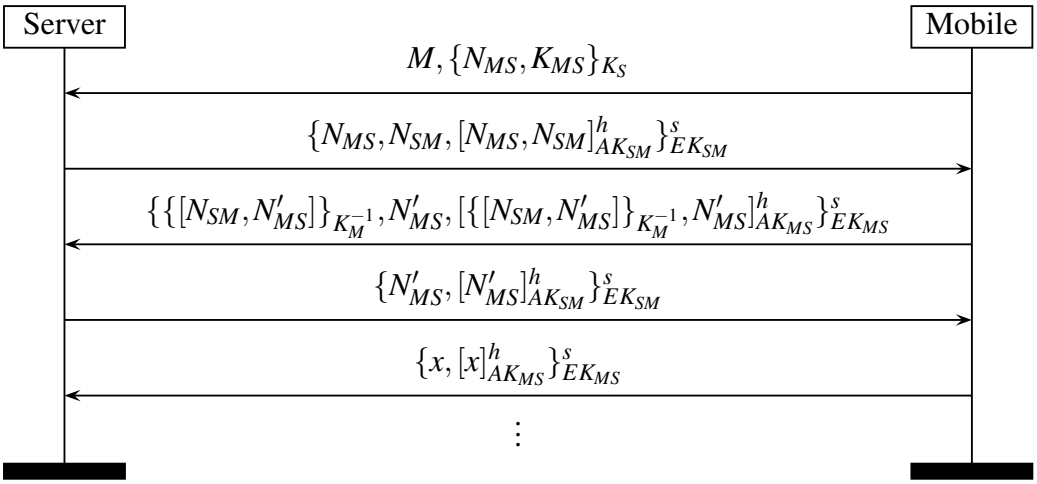



**Figure 2. The mobile unit connects to the server.**

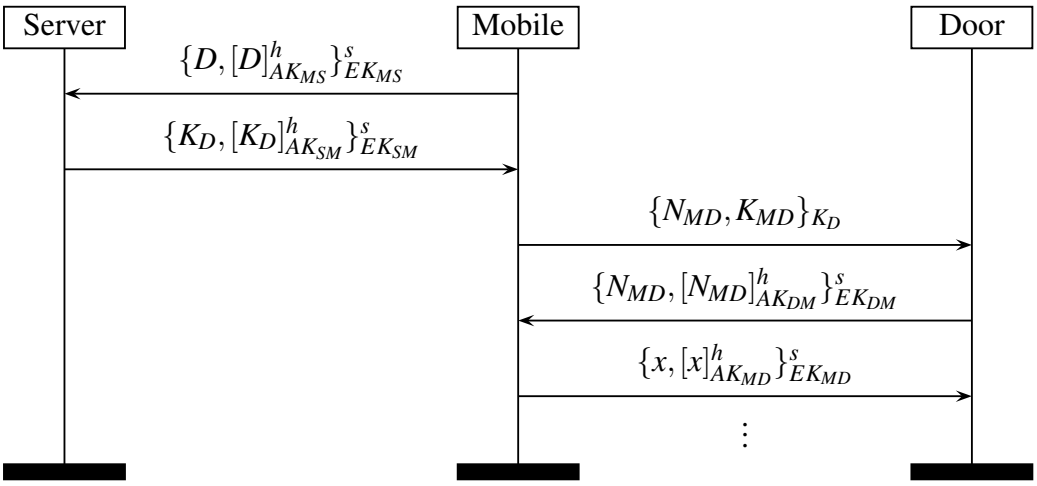



**Figure 3. The mobile unit connects to a door after having established a connection with the server.**

challenges between door and server. When the door is authenticated the server looks up whether the owner of the mobile unit has access to the door at that time and sends "1" or "0" in $m_4$ depending on whether the user is allowed access or not.

## 3. The Applied π-calculus

The applied π-calculus was proposed by Abadi and Fournet [15] as a generalization of the many extensions of the π-calculus [16] in existence. Since then, Blanchet has used a version of the applied π-calculus as the basis of the ProVerif tool.

While the protocol narration describes the messages exchanged, the applied π-calculus model formalizes exactly what properties must be checked by each principal at each step.

 ©2012 SBC — Soc. Bras. de Computação

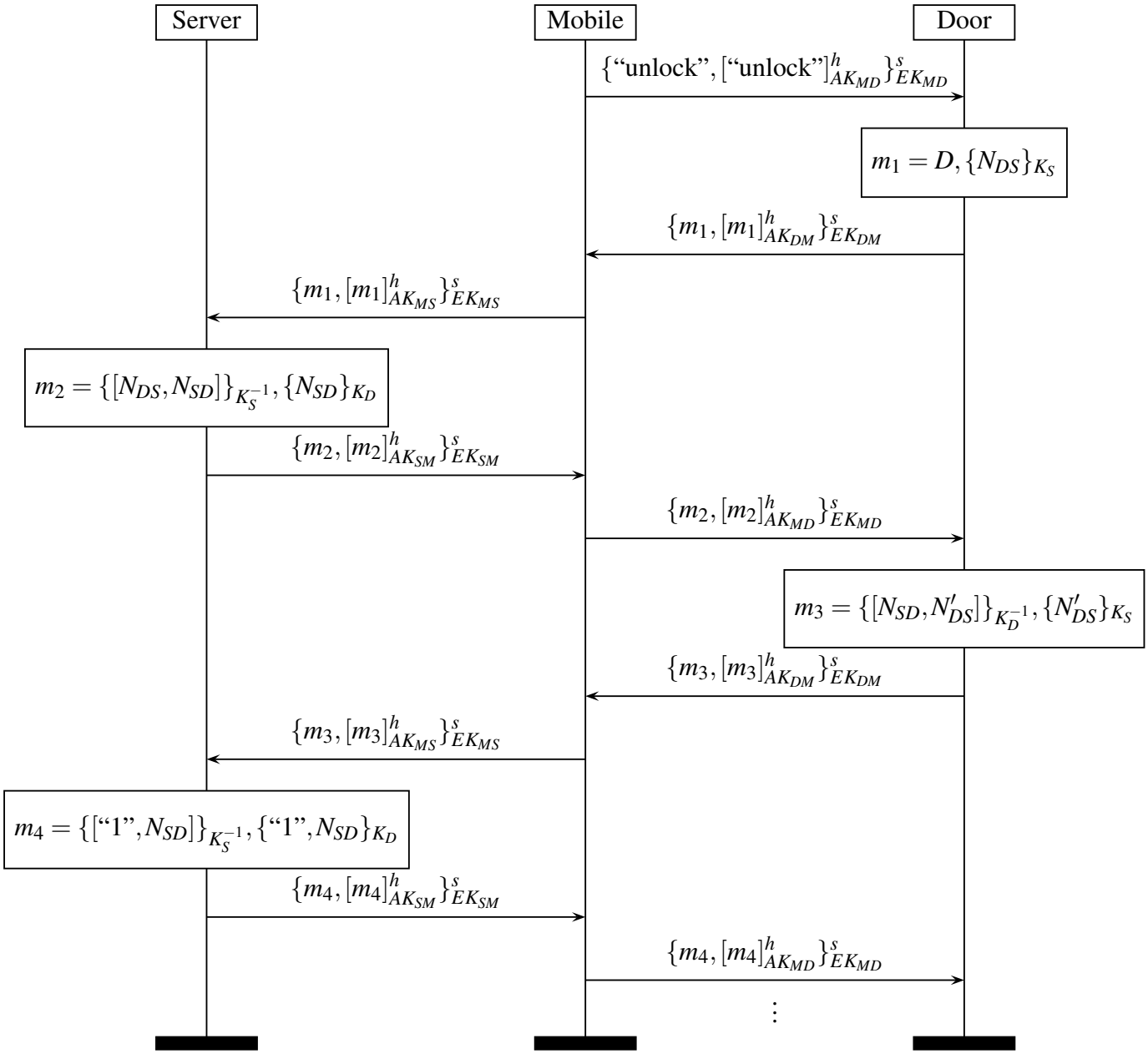



**Figure 4. The door is unlocked after having the mobile unit establish connections to server and door so that the door and server can communicate through the mobile unit.**

The syntax of applied π-calculus processes presupposes a set of terms built from a finite signature and a countable set of names, ranged over by $x, y, z \ldots$. The syntax of processes is then given by the following formation rules where terms are here ranged over by $M$ and $N$.

$$P ::= x(y).P_1 \mid \overline{x}\langle N\rangle.P_1 \mid (P_1|P_2) \mid (\nu x)P_1 \mid !P_1 \mid \{M/y\} \mid [M = N]P_1$$

Here, $x(y).P_1$ denotes an input process that on the channel named $x$ can receive a name, instantiating $y$ within the continuation $P_1$. The output process $\overline{x}\langle N\rangle.P_1$ sends out the term

 ©2012 SBC — Soc. Bras. de Computação

$N$ on channel $x$. $P_1|P_2$ denotes the parallel composition of processes $P_1$ and $P_2$. $(\nu x)P_1$ denotes that the name $x$ is private to and bound within $P_1$. $!P_1$ denotes an unbounded supply of parallel components of the form $P_1$. An active substitution, $\{M/y\}$, denotes that the name $y$ is bound to $M$. Finally, $[M = N]P_1$ is a match process that can only proceed as $P_1$ if the message terms $M$ and $N$ are provably equal.

In our presentation, we sometimes define processes by means of parameterized definitions of the form

$$A(\vec{x}) \stackrel{\text{def}}{=} P$$

where $A$ is an agent identifier that may also occur in the right-hand-side process $P$.

The semantics of the applied π-calculus uses reduction semantics (see Abadi and Fournet for the definition) with reductions of the form

$$P \rightarrow P'$$

We write $P \rightarrow^* P'$, if $P$ becomes $P'$ within zero or more reduction steps.

## 4. The Protocol in the Applied π-calculus

The protocol is represented by the following process, where the pk function is defined in Table 2:

$$\begin{aligned}\mathcal{P}(K_S^{-1}, K_M^{-1}, K_D^{-1}, c_i, c_b, c_m, l_M, f_D) \stackrel{\text{def}}{=} \\ !\,\mathcal{P}_M(K_M^{-1}, \mathsf{pk}(K_S^{-1}), c_i, c_b) \mid \\ !\,\mathcal{P}_S(K_S^{-1}, c_i, c_m, l_M, f_D) \mid \\ !\,\mathcal{P}_D(K_D^{-1}, \mathsf{pk}(K_S^{-1}), c_b, c_m)\end{aligned}$$

The process is parameterized by a number of variables: the private keys of the three parties, the public key of the server, three public channels, a set of mobile units $l_M$ representing the server database of identities of registered mobile units, and a function $f_D$ mapping the Bluetooth addresses of the registered door units to their identities. Channels $c_i$ and $c_b$ represent communication over the Internet and over Bluetooth, respectively. When the communication link between the door and server is set up using the mobile unit as a relay, the public channel $c_m$ is used to model the mobile unit as an untrusted party.

Public channels may be monitored by any attacker and manipulated. Messages in transit may be dropped, altered, replayed, replaced, etc. This is typically how cryptographic adversaries are modeled.

**Table 2. One-way functions**

| | |
|---|---|
| $\mathsf{pk}(K^{-1})$ | Derive public key from private key |
| $\mathsf{hash}(m)$ | Cryptographic hash of message $m$ |
| $\mathsf{mac}(K,m)$ | MAC of message $m$ using key $K$ |
| $\mathsf{ekxy}(K_{XY})$ | Derive encryption key for symmetric communication $X \rightarrow Y$ |
| $\mathsf{ekyx}(K_{XY})$ | Derive encryption key for symmetric communication $Y \rightarrow X$ |
| $\mathsf{akxy}(K_{XY})$ | Derive authentication key for symmetric communication $X \rightarrow Y$ |
| $\mathsf{akyx}(K_{XY})$ | Derive authentication key for symmetric communication $Y \rightarrow X$ |

 ©2012 SBC — Soc. Bras. de Computação

**Table 3. Functions with reductions**

| | |
|---|---|
| $\mathsf{publicEncrypt}(\mathsf{pk}(K^{-1}),x)$ | —Abbreviated as $\{x\}_{\mathsf{pk}(K^{-1})}$ |
| $\mathsf{privateDecrypt}(K^{-1},\mathsf{publicEncrypt}(\mathsf{pk}(K^{-1}),x)) = x$ | |
| $\mathsf{sign}(K^{-1},x)$ | |
| $\mathsf{stripSign}(\mathsf{pk}(K^{-1}),\mathsf{sign}(K^{-1},x)) = x$ | |
| $\mathsf{symEncrypt}(K,IV,x)$ | —Abbreviated as $\{x\}_K^{IV}$ |
| $\mathsf{symDecrypt}(K,IV,\mathsf{symEncrypt}(K,IV,x)) = x$ | |
| $\mathsf{m_i}(x_0,x_1,\ldots,x_n)$ | —Packing components of message $i$ |
| $\mathsf{m_i.j}(\mathsf{m_i}(x_0,x_1,\ldots,x_n)) = x_j \quad \text{for } j \in 0,1,\ldots,n$ | —Unpacking |
| $\mathsf{w}(x_0,x_1,x_2)$ | —Wrapping message with meta data |
| $\mathsf{w.j}(\mathsf{w}(x_0,x_1,x_2)) = x_j \quad \text{for } j \in 0,1,2$ | —Unwrapping |

Table 2 shows the one-way functions used in the model while Table 3 shows the function reductions. Most functions represent standard cryptographic operations in idealized forms. The packing/wrapping functions have no effect on the protocol other than providing a familiar structure for the processes. Packing is used for grouping arbitrary components into a single message while wrapping is used for grouping a message with an initialization vector (IV) and a message authentication code (MAC) for the symmetric encryption.

For convenience, notation for decryption with pattern matching is introduced as in [8] such that the process

$$\text{if } ct^n_{XY} = \{\mathsf{w}(\nu m^n_{XY},\nu IV^n_{XY},\nu H^n_{XY})\}^{IV^{n-1}_{XY}}_{\mathsf{ekyx}(K_{XY})} \text{ using } \mathsf{ekyx}(K_{XY}) \text{ then } Q$$

is equivalent to the following which includes checking that the computed MAC matches the received MAC:

$$\nu m^n_{XY}.\nu IV^n_{XY}.\nu H^n_{XY}.\nu HC^n_{XY}.$$
$$\left(\begin{array}{lcl} \{\ m^n_{XY} & = & \mathsf{w.0}(\mathsf{symDecrypt}(\mathsf{ekyx}(K_{XY}),IV^{n-1}_{XY},ct^n_{XY}))\}\,| \\ \{\ IV^n_{XY} & = & \mathsf{w.1}(\mathsf{symDecrypt}(\mathsf{ekyx}(K_{XY}),IV^{n-1}_{XY},ct^n_{XY}))\}\,| \\ \{\ H^n_{XY} & = & \mathsf{w.2}(\mathsf{symDecrypt}(\mathsf{ekyx}(K_{XY}),IV^{n-1}_{XY},ct^n_{XY}))\}\,| \\ \{\ HC^n_{XY} & = & \mathsf{mac}(\mathsf{akyx}(K_{XY}),(m^n_{XY},IV^n_{XY}))\}\,| \\ \multicolumn{3}{l}{\text{if } ct^n_{XY} = \{\mathsf{w}(m^n_{XY},IV^n_{XY},H^n_{XY})\}^{IV^{n-1}_{XY}}_{\mathsf{ekyx}(K_{XY})} \text{ and } H^n_{XY} = HC^n_{XY} \text{ then } Q} \end{array}\right)$$

The first steps of $\mathcal{P}_M$ and $\mathcal{P}_S$ are shown here in order to explain the meaning of the processes and their interaction. Instead of simply establishing a shared secret in a few steps as many protocols do, this protocol consists of several parts with variable numbers of steps each. One specific, minimal run of the full protocol is modeled. Since the protocol also includes more low-level details (including hybrid encryption) than many other protocols, the applied π-calculus formalization is relatively long. The full processes are found in Figures 5, 6, and 7.

First part of $\mathcal{P}_M$ derives the corresponding public key from its private key and then creates a nonce, an IV and key material for the symmetric communication. The public key is sent over the Internet channel along with the nonce, IV, and key material encrypted

 ©2012 SBC — Soc. Bras. de Computação

with the public key of the server:

$$\text{let } K_M = \mathsf{pk}(K_M^{-1}) \text{ in } \nu N_{MS}.\nu IV_{MS}^1.\nu K_{MS}.$$
$$\overline{c_i}\langle K_M, \{\mathsf{m}_{\mathsf{MS}}^1(N_{MS}, IV_{MS}^1, K_{MS})\}_{K_S}\rangle$$

The server receives the messages sent by the mobile unit and checks that it is registered before continuing. The encrypted message is decrypted and the components extracted. Then, a new nonce is created a sent back with the one from the mobile unit using symmetric encryption with the key material and IV sent by the mobile unit. A new IV is also included in the payload for use with the next message in the communication:

$$c_i(M, ct_{MS}^1).\text{if } M \in l_M \text{ then}$$
$$\text{if } ct_{MS}^1 = \{\mathsf{m}_{\mathsf{MS}}^1(\nu N_{MS}, \nu IV_{MS}^1, \nu K_{MS})\}_{K_S} \text{ using } K_S^{-1}$$
$$\text{then } \nu N_{SM}.\nu IV_{MS}^2.\text{let } m_{MS}^2 = \mathsf{m}_{\mathsf{MS}}^2(N_{MS}, N_{SM}) \text{ in}$$
$$\overline{c_i}\langle\{\mathsf{w}(m_{MS}^2, IV_{MS}^2, \mathsf{mac}(\mathsf{akyx}(K_{MS}), (m_{MS}^2, IV_{MS}^2)))\}_{\mathsf{ekyx}(K_{MS})}^{IV_{MS}^1}\rangle$$

The mobile unit then receives, decrypts, and extracts the reply from the server. If the nonce previously sent by the mobile unit is returned correctly, it continues to create a new message containing a third nonce and this nonce hashed with the one generated by the server. This is then sent to the server:

$$c_i(ct_{MS}^2).\text{if } ct_{MS}^2 = \{\mathsf{w}(\nu m_{MS}^2, \nu IV_{MS}^2, \nu H_{MS}^2)\}_{\mathsf{ekyx}(K_{MS})}^{IV_{MS}^1}$$
$$\text{using } \mathsf{ekyx}(K_{MS}) \text{ and } \mathsf{m}_{\mathsf{MS}}^2.0(m_{MS}^2) = N_{MS} \text{ then}$$
$$\nu N'_{MS}.\nu IV_{MS}^3.\text{let } m_{MS}^3 = \mathsf{m}_{\mathsf{MS}}^3(\mathsf{sign}(K_M^{-1}, \mathsf{hash}(\mathsf{w.1}(m_{MS}^2), N'_{MS})), N'_{MS}) \text{ in}$$
$$\overline{c_i}\langle\{\mathsf{w}(m_{MS}^3, IV_{MS}^3, \mathsf{mac}(\mathsf{akxy}(K_{MS}), (m_{MS}^3, IV_{MS}^3)))\}_{\mathsf{ekxy}(K_{MS})}^{IV_{MS}^2}\rangle$$

## 5. Verification of the Protocol using ProVerif

ProVerif is a tool for verifying cryptographic protocols that supports verification of protocols using common cryptographic primitives including symmetric and asymmetric encryption. By creating a model of a protocol in the applied π-calculus, it is possible to write queries that ProVerif can answer to prove certain properties, e.g., whether or not a message sent between two trusted entities can be obtained by an attacker. The complete ProVerif model of the protocol can be downloaded at [17].

We will use ProVerif to verify these three security properties:

**Secrecy** An attacker is not able to read secret data.
**Strong secrecy** An attacker is not able to discern when secret data changes.
**Authenticity** An attacker is not able to impersonate a legitimate party of the system.

Our ProVerif model is directly based on the applied π-calculus description of Section 4. The remainder of this section shows examples of the use of ProVerif queries to verify the security of the protocol.

 ©2012 SBC — Soc. Bras. de Computação

$$
\begin{aligned}
\mathcal{P}_M(K_M^{-1}, K_S, c_i, c_b) \stackrel{\text{def}}{=} \ & \text{let } K_M = \mathsf{pk}(K_M^{-1}) \text{ in } \nu N_{MS}.\nu IV^1_{MS}.\nu K_{MS}. \\
& \overline{c_i}\langle K_M, \{\mathsf{m}^1_{\mathsf{MS}}(N_{MS}, IV^1_{MS}, K_{MS})\}_{K_S}\rangle.c_i(ct^2_{MS}). \\
& \text{if } ct^2_{MS} = \{\mathsf{w}(\nu m^2_{MS}, \nu IV^2_{MS}, \nu H^2_{MS})\}^{IV^1_{MS}}_{\mathsf{ekyx}(K_{MS})} \text{ using } \mathsf{ekyx}(K_{MS}) \\
& \qquad \text{and } \mathsf{m}^2_{\mathsf{MS}}.0(m^2_{MS}) = N_{MS} \text{ then} \\
& \nu N'_{MS}.\nu IV^3_{MS}.\text{let } m^3_{MS} = \mathsf{m}^3_{\mathsf{MS}}(\mathsf{sign}(K_M^{-1}, \mathsf{hash}(\mathsf{w}.1(m^2_{MS}), N'_{MS})), N'_{MS}) \text{ in} \\
& \overline{c_i}\langle\{\mathsf{w}(m^3_{MS}, IV^3_{MS}, \mathsf{mac}(\mathsf{akxy}(K_{MS}), (m^3_{MS}, IV^3_{MS})))\}^{IV^2_{MS}}_{\mathsf{ekxy}(K_{MS})}\rangle.c_i(ct^4_{MS}). \\
& \text{if } ct^4_{MS} = \{\mathsf{w}(\nu m^4_{MS}, \nu IV^4_{MS}, \nu H^4_{MS})\}^{IV^3_{MS}}_{\mathsf{ekyx}(K_{MS})} \text{ using } \mathsf{ekyx}(K_{MS}) \\
& \qquad \text{and } \mathsf{m}^4_{\mathsf{MS}}.0(m^4_{MS}) = N'_{MS} \text{ then} \\
& c_b(b).\nu IV^5_{MS}.\text{let } m^5_{MS} = \mathsf{m}^5_{\mathsf{MS}}(b) \text{ in} \\
& \overline{c_i}\langle\{\mathsf{w}(m^5_{MS}, IV^5_{MS}, \mathsf{mac}(\mathsf{akxy}(K_{MS}), (m^5_{MS}, IV^5_{MS})))\}^{IV^4_{MS}}_{\mathsf{ekxy}(K_{MS})}\rangle.c_i(ct^6_{MS}). \\
& \text{if } ct^6_{MS} = \{\mathsf{w}(\nu m^6_{MS}, \nu IV^6_{MS}, \nu H^6_{MS})\}^{IV^5_{MS}}_{\mathsf{ekyx}(K_{MS})} \text{ using } \mathsf{ekyx}(K_{MS}) \text{ then} \\
& \text{let } K_D = \mathsf{m}^6_{\mathsf{MS}}(m^6_{MS}) \text{ in} \\
& \nu N_{MD}.\nu IV^1_{MD}.\nu K_{MD}.\text{let } m^1_{MD} = \mathsf{m}^1_{\mathsf{MD}}(N_{MD}, IV^1_{MD}, K_{MD}) \text{ in} \\
& \overline{c_b}\langle\{m^1_{MD}\}_{K_D}\rangle.c_b(ct^2_{MD}). \\
& \text{if } ct^2_{MD} = \{\mathsf{w}(\nu m^2_{MD}, \nu IV^2_{MD}, \nu H^2_{MD})\}^{IV^1_{MD}}_{\mathsf{ekyx}(K_{MD})} \text{ using } \mathsf{ekyx}(K_{MD}) \\
& \qquad \text{and } \mathsf{m}^2_{\mathsf{MD}}.0(m^2_{MD}) = N_{MD} \text{ then} \\
& \nu IV^3_{MD}.\text{let } m^3_{MD} = \text{“unlock” in} \\
& \overline{c_b}\langle\{\mathsf{w}(m^3_{MD}, IV^3_{MD}, \mathsf{mac}(\mathsf{akxy}(K_{MD}), (m^3_{MD}, IV^3_{MD})))\}^{IV^2_{MD}}_{\mathsf{ekxy}(K_{MD})}\rangle
\end{aligned}
$$

**Figure 5. The process describing the mobile unit.**

### 5.1. Secrecy and Strong Secrecy

Certain variables have been marked as secret in the ProVerif model. This has been done to be able to subsequently test if an attacker is able to read encrypted data sent with the protocol. The following excerpt from the ProVerif model exemplifies this:

```
1 free access: bool [private].
2 query attacker(access).
3 noninterf access.
```

The variable `access` defined as secret (or private) in line 1 is sent from the server to the door unit and decides if the door unit will unlock the door. The query in line 2 determines whether secrecy is maintained, and the query in line 3 whether strong secrecy is maintained. Using the ProVerif tool we have verified that these properties indeed hold; this means that an attacker is not able to read whether someone was granted or denied access to a door, and not even whether someone who was previously granted access has now been denied access.

 ©2012 SBC — Soc. Bras. de Computação

$$
\begin{aligned}
\mathcal{P}_S(K_S^{-1}, c_i, c_m, l_M, f_D) \stackrel{\text{def}}{=}\ & c_i(M, ct^1_{MS}).\text{if } M \in l_M \text{ then} \\
& \text{if } ct^1_{MS} = \{\mathsf{m}^1_{\mathsf{MS}}(\nu N_{MS}, \nu IV^1_{MS}, \nu K_{MS})\}_{K_S} \text{ using } K_S^{-1} \text{ then} \\
& \nu N_{SM}.\nu IV^2_{MS}.\text{let } m^2_{MS} = \mathsf{m}^2_{\mathsf{MS}}(N_{MS}, N_{SM}) \text{ in} \\
& \overline{c_i}\langle\{\mathsf{w}(m^2_{MS}, IV^2_{MS}, \mathsf{mac}(\mathsf{akyx}(K_{MS}), (m^2_{MS}, IV^2_{MS})))\}^{IV^1_{MS}}_{\mathsf{ekyx}(K_{MS})}\rangle.c_i(ct^3_{MS}). \\
& \text{if } ct^3_{MS} = \{\mathsf{w}(\nu m^3_{MS}, \nu IV^3_{MS}, \nu H^3_{MS})\}^{IV^2_{MS}}_{\mathsf{ekxy}(K_{MS})} \text{ using } \mathsf{ekxy}(K_{MS}) \\
& \qquad \text{and } \mathsf{stripSign}(K_M, \mathsf{m}^3_{\mathsf{MS}}.0(m^3_{MS})) = \mathsf{hash}(N_{SM}, \mathsf{m}^3_{\mathsf{MS}}.1(m^3_{MS})) \text{ then} \\
& \nu IV^4_{MS}.\text{let } m^4_{MS} = \mathsf{m}^4_{\mathsf{MS}}(\mathsf{m}^3_{\mathsf{MS}}.1(m^3_{MS})) \text{ in} \\
& \overline{c_i}\langle\{\mathsf{w}(m^4_{MS}, IV^4_{MS}, \mathsf{mac}(\mathsf{akyx}(K_{MS}), (m^4_{MS}, IV^4_{MS})))\}^{IV^3_{MS}}_{\mathsf{ekyx}(K_{MS})}\rangle.c_i(ct^5_{MS}). \\
& \text{if } ct^5_{MS} = \{\mathsf{w}(\nu m^5_{MS}, \nu IV^5_{MS}, \nu H^5_{MS})\}^{IV^4_{MS}}_{\mathsf{ekxy}(K_{MS})} \text{ using } \mathsf{ekxy}(K_{MS}) \\
& \qquad \text{and } \mathsf{m}^5_{\mathsf{MS}}.0(m^5_{MS}) \in \text{dom}(f_D) \text{ then} \\
& \text{let } K_D = f_D(\mathsf{m}^5_{\mathsf{MS}}.0(m^5_{MS})) \text{ in } \nu IV^6_{MS}.\text{let } m^6_{MS} = \mathsf{m}^6_{\mathsf{MS}}(K_D) \text{ in} \\
& \overline{c_i}\langle\{\mathsf{w}(m^6_{MS}, IV^6_{MS}, \mathsf{mac}(\mathsf{akyx}(K_{MS}), (m^6_{MS}, IV^6_{MS})))\}^{IV^5_{MS}}_{\mathsf{ekyx}(K_{MS})}\rangle.c_m(D, ct^1_{DS}). \\
& \text{if } D \in \text{codom}(f_D) \text{ and } ct^1_{DS} = \{\nu N_{DS}\}_{K_S} \text{ using } K_S^{-1} \text{ then} \\
& \nu N_{SD}.\text{let } s^2_{DS} = \mathsf{sign}(K_S^{-1}, \mathsf{hash}(N_{DS}, N_{SD})) \text{ and } m^2_{DS} = \mathsf{m}^2_{\mathsf{DS}}(N_{SD}) \text{ in} \\
& \overline{c_m}\langle s^2_{DS}, \{m^2_{DS}\}_{K_D}\rangle.c_m(s^3_{DS}, ct^3_{DS}). \\
& \text{if } ct^3_{DS} = \{\nu N'_{DS}\}_{K_S} \text{ using } K_S^{-1} \\
& \qquad \text{and } \mathsf{stripSign}(K_D, s^3_{DS}) = \mathsf{hash}(N_{SD}, N'_{DS}) \text{ then} \\
& \nu access.\text{let } s^4_{DS} = \mathsf{sign}(K_S^{-1}, \mathsf{hash}(access, N_{SD})) \\
& \qquad \text{and } m^4_{DS} = \mathsf{m}^4_{\mathsf{DS}}(access, N_{SD}) \text{ in} \\
& \overline{c_m}\langle s^4_{DS}, \{m^4_{DS}\}_{K_D}\rangle
\end{aligned}
$$

**Figure 6. The process describing the server.**

## 5.2. Authenticity

Listing 1 shows the correspondence query used to determine if a message received by the door unit about the user's access rights has been sent by the server.

```
query K_D: pubKey, challenge: bool; inj-event(doorUnlockMessageReceived(access)
    ) ==> inj-event(serverSendingFinalUnlock(access)).
```

**Listing 1. Proving authenticity of messages from the server.**

Correspondence queries prove authenticity using two events: begin and end events. Authenticity is proved if the end event cannot be reached without the begin event being triggered [18]. In Listing 1, these two events are: `serverSendingFinalUnlock` which is the begin event and is triggered when the server is about to send the `access` variable to the door, and the end event `doorUnlockMessageReceived` is triggered when the door has received a message at the time it expected to receive the `access` variable. The query thus verifies that

 ©2012 SBC — Soc. Bras. de Computação

$$
\begin{aligned}
\mathcal{P}_D(K_D^{-1}, K_S, c_b, c_m) \stackrel{\text{def}}{=} \ & \text{let } K_D = \mathsf{pk}(K_D^{-1}) \text{ in } \nu b. \\
& \overline{c_b}\langle b\rangle.c_b(ct^1_{MD}). \\
& \text{if } ct^1_{MD} = \{\mathsf{m}^1_{\mathsf{MD}}(\nu N_{MD}, \nu IV^1_{MD}, \nu K_{MD})\}_{K_D} \text{ using } K_D^{-1} \text{ then} \\
& \nu IV^2_{MD}.\text{let } m^2_{MD} = N_{MD} \text{ in} \\
& \overline{c_b}\langle\{\mathsf{w}(m^2_{MD}, IV^2_{MD}, \mathsf{mac}(\mathsf{akyx}(K_{MD}), (m^2_{MD}, IV^2_{MD})))\}^{IV^1_{MD}}_{\mathsf{ekyx}(K_{MD})}\rangle.c_b(ct^3_{MD}). \\
& \text{if } ct^3_{MD} = \{\mathsf{w}(\nu m^3_{MD}, \nu IV^3_{MD}, \nu H^3_{MD})\}^{IV^2_{MD}}_{\mathsf{ekxy}(K_{MD})} \text{ using } \mathsf{ekxy}(K_{MD}) \\
& \qquad \text{and } m^3_{MD} = \text{“unlock” then} \\
& \nu N_{DS}.\text{let } m^1_{DS} = \mathsf{m}^1_{\mathsf{DS}}(N_{DS}) \text{ in} \\
& \overline{c_m}\langle K_D, \{m^1_{DS}\}_{K_S}\rangle.c_m(s^2_{DS}, ct^2_{DS}). \\
& \text{if } ct^2_{DS} = \{\nu N_{SD}\}_{K_D} \text{ using } K_D^{-1} \\
& \qquad \text{and } \mathsf{stripSign}(K_S, s^2_{DS}) = \mathsf{hash}(N_{DS}, N_{SD}) \text{ then} \\
& \nu N'_{DS}.\text{let } s^3_{DS} = \mathsf{sign}(K_D^{-1}, \mathsf{hash}(N_{SD}, N'_{DS})) \text{ and } m^3_{DS} = \mathsf{m}^3_{\mathsf{DS}}(N'_{DS}) \text{ in} \\
& \overline{c_m}\langle s^3_{DS}, \{m^3_{DS}\}_{K_S}\rangle.c_m(s^4_{DS}, ct^4_{DS}). \\
& \text{if } ct^4_{DS} = \{\nu access, N_{SD}\}_{K_D} \text{ using } K_D^{-1} \\
& \qquad \text{and } \mathsf{stripSign}(K_S, s^4_{DS}) = \mathsf{hash}(access, N_{SD}) \text{ then} \\
& \nu c_a.\overline{c_a}\langle access\rangle
\end{aligned}
$$

**Figure 7. The process describing the door unit. The private channel $c_a$ models the door unit controlling the door lock hardware.**

a man-in-the-middle attack is not possible. The fact that the query is injective (`inj-event`) means that the door will never receive more messages from the server than it actually sent, thus preventing replay attacks.

### 5.3. Discovering Security Vulnerabilities

During the development of the protocol, ProVerif was used to detect vulnerabilities. One example of a vulnerability discovered and fixed during the development was found in proving authenticity of the server on the door access control unit with the query in Listing 1. The following lines are used to prove the authenticity of the message received on the door from the server:

```
1 in(mobile, (s4_DS: signature, ct4_DS: asymEncrypted));
2 let message = privDecrypt(Kpriv_D, ct4_DS) in
3   if verifySign(K_S, hash(message), s4_DS) = true then
4     let (access_received: bool, N_SD_returned: bitstring) = message in
5       if N_SD_returned = N_SD then
6         event doorUnlockMessageReceived(access_received);
```

In line 1, the message is received from the server via the mobile unit. It contains two parts: `s4_DS`, the signature of the message, and `ct4_DS`, the asymmetrically encrypted message. In line 2, the message is decrypted using the private key used by the door,

 ©2012 SBC — Soc. Bras. de Computação

```
event serverSendingMobileChallenge(K_M, N_MS);
out(internet, symEncryptWithMAC(ekyx(K_MS), IV1_MS, IV2_MS, akyx(K_MS), (N_MS,
     N_SM)));

in(internet, ct3_MS: symEncrypted);
let (Hc3_MS: mac, H3: mac, IV3_MS: iv, (returnedChallengeSignature: signature,
     Nprime_MS_received: bitstring)) = symDecryptWithMAC(ekxy(K_MS), IV2_MS,
     akxy(K_MS), ct3_MS) in
if Hc3_MS = H3 then
if verifySign(K_M, hash((N_SM, Nprime_MS_received)), returnedChallengeSignature
     ) = true then
event serverAcceptsMobile(K_M, K_MS);

(* Send all ok to mobile *)
out(internet, symEncryptWithMAC(ekyx(K_MS), IV3_MS, IV4_MS, akyx(K_MS),
     Nprime_MS_received));
```

**Listing 2. Excerpt of ProVerif model of the server.**

```
N_SM = Rand.rand_bytes(32)
message = N_MS + N_SM
self.aes_send(message)

#Verifying returned challenge from phone
dec = self.aes_recv()
if len(dec) != 256 + 32:
  raise Error('Message does not have correct length, aborting')
sig, Nprime_MS_received = dec[:256], dec[256:]
if not phone_rsa.verify(sha256(N_SM + Nprime_MS_received).digest(), sig, '
     sha256'):
  raise Error('Unable to verify signature of server challenge, aborting')
#Challenge passed, connection ready for use
self.aes_send(Nprime_MS_received)
```

**Listing 3. Excerpt of Python implementation of the server.**

`Kpriv_D`, and in line 3, the signature of the message is verified using the public key of the server `K_S`. If the signature matches, the message components are extracted in line 4: the `access_received` boolean value that tells the door whether or not the user has been granted access to the door by the server, and `N_SD_returned` that is a nonce included to prevent replay attacks by the mobile unit. In line 5, the nonce is compared with the expected value, and only if it matches is the event `doorUnlockMessageReceived` triggered in line 6.

In an earlier version of the protocol, the check in line 5 was not present. This caused the protocol to be vulnerable to replay attacks allowing the user access to doors where his access had been revoked. However, by verifying the query in Listing 1, ProVerif was able to find this mistake allowing for its correction.

## 6. Implementation of the Protocol

Based on the ProVerif model, a working prototype implementation was created to test the viability for use in an actual end-user product. The server and door units were programmed in Python and the mobile unit was an Android smartphone with a custom application written in Java and C. The level of detail in the protocol specification allowed for an implementation that has a close correspondence with the ProVerif model.

Listing 2 shows a code excerpt from the ProVerif model from the part of the model where the server entity communicates with the mobile device in order to verify the mobile

 ©2012 SBC — Soc. Bras. de Computação

device authenticity. Listing 3 shows the corresponding part of the server implementation in Python, where AES is used for symmetric encryption and RSA is used for public key encryption. In the Python implementation the IVs and keys are hidden within the `aes_send` and `aes_recv` methods, but are otherwise the same as sending and receiving messages with the `symEncryptWithMAC` and `symDecryptWithMAC` functions through the Internet channel in the ProVerif model. Since the type system in ProVerif is different from that of Python, the messages in Python are merely strings, and therefore the length of each message is also verified.

## 7. Conclusion

In this paper, a protocol was presented that allows secure communication over insecure Bluetooth and Internet connections between door locks and a central server. By using only the Internet connection of a smartphone, the cost of deploying an electronic door lock solution can be minimized, especially for large buildings and institutions. Use cases include access control at companies, universities, and for home care.

The protocol is designed with implementation in mind and for real-world use. It is explained as a protocol narration and formalized using applied $\pi$-calculus. The detailed formalization of the protocol allowed us to verify a set of properties using the cryptographic protocol verifier ProVerif. The protocol allows companies such as MVC-Data to improve security for their door lock units, by using the formalization to implement the protocol in a consumer product. By integrating the protocol in their current product range MVC-Data will also be able to address what should happen in (the rare) case where a connection via 3G or Wi-Fi is flaky. In that case the protocol could default to an improved variant of their current protocol where (a limited number of) authorized users credentials are cached in the door unit. The caching of user credentials could even piggy-bag on communication in the protocol proposed in this paper when 3G or Wi-Fi is working correctly. Future work clearly includes a thorough analysis of such a combined protocol.

The current implementations of the protocol assume that both Bluetooth and 3G/Wifi is constantly turned on. To save energy on the mobile device the application could ask the user to turn the relevant radios on and off. However, that would defeat some of the purpose of the application as it currently does not require interaction from the user. The user can keep the mobile device in a pocket and as soon as the user approaches a door, for which the user has authorized access, the system will unlock the door. To preserve this property and save energy, it would make sense to extend the protocol to use location information to make decision on when to turn on and off the relevant radios, i.e. only when the user approaches a door will the relevant radios be turned on. It is a case for future studies to investigate the feasibility and energy efficiency of this extension.

## References

[1] MVC-Data ApS. (2012, January 11th) Wireless Bluetooth access control. [Online]. Available: http://www.mvc-data.com/Home.html

[2] BlueLon. (2011, May 24th) BlueAccess BAL-100-BL. [Online]. Available: http://www.bluelon.com/index.php?id=248

 ©2012 SBC — Soc. Bras. de Computação

[3] FlexiPanel. (2007, March 1st) BlueLock - Access control triggered by Bluetooth on your mobile phone. [Online]. Available: http://www.flexipanel.com/Docs/BlueLock%20DS377%20Cover.pdf

[4] Steab AB. (2012, January 11th) Blue Step. [Online]. Available: http://steab.se/2bluestep.html

[5] ECKey. (2011, May 24th) ECKey - turn your phone into a key! [Online]. Available: http://www.eckey.com

[6] SOREX - Wireless Solutions GmbH. (2012, January 11th) SOREX wireless products. [Online]. Available: http://www.sorex-austria.com/overview_wireless.html

[7] E. R. Wognsen, H. S. Karlsen, M. Calverley, and M. N. Follin. (2011, May 27th) A proposal for a secure relay protocol for door access control. Student report at Aalborg University. [Online]. Available: http://sw8.lmz.dk/report.pdf

[8] M. Abadi, "Private authentication," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, ser. PET'02. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 27–40.

[9] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols," *Inf. Comput.*, vol. 148, pp. 1–70, January 1999.

[10] B. Blanchet. (2011, December 11th) ProVerif. [Online]. Available: http://www.proverif.ens.fr

[11] S. Kremer and M. Ryan, "Analysis of an electronic voting protocol in the applied pi calculus," in *In Proc. 14th European Symposium On Programming (ESOP'05), volume 3444 of LNCS*. Springer, 2005, pp. 186–200.

[12] K. Bhargavan, C. Fournet, R. Corin, and E. Zalinescu, "Cryptographically verified implementations for TLS," in *Proceedings of the 15th ACM conference on Computer and communications security*, ser. CCS '08. New York, NY, USA: ACM, 2008, pp. 459–468. [Online]. Available: http://doi.acm.org/10.1145/1455770.1455828

[13] R. Chang and V. Shmatikov, "Formal analysis of authentication in bluetooth device pairing," 2007.

[14] B. Blanchet. (2011, December 11th) ProVerif users. [Online]. Available: http://www.proverif.ens.fr/proverif-users.html

[15] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, ser. POPL '01. New York, NY, USA: ACM, 2001, pp. 104–115. [Online]. Available: http://doi.acm.org/10.1145/360204.360213

[16] R. Milner, *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.

[17] E. R. Wognsen, H. S. Karlsen, M. Calverley, and M. N. Follin. (2012, Feb) Protocol proverif model. [Online]. Available: http://sw8.lmz.dk/protocol.pv

[18] Woo and S. Lam, "A semantic Model for Authentication Protocols," in *The 1993 Symposium on Research in Security and Privacy*. IEEE Computer Society Press, May 1993, pp. 178–194.

 ©2012 SBC — Soc. Bras. de Computação

# Modified Current Mask Generation (M-CMG): An Improved Countermeasure Against Differential Power Analysis Attacks

**Hedi Besrour[2], Daniel Gomes Mesquita[1], Machhout Mohsen[2], Tourki Rached[2]**

[1]Faculdade de Informática – Universidade Federal de Uberlândia – Brazil

[2]Electronics and Microelectronics Laboratory – Monastir University – Tunisia

mesquita@facom.ufu.br

{hedi.besrour, mohsen.machhout, rached.tourki}@fsm.rnu.tn

***Abstract.*** *It has been demonstrated that encryption device leaks some information, which can be exploited by various attacks such as differential power analysis (DPA). To protect an Advanced Encryption Standard (AES) implementation from DPA without any modification of the cryptographic algorithm, we can use the Current Masking Generation (CMG). The CMG countermeasure consists of stabilizing the power consumption, but it presents some limitations concerning temperature variations and the Early effect. The goal of this paper is to update the CMG to address these problems, evolving to the Modified-Current Masking Generation (M-CMG).*

***Resumo.*** *Durante seu funcionamento, dispositivos criptográficos vazam informações, que podem ser exploradas por vários ataques, tal qual a análise diferencial de consumo de potência (DPA). Para proteger o algoritmo de criptografia AES de ataques desse tipo, sem que seja necessário modificar o algoritmo em si, pode-se usar uma contramedida chamada Geração de Máscara de Corrente (CMG). Essa contramedida consiste na estabilização do consumo do circuito criptográfico, dificultando assim a exploração dessa informação. Entretanto, essa contramedida apresenta limitações em termos de estabilidade, especialmente relacionada ao efeito de Early e à variação de temperatura do circuito. O objetivo deste artigo é demonstrar esses problemas e também propor a solução para ambos, criando assim a contramedida M-CMG: Modified Current Mask Generation.*

## 1. Introduction

In today's ultra connected society, virtually every facet of life involves the flow of information through computer networks. For instance, in online banking transactions or diplomatic communications, the need of a secure flow of information is vital. To provide such security, encryption algorithms are employed, as the Advanced Encryption Standard (AES) [Daemen and Rijmen 2001]. This algorithm is a symmetric block cipher that can process data in 16-byte blocks, using cipher keys with lengths of 128, 192, and 256 bits. The AES won a contest promoted by the US government through its National Institute of Standards and Technology (NIST), but rapidly became an *de facto* standard for commercial transactions and online communications around the world [Parikh and Patel 2007].

 ©2012 SBC — Soc. Bras. de Computação

Generally, the security of cryptographic algorithms is their mathematical basis. The AES basis are operations over the finite field $GF(2^8)$, which means its security can be easily analyzed using mathematical techniques. The AES cryptanalysis have been studied since the algorithm original proposal, and even if the complexity of these kind of attacks is slightly reduced regarding brute force [Bogdanov et al. 2011], the algorithm is still considered secure.

However, a new type of attacks, no longer aiming the mathematical construction of the algorithm, but the physical characteristics of its implementation has been proposed. The information leaked by the physical implementation, also known side channel information, can be exploited to perform attacks [Kocher et al. 1999]. The so called Side Channel Attacks (SCA) have become an increasingly important design concern for cryptographic implementations [Mangard et al. 2007]. A variation of SCA consists on observing the power consumption of an AES implementation, which can lend information about the operations, operands, and ultimately the secret keys. With a modest investment in a digital oscilloscope and third party software, attackers can execute a Differential Power Analysis (DPA) [Kocher et al. 1999] with little knowledge of the system under attack.

This threat drives to search for secure cryptographic hardware implementations, resistant to SCA. The countermeasures against power analysis proposed to counteract SCA can be algorithmic or physical (gate-level) [Mangard et al. 2007]. On one hand, algorithmic countermeasures have as drawback the need of modifications on the original cryptographic algorithms, attempting against its wide applicability and compatibility with already deployed applications. On the other hand, physical countermeasures may generate an area overhead, but can be more effective in maintaining the algorithmic standards, as the Current Mask Generation (CMG) previously proposed by one of the authors of the current paper, in [Mesquita et al. 2007]. As the DPA attack exploits the variation on current consumption concerning the data computed, the CMG aims to minimize the impact of the AES data processing to the chip power consumption.

Nevertheless, by reviewing the CMG, we found some stability issues related to the Early effect [Early 1952] and concerning the circuit behavior in the presence of temperature variations. This second problem may lead to a weakness of the countermeasure, once the current consumption of the CMG increases as the heat rises. These observations leads to some improvements of the original idea, introduced here as the M-CMG, the Modified Current Mask Generation.

This article is organized as follows. Section 2 highlights the Differential Power Analysis attack principles and a real-world application against an AES implementation. The research method is discussed in Section 3. Section 4 explains the advantages and the problems founded at the CMG countermeasure. Section 5 introduces solutions to counteract the temperature and the early effects on the CMG, demonstrated in a theoretical basis. Finally, Section 6 presents the conclusions and further work within this theme.

## 2. Differential Power Analysis (DPA) Attack

In the groundbreaking paper [Kocher et al. 1999], it was presented a method to recover a secret key of real world cryptographic systems from power consumption of computers and microchips. The Differential Power Analysis (DPA) is used to attack on either the

 ©2012 SBC — Soc. Bras. de Computação

first or the last round but it can sometimes be applied to attack on intern rounds of the block ciphers [Messerges et al. 1999]. The main target of a DPA attack is to recover a byte *k* of the secret key *K* that is used to encrypt a plaintext or decrypt ciphertext through a cryptographic chip. In this paper, we use the DPA for the first round of an AES-128 encryption implementation.

### 2.1. DPA Principles

The DPA attack is based to send $N$ known random plaintexts and measures the corresponding chip power consumption waveforms during their encryption. Therefore, $N$power traces $T_{1\cdots M}[1\cdots M]$ are collected by the adversary, each of which consists of $M$ samples, as shown in Figure 1.

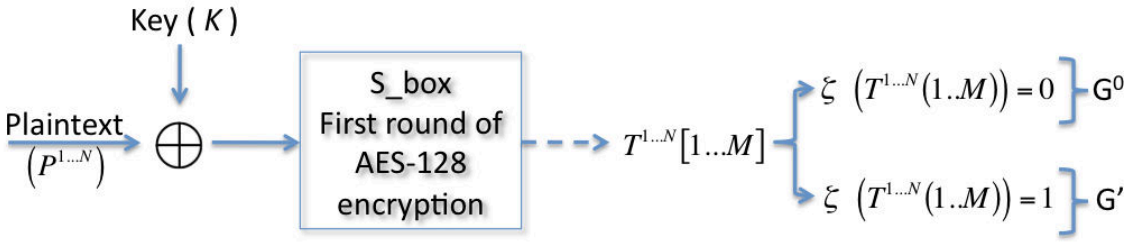



**Figura 1. Basis of the Differential Power Analysis Attack**

An attacker may create several hypotheses of key k and use the selection function $\xi$ to separate the power traces in some $G$ groups[Guilley 2007]. After, it uses statistics to detect the correct key $K$. for this reason, the power traces are immediately classified in two groups, $G_0 = \{T_n|\xi(\cdots) = 0\}$ and $G_1 = \{T_n|\xi(\cdots) = 1\}$.

These two groups $G_0$ and $G_1$ are averaged in order to eliminate random noise into $D_0$ and $D_1$, as represented in Equations 1 and 2, respectively, where $|G_0| + |G_1| = N$.

$$D_0(m) = \left|\frac{1}{G_0}\right| \sum_{G_n(m)\in G_0}^{n,m} G_n(m) \tag{1}$$

$$D_1(m) = \left|\frac{1}{G_1}\right| \sum_{G_n(m)\in G_0}^{n,m} G_n(m) \tag{2}$$

The next step is to subtract the two groups $D_0$ and $D_1$ from each other. This result leaves a differential trace $DPA(m) = D_1(m) - D_0(m)$.

If the key guess was incorrect the resultant trace will tend to zero, but if the key guess was correct the differential trace will contain spikes at points of correlation, verifying the correct key guess. The spike under the correct guess is distinguished only if it is sufficiently greater than the maximum spike under a wrong one.

### 2.2. DPA data acquisition and experimental results

The best evaluation method of DPA attack research is direct measurement. We have developed PCB board with a PIC18F microcontroller, and we have implemented the AES-128

 ©2012 SBC — Soc. Bras. de Computação

in software. We have been using MATLAB to send the messages via RS-232 to the PCB board. At the beginning of the encryption cycle, the output of PIC18F microcontroller triggers a Lecroy WaveRunner104MXI oscilloscope to start acquisition data for power trios. The setup is show in Figure 2.

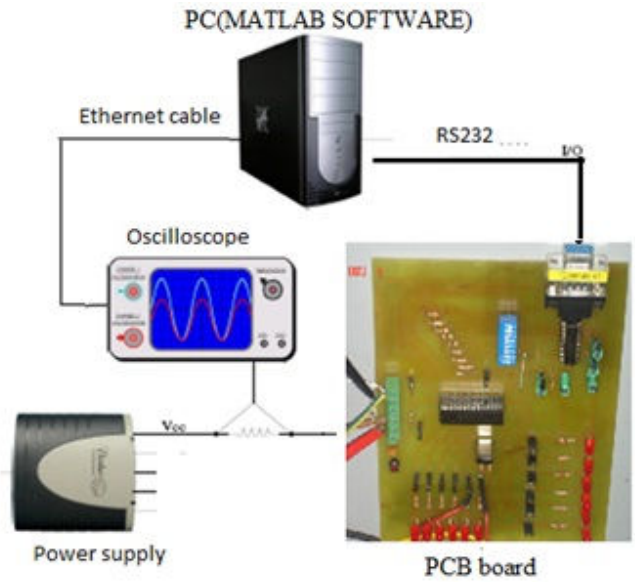



**Figura 2. Experimental setup for data acquisition**

We made an acquisition for 5000 power traces and they are imported into PC from oscilloscope via Ethernet for statistical analysis with our custom software (runned in MATLAB). We had the following results for DPA attacks in Figure 3. In the same figure, we can observe some ghost peaks that may mislead the key guessing. To minimize these ghost peaks, it is necessary to increase the Signal to Noise Ratio (SNR), as suggested in [Alioto et al. 2006]. The SNR is given by Equation 3, where $\varepsilon = l \times \beta$ and $\beta$ is the number of output bits by the selection function $\xi$. The parameter $l$ is constant equal to the voltage difference seen between two data words concerning Hamming weights $N(n)$ and $N(n+1)$.

$$SNR = \frac{\sqrt{N\varepsilon}}{\sqrt{8\sigma^2 + \varepsilon^2(ad + d - 1)}} \tag{3}$$

One way to increase the SNR in Equation 3 is to increase $\varepsilon$. The value of $\varepsilon$ depends on the number of bits output by selection function $\xi$. Then, the power traces are immediately classified in three sets: $G_0 = \{T_n|\xi(\cdots) = 0'\}$, $G_1 = \{T_n|\xi(\cdots) = 1'\}$ and $G_2 = \{T_n| \ni G_0, G_1\}$.

When using the last selection function $\xi$ for DPA attack, the attacker needs more acquisition of power signals. Because all of $1 - 2^{1-l}$ power traces placed in partition $G_2$ are not used. For this reason, an attacker cannot increase $l$ too much without requiring an excessive number of power traces.

The result of this improvement can be seen at Figure 4. The sub key from $DPA(l = 3)$ using $1000$ power samples. We can observe the peak corresponding the

 ©2012 SBC — Soc. Bras. de Computação

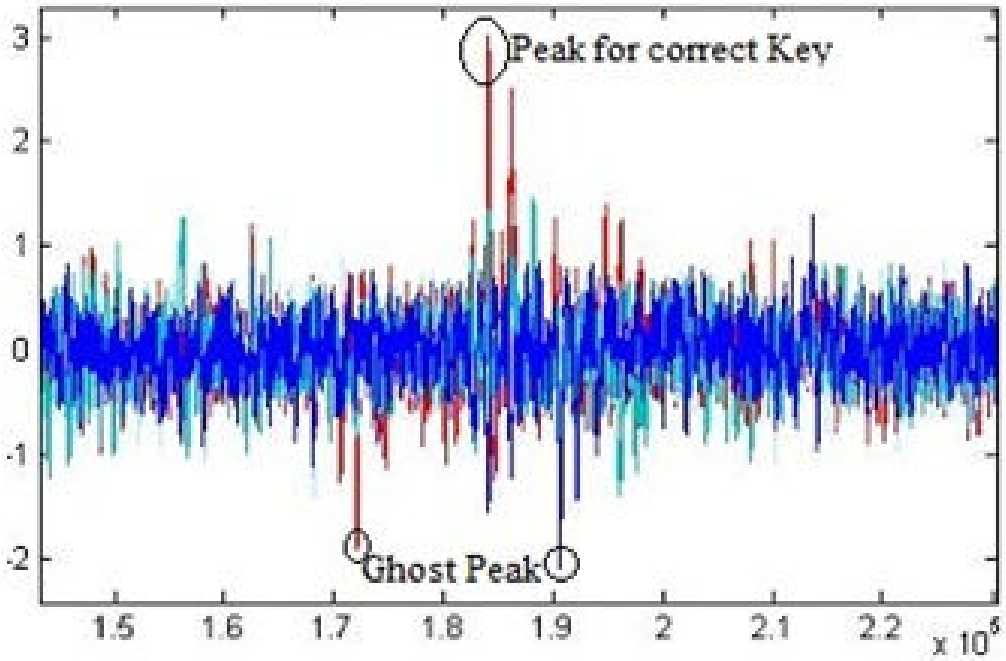



**Figura 3. DPA Attack Experimental Results**

correct key guessing, even if some ghost peaks are still present.

## 3. Method

This work was mainly motivated by the two following research questions:

- *Is the CMG stable regarding the* Early effect *[Early 1952]?*
- *How the CMG countermeasure behaves when the temperature rises?*

To address these two questions, firstly, the CMG architecture was replicated. The initial goal is to perform an electrical simulation to evaluate the CMG's normal behaviour. At this point we could verify the same results as presented in [Mesquita et al. 2007]. The variable measured here is the consumed current of the cryptographic circuit and the current masked by the CMG.

Concerning the *Early effect*, a theoretic study, shown in Section 5.1, was performed to verify the existence of the problem, as well to suggest a practical solution. This study is based on both James Early original paper [Early 1952] and an update to simulate the Early effect given by [McAndrew and Nagel 1996]. The variable we measure here is the external current regarding the input current, as detailed in the Section 4.

After, to learn about its behavior regarding temperature, the circuit was simulated with the Cadence P-Spice tool, varying the temperatures from $10^0$ $Celsius$ to $150^0$ $Celsius$. The variable measured here is the output (masked) current concerning the heat variation. This current must remain stable with the temperature increasing.

Once the problems were plotted, through a theoretical study, modifications on the original CMG were calculated, then simulated, resulting in a updated an upgraded countermeasure.

 ©2012 SBC — Soc. Bras. de Computação

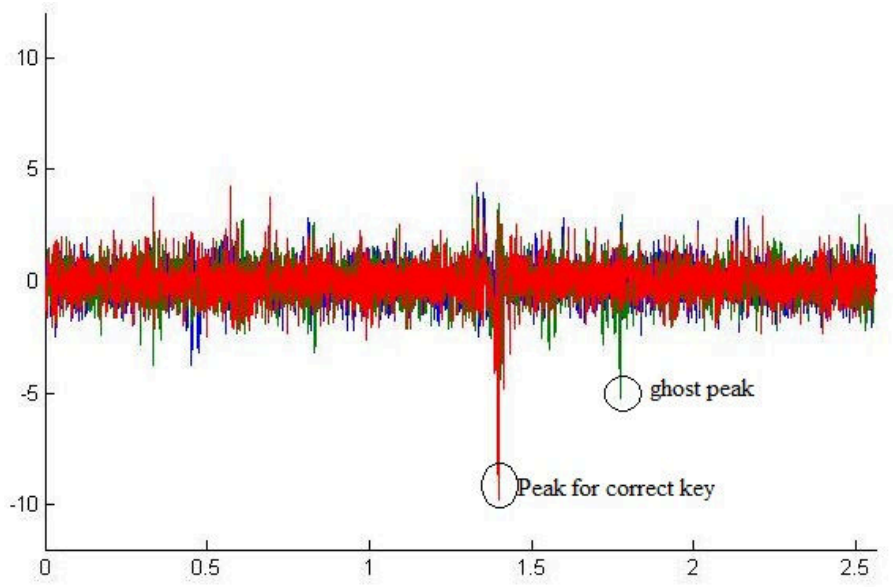



**Figura 4. Improved DPA Attack Experimental Results**

## 4. Current Mask Generation: Principles, advantages and weakness

Several countermeasures have been proposed to against DPA attack, as can be viewed at Chapter 7 from [Mangard et al. 2007]. One efficient way to counteract DPA is to lend the power consumption constant, as proposed in [Mesquita et al. 2005]. With this approach, no changes are needed neither on the cryptographic algorithm nor on the cryptographic core.

As seen in Figure 5, The CMG is composed basically of a current mirror and a follower circuit. The current mirror imposes the fixed current $I_2$ equal at the peak of current consumption for cryptographic circuit.

The operational amplifier receives a voltage sent by the mirror and compares it with a reference tension. If the cryptographic circuit consumes an amount of current less than $I_2$, the tension at the $Op\ Amp$ input will be lower than the reference tension. Then the $Op\ Amp$ output will send $0V$ in the base of the $P_4$ transistor. So, it will consume an $I_L$ current ($I_L = I_{ext} - I_2$).

It is important to take a look at the current mirror from Figure 5, so we expanded it to Figure 6. In an ideal situation $\beta \rightarrow \infty$. It means that while $V_{BE_{N_1}}$ is fixed, $I_0$ grows with $V_{CE_{N_1}}$. Also, the $Early\ Effect$ has been neglected. This ca be written as in the Equation 4.

$$I_0 = \frac{I_{ext}}{1 + \frac{2}{\beta}} \tag{4}$$

But, in reality, if the transistors have the same parameters $\beta_0$, the transistors $N_0$ and $N_1$ needs current as given by Equation 5 (with $V_a$ designating the early voltage), and the relation given by Equation 6.

 ©2012 SBC — Soc. Bras. de Computação

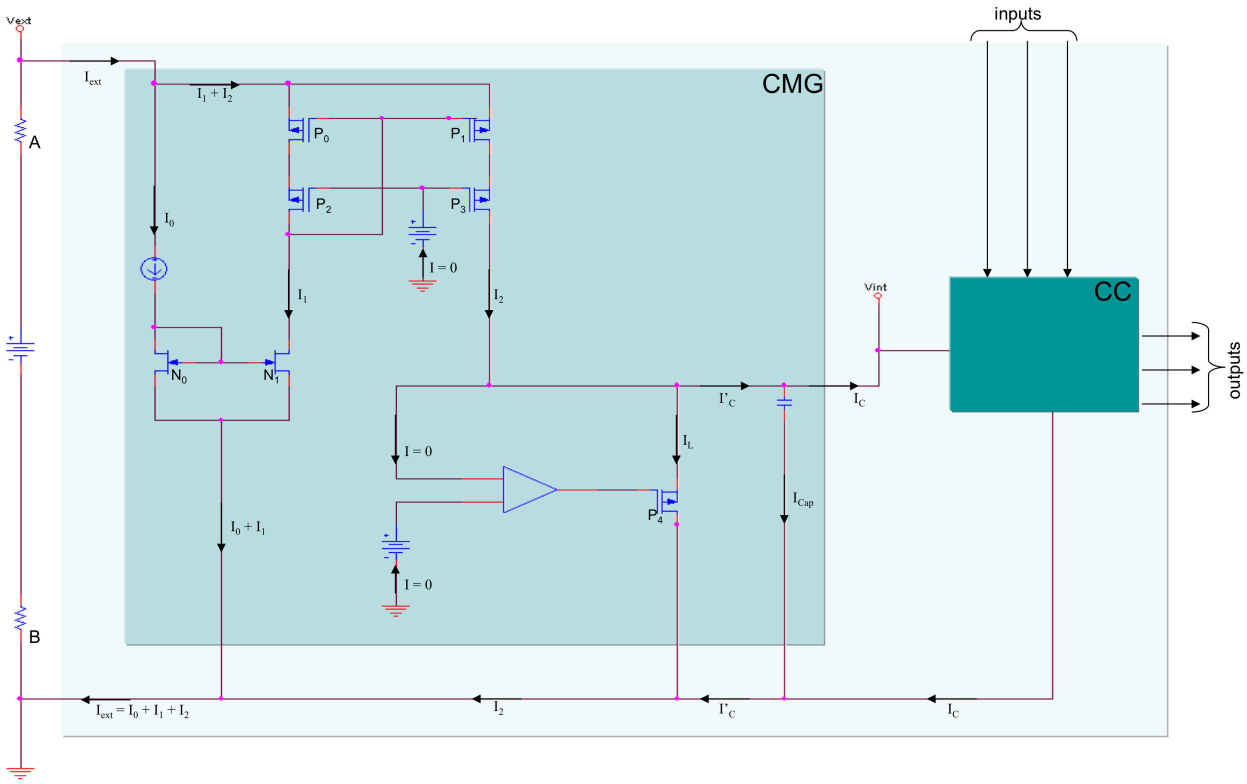



**Figura 5. Original CMG: Current Mask Generation**

$$I_{b_{N_0}} = \frac{I_{C_{N_0}}}{\beta_0} \; and \; I_{b_{N_1}} = \frac{I_0}{\beta_0(1 + \frac{V_{CB_{N_1}}}{V_A}} \tag{5}$$

$$I_0 = \frac{1 + \frac{V_{CB_{N_1}}}{V_A}}{1 + \frac{2}{\beta_0}} I_{ext} \tag{6}$$

The CMG have been demonstrated efficient to counteract DPA in ideal situations [Mesquita et al. 2007], but the Equation 6 shows that system has stability problems, since the output current $I_0$ may be grater than the input current $I_{ext}$ due to the *Early Effect*.

Another problem founded may be critical, because may be exploited by an attacker: the current mirror proposed in CMG (Figure 6) may misbehave if the transistors temperature changes [Munro 1991].

In the next Section we propose a single modification that copes with the two presented problems with the original CMG.

## 5. Modified Current Mask Generation: The M-CMG

To solve the observed CMG problems, we to need introduce a "current-keeping"with negative feedback in the same mirror current used (Figure 6. For this reason, we have added new transistor $N_2$, seen in Figure 7.

 ©2012 SBC — Soc. Bras. de Computação

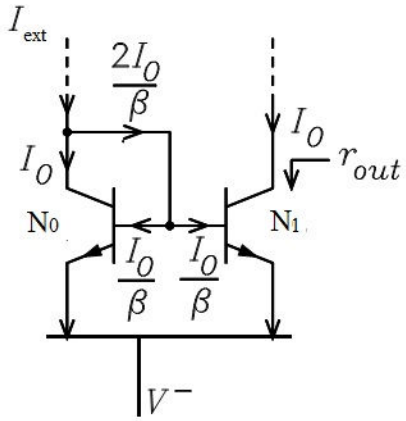



**Figura 6. Current Mirror from the Original CMG**

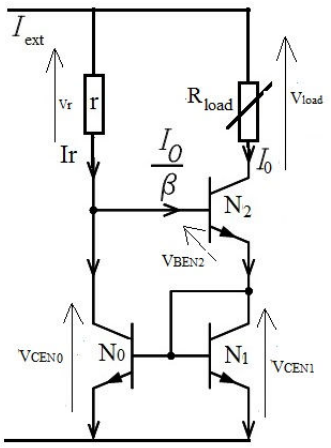



**Figura 7. Proposed Current Mirror for the M-CMG**

Following, we analyze this modification concerning the *Early Effect* and the temperature issues.

### 5.1. The M-CMG and the Early Effect

Regarding Figure 7, the transistor $N_2$ controls the current, through $R_{load}$, by changing its resistance between the collector and the emitter. For instance, $R_{load}$ increase causes a decrease in the current $I_0$ and the voltage $V_{load}$. Or the current $I_0$ is the input for the simple current mirror $N_0$ and $N_1$; thus, the output current will drop and $V_{CE_{N_1}}$ increases. The input voltage $V_{BE_{N_2}}$ of the transistor $N_2$ increases; it begins opening more until the load current restores its previous magnitude. Doing that, it keeps the current $I_r$ and $I_0$ constant. And like that, we reached our objective and solved the problem of early effect. So, the current $I_0$ can be re-written as in Equation 7.

$$I_0 = \frac{\beta^2 + 2\beta}{\beta^2 + 2\beta + 2} I_{ext} \tag{7}$$

Thus, Equation 7 predicts that $I_0 \approx I_{ext}$.

 ©2012 SBC — Soc. Bras. de Computação

### 5.2. The M-CMG and the Temperature Issue

Once the *Early Effect* is avoided, it is important to study the stability of the M-CMG (Figure 7) concerning the influence of heat on the countermeasure behavior. The relation between $I_0$ and temperature is given by Equation 8.

$$\Delta I_0 = S_V \Delta V_{BE} + S_\beta \Delta\beta \tag{8}$$

With:

$$S_V = \left[\frac{\delta I_0}{\delta V_{BR}}\right]_{\beta_{constant}} \tag{9}$$

And:

$$S_\beta = \left[\frac{\delta I_0}{\delta \beta}\right]_{V_{BEconstant}} \tag{10}$$

So we use the relation $I_{ext} = \frac{V_{CC}-2V_{BE}}{r}$ and replace $I_0$ from Equation 7 in Equations 9 and 10, obtaining Equation 11.

$$\Delta I_0 = -\left(\frac{2}{r}\Delta V_{BE} + \frac{4(V_{CC}-V_{BE})}{r\beta^3}\Delta\beta\right) \tag{11}$$

In order to confirm the Temperature Equation 11 stability with heat increase, we simulate de M-CMG circuit with the P-Spice tool. The Figure 8 shows the variation of current consumption in CMG and M-CMG depending on the temperature.

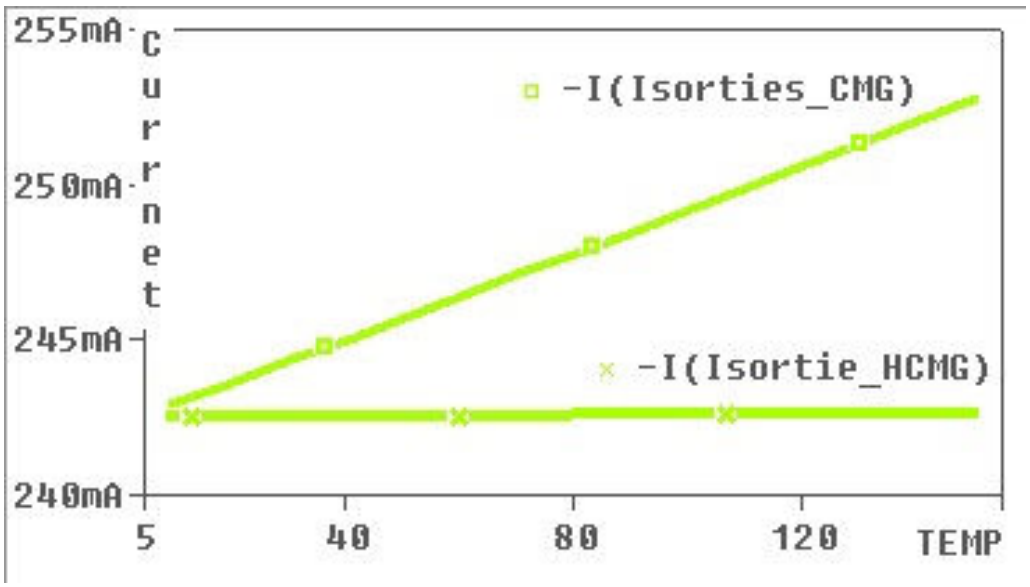



**Figura 8. Current consumption behavior of CMG and M-CMG related to temperature increase**

In Figure 8, the current variation from the CMG countermeasure varies $9mA$ when the temperature increases from $20^o$ to $150^o$, while the current from M-CMG remains stable.

 ©2012 SBC — Soc. Bras. de Computação

## 6. Conclusion and Further Work

In the theoretical terms and in simulation, the M-CMG shows that it can cope with the temperature variations and Early effect more than CMG. The latter characteristics allow the M-CMG to stabilize the power consumption in such a manner that DPA is even more difficult.

Given that the theoretical study of CMG was very rewarding, we are looking forward to implement a single chip with the AES with the M-CMG countermeasure and then to attack it with the DPA, in order to show in practice the efficiency of our countermeasure.

## Acknowledgements

The authors are thankful to Dr Benedikt Gierlichs, Dr Josep Balasch and also to Prof. Bart Preneel, from ESAT/SCD-COSIC, Leuven Belgium, for having given us the opportunity to test the DPA and CPA attacks in their research laboratory.

The presentation of this work was founded by the FAPEMIG (Fundação de Apoio à Pesquisa de Minas Gerais), under the project PEP-00760-12.

A apresentação deste trabalho foi financiada pela FAPEMIG (Fundação de Apoio à Pesquisa de Minas Gerais), através do projeto PEP-00760-12.

## Referências

Alioto, M., Poli, M., Rocchi, S., and Vignoli, V. (2006). Power modeling of precharged address bus and application to multi-bit dpa attacks to des algorithm. In *Proceedings of the 16th Power and Timing Modeling, Optimization and Simulation*, PATMOS'06, pages 593–602, Berlin, Heidelberg. Springer-Verlag.

Bogdanov, A., Khovratovich, D., and Rechberger, C. (2011). Biclique cryptanalysis of the full aes. Cryptology ePrint Archive, Report 2011/449. `http://eprint.iacr.org/`.

Daemen, J. and Rijmen, V. (2001). National bureau of standards - advanced encryption standard. Technical Report FIPS Publication 197, National Bureau of Standards.

Early, J. (1952). Effects of space-charge layer widening in junction transistors. *Proceedings of the IRE*, 40(11):1401 –1406.

Guilley, S. (2007). *Contremesures Géometriques aux Attaques Exploitant les Canaux Cachés*. PhD thesis, Ecole Nationale Supérieure de Télécommunications, ENST.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential power analysis. In *Proceedings of CRYPTO '99*, pages 388–397, Santa Barbara, USA. Springer-Verlag.

Mangard, S., Oswald, E., and Popp, T. (2007). *Power Analysis Attacks: Revealing the secrets of smart cards*. Springer.

McAndrew, C. C. and Nagel, L. (1996). Early effect modeling in spice. *IEEE Journal of Solid-State Circuits*, 31(1):136 –138.

Mesquita, D., Techer, J.-D., Torres, L., Robert, M., Cathebras, G., Sassatelli, G., and Moraes, F. (2007). *IFIP Series: VLSI-SOC: From Systems to Chips*, chapter 7, pages 317–330. Springer Verlag, College Station, Texas.

 ©2012 SBC — Soc. Bras. de Computação

Mesquita, D., Techer, J.-D., Torres, L., Sassatelli, G., Cambon, G., Robert, M., and Moraes, F. (2005). Current mask generation: A new hardware countermeasure for masking signatures of cryptographic cores. In *Proceedings of VLSI-SoC*, page in press, Perth, Australia.

Messerges, T., Dabbish, E., and Sloan, R. (1999). Investigations of power analysis attacks on smartcards. In *Proceedings of USENIX'99*, pages 151–162.

Munro, P. C. (1991). Simulating the current mirror with a self-heating bjt model. *IEEE Journal of Solid–State Circuits*, 26:1321–1324.

Parikh, C. and Patel, P. (2007). Iperformance evaluation of aes algorithm on various development platforms. In *Proceedings of ISCE'07*, pages 1–6.

 ©2012 SBC — Soc. Bras. de Computação

# Uma arquitetura de segurança para medidores inteligentes – verificação prática de dados de energia multitarifada

**Sérgio Câmara[1,2], Raphael Machado[2], Luci Pirmez[1], Luiz F.R.C. Carmo[1,2]**

[1]PPGI iNCE/IM, Universidade Federal do Rio de Janeiro (UFRJ), RJ – Brasil

[2]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro),
Av. Nossa Senhora das Graças, 50, Xerém, Duque de Caxias, 25250-020, RJ – Brasil

{smcamara,rcmachado,lfrust}@inmetro.gov.br, luci@nce.ufrj.br

***Abstract.*** *Energy meters have become functionally complex over time, on the other hand, the correctness of their behavior is still questionable and the effort involved on new smart meters type approval is growing huge. Aiming at establishing trust on the smart meter, this paper proposes a security architecture based on a consumption authenticator. This authenticator is generated inside a secure device, placed on the metering module, using ECPVS signature scheme. Our approach considers different Time-Of-Use scenarios and presents three composing techniques for the authenticator's embedded message. Our concerns are the final size of this message and the necessary data for validating the Time-Of-Use rates consumption values.*

***Resumo.*** *Medidores de energia elétrica tornaram-se equipamentos complexos, por outro lado, seu correto funcionamento ainda é questionável e o esforço envolvido na aprovação de novos modelos aumentou imensamente. Objetivando estabelecer confiança no medidor inteligente, este artigo propõe uma arquitetura de segurança baseada em um autenticador de consumo. Este autenticador é gerado por um dispositivo seguro, localizado no módulo de medição, usando o esquema ECPVS. Nossa abordagem considera diferentes cenários de energia multitarifada e apresenta três técnicas de composição da mensagem contida no autenticador. Nossas preocupações incluem o tamanho total desta mensagem e os dados necessários para validação do consumo em cada faixa de preço.*

## 1. Introdução

As Redes Elétricas Inteligentes (*Smart Grids*) são uma realidade atualmente em diferentes localidades e continuam a demandar um grande número de pesquisas, estabelecimento de novos conceitos e esforços de desenvolvimento em busca de melhorias e confiança na rede. No Brasil, o primeiro passo em direção à Rede Inteligente foi a adoção, no começo dos anos 2000, do Sistema de Medição Centralizado (SMC), que foi desenvolvido com um objetivo específico brasileiro – reduzir as altas taxas de perda não-técnicas devido ao roubo de energia [Boccardo et al. 2010]. A arquitetura SMC determinava que os módulos de medição não mais estivessem localizados em cada residência e, sim, agrupados em um concentrador, geralmente localizado no alto do poste de luz em meio às instalações

*Artigo financiado com recursos da MCT/FINEP PLATCOG (01.10.0549.00).

 ©2012 SBC — Soc. Bras. de Computação

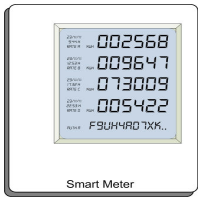



**Figura 1. Exemplo de um dispositivo mostrador exibindo 4 valores de kWh por faixa de preço e o Autenticador de Consumo Distribuído.**

elétricas de média tensão, para dificultar o acesso físico. Cada módulo de medição é associado ao seu respectivo dispositivo mostrador, localizado na residência do consumidor, no qual são atualizados constantemente os valores de consumo de energia através de uma comunicação sem-fio.

Se, por um lado, a adoção desta arquitetura tornou o roubo de energia uma tarefa mais difícil, por outro, vários medidores começaram a se comportar de maneira incorreta com o passar do tempo, apresentando típicas falhas associadas a *software*. Tal problema revelou a necessidade de um controle estrito do *software* embarcado no SMC. Em 2009, a Portaria Inmetro 011/2009 [Inmetro 2009] foi publicada, a qual estipulava requisitos específicos para esse *software* e tornava obrigatória a abertura de seu código-fonte para o Inmetro. Desde então, os medidores inteligentes são aprovados de acordo com esta portaria juntamente com a Portaria Inmetro 371/2007, relacionada à parte metrológica.

A crescente complexidade da arquitetura do medidor e de seu *software* embarcado apresentam um novo desafio para a autoridade metrológica brasileira: como validar e confiar em uma solução de medidor, ao passo que o processo de aprovação de novos modelos se torna cada vez mais dispendioso? A ideia aqui apresentada é a de assinar digitalmente os dados o mais próximo possível do tempo de "consolidação dos dados de medição". Uma vez que estes dados sejam assinados, qualquer manipulação posterior seria detectada e, dessa forma, os elementos de *hardware* e *software* não mais precisariam ser validados metrologicamente – isto é, o processo de validação de *software* pela autoridade metrológica será restrita apenas aos módulos que manipulam os dados "antes da assinatura digital". Ademais, as preocupações relacionadas aos riscos de vazamento de propriedade intelectual associados à abertura e armazenamento de código-fonte dos fabricantes seriam eliminadas.

Enquanto a adoção desta solução simplifica o processo de validação do *software* metrológico, o aumento de confiança nos medidores pela sociedade também se faz necessário. Para isso, a solução requer o estabelecimento de um padrão de assinatura seguro – que nós chamaremos de *Autenticador de Consumo Distribuído* – que poderá ser usado pelo consumidor para verificação da integridade dos valores de kWh consumidos e cobrados em sua conta. O presente artigo propõe algumas técnicas para a composição deste autenticador levando em conta seu tamanho máximo, e considerando uma abordagem prática: o autenticador ACD será lido do dispositivo mostrador (Fig. 1) e transcrito pelo consumidor durante o processo de verificação de integridade dos dados de medição. Tal abordagem permite um baixo custo no processo de verificação, uma vez que não há a necessidade de um segundo dispositivo para possibilitar a leitura dos dados.

Em resumo, nossa proposta apresenta uma arquitetura de segurança, apropriada

 ©2012 SBC — Soc. Bras. de Computação

para a autoridade metrológica, facilitando o processo de aprovação de modelos de medidores inteligentes, e para o consumidor final, permitindo a verificação da autenticidade e integridade dos dados de consumo. A proposta se concentra especificamente no cenário de energia multitarifada, onde não apenas o total do consumo, mas também o "momento do consumo", é importante no cálculo da conta, devido a diferentes preços de energia ao longo do dia.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta esforços estabelecidos e em andamento que abordam a segurança, de forma geral, nos medidores elétricos inteligentes. A Seção 3 explica de forma detalhada a contextualização e definição do problema abordado. Os elementos e mecanismos da arquitetura de segurança proposta são descritos na Seção 4. Na Seção 5, três técnicas de composição do autenticador ACD são descritas e, na Seção 6, essas técnicas são avaliadas e os resultados apresentados. A Seção 7 expõe algumas considerações de segurança da arquitetura e, finalmente, fazemos nossa conclusão na Seção 8.

## 2. Trabalhos Relacionados

A segurança da informação desempenha um papel fundamental nas Redes Inteligentes, portanto é necessário que o *software* embarcado nos medidores inteligentes atenda os requisitos de segurança estipulados pela autoridade metrológica de cada localidade e siga por uma avaliação estrita antes de sua aprovação. No Brasil, as portarias lançadas pelo Inmetro – Portaria Inmetro 011/2009 e Portaria Inmetro 366/2011 – cumprem esse papel normativo. No âmbito internacional, podemos destacar dois documentos chave que serviriam de base para as portarias do Inmetro e que abordam os problemas envolvendo a avaliação de *software* embarcado, são eles: (i) o OIML D31/2009: *General Requirements of Software Controlled Measuring Instruments* [OIML 2009] e (ii) o WELMEC 7.2: *Software Guide – Measuring Instruments Directive* 2004/22/EC [WELMEC 2008]. Ambos os documentos estabelecem requisitos para *software* embarcado em instrumentos de medição, abordando temas como identificação, separação e atualização de *software*, proteção de interfaces, proteção contra mudanças, recuperação de falhas entre outros.

Na Alemanha, o processo de avaliação de medidores inteligentes é baseado nas diretivas estabelecidas pelo WELMEC e pelo PTB – *Physikalisch-Technische Bundesanstalt* – o Instituto Metrológico Nacional alemão. Além dos medidores, a infraestrutura de distribuição alemã conta com um elemento a mais, um *gateway*, que compreende as funcionalidades relacionadas à segurança da informação. Dessa forma, ao separar as particularidades dos medidores MID das funções de segurança da informação, o *gateway* pode ser submetido a uma avaliação de Common Criteria, de acordo com um Protection Profile apropriado. Além dessa vantagem, essa abordagem permite que os Organismos Notificados lidem de forma melhor com as limitações em envolver especialistas de segurança em TI em avaliações de medidores.

Outras soluções de segurança para os medidores inteligentes estão sendo propostas fazendo o uso de módulos criptográficos integrados a arquitetura de medição. Em Teller *et al.* [Treytl et al. 2004], uma versão compacta do TPM – Trusted Platform Module – é implementada, a qual consome poucos recursos e requer um conjunto mínimo de comandos. Este módulo criptográfico é integrado ao medidor e provê proteção de propriedade intelectual, corretude do comportamento e atualização segura do *software*. Em

 ©2012 SBC — Soc. Bras. de Computação

outro projeto, a Infineon, uma fabricante de semicondutores, desenvolveu a família de chipset UMF11x0 [Infineon 2012] para medidores inteligentes, que pode ser conectados a um *smartcard* ou a um módulo de segurança de hardware. Esse *chip* oferece funções para criptografia simétrica (AES-128/256) e para uma Infraestrutura de Chave-Pública, além de proteção das chaves criptográficas e um gerador real de números aleatórios.

Apesar dos trabalhos citados anteriormente apontarem preocupações e soluções sobre a segurança e corretude do comportamento dos medidores inteligentes, o presente trabalho é o primeiro a propor uma arquitetura de segurança que disponibiliza um procedimento prático de verificação de integridade dos dados gerados por um medidor inteligente e capaz de ser realizado por um usuário comum.

## 3. Definição do Problema

O *software* embarcado nos medidores inteligentes não podem ser, *a priori*, considerados confiáveis, isto é, não há garantias que este irá se comportar corretamente ou que este é exatamente o *software* validado na aprovação de modelos. A autoridade metrológica parte do princípio que os medidores inteligentes poderiam ser modificados posteriormente pelos consumidores, pelos fabricantes ou pela distribuidora de energia, para que ajam de maneira maliciosa de acordo com seus interesses.

Medidores convencionais, sem multitarifação de energia, apenas medem acumulativamente o total de energia consumida, logo alguém poderia forçar, por exemplo, um comportamento malicioso que incremente o contador total de energia mais rápido do que o normal, para tirar vantagem desse falso consumo extra todo mês. No entanto, medidores com suporte a multitarifação (horo-sazonal) podem ser adulterados de forma mais sutil – o total de consumo de energia continuaria inalterado, porém uma (pequena) porcentagem da faixa de preço mais barata de energia poderia ser adicionada às faixas de preço de pico, dessa forma, gerando contas mais caras.

Logo, para a validação em um cenário de multitarifação de energia, uma verificação de consumo deve assegurar os valores consumidos em cada faixa de preço. Além disso, se o mecanismo não sabe a configuração destas faixas, que é o nosso caso, ele deverá consolidar dados relevantes para esse propósito e, de alguma forma segura, disponibilizar essa informação para o usuário final. Neste trabalho, isso será feito através de um código autenticador.

Por motivo da não necessidade de outros dispositivos se comunicarem com o medidor para realizar a leitura do autenticador (comentado na Seção 4.1), este será exibido no próprio dispositivo mostrador. Esta condição estipula restrições sobre o tamanho, a integridade e autenticidade do autenticador: ele deve ser tão curto quanto possível e a prova de falsificações.

Nossos esforços a seguir concentram-se na descrição do mecanismo de segurança usado, em como ele consolida os dados relevantes para a validação das faixas de consumo e em como o autenticador pode ser encurtado.

## 4. Proposta

A proposta deste trabalho é a concepção de uma arquitetura de segurança que estabeleça confiabilidade nos dados gerados pelo medidor inteligente para todas as partes interessadas – o consumidor, a distribuidora de energia elétrica e a autoridade metrológica. A

 ©2012 SBC — Soc. Bras. de Computação

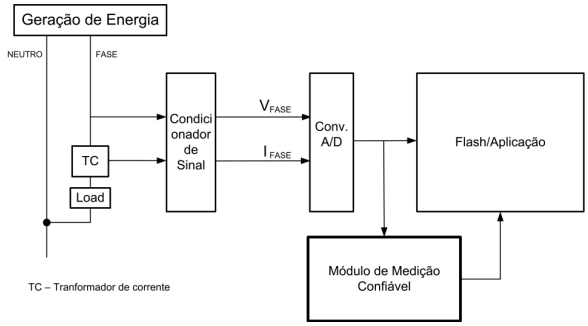



**Figura 2. Diagrama de blocos dos componentes de um módulo de medição incluindo o TMM.**

arquitetura é composta por mecanismos de segurança, como criptografia de chave-pública e assinatura digital, juntamente a procedimentos externos de verificação de dados.

A arquitetura tem sua "raiz de confiança" baseada em um *hardware* que chamaremos de *Trusted Metering Module*. O TMM é um módulo criptográfico acoplado ao começo da cadeia de medição com o objetivo de armazenar dados referentes à energia consumida e repassá-los à frente, para o resto da cadeia de medição, assinados digitalmente. Por definição, a cadeia de medição legalmente relevante é a sequência de elementos de um instrumento, ou sistema de medição, que constitui o trajeto do sinal de medição desde o estímulo até exibição do resultado ao consumidor [VIM 2008]. É mandatório que o TMM esteja posicionado na cadeia de medição antes do *software* proprietário embarcado, que possui seu próprio algoritmo de medição (Fig. 2). Essa restrição assegura que, se dados originais forem alterados de alguma forma após serem assinados digitalmente, este erro, intencional ou não, poderá ser detectado posteriormente através da rastreabilidade provida pela assinatura.

Essa assinatura digital, o Autenticador de Consumo Distribuído (ACD), é uma sequência alfa-numérica disponibilizada no visor do dispositivo mostrador. O autenticador ACD é gerado dentro do TMM e é, então, repassado para à aplicação principal do medidor, que fica encarregada de mandá-lo, juntamente com os valores de consumo de energia, para o dispositivo mostrador. Um novo autenticador ACD é exibido para cada atualização de valores no mostrador, o que acontece a cada unidade de kWh consumido em qualquer faixa de preço.

Chamaremos de Processo de Validação da Medição (PVM) o processo pelo qual o consumidor pode checar a integridade dos dados de consumo. O PVM consiste basicamente da transcrição e inserção de dados relevantes em um sistema computacional, além da execução do algoritmo de validação. As entradas que o algoritmo precisa saber são: o autenticador ACD e os valores de consumo em cada faixa horária no momento da leitura, a configuração das faixas de preço e a chave-pública do TMM.

### 4.1. Objetivos e Requisitos

Nesta Seção, os requisitos mandatórios que a arquitetura de segurança deve atender são descritos. Seguem dois objetivos gerais do projeto:

1. Diminuir o esforço empregado durante o processo de aprovação de modelos de medidores inteligentes;

 ©2012 SBC — Soc. Bras. de Computação

2. Prover confiança no comportamento do medidor inteligente para as partes interessadas, garantindo a autenticidade e integridade dos dados de consumo.

Durante o processo de aprovação de modelos de medidores, a autoridade metrológica restringiria sua análise apenas aos componentes legalmente relevantes que se encontram antes do TMM na cadeia de medição, uma vez que seria apenas necessário garantir que os dados de entrada do TMM sejam genuinamente dos sensores de corrente e voltagem. Dessa forma, o TMM garante a rastreabilidade dos dados originais. Como consequência do uso dessa arquitetura, a autoridade metrológica mitigaria os riscos associados à abertura e armazenamento do código proprietário da aplicação metrológica dos fabricantes de medidores.

Em relação ao item 2, através do Autenticador de Consumo Distribuído e do Processo de Validação da Medição, os consumidores terão meios de estabelecer confiança nos dados de consumos exibidos no visor e cobrados em sua conta de luz. Para as distribuidoras de energia elétrica, sua confiança nos medidores inteligentes em operação também aumenta, uma vez que elas poderão recuperar os autenticadores ACD remotamente e atestar se seus medidores sofreram algum tipo de violação que comprometa a integridade de seus dados.

Além dos objetivos gerais, os requisitos atendidos pelo projeto são:

- Estabelecer as mudanças necessárias na arquitetura de um medidor inteligente, priorizando a mitigação de vulnerabilidades.
- Dar suporte à funcionalidade de Postos Tarifários (*Time-Of-Use*).
- Manter um baixo custo para a solução.
- Estabelecer maneiras de compor o Autenticador de Consumo Distribuído.

O esboço do TMM e sua posição são descritos de forma a dificultar a inserção de *backdoors* no mecanismo. Mais detalhes serão discutidos na Seção 4.2. É mandatório no projeto que não haja necessidade de um segundo dispositivo se comunicar com o medidor para possibilitar o PVM, logo a leitura dos dados deverá ser possível e factível de forma manual. Dessa forma, o autenticador ACD deverá apresentar o menor número de caracteres e ainda capaz de oferecer um alto nível de acurácia e segurança para a informação que carrega. Algumas técnicas de composição do ACD serão apresentadas na Seção 5.

### 4.2. *Trusted Metering Module* (TMM)

Nesta Seção descrevemos, como segue, um esboço do *Trusted Metering Module*. O TMM é um módulo criptográfico capaz também de contabilizar o consumo de energia, além de realizar procedimentos particulares ao projeto. Seus componentes internos são: (i) o processador, (ii) o código do programa, o qual inclui as técnicas de composição do autenticador ACD, (iii) o motor de execução, (iv) a memória persistente e protegida, para a guarda de chaves criptográficas, (v) a memória versátil, que armazena variáveis e valores de energia consumidos, (vi) o motor de criptografia e assinatura digital, (vii) o motor de *hash* (resumo criptográfico) e (viii) o *Real Time Clock*, RTC.

O TMM possui duas entradas de dados: uma referente às grandezas de voltagem e corrente, provenientes das fases do circuito elétrico da instalação, amostrados por um conversor Analógico/Digital, e a outra aos comandos para correção de sincronização do

 ©2012 SBC — Soc. Bras. de Computação

RTC, localizado no TMM, com o relógio da aplicação. Essa sincronização é de responsabilidade da aplicação do fabricante, onde apenas serão permitidas correções de acordo com a deriva máxima em um ano do relógio interno do TMM. O RTC usado deve ser bem preciso, com erro abaixo de 5ppm (partes por milhão) [Arora 2011]. Em princípio, nenhuma outra entrada é considerada para qualquer propósito, de forma que as vulnerabilidades em potencial sejam minimizadas.

Na memória protegida, o TMM armazena uma chave-privada única, gerada em *hardware* em tempo de fabricação, juntamente com sua chave-pública. Esse par de chaves é baseada em curvas elípticas. Além disso, um calendário é também armazenado na memória protegida.

De fato, o TMM desconhece a configuração das faixas de preço, isto é, ele não sabe determinar o começo/término dos períodos de pico, fora-de-pico e intermediário. Portanto, o módulo lida com a multitarifação de energia da forma descrita na sequência. O TMM converte os dados de corrente e voltagem para valores de mesma magnitude de quilowatt-hora e, então, calcula e armazena os valores acumulados de energia total consumidos por períodos fixados de uma hora de duração (por exemplo, 11:00:00h – 11:59:59h). Dessa forma, existem 24 registradores armazenando valores acumulados de consumo desde o tempo de instalação do medidor até o momento presente. Além disso, existe um registrador a mais para armazenar valores de energia fora-do-pico, consumidos durante os fins-de-semana e feriados. Esses registradores precisam ser mantidos atualizados para que o autenticador ACD seja composto corretamente.

### 4.3. Autenticador de Consumo Distribuído (ACD)

Tecnicamente, o autenticador ACD é uma assinatura ECPVS (*Elliptic Curve Pintsov-Vanstone Signature*) codificado em base64. Uma vez que o esquema ECPVS, especificado em ISO/IEC 9796-3:2006, permite assinaturas de tamanho reduzido – em torno de seis vezes menor do que o RSA e metade do tamanho do ECDSA [Modares 2009] – e recuperação parcial ou total de mensagem, essas características se encaixam idealmente aos requisitos do ACD. Um exemplo de aplicação real deste esquema é o uso de assinaturas ECPVS em *Digital Postage Marks* [Certicom 2004].

De acordo com os algoritmos de assinatura digital e funções de *hash* recomendados no FIPS (*Federal Information Processing Standards Publications*) publicado pelo NIST [Gallagher et al. 2009, Barker and Roginsky 2011], os parâmetros de domínio escolhidos para o processo de assinatura do ACD são:

- Curvas elípticas de 224 bits sobre conjuntos finitos primos;
- Função *Hash* SHA-224;
- XOR Encryption Scheme (Aqui usamos o XOR como cifrador simétrico uma vez que não há real necessidade de confidencialidade da mensagem-ACD, porém a previne de ataques passivos, adicionando uma camada fraca de confidencialidade [Certicom 2000])

O processo de geração da assinatura ECPVS é descrito na sequência [ISO/IEC 2006]. A entrada para o processo consiste dos parâmetros de domínio, a chave-privada $x_A$ e uma mensagem $M$ para ser assinada. Então, o assinante divide a mensagem $M$ em $M_{clr} \mid\mid M_{rec}$, onde $M_{clr}$ é a parte da mensagem enviada em

 ©2012 SBC — Soc. Bras. de Computação

claro e $M_{rec}$ é a parte recuperável da mensagem. $M_{clr}$ e $M_{rec}$ devem ser formatadas e codificadas de maneira acordada anteriormente entre o assinante e o verificador. Seja $d$ o dado de entrada $M_{rec}$ (com redundância natural ou adicionada), $n$ um número primo, $k$ um inteiro aleatório no intervalo $[1, n-1]$, $\Pi$ a chave simétrica computada por uma *Key Derivation Function* (KDF) a partir da chave-pública do assinante, a assinatura ECPVS deve ser computada pela seguinte, ou equivalente, sequência de passos:

1. Computar $r = Sym(d, \Pi)$;
2. $u = Hash(r||M_{clr})$;
3. Converta $t = OS2IP(u)$; (octet-string-to-integer primitive function), note que $t \in [0, n-1]$;
4. Se $t = 0$, então o processo de assinatura deve ser repetido com um novo valor aleatório $k$;
5. Computar $s = (k - x_A t) \quad mod \quad n$;
6. Se $s = 0$, então o processo de assinatura deve ser repetido com um novo valor aleatório $k$;
7. Apague $k$;

Gere como saída a assinatura $(r, s)$ e a mensagem parcial $M_{clr}$ (que pode ser nula).

Para o autenticador ACD, chamamos $M$ de mensagem-ACD, onde $M_{rec}$ é exatamente $M$ e $M_{clr}$ é nula. O tamanho final do autenticador ACD depende do tamanho de $s$, o *overhead* criptográfico, e $r$. Uma vez que o tamanho de $r$ está diretamente associado ao tamanho da mensagem-ACD, otimizar a composição desta mensagem se faz necessário, como apresentaremos na próxima seção.

## 5. Técnicas de Composição da mensagem-ACD

A mensagem-ACD deve conter dados suficientes para ser possível validar os valores de consumo em kWh em cada faixa de preço. Uma vez que o TMM não conhece as faixas definidas, a mensagem-ACD representará, de alguma forma, os valores dos registradores do TMM tornando possível o Processo de Validação da Medição.

Para escolher o número apropriado de bits para cada dado na mensagem, tomaremos em consideração as seguintes suposições: (i) o consumo mensal máximo de uma residência é de 4.000 kWh. No Brasil, o consumo médio é bem abaixo disso [Francisquini 2006], portanto estamos considerando a grande maioria de classes de consumidores e (ii) a vida útil máxima de operação de um medidor inteligente é de 20 anos. Neste caso, estamos também considerando um valor razoavelmente alto em relação à vida útil de um medidor eletrônico, que é de 12 anos aproximadamente [Guimarães ].

### 5.1. Técnica por Valores Absolutos

A Técnica por Valores Absolutos é a nossa primeira tentativa de construir a mensagem-ACD. Sejam os valores nos registradores do TMM $Val_i$, para $i \in [0, 23]$, logo suas representações ($Rep_i$) na mensagem-ACD são determinadas por

$$Rep_i = Val_i \tag{1}$$

De acordo com as suposições feitas acima, sobre consumo mensal máximo de uma residência e vida útil máxima de um medidor, o valor de $Val_i$ não deverá ultrapassar

 ©2012 SBC — Soc. Bras. de Computação

40.000 kWh. Logo, fixaremos $k = 16$, onde $k$ é o número de bits necessários para $Rep_i$ (podendo variar de 0 a 65.535) e compor a mensagem-ACD como segue:

- Soma dos períodos fora-de-pico (fins-de-semana e feriados): 20 bits;
- As 24 representações dos valores dos registradores ($Rep_0 - Rep_{23}$): 16 bits para cada representação;
- *Hash* dos 24 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

A primeira parte da assinatura, chamada $r$, compreende a mensagem-ACD criptografada com o cifrador simétrico. A mensagem é o resultado da concatenação dos três *bit strings* apresentados acima. A segunda parte, $s$, é o *overhead* criptográfico de 28 bytes (esperado de uma chave-privada ECC-224). Logo, o tamanho final da assinatura $(r, s)$ é 54,5 + 28 = 82,5 bytes, gerando 110 caracteres codificados em base64.

Como uma primeira manobra para reduzir o tamanho da assinatura, sugerimos que uma compactação de dados seja realizada. Uma vez que, aos nossos conhecimentos, todas as modalidades existentes atualmente de multitarifação de energia consideram o período entre 23:00h e 5:59h como fora-de-pico, poderemos somar todos os valores dos sete registradores correspondentes ao bloco de bits para períodos fora-de-pico (antes apenas reservado para fins-de-semana e feriados) e, dessa forma, reduzir o tamanho da mensagem-ACD sem prejudicar o resultado final. Com essa alteração, o novo tamanho do autenticador ACD $(r, s)$ seria de 40,5 + 28 = 68,5 bytes, gerando 92 caracteres codificados em base64 acrescido de dois caracteres de *padding* ("=="). Com a Técnica por Valores Absolutos, essa é a menor assinatura esperada. Nas seções seguintes, apresentamos técnicas que utilizam compressão lógica de dados.

### 5.2. Técnica por Contadores em Módulo

Essa técnica de compressão lógica estabelece que o valor do registrador, $Val_i$, é representado na mensagem-ACD por um contador sequencial que incrementa ao passo que $Val_i$ é incrementado, porém restringido ao intervalo de inteiro $[0, 2^{k_{mod}} - 1]$, $k < 16$. Portanto, seja $k_{mod}$ o número de bits usados para representar os valores dos registradores, $Rep_i$ é calculado pela seguinte fórmula:

$$Rep_i = Val_i \; mod \; 2^{k_{mod}} \tag{2}$$

A mensagem-ACD é composta, portanto, pelas seguintes *bit strings* concatenadas: (i) soma total dos períodos fora-de-pico (incluindo 23h-5:59h), (ii) as 17 representações

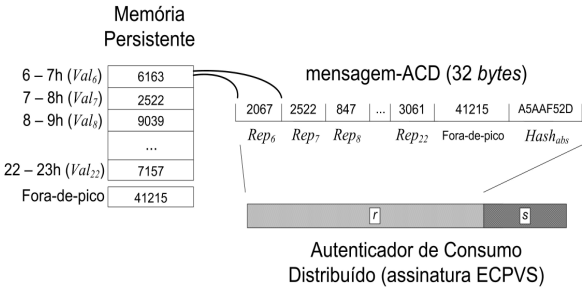



**Figura 3. Exemplo de composição da mensagem-ACD pela Técnica por Contadores em Módulo ($k_{mod} = 12$).**

 ©2012 SBC — Soc. Bras. de Computação

dos valores nos registradores ($Rep_6 - Rep_{22}$) e (iii) *Hash* (por exemplo, SHA-224) dos 17 valores originais dos registradores concatenados e separados por ","($Hash_{abs}$).

No Processo de Validação da Medição (PVM), o algoritmo de descompressão recupera os dados originais ($Val_6 - Val_{22}$) a partir das representações ($Rep_6 - Rep_{22}$) na mensagem-ACD. Este algoritmo opera da seguinte forma:

1. Computar lista $Lcand_i$ de todos os possíveis candidatos para $Rep_i$, $i \in [6, 22]$;
2. Computar todas as combinações de elementos de ($Lcand_i, \cdots, Lcand_{i+j}$), o qual este grupo pertence a uma faixa de preço e a soma dos seus elementos é igual ao consumo da faixa de preço apresentado como entrada;
3. Incluir a combinação de elementos em uma lista de Sequências Candidatas, $Lseq_{cand}$ ;
4. Concatenar todas as listas $Lseq_{cand}$ de todas as faixas de preço, comparando com $Hash_{abs}$ ;
5. Quando houver igualdade, os valores de consumo originais foram recuperados.

Com os dados originais recuperados, isso significa que a validação da medição foi positiva e o PVM realizado com sucesso.

### 5.3. Técnica por Contadores em Função de Dispersão

A Técnica por Contadores em Função de Dispersão apresenta uma estrutura parecida com a Técnica por Contadores em Módulo. Sua diferença está no fato de que o valor de consumo de um registrador, $Val_i$, é representado por um contador aleatório, e não mais por um contador sequencial como na técnica anterior. Os valores desse contador aleatório também estão restringidos por um intervalo de inteiros $[0, 2^{k_{hash}} - 1]$, $k < 16$. O cálculo de $Rep_i$ é dado pela fórmula:

$$Rep_i = ultimosBits(\ k_{hash}, Hash(\ concat(\ Val_i, ID_i))) \tag{3}$$

A função $Hash()$ representa qualquer função com propriedade unidirecional, com resistência a colisões e que gere um dado de saída com pelo menos $k_{hash}$ bits (SHA-224, por exemplo). Para essa técnica, convencionou-se utilizar os últimos $k_{hash}$ bits da saída da função $Hash()$ para preencher $Rep_i$, no entanto, qualquer outra combinação poderia ter sido escolhida.

A concatenação de $Val_i$ com um identificador próprio do registrador é justificado por uma questão de melhoria na segurança do autenticador. Com esse identificador, a saída de $Hash()$ para um mesmo $Val$ será diferente dependendo do registrador $i$ em que o valor estiver localizado. Essa manobra atribui aos $Rep_i$ uma aleatoriedade entre si, dificultando ainda mais um possível ataque de injeção de dados falsos.

O algoritmo de descompressão usa Tabelas $Rainbow$ durante o processo de listagem de valores candidatos para $Lcand_i$. Essas tabelas mapeiam todos os valores de *Hash()* para suas respectivas entradas $concat(Val_i, ID_i)$, onde $Val_i \in [0, 40.000]$ e $i \in [0, 16]$. O algoritmo, então, procede da mesma forma descrita na Seção 5.2.

## 6. Análise Experimental das Técnicas e Resultados

Nesta seção apresentamos os testes realizados com as técnicas de compressão descritas anteriormente e resolvemos o valor de $k$ apropriado para cada uma. De fato, essas

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Tempo de descompressão x $Interval_{max}$, $k_{mod} \in [8, 14]$ – Técnica por Contadores em Módulo**

| $Interval_{max}$ | Tempo de descompressão (seg) | | | | | | |
|---|---|---|---|---|---|---|---|
| (kWh) | $k_{mod}$ = 8 | $k_{mod}$ = 9 | $k_{mod}$ = 10 | $k_{mod}$ = 11 | $k_{mod}$ = 12 | $k_{mod}$ = 13 | $k_{mod}$ = 14 |
| 1000 | 8,9E+02 | 7,6E-02 | 5,6E-05 | 5,6E-05 | 5,6E-05 | 5,6E-05 | 5,6E-05 |
| 2000 | 1,4E+07 | 9,5E+02 | 7,7E-02 | 5,6E-05 | 5,6E-05 | 5,6E-05 | 5,6E-05 |
| 3000 | 1,8E+09 | 2,5E+05 | 1,6E+01 | 4,0E-03 | 5,6E-05 | 5,6E-05 | 5,6E-05 |
| 4000 | - | 1,4E+07 | 7,7E+02 | 8,5E-02 | 5,6E-05 | 5,6E-05 | 5,6E-05 |
| 5000 | - | 1,0E+08 | 1,6E+04 | 2,1E+00 | 3,5E-04 | 5,6E-05 | 5,6E-05 |
| 6000 | - | - | 2,2E+05 | 1,9E+01 | 4,1E-03 | 5,6E-05 | 5,6E-05 |
| 7000 | - | - | 2,1E+06 | 1,5E+02 | 2,1E-02 | 5,6E-05 | 5,6E-05 |
| 8000 | - | - | 1,3E+07 | 9,1E+02 | 8,0E-02 | 5,6E-05 | 5,6E-05 |
| 9000 | - | - | 4,3E+07 | 4,7E+03 | 4,6E-01 | 9,7E-05 | 5,6E-05 |
| 10000 | - | - | 2,0E+07 | 1,7E+04 | 2,1E+00 | 3,3E-04 | 5,6E-05 |
| 11000 | - | - | - | 7,2E+04 | 5,8E+00 | 1,3E-03 | 5,6E-05 |
| 12000 | - | - | - | 2,3E+05 | 1,5E+01 | 4,8E-03 | 5,6E-05 |
| 13000 | - | - | - | 7,9E+05 | 4,6E+01 | 1,2E-02 | 5,6E-05 |
| 14000 | - | - | - | 1,8E+06 | 1,5E+02 | 2,6E-02 | 5,6E-05 |
| 15000 | - | - | - | 5,6E+06 | 3,8E+02 | 4,4E-02 | 5,6E-05 |
| 16000 | - | - | - | 1,3E+07 | 8,0E+02 | 7,1E-02 | 5,6E-05 |
| 17000 | - | - | - | 2,9E+07 | 2,2E+03 | 1,3E-01 | 5,9E-05 |
| 18000 | - | - | - | - | 4,5E+03 | 4,7E-01 | 1,1E-04 |
| 19000 | - | - | - | - | 1,1E+04 | 8,3E-01 | 1,8E-04 |
| 20000 | - | - | - | - | 1,8E+04 | 2,0E+00 | 2,9E-04 |

técnicas utilizam algoritmos de compressão assimétricos, uma vez que o tempo gasto para recuperar os dados comprimidos pode ser bem maior do que o tempo de compressão, isso dependendo do valor usado em $k_{mod}$, $k_{hash}$. Outro conceito importante é o intervalo entre os valores maior e menor dos registradores em cada faixa de preço ($Interval_{max} = Val_{max} - Val_{min}$). Conforme $Interval_{max}$ aumenta, e devido a uma etapa de força-bruta nos algoritmos de descompressão, este processo pode demorar bastante. O tempo máximo assumido para descompressão é de até um minuto, embora este não seja um requisito mas apenas uma escolha razoável.

Para determinar o número apropriado de bits para $Rep_i$, e encontrar o melhor *trade-off* entre o tempo de descompressão e o tamanho total da mensagem-ACD, para cada $k_{mod}, k_{hash} \in [8, 14]$ e para cada $Interval_{max} \in [1000, 20000]$ foram simulados 100 ACDs sob a modalidade tarifária do Ontario Energy Board [OEB ]. A simulação foi realizada em um PC, Intel Q6600 CPU 2.40GHz, 8GB RAM. Os autenticadores ACD simulados e as técnicas de compressão de dados foram implementados em linguagem C++.

A Tabela 1 apresenta os resultados da simulação obtidos com a Técnica por Contadores em Módulo. Dessa tabela, devemos considerar, para o tempo de descompressão abaixo de um minuto, $k_{mod} = 11$ permitindo um $Interval_{max}$ = 6.000 kWh e $k_{mod} = 12$ permitindo um $Interval_{max}$ = 13.000 kWh. Apesar da primeira opção oferecer um tamanho menor para a mensagem-ACD, seu $Interval_{max}$ permitido é considerado pequeno para garantir um tempo de descompressão aceitável durante a vida útil do medidor, logo definiremos $k_{mod} = 12$ como a opção de melhor custo-benefício. Assim sendo, pela Técnica por Contadores em Módulo, o tamanho final do autenticador ACD $(r, s)$ é de 32 + 28 = 60 bytes, resultando em 80 caracteres codificados em base64. A mensagem-ACD é formada com os seguintes dados (Fig. 3):

- Soma dos períodos fora-de-pico (incluindo 23h-5:59h): 20 bits;
- As 17 representações dos valores dos registradores ($Rep_6 - Rep_{22}$): 12 bits para cada representação;
- *Hash* dos 17 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 2. Tempo de descompressão x $Interval_{max}$, $k_{hash} \in [8, 14]$ – Técnica por Contadores em Função de Dispersão**

| $Interval_{max}$ (kWh) | Tempo de descompressão (seg) | | | | | | |
|---|---|---|---|---|---|---|---|
| | $k_{hash}$ = 8 | $k_{hash}$ = 9 | $k_{hash}$ = 10 | $k_{hash}$ = 11 | $k_{hash}$ = 12 | $k_{hash}$ = 13 | $k_{hash}$ = 14 |
| 1000 | 1,37E-02 | 3,28E-03 | 3,40E-03 | 3,25E-03 | 3,21E-03 | 3,17E-03 | 3,30E-03 |
| 2000 | 8,06E+00 | 5,14E-03 | 2,92E-03 | 2,75E-03 | 2,79E-03 | 2,84E-03 | 2,80E-03 |
| 3000 | 7,88E+02 | 7,15E-02 | 2,98E-03 | 2,69E-03 | 2,61E-03 | 2,73E-03 | 2,99E-03 |
| 4000 | 2,66E+04 | 8,97E-01 | 3,90E-03 | 2,77E-03 | 2,72E-03 | 2,91E-03 | 2,65E-03 |
| 5000 | 6,46E+05 | 1,03E+01 | 6,29E-03 | 2,80E-03 | 2,73E-03 | 2,84E-03 | 2,79E-03 |
| 6000 | 1,15E+07 | 9,79E+01 | 1,51E-02 | 2,95E-03 | 2,61E-03 | 2,88E-03 | 3,30E-03 |
| 7000 | - | 1,10E+03 | 6,03E-02 | 3,08E-03 | 2,94E-03 | 2,83E-03 | 2,69E-03 |
| 8000 | - | 6,81E+03 | 1,67E-01 | 3,55E-03 | 2,80E-03 | 2,74E-03 | 2,77E-03 |
| 9000 | - | 1,99E+04 | 8,03E-01 | 3,86E-03 | 2,88E-03 | 2,80E-03 | 2,81E-03 |
| 10000 | - | 8,02E+04 | 1,67E+00 | 5,09E-03 | 2,85E-03 | 2,65E-03 | 2,66E-03 |
| 11000 | - | 3,33E+05 | 6,81E+00 | 5,48E-03 | 2,75E-03 | 2,76E-03 | 2,87E-03 |
| 12000 | - | - | 1,84E+01 | 7,88E-03 | 3,00E-03 | 2,92E-03 | 3,15E-03 |
| 13000 | - | - | 3,84E+01 | 1,12E-02 | 2,85E-03 | 2,84E-03 | 2,84E-03 |
| 14000 | - | - | 9,31E+01 | 2,03E-02 | 3,03E-03 | 2,92E-03 | 3,11E-03 |
| 15000 | - | - | 2,35E+02 | 2,19E-02 | 3,05E-03 | 2,84E-03 | 2,88E-03 |
| 16000 | - | - | 4,71E+02 | 4,42E-02 | 3,27E-03 | 2,90E-03 | 2,88E-03 |
| 17000 | - | - | 1,19E+03 | 6,29E-02 | 3,34E-03 | 2,99E-03 | 2,80E-03 |
| 18000 | - | - | 1,97E+03 | 1,14E-01 | 3,50E-03 | 2,92E-03 | 2,96E-03 |
| 19000 | - | - | 5,83E+03 | 2,03E-01 | 3,49E-03 | 2,80E-03 | 2,68E-03 |
| 20000 | - | - | 1,03E+04 | 3,33E-01 | 4,13E-03 | 2,94E-03 | 3,31E-03 |

Da simulação, calculamos que, para $k_{mod} = 12$ e $Hash_{abs} = 32$ bits, a chance máxima de encontrar uma solução não-única no processo de descompressão é de $0,001\%$, e a chance média de solução não-única é de $1,902$ x $10^{-4}\%$.

A Tabela 2 apresenta os dados simulados referentes à Técnica por Contadores em Função de Dispersão. Pela tabela constatamos que para $k_{hash} = 10$, com um $Interval_{max}$ = 13.000, temos um tempo de descompressão médio de 38,4 segundos. Logo, para a Técnica por Contadores em Função de Dispersão, definiremos $k_{hash} = 10$ como a opção de melhor custo-benefício. Dessa forma, o tamanho do novo autenticador ACD $(r, s)$ seria de 27,75 + 28 = 55,75 bytes, resultando em 75 caracteres codificados em base64 acrescido de dois caracteres de *padding* ("=="). Para essa técnica, a mensagem-ACD será composta pelos seguintes dados:

- Soma dos períodos fora-de-pico (incluindo 23h-5:59h): 20 bits;
- As 17 representações dos valores dos registradores ($Rep_6 - Rep_{22}$): 10 bits para cada representação;
- *Hash* dos 17 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

Com a Técnica por Contadores em Função de Dispersão, conseguimos diminuir o tamanho da mensagem-ACD para 222 bits, contra 256 bits da Técnica por Contadores em Módulo e 324 bits da Técnica por Valores Absolutos. Isso representa 68,5% (quociente de compressão) do tamanho da mensagem-ACD usada na primeira técnica apresentada. O autenticador ACD, como um todo, teve uma redução de 67,57% no tamanho desde sua primeira tentativa de composição, caindo de 110 para 75 caracteres, tornando-se mais prático para transcrição.

A redução em aproximadamente $1/3$ do tamanho da mensagem-ACD foi conseguida por técnicas de compressão lógica elaboradas especificamente para esse problema. Algoritmos mais comuns usados para compressão de dados, como os baseados em dicionário, não teriam o mesmo sucesso ao processarem uma mensagem de tamanho pequeno como entrada. Em geral, esses algoritmos necessitam de, no mínimo, 1000 bytes de dados de entrada para tornarem-se eficientes [Chu 1996].

 ©2012 SBC — Soc. Bras. de Computação

## 7. Considerações de Segurança

Para qualquer tentativa da aplicação do medidor, ou da distribuidora, de falsificar o autenticador ACD (apresentando mais kWh nos períodos de pico, por exemplo), o atacante deverá estar consciente de que todos os ACDs seguintes devem estar coerentes com os ACDs verificados anteriormente, uma vez que os autenticadores carregam valores acumulados. A probabilidade de falsificação está diretamente associada ao tamanho da chave-privada e função de *hash* usados para assinar, juntamente com a redundância conhecida na mensagem embutida. Ataques de disponibilidade, quando o autenticador não é exibido pelo dispositivo mostrador, seria facilmente detectado pelo usuário.

O *overhead* criptográfico da assinatura ECPVS poderia ser também reduzido, porém isso implicaria na mudança do tamanho da chave usada para assinar. Por exemplo, chaves ECC de 160 bits gerariam um *overhead* de 20 bytes. No entanto, de acordo com o NIST, chaves baseadas em curvas elípticas de 160 bits não estão aprovadas para uso após 2013 [Barker and Roginsky 2011].

## 8. Conclusão

Neste trabalho, apresentamos uma proposta de arquitetura de segurança que oferece confiança no comportamento e autenticidade dos dados do medidor inteligente para o usuário comum e a distribuidora de energia. Além disso, essa arquitetura pode ser usada para diminuir os esforços da autoridade metrológica durante o processo de aprovação de novos modelos de medidores e, como consequência, mitigar os riscos associados com a abertura e armazenamento de código proprietário dos fabricantes.

A proposta considera diferentes cenários de energia multitarifada e introduz o esboço de um mecanismo baseado em *hardware* confiável e um autenticador de consumo. Cada novo autenticador ACD é exibido pelo medidor para cada atualização de valores de consumo e carrega dados relevantes para validação das medições realizadas. Nós mostramos que, com técnicas de compressão e compactação de dados, o autenticador ACD pode ser reduzido para 67,57% de seu tamanho original, permitindo uma transcrição mais prática para o usuário.

Uma ideia considerada para trabalhos futuros é a criação de um autenticador ACD "relativo" e ainda menor, o qual poderia validar as medições de um medidor inteligente em um período curto de tempo, compreendido entre a coleta de dois autenticadores ACD diferentes.

## Referências

Arora, M. (2011). Prevent tampering in energy meters. `http://www.eetimes.com/design/smart-energy-design/4014235/Prevent-tampering-in-energy-meters/`. Acessado em Novembro de 2011.

Barker, E. and Roginsky, A. (2011). NIST special publication 800-131A transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths.

Boccardo, D., Gomes dos Santos, L., da Costa Carmo, L., Dezan, M., Machado, R., and de Aguiar Portugal, S. (2010). Software evaluation of smart meters within a legal metrology perspective: A brazilian case. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1 –7.

 ©2012 SBC — Soc. Bras. de Computação

Certicom (2000). *Standards for Efficient Cryptography 1: Elliptic Curve Cryptography*. Certicom Research, Ontario, Canada. Version 1.0.

Certicom (2004). Code and cipher. Certicom's Bulletin of Security and Cryptography, 1(3), 1 -5.

Chu, K.-C. (1996). Composite dictionary compression system. Patent, US 5530645, Estados Unidos.

Francisquini, A. A. (2006). Estimação de curvas de carga em pontos de consumo e em transformadores de distribuição. Master of electrical engineering, Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista Júlio de Mesquita Filho.

Gallagher, P., Foreword, D. D., and Director, C. F. (2009). FIPS PUB 186-3 federal information processing standards publication digital signature standard (DSS).

Guimarães, L. C. Audiência pública 012/2006. `http://www.aneel.gov.br/aplicacoes/audiencia/arquivo/2006/012/contribuicao/abradee_luiz_carlos_guimaraes.pdf`. Associação Brasileira de Distribuidores de Energia Elétrica. Acessado em Abril de 2012.

Infineon (2012). Electric metering – product brief. `http://www.infineon.com/smartmeter`. Acessado em Fevereiro de 2012.

Inmetro (2009). Portaria Inmetro nº 011 de 13 de janeiro de 2009. Instituto Nacional de Metrologia, Qualidade e Tecnologia.

ISO/IEC (2006). *ISO/IEC 9796-3:2006: Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms.* International Organization for Standardization, Geneva, Switzerland, second edition.

Modares, H. (2009). A scalar multiplication in elliptic curve cryptography with binary polynomial operations in galois field. Master of computer science, The Faculty of Computer Science and Information Technology, University Malaya.

OEB. Electricity prices. `http://www.ontarioenergyboard.ca/OEB/Consumers/Electricity/Electricity+Prices`. Ontario Energy Board. Acessado em Abril de 2012.

OIML (2009). OIML D31 requirements of software controlled measuring instruments. International Organization of Legal Metrology.

Treytl, A., Roberts, N., and Hancke, G. (2004). Security architecture for power-line metering system. In *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pages 393 – 396.

VIM (2008). *International vocabulary of metrology — Basic and general concepts and associated terms*. Joint Committee on Guides for Metrology (JCGM), third edition.

WELMEC (2008). WELMEC 7.2 software guide – measuring instruments directive 200/22/ec. European Cooperation in Legal Metrology.

 ©2012 SBC — Soc. Bras. de Computação

# Redes Virtuais Seguras: Uma Nova Abordagem de Mapeamento para Proteger contra Ataques de Disrupção na Rede Física

**Rodrigo R Oliveira[1], Leonardo R Bays[1], Daniel S Marcon[1], Miguel C Neves[1], Luciana S Buriol[1], Luciano P Gaspary[1], Marinho P Barcellos[1]**

[1]Universidade Federal do Rio Grande do Sul (UFRGS),
Instituto de Informática

{ruas.oliveira, lrbays, daniel.stefani, mcneves, buriol, paschoal, marinho}@inf.ufrgs.br

***Resumo.*** *Na virtualização de redes, roteadores e enlaces virtuais são alocados sobre uma infraestrutura de rede física. Tal característica representa uma vulnerabilidade a ataques de negação de serviço na rede física, visto que um único dispositivo físico comprometido afeta todos os virtuais sobrepostos. Trabalhos anteriores propõem a reserva de recursos sobressalentes. Apesar de funcional, esse tipo de solução agrega custo ao provedor da rede física. Neste artigo, propõe-se uma abordagem para alocação de redes virtuais que explora o compromisso entre a resiliência a ataques e a eficiência na utilização de recursos. A abordagem é separada em duas estratégias, uma preventiva e uma reativa. A primeira aloca enlaces virtuais em múltiplos caminhos do substrato, enquanto a segunda tenta recuperar a capacidade dos enlaces virtuais afetada por um ataque de negação de serviço subjacente. Ambas as estratégias são formuladas como problemas de otimização. Resultados numéricos demonstram o nível de resiliência a ataques propiciado pela abordagem e o baixo custo decorrente da mesma.*

***Abstract.*** *In network virtualization, virtual routers and links are embedded into a physical network infrastructure. Such characteristic represents a vulnerability as a compromised physical device affects all the overlaid virtual ones. Previous work proposes setting aside backup resources. Although effective, this solution aggregates cost to infrastructure providers. In this paper, we propose a virtual network allocation approach which explores the trade-off between resilience to attacks, and efficiency in resource utilization. Our approach is composed of two strategies, one preventive and the other reactive. The former allocates virtual links into multiple substrate paths, while the latter attempts to recover the capacity of virtual links affected by an underlying DoS attack. Both strategies are formulated as optimization problems. Numerical results show the level of resilience to attacks and the low cost demanded by our approach.*

## 1. Introdução

A virtualização de redes consiste na instanciação de múltiplas redes virtuais sobre um mesmo substrato de rede física. Esse paradigma permite o isolamento entre redes virtuais, propiciando a independência funcional entre as mesmas. Uma de suas mais promissoras vantagens é a capacidade de limitar o escopo de ataques, através da organização de uma infraestrutura em múltiplas redes virtuais, isolando o tráfego das mesmas [Khan *et al* 2012]. De maneira geral, as redes virtuais são requisitadas por diferentes "Provedores de Serviço" (*Service Providers* – SP), e alocadas sob-demanda

 ©2012 SBC — Soc. Bras. de Computação

nos recursos físicos de "Provedores de Infraestrutura" (*Infrastructure Provider* – InP) [Belbekkouche *et al* 2012]. A alocação significa designar uma parcela dos nós e enlaces físicos do substrato aos nós e enlaces virtuais sobrepostos. Dessa forma, as redes virtuais compartilham os recursos dos dispositivos da rede física [Chowdhury and Boutaba 2010]. Tal particularidade tende a aumentar a dependência de certos recursos físicos. Consequentemente, um atacante pode lançar ataques de negação de serviço às redes virtuais, comprometendo para isso roteadores e/ou enlaces do substrato físico subjacente. Especificamente, caso determinado enlace do substrato seja comprometido, todos os enlaces virtuais sobrepostos (ou seja, alocados neste) serão afetados[1].

Para lidar com esse problema, a literatura propõe estratégias que reservam determinada quantidade de recursos do substrato como sobressalentes. Essa reserva pode ser uma fração de todos os recursos da rede [Rahman *et al* 2010] ou a alocação extra de nós disjuntos [Yeow *et al* 2010], caminhos disjuntos [Chen *et al* 2010,Guo *et al* 2011] ou ambos [Yu *et al* 2011]. Apesar de eficaz, essa abordagem pode ser muito custosa ao Provedor de Infraestrutura visto que os recursos sobressalentes (a) ficam ociosos caso não haja disrupções e (b) a ocupação extra pode limitar a alocação de novas redes. Dessa forma, a aplicabilidade dessas estratégias pode ficar limitada a nichos de mercado, deixando pragmaticamente as futuras redes virtuais vulneráveis a ataques de negação de serviço.

Neste artigo é proposta uma abordagem para aumentar a resiliência das redes virtuais dos Provedores de Serviço contra ataques de DoS, sem prejudicar a alocação dos recursos físicos do Provedor de Infraestrutura. A abordagem pode ser dividida em duas estratégias complementares, formuladas como problemas de otimização linear inteira. Primeiro utiliza-se uma estratégia **preventiva** para mitigar o impacto de um ataque. Nessa estratégia os enlaces virtuais são alocados em *múltiplos caminhos preferencialmente disjuntos* para impedir que toda a capacidade dos enlaces virtuais seja afetada quando um enlace físico subjacente é comprometido. Após a ocorrência de um ataque, utiliza-se uma estratégia **reativa** para recuperar, *parcial ou integralmente*, a capacidade perdida nos enlaces virtuais afetados.

A abordagem proposta acarreta dois problemas principais: (i) a alocação com *priorização por caminhos disjuntos* levar à sobrecarga de determinados enlaces físicos, deixando pouca ou nenhuma capacidade disponível, potencialmente impedindo a alocação de novas redes que dependem daquele enlace; e (ii) o uso de múltiplos caminhos, com atrasos de propagação potencialmente distintos, causa o problema da entrega de pacotes fora de ordem. Para mitigar o primeiro problema propõe-se o balanceamento de carga entre os enlaces do substrato, evitando saturar os caminhos escolhidos. Para tratar do segundo, alocações são feitas buscando minimizar a diferença dos atrasos entre caminhos, em linha com a literatura [Zhang *et al* 2010]. Tais escolhas serão discutidas na Seção 3.

As principais contribuições deste artigo são sumarizadas a seguir:

- **Modelagem dos três fatores do problema, combinando relativamente o impacto de caminhos disjuntos, balanceamento de carga e alocação eficiente de recursos (Seção 3).** Obtém-se assim um equilíbrio entre a resiliência das redes virtuais e a eficiência de alocação de recursos.

[1]Este artigo concentra-se em ataques, mas as soluções propostas também aplicam-se a falhas.

 ©2012 SBC — Soc. Bras. de Computação

- **Formulação das estratégias como problemas de otimização (Seção 4).** As duas estratégias propostas neste artigo, a *alocação em múltiplos caminhos* e a *redistribuição oportunística de carga*, são formuladas em programação linear inteira. As formulações geram o *resultado ótimo* de acordo com a priorização desejada.
- **Análise do compromisso gerado pela solução (Seção 5).** Experimentos demonstram que a abordagem proposta pode reduzir substancialmente o impacto de ataques de negação de serviço, porém ao custo de uma taxa de aceitação menor. Este artigo responde a seguinte questão: *como sobreviver à disrupção nas redes virtuais aumentando minimamente o custo?*

O restante do artigo está organizado da seguinte forma: a Seção 2 descreve trabalhos relacionados. A Seção 3 define o modelo de ataque e os principais conceitos relacionados à alocação de redes virtuais, ao atraso diferencial e ao objetivo. A Seção 4 apresenta as formulações em programação linear inteira das duas estratégias desenvolvidas. Por fim, os resultados da avaliação dos modelos são discutidos na Seção 5, e a Seção 6 conclui o trabalho.

## 2. Trabalhos relacionados

O presente trabalho se concentra em ataques de negação de serviço a redes virtualizadas através do comprometimento de enlaces físicos. O problema pode ser relacionado à pesquisa em sobrevivência em redes. Embora tal assunto não seja novo, apenas recentemente a comunidade científica começou a tratar dessa questão no contexto de redes virtuais. Esta seção discute os trabalhos relacionados, tanto em um contexto mais amplo como especificamente em redes virtualizadas.

A principal diferença entre o tratamento de disrupções nas redes virtuais para as redes convencionais (p.ex., redes ópticas, MPLS ou no planejamento de tráfego IP) é que na virtualização de redes o tráfego é gerado pelas redes virtuais, as quais podem ser instaladas e retiradas da rede sob-demanda. Em contraste, nas redes convencionais o tráfego entre os nós é conhecido e considerado como um parâmetro do modelo [Medhi 2006]. Ademais, as redes virtuais devem garantir a topologia, ou seja, os enlaces virtuais devem permanecer ativos. Por outro lado, nas redes convencionais basta manter a conectividade [Belbekkouche *et al* 2012].

De maneira geral, a resiliência das redes virtuais é provida através da alocação de recursos sobressalentes. [Rahman *et al* 2010], [Chen *et al* 2010] e [Guo *et al* 2011] estudam a *disrupção de enlaces do substrato físico*. [Rahman *et al* 2010] reservam determinado percentual de banda de cada enlace do substrato para proteger enlaces virtuais contra disrupções. Os autores calculam "k-caminhos mais curtos" entre os nós de cada enlace físico, resultando em micro-desvios. Caso um enlace físico seja interrompido, o fluxo é redirecionado pelos recursos sobressalentes desses desvios. Por sua vez, [Chen *et al* 2010] e [Guo *et al* 2011] utilizam a alocação de caminhos disjuntos, entre os mapeamentos primário e sobressalente, como estratégia para prover resiliência a disrupções.

Por outro lado, [Yeow *et al* 2010] estudam a *negação de serviço em nós da rede física*. Os autores consideram a existência de nós virtuais críticos. Para proteger esses nós, os autores oferecem redundância por meio da reserva de nós físicos sobressalentes. Apesar de não considerar disrupções em enlaces físicos, a estratégia também reserva caminhos sobressalentes para manter a topologia. Isso pode resultar em um número potencialmente grande de enlaces físicos alocados como

 ©2012 SBC — Soc. Bras. de Computação

sobressalentes, visto que cada nó redundante precisa manter a conectividade dos nós críticos mapeados.

[Yu *et al* 2011] consideram eventos de *disrupção regionalizada* no substrato. Tal evento ocorre quando uma única ação compromete múltiplos dispositivos de uma mesma localidade. Para sobreviver a esses eventos, cada nó virtual é replicado em regiões distintas. Além disso, os caminhos entre esses nós sobressalentes não podem passar pelas mesmas regiões dos primários. Essa solução, no entanto, pode restringir bastante o espaço de soluções.

As soluções anteriores fazem uso da reserva de recursos sobressalentes para oferecer resiliência. Tal abordagem pode ser muito custosa ao substrato pois os recursos sobressalentes limitam a alocação de novas redes e ficam ociosos caso não haja disrupções. Em contraste, [Houidi *et al* 2010] dispensam recursos sobressalentes e propõem o uso de multiagentes para lidar com *negação de serviço em nós e enlaces físicos*. Na proposta, os roteadores físicos devem implementar um agente que se comunica com os demais para realocar nós e enlaces virtuais de dispositivos sob ataque. Apesar de não alocar recursos sobressalentes, a estratégia precisa recalcular novos caminhos para os enlaces virtuais afetados. Essa estratégia demanda um período de convergência e pode deixar as redes virtuais inoperantes durante o mesmo.

A proposta apresentada neste artigo difere das anteriores pelas seguintes razões: i) ela não considera a alocação de recursos sobressalentes; ii) a recuperação de capacidade dos enlaces virtuais não necessita recalcular os caminhos fim-a-fim, evitando o custo de realizar esta computação na recuperação; iii) as restrições de resiliência são relaxadas de forma a oferecer uma estratégia menos custosa ao provedor de infraestrutura.

## 3. Definições Preliminares

### 3.1. Modelo de Ameaça

A principal ameaça considerada neste artigo é a disrupção de comunicação entre os nós das redes virtuais. Essa disrupção pode ocorrer por meio de ataques de negação de serviço nos roteadores ou enlaces da rede física.

**Ataques a roteadores e enlaces físicos.** Um ataque de negação de serviço pode ser feito por meio do acesso físico ao dispositivo, ou exploração de alguma vulnerabilidade. No primeiro caso, um atacante obtém a localização de uma fibra óptica ou roteador e interrompe seu funcionamento (p.ex., cortando a fibra ou desligando a energia). No segundo caso, o atacante explora alguma vulnerabilidade no protocolo ou *software* de controle para comprometer algum dispositivo.

**Premissas.** O escopo deste trabalho é delimitado pelas seguintes premissas:

- Considera-se que um atacante pode obter informação sobre a utilização de determinado dispositivo. Dessa forma, o ataque tem como alvo o nó ou enlace com maior utilização, de forma a causar o maior impacto.
- Há isolamento entre as redes virtuais. Portanto, uma rede virtual não pode interferir no funcionamento de outra ao tentar consumir mais recursos do que requisitou.
- A disrupção de um nó físico pode ser reduzida a disrupções em múltiplos enlaces físicos do substrato.

 ©2012 SBC — Soc. Bras. de Computação

### 3.2. Alocação de Redes Virtuais

A alocação de redes virtuais consiste em alocar nós e enlaces virtuais em nós e caminhos do substrato físico, respectivamente. Cada nó virtual é alocado em um nó físico distinto e cada enlace virtual é alocado em *um caminho* (ou *subconjunto dos caminhos*) do substrato. Ademais, as requisições de redes virtuais são enviadas ao substrato físico *sob-demanda*, ou seja, não é possível determinar com antecedência quais redes virtuais deverão ser alocadas. Essas e as demais definições estão em linha com trabalhos anteriores de alocação [Chowdhury *et al* 2009, Cheng *et al* 2012].

**Substrato.** Representado por um grafo dirigido $G^S(N^S, E^S, A^S)$, onde $N^S$ é o conjunto de nós, $E^S$ o conjunto de enlaces e $A^S$ o conjunto de atributos dos nós e enlaces. Neste artigo, são considerados os seguintes atributos: *capacidade computacional* dos nós, *largura de banda* e *atraso de propagação* dos enlaces.

**Requisição de rede virtual.** Cada requisição é definida por um grafo dirigido $G^V(N^V, E^V, A^V)$, onde $N^V$, $E^V$ e $A^V$ são conjuntos similares aos supracitados.

### 3.3. Atraso Diferencial

A divisão do tráfego de dados nem sempre pode ser realizada por caminhos com mesmo atraso fim-a-fim. Particularmente, a utilização de caminhos com diferenças de atraso pode intensificar a entrega de pacotes fora de ordem. Esse problema (definido como *problema do atraso diferencial*) acaba impactando o funcionamento geral dos protocolos de entrega confiável (p.ex., TCP). De maneira geral, pode-se realizar a multiplexação de três formas [He and Rexford 2008]: por fluxo, por rajadas (*flowlets*) ou por pacotes. A primeira forma possui baixa granularidade, visto que cada fluxo está limitado à capacidade de um único caminho. A segunda permite que o mesmo fluxo modifique o caminho de tempos em tempos, o que pode ser ineficiente em fluxos que possuam comportamento contínuo. A terceira alternativa possui a melhor granularidade, permitindo que os fluxos agreguem toda a capacidade dos caminhos.

Neste trabalho optou-se por esta última alternativa, a qual permite aos enlaces das redes virtuais utilizar toda a capacidade requisitada. Apesar dessa grande vantagem, a multiplexação por pacotes pode impor um custo adicional ao destino, seja aos roteadores de borda ou aos hospedeiros. Tal custo, em memória e processamento, é diretamente proporcional às diferenças no atraso entre pacotes. Portanto, de forma similar a trabalhos anteriores [Zhang *et al* 2010], define-se como subobjetivo de otimização a minimização do atraso diferencial.

### 3.4. Objetivos de Otimização

A formulação apresentada neste trabalho possui por objetivo oferecer resiliência às redes virtuais, levando em consideração o custo gerado ao substrato físico. Para isso, o modelo proposto considera três fatores principais.

**Priorização por multi-caminhos disjuntos.** A alocação em múltiplos caminhos permite que um enlace virtual possa resistir a uma disrupção na rede física. No entanto, tal característica só é eficiente se os caminhos selecionados são suficientemente diferentes, caso contrário, uma disrupção em um enlace físico poderia afetar a maior parte dos caminhos utilizados por determinado enlace virtual. No pior caso, uma disrupção pode afetar completamente um ou mais enlaces virtuais. Para lidar com esse problema, define-se a função $\Xi = \sum_{\forall e^S \in E^S} \sum_{\forall e^V \in E^V} \xi_{e^S, e^V}$, a qual descreve a penalidade total dada pela similaridade entre os caminhos.

 ©2012 SBC — Soc. Bras. de Computação

**Balanceamento de carga.** Conforme mencionado anteriormente, a alocação de redes virtuais é feita sob-demanda. Com isso, a alocação a longo prazo pode causar a saturação de um subconjunto dos enlaces da rede física. Essa saturação, por sua vez, pode acabar gerando rejeição de novas requisições que possuam preferência por determinado (subconjunto de) enlace(s) físico(s). O balanceamento de carga é utilizado para evitar essa saturação de enlaces físicos. Para isso, define-se a função $\Phi = \sum_{\forall e^S \in E^S} \phi_{e^S}$, expressando a penalização pela saturação do substrato.

**Minimização do atraso diferencial.** Supondo $P_{e^V}$ o conjunto de caminhos selecionados para alocar determinado enlace virtual, $D_p$ como o atraso no caminho $p$ e $\delta_{e^V} = \max_{\forall p,q \in P_{e^V}}(\mid D_p - D_q \mid)$ como a diferença entre o maior e menor atraso no conjunto de caminhos de um enlace virtual. A função $\Delta = \sum_{\forall e^V \in E^V} \delta_{e^V}$ penaliza o atraso diferencial acumulado de todos os enlaces virtuais.

## 4. Estratégias para Mitigar a Disrupção na Rede Física

Nesta seção são definidas as duas estratégias, preventiva e reativa, para oferecer resiliência contra ataques de negação de serviço às redes virtuais. A primeira estratégia consiste no mapeamento dos enlaces virtuais em múltiplos caminhos, de forma que o comprometimento de um ou mais enlaces físicos não afete completamente os enlaces virtuais. A segunda estratégia é utilizada quando um ataque obtém sucesso em derrubar um enlace. Essa estratégia tenta realocar a capacidade perdida nos caminhos que não foram afetados.

### 4.1. Mapeamento de Redes Virtuais considerando Resiliência

A estratégia utilizada para prover resiliência visa mapear cada enlace virtual das requisições em conjuntos de caminhos físicos, tal que: $(i)$ toda a capacidade do enlace virtual seja distribuída por esses caminhos e $(ii)$ tais caminhos tenham pouca similaridade entre si. A formulação da estratégia é dada conforme segue:

**Variáveis:**

- $\mathcal{X}_{n^S,n^V} \in \mathbb{B}$: indica que o nó $n^S \in N^S$ do substrato físico está sendo utilizado para mapear o nó virtual $n^V \in N^V$.
- $\mathcal{P}_{p,e^S,e^V} \in \mathbb{B}$: indica que o caminho $p$ utiliza o enlace do substrato $e^S \in E^S$ para mapear o enlace virtual $e^V \in E^V$.
- $\mathcal{F}_{p,e^S,e^V} \in \mathbb{R}$ no intervalo $[0;1]$: indica a parcela do fluxo do enlace virtual $e^V \in E^V$ a ser alocada no enlace físico $e^S$ para o caminho $p$.

**Entrada:**

- $c(.) \in A^*$: poder computacional de determinado nó (físico ou virtual).
- $b(.) \in A^*$: largura de banda de determinado enlace (físico ou virtual).
- $d(.) \in A^S$: atraso de determinado enlace físico.
- $P, |P|$: representam, respectivamente, o conjunto de índices dos caminhos e número máximo de caminhos por enlace virtual.

**Objetivo:** *minimizar o impacto causado pelos três fatores do problema*,

$$\min Z = w_\Xi \cdot \Xi + w_\Phi \cdot \Phi + w_\Delta \cdot \Delta \tag{1}$$

onde $w_\Phi$, $w_\Xi$ e $w_\Delta$ são pesos que representam a importância relativa de cada parte da função objetivo. A definição das funções de penalização que compõem $\Xi$, $\Phi$ e $\Delta$ serão apresentadas após as restrições do modelo.

 ©2012 SBC — Soc. Bras. de Computação

**Restrições:**

$$\sum_{\forall n^S \in N^S} \mathcal{X}_{n^S,n^V} = 1 \qquad \forall n^V \in N^V \tag{2}$$

$$\sum_{\forall n^V \in N^V} \mathcal{X}_{n^S,n^V} \leq 1 \qquad \forall n^S \in N^S \tag{3}$$

$$\sum_{\forall n^V \in N^V} \mathcal{X}_{n^S,n^V} \cdot c(n^V) \leq c(n^S) \qquad \forall n^S \in N^S \tag{4}$$

$$\sum_{\forall p \in P} \sum_{\forall e^V \in E^V} \mathcal{F}_{p,e^S,e^V} \cdot b(e^V) \leq b(e^S) \qquad \forall e^S \in E^S \tag{5}$$

$$\sum_{\forall b^S \in N^S:(a^S,b^S) \in E^S} \mathcal{P}_{p,(a^S,b^S),e^V} - \sum_{\forall b^S \in N^S:(b^S,a^S) \in E^S} \mathcal{P}_{p,(b^S,a^S),e^V} = \mathcal{X}_{a^S,s^V} - \mathcal{X}_{a^S,t^V} \tag{6}$$

$$\forall p \in P, \forall a^S \in N^S, \forall e^V = (s^V, t^V) \in E^V$$

$$\sum_{\forall p \in P} \sum_{\forall b^S \in N^S:(a^S,b^S) \in E^S} \mathcal{F}_{p,(a^S,b^S),e^V} - \sum_{\forall p \in P} \sum_{\forall b^S \in N^S:(a^S,b^S) \in E^S} \mathcal{F}_{p,(b^S,a^S),e^V} = \mathcal{X}_{a^S,s^V} - \mathcal{X}_{a^S,t^V} \tag{7}$$

$$\forall a^S \in N^S, \forall e^V = (s^V, t^V) \in E^V$$

$$\mathcal{F}_{p,e^S,e^V} \leq \mathcal{P}_{p,e^S,e^V} \qquad \forall p \in P, \forall e^S \in E^S, \forall e^V \in E^V \tag{8}$$

As restrições (2) e (3) garantem respectivamente que todo nó virtual será alocado a exatamente um nó físico e que nós virtuais serão alocados em nós físicos diferentes. As restrições (4) e (5) asseguram que as capacidades dos nós e enlaces físicos serão respeitadas. A formação de um caminho fim-a-fim válido é representada pela restrição (6). Por fim, as duas últimas restrições definem a quantidade de banda em cada caminho. Elas asseguram a conservação de fluxo ao longo dos caminhos (7) e que o fluxo só será definido em enlaces físicos selecionados (8).

O conjunto de restrições anterior não previne a formação de laços na rede. Dessa forma, adiciona-se a variável auxiliar $\mathcal{H}_{p,a^S,e^V}$ e a restrição não linear $\mathcal{H}_{p,b^S,e^V} = \mathcal{P}_{p,(a^S,b^S),e^V} \cdot \mathcal{H}_{p,a^S,e^V} + \mathcal{P}_{p,(a^S,b^S),e^V}$. Essa restrição "conta o número de saltos" do roteador origem ao destino de cada caminho. Com isso, mapeamentos com laços serão naturalmente descartados pois não seria possível encontrar solução factível. A linearização da restrição é feita pela substituição da multiplicação de variáveis por uma variável auxiliar $\tau$. Após, com o uso de técnicas padrões é possível atribuir relação entre as mesmas. Esse processo culminou nas seguintes restrições:

$$\mathcal{H}_{p,n^S,e^V} \leq \sum_{\substack{m^S \in N^S: \\ (m^S,n^S) \in E^S}} \sigma \cdot \mathcal{P}_{p,(m^S,n^S),e^V} \qquad \forall p \in P, \forall n^S \in N^S, \forall e^V \in E^V : \sigma \to \infty \tag{9}$$

$$\mathcal{H}_{p,n^S,e^V} = \sum_{\substack{m^S \in N^S: \\ (m^S,n^S) \in E^S}} \left( |E^S| \cdot \tau_{p,(m^S,n^S),e^V} + \mathcal{P}_{p,(m^S,n^S),e^V} \right) \qquad \forall p \in P, \forall n^S \in N^S, \forall e^V \in E^V \tag{10}$$

$$\tau_{p,e^S,e^V} \leq \frac{\mathcal{H}_{p,n^S,e^V}}{|E^S|} \qquad \forall p \in P, \forall e^S = (n^S, m^S) \in E^S, \forall e^V \in E^V \tag{11}$$

$$\tau_{p,e^S,e^V} \leq \mathcal{P}_{p,e^S,e^V} \qquad \forall p \in P, \forall e^S \in E^S, \forall e^V \in E^V \tag{12}$$

$$\tau_{p,e^S,e^V} \geq \frac{\mathcal{H}_{p,n^S,e^V}}{|E^S|} + \mathcal{P}_{p,e^S,e^V} - 1 \qquad \forall p \in P, \forall e^S = (n^S, m^S) \in E^S, \forall e^V \in E^V \tag{13}$$

$$\mathcal{H}_{p,n^S,e^V},\ \tau_{p,e^S,e^V} \geq 0 \qquad \forall p \in P, \forall n^S \in N^S, \forall e^S \in E^S, \forall e^V \in E^V \tag{14}$$

 ©2012 SBC — Soc. Bras. de Computação

**Funções de penalização:** a priorização por caminhos disjuntos é dada pela penalização da similaridade de caminhos. Tal restrição é definida por uma aproximação linear de uma função que cresce exponencialmente com o compartilhamento de enlaces físicos. A expressão generalizada é dada pela seguinte formula:

$$\xi_{e^S,e^V} \geq w_K \cdot \sum_{p \in P} \mathcal{P}_{p,e^S,e^V} - c_K \qquad \forall e^S \in E^S, \forall e^V \in e^V \tag{15}$$

onde a constante $K$ ($\leq |P|$) indica quantos caminhos do enlace virtual $e^V$ utilizam o mesmo enlace físico $e^S$, e $w_K = s^{K-1}$ ($s \geq 2$) indica o peso dessa sobreposição. Por sua vez, $c_K$ é dado pela seguinte equação: $c_K = w_K \cdot (K-1) - [w_{K-1} \cdot (K-1) - c_{K-1}]$, $\forall e^S \in E^S, \forall e^V \in e^V, c_0 = 0, w_0 = 0$. A expressão da Eq. (15) é utilizada para criar uma restrição para cada caminho. Portanto, no caso de 2 caminhos serão criadas duas restrições da seguinte forma:

$$\xi_{e^S,e^V} \geq \sum_{p \in P} \mathcal{P}_{p,e^S,e^V} \qquad \forall e^S \in E^S, \forall e^V \in e^V$$

$$\xi_{e^S,e^V} \geq s^1 \cdot \sum_{p \in P} \mathcal{P}_{p,e^S,e^V} - c_1 \qquad \forall e^S \in E^S, \forall e^V \in e^V$$

Para prover a penalidade na saturação de enlaces ($\phi$) utilizou-se a função clássica de engenharia de tráfego proposta por [Fortz and Thorup 2004]. Tal função define um custo exponencial (com aproximação linear) proporcional à utilização do enlace físico e a capacidade do mesmo.

Por fim, para calcular o atraso diferencial, é preciso encontrar os atrasos mínimo e máximo ao longo dos caminhos. Como a ordem dos caminhos não é importante para a formulação, é possível adicionar a seguinte restrição sem alterar o resultado da otimização:

$$\sum_{\forall e^S \in E^S} \mathcal{P}_{p,e^S,e^V} \cdot d(e^S) \leq \sum_{\forall e^S \in E^S} \mathcal{P}_{p+1,e^S,e^V} \cdot d(e^S) \qquad \forall p \in P = \{1,\ldots,k\} \setminus \{k\}, \forall e^V \in E^V$$

a qual ordena os caminhos selecionados de menor a maior atraso. Logo, o atraso diferencial em um enlace virtual $\delta_{e^V}$ pode ser calculado pela seguinte expressão:

$$\delta_{e^V} = \sum_{\forall e^S \in E^S} \mathcal{P}_{k,e^S,e^V} \cdot d(e^S) - \sum_{\forall e^S \in E^S} \mathcal{P}_{1,e^S,e^V} \cdot d(e^S) \qquad \forall e^V \in E^V : P = \{1,\ldots,k\}$$

## 4.2. Recuperação Oportunística de Carga

Uma contribuição chave do presente trabalho é o que cunhamos de **recuperação oportunística**. Ao invés de reservar recursos sobressalentes, a capacidade perdida após um ataque de DoS bem sucedido é redistribuída sobre a banda disponível nos caminhos remanescentes dos enlaces virtuais afetados (denominados de *caminhos ativos*).

Quando um enlace físico torna-se inacessível, a capacidade dos enlaces virtuais sobrepostos é totalmente ou parcialmente comprometida. Caso determinado enlace virtual seja parcialmente afetado, ele ainda possuirá um conjunto de caminhos ativos no substrato. Dessa forma, é possível utilizar qualquer banda disponível nesses caminhos para tentar recuperar a capacidade do enlace virtual. A formulação da estratégia de recuperação é dada conforme segue:

 ©2012 SBC — Soc. Bras. de Computação

**Variável:**

- $Q_{p,e^V} \in \mathbb{R}$ no intervalo $[0;1]$: indica a taxa de fluxo do enlace virtual $e^V \in \dot{E}^V$ utilizada no caminho $p$. Essa taxa é relativa apenas ao fluxo afetado, visto que o restante do fluxo do enlace virtual não é redistribuído.

**Entrada:**

- $\dot{E}^S \subset E^S$: conjunto de enlaces físicos disponíveis após a disrupção.
- $P_{e^V}$: conjunto de caminhos ativos para cada enlace virtual após a disrupção.
- $P_{p,e^S,e^V} \in \mathbb{B}$: indica que o enlace físico $e^S$ está sendo utilizado pelo caminho ativo $p$ do enlace virtual $e^V$.
- $\dot{E}^V \subset E^V$: indica o conjunto de enlaces virtuais afetados.

**Objetivo:** *maximizar a capacidade de recuperação após uma disrupção.* Dessa forma, a função

$$\max T = \sum_{\forall e^V \in E^V} \sum_{\forall p \in P_{e^V}} Q_{p,e^V} \tag{16}$$

visa realocar todo o tráfego afetado sobre o conjunto de caminhos ativos.

**Restrições:**

$$\sum_{\forall p \in P_{e^V}} Q_{p,e^V} \leq F_{e^V} \qquad \forall e^V \in \dot{E}^V \tag{17}$$

$$\sum_{\forall e^V \in E^V} \sum_{\forall p \in P_{e^V}} P_{p,e^S,e^V} \cdot Q_{p,e^V} \cdot b(e^V) \leq b(e^S) \qquad \forall e^S \in \dot{E}^S \tag{18}$$

A primeira restrição (17) impede que o modelo aloque mais banda do que o necessário (ou seja, somente aquela afetada pelo ataque). A segunda restrição (18) assegura que as capacidades de banda dos enlaces físicos serão respeitadas.

## 5. Avaliação

Os principais objetivos das estratégias descritas na seção anterior são (i) prevenir ataques de disrupção, alocando redes virtuais com múltiplos caminhos e de forma eficiente, e (ii) reagir perante ataques, realocando a capacidade comprometida em outros enlaces (quando possível). Esta seção avalia a abordagem proposta através da quantidade de enlaces virtuais afetados por ataques, com e sem proteção, e da quantidade de requisições aceitas na alocação determinada pela abordagem. Além disso, avalia-se o tempo necessário para realizar a alocação ótima de redes virtuais.

A formulação matemática descrita nas seções anteriores foi implementada no software CPLEX 12.3. Esse software utiliza variações do algoritmo Simplex e da técnica de *Branch-and-Bound* para encontrar a solução ótima dentro de determinada margem de erro (ou distância para a solução ótima). Todos os experimentos foram executados em um Intel Core i7 com 8 núcleos de 2,93 GHz e 8GB de memória ram. As condições dos experimentos e seus parâmetros são descritos a seguir.

### 5.1. Configuração dos Experimentos

Durante os experimentos o tempo foi discretizado, organizado em rodadas. Cada rodada pode receber uma ou mais requisições e executar o algoritmo de alocação para cada uma delas. Ademais, caso ocorra um evento de ataque, o algoritmo de recuperação também é executado. A rodada é encerrada somente após a conclusão de todas as etapas pertinentes à alocação e/ou recuperação.

 ©2012 SBC — Soc. Bras. de Computação

As topologias de rede dos experimentos foram geradas sinteticamente utilizando o *software* BRITE com o modelo de redes Barabási-Albert (BA-2). As configurações das redes e da carga de trabalho são similares às definidas em trabalhos anteriores [Chowdhury *et al* 2009, Cheng *et al* 2012].

**Substrato.** A rede física é composta por 30 nós e 114 enlaces em uma grade de 60x60. As capacidades de processamento dos nós e de largura de banda dos enlaces são uniformemente distribuídas no intervalo [50,100][2]. O atraso dos enlaces é diretamente proporcional à distância entre os nós, conforme gerado pelo BRITE, normalizados para o maior valor (ou seja, em intervalos entre 0 e 1).

**Redes virtuais.** O número de nós de uma requisição de criação de rede virtual é gerado uniformemente entre 2 e 4. Conforme será discutido ao final da Subseção 5.2, o uso de valores assim baixos foi necessário para viabilizar a realização do conjunto de experimentos, visto que a etapa de alocação demanda muito tempo de processamento e memória. A capacidade dos nós e enlaces é uniformemente distribuída nos intervalos [5,20] e [50,100], respectivamente. Essas configurações de capacidade foram escolhidas de modo que um único atributo não seja o principal fator de impacto nos experimentos.

**Cargas de trabalho.** A carga de trabalho é composta por requisições de redes virtuais e por ataques de negação de serviço às mesmas via disrupção da rede física. A chegada de requisições é dada por um processo de Poisson com média de 7 para cada 100 rodadas. A duração de cada rede virtual segue uma distribuição geométrica com média de 1000 rodadas. Por sua vez, os ataques são modelados por um processo de Poisson com média de 1 para cada 100 rodadas. Conforme mencionado anteriormente (Seção 3.1), cada ataque é lançado contra o nó que possui a maior alocação de largura de banda em seus enlaces.

Os experimentos executaram durante 5000 rodadas. Dessa forma, durante cada experimento foram geradas, em média, 350 requisições e 50 ataques. Por fim, a margem de erro do CPLEX foi definida para 1%, de modo que soluções fossem encontradas em tempo factível.

### 5.2. Avaliação dos Resultados

*As estratégias propostas ajudam a evitar o impacto de ataques.* A abordagem proposta divide-se em duas estratégias: mapeamento em múltiplos caminhos (preventiva) e recuperação da capacidade perdida (reativa). A Fig. 1 ilustra o efeito da disrupção da rede física com e sem o uso das estratégias, em ambos os casos tendo os parâmetros $w_{\Xi}$, $w_{\Phi}$ e $w_{\Delta}$ fixados em $1/3$ (de modo que seu somatório seja $1$). Os eixos x e y representam respectivamente o número de caminhos a ser usado entre dois nós e a perda média de largura de banda (proporcional) para cada enlace virtual causada pelo ataque. Intuitivamente, um número maior de caminhos deve tornar a rede mais resiliente, e quanto menor a barra, melhor.

No caso padrão, sem a abordagem proposta, há apenas 1 caminho por enlace virtual. Neste caso, quando ocorre um ataque, toda a banda alocada para os enlaces virtuais afetados é perdida e o mesmo fica completamente inoperante. Com o incremento no número de caminhos é possível utilizar as duas estratégias propostas para

[2]Em linha com trabalhos similares, mantém-se unidade no nível abstrato; a mesma poderia representar Mbps no caso de enlaces ou número de fatias de tempo no caso de nós.

 ©2012 SBC — Soc. Bras. de Computação

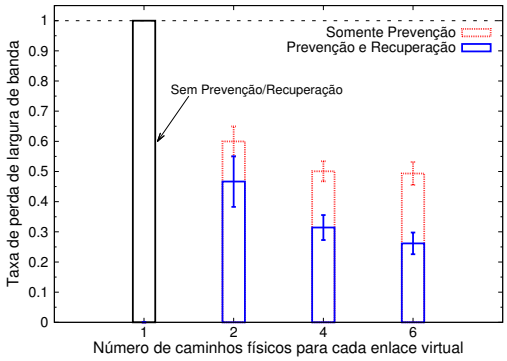



**Figura 1. Perda e recuperação da largura de banda para cenários com 1, 2, 4 e 6 caminhos por enlace virtual. Neste experimento, $w_\Xi = w_\Phi = w_\Delta = 1/3$.**

mitigar o efeito de ataques. O comportamento da estratégia preventiva é representado pelas barras vermelhas (contorno tracejado), enquanto o comportamento das duas estratégias em conjunto é representado pelas barras azuis (contorno contínuo). Considerando o cenário com 6 caminhos por enlace virtual, é possível perceber que a estratégia preventiva consegue proteger, em média, 50% da capacidade do enlace. Por sua vez, a utilização posterior da estratégia reativa para recuperar a capacidade comprometida reduz a perda para aproximadamente 25%. Como esperado, a abordagem proposta tem melhor resultado quando ambas as estratégias são utilizadas. Ademais, apesar da eficiência das estratégias aumentar com o número de caminhos utilizados, observa-se que seu comportamento não é linear. Tal é explicado pelas limitações na quantidade de caminhos disjuntos do substrato conforme evidenciado a seguir.

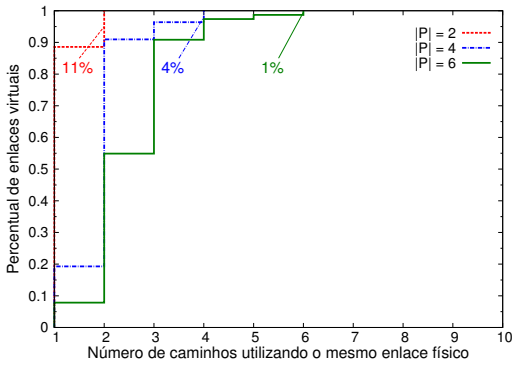
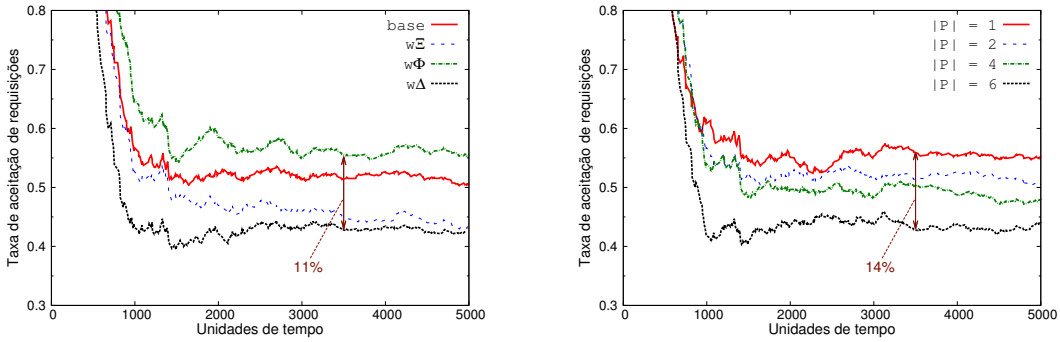



**Figura 2. Nível de compartilhamento de enlaces físicos pelos caminhos alocados aos enlaces virtuais.**

*A alocação em múltiplos caminhos reduz a severidade de ataques a pontos críticos.* Define-se um "ponto crítico" como um recurso, seja ele nó ou caminho, que por uma circunstância da rede física e/ou da alocação, acaba por concentrar toda a capacidade de um enlace virtual. Um ataque a um ponto crítico pode comprometer um ou mais enlaces virtuais por completo. A Função de Distribuição Acumulada (*Cumulative Distribution Function* – CDF) apresentada na Fig. 2 permite avaliar

 ©2012 SBC — Soc. Bras. de Computação

a quantidade de enlaces virtuais suscetíveis a esse tipo de ataque. O eixo x indica quantos caminhos de um mesmo enlace virtual utilizam determinado enlace físico (ou seja, quando for 1 não existe compartilhamento). Considerando dois caminhos por enlace virtual ($|P| = 2$), aproximadamente 11% desses enlaces possuem ambos os caminhos compartilhando o mesmo enlace físico. Ao incrementar o número de caminhos, a quantidade de enlaces virtuais suscetíveis ao ataque cai para 4% ($|P| = 4$) e então para 1% ($|P| = 6$). Por outro lado, é possível perceber que o número de caminhos disjuntos também decresce. A importância de tais caminhos está no fato de que se dois ou mais caminhos compartilham algum enlace físico, a eficiência da recuperação fica limitada à capacidade do mesmo. Quando os enlaces virtuais são mapeados em dois caminhos, 89% deles possuem caminhos disjuntos. Esse número decresce para 7% quando são usados 6 caminhos por enlace virtual. Tal evidência corrobora a afirmação anterior de que a quantidade de caminhos disjuntos no substrato pode afetar o desempenho da abordagem proposta.




(a) Número de caminhos por enlace virtual ($|P|$) fixado em 2

(b) Parâmetros $w_\Xi = w_\Phi = w_\Delta = 1/3$

**Figura 3. Taxa de aceitação de acordo com a variação (a) dos parâmetros $w_\Xi$, $w_\Phi$ e $w_\Delta$ e (b) no número de caminhos por enlace virtual.**

*O custo da solução é diretamente proporcional ao nível de resiliência contra disrupção.* A Fig. 3 ilustra as diferenças na taxa de aceitação de redes virtuais para diferentes parâmetros de entrada ao longo do tempo, com a taxa de aceitação sendo representada no eixo y. Em um primeiro momento ($0 \sim 900$), as requisições de rede virtual encontram um substrato ocioso com ampla disponibilidade de recursos. À medida que os recursos vão ficando escassos ($900 \sim 2000$), a taxa de aceitação diminui. Por fim, o substrato alcança estabilidade ($2000 \sim \infty$) quando a quantidade de recursos liberados pelas redes virtuais extintas é similar à quantidade demandada pelas novas requisições e 40 a 60% são aceitas. A Fig. 3(a) demonstra a diferença na taxa de aceitação causada pela escolha de prioridades distintas na alocação. O caso base (primeira curva da legenda e segunda de cima para baixo) foi utilizado para verificar a importância relativa de cada parâmetro. O aumento na *priorização por caminhos disjuntos* (parâmetro $w\Xi$, segunda curva na legenda e terceira de cima para baixo) provoca uma sobrecarga na alocação de recursos do substrato e diminui a taxa de aceitação. Tal se deve ao algoritmo de alocação, que opta por caminhos maiores para evitar a sobreposição de enlaces físicos. A escolha por *enlaces menos saturados* (parâmetro $w\Phi$) ou pela *minimização do atraso diferencial* (parâmetro $w\Delta$) acarretaram respectivamente na melhor e pior taxas de aceitação (com uma diferença de aproximadamente 11%). O aumento de $w\Phi$ ajuda a evitar a formação de gargalos, deixando o acesso mais livre aos nós da rede. Por outro lado, o aumento

 ©2012 SBC — Soc. Bras. de Computação

de $w\Delta$ prioriza a escolha de caminhos maiores com atrasos similares, ao invés de caminhos menores com maior diferença nos atrasos.

A Fig. 3(b) demonstra a eficiência do algoritmo de alocação ao longo do tempo com a variação no número de caminhos por enlace virtual. Apesar de possibilitar maior granularidade na alocação, o aumento no número de caminhos pode resultar em alocações piores. No pior caso, ao considerar 6 caminhos por enlace virtual, a taxa de aceitação decresce 14%. Isso ocorre porque ao considerar somente um único caminho, os efeitos colaterais causados pela priorização de caminhos disjuntos $\Xi$ e pela minimização do atraso diferencial $\Delta$ são anulados. Dessa forma, o critério dominante na alocação passa a ser o balanceamento de carga $\Phi$. No entanto, quanto maior for a quantidade de caminhos, maior será a importância relativa destes dois outros critérios.

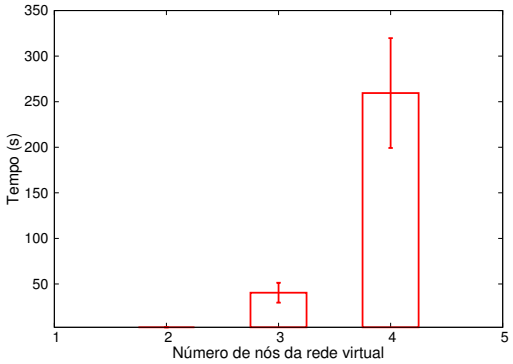



**Figura 4. Tempo médio para alocar cada requisição de rede virtual.**

*A característica exponencial do problema de alocação justifica o estudo de heurísticas.* Diversos trabalhos da literatura estudam heurísticas para lidar com o problema da alocação de redes virtuais [Chowdhury *et al* 2009, Alkmim *et al* 2011, Cheng *et al* 2012]. Sua similaridade com o *multiway separator problem* [Andersen 2002] é um forte indicio de que o problema é NP-Difícil. A Fig. 4 demonstra o tempo médio para a execução do algoritmo de alocação de redes sobre uma única requisição. Conforme pode ser observado, o problema de alocação limita a escalabilidade da solução, demandando em torno de 250s para resolver de forma ótima *uma única instância* de rede virtual com 4 nós na rede física descrita anteriormente. Dessa forma, a evolução natural deste trabalho é a definição de heurísticas para a abordagem proposta.

## 6. Conclusão

Apesar das potenciais vantagens de redes virtuais, como o isolamento, elas ainda estão sujeitas a ataques de negação de serviço lançados à rede física. Este trabalho apresentou uma abordagem inovadora para aumentar a proteção de redes virtuais contra ataques de DoS via rede física. Entre suas contribuições, destaca-se a que a abordagem de alocação proposta, diferentemente dos trabalhos anteriores, não precisa reservar recursos sobressalentes ou recalcular os caminhos utilizados pelos enlaces virtuais. A mesma é dividida em duas estratégias complementares. A primeira estratégia visa prevenir os ataques pelo uso de múltiplos caminhos *prioritariamente* disjuntos. A segunda visa recuperar a capacidade comprometida utilizando os caminhos ativos (ou seja, aqueles não afetados pelo ataque).

 ©2012 SBC — Soc. Bras. de Computação

Conforme discutido na seção de avaliação, a solução proposta oferece um ganho na proteção e recuperação de capacidade, ao custo de uma menor taxa de aceitação. Ademais, esse ganho é incrementalmente menor à medida que a resiliência é priorizada sobre a alocação. Portanto, é possível adequar a solução aos requisitos do ambiente, priorizando a resiliência em detrimento da alocação ou adotando uma estratégia mais conservadora, mas ainda assim mitigar os ataques de disrupção.

Quanto aos trabalhos futuros, vislumbramos a definição de heurísticas para tornar a abordagem mais rápida, possibilitando sua aplicação em cenários maiores. Ademais, estamos investigando novas estratégias para o processo de recuperação, como a migração de nós, a qual possibilitará tornar a rede ainda mais robusta a ataques de negação de serviço.

## Referências


Alkmim *et al* (2011). Optimal mapping of virtual networks. In *GLOBECOM, IEEE*, pages 1–6.

Andersen, D. G. (2002). Theoretical approaches to node assignment. Unpublished manuscript. `http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps`.

Belbekkouche *et al* (2012). Resource discovery and allocation in network virtualization. *Comm. Surv. Tut., IEEE*, PP(99):1–15.

Chen *et al* (2010). Resilient virtual network service provision in network virtualization environments. In *ICPADS, IEEE*, pages 51–58.

Cheng *et al* (2012). Virtual network embedding through topology awareness and optimization. *Comput. Netw.*, 56(6):1797–1813.

Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Comp. Netw.*, 54(5):862–876.

Chowdhury *et al* (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM, IEEE*, pages 783–791.

Fortz, B. and Thorup, M. (2004). Increasing internet capacity using local search. *Comput. Optim. and Applic.*, 29:13–48.

Guo *et al* (2011). Shared backup network provision for virtual network embedding. In *ICC, IEEE*, pages 1–5.

He, J. and Rexford, J. (2008). Toward internet-wide multipath routing. *Network, IEEE*, 22(2):16–21.

Houidi *et al* (2010). Adaptive virtual network provisioning. In *2nd SIGCOMM VISA workshop, ACM*, pages 41–48.

Khan *et al* (2012). Network virtualization: a hypervisor for the internet? *Comm. Mag., IEEE*, 50(1):136–143.

Medhi, D. (2006). Network restoration. In Resende, M. G. C. and Pardalos, P. M., editors, *Handbook of Optimization in Telecomm.*, pages 801–836. Springer US.

Rahman *et al* (2010). Survivable virtual network embedding. In Crovella, M. *et al.*, editor, *NETWORKING 2010*, volume 6091 of *LNCS*, pages 40–52. Springer Berlin / Heidelberg.

Yeow *et al* (2010). Designing and embedding reliable virtual infrastructures. In *2nd SIGCOMM VISA workshop, ACM*, pages 33–40.

Yu *et al* (2011). Cost efficient design of survivable virtual infrastructure to recover from facility node failures. In *ICC, IEEE*, pages 1–6.

Zhang *et al* (2010). Reliable adaptive multipath provisioning with bandwidth and differential delay constraints. In *Proc. INFOCOM, IEEE*, pages 2178–2186.


 ©2012 SBC — Soc. Bras. de Computação

# Um Modelo para Mapeamento Ótimo de Redes Virtuais com Requisitos de Segurança

**Leonardo Richter Bays[1], Rodrigo Ruas Oliveira[1], Luciana Salete Buriol[1], Marinho Pilla Barcellos[1], Luciano Paschoal Gaspary[1]**

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{lrbays,ruas.oliveira,buriol,marinho,paschoal}@inf.ufrgs.br

***Abstract.*** *Network virtualization enables the creation of multiple instances of virtual networks on top of a single physical infrastructure. Given its wide applicability, this technique has attracted a lot of interest both from academic researchers and major companies within the segment of computer networks. Although recent efforts (motivated mainly by the search for mechanisms to enable the evaluation of Future Internet proposals) have contributed substantially to materialize this concept, none of them has attempted to combine efficient resource allocation with fulfillment of security requirements (e.g., confidentiality). It is important to note that, in the context of virtual networks, the protection of shared network infrastructures constitutes a fundamental condition to enable its use in large scale. To address this problem, in this paper we propose a virtual network embedding model that aims to provide the desired level of security while optimizing physical resource usage. The results obtained demonstrate that the model is able to correctly and optimally map virtual networks to a physical substrate, minimizing bandwidth costs for infrastructure providers.*

***Resumo.*** *A virtualização de redes permite a criação de múltiplas instâncias de redes virtuais sobre uma única infraestrutura física. Devido à sua ampla aplicabilidade, tal técnica tem atraído grande interesse tanto de pesquisadores quanto de empresas importantes do segmento de redes de computadores. Apesar de esforços recentes (motivados principalmente pela busca de mecanismos para viabilizar a avaliação de propostas na temática Internet do Futuro) terem contribuído substancialmente para a materialização do conceito, nenhum preocupou-se em conciliar alocação eficiente de recursos e satisfação de requisitos de segurança (ex: confidencialidade). Ressalta-se que, no contexto de redes virtuais, a proteção de infraestruturas de rede compartilhadas constitui condição fundamental para seu uso em larga escala. Para abordar o referido problema, neste artigo propõe-se um modelo de alocação de redes virtuais que busca satisfazer o nível especificado de segurança e, ao mesmo tempo, otimizar a utilização dos recursos físicos. Os resultados obtidos demonstram que o modelo é capaz de alocar redes virtuais a um substrato físico de forma correta e ótima, minimizando custos de largura de banda para provedores de infraestrutura.*

## 1. Introdução

Nos últimos anos, têm surgido demandas cada vez maiores por serviços de rede específicos, com requisitos peculiares e distintos. Motivados por tais demandas, e estimulados pelo sucesso no emprego de virtualização para hospedagem de servidores personalizados, pesquisadores passaram a explorar o uso dessa técnica em infraestruturas de

 ©2012 SBC — Soc. Bras. de Computação

rede. A virtualização de redes permite a criação de múltiplas topologias virtuais sobre um mesmo substrato físico. Isso é possível por meio da instanciação de um ou mais roteadores virtuais em dispositivos físicos e do estabelecimento de enlaces virtuais entre esses roteadores, formando topologias arbitrárias.

Entre outras vantagens, o uso de virtualização de redes permite a um provedor de infraestrutura acomodar simultaneamente múltiplas pilhas de protocolo no mesmo substrato. Isso possibilita a criação de infraestruturas de rede adaptadas às necessidades de aplicações de rede específicas [Fernandes et al. 2010]. Ademais, essa técnica pode ser usada para a execução de experimentos sem interferir com tráfego de produção, em escala e com um alto grau de similaridade com infraestruturas reais. Dessa forma, é possível criar ambientes favoráveis ao desenvolvimento e avaliação de novas arquiteturas e protocolos, o que pode contribuir para o avanço de pesquisas relacionadas à Internet do Futuro [Anderson et al. 2005].

A virtualização de redes também tem recebido grande apoio no mercado. Empresas importantes passaram a oferecer dispositivos com suporte nativo à virtualização. Essa nova funcionalidade permite que provedores de infraestrutura passem também a oferecer novos serviços. O suporte de grandes nomes da indústria a esse tipo de iniciativa pode ser observado, por exemplo, na lista de membros da *Open Networking Foundation*[1], que promove o desenvolvimento e o uso de redes virtualizadas definidas por software.

Apesar de sua ampla aplicabilidade, manter um ambiente de virtualização de redes requer uma distribuição adequada dos recursos. Por um lado, há provedores de infraestrutura, que desejam obter o máximo de lucro hospedando a maior quantidade possível de redes virtuais, minimizando seus custos. Por outro, há uma série de clientes que solicitam redes virtuais com demandas de recursos específicas. O método de alocação deve garantir que os recursos requisitados estarão disponíveis para esses clientes e, ao mesmo tempo, minimizar os custos do provedor de infraestrutura. Além disso, o resultado do processo de mapeamento deve ser entregue em um tempo aceitável.

Outra grande preocupação que surge com o uso compartilhado de dispositivos de roteamento e canais de comunicação é a segurança. Sem a proteção adequada, é possível que usuários de uma rede virtual capturem ou até mesmo adulterem dados de outras redes virtuais no mesmo substrato. Tais ações violariam propriedades de segurança tais como confidencialidade e integridade. Portanto, é de grande importância que arquiteturas de virtualização ofereçam proteção contra essas e outras ameaças que possam comprometer sua segurança.

Para viabilizar o uso de virtualização em ambientes reais, tanto alocação eficiente de recursos quanto segurança devem ser levados em consideração. O problema da alocação de recursos é considerado *NP-hard* devido a sua similaridade com o *multi-way separator problem* [Andersen 2002], e tem sido geralmente abordado na literatura com algoritmos de alocação modelados por meio de programação linear. Há uma série de trabalhos focando no problema da alocação de recursos de redes virtuais [Yu et al. 2008, Chowdhury et al. 2009, Alkmim et al. 2011, Cheng et al. 2011, Davy et al. 2011]. No entanto, os autores deste artigo desconhecem propostas que levam em consideração requisitos de segurança. Para preencher essa lacuna, no presente artigo é proposto um modelo de alocação de redes virtuais que otimiza a utilização de recursos físicos ao mesmo tempo em que atende requisitos de segurança. Além de requisitos de capacidade e

[1]http://www.opennetworking.org/membership

 ©2012 SBC — Soc. Bras. de Computação

localização, solicitantes de redes virtuais podem especificar requisitos de segurança para suas redes, que devem ser atendidos pelo provedor de infraestrutura. O modelo proposto recebe requisições de forma *on-line* e determina a melhor alocação possível em termos de utilização de recursos, considerando todos os requisitos de segurança dos solicitantes.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados às áreas de mapeamento de recursos e segurança em redes virtuais. Na seção 3 é descrito o modelo proposto e sua formulação. A Seção 4 relata a avaliação realizada e apresenta os resultados obtidos. Por fim, na Seção 5 são apresentadas as considerações finais e perspectivas para trabalhos futuros.

## 2. Trabalhos Relacionados

Nessa seção, serão apresentados os trabalhos relacionados focando no mapeamento de redes virtuais, bem como algumas das principais propostas visando prover segurança a redes virtuais. Cada trabalho será descrito brevemente, ressaltando suas principais características.

Yu et al. [Yu et al. 2008] propõem um modelo de alocação de redes virtuais com suporte à separação de caminhos (fragmentação de enlaces virtuais por múltiplos caminhos do substrato físico) e migração. O modelo proposto aproveita-se do ganho em flexibilidade obtido pela separação de enlaces virtuais em múltiplos caminhos (caso tal separação seja permitida pelo solicitante), reduzindo o tempo necessário para completar o processo de mapeamento. Além disso, o substrato é capaz de reotimizar sua utilização de recursos periodicamente por meio da migração de roteadores e enlaces virtuais previamente alocados. O modelo considera que requisições de redes virtuais não são conhecidas *a priori*, e leva em consideração requisitos de CPU e largura de banda, bem como o tempo máximo que uma requisição pode aguardar antes de ser atendida.

Outro modelo, formulado por Chowdhury et al. [Chowdhury et al. 2009], visa aprimorar a coordenação entre a alocação de roteadores e enlaces, que é realizada em duas fases separadas. Isso é feito por meio da pré-seleção de alocações de roteadores de forma a auxiliar o estágio de mapeamento de enlaces. Assim como o modelo previamente mencionado, esse também permite a separação de caminhos, e considera requisitos de CPU e largura de banda. Requisições de redes virtuais são alocadas de forma *on-line*, e podem especificar as localizações físicas em que certos roteadores virtuais devem ser mapeados.

O modelo desenvolvido por Alkmim et al. [Alkmim et al. 2011] estende os trabalhos anteriores combinando requisitos de capacidade com restrições relacionadas à transferência de imagens utilizadas nos roteadores virtuais. O modelo visa minimizar o tempo necessário para transferir tais imagens, ao mesmo tempo que considera requisitos de CPU, memória, banda e localização. Como nos trabalhos anteriores, requisições são recebidas e alocadas de forma *on-line*.

Cheng et al. [Cheng et al. 2011] apresentam uma abordagem baseada na classificação de nós (*node ranking*), que considera tanto a capacidade de roteadores e enlaces quanto as capacidades de seus vizinhos. Por exemplo, a classificação de roteadores físicos é afetada não só por sua própria capacidade, mas também pela capacidade disponível em outros roteadores conectados ao mesmo. De forma análoga, a classificação de roteadores e enlaces virtuais também leva em consideração as características de seus vizinhos. O processo de mapeamento aloca roteadores e enlaces virtuais a elementos da

 ©2012 SBC — Soc. Bras. de Computação

rede física com classificações similares. Tal estratégia, segundo os autores, tende a reduzir possíveis gargalos criados pela alocação de redes virtuais.

Ao contrário das propostas apresentadas anteriormente, o modelo proposto por Davy et al. [Davy et al. 2011] não recebe como entrada uma topologia de rede completa. Em vez disso, requisições contêm apenas os *end points* que devem ser interconectados (uma origem e um ou mais destinos). A solução constrói redes virtuais em forma de árvore, partindo da localização de origem até os destinos requisitados. Além de restrições de localização, o modelo também considera a preferência dos solicitantes por redes de baixo custo, ou redes com menor atraso e custo maior. As redes virtuais são então instanciadas, obedecendo os requisitos previamente mencionados e visando minimizar os custos do provedor de infraestrutura.

Apresentados e discutidos os principais trabalhos relacionados à alocação de recursos, passa-se, agora, a uma síntese das investigações que buscam oferecer segurança em redes virtualizadas. Cabuk et al. [Cabuk et al. 2007] apresentam um arcabouço para criação de redes virtuais seguras. Os autores utilizam "Domínios Virtuais Confiáveis" (*Trusted Virtual Domains* – TVDs) para prover controle de acesso, confidencialidade e integridade a comunicações de rede. Cada TVD representa um domínio isolado, composto de entidades virtuais e de enlaces entre as mesmas. Certificados digitais são usados para assegurar que somente entidades que satisfaçam um determinado conjunto de condições sejam capazes de participar de um TVD. Os autores usam VLANs para isolar o tráfego dentro de cada rede confiável, e VPNs para interconectar tais redes.

Huang et al. [Huang et al. 2010] propõem um método que lança mão de criptografia para proteger informações de roteamento, bem como caminhos variáveis para mitigar ataques de análise de tráfego. Tal método classifica roteadores em diferentes grupos, e distribui chaves de grupo para cada roteador. Dessa forma, somente roteadores pertencentes a um determinado grupo são capazes de acessar as informações protegidas de tal grupo. Além disso, cada enlace virtual é mapeado a múltiplos caminhos físicos. Fluxos transmitidos por tais enlaces são divididos aleatoriamente entre os caminhos físicos disponíveis, visando evitar análise de tráfego.

Ao mesmo tempo em que há trabalhos focando no problema de alocação de redes virtuais, há outros visando oferecer serviços de segurança a ambientes de redes virtuais. No entanto, os autores deste artigo desconhecem trabalhos que abordem ambas as áreas simultaneamente. Ao não considerarem segurança, um aspecto essencial em ambientes de virtualização de redes devido ao compartilhamento de recursos físicos, os trabalhos anteriores na área de alocação de redes virtuais acabam por subestimar a quantidade de recursos necessária para acomodar tais redes. Nesse contexto, busca-se, neste artigo, suprir tal lacuna, ao propor uma solução que concilia alocação eficiente de recursos com satisfação de requisitos de segurança, fatores fundamentais para ampla adoção de redes virtuais em ambientes de produção.

## 3. Modelo Proposto

Para abordar o problema da otimização do uso de recursos considerando requisitos de segurança, foi desenvolvido um modelo por meio de Programação Linear Inteira. Para criar um modelo que represente o cenário de alocação de redes virtuais com um nível desejável de fidelidade, diversos detalhes foram levados em consideração. Vislumbra-se um cenário em que um provedor de infraestrutura fornece redes virtuais a diversos clientes. Para solicitar a criação de uma rede virtual, clientes devem firmar um Acordo de

 ©2012 SBC — Soc. Bras. de Computação

Nível de Serviço (*Service Level Agreement* – SLA) com o provedor de infraestrutura. Tal SLA descreve as características da rede virtual solicitada e seus requisitos de segurança, que devem ser atendidos pelo provedor.

Assume-se que um provedor de infraestrutura receberá uma série de requisições de redes virtuais ao longo do tempo. Portanto, tais requisições devem ser tratadas de forma *on-line*, isto é, uma a uma conforme são recebidas. Caso haja recursos suficientes no substrato para que a alocação seja possível, a saída do modelo deve indicar a melhor alocação em termos de utilização de recursos, maximizando os recursos disponíveis para futuras requisições. Caso não haja recursos suficientes para alocar uma rede virtual, a requisição é negada.

Antes de apresentar o modelo proposto, será descrita a sintaxe usada na sua formulação. Letras maiúsculas são usadas para representar conjuntos ou variáveis. Letras sobrescritas indicam se um conjunto refere-se a recursos virtuais ($V$) ou físicos ($P$), ou se o mesmo se refere a roteadores ($R$) ou enlaces ($L$). Ademais, letras subscritas representam índices associados a variáveis.

**Topologias.** Requisições de redes virtuais devem especificar a topologia desejada, isto é, o número de roteadores virtuais na rede e as interconexões entre os mesmos. Cada topologia de rede virtual, bem como a topologia da rede física, é representada como um grafo direcionado $N = (R, L)$, no qual os vértices $R$ e as arestas $L$ representam, respectivamente, roteadores e enlaces. Além disso, um enlace entre dois roteadores $a$ e $b$ é representado por um par de arestas simétricas com direções opostas, $(a, b)$ e $(b, a)$. Roteadores virtuais são mapeados a exatamente um roteador físico, enquanto que enlaces virtuais podem ser mapeados a um único enlace físico ou a um caminho composto por dois ou mais enlaces físicos.

**Capacidades físicas e virtuais.** Roteadores físicos possuem capacidades limitadas de CPU e memória, expressas por $C_i^P$ e $M_i^P$, respectivamente (em que $i$ é o índice do roteador). Por sua vez, enlaces possuem capacidade de banda limitada $B_{i,j}^P$, em que o par $(i, j)$ representa um enlace físico entre $i$ e $j$. De forma similar, $C_{r,i}^V$ e $M_{r,i}^V$ representam os requisitos de CPU e memória de um roteador virtual de uma rede $r$. Além disso, $B_{r,i,j}^V$ representa o requisito de largura de banda de um enlace virtual entre os roteadores virtuais $i$ e $j$ de uma rede $r$. Os requisitos desses elementos virtuais definem a parcela dos recursos físicos que deve ser alocada para seu consumo. Presume-se que a arquitetura de virtualização é capaz de isolar adequadamente os recursos físicos, garantindo o cumprimento desses limites.

**Localidades.** Assume-se que a maioria dos clientes requisitará redes virtuais fixando um ou mais pontos onde roteadores virtuais deverão ser hospedados. Portanto, cada roteador físico está associado a um identificador de localização $S^P$, e requisições de redes virtuais podem ou não requerer que alguns de seus roteadores sejam mapeados a roteadores físicos em localidades específicas. Roteadores virtuais com requisitos de localidade são armazenados no conjunto $S^V$.

**Segurança.** O modelo também permite que cada requisição de rede virtual possua requisitos de segurança associados. O oferecimento de serviços de confidencialidade visa tratar preocupações relacionadas ao uso compartilhado de roteadores físicos e canais de comunicação, o que pode fazer com que dados sensíveis sejam expostos a terceiros. Tais requisitos, se presentes, indicam um de três níveis distintos de confidencialidade que devem ser fornecidos às comunicações dessas redes:

 ©2012 SBC — Soc. Bras. de Computação

- Criptografia fim-a-fim: caso esse nível de confidencialidade seja solicitado, os roteadores de borda da rede virtual devem ser hospedados em roteadores físicos que suportam tal característica. Na prática isso significa que esses roteadores físicos devem dar suporte a suítes de protocolos tais como IPSec [Kent and Seo 2005], que fornece criptografia fim-a-fim quando usado em *modo de transporte*.
- Criptografia ponto-a-ponto: nesse nível de confidencialidade, pacotes inteiros são criptografados, protegendo não só os dados contidos nos mesmos mas também seu cabeçalho. Isso significa que pacotes precisam ser decriptografados e recriptografados em cada salto para serem roteados adequadamente. Portanto, cada roteador em uma rede virtual que requer esse nível de confidencialidade deve ser mapeado a um roteador físico capaz de dar suporte a tais operações. Esse nível corresponde ao *modo de túnel* do IPSec, o que significa que roteadores físicos com suporte a esse protocolo são capazes de prover tal característica.
- Não-sobreposição de redes: uma requisição de rede virtual pode também exigir que seus roteadores e enlaces virtuais não compartilhem roteadores nem caminhos físicos com uma ou mais redes virtuais. Tal caso extremo pode ser usado, por exemplo, para proteger informações altamente sigilosas de empresas concorrentes.

Para prover os dois primeiros níveis de segurança, requisições de redes virtuais devem ser capazes de indicar quais de seus roteadores devem ser capazes de criptografar e decriptografar pacotes de rede, caso desejado. Portanto, o modelo também incorpora os conjuntos $K_i^P$ e $K_{r,i}^V$, que indicam se um roteador físico é capaz de oferecer tal característica, e se um roteador virtual a requer.

Já quanto ao terceiro nível, requisições devem ser capazes de especificar o conjunto de redes virtuais com o qual roteadores e enlaces físicos não serão compartilhados. Para oferecer tal nível, é usado o conjunto $X$. Esse conjunto é composto por pares de redes virtuais que não devem compartilhar recursos do substrato (por exemplo, se $(i, j) \in X$, não é permitido que as redes virtuais $i$ e $j$ compartilhem recursos).

**Alocação prévia.** Por fim, os conjuntos $E_{i,r,j}^R$ e $E_{i,j,r,k,l}^L$ indicam onde encontram-se alocados, respectivamente, os roteadores e enlaces das redes virtuais já alocadas no substrato. Caso não haja nenhuma rede virtual alocada no momento em que uma requisição é recebida, tais conjuntos estarão vazios. A seguir, para maior clareza, apresenta-se um sumário das entradas do modelo proposto.

- $N^P = \{R^P, L^P\}$ – Representa a rede física, composta por um conjunto de roteadores físicos $R^P$ e um conjunto de enlaces físicos $L^P$.
- $N^V = \{R^V, L^V\}$ – Representa uma requisição de rede virtual, composta por um conjunto de roteadores virtuais $R^V$ e um conjunto de enlaces virtuais $L^V$.
- $X \in N^V \times N^V$ – Conjunto de redes virtuais conflitantes. Representa redes virtuais que não devem ser mapeadas aos mesmos elementos do substrato físico.
- $S \in \mathbb{N}$ – Conjunto de todas as possíveis localidades físicas onde roteadores físicos podem residir, representadas por números naturais.
- $S^P \in R^P \times S$ – Indica a localização de roteadores da rede física.
- $S^V \in R^V \times S$ – Indica requisitos de localização de roteadores em requisições de redes virtuais.
- $C_i^P \in \mathbb{N}$ – Indica a capacidade total de CPU de um roteador físico $i$.
- $M_i^P \in \mathbb{N}$ – Indica a capacidade total de memória de um roteador físico $i$.
- $B_{i,j}^P \in \mathbb{N}$ – Indica a largura de banda de um enlace físico $(i, j)$.

 ©2012 SBC — Soc. Bras. de Computação

- $K_i^P \in \{0,1\}$ – Indica se um roteador físico $i$ suporta protocolos que o permitam criptografar e decriptografar pacotes de rede. Se um roteador físico é capaz de dar suporte a tais protocolos, o valor é definido como 1; caso contrário, é definido como 0.
- $C_{r,i}^V \in \mathbb{N}$ – Indica a capacidade de CPU exigida por um roteador virtual $i$ de uma rede virtual $r$.
- $M_{r,i}^V \in \mathbb{N}$ – Indica a capacidade de memória necessária a um roteador virtual $i$ de uma rede virtual $r$.
- $B_{r,i,j}^V \in \mathbb{N}$ – Indica a largura de banda exigida por um enlace virtual $(i,j)$ de uma rede virtual $r$.
- $K_{r,i}^V \in \{0,1\}$ – Indica se um roteador virtual $i$ de uma rede virtual $r$ deve ser capaz de criptografar e decriptografar pacotes de rede. Se um roteador virtual requer tal característica, o valor é definido como 1; caso contrário, é definido como 0.
- $E_{i,r,j}^R \in \{0,1\}$ – Indica se um roteador virtual $j$ de uma rede virtual $r$ previamente recebida encontra-se alocado no roteador físico $i$. Em caso positivo, assume o valor 1; caso contrário, assume o valor 0.
- $E_{i,j,r,k,l}^L \in \{0,1\}$ – Indica se um enlace virtual $(k,l)$ de uma rede virtual $r$ previamente recebida encontra-se alocado no enlace físico $(i,j)$. Em caso positivo, assume o valor 1; caso contrário, assume o valor 0.

De forma similar, as variáveis de saída do modelo proposto são apresentadas a seguir. Os valores retornados por tais variáveis indicam a alocação de elementos virtuais no substrato físico, representando a solução do problema. Uma vez que o problema é solucionado, cada roteador virtual estará mapeado a um único roteador físico, e cada enlace virtual estará mapeado a um caminho no substrato físico. Tal caminho pode ser um único enlace físico, ou uma série de enlaces físicos consecutivos.

- $A_{i,r,j}^R \in \{0,1\}$ – Alocação de roteadores, indica se o roteador físico $i$ está hospedando o roteador virtual $j$ da rede virtual $r$.
- $A_{i,j,r,k,l}^L \in \{0,1\}$ – Alocação de enlaces, indica se o enlace físico $(i,j)$ está hospedando o enlace virtual $(k,l)$ da rede virtual $r$.

Por fim, é apresentada a função objetivo do modelo e suas restrições. A função objetivo visa minimizar a largura de banda física consumida pelos enlaces virtuais nas redes solicitadas, dessa forma minimizando custos e preservando largura de banda para alocações futuras. Por sua vez, as restrições garantem que os requisitos serão atendidos, e que as capacidades físicas não serão excedidas.

**Objetivo:**

$$min \sum_{(i,j)\in L^P} \sum_{r\in N^V,(k,l)\in L^V} A_{i,j,r,k,l}^L B_{r,k,l}^V$$

**Sujeito a:**

$$\sum_{r\in N^V, j\in R^V} C_{r,j}^V A_{i,r,j}^R \leq C_i^P \qquad \forall i \in R^P \quad \text{(R1)}$$

$$\sum_{r\in N^V, j\in R^V} M_{r,j}^V A_{i,r,j}^R \leq M_i^P \qquad \forall i \in R^P \quad \text{(R2)}$$

 ©2012 SBC — Soc. Bras. de Computação

$$\sum_{r \in N^V, (k,l) \in L^V} B^V_{r,k,l} A^L_{i,j,r,k,l} \leq B^P_{i,j} \qquad \forall (i,j) \in L^P \quad \text{(R3)}$$

$$K^V_{r,j} A^R_{i,r,j} \leq K^P_i \qquad \forall i \in R^P, r \in N^V, j \in R^V \quad \text{(R4)}$$

$$\sum_{i \in R^P} A^R_{i,r,j} = 1 \qquad \forall r \in N^V, j \in R^V \quad \text{(R5)}$$

$$\sum_{j \in R^P} A^L_{i,j,r,k,l} - \sum_{j \in R^P} A^L_{j,i,r,k,l} = A^R_{i,r,k} - A^R_{i,r,l} \qquad \forall r \in N^V, (k,l) \in L^V, i \in R^P \quad \text{(R6)}$$

$$\sum_{q \in N^V, k \in R^V} A^R_{i,q,k} + \sum_{r \in N^V, l \in R^V} A^R_{i,r,l} \leq 1 \qquad \forall q, r \in X, i \in R^P \quad \text{(R7)}$$

$$\sum_{q \in N^V, (k,l) \in L^V} A^L_{i,j,q,k,l} + \sum_{r \in N^V, (o,p) \in L^V} A^L_{i,j,r,o,p} \leq 1 \qquad \forall q, r \in X, (i,j) \in L^P \quad \text{(R8)}$$

$$j A^R_{i,r,k} = l A^R_{i,r,k} \qquad \forall (i,j) \in S^P, r \in N^V, (k,l) \in S^V \quad \text{(R9)}$$

$$A^R_{i,r,j} = E^R_{i,r,j} \qquad \forall (i,r,j) \in E^R \quad \text{(R10)}$$

$$A^L_{i,j,r,k,l} = E^L_{i,j,r,k,l} \qquad \forall (i,j,r,k,l) \in E^L \quad \text{(R11)}$$

$$\sum_{j \in R^V} A^R_{i,r,j} \leq 1 \qquad \forall i \in R^P, r \in N^V \quad \text{(R12)}$$

As primeiras três restrições garantem que os requisitos de capacidade dos roteadores e enlaces virtuais serão atendidos. A restrição R1 garante que a quantidade de CPU requisitada por roteadores virtuais mapeados a um roteador físico não excederá sua capacidade máxima. A restrição R2 aplica o mesmo controle à capacidade de memória dos roteadores físicos, e a restrição R3, à largura de banda dos enlaces físicos.

A restrição R4 garante que todos os roteadores virtuais que devem realizar criptografia e decriptografia de pacotes serão mapeados a roteadores físicos que suportam tais operações. Tais roteadores virtuais são os roteadores de borda no caso de redes virtuais que solicitam criptografia fim-a-fim, e todos os roteadores no caso de redes virtuais que requerem criptografia ponto-a-ponto.

A restrição R5 garante que cada roteador virtual será mapeado a um roteador físico. De forma complementar, a restrição R6 garante que o caminho formado por um conjunto de enlaces físicos hospedando um enlace virtual será válido. Em outras palavras, o caminho físico hospedando um enlace virtual $(a, b)$ deve ser um caminho válido entre o

 ©2012 SBC — Soc. Bras. de Computação

roteador físico hospedando o roteador virtual $a$ e o roteador físico hospedando o roteador virtual $b$.

As restrições R7 e R8 referem-se a pares de redes virtuais conflitantes – isto é, redes virtuais que não podem compartilhar recursos físicos. A restrição R7 não permite que roteadores virtuais que pertencem a redes virtuais conflitantes sejam mapeados aos mesmos roteadores físicos. De forma análoga, a restrição R8 garante que enlaces virtuais dessas redes conflitantes não compartilharão quaisquer caminhos físicos.

A restrição R9 garante que todo roteador virtual que possua um requisito de localidade será mapeado a um roteador físico na localidade solicitada. As restrições R10 e R11 garantem que os elementos das redes virtuais previamente alocadas continuarão alocados aos mesmos elementos físicos. A alocação dos roteadores será mantida pela restrição R10, enquanto que a alocação dos enlaces, pela restrição R11. Por fim, a restrição R12 impede que múltiplos roteadores virtuais de uma mesma rede virtual sejam hospedados no mesmo roteador físico.

## 4. Avaliação

Para avaliar o modelo em Programação Linear Inteira apresentado na seção anterior, o mesmo foi implementado e executado no *CPLEX Optimization Studio*[2] versão 12.3. Os experimentos foram realizados em uma máquina com quatro processadores AMD Opteron 6276, usando no máximo quatro *threads* simultâneas. A máquina possui 64 GB de RAM e usa o sistema operacional Ubuntu GNU/Linux Server 11.10 x86_64.

### 4.1. Cenários

Para realizar os experimentos, foi desenvolvido um simulador capaz de gerar requisições de redes virtuais de forma aleatória. O simulador é executado por 500 janelas de tempo, e são geradas em média cinco requisições em cada uma, seguindo uma distribuição de Poisson. Cada requisição permanece alocada por, em média, cinco janelas de tempo, seguindo uma distribuição exponencial. Ressalta-se que essa forma de instanciação, isto é, o emprego de janelas de tempo e os modelos de chegada de requisições e de duração de redes virtuais na infraestrutura, é empregada em trabalhos importantes da área, com destaque para o realizado por Yu et al. [Yu et al. 2008].

A topologia da rede física e de cada rede virtual é gerada por meio da ferramenta BRITE[3], usando o modelo Barabási-Albert (BA-2) [Albert and Barabási 2000]. A rede física possui 100 roteadores, cada um com capacidade total de CPU definida como 100%, e 256 MB de memória. Além disso, os roteadores são distribuídos uniformemente entre 16 localidades, e 95% suportam protocolos que os permitem oferecer serviços de criptografia. A largura de banda dos enlaces físicos é distribuída uniformemente entre 1 e 10 Gbps.

As requisições de redes virtuais possuem entre 2 e 5 roteadores cada. Em cada rede virtual, dois roteadores (os *end points* dessa rede) possuem requisitos de localidade, gerados aleatoriamente entre as 16 localidades existentes. 35% das requisições geradas não possuem requisitos de criptografia, enquanto que 35% requerem criptografia fim-a-fim, e as demais 30%, criptografia ponto-a-ponto. De forma independente, 5% das requisições possuem conflito com uma rede já alocada no substrato.

[2]http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/
[3]http://www.cs.bu.edu/brite/

 ©2012 SBC — Soc. Bras. de Computação

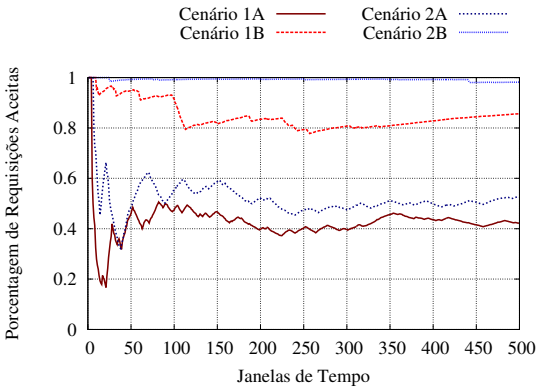



**Figura 1. Porcentagem média de requisições aceitas nos experimentos realizados.**

Foram criados dois cenários para avaliar o modelo, os quais se diferenciam pelos requisitos de capacidade dos elementos das redes virtuais. No primeiro, denominado *cenário 1*, cada roteador virtual requer entre 10 e 50% de CPU, e entre 32 e 128 MB de memória. Os enlaces das redes virtuais nesse cenário requerem entre 1 e 5 Gbps. Já no *cenário 2*, roteadores requerem entre 10 e 25% de CPU e entre 32 e 64 MB de memória, e enlaces requerem entre 1 e 2.5 Gbps. Os limites superiores dos requisitos do cenário 1 equivalem a 50% da capacidade disponível em roteadores e enlaces físicos, enquanto que no cenário 2, tais limites equivalem a 25% da capacidade dos elementos físicos. Todos os parâmetros previamente descritos seguem uma distribuição uniforme.

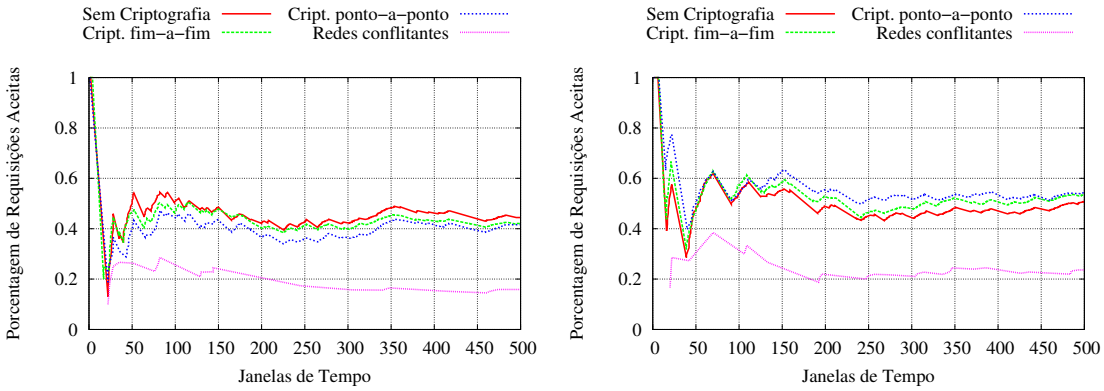
## 4.2. Resultados

Inicialmente, foram realizados experimentos seguindo os dois cenários descritos na subseção anterior. Em seguida, os experimentos foram repetidos, utilizando as mesmas redes físicas e as mesmas requisições geradas em cada janela de tempo, porém com uma versão modificada do modelo que desconsidera requisitos de segurança. As versões dos cenários 1 e 2 em que são considerados os requisitos de segurança são denominadas *1A* e *2A*. Já as versões modificadas para ignorar tais requisitos são denominadas *1B* e *2B*. Os resultados dos experimentos realizados com essas diferentes versões foram comparados para caracterizar o impacto causado pelo emprego de serviços relacionados à confidencialidade.

Analisou-se a taxa de aceitação de requisições de redes virtuais nos experimentos realizados. Ressalta-se que requisições somente são negadas caso não seja possível acomodar a rede virtual solicitada no substrato atendendo todos os seus requisitos. A Figura 1 ilustra a taxa média de aceitação obtida em cada cenário. Cada ponto no gráfico denota a taxa média de aceitação obtida desde o início do experimento até a janela de tempo em questão.

Analisando o gráfico, percebe-se, de forma clara, o impacto causado pelo fornecimento de serviços relacionados à confidencialidade. Nos cenários 1A e 2A, em que os requisitos de segurança são considerados, a taxa média ao fim dos experimentos é de, respectivamente, 42,2% e 52,5%. Já nos demais cenários, 1B e 2B, as taxas são de respectivamente 85,6% e 98,2%. Observa-se, ainda, uma taxa de aceitação maior nas variantes do cenário 2 em relação às do cenário 1, devido ao fato de que nos cenários 2A e 2B as requisições possuem requisitos de capacidade mais baixos. Além disso, é possível perceber que, em todos os casos, a taxa de aceitação inicial é de 100%, visto que o substrato

 ©2012 SBC — Soc. Bras. de Computação




(a) Cenário 1A (b) Cenário 2A

**Figura 2. Porcentagem média de requisições aceitas nos cenários 1A e 2A, divididas por tipo de requisição.**

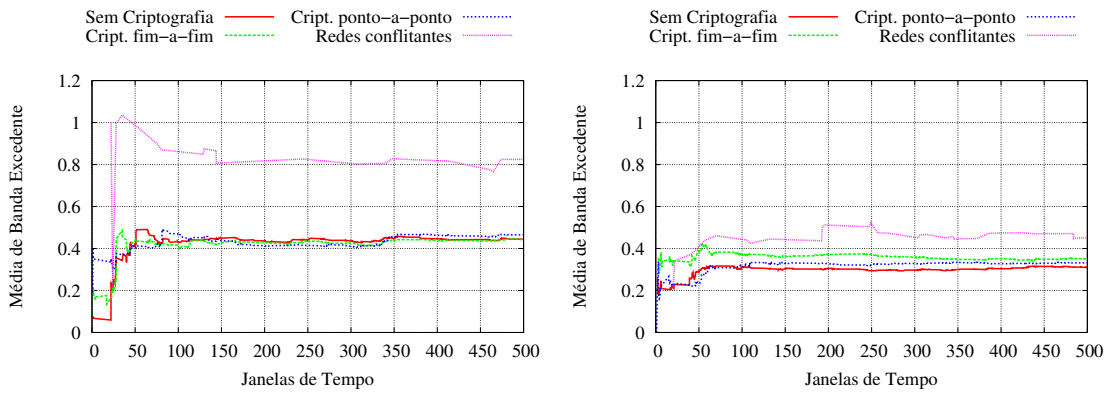
encontra-se desocupado no início do experimento. Após algum tempo, devido à saturação dos elementos físicos, requisições passam a ser negadas, causando quedas na taxa média. A mesma, então, volta a subir conforme algumas das redes virtuais alocadas alcançam sua duração máxima e são removidas do substrato. Esse comportamento se repete ao longo dos experimentos, gradualmente convergindo para um valor médio.

A seguir, a Figura 2 apresenta, em maiores detalhes, a taxa de aceitação dos diferentes tipos de requisições presentes nos cenários 1A e 2A. Os gráficos exibem um comportamento em grande parte similar ao anterior, porém nota-se que em ambos a taxa de aceitação de requisições com conflitos é significativamente mais baixa do que a média geral. Isso se deve à dificuldade de alocar redes virtuais conflitantes sem que nenhum de seus roteadores e enlaces se sobreponham. A taxa média de aceitação de redes virtuais com conflitos é de 15,9% no cenário 1A, e de 23,6% no cenário 2A.

Ainda observando o gráfico ilustrado na Figura 2, percebe-se que não há uma grande diferença entre a porcentagem de redes aceitas sem criptografia, com criptografia fim-a-fim e com criptografia ponto-a-ponto. As taxas médias ao longo do cenário 1A são de, respectivamente, 44,4%, 42,1% e 41,6%. Já no cenário 2A, as médias são de, respectivamente, 50,8%, 53,3% e 54,2%. É importante salientar que, nos experimentos realizados, 95% dos roteadores da rede física oferecem suporte a protocolos que permitem a criptografia e decriptografia de pacotes, e que o modelo não considera custos adicionais de processamento e memória necessários para tais operações. Considerando-se uma taxa menor de equipamentos com suporte a tais operações, ou os custos adicionais associados às mesmas, julga-se que haveria uma diferença mais expressiva entre os diferentes tipos de requisições. Contudo, acredita-se conseguir, com o experimento realizado, oferecer uma boa visão global do custo associado para satisfazer requisitos de segurança no contexto investigado.

A próxima análise foca na largura de banda total necessária para alocar cada requisição em relação à largura de banda requisitada. A largura de banda ocupada por redes virtuais alocadas no substrato tende a ser mais alta do que a requisitada, visto que um único enlace virtual pode ser alocado em um caminho composto por uma série de enlaces físicos. A Figura 3 apresenta a média de largura de banda excedente das requisições aceitas nos cenários 1A e 2A, separadas pelo nível de confidencialidade solicitado. Ressalta-

 ©2012 SBC — Soc. Bras. de Computação




(a) Cenário 1A (b) Cenário 2A

**Figura 3. Média de largura de banda excedente necessária para acomodar as requisições de diferentes tipos aceitas nos cenários 1A e 2A.**

se que o modelo visa minimizar a largura de banda consumida pelas redes virtuais alocadas, e que todas as soluções resultantes dos experimentos realizados são ótimas. Nota-se que requisições com conflitos tendem a consumir uma banda significativamente maior em relação às demais. No cenário 1A, tais requisições consomem em média 82,5% de largura de banda além do solicitado, e no cenário 2A, 45,1%. Em outras palavras, o custo da alocação de redes conflitantes para provedores de infraestrutura torna-se muito maior. Isso ocorre devido à necessidade de se utilizar caminhos mais longos para evitar a sobreposição dos elementos dessas redes.

Quanto à largura de banda excedente dos demais tipos de requisições, no início do experimento 1A há uma diferença visível entre as mesmas. Até a vigésima janela de tempo, requisições sem criptografia possuem banda média excedente de 6,7%, enquanto que requisições com criptografia fim-a-fim, 13,4%, e as com criptografia ponto-a-ponto, 35,1%. Ou seja, até esse momento, requisições com níveis mais altos de criptografia exigem uma quantidade maior de recursos para serem alocadas. No entanto, de forma similar aos gráficos da Figura 2, ao longo da execução tais valores convergem para porcentagens muito próximas. Ao término da execução, as médias de largura de banda excedente situam-se entre 44,3% e 46,5%. No cenário 2A, há sobreposições entre tais médias desde o início do experimento, e ao término do mesmo, novamente as médias encontram-se muito próximas, entre 31,1% e 35,1%.

Por fim, é apresentado o tempo médio necessário para encontrar a alocação ótima de cada requisição aceita nos experimentos realizados. Em todos os cenários, a média permanece abaixo dos 4,5 segundos do início ao fim dos experimentos. Apesar de pequena, é possível notar uma diferença na média de tempo entre os diferentes cenários. Os cenários em que os requisitos de segurança são considerados possuem média de tempo mais baixa do que os cenários nos quais os mesmos são ignorados. De forma similar, os cenários com requisitos de capacidade mais altos são resolvidos mais rapidamente do que os cenários em que tais requisitos são mais baixos. Tais diferenças podem ser explicadas pela diminuição no espaço de busca causada tanto por restrições relacionadas à segurança quanto pelos requisitos de capacidade mais altos. A presença de um número maior de restrições ou requisitos de capacidade tendem a diminuir o espaço de soluções factíveis, o que pode a tornar a busca pela solução ótima mais rápida. Nos cenários 1A e 1B, o tempo médio é de respectivamente 2,29 e 2,75 segundos, enquanto que nos cenários 2A e 2B os

 ©2012 SBC — Soc. Bras. de Computação

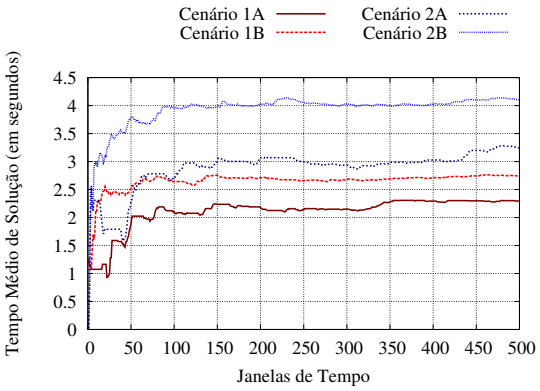



**Figura 4. Tempo médio necessário para encontrar a alocação ótima nos experimentos realizados.**

mesmos são de 3,24 e 4,1 segundos.

Em resumo, os resultados apresentados mostram que há um impacto significativo em termos de aceitação de requisições de redes virtuais e consumo de banda por parte das requisições aceitas ao se considerar o emprego de serviços de segurança. Esses fatores afetam negativamente a possibilidade de lucro que um provedor de infraestrutura pode obter, visto que haverá menos redes alocadas no substrato, e o custo das mesmas tenderá a ser mais alto. Portanto, julga-se importante considerar o custo necessário para prover segurança às redes virtuais no momento da alocação. Além disso, a avaliação do tempo de resolução do problema mostra que a implementação do modelo proposto é capaz de produzir resultados ótimos em um tempo adequado para uso em ambientes de produção.

## 5. Conclusões

Virtualização de redes é um tópico importante e que tem recebido atenção da comunidade científica e da indústria, resultando na proposta de uma série de abordagens de alocação. Mais recentemente, surgiram propostas visando prover segurança a ambientes de redes virtuais. No entanto, os autores desconhecem tentativas anteriores de combinar ambas as áreas, provendo alocação ótima e orientada a segurança de recursos de redes virtuais.

Considerando que a alocação de recursos e a segurança são igualmente importantes, desenvolveu-se um modelo que combina restrições de CPU, memória, largura de banda e localidade com requisitos de segurança. Redes virtuais podem solicitar diferentes níveis de criptografia em comunicações entre seus roteadores, visando prover confidencialidade às mesmas, ou podem exigir que seus roteadores e enlaces virtuais não compartilhem dispositivos e caminhos físicos com outras redes virtuais específicas.

Os resultados obtidos demonstram o impacto significativo causado pelo aprovisionamento de serviços de segurança na alocação de redes virtuais, salientando a importância de considerá-los no processo de mapeamento. Além disso, o modelo proposto mostra-se capaz de produzir resultados ótimos em um tempo adequado. Ainda que não sejam considerados custos adicionais de processamento e memória associados aos processos de criptografia e decriptografia, os resultados são capazes de prover uma boa visão global desse impacto. Pretende-se realizar uma revisão mais profunda de trabalhos relacionados à segurança, visando obter medidas reais de tais custos para incorporá-los no modelo. Acredita-se que isso permitirá analisar as consequências do fornecimento de serviços de segurança com uma granularidade mais fina.

 ©2012 SBC — Soc. Bras. de Computação

Outra perspectiva para trabalhos futuros é permitir a reotimização de redes virtuais já alocadas, migrando recursos virtuais entre roteadores e enlaces do substrato. O processamento de requisições em tempo real pode levar à fragmentação dos recursos físicos, visto que as requisições não são conhecidas *a priori*. Por esse motivo, a reotimização periódica pode beneficiar o provedor de infraestrutura, diminuindo custos e permitindo que uma quantidade maior de requisições sejam atendidas. No entanto, o tempo necessário para avaliar possíveis realocações pode tornar proibitiva a obtenção de soluções ótimas. Por esse motivo, pretende-se criar um algoritmo baseado em metaheurísticas, produzindo soluções sub-ótimas porém minimizando o tempo necessário para obtê-las.

## Referências

Albert, R. and Barabási, A.-L. (2000). Topology of evolving networks: Local events and universality. http://link.aps.org/doi/10.1103/PhysRevLett.85.5234. *Phys. Rev. Lett.*, 85:5234–5237.

Alkmim, G. P., Batista, D. M., and Fonseca, N. L. S. (2011). Optimal mapping of virtual networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6.

Andersen, D. (2002). Theoretical approaches to node assignment. http://www.cs.cmu.edu/~dga/papers/andersen-assign.ps. Unpublished manuscript.

Anderson, T., Peterson, L., Shenker, S., and Turner, J. (2005). Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41.

Cabuk, S., Dalton, C. I., Ramasamy, H., and Schunter, M. (2007). Towards automated provisioning of secure virtualized networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 235–245, New York, NY, USA. ACM.

Cheng, X., Su, S., Zhang, Z., Wang, H., Yang, F., Luo, Y., and Wang, J. (2011). Virtual network embedding through topology-aware node ranking. In *SIGCOMM Computer Communication Review*, volume 41, pages 38–47, New York, NY, USA. ACM.

Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783 –791.

Davy, S., Serrat, J., Astorga, A., Jennings, B., and Rubio-Loyola, J. (2011). Policy-assisted planning and deployment of virtual networks. In *Network and Service Management (CNSM), 2011 7th International Conference on*, pages 1 –8.

Fernandes, N., Moreira, M., Moraes, I., Ferraz, L., Couto, R., Carvalho, H., Campista, M., Costa, L., and Duarte, O. (2010). Virtual networks: Isolation, performance, and trends. In *Annals of Telecommunications*.

Huang, D., Ata, S., and Medhi, D. (2010). Establishing secure virtual trust routing and provisioning domains for future internet. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1–6.

Kent, S. and Seo, K. (2005). Rfc 4301: Security architecture for the internet protocol. http://tools.ietf.org/rfc/rfc4301.txt.

Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking virtual network embedding: substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38(2):17–29.

 ©2012 SBC — Soc. Bras. de Computação

# Uma Arquitetura para Auditoria de Nível de Serviço para Computação em Nuvem

**Juliana Bachtold, Altair O. Santin, Maicon Stihler, Arlindo L. Marcon Jr, Eduardo Viegas**

Pontifícia Universidade Católica do Paraná (PUCPR) – Escola Politécnica – Programa de Pós-Graduação em Informática (PPGIa) – Curitiba – PR – Brasil

{jbachtold, santin, sthiler, almjr, eduardo.veigas}@ppgia.pucpr.br

***Abstract.*** *This paper's purpose is a multiparty Service Level Agreement (SLA) for cloud computing auditing. Audits are performed on the IaaS provider, IaaS contractor (SaaS provider) and SaaS client. The objective is to audit the cloud environment problems, internally and externally, in an unquestionable way by the parties. The proposal uses inspectors (auditing collector agents) and an independent auditor (third party), capable of identifying SLA deviations through information collected in the parties' environments. The results show that it is possible to audit and diagnose problems in the cloud by combining information from the parties with the independent auditing, in addition to avoiding conflicts of interest of the inspectors.*

***Resumo:*** *Este artigo apresenta uma arquitetura para auditoria multipartes de acordo de serviço (SLA) em computação em nuvem. São auditados o provedor de IaaS, contratante de IaaS (provedor de SaaS) e cliente de SaaS. O objetivo é auditar problemas internos e externos ao ambiente de nuvem de modo incontestável pelas partes. A proposta emprega inspetores (agentes de coleta para auditoria) e auditores independentes (terceiros), capazes de identificar desvios de SLA através de informações coletadas nos ambientes das partes. Os resultados mostram que é possível auditar e diagnosticar problemas na nuvem combinando informações das partes, com a autoria independente, inclusive evitando-se conflito de interesse dos inspetores.*

## 1. Introdução

A utilização da computação em nuvem cresceu intensamente nos últimos anos, devido a sua adaptação às demandas dos negócios e ao encapsulamento da complexidade da tecnologia adjacente.

Basicamente, a nuvem computacional é formada por provedores que fornecem infraestrutura (*Infrastructure as a Service*, *IaaS*) para contratantes de *IaaS,* que desenvolvem aplicações (*Software as a Service*, *SaaS*) consumidas por seus clientes (usuário final ou consumidor de *SaaS*). Alguns provedores ofertam diretamente o ambiente para desenvolvimento e execução da aplicação (*Plataform as a Service*, *PaaS)* para que contratantes de *PaaS*, da mesma forma, desenvolvam aplicações para ofertar diretamente ao cliente (consumidor de SaaS).

As características que fazem o modelo de computação em nuvem ser atrativo para o contratante do serviço também restringem as ações de administração do contratante à interface de gerenciamento fornecida pelo provedor de *IaaS*.

Ao transferir seus sistemas para a nuvem computacional as organizações

*Este trabalho é parcialmente financiado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC/MCTI), contrato 1313 e CNPq, processos: 310319/2009-9 e 478285/2011-6.*

 ©2012 SBC — Soc. Bras. de Computação

dividem com terceiros a responsabilidade pela tutela das informações e a execução de operações críticas aos seus negócios.

O estabelecimento de um Acordo de Nível de Serviço (*Service Level Agreement*, SLA) é uma forma de garantir que o serviço contratado venha a atender às necessidades da organização contratante. O SLA é um documento que contém os níveis de serviço esperados pelo contratante e garantidos pelo provedor, implementados como níveis de qualidade de serviço (*Quality of Service*, QoS) ou outros indicadores (métricas) definidos pelas partes.

Entretanto, o estabelecimento do SLA é apenas um dos componentes de governança para *cloud computing*. Considera-se que governança de *cloud computing* inclua, além do estabelecimento do SLA, a definição de políticas de uso e a implantação de rotinas de auditoria (Guo et. al., 2010).

A implantação de auditoria de SLA deve ser capaz de monitorar as ações de contratante e provedor dentro da nuvem, além de identificar responsabilidades externas sobre a degradação de serviço para qualquer uma das partes.

Em nuvem computacional o comportamento das partes (provedor, contratante e cliente), é responsável pelo desempenho do serviço computacional. Desta forma, a monitoração de SLA neste ambiente deve ser capaz de capturar informações sobre cada uma das partes para permitir a auditoria efetiva. Além de considerar o comportamento dos participantes, a auditoria efetiva deve ser capaz de detectar que fatores externos à nuvem alteraram a experiência (percepção) do cliente com relação ao serviço provido, como um problema com a conexão de rede entre as partes, por exemplo.

Se ocorrerem experiências negativas do cliente durante o consumo de serviços, deve ser possível para a auditoria identificar se o problema reside na aplicação do contratante, na conexão com o ambiente de nuvem computacional ou no provedor de infraestrutura, de maneira inquestionável para todas as partes. Ou seja, a auditoria multipartes necessita de pontos de monitoração com coleta continua de métricas sobre os serviços e recursos, considerados em cada uma das partes.

Entretanto, em *cloud computing,* o ambiente computacional das partes é segregado. Por este motivo, uma das partes não tem acesso às informações de auditoria das demais. Para contornar esta limitação e também para minimizar conflitos de interesse entre as partes, agentes de auditoria (inspetores) devem pertencer a uma entidade externa à nuvem para que esta tenha visibilidade sobre todas as partes. Os inspetores coletam e enviar as informações para um ambiente externo à nuvem. Então, um auditor independente e externo pode consolidar tais informações para identificar desvio de SLA e reportá-los à governança de TI do contratante. Este é o objetivo perseguido nesta proposta.

É importante frisar que o terceiro deve ser independente e confiável para todas as partes (provedor, contratante e cliente) e que as informações coletadas podem ser utilizadas no caso de questionamento quanto à aderência aos SLAs por qualquer e contra qualquer uma das partes.

Este trabalho apresenta uma arquitetura para implantação da auditoria multipartes em nível de IaaS e SaaS em nuvem computacional, com coleta de indicadores de forma anônima, implantados num protótipo que evidencia sua viabilidade.

O restante do trabalho está dividido da seguinte forma: a seção 2 apresenta a fundamentação; a seção 3 aborda os trabalhos relacionados; a seção 4 descrever a

 ©2012 SBC — Soc. Bras. de Computação

arquitetura proposta; a seção 5 apresenta o protótipo desenvolvido e os testes realizados e a seção 6 desenvolve a conclusão do trabalho.

## 2. Fundamentação

Esta seção aborda os tipos de serviços em ambiente de nuvem computacional, conceitos aplicáveis a acordos em nível de serviço e fundamentos sobre virtualização.

### 2.1. Tipos de serviço em nuvem computacional

Os serviços de computação em nuvem podem ser disponibilizados de diferentes formas. O NIST define três modelos de arquitetura de nuvem de acordo com os recursos que são entregues para o cliente (Mell et. al., 2011):

- Software em Nuvem como um Serviço (*SaaS*). O serviço oferecido ao cliente consiste em aplicações disponibilizadas pelo provedor de *SaaS* executando em uma infraestrutura de nuvem. É o serviço que o cliente consome, e geralmente consiste numa aplicação.
- Plataforma em Nuvem como um Serviço (*PaaS*). Conjunto de ferramentas, ambiente de desenvolvimento e produção oferecida ao contratante para facilitar o desenvolvimento de sua aplicação.
- Infraestrutura em Nuvem como um Serviço (*IaaS*). São recursos computacionais (infraestrutura física) oferecidos ao contratante. Recurso computacional, em geral, consiste numa máquina virtual que oferece armazenamento, processamento e rede para que o contratante possa instalar e executar os softwares, desde o sistema operacional até as suas aplicações.

### 2.2. Acordos em nível de serviço

Um SLA pode especificar as necessidades e níveis a serem obtidos no fornecimento do serviço com alto nível de abstração. Os serviços são avaliados através de métricas de desempenho, conhecidas como Indicadores de Nível de Serviço (*Service Level Indicators*, SLIs). A integração da métrica de serviço (SLI) com o seu (valor) objetivo é denominado Objetivo de Nível de Serviço (*Service Level Objective*, SLO). "Disponibilidade" e "tempo médio de resposta do serviço" são exemplos de SLIs utilizados em um SLA enquanto que valores como 95% de disponibilidade ou 1,3 segundos de tempo de resposta, respectivamente, são exemplos de valores para SLOs (Sauvé et. al., 2005).

Para uma aplicação atender a um SLA, os recursos de infraestrutura que a suportam, por exemplo: processador, banda de rede e memória, devem ser adequadamente dimensionados. Cada recurso possui seus atributos como utilização do processador, largura de banda e memória disponível. As medições destes atributos são as métricas de recurso. São exemplos de métricas de recurso: percentual de memória disponível, percentual de consumo de rede, percentual de utilização de CPU etc.

Um conjunto de métricas de recurso reunidas para especificar requisitos de um serviço é denominado parâmetros de Qualidade de Serviço (*Quality of Service*, QoS). Métricas de recurso também são utilizadas para compor SLIs, quando estes SLIs compõem SLAs que regem o fornecimento de *IaaS*.

### 2.3. Virtualização

Virtualização de servidores implica em abstrair os recursos físicos de um servidor, de modo a parecerem diferentes logicamente do que são fisicamente, em geral

 ©2012 SBC — Soc. Bras. de Computação

representando uma fração do hardware físico. Virtualização é utilizada por todas as nuvens computacionais e é considerada uma de suas características – característica esta não restrita apenas às nuvens que ofertam IaaS.

Num ambiente virtualizado a camada de software abstrai o hardware físico e instancia várias máquinas virtuais, cada uma com seus processos, sistema operacional e aplicações. Esta camada de abstração é controlada pelo *hypervisor* (hipervisor). É o *hypervisor* que disponibiliza a abstração de recursos de hardware para os sistemas operacionais.

Um sistema operacional executado numa instância de máquina virtual é chamado de SO hóspede (*guest*). As máquinas virtuais instanciadas no mesmo hardware compartilham os "mesmos recursos físicos". Esse compartilhamento de recursos pode significar um melhor aproveitamento do hardware, uma vez que os servidores oscilam quanto ao consumo de recursos de hardware, apresentando picos e ociosidade de uso em diferentes momentos.

## 3. Trabalhos relacionados

O trabalho de Skene (Skene et.al., 2010) modela o consumo de serviços na Internet (não especificamente em nuvem computacional), apresentando a influência dos fatores externos para contratante e provedor no consumo dos serviços. Problemas de conexão do cliente podem influenciar a percepção de tempo de resposta no lado servidor, por exemplo. Para determinar o cumprimento do SLA, torna-se importante que as influências externas sejam filtradas, utilizando-se o exemplo do SLI "tempo de resposta", a influencia da latência da conexão entre cliente-servidor pode ser obtida através do registro de *timestamps* entre o envio e o recebimento de pacotes.

No trabalho de Chen (Chen, Y. et. al., 2011) é apresentada uma cadeia de confiança estabelecida na utilização da nuvem: contratantes podem ter SLA estabelecido com provedores de *SaaS*, que por sua vez possuem outro SLA com provedores de *IaaS* – o que nem sempre é visível para o cliente. Porém, o comportamento do contratante pode influenciar o nível de serviço do provedor *IaaS*, em determinadas situações, por exemplo, quando seus clientes demandarem muito a rede. A interdependência entre as partes introduzida pela computação em nuvem requer auditabilidade mútua – provedores e contratantes devem ser auditados nos níveis de serviço que provem. Entretanto, em seu trabalho Chen não relaciona a terceira entidade auditável, o cliente. Ao considerá-lo estamos estendendo a proposta original de auditoria mútua para auditoria multipartes.

A inclusão do cliente como entidade auditável baseia-se no trabalho de Morsel (Morsel, 2001), onde são apresentados exemplos de métricas para obtenção da Qualidade da Experiência (*Quality of Experience*, QoE), como a utilização do parâmetro "tempo de resposta" a ser medido na estação do usuário. Segundo os autores, as métricas de Qualidade de Serviço (*Quality of Service*, QoS) não traduzem a percepção do usuário quanto ao nível de serviço consumido.

Na proposta de Barbosa (Barbosa et. al., 2006) são discutidas arquiteturas de auditoria de SLA em *grid computing*. As alternativas são avaliadas quanto à eficiência de cada arquitetura em diferentes cenários. As arquiteturas apresentadas no artigo são baseadas na existência de uma entidade terceira, que seja considerada confiável por ambos, para mediar a relação cliente e servidor. As arquiteturas apresentadas no trabalho não consideram a existência de ambientes computacionais segregados para provedor e contratante, uma característica do ambiente de nuvem computacional.

 ©2012 SBC — Soc. Bras. de Computação

Haeberlen (Haeberlen, 2010) considera que para tornar a nuvem computacional *accountable* é necessário que seja definida uma taxa de amostragem adequada, todas as entidades monitoradas devem ter relógios sincronizados para captura e comparação adequada de *timestamps*. Os logs devem ser armazenados em outro ambiente, diferente daquele onde foram feitas as coletas. Além disto, os logs serem invioláveis, e.g. *write-once.* Apesar de definir os requisitos de auditabilidade, Haeberlen não define uma arquitetura para auditoria em nuvem.

Emeakaroha (Emeakaroha et. al., 2010) apresenta uma nuvem que utiliza a auditoria de SLA integrada ao funcionamento de um *broker* (negociador), utilizando nas medições da infraestrutura para redimensionar a nuvem computacional. Na arquitetura apresentada no artigo, não existe a figura do inspetor para o ambiente do cliente, fornecendo apenas informações sobre consumo de recursos de infraestrutura na visão do provedor.

Nenhum dos trabalhos relacionados oferece uma arquitetura para auditoria em nuvem que implemente a auditabilidade multipartes para computação em nuvem e para tal obtenha informações de cliente, contratante e provedor. Além disto, nenhuma proposta mostra-se capaz de identificar as causas de desvio de SLA dentro e fora da nuvem computacional ou preocupa-se com o possível conflito de interesses do auditor.

## 4. Proposta

A seguir são apresentadas as premissas e a arquitetura da proposta com seus principais componentes.

### 4.1. Premissas para auditoria de SLA

A arquitetura proposta para auditoria de SLA em computação em nuvem considerou os seguintes aspectos:

- As informações de auditoria devem ser obtidas por um terceiro; entidade independente em relação ao cliente, contratante e provedor; porém confiável a todos;
- Os inspetores coletam os dados do cliente, contratante e provedor continuamente e, os fornecem ao auditor que cria o SLI e o compara ao respectivo SLO;
- As informações obtidas no ambiente do cliente e contratante devem auxiliar na identificação de fatores externos, que influenciam negativamente a percepção da nuvem;
- As informações obtidas no ambiente do contratante e do cliente podem ser confrontadas com as informações do provedor para identificar não conformidade de SLA causado por fatores internos à nuvem (sejam de responsabilidade do contratante ou do provedor);
- O inspetor não deve ser capaz de associar o consumo de serviços ao cliente ou contratante dentro da nuvem, mas o rastreio das ações de clientes e contratantes dentro da nuvem deve ser possível para o pessoal de governança de TI do contratante;
- Através do comportamento de contratante e provedor é possível responsabilizar as partes por suas ações dentro do ambiente de nuvem;
- A coleta de informações do contratante permite que o nível de serviço seja auditado, considerando-se também a experiência de consumo do cliente.

 ©2012 SBC — Soc. Bras. de Computação

### 4.2. Visão geral do cenário da proposta

O mecanismo de auditoria baseia-se em três componentes: inspetores, auditores e camada de governança. Os inspetores são posicionados dentro da nuvem para realização da coleta, sendo inseridos na infraestrutura do provedor (camada *IaaS*) e na camada de aplicação do contratante (camada *SaaS*). Cada inspetor é responsável por coletar dados que irão gerar um SLI ou parte desse.

Os auditores permanecem lógica e fisicamente fora da nuvem auditada e recebem os dados dos inspetores para calcular os SLIs. Pode haver um auditor para cada SLI ou família de SLIs correlacionados (i.e. calculados a partir dos mesmos dados).

Após calcular o SLI, o auditor confronta os resultados com o SLO a fim de identificar ocorrências de não-conformidade com o SLA.

A camada de governança obtém dos auditores os valores calculados para os SLIs e as sinalizações para os SLIs que apresentaram desvio. Por conhecer todos os SLIs calculados para cada cliente, a camada de governança, em caso de desvio de SLA, torna-se capaz de isolar causas de problemas a partir das medições obtidas. Na seção 4.3 será abordado a interpretação das medições.

Na Figura 1 é mostrado um exemplo de auditoria para um tipo de SLI para *IaaS* e um SLI medido em *SaaS*, mas é possível monitorar mais SLIs através da utilização em paralelo de outros inspetores e auditores (representado pela linha pontilhada na Figura 1). Por fim, uma camada de governança pode gerar dados estatísticos, a partir dos dados coletados, e avaliar o serviço de determinado provedor ou as causas do desvio de SLA para determinado cliente – informações que não são geradas pelos os auditores porque hierarquizamos a informação contextualizada para evitar o conflito de interesses.

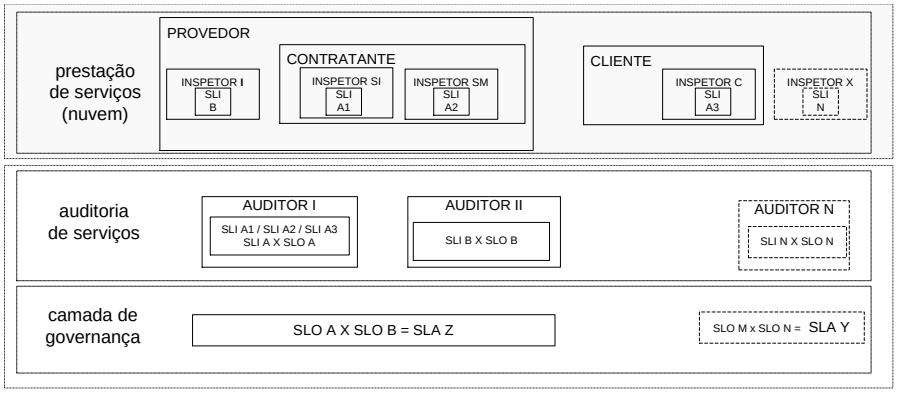



**Figura 1. Visão geral da arquitetura da proposta**

Na proposta um inspetor só lê o valor de um indicador relacionado a um pseudônimo. O mesmo acontece com o auditor que cria SLIs do mesmo tipo para o mesmo pseudônimo e os confronta com atributos de SLO para este pseudônimo, disponibilizando-os a camada de governança de modo consolidado. Porém, nem o inspetor, nem o auditor, conseguem relacionar o pseudônimo a uma identidade do mundo real – só a camada de governança tem esta habilidade. Assumimos que a homologação dos códigos que coletam os indicadores deve ser feita antes que os mesmos sejam colocados nos pontos de coleta do inspetor. Estes códigos podem ser inspecionados a qualquer momento pelo auditor através de geração de *hash code* criptográfico. Neste caso, é sugerido que o código do coletor seja desenvolvido usando a técnica de reentrância (Wloka et al., 2009).

 ©2012 SBC — Soc. Bras. de Computação

Uma vez que os inspetores consultam as camadas de infraestrutura e aplicação, é necessário definir as premissas iniciais de ambiente para obtenção de métricas de auditoria. A infraestrutura da arquitetura presume a existência de virtualização. Para a camada de aplicação a arquitetura deve estar preparada para funcionar com aplicações disponibilizadas em servidores web – nos papéis de servidor de interface (para o cliente) e motor de aplicação.

Na camada de infraestrutura o inspetor (Inspetor I) obtém informações de virtualização sobre recursos disponibilizados e consumidos, controlados pelo *hypervisor*, i.e., o Inspetor I extrai as medições de SLIs da infraestrutura.

**Tabela I – Descrição dos tempos coletados pelos inspetores**

| Inspetor | Onde | ID Registro | Função interna a operação |
|---|---|---|---|
| Inspetor C | Cliente (estação do usuário) | Tempo 1 (T1) | Registra o *timestamp* do envio de página web pelo cliente, i.e, no momento da requisição de operação a aplicação. |
| Inspetor SI | Servidor de *interface* | Tempo 2 (T2) | Registra o *timestamp* do recebimento do pedido de operação pelo servidor. |
| Inspetor SI | Servidor de *interface* | Tempo 3 (T3) | Registra o *timestamp* imediatamente anterior ao envio da solicitação para o servidor de *motor* da aplicação. |
| Inspetor SA | Servidor *de motor* da aplicação | Tempo 4 (T4) | Registra o *timestamp* imediatamente posterior ao recebimento da requisição de operação. |
| Inspetor SA | Servidor de *motor* da aplicação | Tempo 5 (T5) | Registra o *timestamp* imediatamente anterior ao envio do resultado da operação. |
| Inspetor SI | Servidor de *interface* | Tempo 6 (T6) | Registra o *timestamp* imediatamente posterior ao recebimento da resposta do motor da aplicação. |
| Inspetor SI | Servidor de *interface* | Tempo 7 (T7) | Registra o *timestamp* imediatamente anterior ao envio do resultado para o cliente. |
| Inspetor C | Cliente (estação do usuário) | Tempo 8 (T8) | Registra o *timestamp* imediatamente posterior ao recebimento do resultado da operação. |

Na camada de aplicação os inspetores devem ser instalados no servidor de interface, no cliente e no servidor do motor de aplicação. As informações coletadas pelos inspetores são base para os auditores confeccionarem o SLI "tempo de resposta para o cliente (usuário) *por* operação".

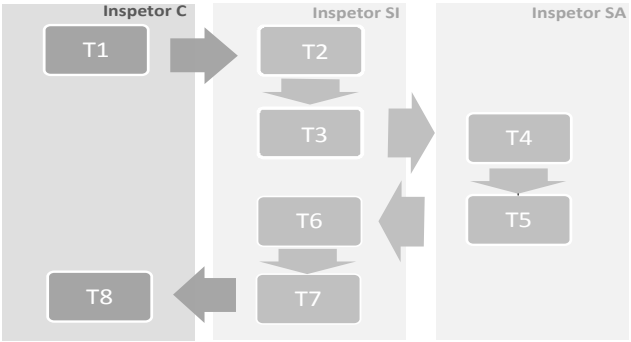



**Figura 2 – Coleta de *timestamps* pelos inspetores**

Os *timestamps* gerados por cada inspetor de software estão descritos na Tabela I e são apresentados abaixo:

- Um inspetor de aplicação é executado no cliente (Inspetor C). O objetivo do inspetor C é coletar *timestamps* das transações que ocorrem dentro da estação do cliente (T1 e T8, Figura 2).

 ©2012 SBC — Soc. Bras. de Computação

- Os inspetores de aplicação que são executados no ambiente de aplicação do contratante estão assim alocados:
    a) O Inspetor SI é executado no servidor de interface e registra *timestamps* na interação com o cliente e com o servidor de motor da aplicação (T3, T6 e T7, Figura 2).
    b) O Inspetor SA é executado no servidor de motor de aplicação (Inspetor SA) e registra um *timestamp* (T4, Figura 2) ao receber a requisição para execução de operação e outro ao devolver a resposta para o solicitante (T5, Figura 2).

### 4.3. O papel do auditor e do módulo de governança

Além do SLI principal "tempo de resposta para o cliente por operação" (*Tru),* os auditores geram outros indicadores intermediários à partir dos *timestamps* coletados (Tabela II). A geração destes indicadores é necessária para localização das causas de ocorrências de desvios de SLA, interpretadas da seguinte maneira:

- Problemas de desempenho externos à nuvem, nos serviços de conexão do provedor e/ou do cliente à Internet. Identificam-se altas taxas de latência nas conexões do cliente, do servidor de interface ou do servidor de aplicação, através de SLIs calculados a partir de indicadores no lado do cliente ou servidor de interface. A identificação ocorre quando observados valores superiores aos limites definidos para T8 - T7 (Trd2) e T2 – T1 (Trd1), respectivamente.
- Problemas de desempenho dentro da nuvem são evidenciados por indicadores calculados com a diferença entre *timestamps* coletados dentro do mesmo servidor, caracterizando maior tempo para conclusão de determinada operação, nos seguintes casos:
    a) no servidor de interface, evidenciado por T3 – T2 (Tsi).
    b) no servidor de motor da aplicação, evidenciado por T5 – T4 (Tsa).

**Tabela II – SLIs gerados pelo auditor**

| SLI | Operação representada pelo indicador | Cálculo | Análise do módulo de governança |
|---|---|---|---|
| Tru | Identifica o tempo de resposta para o usuário final (execução completa de uma operação). | Tru = T8– T1 | Caso o SLO seja extrapolado, sugere a análise de mais indicadores, para identificar a operação gargalo. |
| Trd1 | O indicador obtém o tempo para o envio de conteúdo do cliente para o servidor de *front-end* (aplicação). | Trd1= T2– T1 | Tempo elevado entre o envio e o recebimento do conteúdo indica gargalos na comunicação em rede entre o cliente e o servidor. |
| Tsi | O indicador obtém o tempo de processamento do servidor de interface para executar a operação de recebimento e empacotamento de conteúdo. | Tsi= T3 - T2 | Tempo elevado para receber e transmitir o conteúdo indica gargalo no servidor de interface, (não identificando se há problema na disponibilização do hardware contratado ou sobrecarga de recursos). No caso de medições mais altas são necessárias informações adicionais do Inspetor I. |
| Tsa | O indicador obtém o tempo de processamento do servidor de motor de aplicação para executar o envio de conteúdo. | Tsa= T5– T4 | Tempo elevado para receber e transmitir o conteúdo indica gargalo no servidor de motor de aplicação, (não identificando se há problema na disponibilização do hardware contratado ou sobrecarga de recursos). No caso de medições mais altas são necessárias informações adicionais do Inspetor I. |
| Trd2 | O indicador obtém o tempo para o envio do conteúdo do servidor de interface para o cliente. | Trd2= T8– T7 | Tempo elevado entre o envio e o recebimento do conteúdo indica gargalos na comunicação em rede entre o servidor de interface e o cliente. |

 ©2012 SBC — Soc. Bras. de Computação

A identificação de problemas dentro da nuvem (evidenciada pelos indicadores Tsi e Tsa), em si, não caracteriza não-conformidade do provedor de acesso ou problemas no ambiente do cliente. Estas medições precisam ser combinadas com as informações obtidas pelo Inspetor I. Para as operações que levaram mais tempo para serem concluídas, o auditor busca evidências de maior consumo (ou até mesmo esgotamento) de recursos de hardware, como CPU e memória.

Anomalias em funções executadas dentro de uma operação podem ser reveladas pelas correlações entre *timestamps* obtidos pelos inspetores. A Tabela II sintetiza os indicadores e as respectivas análises que podem ser geradas pelo módulo de governança

### 4.4. Mapeamento entre contratante e cliente

O contratante do serviço de nuvem pode ser uma organização pública ou privada que oferece serviços desenvolvidos em nível *SaaS* a um conjunto de clientes; são os clientes que efetivamente realizam as operações nos serviços e aplicações oferecidas na camada *SaaS* da nuvem computacional. Assim, um provedor pode atender a vários contratantes (locatários) e este contratante pode, por sua vez, representar vários clientes.

Os *timestamps* obtidos pelos inspetores nas operações realizadas por clientes são transmitidos ao repositório do auditor. Naturalmente, cada cliente é identificado por um pseudônimo para evitar conflito de interesse de auditores e inspetores.

Um servidor de pseudônimos externo ao ambiente do mecanismo de monitoração e coleta (i.e., no domínio de governança) fornece os pseudônimos ao cliente. Ainda que o inspetor observe este pseudônimo a cada operação feita nos serviços da nuvem e o registre para fins de rastreamento, não deverá obter outras informações a ponto de identificar o cliente no mundo real.

O servidor de pseudônimos os renova com certa periodicidade para evitar o rastreamento de clientes que, por alguma razão – externa ao sistema de identificação, tenham seus pseudônimos reconhecidos. O registro da operação é armazenado associado ao pseudônimo e somente quando necessário poderá ser associado à verdadeira identidade do cliente. Somente com intervenções externas é fornecido o mapeamento entre pseudônimo e cliente, em geral por questões legais, que geralmente surgem quando há impasse quanto à conformidade de SLAs.

## 5. Protótipo

Para avaliar a viabilidade da proposta, a arquitetura para auditoria de SLA foi implementada em um protótipo e testada.

### 5.1. Desenvolvimento do Protótipo

O protótipo foi desenvolvido sobre uma aplicação Web, simples, baseada no envio de e-mails (mensagens e anexos). A aplicação está subdividida em camada de interface e motor de aplicação. A camada de interface é responsável por receber e enviar os formulários de interação (página web) com o cliente, tratando as respostas recebidas e encaminhando as solicitações de processamento ao motor de aplicação (um servidor SMTP – *Simple Mail Transfer Protocol*), que recebe o formulário de envio e seus anexos e realiza o envio das mensagens via SMTP.

Os servidores foram instanciados numa nuvem computacional e devem possuir configuração mínima de 01 núcleo virtual e 256 MB RAM. O servidor de interface se utiliza do Apache TomCat e acesso a um SGBD (Sistema de Gerenciamento de Banco de Dados) externo, nos testes adotou-se o MySQL. Todos os servidores precisam ter

 ©2012 SBC — Soc. Bras. de Computação

seus relógios sincronizados a partir de um mesmo servidor NTP (*Network Time Protocol*).

As mensagens trocadas entre o servidor de interface e o servidor de motor de aplicação estão no formato JSON (formato de troca de mensagens alternativo ao XML – *Extensible Markup Language*). O serviço disponibilizado como motor de aplicação possui o estilo arquitetural REST (*REpresentational State Transfer*).

O Inspetor I conecta-se no *hypervisor* obtendo informações diretamente do *Xentrace*, que é um gerador de eventos disponível nativamente no virtualizador *XEN* (*bare-metal hypervisor*). As informações são enviadas para o auditor para registro da operação como descrito na seção 4.4. O Inspetor I, desenvolvido em 'C', salva as informações coletadas em arquivos texto para posterior consolidação no auditor.

Para fins de protótipo, o Inspetor I irá obter apenas o consumo de CPU por servidor para identificar esgotamento de recursos de processamento, entretanto, o Inspetor I é capaz de obter do *hypervisor* métricas de outros recursos de hardware. Como estas métricas não serão utilizadas nos testes a seguir, optamos por não mencioná-las.

Os inspetores da camada de aplicação, desenvolvidos em Java, foram nomeados de acordo com o ambiente onde são de executados, conforme apresentado na seção 4.2.

### 5.2. Definição dos cenários de testes

A escolha do ambiente de testes apenas em nuvem privada foi determinada pela impossibilidade de se utilizar nuvens públicas, uma vez que nestas o Inspetor I não teria acesso ao *hypervisor*. Desta forma, todos os cenários propostos exprimem variações de condições dentro de um mesmo ambiente computacional.

**Tabela III – Planejamento dos cenários de testes da proposta**

| Objetivo do Cenário | Descrição | Implantação | Indicadores afetados |
|---|---|---|---|
| A – Definição de baseline, criação de indicadores de referência. | Recursos de processamento de ambos os servidores dedicados exclusivamente ao processamento da aplicação. Utiliza-se a banda de rede disponível para este cenário. | Execução de simulações no ambiente, sem implantação de restrição de banda ou adição de carga de processamento nas VMs dos servidores de interface e motor de aplicação. | Todos os indicadores são obtidos a fim de gerar as medições iniciais |
| B – Verificar a capacidade dos inspetores capturarem ocorrências de sobrecarga de processamento nos dois servidores e do auditor correlacionar as medições a gargalos causados pela infraestrutura. | Recursos de processamento de ambos os servidores comprometidos com carga de aplicações concorrentes. Utiliza-se a banda de rede disponível para este cenário. | Utilização de aplicação Java, com algoritmo de criptografia, chave de 168 bits e o algoritmo DES gerando consumo adicional de CPU, dentro das VMs dos servidores interface e motor. A execução desta aplicação ocorre em paralelo à execução da aplicação de interface e da aplicação principal (*core*). | Os indicadores que devem refletir alteração são Tsi e Tsa |
| C – Avaliar a eficiência do protótipo em identificar gargalos relacionados à conectividade. | Recursos de processamento de ambos os servidores dedicados exclusivamente ao processamento da aplicação. Restringe-se a banda de rede | Execução de simulações no ambiente, sem adição de carga de processamento nas VMs dos servidores interface e motor.<br><br>Foi implantado a restrição de banda através do uso do CBQ (*Class Based Queue*), utilitário Linux para implantação de QoS. A velocidade medida entre o cliente e o servidor foi de 100 Kbps/s. | Os indicadores que devem refletir alteração são Trd1 e Trd2 |

Quando são executadas operações nos cenários B e C (Tabela III), obtêm-se resultados que desviam do SLA proposto (referência de SLA no protótipo é obtida em A). Como o

 ©2012 SBC — Soc. Bras. de Computação

desvio de SLA em B e C é causado por intervenções em pontos distintos da nuvem, planejou-se medir o resultado através de indicadores relacionados a estes componentes.

A nuvem privada foi implantada na rede da PUCPR. Os servidores foram instanciados numa nuvem computacional *open source*, o Eucalyptus (Numi et. al., 2010) versão 2.0.3, utilizando para virtualização o XEN versão 4.1.2. O servidor de interface executou em sistema operacional Linux Ubuntu 10.04 e Apache TomCat versão 7.0.27, com 01 núcleo virtual e 256 MB RAM. O servidor de motor da aplicação também executou em sistema operacional Linux Ubuntu 10.04 com 01 núcleo virtual e 256 MB RAM.

O objetivo dos testes foi identificar a capacidade do protótipo para:

- Identificar interferências externas – não associadas ao desempenho da nuvem (no cliente);
- Identificar ocorrências de não-conformidade de SLA através das informações obtidas no ambiente do cliente, contratante e do provedor;
- Obter informações que permitissem identificar o responsável pela não-conformidade com o SLA, usando a análise dos SLIs.

### 5.3. Resultado dos testes e análises

Os resultados dos testes mostram a capacidade do auditor para identificar ocorrências de violação de SLA, através da formulação dos SLIs para cada função interna as operações executadas pelo cliente e sua comparação com os respectivos SLOs.

O módulo de governança pode identificar interferências externas ou internas à nuvem com sucesso e relacioná-las ao responsável pelo fato. Os indicadores extraídos em cada cenário são mostrados nas tabelas IV, V e VI. Para todos os casos o coeficiente de variabilidade foi inferior a 5%. Cada conjunto de testes foi repetido quarenta vezes por cenário e o tempo de uma operação é medido a partir do envio do formulário de mensagem pelo usuário final até o processamento da mensagem pelo servidor SMTP. Os indicadores gerados foram previamente apresentados na Tabela II.

No cenário A (Tabela III) o objetivo foi capturar os indicadores em cenário neutro (Tabela IV), sem intervenções internas ou externas à nuvem.

**Tabela IV – Resultado dos testes para o cenário A**

| SLI | Tru | Trd1 | Tsi | Tsa | Trd2 |
|---|---|---|---|---|---|
| Média (em mSec) | 4.354 | 641 | 427 | 1.130 | 519 |

Neste cenário (Tabela IV), os valores de referência foram estabelecidos para os SLIs medidos nos próximos cenários, compondo os SLOs para cada operação. Na simulação A os indicadores de consumo de CPU não indicaram esgotamento (ou consumo excessivo) de recursos, ficando o consumo médio de CPU em 12,6% para o servidor de interface e 7,5% para o servidor de motor de aplicação durante a execução de cada operação.

**Tabela V - Resultado dos testes para o cenário B**

| SLI | Tru | Trd1 | Tsi | Tsa | Trd2 |
|---|---|---|---|---|---|
| Média (em mSec) | 18.466 | 775 | 1.162 | 10.182 | 579 |

No cenário B (Tabela III), foram iniciadas outras aplicações para geração de carga de processamento adicional nos servidores de interface e motor de aplicação, durante os acessos a aplicação do experimento (Tabela V). Com a carga adicional,

 ©2012 SBC — Soc. Bras. de Computação

esperou-se que fosse aumentado o tempo total das operações na aplicação e que os inspetores SI e SA fossem capazes de identificar quais as operações foram impactadas pela alteração no ambiente.

O auditor A, através das medições dos inspetores C, SI e SA identificou com sucesso o aumento nos tempos de funções (internas as operações) executadas no ambiente servidor, representadas pelos SLIs Tsi e Tsa. Comparados ao SLO estabelecido com as medições do Cenário A (Tabela IV), os indicadores desviaram-se em 270% e 900%, respectivamente. O maior tempo para execução das funções processadas nos servidores impactou em 420% o SLI Tru, que exprime o tempo total da operação.

Uma vez que as operações identificadas com a maior latência são realizadas dentro da nuvem – pois dependem exclusivamente de seu processamento pelo servidor, o módulo de governança requisita que o Auditor B (Figura 1) busque as medições do Inspetor I para obter o consumo de CPU dos servidores. Neste cenário foi identificado 86,5% de consumo de CPU no servidor de interface e 93,2% no servidor de motor de aplicação. Ficou evidenciado para o módulo de governança que a causa de desvio do SLO para esta operação foi causado pelo contratante por consumo intensivo de CPU.

No cenário C (Tabela III) a banda disponível para conexão do cliente ao servidor de interface foi reduzida, durante os acessos a aplicação do experimento (Tabela VI). Com esta alteração esperou-se que fosse aumentado o tempo total da operação e que os Inspetores C, SI e SA fossem capazes de identificar quais as funções que foram impactadas pela alteração no ambiente.

**Tabela VI - Resultado dos testes para o cenário C**

| SLI | Tru | Trd1 | Tsi | Tsa | Trd2 |
|---|---|---|---|---|---|
| Média (em mSec) | 15.557 | 12.009 | 189 | 1.185 | 654 |

Através dos dados dos inspetores C, SI e SA, o Auditor A identificou com sucesso o aumento nos tempos de operações que dependem da conexão de rede entre cliente e servidor de interface, representadas pelos SLIs Trd1 e Trd2. As medições obtidas neste cenário para estes indicadores desviaram-se em 1870% e 130% do SLO. O desvio foi expressivamente maior para o indicador Trd1 em relação ao Trd2 porque o arquivo (transmitido conjuntamente com o formulário de mensagem para o servidor de interface) utilizado nos experimentos foi de 3,5MB e a operação medida por Trd1 contempla todo o tempo da operação (incluindo a função de envio do formulário e de transmissão do arquivo). Já em Trd2, apenas um página de confirmação é enviado do servidor de interface para o cliente.

No cenário C o módulo de governança identificou que as operações com maior desvio do SLO são operações dependentes de conexão de rede e pode confirmar que a responsabilidade do descumprimento do SLA foi externa à nuvem.

Avaliando-se conjuntamente o resultado nos cenários B e C percebe-se que o módulo de governança pode identificar causas distintas para ocorrências de não conformidade com o SLA nas operações onde foi identificado maior tempo de resposta.

## 6. Conclusão

A definição de um SLA exprime a expectativa de clientes e contratantes e os compromissos dos provedores de serviços em nuvem. Entretanto, a qualidade dos serviços percebida pelo cliente não é afetada unicamente pelo provedor. Fatores

 ©2012 SBC — Soc. Bras. de Computação

externos a nuvem e o próprio comportamento dos contratantes e clientes podem impactar em seu funcionamento.

A simples extração de informações de consumo de recursos infraestrutura por parte do provedor entrega uma visão parcial e não o isenta de conflitos de interesse sobre a entrega do serviço. As medições de SLIs extraídas pelo provedor podem não corresponder à forma como o usuário percebe e avalia o serviço. O usuário avalia o serviço através da sua experiência de uso, QoE, que é afetada pelo desempenho de uma cadeia de provedores, incluindo o próprio provedor da nuvem, e as operadoras que fornecem conexão do usuário à Internet.

A utilização de SLIs baseados em QoE podem auxiliar na identificação do componente onde há degradação de nível serviço e quem seja causador de descumprimento de SLA, mas, esta informação é visível parcialmente ao cliente e parcialmente ao contratante. Desta forma, informações, que eventualmente podem identificar gargalos em operações fora da nuvem, retirando a responsabilidade sobre as inconformidades de SLA do provedor, também não são visíveis a esse.

Através da geração dos indicadores para cada operação, torna-se possível combinar informações oriundas do cliente, contratante e provedor, obtendo-se uma única visão sobre a entrega de serviços e eventual causa sobre desvios de SLA.

O trabalho mostrou que para evitar conflitos de interesse entre cliente, contratante e provedor deve haver a participação de uma entidade neutra, confiável e independente, obtendo e relacionando as informações obtidas destes tornando todas as partes auditáveis, promovendo assim a auditabilidade multipartes. Para isso, a entidade de auditoria deve ter acesso às informações diretamente dos ambientes do provedor, do contratante e do cliente, consolidá-las em ambiente externo e livre de interferências.

A arquitetura de auditoria considerou que cliente, contratante e provedor estejam cientes de sua execução, uma vez que é necessária a adição de *plug-in* no lado do cliente e do contratante, e a disponibilização do código coletor no provedor, especificamente na camada de virtualização. Este foi o motivo que inviabiliza a execução de testes do protótipo em nuvem pública. Porém, entende-se que é interesse das partes auditarem-se entre si, quando isto acontecer recursos como os comentados neste trabalho deverão se tornar comuns e tais limitações não serão mais identificadas.

Os testes realizados em nuvem privada evidenciaram a capacidade do protótipo identificar o componente responsável pelo desvio, de acordo com a operação afetada, tornando possível atribuir responsabilidade sobre o descumprimento de SLA. Adicionalmente, a utilização dos auditores de infraestrutura e de aplicação, a arquitetura permite o desenvolvimento de novos inspetores e auditores, associados a outros SLIs.

Ainda que a tendência de utilização de computação em nuvem venha se intensificando no ambiente empresarial, a baixa confiança no nível de serviço obtido do provedor ainda é uma barreira de entrada para muitas empresas. Porém, a criação de serviços que sejam auditáveis em multipartes, somada a existência de entidades de auditoria de reconhecida boa reputação pode aumentar o nível de confiança dos futuros contratantes de computação em nuvem.

## 7. Referências

BARBOSA, AC; SAUVÉ, J.; CIRNE, W.; CARELLI, M. (2006). Evaluating architectures for independently auditing service level agreements, Elsevier FGCS, 22, 7, 721-731.

 ©2012 SBC — Soc. Bras. de Computação

BUYYA, R.; YEO CS.; VENUGOPAL, S. (2008). Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, High Performance Computing and Communications. Proc. HPCC, pp. 25-27.

CHEN, Y. PAXSON, V., Katz R.H. (2011). New About Cloud Computing Security?, University of California at Berkeley, disponível em <eecs.berkeley.edu>. Acesso Fev. 2012.

CHOW, R.; GOLLE, P.; JAKOBSSON, M.; SHI, E.; STADDON, J.; MASUOKA, R.; MOLINA, J. (2009). Controlling data in the cloud: outsourcing computation without outsourcing control. Proceedings of ACM CCSW. ACM, pp. 85-90.

COMUZZI, M; KOTSOKALIS, C.; SPANOUDAKIS, G.; YAHYAPOUR, R. (2009). Establishing and Monitoring SLAs in complex Service Based Systems. ICWS, pp.783-790.

CSA GRC STACK RESEARCH GROUP. (2010). Cloud Security Alliance GRC Stack, disponível em: <cloudsecurityalliance.org/research/projects/grc-stack>. Acesso: Jan. 2012.

CSA RESEARCH GROUP. (2010). Guia de Segurança para Áreas Críticas Focado em Computação em Nuvem, disponível: <cloudsecurityalliance.org/guidance>. Acesso em Dez. 2012.

EMEAKAROHA, VC.; BRANDIC, I.; MAURER, M.; DUSTDAR, S. (2010). Low Level Metrics to High Level SLAs - LoM2HiS Framework: Bridging the GapBetween Monitored Metrics and SLA Parameters in Cloud Environments. Proc. HPCS, pp.48-54.

FIEDLER, M.; HOSSFELD, T.; PHUOC, TG. (2010). A generic quantitative relationship between quality of experience and quality of service, Network, IEEE , 24, 2, pp. 36-41.

GUO, Z.; SONG, M.; SONG, J. (2010). A Governance Model for Cloud Computing, Proc. MASS, pp.1-6.

HAEBERLEN, A. (2010). A Case for the Accountable Cloud, ACM SIGOPS Operating Systems Review, 44, 2, pp. 52-57.

HUBBARD, D; SUTTON, M. (2012). Top Threats to Cloud Computing, Cloud Security Alliance, disponível em: <cloudsecurityalliance.org/topthreats>. Acesso em Fev. 2012.

IDC. (2010). It cloud services user survey, pt.2: Top benefits & challenges, disponível em: <blogs.idc.com/ie/?p=210>. Acesso em Dez. 2011.

JANSEN, W.; GRANCE, T.(2011). Guidelines on Security and Privacy in Public Cloud Computing, disponível em: <nist.gov.br>. Acesso em Nov. 2011.

LANDWEHR, C.E. (2001). Computer Security**,** International Journal of Information Security, Springer Berlin / Heidelberg, pp. 3-13.

MELL, P; GRANCE T.(2011). The NIST Definition of Cloud Computing**,** disponível em <nist.gov>. Acesso em Out. 2011.

MOORSEL, AV. (2001). Metrics for the Internet Age: Quality of Experience and Quality of Business, disponível em: <www.hp.com>. Acesso em Mai 2012.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. (2009). The Eucalyptus Open-Source Cloud-Computing System. Proc. of IEEE/ACM CCGRID. IEEE Computer, pp. 124-131.

OPEN CLOUD STANDARDS INCUBATOR – DMTF. (2010). Architecture for Managing Clouds, disponível em: <www.dmtf.org >. Acesso em Jan. 2012.

SAUVÉ, J., MARQUES, F.; MOURA, A.; SAMPAIO, M.; JORNADA, J.; RADZIUK, E.; SCHÖNWÄLDER, J. (2005). SLA design from a business perspective, Ambient Networks - Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 72-83.

SKENE, J.; RAIMONDI, F.; EMMERICH, W. (2010). Service-Level Agreements for Electronic Services, IEEE Transactions on Software Engineering, 36, 2, pp. 288-304.

WLOKA J., SRIDHARAN, M., TIP, F. (2009). Refactoring for reentrancy. Proc. ACM SIGSOFT and ESEC/FSE. ACM, pp. 173-182.

 ©2012 SBC — Soc. Bras. de Computação

# Identificação de Autoria de Documentos Eletrônicos

**Walter R. de Oliveira Jr.[1], Luiz E. S. Oliveira[2], Edson J. R. Justino[1]**

[1]PRograma de Pós-Graduação em Informática – Pontifícia Universidade Católica do Paraná (PUCPR)
Rua Imaculada Conceição, 1155 – 80.215-901 – Curitiba – PR – Brasil

[2]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba, PR – Brasil

woliveirajr@gmail.com, justino@ppgia.pucpr.br, lesoliveira@inf.ufpr.br

***Abstract.*** *The identification of authorship of electronic documents lies among the demands of forensic analysis. Data compressors and the Normalized Compression Distance (NCD) are tools that can help the expert to perform this task. In this work the performance of these tools were analyzed in a dataset of 3,000 electronic documents in Brazilian Portuguese, from 100 different authors, and correct attribution was made in more than 70% of the cases, indicating that this approach might have promising results. It was also verified the influence of the number of training documents in the results.*

***Resumo.*** *Entre as demandas das perícias forenses está a identificação da autoria de documentos eletrônicos. Compressores de dados e a Distância Normalizada de Compressão (NCD) são ferramentas que auxiliam o perito a executar esta tarefa. Estudou-se o desempenho destas ferramentas em uma base de dados com 3000 documentos eletrônicos em português, de 100 diferentes autores, e foram obtidas taxas médias de acerto superiores a 70%, indicando que esta técnica é promissora. Verificou-se, também, a influência da quantidade de documentos de treinamento no desempenho desta técnica.*

## 1. Introdução

Entre as tarefas de perícia desenvolvidas em relação a documentos está a determinação de autoria de documentos. Nesta atividade busca-se determinar quem produziu um documento a partir da análise de elementos deste documento. Em relação a documentos eletrônicos, entretanto, pode haver dificuldades pela ausência de determinados elementos físicos que auxiliam o perito em sua tarefa. Por exemplo, não estão presentes elementos como a tinta utilizada para a escrita, a maneira como o autor grafa determinadas letras ou mesmo desgastes físicos de peças de impressoras ou máquinas datilográficas que deixem marcas em uma folha de papel.

Poderá ocorrer, também, que elementos comuns em documentos eletrônicos (como metadados de processadores de texto) não possam ser obtidos, e desta forma o perito tenha que utilizar apenas o conteúdo do documento para realizar sua perícia. Neste caso, o perito poderá utilizar a análise de características estilísticas do autor para a tarefa de atribuição de autoria. Considera-se que estas características estilísticas são produzidas de maneira inconsciente pelo autor e são pessoais, havendo variação da maneira como cada autor expressa suas ideias e pensamentos (Stamatatos, 2009 e Pinker, 2007).

 ©2012 SBC — Soc. Bras. de Computação

Existem diversas características estilométricas que podem ser analisadas pelo perito. Stamatatos (2009) dividiu-as em características de caracteres, léxicas, sintáticas e específicas a atividades (são as características que consideram elementos de atividades específicas, por exemplo maneira como o autor formata um determinado conteúdo de um documento).

O uso de compressão de dados para a verificação de autoria é uma análise que utiliza características de caracteres, pois os compressores de dados utilizam informações sobre os símbolos encontrados para diminuir a quantidade de símbolos necessários para representar os mesmos dados (no caso de compressores sem perdas) ou os dados degradados em um nível controlado (no caso de compressores com perdas).

No presente trabalho são utilizados os compressores de dados para a tarefa de atribuição de autoria a documentos eletrônicos. São comparadas duas abordagens que utilizam compressores de dados. A primeira utiliza a distância normalizada de compressão (*normalized compression distance* - NCD), proposta por Cilibrasi e Vitányi (2005), e a segunda abordagem utiliza a complexidade condicional de compressão (*conditional complexity of compression* – CCC), proposta por Malyutov et al. (2007). Estas abordagens são detalhadas na seção 2.

A base de dados utilizada neste trabalho é a mesma que Varela (2010) utilizou em um trabalho prévio. Esta base de dados é melhor explicada no seção 3. Naquele trabalho, Varela (2010) atribuía a autoria de documentos eletrônicos com o uso de classificadores SVM e estatísticas de palavras-função presentes nos documentos. Por exemplo, para cada autor eram geradas estatísticas sobre os pronomes utilizados em seus documentos, e os perfis de cada autor eram considerados em um classificador SVM para a atribuição de autoria.

O método proposto para o presente trabalho é apresentado no seção 4, onde detalha-se como a NCD e a CCC foram utilizados para atribuição de autoria.

No seção 5 são apresentados os resultados obtidos. Como mencionado, o fato de utilizar a mesma base de dados de outro trabalho e buscar-se o desempenho da mesma tarefa, é possível efetuar uma comparação dos resultados obtidos por Varela (2010).

Por fim, no seção 6 serão apresentadas as conclusões obtidas com este trabalho e sugestões para trabalhos futuros.

## 2. Distância Normalizada de Compressão e Complexidade Condicional de Compressão

O uso de compressores para a verificação da autoria de documentos decorre da teoria de Complexidade de Kolmogorov (1965). De maneira resumida, a teoria da Complexidade de Kolmogorov afirma que a complexidade de uma determinada informação pode ser medida pela quantidade de símbolos necessários para expressar esta informação em alguma linguagem universal. Logo, a complexidade de Kolmogorov $K()$ de uma informação $x$ é representada pelo tamanho da representação desta informação em uma linguagem universal.

É possível verificar a complexidade condicional de Kolmogorov $K(x|y)$. Dada uma informação $x$, sua complexidade condicional será o tamanho do programa $y$ que é processado por uma máquina de Turing e gera a saída $x$ novamente.

Estas complexidades são incomputáveis, pois não é possível saber se o menor

 ©2012 SBC — Soc. Bras. de Computação

programa *y* foi gerado (Li e Vitányi, 1997).

### 2.1. Distância Normalizada de Compressão

A partir da teoria de Kolmogorov, Li e Vitányi (1997) propuseram que é possível verificar a similaridade entre duas informações, e que era possível apresentar esta similaridade em uma medida normalizada. A Distância Normalizada de Informação (NID – *Normalized Information Distance*) é expressa na equação 1 a seguir

$$NID(x, y) = \frac{max\{K(x|y), K(y|x)\}}{max\{K(x), K(y)\}} \quad (1)$$

sendo que *x* e *y* são as informações consideradas, *K(x|y)* é a complexidade condicional de Kolmogorov, *K(x)* é a complexidade de Kolmogorov da informação *x* e *max{}* é a função que retorna o maior entre dois valores.

Como *K(x)* é incomputável, a NID também o é.

Os mesmos autores propuseram que é possível aproximar a complexidade de Kolmogorov ao se utilizar compressores de dados. Para comprimir uma informação, o compressor processa esta informação e busca, através do uso de uma codificação otimizada, gerar um programa *y* que possa ser novamente processado e que gere a informação *x* como saída, e que este programa *y* tenha um tamanho menor que a informação *x* original.

Com o uso de compressores, é possível estabelecer uma Distância Normalizada de Compressão (NCD – *Normalized Compression Distance*), expressa na equação 2 a seguir.

$$NCD(x, y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \quad (2)$$

sendo que *min{ }* é a função que retorna o menor valor entre dois valores dados, *C(x)* é o tamanho da informação *x* após ser comprimido por um compressor *C*, *xy* é a concatenação da informação *x* com a *y* e os demais elementos já foram definidos anteriormente.

A NCD é uma medida normalizada de similaridade entre duas informações e seu resultado, idealmente, deveria estar entre 0 e 1. Cilibrasi e Vitányi (2005) mencionam que, para determinados compressores de dados, o valor obtido poderá ser ligeiramente superior a 1, quando pouca ou nenhuma compressão for obtida e o compressor não for otimizado para esta situação, utilizando uma grande quantidade de informações como cabeçalho da compressão. Valores de NCD próximos a 0 indicam uma maior similaridade entre duas informações, e idealmente a *NCD(x,x)* deveria ser exatamente 0, pois a similaridade entre duas informações idênticas é total.

Destaca-se que a compressão de dados para cálculo da NCD é feita utilizando-se o documento completo, tal como ele se apresenta. Não é necessário realizar a extração ou pré-seleção de características específicas a serem utilizadas. Isto é um diferencial, por exemplo, em relação ao uso de SVM como classificador, pois para a utilização do SVM requer que características sejam selecionadas para a construção dos vetores de suporte.

### 2.2. Complexidade Condicional de Compressão

Malyutov et al. (2007), também fundamentando-se na complexidade de Kolmogorov e

 ©2012 SBC — Soc. Bras. de Computação

na sua aproximabilidade com o uso de compressores de dados, propuseram a medida da Complexidade Condicional de Compressão (CCC – *Conditional Complexity of Compression*).

Esta CCC é expressa na equação 3 a seguir.

$$CCC(x|y)=C(yx)-C(x) \tag{3}$$

sendo que *yx* é a concatenação da informação *y* com a informação *x* e os demais elementos já foram definidos anteriormente.

Para a medida de similaridade de informações, um valor menor de CCC indica que as duas informações concatenadas puderam ser comprimidas melhor do que outras duas informações que apresentem um CCC maior, e portanto o compressor de dados conseguiu gerar um modelo melhor de compressão por ter encontrado elementos mais semelhantes entre as duas informações.

Nesta medida também não é feita a pré-seleção ou extração de características, pois os documentos a serem testados são considerados em sua integralidade.

### 2.3. Compressores de dados

Para aproximação da complexidade de Kolmogorov são utilizados compressores de dados. Não há, entretanto, determinação sobre qual compressor deve ser utilizado pelos métodos NCD e CCC, então foram utilizados três compressores de dados, com três métodos de compressão diferentes: método baseados em estatísticas, em dicionários e em blocos de dados.

O compressor estatístico utilizado foi o PPM-D (*prediction by partial matching D*). Este compressor gera um modelo estatístico dos símbolos encontrados no documento considerando a probabilidade da ocorrência de um símbolo em função da ocorrência anterior de outros símbolos (Shkarin, 2002). Por exemplo, na língua portuguesa, após um símbolo "ç", há uma maior probabilidade de encontrar um símbolo "ã" ou "õ" do que um símbolo "r", e estas probabilidades são consideradas para a geração de modelos estatísticos otimizados.

Para compressores baseados em dicionários foi escolhido o compressor Zip. Este compressor implementa o algoritmo de compressão baseado em dicionário LZ77 (Lempel-Ziv77) e uma codificação de Huffmann. De maneira simplificada, os compressores baseados em dicionário utilizam as informações visualizadas anteriormente no documento como uma tabela de dicionário, e quando esta informação é encontrada novamente, é feita apenas uma referência à tabela.

Por fim, o compressor baseado em bloco de dados utilizado é o compressor Bzip, que implementa o algoritmo de compressão de blocos de Burrows e Wheeler (Burrows e Wheeler, 1994). Este compressor caracteriza-se por separar o conteúdo a ser comprimido em blocos de dados e efetuar operações em seu conteúdo, de maneira a agrupar informações semelhantes e assim conseguir representá-las de uma maneira que possam ser codificados mais eficientemente por alguma codificação posterior. No caso do compressor Bzip, é utilizada a codificação de Huffmann.

## 3. Base de dados

Conforme menciona Stamatatos (2009), uma característica desejável em uma base de

 ©2012 SBC — Soc. Bras. de Computação

dados para pesquisas de autoria de documentos é que a autoria dos documentos seja previamente conhecida, permitindo que os resultados obtidos possam ser conferidos e a taxa de acerto de um determinado método possa ser verificada. Também é desejável que testes sejam realizados sobre as mesmas bases de dados, controladas, com protocolos iguais ou semelhantes. Desta maneira os resultados de diferentes métodos podem ser comparados, permitindo uma verificação de ganhos de um determinado método em relação a métodos utilizados anteriormente.

Neste trabalho foi usada a mesma base de dados utilizada anteriormente por Varela (2010). Esta base de dados é composta por 3.000 documentos, de 100 diferentes autores, com 30 documentos de cada autor. Estes autores e documentos estão separados em 10 categorias: direito, economia, esportes, gastronomia, literatura, politica, saúde, tecnologia, turismo e assuntos gerais. O tamanho médio dos documentos é de 2989 bytes com um desvio padrão de 1531, com um valor máximo de 735 *tokens* (palavras) em um documento.

Para os testes os documentos foram separados em dois grupos. O primeiro grupo era composto pelos documentos de perfil dos autores. Cada autor era representado por 7 documentos, escolhidos aleatoriamente, em um total de 70 documentos por área. O segundo grupo compunha o conjunto de testes e era composto pelos documentos restantes, com 23 documentos sendo utilizados para cada autor em um total de 230 documentos testados por área, em um total de 2300 documentos testados.

## 4. Método

Para a atribuição de autoria dos documentos de teste foi utilizado o seguinte método. Inicialmente os documentos de cada área foram separados em dois grupos (perfil do autor e documentos de teste), conforme mencionado na seção anterior.

A seguir foram efetuados testes com os dois métodos de cálculo de similaridade entre documentos: NCD e CCC. Para cada um dos métodos foram utilizados os três compressores de dados mencionados: Zip, Bzip e PPM-D.

Para cada teste, o documento testado tinha sua medida NCD ou CCC efetuada em relação a todos os documentos de perfil de autor daquela área. Ou seja, para cada documento testado eram obtidas 70 medidas, sendo que cada autor era representado por 7 medidas. Para cada teste, em cada área de conhecimento, eram possíveis 10 autores, e a probabilidade de um documento ser atribuído aleatoriamente a um autor era de 10%.

Após obtidos os valores NCD e CCC de cada teste, era necessário utilizar algum mecanismo de escolha do resultado para a atribuição de autoria ao documento. Neste trabalho foram utilizados dois mecanismos de escolha: melhor valor e votação.

A escolha pelo melhor valor é feita considerando-se o melhor valor de cada teste. Por exemplo, um documento $x$ era testado em relação a 7 documentos do autor A, 7 documentos do autor B, e assim por diante, gerando um total de 70 medidas. Dentre estas medidas, verificava-se qual era o melhor resultado, e a atribuição de autoria era feita ao autor cujo documento de perfil apresentou o melhor resultado.

A escolha por votação, por sua vez, é feita considerando-se o número de votos que cada autor recebeu. Determinou-se que seriam computados como votos os sete melhores resultados de medida, e a atribuição de autoria era feita ao autor mais votado. Por exemplo, supondo que um documento $x$ teve as 7 melhores medidas em relação aos

 ©2012 SBC — Soc. Bras. de Computação

autores "B, A, A, C, D, A, B". O autor mais votado foi o autor *A*, e a autoria era atribuída a ele. Em caso de empate, a autoria foi atribuída ao autor que estivesse com o melhor resultado. Por exemplo, se os votos fossem para os autores "B, A, C, A, B, D, E", os autores *B* e *A* teriam a mesma quantidade de votos e o autor *B* seria escolhido.

## 5. Resultados e Análise

São apresentados, a seguir, os resultados dos testes efetuados.

### 5.1. Escolha pelo melhor resultado

O primeiro teste efetuado utilizou a atribuição de autoria pelo melhor resultado dos métodos CCC e NCD, com os três compressores de dados mencionados.

Os testes eram efetuados entre os documentos de cada área, havendo 10 autores prováveis para cada documento testado. Foram obtidos os seguintes resultados, expressos na tabela 1. A coluna SVM refere-se aos resultados obtidos por Varela (2010) utilizando um classificador SVM e estatísticas de palavras-função encontradas no documento.

**Tabela 1. Escolha de autor pelo melhor resultado**

| | | NCD | | | CCC | | |
|---|---|---|---|---|---|---|---|
| Área | SVM | Bzip | PPM-D | Zip | Bzip | PPM-D | Zip |
| Assuntos Variados | 70,70% | 79,57% | 81,74% | **83,04%** | 69,70% | 72,16% | 73,02% |
| Direito | **72,20%** | 63,91% | 68,26% | 65,65% | 52,75% | 60,75% | 57,32% |
| Economia | 64,80% | 77,83% | 78,70% | **79,57%** | 68,63% | 67,47% | 72,77% |
| Esportes | 68,30% | 82,61% | 85,65% | **87,39%** | 75,74% | 74,10% | 80,78% |
| Gastronomia | **75,70%** | 44,78% | 54,35% | 53,04% | 34,48% | 45,80% | 47,51% |
| Literatura | **72,20%** | 59,13% | 66,96% | 61,74% | 47,27% | 56,97% | 50,69% |
| Política | 68,70% | 81,74% | **83,91%** | 83,04% | 71,02% | 72,84% | 77,38% |
| Saúde | **72,20%** | 58,26% | 61,30% | 63,91% | 50,07% | 54,72% | 52,57% |
| Tecnologia | 73,90% | 74,78% | 77,39% | **79,13%** | 64,75% | 67,04% | 70,29% |
| Turismo | 78,30% | 80,00% | 82,17% | **83,04%** | 69,41% | 72,63% | 73,95% |
| Média | 71,70% | 70,26% | **74,04%** | 73,96% | 60,38% | 64,45% | 65,63% |
| Desvio padrão (pontos percentuais) | 3,86 | 12,91 | 10,67 | 11,75 | 13,38 | 9,53 | 12,27 |

Em negrito estão destacados os melhores resultados de cada área.

Verifica-se que para a escolha pelo melhor resultado, o método que apresenta uma maior taxa de acerto média é o NCD com o uso do compressor PPM-D. O NCD com o compressor Zip apresenta o melhor resultado em 5 categorias, seguido pelo método SVM.

O método CCC apresentou resultados inferiores de aproximadamente 12 pontos percentuais aos obtidos pelo método NCD. Em duas áreas (Tecnologia e Turismo) o método CCC teve resultado inferior ao classificador SVM enquanto o NCD apresentava um resultado superior. A figura 1 ilustra o desempenho médio e o desvio padrão dos

 ©2012 SBC — Soc. Bras. de Computação

métodos NCD e CCC com cada compressor.

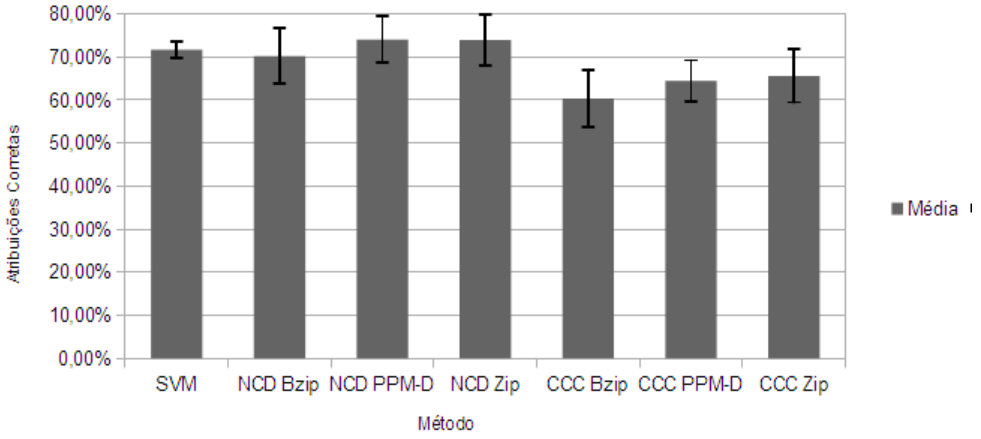



**Figura 1. Escolha de autor pelo melhor resultado**

A diferença de desempenho entre o método NCD com compressor PPM-D e com compressor Zip foi de aproximadamente 0,1 ponto percentual, e no gráfico esta diferença é imperceptível.

O método que utilizou o classificador SVM apresentou o menor desvio padrão, e conforme pode ser visto pelo desempenho em cada área, os seus resultados foram mais homogêneos.

## 5.2. Escolha por votação

O segundo teste efetuado utilizou a atribuição de autoria através de votação. Os resultados obtidos estão expressos na tabela 2 a seguir.

**Tabela 2. Escolha de autor por votação**

| | | NCD | | | CCC | | |
|---|---|---|---|---|---|---|---|
| Tema | SVM | NCD Bzip | NCD PPM-D | NCD Zip | Bzip | PPM-D | Zip |
| Assuntos Variados | 81,74% | **84,78%** | 83,04% | 79,57% | 69,72% | 73,64% | 62,76% |
| Direito | **72,20%** | 60,87% | 64,35% | 67,83% | 46,59% | 49,44% | 55,35% |
| Economia | 64,80% | **75,65%** | 70,00% | 74,78% | 64,98% | 59,53% | 63,20% |
| Esportes | 68,30% | 80,43% | **83,91%** | 83,04% | 69,59% | 75,95% | 74,06% |
| Gastronomia | **75,70%** | 50,87% | 48,26% | 51,74% | 38,23% | 33,35% | 38,66% |
| Literatura | **72,20%** | 61,30% | 59,13% | 57,83% | 46,91% | 43,28% | 46,07% |
| Política | 68,70% | 80,87% | 80,87% | **84,35%** | 72,13% | 69,09% | 72,99% |
| Saúde | **72,20%** | 60,87% | 63,48% | 66,09% | 51,07% | 55,32% | 54,74% |
| Tecnologia | 73,90% | 75,22% | 74,35% | **78,26%** | 61,20% | 61,25% | 70,12% |
| Turismo | 78,30% | 77,39% | 78,26% | **81,30%** | 66,94% | 64,69% | 69,16% |
| Média | 71,70% | 70,83% | 70,43% | **72,35%** | 58,74% | 58,56% | 60,71% |
| Desvio padrão (pontos percentuais) | 4,97% | 11,36% | 11,69% | 11,19% | 12,00% | 13,54% | 11,85% |

Em negrito estão destacados os melhores resultados de cada área. Observa-se

 ©2012 SBC — Soc. Bras. de Computação

que novamente o método CCC teve resultados inferiores ao método NCD, para qualquer dos compressores de dados considerados. Apesar do método NCD com compressor Zip ter tido a melhor média de atribuições corretas, seus resultados foram melhores que os demais métodos ou compressores apenas em 3 temas. O compressor PPM-D apresentou o melhor resultado em um tema e o compressor Bzip em dois temas.

Observa-se, também, que o método de escolha de resultado por votação apresenta um desempenho inferior à escolha pelo melhor resultado. A figura 2 ilustra o desempenho médio e o desvio padrão dos métodos NCD e CCC com cada um dos compressores.

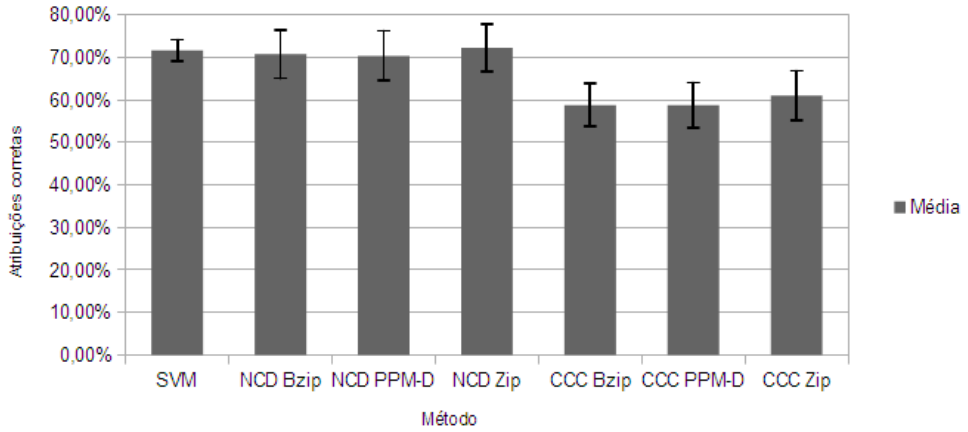



**Figura 2. Escolha de autor por votação**

### 5.3. Influência da quantidade de documentos do estilo do autor

Hipotetiza-se que com uma disponibilidade menor de documentos que representem o estilo de um autor o desempenho da atribuição de autoria apresente resultados inferiores. Para verificar a influência da diminuição da disponibilidade destes documentos de treinamento, foram efetuados testes com o método NCD, pois este apresentou os melhores resultados nos testes anteriores. Os três compressores de dados (Zip, Bzip e PPM-D) foram utilizados.

Os testes foram conduzidos nos documentos de apenas duas áreas de conhecimento: Economia e Saúde. Esta escolha é justificada pelo fato que o tema Economia teve um melhor desempenho quando o mecanismo de escolha era o melhor resultado e que o tema Saúde teve um melhor desempenho quando o mecanismo de escolha era a votação.

Os testes foram efetuados através da retirada, aleatória, dos documentos de perfil do autor, reduzindo-se a quantidade inicial de 7 documentos por autor até apenas 2 documentos por autor.

Os resultados para o tema Economia estão representados na figura 3, para o compressor Zip. Para os compressores Bzip e PPM-D os resultados obtidos são bastante semelhantes ao do compressor Zip.

 ©2012 SBC — Soc. Bras. de Computação

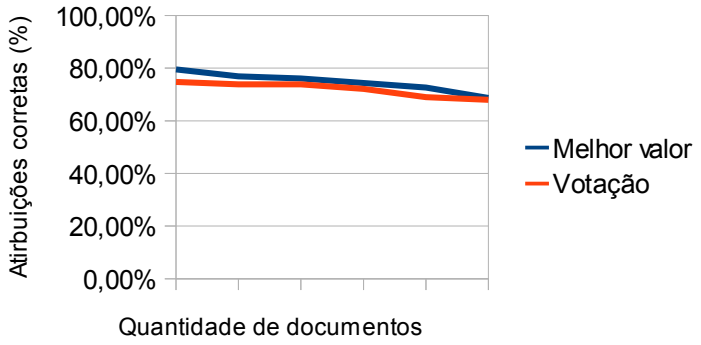



**Figura 3. Tema Economia, compressor Zip**

Como pode ser observado, há uma queda gradativa na taxa de atribuições corretas conforme a diminuição na quantidade de documentos que representam o estilo do autor, quando o método de atribuição é o do melhor valor de NCD.

A diminuição de atribuições corretas com a escolha sendo feita por votação é também gradativa. Quando existem apenas dois documentos de perfil do autor o resultado é idêntico à escolha pelo melhor resultado porque só restarão dois votos, e caso cada um destes votos seja dado a um autor diferente, o autor com melhor resultado é escolhido, e isto equivale a dizer que a escolha é feita pelo melhor resultado.

Os resultados para o tema Saúde estão representados na figura 4, para o compressor Zip. Para os demais compressores os resultados obtidos foram bastante semelhantes.

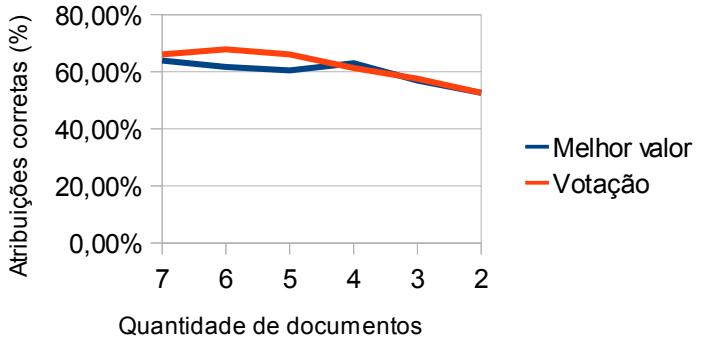



**Figura 4. Tema Economia, compressor Zip**

Como pode ser observado nos testes deste tema, a escolha pelo critério de votação apresentou um melhor resultado quando o perfil do autor era representado por 5 ou mais documentos. Com a diminuição da quantidade de documentos houve uma inversão de desempenho e o método de escolha pelo melhor resultado passou a apresentar um maior índice de atribuições corretas.

Uma das explicações possíveis para este comportamento é a existência de algum documento, dentro do tema, que possuísse características (por exemplo, uma grande quantidade de palavras que sejam comuns a alguns dos documentos testados) e isto fazia

 ©2012 SBC — Soc. Bras. de Computação

com que todos os documentos fossem considerados similares a ele. Com a sua remoção, este documento deixou de influir na atribuição de autoria, e a escolha pelo melhor valor da NCD passou a ser determinante.

Verifica-se, assim, que apesar do método de escolha pelo melhor resultado NCD apresentar o melhor desempenho na tarefa de atribuição de autoria, há uma suscetibilidade a documentos que representem mal o perfil de um autor, e que por diversos fatores possíveis todos os documentos apresentem uma melhor compressão (e consequentemente uma melhor NCD) quando concatenados a ele.

Também é possível observar que mesmo com a diminuição da quantidade de documentos que fornecem o perfil do autor, há um desempenho superior ao que seria obtido por uma atribuição aleatória. No tema Economia esta atribuição foi feita corretamente em mais de 60% dos casos, mesmo com apenas 2 documentos de cada autor sendo utilizados, sendo que uma atribuição aleatória teria uma taxa de sucesso de apenas 10%. Em relação ao tema Saúde este índice, para 2 documentos, foi superior a 40%.

## 6. Conclusões

A tarefa de atribuição de autoria a documentos eletrônicos pode apresentar desafios pela ausência de características físicas que possam auxiliar o perito no desempenho de sua tarefa.

O uso de técnicas computacionais pode ser relevante, automatizando algumas atividades ou fornecendo resultados objetivos, reprodutíveis, sobre a semelhança entre documentos. Os métodos mostrados neste trabalho mostram que bons resultados podem ser obtidos com o uso de compressores de dados, e que a teoria da complexidade de Kolmogorov é uma fundamentação teórica importante a embasar o desempenho visto.

A utilização de uma base de documentos já utilizada em trabalho prévio permite que os resultados obtidos possam ser comparados e assim seja possível a confrontação de características de cada uma das técnicas. No trabalho presente, foi possível verificar que para estes documentos o uso da NCD possui um desempenho superior ao obtido pela CCC e pelo uso de classificadores SVM, quando estes são utilizados com determinadas características estatísticas.

Entre as vantagens do uso da compressão de dados com a NCD ou com a CCC está a desnecessidade de uma seleção prévia de características do estilo do autor e a ausência da necessidade de um treinamento prévio da ferramenta. Desta forma, caso existam outros autores prováveis incluídos posteriormente, bastará efetuar o cálculo da NCD ou CCC em relação aos novos autores e confrontar com medidas obtidas anteriormente.

A possibilidade de escolha do compressor de dados a ser utilizado, havendo em alguns casos pequenas diferenças no resultado obtido, é relevante, pois cada compressor possui métodos diferentes de extração de características dos documentos. Também existem exigências diferenciadas quanto a capacidade computacional, necessidade de memória e complexidade de processamento, e fatores externos pode levar à necessidade que apenas um determinado compressor possa ser utilizado.

A comparação entre o desempenho obtido com diferentes quantidades de documentos representando o perfil dos autores prováveis mostra que há uma influência

 ©2012 SBC — Soc. Bras. de Computação

no desempenho da atribuição de autoria, mas que mesmo com poucos documentos ainda há uma taxa de atribuições corretas superior ao que seria obtido por uma atribuição aleatória.

Os métodos analisados neste artigo podem ser aperfeiçoados em trabalhos futuros. Sugere-se que mais pesquisas sejam feitas com diferentes compressores, inclusive considerando-se a hipótese de pré-processamento dos documentos. Por exemplo, deve ser verificado se a eliminação de alguns símbolos (como caracteres acentuados ou sinais de pontuação) permite uma melhor compressão dos documentos (gerando melhores medidas NCD) ou se esta diminuição afeta a representação do estilo de um autor.

## Agradecimentos

Esta pesquisa foi parcialmente apoiada pelo Conselho Nacional para Desenvolvimento Científico e Tecnológico (CNPq) – concessão nº 301653/2011-9.

## Referências

Cilibrasi, R. e Vitányi, P. M. B. (2005) "Clustering by compression", In *IEEE Transactions on Information Theory*, 51:4, pp. 1523–1545.

Kolmogorov, A. N. (1965) "Three approaches to the quantitative definition of information". In *Problems Inform. Transmission*, 1, pp. 1–7

Li, M. e Vitányi, P. M. B. (1997) "An Introduction to Kolmogorov Complexity and Its Applications", Springer, 2nd edition.

M. Burrows , D. J. Wheeler (1994) "A block-sorting lossless data compression algorithm" In *Technical Report 124*, System Research Center - Digital, Palo Alto.

Malyutov, M.B., Wickramasinghe, C.I. and Li, S. (2007): Conditional Complexity of Compression for Authorship Attribution, In *SFB 649 Discussion Paper No. 57*, Humboldt University, Berlin.

Pinker, S. (2007) "The stuff of thought: language as a window into human nature", Viking Adult, 1st edition.

Shkarin, D. (2002) "PPM: One step to practicality." In *Proceedings of the Data Compression Conference*, April 2-4, IEEE Computer Society, Washington, DC., USA., pp: 202-211.

Stamatatos, E. (2009) "A survey of modern authorship attribution methods" In *Journal of the American Society for Information Science and Technology*, Volume 60, Issue 3, pp. 538–556

Varela, P. J. (2010) “O uso de atributos estilométricos na identificação da autoria de textos”. Dissertação de Mestrado apresentada no Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Brasil.

 ©2012 SBC — Soc. Bras. de Computação

# Modelagem de Aliciamento de Menores em Mensagens Instantâneas de Texto

**Priscila L. L. Santin, Cinthia O. A. Freitas, Emerson C. Paraíso, Altair O. Santin**

Pontifícia Universidade Católica do Paraná (PUCPR) – Escola Politécnica – Programa de Pós-Graduação em Informática (PPGIa) – Curitiba – PR – Brasil

{priscila.santin, cinthia, paraiso, santin}@ppgia.pucpr.br

***Abstract.*** *The approaches presented in the literature are not suitable for detection of children sexual grooming, because the proposals are usually limited to the identification of stages of communication between the predator and the victim. These approaches are inefficient due to the use of a unified profile to identify a sexual grooming stage. Our proposal takes into account the identification of the communication's stage applying detached profiles, one for the victim and another for the predator. This approach, based on a stochastic technique (i.e. HMM), aims at the individual modeling of each profile to enhance the detection hit rate. The experiments achieved promising identification rates with results closer to 91%.*

***Resumo.*** *As abordagens existentes na literatura não são modeladas para detecção do aliciamento sexual de menores, mas apenas fazem a descoberta do estágio de aliciamento numa comunicação entre o agressor e sua vítima. Além disto, mostram baixa eficiência em função do emprego de um perfil único para a descoberta dos estágios. Este artigo considera a descoberta dos estágios de aliciamento com o perfil do agressor e da vítima separadamente. Esta abordagem, baseada em avaliação estocástica (i.e. HMM), visa a modelagem individual de cada perfil para aumentar a eficiência da detecção. Os experimentos mostram taxas de acerto promissoras com resultados próximos a 91%.*

## 1. Introdução

A *Internet* oferece inúmeros recursos para interação entre indivíduos das mais variadas faixas etárias. Atualmente, crianças (pessoas de até 12 anos de idade incompletos) e adolescentes (pessoas entre 12 e 18 anos) tomam contato com a rede mundial de computadores ainda nos seus primeiros anos de vida. A partir do uso rotineiro da *Internet,* estes menores de idade vão assimilando os diferentes tipos de recursos à medida que passam pelas diferentes fases de seu crescimento. A interatividade fornecida por recursos como salas de bate-papo (*chats*) traz para os menores uma espécie de local onde encontram todas as respostas para as suas dúvidas e curiosidades, porém, em suas visões, sem se expor a "censura" tradicional dos familiares.

Indivíduos mal intencionados conhecem esta tendência comportamental das crianças e adolescentes e usam as salas de bate-papo para o aliciamento (*grooming/enticement*) de menores com o intuito de praticar abuso sexual [Rashid 2008].

*Este trabalho é parcialmente apoiado pela Rede Marista de Solidariedade – Projetos Sociais Nº 01/2010.*

 ©2012 SBC — Soc. Bras. de Computação

Nos Estados Unidos (EUA), o Centro Nacional para Crianças Desaparecidas e Exploradas (*National Center for Missing and Exploited Children – NCMEC*) recebe informações de casos de exploração sexual de menores, incluindo o aliciamento *online*. Nesta categoria de aliciamento *online*, desde 9 de março de 1998 até o final de 2011 já haviam sidos informados 54.492 casos ao NCMEC, sendo que em 2011 a média foi de 77 casos por semana [NCMEC 1984].

No *site* brasileiro sobre a Campanha Nacional de Combate à Pedofilia na *Internet*[1] encontra-se que: "Também constam dados de uma pesquisa realizada nos EUA, dizendo que de cada 5 crianças que navegam na *Internet*, uma recebeu proposta de um pedófilo, e uma a cada 33 já se comunicou, através de telefone e recebeu dinheiro ou passagem para se encontrar com um criminoso". O *site* ainda alerta: "Pais e filhos, inconscientes dos perigos da rede são presas fáceis de pedófilos. Uma criança ingenuamente não identifica um adulto se passando por um amiguinho da mesma idade".

Em 2006, o Brasil já ocupava o terceiro lugar entre os países do mundo em número de denúncias de crimes *online* que ferem os direitos humanos, tais como: racismo, intolerância religiosa e pornografia infantil. Nesta época, tinham ocorrido em 20 dias um total de 2,25 mil denúncias, como declarado por Thiago Tavares, presidente da SaferNet, uma organização não-governamental (ONG) que luta contra crimes virtuais, durante a cerimônia de assinatura do acordo entre a entidade e o Ministério Público Federal (MPF) no dia 29 de março de 2006 em São Paulo[2]. Em janeiro de 2012, a violência de abuso sexual de menores (crianças e adolescentes), segundo a SaferNet[3], representava 60% dos casos reportados.

Diante desta problemática, pesquisadores têm estudado e desenvolvido técnicas para analisar conversas em salas de bate-papo [Kontostathis et al. 2009a] [Ho e Watters 2004] [Kontostathis et al. 2009b] [McGhee et al. 2011], de modo a auxiliar na descoberta dos perfis de agressores e no estabelecimento e identificação dos estágios de comunicação que estes agressores utilizam no contato com menores. A definição de tais perfis tem o intuito de contribuir para detectar a agressividade do aliciamento sexual e consequente a susceptibilidade ao abuso.

O trabalho de Kontostathis [Kontostathis et al. 2009a] relata que a identificação dos estágios de aliciamento é prejudicada quando considerado o uso do modelo do agressor e da vítima em conjunto. Diante disso, tem-se como objetivo estudar a correlação entre os dois modelos com intuito de superar esta limitação. Assim, este artigo propõe um método para análise de mensagens instantâneas de texto, realizadas em salas de bate-papo na *Internet,* visando a modelagem do perfil do agressor e da vítima separadamente. Esta modelagem é baseada na identificação dos estágios do diálogo entre um agressor e sua possível vítima. Esta abordagem permite detectar a correlação (*match*) entre os estágios de cada perfil com intuito de melhorar a precisão da detecção de aliciamento e ainda, de maneira análoga, inferir a susceptibilidade (probabilidades) a exposição das vítimas ao risco do abuso sexual.

Este artigo está organizado de tal forma que a Seção 2 traz uma introdução dos

[1] *http://www.censura.com.br/*

[2] *Disponível em: http://www.safernet.org.br/twiki/bin/view/SaferNet/Noticia20060329194811. Acesso em julho/2009.*

[3] *http://www.safernet.org.br/site/indicadores*

 ©2012 SBC — Soc. Bras. de Computação

conceitos fundamentais estudados para a realização do trabalho proposto. A Seção 3 apresenta os trabalhos relacionados. Na Seção 4 encontra-se descrito o método proposto para o desenvolvimento deste trabalho. A Seção 5 apresenta os testes experimentais e os resultados obtidos. Por fim, a Seção 6 aborda conclusões e trabalhos futuros.

## 2. Fundamentação

Esta seção apresenta a fundamentação para aliciamento e modelos estocásticos Markovianos.

### 2.1. Aliciamento

O termo de consenso para se referir ao aliciamento na área computacional é *grooming* ou *enticement* [Michalopoulos e Mavridis 2010]. Neste caso, o aliciamento sexual de menores se refere ao ato de preparar a vítima, por meio de salas de bate-papo, via troca de mensagens instantâneas para cometer um abuso sexual [Michalopoulos e Mavridis 2010].

Pelas características de interatividade e recursos que os meios computacionais atuais oferecem o *grooming/enticement* ocorre com muita frequência em salas de bate-papo. Neste caso, a detecção do aliciamento precisa ser feita *online*, avaliando as mensagens de texto que trafegam entre o possível agressor e a possível vítima [Michalopoulos e Mavridis 2010]. A detecção do aliciamento por meio da avaliação das mensagens de textos de conversas de salas de bate-papo pode ser executada identificando os estágios da conversa que o agressor utiliza para conquistar sua vítima com o objetivo de se relacionar sexualmente com a mesma. A seguir apresenta-se a definição dos estágios de comunicação para aliciamento encontrados na literatura e depois sua adaptação para a área de computação.

#### 2.1.1 Estágios do Aliciamento

Para entender o aliciamento deve-se entender como este ocorre, quais seus estágios e como estão caracterizados. Assim, com base nos estudos da teoria de comunicação sedutiva, Olson [Olson et al. 2007] propôs um modelo de comunicação que descreve o perfil do agressor identificando seus estágios.

Este modelo apresenta três fases mais relevantes: a persuasão (ganho de acesso) à vítima, o envolvimento da vítima numa relação enganosa e a iniciação e manutenção de um relacionamento sexualmente abusivo (Figura 1).

Olson mostra que a primeira fase, persuasão (ganho de acesso à vítima), tem três divisões. Duas referem-se à obtenção de características individuais do agressor e da vítima, que correspondem basicamente a informações como local de residência, idade, sexo etc. A terceira é o posicionamento estratégico – definido como um encontro de curto prazo, em *shopping* ou parques, ou um encontro de longo prazo, onde o agressor constrói um relacionamento com a vítima.

A segunda fase do modelo de comunicação de Olson trata de um ciclo para envolver a vítima numa relação enganosa. A preparação (*grooming/enticement*) consiste nas estratégias de comunicação que o agressor utiliza para que a vítima aceite o contato sexual. O desenvolvimento de uma verdade enganosa é entendido pelos autores deste modelo como a capacidade do agressor em cultivar um relacionamento de amizade com

 ©2012 SBC — Soc. Bras. de Computação

sua vítima, podendo também incluir seus familiares. Normalmente o agressor é visto como alguém importante ou uma autoridade, sendo que seu comportamento é pouco questionado pela vítima e seus parentes. Então vem o isolamento, que pode ser apresentado tanto como um isolamento físico, em que o agressor se propõe a tomar conta da vítima, como um isolamento mental, onde o agressor se mostra para a vítima como a única pessoa em quem essa pode confiar. Finalizando este ciclo tem-se a aproximação, que é definida por Olson [Olson et al. 2007], como o contato físico inicial ou introdução verbal que ocorre antes do ato sexual.

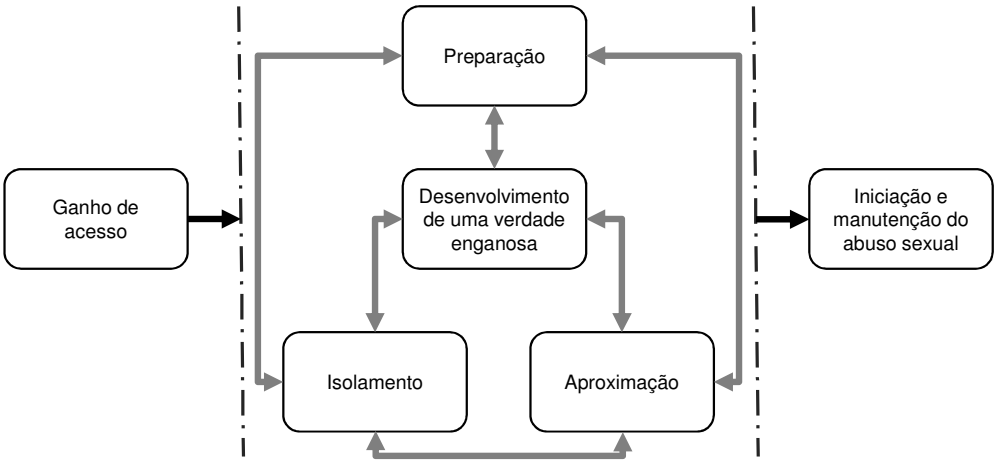



**Figura 1 - Adaptação do Modelo de Comunicação [OLS07]**

Na terceira fase de Olson tem-se a iniciação e manutenção do abuso sexual. A iniciação consiste em realmente praticar o ato sexual. Então, torna-se importante para o agressor o sigilo da vítima. Normalmente os agressores obrigam as vítimas ao silêncio coagindo-as ou simplesmente convencendo-as de que ao relatar o abuso não serão acreditadas ou até mesmo poderão causar um desconforto familiar.

A seguir é mostrado como as fases do modelo de Olson foram adaptados para compor os estágios do aliciamento sexual de menores em ambientes virtuais.

### 2.1.2 Estágios do Aliciamento para o Contexto Virtual

Leatherman [Leatherman 2009] fez uma adaptação para o contexto virtual do modelo de comunicação proposto por Olson [Olson et al. 2007]. O resultado de seu trabalho são os estágios descritos sumariamente a seguir:

- Acesso: envolve a troca de informações pessoais tanto do agressor quanto da vítima. No aliciamento *online* o agressor ganha acesso (*persuade*) à vítima usando recursos baseados em mensagens instantâneas ou salas de bate-papo *online* e *sites* de redes sociais;
- Desenvolvimento de um confiança enganosa: refere-se a capacidade do agressor cultivar relacionamentos com potenciais vítimas e possivelmente com suas famílias para se benficiar em seus interesses sexuais. No ambiente *online* essa confiança é praticada com base na utilização de quatro recursos:
    - ✓ *Informações Pessoais*: consiste na obtenção de detalhes sobre a vítima – idades, nomes, data de aniversários, fotos etc;
    - ✓ *Informações de Relacionamento*: considera a discussão de sentimentos e atitudes em relação à manutenção, construção e dissociação de relações com amigos, outras pessoas importantes e membros da família;

 ©2012 SBC — Soc. Bras. de Computação

  - ✓ *Atividades*: é o recurso mais utilizado pelo agressor para se aproximar da vítima compartilhando com esta preferências por músicas, filmes, livros, esportes, *hobbies*, comidas favoritas etc;
  - ✓ *Elogios*: consiste em envolver a vítima elogiando sua aparência, as atividades que desenvolve e caracteríticas pessoais.
- Preparação: compreende estratégias sutis de comunicação que os agressores utilizam para preparar suas potenciais vítimas a aceitar aspectos de conotação sexual. Conquistando a confiança da vítima, o agressor começa a prepará-la para aceitar ofertas de contato sexual, e também tornar a vítima menos sensível a comentários sexuais e linguagens obcenas. Há duas etapas de preparação para o aliciamento:
  - ✓ *Dessensibilização Comunicativa*: refere-se ao uso frequente de linguagem sexual vulgar, utilizada pelo agressor para dessensibilizar a vítima sobre seu uso;
  - ✓ *Reenquadramento*: ocorre quando o agressor se esforça para tornar a vítima confortável com experimentos sexuais por meio da *Internet*. Para este fim, a conversa sexual é apresentada de uma forma positiva e muitas vezes referida como uma experiência de aprendizagem, um jogo ou uma habilidade importante para aprender a fim de participar de relacionamentos amorosos futuros.
- Isolamento: pode envolver isolamento físico ou mental. O isolamento físico é definido como a dedicação de um tempo para ficar a sós com a vítima, já o isolamento mental é definido como o aumento da dependência que a vítima tem do agressor com relação a amizade e orientação. O agressor consegue o isolamento quando a vítima está em salas de bate-papo sem supervisão;
- Aproximação: o agressor tenta a abordagem da vítima, sugerindo encontros para fins sexuais. Esta é a etapa final no modelo *online*.

A caracterização destes estágios é de suma importância para o entendimento do modo de agir dos agressores visando o aliciamento sexual, bem como, para o desenvolvimento de sistemas computacionais para detecção deste tipo de problema. Considera-se que qualquer abordagem preventiva é relevante, pois na maioria dos casos no mundo real a abordagem é somente punitiva. Neste contexto, as ferramentas computacionais podem atuar preventivamente e auxiliar na detecção antecipada da intenção do abusador (agressor).

### 2.2. Modelos Escondidos de Markov

Os Modelos Escondidos de Markov (HMM – *Hidden Markov Models*) possuem habilidade para modelar eficientemente diferentes fontes de conhecimento, ou seja, permitem a integração entre dois diferentes níveis de modelagem: estrutural e probabilística. Para a modelagem de aliciamento tal característica é essencial, pois permite a entrada de sequências de observações $O$ de tamanhos diversos, o que não ocorre em outros classificadores, tais como: SMV (*Support Vector Machine*) ou RN (Redes Neurais).

Esta é a razão da designação "escondido" ao modelo de Markov, pois não se sabe quantas observações $O_i$ correspondem a cada estado no modelo Markoviano treinado. A formulação teórica de HMM está além do escopo deste artigo, porém uma excelente introdução ao assunto pode ser encontrada em [Rabiner e Juang 1993].

 ©2012 SBC — Soc. Bras. de Computação

Assim, o HMM integra corretamente os diferentes níveis de modelagem e também fornece algoritmos eficientes para determinar valores ideais para os parâmetros do modelo. A modelagem Markoviana assume que um modelo é representado por uma sequência de observações. Estas observações devem ser estatisticamente independentes uma vez que a sequência oculta (*hidden*) de estado subjacente é conhecida. Primeiramente deve ser estabelecida uma estrutura para o modelo e então se utiliza de estimativa de parâmetros para melhorar a probabilidade de geração de dados de treinamento por parte desses modelos. O HMM absorve a sequência de observação durante a fase de treinamento, logo é capaz de associar cada estado de acordo com alguma função de densidade de probabilidade.

Os modelos de HMM para detecção de aliciamento sexual de menores são baseados em uma abordagem global e uma topologia ergódica [Rabiner 1989]. Ou seja, cada nó do modelo pode ser alcançado a partir de um outro nó qualquer por um número finito de passos.

A formação do modelo baseia-se no algoritmo de *Baum-Welch* [Rabiner e Juang 1993]. Neste trabalho, o processo de classificação consiste em determinar a máxima probabilidade *a posteriori* para as linhas da conversação do agressor ou da vítima, sendo que cada linha da conversa $w$ gera uma sequência de observações $O$ não conhecidas pelo processo de treinamento (ou seja, os exemplos de treinamento são distintos dos exemplos de teste). Sendo válida a seguinte equação:

$$\Pr(\hat{w} \mid O) = \max_{w} \Pr(w \mid O) \tag{1}$$

Aplicando-se o teorena de Bayes, tem-se a equação fundamental de reconhecimento de padrões:

$$\Pr(w \mid O) = \frac{\Pr(O \mid w).\Pr(w)}{\Pr(O)} \tag{2}$$

Então, sabendo-se que $\Pr(O)$ não depende de $w$, a classificação torna-se equivalente a maximizar a probabilidade conjunta dada por:

$$\Pr(w, O) = \Pr(O \mid w).\Pr(w) \tag{3}$$

Finalmente, tem-se que $\Pr(w)$ é a probabilidade *a priori* da conversa $w$ e está relacionada com o problema modelado. A estimativa de $\Pr(O|w)$ requer um modelo probabilístico que seja capaz de considerer as variações das conversas $w$ representadas pelas sequências de observações $O$. Durante os experimentos, o confronto dos *scores* entre cada modelo e as sequências de observações dos conjuntos de teste foi gerado pelo algoritmo de *Viterbi* [Rabiner e Juang 1993].

## 3. Trabalhos Relacionados

A seguir serão apresentados os trabalhos relacionados com o tema de modo a evidenciar as abordagens utilizadas nestes trabalhos. As abordagens podem estar baseadas no uso de técnicas com filtro de palavras, regras e classificadores.

### 3.1. Detecção dos estágios baseada em filtro de palavras

No estudo de Kontostathis [Kontostathis et al. 2009a] é desenvolvido um *software* denominado *ChatCoder1* que faz a busca dos estágios de comunicação adaptados por [Leatherman 2009] em transcrições de conversas de salas de bate-papo. Naquele

 ©2012 SBC — Soc. Bras. de Computação

trabalho, os autores usaram como base de dados 288 transcrições de conversas existentes em [Perverted 2003]. Este *site* está descrito na Seção 4, pois a base de dados utilizada nos experimentos realizados também provém deste *site*. Kontostathis desenvolveu um léxico de palavras e um manual de regras para definir os estágios do modelo de comunicação. O léxico foi desenvolvido utilizando-se de termos de aliciamento, palavras, ícones, frases e linguagem virtual para cada estágio do modelo. O *software ChatCoder1* captura os diálogos e já classifica, por meio do uso do léxico, as frases em seus estágios. Porém, se as palavras ou frases não fossem apresentadas exatamente como no léxico (técnica "caça-palavras"), não eram marcadas como um dos estágios do modelo de comunicação, o que gerava baixa taxa de acerto desta abordagem.

É importante observar que as técnicas de detecção de aliciamento sexual baseadas em análise dos elementos gramaticais de uma frase apresentam-se ineficientes porque os diálogos em salas de bate-papo são muito próximos da linguagem coloquial, que não segue o rigor gramatical da linguagem escrita.

### 3.2. Detecção dos estágios baseada em regras

Num segundo trabalho, Kontostathis [Kontostathis et al. 2009b] faz o uso de uma técnica baseada em regras para desenvolver o *software ChatCoder2*, uma evolução do *ChatCoder1*. As regras para o desenvolvimento do *ChatCoder2* foram criadas com base na transcrição de 15 conversas disponíveis em [Perverted 2003]. O algoritmo do *ChatCoder2* foi desenvolvido utilizando-se das regras e análise da composição gramatical das frases do diálogo para cada um dos estágios do modelo de comunicação adaptados por Leatherman [Leatherman 2009]. Também foram acrescentados valores nominais para os estágios do modelo para que se uma palavra ou frase estivesse em mais de em estágio, seria considerada no estágio com maior valor nominal. Para concluir que o método baseado em regras proposto por Kontostathis apresentava resultados melhores que o método "caça-palavras" (*ChatCoder1*), os autores calcularam a confiabilidade entre os codificadores (*Intercoder Reliability*). Usando o método de [Holsti 1969] foram feitas comparações entre dois codificadores humanos, o *ChatCoder1* e o *ChatCoder2*, e os autores avaliaram ser possível obter uma melhora na confiabilidade máxima de 13,21% e na média de 5,81%.

Porém cabe ressaltar que as técnicas de detecção baseadas em regras são dependentes da linguagem cotidiana analisada e dos vícios de linguagem de seus interlocutores.

### 3.3. Detecção dos estágios baseada em classificadores

McGhee e seus colegas [McGhee et al. 2011] fazem uso de algoritmos de aprendizagem de máquina para classificar as linhas de conversas de salas de bate-papo. Neste estudo, o modelo de comunicação adaptado por Leatherman [Leatherman 2009] foi consolidado em apenas quatro estágios: (1) <u>troca de informações pessoais</u> – inclui informações pessoais, gostos/preferências e sentimentos, e aspecto de construção ou manutenção do relacionamento entre o agressor e a vítima; (2) <u>preparação</u> – coincide com o proposto por Olson; (3) <u>aproximação</u> – envolve a obtenção de telefone, endereço, marcação de encontro, compartilhamento de segredos e o isolamento da vítima e (4) <u>outros</u> – enunciados que não estão contemplados nos itens anteriores são classificados como do quarto estágio. Para este estudo foram utilizadas 33 transcrições de conversas

 ©2012 SBC — Soc. Bras. de Computação

disponíveis em [Perverted 2003] como base. Três conversas eram comuns a ambos os codificadores e foram utilizadas para verificar a concordância de classificação dos codificadores. Como o número de estágios do modelo de comunicação foi reduzido neste trabalho, as regras para marcar cada linha do diálogo foram também adaptadas. O léxico utilizado em Kontostathis [Kontostathis et al. 2009b] foi também usado neste trabalho para desenvolver os atributos de entrada do algoritmo de aprendizagem de máquina. Neste trabalho os autores utilizaram a precisão (*Accuracy*) como métrica de avaliação para a comparação entre os vários experimentos realizados com diversos algoritmos de aprendizagem de máquina. Quando as 33 conversas foram submetidas aos algoritmos individualmente, pode-se dizer que a aprendizagem de máquina apresentou um melhor desempenho se comparado às regras. Quando as conversas formaram um único bloco de entrada o desempenho dos classificadores foi muito semelhante ao método baseado em regras, porém os atributos apresentados na forma numérica tiveram um desempenho um pouco melhor.

As técnicas baseadas em classificadores sofrem com problemas de seleção das características, palavras ou frases, de modo a formar um léxico válido e representativo do problema em questão. Além disto, uma mesma palavra ou frase pode ser encontrada em diferentes estágios do processo de aliciamento, dificultando assim a classificação ou identificação do problema.

## 4. Proposta de Método para Modelagem de Aliciamento

A proposta está baseada na identificação dos estágios de comunicação de uma conversa entre um possível agressor e sua vítima, bem como na definição dos modelos que representam o perfil do agressor e da vítima. A Figura 2 ilustra uma visão geral da proposta.

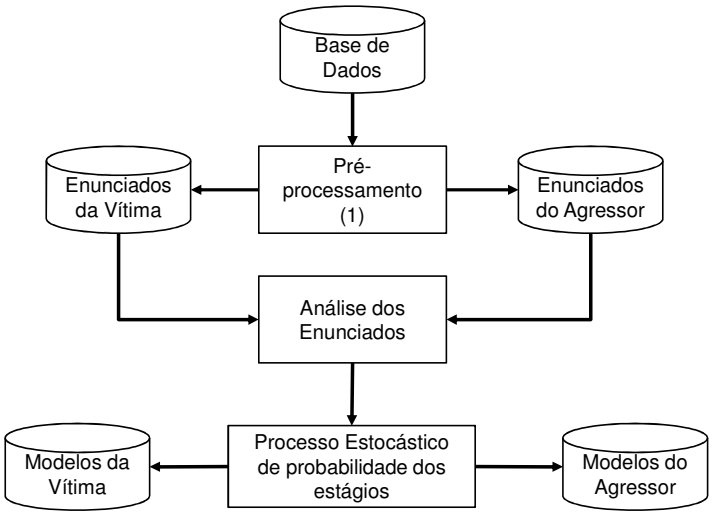



**Figura 2 – Visão geral da proposta**

Os esforços inicialmente se concentraram na formação da base de dados de conversações que está baseada no *site Perverted-justice.com* [Perverted 2003]. O *site* disponibiliza em seu conteúdo público transcrições de conversas de salas de bate-papo que levaram os agressores à prisão. Voluntários do *Perverted Justice* se passam por menores em salas de bate-papo com intuito de encontrar possíveis agressores que tinham como objetivo iniciar um abuso sexual de menores. Quando esse agressor é

 ©2012 SBC — Soc. Bras. de Computação

preso e tem sua sentença definida por um juiz de direito, a transcrição da conversa é tornada publica no *site*. Um exemplo de conversação é mostrado na Figura 3, onde se pode observar os enunciados do agressor (cobbler1976) e da vítima (that_polish_chick94).

**cobbler1976** (02/10/08 9:49:05 PM): what do ur panties look like?
that_polish_chick94 (02/10/08 9:49:26 PM): black bikini type
**cobbler1976** (02/10/08 9:49:31 PM): sexy
that_polish_chick94 (02/10/08 9:49:35 PM): ty
**cobbler1976** (02/10/08 9:50:04 PM): are u horny?
that_polish_chick94 (02/10/08 9:50:45 PM): i dunno not really
**cobbler1976** (02/10/08 9:51:18 PM): i could get u horny if i was there
that_polish_chick94 (02/10/08 9:51:25 PM): yeah
**cobbler1976** (02/10/08 9:51:34 PM): yep
**cobbler1976** (02/10/08 9:52:21 PM): well then?
that_polish_chick94 (02/10/08 9:52:34 PM): what
**cobbler1976** (02/10/08 9:52:57 PM): will u show me my cum in ur mouth?
that_polish_chick94 (02/10/08 9:53:12 PM): yeah if u wanted
**cobbler1976** (02/10/08 9:53:31 PM): i could fuck u?
that_polish_chick94 (02/10/08 9:53:47 PM): yeah
**cobbler1976** (02/10/08 9:54:12 PM): condom right?
that_polish_chick94 (02/10/08 9:54:28 PM): i would get preg if u didnt wouldnt i?
**cobbler1976** (02/10/08 9:54:37 PM): yep
that_polish_chick94 (02/10/08 9:54:44 PM): then yeah
**cobbler1976** (02/10/08 9:55:01 PM): when?
that_polish_chick94 (02/10/08 9:55:52 PM): when my mom goes outta town
that_polish_chick94 (02/10/08 9:56:00 PM): she goes somtimes
that_polish_chick94 (02/10/08 9:56:09 PM): for like at least a day
that_polish_chick94 (02/10/08 9:56:24 PM): i dunno when yet tho
**cobbler1976** (02/10/08 9:56:30 PM): ok
**cobbler1976** (02/10/08 9:57:03 PM): just sex or party and sex?
that_polish_chick94 (02/10/08 9:57:40 PM): i dunno both? lol
**cobbler1976** (02/10/08 9:57:48 PM): sure....lol
**cobbler1976** (02/10/08 9:57:54 PM): drinks?

**Figura 3 - Visão geral das transcrições das conversas [Perverted 2003]**

A segunda fase do trabalho consistiu no pré-processamento do texto das transcrições das conversações, separando as linhas da conversa em enunciados do agressor e enunciados da vítima. O objetivo da separação dos enunciados é trabalhar com cada um independentemente no intuito de obter os modelos de agressor e vítima, sem que um venha a interferir na obtenção do outro.

Após esta separação de conteúdos as transcrições foram analisadas manualmente para que fosse possível identificar nos enunciados os estágios descritos por [Leatherman 2009].

Os modelos do agressor e da vítima foram definidos usando técnicas estocásticas com a sequência de estágios que as conversas analisadas apresentavam. Estas técnicas constituem os Modelos Escondidos de Markov, que foram utilizados neste trabalho para a modelagem de aliciamento sexual de menores. Com o HMM (*Hidden Markov Models*) se pode obter a probabilidade da ocorrência (susceptibilidade) do aliciamento sem ter todos os estágios da sequência de observação, pois não se sabe *a priori* o tamanho que uma conversação pode atingir, nem quantas destas linhas de enunciados permitirão a identificação de um estágio do aliciamento.

Outra característica importante dos HMMs para o problema do aliciamento é que as observações são geradas independentemente para o agressor e vítima, não havendo interdependência entre os elementos a serem modelados. Por exemplo, em uma mesma conversação, o agressor pode utilizar-se de 1.000 linhas de enunciados e a vítima somente de 900 linhas. Nos experimentos realizados isto não interfere nem no processo de treinamento nem no de classificação, simplesmente tem-se que a cada linha de enunciado é associado um dos nove possíveis estágios definidos por [Leatherman

 ©2012 SBC — Soc. Bras. de Computação

2009] ou nenhum estágios é observado. Assim, tem-se que o modelo utiliza da estrutura conceitual de [Leatherman 2009], mas não está vinculada a essa. Ou seja, os estágios do aliciamento definidos por [Leatherman 2009] não obrigam que os modelos treinados tenham nove estados, mas sim um número ideal de estados que permita a cada um desses absorver parte da sequência de observações *O* (Seção 2.2.).

## 5. Testes Experimentais e Análise dos Resultados

Para a realização dos experimentos, 21.989 linhas de enunciados de transcrições de conversas foram analisadas por quatro avaliadores, separando-os em linhas de enunciados do agressor e da vítima. A análise de cada enunciado se fez necessária para identificar os estágios de aliciamento definidos por [Leatherman 2009] para o contexto virtual de aliciamento sexual de menores. Estes 21.989 enunciados foram observados em 20 conversas disponíveis em [Perverted 2003]. Nas conversar analisadas as idades dos agressores variaram entre 19 e 49 anos e das vítimas entre 13 e 15 anos.

**Tabela 1 - Representação numérica para os estágios de aliciamento**

| ESTÁGIO | REPRESENTAÇÃO |
|---|---|
| *Acesso* | *1* |
| *Informações Pessoais* | *2* |
| *Informações de Relacionamento* | *3* |
| *Atividades* | *4* |
| *Elogios* | *5* |
| *Dessensibilização Comunicativa* | *6* |
| *Reenquadramento* | *7* |
| *Isolamento* | *8* |
| *Aproximação* | *9* |

Para cada um dos estágios de aliciamento observado foi atribuído um valor numérico conforme ilustra a Tabela 1, para que fosse possível o treinamento usando HMM. Nos enunciados em que não foi possível identificar nenhum estágio foi marcada a sigla NEO – Nenhum Estágio Observado.

**Tabela 2 – Estágios observados nos enunciados dos agressores**

| ENUNCIADOS DOS AGRESSORES | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Conversa Analisada* | *Estágio Observado* | | | | | | | | | | | *Total de Enunciados* |
| | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *Mais de um estágio* | *NEO* | |
| **Conversa 01** | 10 | 26 | 2 | 3 | 7 | 98 | 8 | 9 | 16 | 3 | 350 | **532** |
| **Conversa 02** | 41 | 60 | 6 | 48 | 27 | 92 | 14 | 7 | 24 | 5 | 255 | **579** |
| **Conversa 03** | 17 | 12 | 6 | 97 | 41 | 39 | 3 | 5 | 17 | 19 | 330 | **586** |
| **Conversa 04** | 7 | 21 | 2 | | | 6 | 61 | 21 | 27 | 3 | 323 | **471** |
| **Conversa 05** | 9 | 14 | 19 | 47 | 31 | 36 | | 7 | 11 | 16 | 194 | **384** |
| **Conversa 06** | 10 | 35 | 16 | 75 | 23 | 106 | 13 | 1 | 15 | 14 | 316 | **624** |
| **Conversa 07** | 4 | 19 | 7 | 52 | 19 | 38 | | 2 | 33 | 0 | 258 | **432** |
| **Conversa 08** | 11 | 23 | 6 | 4 | 27 | 25 | 5 | 3 | 26 | 4 | 551 | **685** |
| **Conversa 09** | 12 | 23 | 6 | 11 | 12 | 93 | 12 | 17 | 34 | 4 | 605 | **829** |
| **Conversa 10** | 16 | 28 | 14 | 57 | 10 | 15 | 4 | 6 | 79 | 8 | 355 | **592** |
| **Conversa 11** | 8 | 25 | 4 | 7 | 33 | 23 | | 4 | 10 | 5 | 630 | **749** |
| **Conversa 12** | 9 | 9 | | 9 | 3 | 123 | 5 | 13 | 29 | 13 | 220 | **433** |
| **Conversa 13** | 3 | 16 | 6 | 8 | 26 | 40 | 17 | 26 | 26 | 13 | 384 | **565** |
| **Conversa 14** | 5 | 7 | | | 6 | 17 | | 8 | 16 | 0 | 613 | **672** |
| **Conversa 15** | 3 | 15 | 8 | 52 | 8 | 19 | | 2 | 21 | 6 | 289 | **423** |
| **Conversa 16** | 22 | 24 | 9 | 49 | 5 | 50 | | 12 | 16 | 0 | 391 | **578** |
| **Conversa 17** | 7 | 10 | 1 | | 3 | 70 | 2 | 8 | 5 | 1 | 333 | **440** |
| **Conversa 18** | 19 | 14 | 15 | 45 | 2 | 26 | | 2 | 14 | 4 | 294 | **435** |
| **Conversa 19** | 11 | 26 | 6 | | | 134 | | 11 | 10 | 1 | 383 | **582** |
| **Conversa 20** | 4 | 27 | 8 | 82 | 10 | 25 | 2 | 1 | 20 | 2 | 409 | **590** |
| **Total de Estágios** | **228** | **434** | **141** | **646** | **293** | **1.075** | **146** | **165** | **449** | **121** | **7.483** | **11.181** |

A Tabela 2 mostra que todos os nove estágios foram observados nas conversas analisadas, assim como o respectivo número de ocorrência para cada um. No total foram analisadas 11.181 linhas com enunciados de agressores. A Tabela 3 é similar a Tabela 2

 ©2012 SBC — Soc. Bras. de Computação

e mostra o número de ocorrências para cada estágio considerando os enunciados da vítima. Neste caso, foram analisadas um total de 10.808 linhas com os enunciados da vítima.

**Tabela 3 - Estágios observados nos enunciados das vítimas**

| ENUNCIADOS DAS VÍTIMAS | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Conversa Analisada* | *Estágio Observado* | | | | | | | | | | | *Total de Enunciados* |
| | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *Mais de um estágio* | *NEO* | |
| **Conversa 01** | 2 | 19 | 2 | 1 | 13 | 18 | 1 | 10 | 11 | 4 | 201 | **282** |
| **Conversa 02** | 23 | 42 | 15 | 46 | 22 | 9 | 2 | 6 | 8 | 2 | 325 | **500** |
| **Conversa 03** | 15 | 10 | 20 | 106 | 42 | 16 | | 10 | 13 | 20 | 481 | **733** |
| **Conversa 04** | 8 | 20 | 8 | | 5 | 1 | 7 | 9 | 15 | 0 | 610 | **683** |
| **Conversa 05** | 15 | 6 | 14 | 45 | 11 | 13 | | 3 | 19 | 3 | 347 | **476** |
| **Conversa 06** | 11 | 12 | 11 | 46 | 11 | 6 | 1 | 1 | 8 | 11 | 490 | **608** |
| **Conversa 07** | 3 | 13 | 19 | 43 | 29 | 9 | | 4 | 26 | 2 | 285 | **433** |
| **Conversa 08** | 8 | 12 | 4 | 5 | 18 | 5 | | 4 | 15 | 1 | 543 | **615** |
| **Conversa 09** | 19 | 19 | 2 | 15 | 4 | 21 | | 17 | 15 | 3 | 486 | **601** |
| **Conversa 10** | 9 | 15 | 26 | 37 | 7 | 5 | | 4 | 18 | 1 | 357 | **479** |
| **Conversa 11** | 10 | 13 | 2 | 4 | 22 | 3 | | 1 | 3 | 0 | 541 | **599** |
| **Conversa 12** | 17 | 7 | 4 | 12 | 3 | 11 | | 18 | 16 | 0 | 508 | **596** |
| **Conversa 13** | 7 | 4 | 1 | 5 | 4 | 3 | | 7 | 8 | 0 | 422 | **461** |
| **Conversa 14** | 3 | 9 | | 1 | 3 | 4 | | 10 | 13 | 2 | 385 | **430** |
| **Conversa 15** | 15 | 18 | 12 | 92 | 11 | 14 | 1 | | 16 | 0 | 458 | **637** |
| **Conversa 16** | 9 | 20 | 33 | 68 | 1 | 6 | | 3 | 13 | 0 | 360 | **513** |
| **Conversa 17** | 9 | 13 | 1 | 2 | 7 | 28 | | 12 | 3 | 1 | 505 | **581** |
| **Conversa 18** | 30 | 30 | 18 | 77 | 3 | 3 | | 1 | 16 | 0 | 383 | **561** |
| **Conversa 19** | 11 | 21 | 10 | | 13 | 23 | 1 | 14 | 7 | 1 | 613 | **714** |
| **Conversa 20** | 4 | 13 | 13 | 42 | 11 | 4 | | | 12 | 7 | 200 | **306** |
| **Total de Estágios** | **228** | **316** | **215** | **647** | **240** | **202** | **13** | **134** | **255** | **58** | **8.500** | **10.808** |

Para a modelagem do aliciamento dos agressores e das vítimas foram utilizados os algoritmos dos Modelos Escondidos de Markov disponíveis no *toolkit* HTK [HTK 2012] considerando HMM discreto e topologia ergódiga [Grundy 1997] [Szoke 2004]. Para o treinamento dos modelos foram utilizadas as ferramentas do *toolkit HInit* e *HRest*, que implementam o algoritmo de *Baum-Welch*. Para o teste dos modelos resultantes foi utilizada a ferramenta *HVite* do *toolkit* HTK, que implementa o algoritmo de *Viterbi*.

Os 21.989 enunciados foram separados em quatro conjuntos (Tabela 4) e para execução de dois experimentos diferentes. No primeiro o objetivo foi estudar se a composição das sequências de observação considerando diferentes critérios influenciaria no resultado da classificação resultante, considerando cada um dos modelos (agressor e vítima) – teste (*i*).

**Tabela 4 - Composição dos dados para os conjuntos de treinamento e teste**

| CONJUNTO | DADOS |
|---|---|
| *Conjunto 1* | *Sequência dos estágios observados, excluindo-se os enunciados em que mais de um estágio foi observado e também os NEO.* |
| *Conjunto 2* | *Sequência dos estágios observados, excluindo-se os NEO.* |
| *Conjunto 3* | *Sequência dos estágios observados, excluindo-se os enunciados em que mais de um estágio foi observado.* |
| *Conjunto 4* | *Sequência com todos os eságios observados.* |

Para obtenção dos modelos (oito no total, dois para cada conjunto), primeiramente foi executado o treinamento com *HInit* e *HRest* para as sequências de observação do agressor e da vítima, separadamente. O treinamento foi executado utilizando-se de 80% das transcrições das conversas analisadas (e.g. 8.945 estágios nas sequências de observação do agressor e 8.647 estágios nas sequências de observação da vítima). A Tabela 5 mostra o número de estados HMM para os modelos treinados

 ©2012 SBC — Soc. Bras. de Computação

considerando cada conjunto de dados (Tabela 4). Observa-se que o número de estados resultante do treinamento é variável, sendo maior à medida que as sequências de observações são maiores (Conjunto 2 e Conjunto 4). Isto corresponde ao esperado e mostra que o procedimento de treinamento ajusta os modelos às sequências de observações.

**Tabela 5 - Número de estados resultantes do treinamento com HMM**

| | CONJUNTO 1 | | CONJUNTO 2 | | CONJUNTO 3 | | CONJUNTO 4 | |
|---|---|---|---|---|---|---|---|---|
| | Agressor | Vítima | Agressor | Vítima | Agressor | Vítima | Agressor | Vítima |
| Nº de Estados | 8 | 8 | 11 | 12 | 6 | 8 | 10 | 12 |

Para testar os modelos resultantes foi usado o *HVite*, utilizando os 20% das transcrição de conversações não utilizadas no treinamento (e.g. 2.236 estágios nas sequências de observação do agressor e 2.161 estágios nas sequências de observação da vítima). Os testes foram executados de maneira tradicional, *all-against-all*. Estes testes permitiram não somente verificar a eficácia do treinamento, mas também o grau de confusão entre os comportamentos do agressor e da vítima de modo a constatar se observações distintas guardavam correlações intrínsecas entre os modelos obtidos – teste (*ii*).

A Tabela 6 apresenta a matriz de confusão com o resumo dos resultados dos testes (*i*) e (*ii*). Pode-se observar pelo percentual de acerto que a separação dos modelos (agressor e vítima, leitura da Tabela 6 pela linha) não trouxe benefícios importantes, porque os resultados são muitos similares em cada um dos conjuntos. Já a separação em conjuntos apresentou uma leve variação, sendo menor a taxa de acerto para os conjuntos com maiores sequências de observações. Logo, pode-se inferir que o Conjunto 1, mais simples – constituído de menores sequências de observação (Tabela 2 e Tabela 3) e menos estados HMM (Tabela 5) é equivalente ao Conjunto 2, mais complexo (maiores sequências de observação e número de estados). Analogamente, o mesmo ocorre entre os Conjuntos 3 e 4. Assim, um resultado importante deste trabalho é a constatação que o Conjunto 1 pode representar com vantagem os demais por apresentar as sequências de observação com os estágios puros (únicos no mesmo enunciado) e com melhores taxas de acerto.

**Tabela 6 - Matriz de confusão**

| | CONJUNTO 1 | | CONJUNTO 2 | | CONJUNTO 3 | | CONJUNTO 4 | |
|---|---|---|---|---|---|---|---|---|
| | Agressor | Vítima | Agressor | Vítima | Agressor | Vítima | Agressor | Vítima |
| Agressor | 89,18% | 90,48% | 88,22% | 89,65% | 77,74% | 79,99% | 77,23% | 79,28% |
| Vítima | 88,61% | 91,15% | 86,39% | 90,07% | 76,08% | 80,66% | 75,20% | 79,91% |

No segundo experimento, os conjuntos de sequências de observações do agressor e da vítima foram considerados combinados (agrupados). Os resultados obtidos podem ser observados na Tabela 7. O objetivo deste experimento foi avaliar se a separação dos modelos (agressor e vítima) trazia benefícios efetivos à modelagem ou não de aliciamento. A Tabela 7 mostra que ao se aplicar os dados de entrada do agressor e da vítima como um único conjunto se obtém uma redução nas taxas de acerto, porém as taxas de acerto nos conjuntos são analogamente proporcionais aos modelos em separado (Tabela 6). Assim, pode-se concluir que a modelagem considerando agressor e vítima em separado é mais efetiva que o modelo combinado. Observa-se também que o

 ©2012 SBC — Soc. Bras. de Computação

modelo combinado é mais complexo que o modelo separado (composto de aproximadamente a metade dos enunciados), deve-se então preferir este último na escolha entre os dois, pois o modelo em separado é mais simples e resulta em melhor eficiência (taxa de acerto) que o combinado.

**Tabela 7 - Taxas de acerto para o modelo combinado**

| CONJUNTO 1 | CONJUNTO 2 | CONJUNTO 3 | CONJUNTO 4 |
|---|---|---|---|
| 82,02% | 81,27% | 62,66% | 62,10% |

Avaliando os resultados dos experimentos pode-se concluir que o conjunto puro (Conjunto 1, Tabela 5) é o mais simples, como mostrado no primeiro experimento. O mesmo ocorre com o modelo em separado, como mostrado no segundo experimento. Assim, os experimentos mostram que se pode trabalhar apenas com um dos modelos, o do agressor por exemplo – pois o objetivo deste trabalho é detecção de aliciamento, e com o conjunto puro, pois os resultados serão equivalentes a trabalhar com o modelo com tudo combinado.

## 6. Conclusão

Este artigo propôs um método para análise de mensagens instantâneas de texto, realizadas em salas de bate-papo na *Internet*. O objetivo da proposta foi modelar o perfil do agressor e da vítima, usando HMM, baseando-se na identificação dos estágios de aliciamento observados nas transcrições de conversas entre ambos.

Os experimentos realizados mostraram que o uso dos modelos do agressor e da vítima em separado fornecem resultados mais promissores que quando combinados. Além disso, foi mostrado que há equivalência entre os pares de conjuntos das sequências de observação. Assim, concluiu-se que se pode trabalhar com o modelo do agressor e com o conjunto de sequências de observações simples (puras), resultando em modelagem mais simples que as relatadas nos trabalhos da literatura.

É importante observar que os resultados obtidos com as sequências de teste permitem inferir que a susceptibilidade ao abuso é obtida de maneira análoga, dado que as sequências de diferentes tamanhos foram bem identificadas pelo modelo do agressor e conjunto de sequências de observações simples. Assim, esse trabalho contribui inovando com o uso do HMM na detecção de aliciamento. Esta abordagem permite inferir a susceptibilidade ao abuso sexual de menores, diferentemente do que é proposto na literatura que apenas se limita a detectar os estágios de aliciamento.

A susceptibilidade ao abuso, calculada em tempo real (durante o desenvolvimento da conversação na troca de mensagens instantâneas), pretendida como trabalhos futuros, só é possível devido ao emprego do HMM. Pois, com o HMM é possível estimar a susceptibilidade (usando probabilidades) ao abuso com diferentes tamanhos de sequências de observações.

## 7. Agradecimento

Os autores agradecem aos bolsistas de PIBIC Caio Carnelos, Paola Sayuri e Tiago Betiato pela participação no tratamento das conversações das salas de bate-papo.

 ©2012 SBC — Soc. Bras. de Computação

## 8. Referências

Grundy, W. N.. "Modeling Biological Sequences Using HTK". Technical Report, 1997.

Holsti, O. R.. "Content Analysis for the Social Sciences and Humanities". Addison-Wesley, 1969.

Ho, W. H.; Watters, P. A.. "Statistical and Structural Approaches to Filtering Internet Pornography". IEEE International Conference on Systems, Man and Cybernetics, 2004. p. 4792-4798.

HTK toolkit. Disponível em <http://htk.eng.cam.ac.uk/>. Acesso em 04 de junho de 2012.

Kontostathis, A.; Edwards, L.; Leatherman, A.. "ChatCoder: Toward the Tracking and Categorization of Internet Predators". Proceedings of 2009 Text Mining Workshop – Society for Industrial and Applied Mathematics International Conference on Data Mining, May 2009.

Kontostathis, A.; Edwards, L.; Bayzick, J.; McGhee, I.; Leatherman, A.; Moore, K.. "Comparison of Rule-based to Human Analysis of Chat Logs". International Workshop on Mining Social Media, November 2009.

Leatherman, A.. "Luring language and virtual victims: Coding cyber-predators online communicative behavior". Technical report of Ursinus College, 2009.

McGhee, I.; Bayzick, J.; Kontostathis, A.; Edwards, L.; McBride, A.; Jakubowski, E.. "Learning to Identify Internet Sexual Predation". International Journal of Electronic Commerce, 2011. Vol. 15, p. 103-122.

Michalopoulos, D.; Mavridis, I.. "Towards Risk Based Prevention of Grooming Attacks". International Conference on Security and Cryptography, July 26-28, 2010. p. 1-4.

National Center for Missing and Exploited Children. Disponível em <http://www.missingkids.com/en_US/documents/CyberTiplineFactSheet.pdf>. Acesso em 17 de fevereiro de 2012.

Olson, L. N.; Daggs, J. L.; Ellevold, B. L.; Rogers, T. K. K.. "Entrapping the Innocent: Toward a Theory of Child Sexual Predators' Luring Communication". Communication Theory, 2007. Vol. 17, p. 231-251.

Perverted Justice Foundation Incorporated. Disponível em < www.perverted-justice.com >. Acesso em janeiro de 2012.

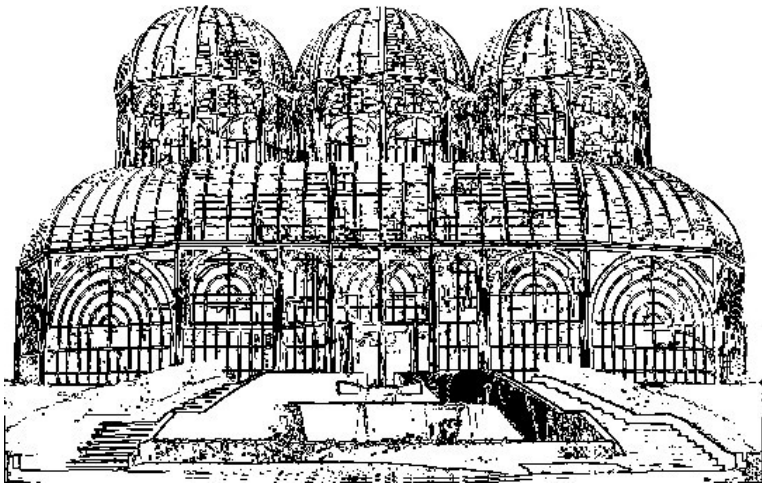
Rabiner, L.R.; Juang, B-H.. "Fundamentals of speech recognition". Prentice Hall Inc., Englewood Clifss, New Jersey, 1993.

Rabiner, L. R.. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, February, 1989. Vol. 77, No. 2, p.257-286.

Rashid, A.. "The Talk of Crime". Investigative Practice Journal, July 10, 2008. p. 28-29.

Szoke, I.. "Speech Units Automatically Generated by Ergodic Hidden Markov Model". Proceedings of 10th Conference and Competition Student EEICT, 2004. Vol. 1.

 ©2012 SBC — Soc. Bras. de Computação



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# Trilha principal – Artigos Curtos

# Arquitetura de um sistema integrado de defesa cibernética para detecção de *botnets*

**Sérgio dos Santos Cardoso Silva**[1] **e Ronaldo Moreira Salles**[1] **(IME)**
**Praça General Tiburcio, 80 – 22290-270 – Rio de Janeiro – RJ – Brasil**

{scardoso,salles}@ime.eb.br

[1]Programa de Engenharia de Defesa – Instituto Militar de Engenharia

***Abstract.*** *This work proposes a novel architecture for for the design of a cyber defense system able to detect bots and block communication between bots and botnets. It is also proposed an algorithm based on graphs for detecting bots. The architecture was tested using DNS logs of an academic institute. Experiments were carried out by analyzing records of DNS queries in order to find machines suspected of being zombies. Preliminary results show that mechanisms employed can filter DNS records and identify machines suspected of belonging to a botnet.*

***Resumo.*** *Este trabalho tem por objetivo apresentar uma proposta de arquitetura para um sistema de defesa cibernética capaz de detecar* bots *e bloquear a comunicação entre os bots e as* botnets. *Também é proposto um algoritmo de detecção de* bots *baseado em grafos de relacionamento. Foram realizados experimentos por meio da análise de registros de consultas DNS com o objetivo de encontrar máquinas suspeitas de serem zumbis. Resultados preliminares mostram que os mecanismos empregados permitem filtrar os registros de DNS e identificar máquinas suspeitas de pertencerem a uma botnet.*

## 1. Introdução

Diversos tipos de ataques cibernéticos contra estruturas críticas e usuários comuns têm sido relatados nos últimos anos, trazendo à tona a preocupação do emprego desses ataques em ações de Guerra Cibernética. Em algumas dessas ações foram empregadas estruturas conhecidas como *botnets*. As *botnets* são redes formadas por *hosts* escravos, denominados *bots*, que são controladas por um ou mais atacantes, denominados *botmasters*, que tem por objetivo realizar algum tipo de ação maliciosa [Rajab et al. 2006, Zeidanloo et al. 2010].

Atualmente, as *botnets* são uma das principais ameaças ao funcionamento dos serviços de rede [Ceron et al. 2010], pois permitem ao atacante que controla essas máquinas realizar diversas ações maliciosas, como roubo de informações, envio de mensagens não solicitadas (*spam*), ataques de negação de serviços, manipulação de jogos ou pesquisas online, entre outras.

Sendo assim, é necessário que sejam realizados esforços para combater essas redes, identificando as máquinas componentes e desativando a comunicação entre elas. Entretanto, realizar a detecção da *botnet* é uma atividade difícil. As *botnets* utilizam técnicas que permitam ficar a maior parte do tempo ativas, ocultando o tráfego gerado pelas botnets, tornando-as invisíveis [Dagon et al. 2007]. Um exemplo desses mecanismos é a redução do número de trocas de mensagens entre os componentes da rede.

 ©2012 SBC — Soc. Bras. de Computação

Os trabalhos existentes sobre *botnets* normalmente descrevem soluções isoladas apresentando mecanismos de detecção [Gu et al. 2008, Coskun et al. 2010, Zhang et al. 2011] ou técnicas para bloqueio e desativação das *botnets* [Stone-Gross et al. 2011, Sanchez et al. 2011]. Além disso, as soluções comerciais existentes [Snort 2006] empregam técnicas baseadas em assinaturas, que tem limitação em detectar novos *malwares* ou *bots*, devido as caracteríticas furtivas que estes possuem.

Considerando a pesquisa bibliográfica realizada, não existe uma proposta de solução integrada que permita tanto a detecção quanto a desativação das *botnets*. Portanto, este trabalho tem por objetivo apresentar uma proposta de arquitetura de sistema de defesa cibernética capaz de realizar a detecção de máquinas *bots* e bloquear a comunicação entre os bots e as *botnets*, diminuindo a sua efetividade e, possivelmente, desarticulando-as. Também é proposta uma técnica de detecção de *bots* que utilizam o protocolo DNS. Resultados preliminares mostram que os mecanismos empregados permitem filtrar os registros de DNS e identificar máquinas suspeitas.

As principais contribuições desse artigo são:

- Apresentar uma proposta de arquitetura para um sistema integrado de defesa cibernética.
- Propor uma técnica de detecção de *bots* baseada em análise de logs DNS, empregando a arquitetura proposta.

Este trabalho está organizado conforme descrito a seguir. Na Seção 2 é apresentado o ciclo de vida de um *bot*. Na Seção 3 é apresentada a arquitetura de um sistema integrado de defesa cibernética e o estudo de caso para o DNS na seção 4. A Seção 5 apresenta os resultados iniciais com a análise de registros de acesso ao DNS. Por fim, a Seção 6 mostra as conclusões e possibilidades de trabalhos futuros.

## 2. Ciclo de vida dos *bots*

Para um melhor entendimento do problema e das técnicas propostas, será apresentado o ciclo de vida de um *bot*, que são as etapas que um *host* deve realizar para que se se torne um *bot* e pertença a uma *botnet* [Zhu et al. 2008]. Estas etapas podem receber nomes diferentes, mas os mais adotados estão ilustrados na figura 1.

A infecção inicial é o primeiro passo para transformar uma máquina vulnerável em um *bot*. O invasor realiza uma varredura em uma sub-rede buscando vulnerabilidades conhecidas e infecta a vítima por diversos métodos, por exemplo, através de *exploits*, anexos de mensagens, entre outros [Rajab et al. 2006, Dagon et al. 2007].

Após o invasor infectar a máquina, esta passa para a fase de Injeção Secundária. Nesta etapa, a máquina infectada executa um *script* que busca o código do *malware* em um repositório, transformando efetivamente essa máquina em um *bot* [Rajab et al. 2006].

Após a máquina se tornar um *bot*, ela contacta o servidor de comando e controle (C&C), processo denominado *rally* ou conexão. Esta etapa irá ocorrer toda vez que a máquina infectada for reiniciada, pois, é necessário que ela contate o C&C para informar a sua participação na *botnet*, assim como, para receber os comandos para realização de alguma atividade maliciosa [Dagon et al. 2007].

Devido a essa necessidade de contato com os servidores de C&C, essa fase pode ser considerada como uma etapa vulnerável da *botnet*, principalmente porque existirá um

 ©2012 SBC — Soc. Bras. de Computação

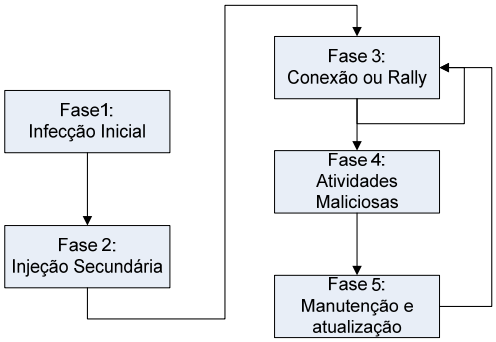



**Figura 1. Ciclo de Vida dos Bots**

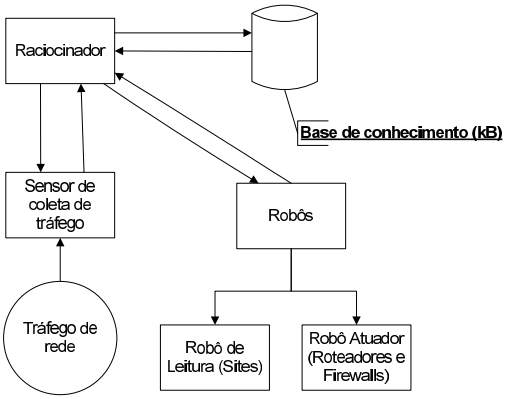



**Figura 2. Arquitetura de sistema integrado de defesa cibernética**

comportamento padrão dos *bots* para realizar a conexão, permitindo que sejam criados mecanismos que possam identificar esse padrão de tráfego e, consequentemente, identificar os componentes da *botnet*.

Após estabelecido o canal de comando e controle, o *bot* aguarda por comandos para realizar alguma atividade maliciosa [Rajab et al. 2006]. A partir dessa fase o *bot* está pronto para realizar algum tipo de ataque.

A última fase do ciclo de vida dos *bots* é a manutenção e atualização dos *malwares* utilizados. A manutenção está associada ao fato de que o *botmaster* deseja manter o seu exército de *bots*. Para isso, é necessário atualizar os códigos por diversos motivos, tais como, evadir das técnicas de detecção, adicionar novas funcionalidades ou migrar para outro servidor de C&C [Zhu et al. 2008].

## 3. Arquitetura do Sistema

Devido à gravidade do problema é necessário o desenvolvimento de soluções que possam integrar as duas fases importantes para combater as *botnets*: detecção e mitigação. A arquitetura de sistema integrado de defesa cibernética proposta na figura 2 integra essas duas fases necessárias ao tratamento do problema detectando a atividade maliciosa e realizando alguma medida para mitigar o problema.

Esta arquitetura é desenvolvida de forma modular permitindo, além da análise do tráfego da rede, a análise semântica dos dados das máquinas atacantes e atacadas, facilitando a identificação das máquinas *bots* e a adoção de medidas de proteção necessárias. A modularidade também permite uma visão abrangente dos vários componentes das redes de computadores monitoradas, fornecendo diferentes funcionalidades, dependendo da perspectiva considerada. A arquitetura é formada pelos seguintes componentes:

- **Sensor de coleta de tráfego**: monitora de forma passiva o tráfego de rede, fornecendo informações importantes na detecção e prevenção de ataques. Este componente está relacionado com as funções de sensoriamento do sistema;
- **Raciocinador**: executa funções voltadas para a análise de informações, processando e analisando os dados coletados, considerando aspectos estatísticos e semânticos dessas informações. A inteligência da arquitetura considera a análise

 ©2012 SBC — Soc. Bras. de Computação

do tráfego aliada à análise semântica como uma forma de aprimorar a detecção e proteção contra ataques;

- **Base de conhecimento (kb)**: componente responsável por armazenar aspectos estatísticos e semânticos do tráfego de rede que são utilizados pelo raciocinador na análise das informações;
- **Robôs de atuação e leitura**: são responsáveis pelas funções de proteção. Podem alterar o funcionamento de dispositivos da rede de modo a melhorar a sua proteção, evitando potenciais ataques, ou coletar informações adicionais para confirmar algum tipo de atividade maliciosa.

O tráfego de rede analisado pelas funções de sensoriamento pode ser obtido de forma bruta, através da coleta de registros de tráfego (*traces*), com algum nível de agregação (fluxos) ou análise de registros (*log*). Para cada tipo de registro a ser analisado é implementado um módulo específico de sensoriamento e coleta de dados.

Na função de análise de informações é empregado um raciocinador modular que emprega algoritmos específicos para análise dos dados obtidos pelo sensor de coleta de tráfego. A partir do resultado dos algoritmos empregados, o raciocinador irá acionar robôs de leitura ou atuação para realizar alguma atividade complementar. O disparo dos robôs é baseado em valores pré-definidos armazenados na base de conhecimento do sistema.

Os robôs são agentes de softwares que realizam atividades complementares ao raciocinador. O robô de leitura pode ser acionado com o objetivo de obter informações adicionais necessárias a tomada de decisão do raciocionador. Por exemplo, se um determinado domínio foi considerado suspeito pelo algoritmo de análise de DNS, o raciocinador pode ativar o robô de leitura para obter mais informações, como, *PageRank* ou dados semânticos do *site*, para poder confirmar que aquele domínio é um C&C. Essas informações adicionais são adicionadas à base de conhecimento.

Confirmado que determinado domínio e/ou máquina é um *bot* ou C&C, o raciocinador pode, então, acionar o robô de atuação que irá adotar alguma medida para mitigar o problema. Os robôs de atuação podem ser especializados, por exemplo, como robôs para *firewalls*, servidores e roteadores. Um exemplo de medida preventiva que pode ser tomada é a adição de uma regra em um *firewall* com o objetivo de bloquear, exclusivamente, a comunicação entre o *bot* e o centro de comando e controle.

## 4. Estudo de caso - DNS

O primeiro módulo implementado na arquitetura proposta é para análise de registros de acesso (*log*) DNS. As consultas DNS são necessárias e importantes na *botnet*, especialmente nas centralizadas, porque durante a fase de *rally* o *bot* precisa encontrar o C&C e, para isso, precisa traduzir o domínio especificado para o centro de comando e controle.

Os registros de acesso obtidos pelo sensor de coleta são mapeados em Grafos de Contatos, modelados através de um grafo direcionado bipartido e ponderado $Gdns = (U, V, E, P)$, cujos nós são formados pelos endereços IP ($U$) e os domínios ($V$). Existirá uma aresta entre $u$ ($u \in U$) e $v$ ($v \in V$) se $u$ solicitar uma consulta DNS para o domínio $v$. A quantidade de consultas determinará o peso das arestas ($P$).

O objetivo é a partir do grafo de contato identificar sub-grafos que formam comunidades de interesses suspeitas, permitindo encontrar o conjunto de *bots* que pertençam

 ©2012 SBC — Soc. Bras. de Computação

a esta *botnet*. A identificação das comunidades deve ser baseada nas características dos *bots*, mapeadas nas propriedades dos grafos através da adoção das seguintes premissas:

I. **Interesse comum**: Os *bots* normalmente infectam mais de uma máquina na mesma rede [Coskun et al. 2010]. Essa caracterítica é mapeada através da análise do grau de entrada do nó de destino, que deve ser maior que 2 ($\theta \geq 2$);
II. **Comportamento robótico**: O *bot* é um *software* pré-programado, então a comunicação dos *hosts* infectados com a *botnet* terá um comportamento semelhante, por exemplo, realizando o mesmo número de consultas. Essa característica é mapeada através da análise dos pesos das arestas que incidem em determinado domínio. O comportamento robótico será caracterizado pelo desvio padrão ($\sigma$) do peso das arestas ($P$) estar abaixo de um determinado limiar ($\sigma \leq \alpha$) ou ser igual a zero;
III. ***Stealthiness***: Com o objetivo de evadir dos mecanismos de detecção, os *bots* empregam mecanismos que tornam sua comunicação de difícil detecção, realizando poucas consultas a muitos domínios distintos. Essa característica pode ser mapeada através do peso baixo das arestas que incidem em determinado domínio ($P \leq \beta$).

Com base nessas premissas foi implementado no raciocinador um algoritmo para análise do grafo $Gdns$ composto pelos seguintes passos:

1. Filtragem inicial;
2. Separação em subgrafos desconexos para cada domínio $v$ ($v \in V$);
3. Busca por domínios que possuem arestas incidentes com desvio padrão baixo ($\sigma \leq \alpha$);
4. Busca por arestas com baixo peso ($P \leq \beta$).

A filtragem inicial tem por objetivo reduzir o tamanho do grafo e otimizar a busca por domínios suspeitos, removendo aqueles que com certeza não são C&C e que não se enquadram na premissa (I). A remoção foi realizada através da aplicação das seguintes regras da base de conhecimento:

- Aplicar uma *whitelist* com o objetivo de retirar os nós que não são C&C, por exemplo, domínios $gTLD$ .gov, .edu, .mil; $ccTLD$ .gov.br, .edu.br, .mil.br, .eb.br; ou domínios legítimos, como, facebook.com, google.com, entre outros;
- Retirar os nós de destino que possuem grau de entrada menor que 1 ($\theta \leq 1$);

O grafo de contatos filtrado ($Gdns' = (U, V, E, P)$) é separado em subgrafos desconexos para cada domínio $v$ ($v \in V$). O subgrafo representa a comunidade de nós que acessam determinado domínio. Com o objetivo de mapear a característica robótica de um *bot*, são mantidos apenas os subgrafos que possuem arestas com baixa variação no número de contatos ($\sigma \leq \alpha$), de acordo com a premissa (II). O valor de $\alpha$ é obtido da base de conhecimento.

Os subgrafos resultantes da etapa (3) indicam nós suspeitos de pertencerem a uma *botnet*, porém, ainda podem existir nós que não realizem atividade maliciosa, mas se enquadram nessas características. Um exemplo, são as consultas realizadas para domínios que são contadores de acesso a páginas *web* (http://statcounter.com/) que possuem comportamento semelhante, pois, também são *softwares* pré-programados.

Com o objetivo de remover os domínios que não são *bots* é aplicada a etapa (4) do algoritmo, baseada na premissa (III). São removidos todos os subgrafos que possuem arestas com pesos superior a $\beta$. O valor de $\beta$ é obtido da base de conhecimento.

 ©2012 SBC — Soc. Bras. de Computação

Os subgrafos resultantes das 4 etapas do algoritmos contém os endereços IP e domínios suspeitos de pertencerem a uma *botnet*. Para uma confirmação, o raciocinador aciona o robô de leitura com o objetivo de confirmar se determinado subgrafo é uma comunidade *botnet*. O robô pode confirmar, por exemplo, através da análise semântica do site ou da verificação se determinado domínio está em uma *blacklist*.

Confirmada a participação, o raciocinador aciona o robô de atuação para realizar alguma ação de bloqueio, por exemplo, criando uma regra no *firewall*, removendo o domínio, comunicando o administrador da rede, entre outras.

## 5. Resultados Preliminares

Os primeiros resultados foram obtidos com a análise do *Log* DNS de uma instituição de ensino, no período de período de 07/07 à 19/07/2011 e 24/02 à 04/04/2012. Entretanto, devido a limitação de espaço para o artigo serão apresentados resultados para 3 dias.

A tabela 1 apresenta um resumo do número de endereços IP e domínios distintos presentes no *log*, de acordo com cada fase do algoritmo implementado no raciocinador. Como pode ser verificado, após a etapa (1) houve uma redução significativa, superior a 97%, do número de endereços IP e domínios distintos.

Para determinar os valores para $\alpha$ e $\beta$ aplicados nas etapas (3) e (4), respectivamente, realizou-se uma busca de domínios suspeitos nos acessos ao servidor de DNS. Foram considerados domínios suspeitos os que pertenciam ao $ccTLD$ .cn, de acordo com [Rajab et al. 2006], 95% destes domínios estão infectados, e ao .ws, devido a não ser comum um acesso a esse site considerando acessos iniciados no Brasil.

A partir dos domínios considerados suspeitos foi adotada a mediana e o desvio padrão do número de contatos para determinar os valores de corte $\alpha = 0.3$ e $\beta = 1.3$. O número de endereços e domínios restantes após cada fase são apresentados na tabela 1.

**Tabela 1. Número de endereços e domínios distintos para cada fase do algoritmo**

| Etapa | Dia 12/03/12 | | Dia 13/03/12 | | Dia 14/03/12 | |
|---|---|---|---|---|---|---|
| | # IP | # Dom | # IP | # Dom | # IP | # Dom |
| Gdns | 20.529 | 238.852 | 30.027 | 383.977 | 37.355 | 499.777 |
| 1 | 540 | 6.962 | 784 | 11.428 | 949 | 15.040 |
| 2 | 540 | 6.962 | 784 | 11.428 | 949 | 15.040 |
| 3 | 290 | 1.359 | 260 | 2.244 | 122 | 2.918 |
| 4 | 245 | 1.149 | 221 | 1.899 | 104 | 2.492 |

Para confirmar se os hosts e os domínios pertenciam ou não a uma *botnet* foi realizada uma verificação dos sites e foi detectado que 5 endereços IPs e 253 domínios pertenciam à *botnet* Conficker. Dos 5 endereços encontrados, 3 pertencem a mesma subrede (200.213.8y.xxx) e os outros dois endereços são de subredes distintas (200.20.12y.xxx e 200.20.17z.kkk). Os domínios consultados não possuem um nome que apresente um significado relevante, por exemplo, *vbhtzo.ws* e *vqtqq.cn*.

A vantagem da solução está na taxa de acerto para os domínios e endereços suspeitos, pois, foram todos detectados. Entretanto, a desvantagem da solução está relacionada a alta taxa de falsos positivos, superior a 90%. O aprimoramento na criação da *whitelist* contribuirá para a redução desse valor.

 ©2012 SBC — Soc. Bras. de Computação

## 6. Conclusão

Os resutlados preliminares demonstraram que a arquitetura e o algoritmo proposto para o protocolo DNS apresentaram resultados satisfatórios na detecção de máquinas e domínios suspeitos. Entretanto são necessários estudos para aprimorar a obtenção da *whitelist*, com o objetivo de reduzir significativamente a taxa de falso positivos; para a determinação de $\alpha$ e $\beta$ com o objetivo de aumentar a precisão na detecção dos *bots*; e para a criação de mecanismos para permitir ao robô de leitura confirmar se um determinado domínio ou endereço suspeitos pertencem a uma *botnet*.

## Referências

Ceron, J. M., Granville, L. Z., and Tarouco, L. M. R. (2010). Uma arquitetura baseada em assinaturas para mitigação de botnets. In *X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 105–118.

Coskun, B., Dietrich, S., and Memon, N. (2010). Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. In *Proc. of the 26th Annual Computer Security Applications Conference*, ACSAC 10, New York. ACM.

Dagon, D., Gu, G., Lee, C. P., and Lee, W. (2007). A taxonomy of botnet structures. In *Computer Security Applications Conf., 2007. ACSAC 2007. 23th Annual*.

Gu, G., Perdisci, R., Zhang, J., and Lee, W. (2008). BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proc. of the 17th Conf. on Security symposium*, pages 139–154, Berkeley, CA, USA. USENIX.

Rajab, M. A., Zarfoss, J., Monrose, F., and Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, page 41–52, New York. ACM.

Sanchez, F., Duan, Z., and Dong, Y. (2011). Blocking spam by separating end-user machines from legitimate mail server machines. In *Proc. of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS '11, New York. ACM.

Snort (2006). Snort. http://www.snort.org.

Stone-Gross, B., Cova, M., Gilbert, B., Kemmerer, R., Kruegel, C., and Vigna, G. (2011). Analysis of a botnet takeover. *Security Privacy, IEEE*, 9(1):64 –72.

Zeidanloo, H. R., Shooshtari, M. J., Amoli, P. V., Safari, M., and Zamani, M. (2010). A taxonomy of botnet detection techniques. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE Inter. Conf. on*, volume 2, pages 158 –162.

Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., and Luo, X. (2011). Detecting stealthy P2P botnets using statistical traffic fingerprints. DNS 2011, pages 121–132, Los Alamitos, CA, USA. IEEE Computer Society.

Zhu, Z., Lu, G., Chen, Y., Fu, Z. J., Roberts, P., and Han, K. (2008). Botnet research survey. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 967 –972.

 ©2012 SBC — Soc. Bras. de Computação

# Avaliação do Classificador ARTMAP Fuzzy em redes 802.11 com Criptografia Pré-*Robust Security Network* (WEP e WPA)

**Nelcileno Araújo[1], Ruy de Oliveira[2], Ailton Akira Shinoda[3], Ed'Wilson Tavares Ferreira[2], Valtemir Emerêncio do Nascimento[2]**

[1]Instituto de Computação - Universidade Federal de Mato Grosso (UFMT) – Cuiabá, MT – Brasil

[2]Departamento de Informática - Instituto Federal de Educação. Ciência e Tecnologia de Mato Grosso (IFMT) – Cuiabá, MT – Brasil

[3]Departamento de Engenharia Elétrica - Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP) - Ilha Solteira, SP – Brasil

nelcileno@ic.ufmt.br,{ruy,ed,valtemir}@cba.ifmt.edu.br, shinoda@dee.feis.unesp.br

***Abstract.*** *Recent years have seen strong growth in the use of 802.11 wireless technology (Wireless LAN) and the security mechanisms implemented by IEEE 802.11i e IEEE 802.11w amendments have been shown to be ineffective in combating attacks against the availability of WLAN services. This paper evaluates the performance of Fuzzy ARTMAP classifier in the detection of a group of denial of service (DoS) attacks in a real WLAN supporting WEP and WPA encryption. Fuzzy ARTMAP neural network was selected because it is able to retain previously acquired knowledge and adapt to new classification patterns. The results obtained in terms of detection rate and false alarms rate suggest that there is a need for a metaheuristic to provide more reliable parameters for the performance of the classifier and the selection of most representative features about intrusive behavior exists in DoS attacks.*

***Resumo.*** *Nos últimos anos têm-se percebido um forte crescimento no uso da tecnologia sem fio 802.11 (Wireless LAN) e os mecanismos de segurança implementados pelas emendas IEEE 802.11i e IEEE 802.11w têm se mostrado pouco eficazes no combate a ataques contra a disponibilidade dos serviços da WLAN. Neste artigo avalia-se o desempenho do classificador ARTMAP Fuzzy na detecção de um grupo de ataques de negação de serviço (DoS) numa rede WLAN real com suporte a criptografia WEP e WPA. A rede neural ARTMAP Fuzzy foi escolhida pela sua capacidade de preservar o conhecimento anteriormente adquirido e adaptar-se a novos padrões de classificação. Os resultados obtidos demonstram que há a necessidade de uma metaheurística para fornecer parâmetros mais confiáveis para a execução do classificador e a seleção de atributos mais representativos do comportamento intrusivo existente nos ataques de DoS.*

## 1. Introdução

Os números apresentados pela IDC Brasil, onde 55% dos computadores vendidos em 2011 foram *notebooks*/*netbooks* e as vendas de *smartphones* em 2011 cresceram 84%

 ©2012 SBC — Soc. Bras. de Computação

com relação ao ano anterior [IDC - Brasil, 2012a][IDC - Brasil, 2012b] demonstram uma popularização de dispositivos com interface de comunicação sem fio 802.11, propiciando uma disseminação do uso de redes locais sem fio (WLAN) em ambientes que fornecem acesso à Internet para seus clientes, tais como: shopping centers, universidades, aeroportos, dentre outros.

No entanto, é bem conhecido que devido à natureza aberta do acesso sem fio, a segurança de uma WLAN levanta muitas considerações e um conjunto de extensões de segurança ao padrão IEEE 802.11 já foram propostas [IEEE 802.11, 1999], [IEEE 802.11i, 2004] e [IEEE 802.11w, 2009]. Contudo, estas extensões combatem com sucesso as vulnerabilidades relacionadas ao acesso não autorizado, integridade e a confidencialidade, mas deixam a desejar na proteção da disponibilidade dos serviços de redes numa WLAN [Ahmad e Tadakamadla, 2011].

Os ataques contra a disponibilidade dos recursos da rede são popularmente conhecidos como negação de serviço (DoS – *Denial of Service*). Uma das principais fontes para ataques de DoS numa WLAN são os quadros de gerenciamento e controle pois são responsáveis pela associação e controle de acesso ao meio sem fio entre os clientes da rede e o ponto de acesso [Bicakci e Tavli, 2009].

Uma contramedida para evitar os problemas de indisponibilidade na WLAN é a utilização de sistemas detectores de intrusão (IDS) como uma segunda camada de defesa que monitore constantemente as ondas de rádio, de forma a detectar possíveis explorações destas vulnerabilidades. O IDS pode analisar o tráfego monitorado a partir de um conjunto de assinaturas de ataques (detecção baseada em assinaturas) ou pela definição de um perfil esperado para um dispositivo pertencente à rede (detecção baseada em anomalia).

Este artigo apresenta uma avaliação do classificador ARTMAP *Fuzzy* [Carpenter, G. *et al*., 1992] na tarefa de identificação de um grupo de ataques DoS por meio dos quadros de gerenciamento de uma rede WLAN real com criptografia Pré-RSN (*Robust Security Network*), ou seja empregam apenas os protocolos criptográficos WEP (*Wired Equivalent Privacy*) e WPA (*Wi-Fi Protected Access*).

## 2. Trabalhos Relacionados

A proposta apresentada por [Ahmad, Abdullah e Alghamdi, 2010] mostra que o classificador ARTMAP Fuzzy na tarefa de detecção de intrusão possui um baixo desempenho nas avaliações realizadas, mas a base de dados utilizada nos experimentos é originada de uma rede cabeada e não utilizam nenhum mecanismo de otimização computacional para maximizar a classificação correta.

Já a proposta relatada por [Vilakazi e Marwala, 2007] emprega a técnica de algoritmos genéticos, como mecanismo de otimização computacional, para obter parâmetros de configuração que proporciona ao classificador ARTMAP Fuzzy alcançar um ótimo desempenho de 100% na fase de detecção. No entanto, a base de dados avaliada não representa uma rede local sem fio.

A abordagem relatada por [Santra, Nagaranjan e Jinesh, 2012] utiliza a rede neural ARTMAP Fuzzy na detecção de intrusão em redes heterogêneas sem fio e aplica a metaheurística otimização por enxame de partículas (PSO) para minimizar os erros da classificação. Apesar disso, o mecanismo proposto não se enquadra na identificação de ataques DoS em quadros de gerenciamento pois os autores utilizam sua proposta para fornecer um modelo de controle de acesso.

 ©2012 SBC — Soc. Bras. de Computação

## 3. Abordagem proposta para avaliação do classificador ARTMAP Fuzzy

Para avaliar o classificador ARTMAP Fuzzy, foram realizados experimentos num cenário controlado, cuja topologia é representada na Figura 1, para gerar a base de dados a ser empregada como base de conhecimento.

A rede analisada tinha a seguinte composição: três estações sem fio e um ponto de acesso (AP). A estação 1 injetava tráfego normal na rede (HTTP, FTP). A estação 2 utilizou a ferramenta *Aireplay* [Aireplay, 2011] para realizar, simultaneamente, os quatro ataques (*chopchop*, duração, deautenticação e fragmentação) pré-definidos. A estação 3 usou a ferramenta *Wireshark* [Wireshark, 2011] para realizar a captura do tráfego transeunte (normal e intruso) na rede.

A seguir, foi realizado um pré-processamento nos dados capturados para se extrair os seguintes campos do cabeçalho MAC (*Media Access Control*) dos quadros de gerenciamento: *protocol version*, *type*, *subtype*, *to DS*, *from DS*, *more fragment*, *retry*, *power managment*, *more data*, WEP, *order*, *duration*, *address1*, *address2*, *address3* e *sequence control*, , pois também faz parte da pesquisa ver o impacto destas informações na especificação de assinaturas para os ataques de DoS. Por último, inseriu-se em cada amostra sua respectiva categoria de reconhecimento.

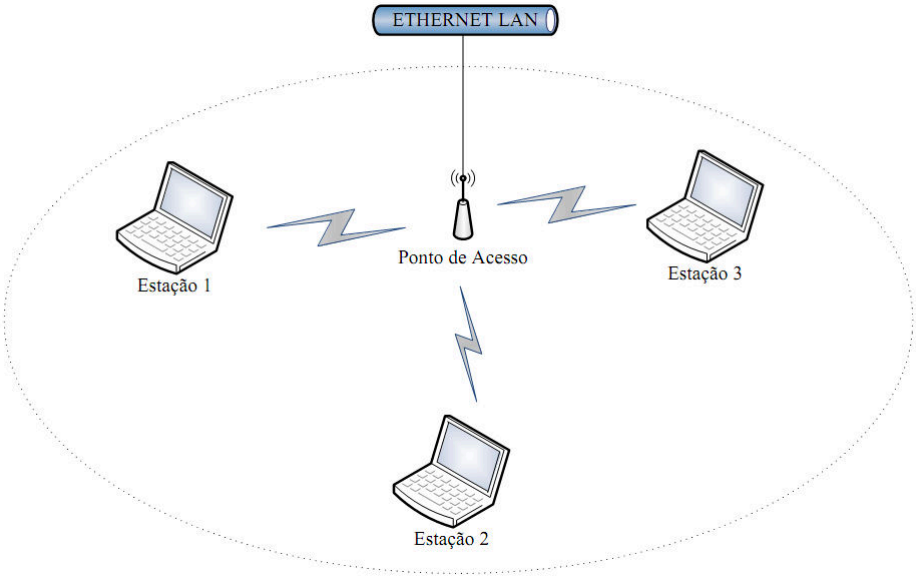



**Figura 1. Topologia da rede WLAN aplicada na geração da base de dados**

A base de dados empregada na avaliação do classificador é criada pela captura de quadros de gerenciamento da rede WLAN utilizada nos experimentos, em que a rede esteve sob condição "normal", sem ataques, e sob os ataques *Chopchop* [Bittau, Rendley e Lackey, 2006], deautenticação [Bellardo e Savage, 2003], duração [ Bellardo e Savage, 2003] e fragmentação [Bittau, Rendley e Lackey, 2006].

Essas quatro categorias usadas na geração dos ataques foram escolhidas porque elas exploram de forma efetiva as vulnerabilidades de disponibilidade nas redes 802.11 com criptografia Pré-RSN. Os ataques de duração e deautenticação afetam a capacidade da estação base de gerenciar o acesso à infraestrutura da rede [Berlado e Savage, 2003]. Os ataques de *chopchop* e fragmentação exploram as vulnerabilidades dos mecanismos

 ©2012 SBC — Soc. Bras. de Computação

criptográficos (WEP e WPA) para indisponibilizar os serviços da rede [Bittau, Rendley e Lackey, 2006].

A base de dados gerada pelos dados coletados nos experimentos foi dividida em três conjuntos: treinamento, validação e teste.

O conjunto de treinamento é formado por 9600 amostras com a seguinte constituição: 6000 amostras de tráfego normal e 3600 amostras de tráfego intruso. A validação da rede neural ARTMAP Fuzzy ocorre pela aplicação de um conjunto de validação formado por novas 6400 amostras, constituída de 4000 amostras de tráfego normal e 2400 amostras de tráfego intruso. Após a rede treinada e validada obtêm-se os parâmetros da rede neural ARTMAP Fuzzy, os quais são: parâmetro de escolha ($\alpha$) = 0.01, parâmetro de vigilância da rede $ART_a$ ($\rho_a$) = 0.7, parâmetro de vigilância da rede $ART_b$ ($\rho_b$) = 1, parâmetro de vigilância do mapa associativo ($\rho_{ab}$) = 0.99 e taxa de treinamento ($\beta$) = 1. A partir disso, pode-se entrar com o conjunto de teste para que o classificador diagnostique suas saídas. O conjunto de teste possui novas 8200 amostras divididas em 5000 amostras de tráfego normal e 3200 amostras de tráfego intruso.

Para avaliar o desempenho do classificador ARTMAP Fuzzy, os seus resultados foram comparados com outros três classificadores: *Suport Vector Machine* (SVM), Rede Neural *Multilayer Perceptron* com treinamento *Backpropagation* (MPBP) e Rede Neural *Radial Basis Function* (RBF).

A ferramenta WEKA [WEKA, 2011] é a solução computacional utilizada para implementar a avaliação do classificador ARTMAP Fuzzy com os outros classificadores devido ao fácil manuseio e uma biblioteca rica de técnicas de classificação de padrões.

Os critérios de avaliação de um classificador de padrões na detecção de intrusão podem ser realizados pelo cálculo das seguintes métricas: Verdadeiros Positivos (TP) - identifica uma atividade intrusiva corretamente; Verdadeiro Negativo (TN) - identifica uma atividade não-intrusiva corretamente; Falso Positivo (FP): identifica uma ação não-intrusiva como sendo intrusiva; Falso Negativo (FN): identifica uma atividade intrusiva como sendo não-intrusiva.

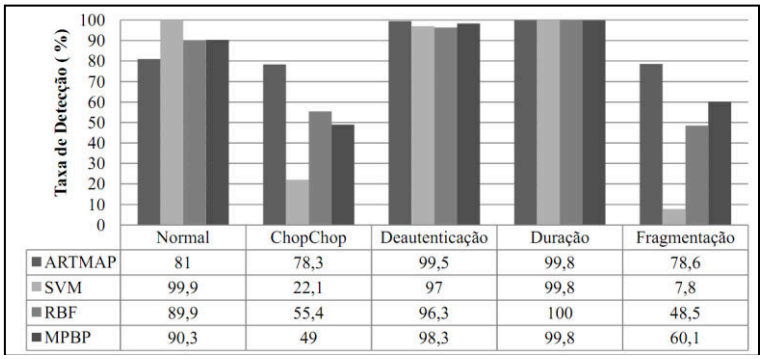
As métricas mais populares na avaliação de desempenho de um IDS são taxa de detecção e taxa de falsos alarmes [Wu e Banzhaf, 2010]. A taxa de detecção é definida como TP/(TP+FN) e por fim, a taxa de falsos alarmes é definida como FP/(TN+FP).Um classificador de padrões obtém um bom desempenho na detecção de intrusão quando atinge uma alta taxa de detecção e uma baixa taxa de falsos alarmes [Wu e Banzhaf, 2010]. A seguir, apresentam-se os resultados obtidos nesta avaliação de desempenho.

## 4. Resultados Obtidos

Para verificar o desempenho do classificador de padrões foram aplicadas 8200 amostras do conjunto de teste nos dois cenários de comparação apresentados neste artigo.

A Figura 2 apresenta o primeiro cenário, usado na avaliação da taxa de detecção dos classificadores, segundo as categorias relatadas na base de dados para reconhecimento. Observa-se que o classificador ARTMAP Fuzzy possui um baixo desempenho na identificação de tráfego normal, enquanto no reconhecimento dos ataques de DoS apresentados na base de conhecimento atingiu desempenho igual ou superior aos classificadores avaliados.

 ©2012 SBC — Soc. Bras. de Computação

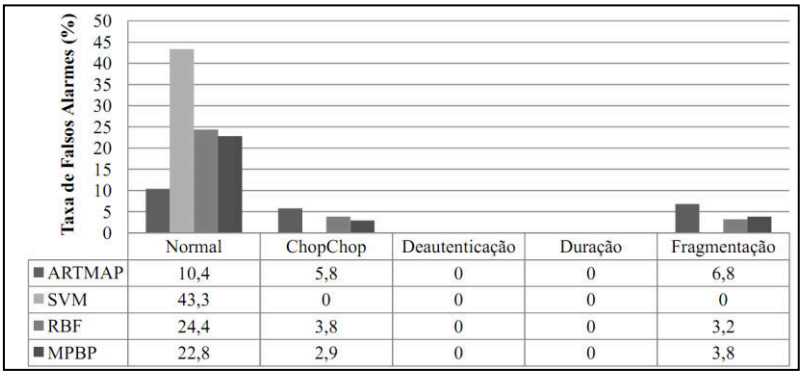



| | Normal | ChopChop | Deautenticação | Duração | Fragmentação |
|---|---|---|---|---|---|
| ARTMAP | 81 | 78,3 | 99,5 | 99,8 | 78,6 |
| SVM | 99,9 | 22,1 | 97 | 99,8 | 7,8 |
| RBF | 89,9 | 55,4 | 96,3 | 100 | 48,5 |
| MPBP | 90,3 | 49 | 98,3 | 99,8 | 60,1 |

**Figura 2. Taxa de detecção dos classificadores**

A baixa eficiência na detecção de atividade normal pode ser ocasionada por não ocorrer uma seleção de atributos mais representativos nas instâncias pertence aos registros auditados ou por não implementar um mecanismo de otimização que ofereça melhores valores para os parâmetros de configuração da rede ARTMAP Fuzzy.

A avaliação da taxa de falsos alarmes apresentada pelos classificadores, ilustrado na Figura 3, demonstra que o classificador ARTMAP Fuzzy apresenta a maior taxa de falsos alarmes entre os classificadores avaliados no reconhecimento dos quatro ataques de DoS executados no cenário. Uma possível razão para este baixo desempenho do classificador ARTMAP Fuzzy é os campos do quadro MAC serem insuficientes para especificar assinaturas bem definida das categorias de ataques representadas nos experimentos.




| | Normal | ChopChop | Deautenticação | Duração | Fragmentação |
|---|---|---|---|---|---|
| ARTMAP | 10,4 | 5,8 | 0 | 0 | 6,8 |
| SVM | 43,3 | 0 | 0 | 0 | 0 |
| RBF | 24,4 | 3,8 | 0 | 0 | 3,2 |
| MPBP | 22,8 | 2,9 | 0 | 0 | 3,8 |

**Figura 3. Taxa de falsos alarmes dos classificadores**

Um dos maiores entraves no aumento da classificação correta da rede neural ARTMAP Fuzzy refere-se à sua sensibilidade à sobreposição estatística entre as classes. A sensibilidade pode ocasionar crescimento descontrolado da quantidade de categorias de reconhecimento, causando aumento na complexidade de memória e computacional. Como também, possível degradação na classificação correta [Lerner e Guterman, 2008].

## 6. Conclusões

Neste artigo é relatada uma avaliação do classificador ARTMAP Fuzzy na detecção de um grupo de ataques DoS (*chopchop*, deautenticação, duração e fragmentação) numa rede WLAN real com suporte a criptografia WEP e WPA. Esta avaliação ocorre por meio de uma comparação do classificador estudado com um grupo de classificadores comumente empregados na literatura para detecção de intrusos.

 ©2012 SBC — Soc. Bras. de Computação

A rede neural ARTMAP Fuzzy apresenta um baixo desempenho no reconhecimento de atividade normal e uma alta de taxa de falsos alarmes na identificação do grupo de ataques DoS analisados. Isto demonstra que os campos do quadro MAC utilizados são insuficientes para gerar assinaturas confiáveis para a identificação dos ataques, necessitando um processo de seleção de atributos que possa escolher características mais representativas do comportamento intrusivo existente nos ataques DoS estudados. Além disso, a ausência de uma técnica de otimização computacional para a geração dos parâmetros de configuração do classificador pode ter contribuído para o baixo desempenho dele, visto que em trabalhos, tais como [Vilakazi e Marwala, 2007] e [Santra, Nagaranjan e Jinesh, 2012], o uso de uma metaheurística auxilia na definição dos parâmetros da rede neural ARTMAP Fuzzy.

Na fase atual da pesquisa pretende-se tratar o problema da sensibilidade da rede neural ARTMAP Fuzzy implementando a técnica de metaheurística baseada em inteligência de enxame para obter melhores parâmetros de configuração para o classificador. Como também implementar as emendas de segurança IEEE 802.11i e IEEE 802.11w na WLAN para ver o desempenho do classificador no reconhecimentos de ataques de DoS referente a este tipo de emendas.

## Referências

Ahmad, I., Abdullah, A. e Alghamdi, A. (2010) "Towards the selection of best neural network system for intrusion detection", International Journal of the Physical Sciences, vol. 5, n. 12, p. 1830-1839.

Ahmad, M. S., Tadakamadla, S. (2011) "Short paper: security evaluation of IEEE 802.11w specification", Proceedings of the fourth ACM conference on Wireless network security, p. 53 - 58.

Aircrack (2011), http://www.aircrack-ng.org/.

Bellardo, J. e Savage, S. (2003) "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", Proceedings of the 12th Conference on USENIX Security Symposium, vol. 12, p. 15-28.

Bicakci, K. e Tavli, B. (2009) "Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks", Computer Standards & Interfaces, v. 31, n. 5, p. 931-941.

Bittau, A., Handley, M. e Lackey, J. (2006) "The Final Nail in WEP's Coffin", Proceedings of the 2006 IEEE Symposium on Security and Privacy, p. 386-400.

Carpenter, G., Grossberg, S., Markuzon, N., Reynold, J. e Rosen, D. (1992) "Fuzzy ARTMAP: A neural network for incremental supervised learning of analog multidimensional maps", IEEE Transactions on Neural Network, vol. 3, n. 5, p. 689-713.

IDC – BRASIL (2012) "Estudo da IDC revela que foram vendidos aproximadamente 9 milhões de smartphones no Brasil em 2011", http://www.idclatin.com/news.asp?ctr=bra&year=2012&id_release=2213, Abril.

IDC – BRASIL (2012) "Brasil comercializa 15,4 milhões de computadores em 2011 e se consolida na terceira posição do mercado mundial, segundo pesquisa da IDC", http://www.idclatin.com/news.asp?ctr=bra&year=2012&id_release=2200, Abril.

 ©2012 SBC — Soc. Bras. de Computação

IEEE Std. 802.11 (1999) "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", p. 1-512.

IEEE Std. 802.11 (2004) "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment 6: Medium Access Control (MAC) Security Enhancements ", p. 1-175.

IEEE Std. 802.11w (2009) "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Protected Management Frames", p. 1-92.

Lerner, B. e Guterman, H. (2008) "Advanced Developments and Applications of the Fuzzy ARTMAP Neural Network in Pattern Classification", In *Studies in Computational Intelligence*, vol. 137, p. 77-107, Spring-Verlag.

Santra, A. K., Nagarajan, S. e Jinesh, V. N. (2012) " Intrusion Detection in Wireless Networks using FUZZY Neural Networks and Dynamic Context-Aware Role based Access Control Security (DCARBAC)", International Journal of Computer Applications, vol. 39, n. 4, p. 23-31.

Vilakazi, C. e Marwala, T. (2007) "Application of Feature Selection and Fuzzy ARTMAP to Intrusion Detection", Proceedings of the Conference International on Systems, Man and Cybernetics, p. 4880-4885.

Weka (2011), http://www.cs.waikato.ac.nz/ml/weka/.

Wireshark (2011), http://www.wireshark.org/.

Wu, S. e Banzhaf, W. (2010) "The Use of Computational Intelligence in Intrusion Detection Systems: A Review", In *Applied Soft Computing*, vol.10, p. 1-35.

 ©2012 SBC — Soc. Bras. de Computação

# Identity Management Requirements in Future Internet

**Jenny Torres**[1], **Ricardo Macedo**[2], **Michele Nogueira**[2], **Guy Pujolle**[1,3]

[1]Laboratory in Computer Science (LIP6) – University Pierre and Marie Curie
Paris – France

[2]Informatics Department – Federal University of Paraná
Curitiba – Brazil

[3]IT Convergence and Engineering – POSTECH
Korea

{Jenny.Torres,Guy.Pujolle}@lip6.fr, {rmacedo,michele}@inf.ufpr.br

***Abstract.*** *The characteristics of the Future Internet and the emerging technologies result in new requirements, in which user security issues are highlighted. Identity Management requirements are linked to the development of systems able to prevent unauthorized use of digital identities, information overload, and to enhance user privacy. This paper highlights the Identity Management requirements in Future Internet context, based on its characteristics. Security and privacy were identified as key factors since they determine the overall trustworthiness of a system in terms of confidentiality, integrity and availability. Further, we introduce a discussion, in which we present the relationship between the Identity Management requirements and a future Internet scenario.*

## 1. Introduction

The increasing use of the Internet and the fast advance of new technologies have motivated the development of the Future Internet (FI) [Paul et al. 2011]. FI is supported by a network infrastructure, which increases the dependence on distributed information and decentralized control, requiring strong guarantees of security [Chim et al. 2011, Gomez-Skarmeta et al. 2010, Weber et al. 2010]. The main goal of the network infrastructure, known as Future Network, is to provide service-related functionalities independent of subjacent technologies. This network has as main characteristics the heterogeneity, dynamicity, interdependence and autonomy, increasing security issues and making difficult to protect services and applications.

In FI, Identity Management (IdM) plays a fundamental role. As a FI service, an IdM can assist the network infrastructure by providing information about user's profile, service features and access policies, in order to improve the efficiency of other services and to ensure network operation transparency, such as mobility, routing and others. IdM has attracted attention in recent years as an efficient way to provide trust between entities, protect or mitigate the effects of malicious entities, manage user's identities, identify entities in a system and control their access to resources [Sabena et al. 2010, Sarma and Girao 2009]. Features, such as security, determine the overall trustworthiness of a system in terms of performance, robustness and privacy. In the new paradigm of FI, many security problems are related to the concept of identity, such as the explosion of the number of identities, identity theft and identity impersonation [Van Rooy and Bus 2010, Wan et al. 2010].

 ©2012 SBC — Soc. Bras. de Computação

The goal of this paper is to identify IdM requirements in FI based on its characteristics. There are many factors driving the adoption of IdM solutions, since their requirements vary depending on the proposal, applications and environment. The paper proceeds as follows. Section 2 provides an overview about the main characteristics of FI. Section 3 describes the core of IdM, which includes main definitions, operations and components. Section 4 presents the requirements for IdM systems in FI. Section 5 presents a discussion on the interaction between the requirements previously defined with a specific scenario. Finally, Section 6 concludes the paper.

## 2. Future Internet

In this section, we present FI from a service-oriented perspective and less protocol-oriented. For most users, the Internet is defined as a set of services, since they associate it with easy access to information and search engines, video and music availability or entertainment services. Understanding how users employ FI and the relationship between services with underlying communication networks is important to identify the requirements that the network infrastructure must support. Among those requirements, we have the connection of existing networks, survivability, support of multiple types of services, distributed management, cost-effectiveness, and resource accountability [Levin et al. 2009].

Among the advances of FI, shaped by Internet users behavior, we highlight the *personalized services*, which are strongly related to social networks. They digitally represent personal relationships and provide an important reference for identity and confidence issues in the Internet. Further, advances that should assist in defining FI are privacy and anonymity, which are becoming increasingly important for Internet users. On the other hand, as devices on cellular networks had been enabled for the Internet Protocol (IP) and sensors have been added to some networks, the Internet has gone mobile. The role of mobility has been changing over time and now it is considered one of the major features in FI. New challenges are centered on mobile access to networked information objects. FI is mainly related to the delivery of new services that will be provided in a pervasive and ubiquitous way, that is, anywhere, anytime and on any device through a selected communication technology. This is a heterogeneous environment where there are challenges as: security, quality of service (QoS) and costs. FI will integrate services offered by traditional networks and innovative IP services in a single service platform, and this integration should be as transparent as possible to the end user [Salsano et al. 2008].

## 3. Identity Management

This section overviews fundamentals of IdM, main concepts, operations and actors in IdM systems [Bosworth et al. 2005, ITU-T 2009]. An *entity* can be a person, a network service, a network computing device or a mobile telephone device. They use credentials and have a life cycle separated from any associated identity, identifier or credential life cycle. A *credential* is used to prove an identity to a system. There can be various types of credentials, but all are involved in ensuring a system that an entity truly has the right to use a particular identity. *Identity* is an instrument used by an entity in order to provide information about itself to the system. It is always associated with an entity or generally formed by a unique identifier, which is used to prove ownership of the identity. An *identifier* is a unique index for an identity. It must be unique within any given system, but might be reused across several. The main difference between identifiers and credentials

 ©2012 SBC — Soc. Bras. de Computação

is the fact that an identifier must necessarily be unique and the credential not. Even being unique, the identifier can be the union of others identifiers not unique.

The four main operations of an IdM system are: identification, authentication, authorization and accounting [Bosworth et al. 2005]. *Identification* is the action of an entity providing an identity to the system through an identifier. The identification tells the system who user is trying to connect, but it offers no proof. To obtain proof of this identity, the system makes the *authentication* process where the connecting entity provides credentials for the claimed identity, known only by it. After authentication, the system can be certain that the entity is the rightful owner of the chosen identity. Each entity will have a set of actions allowed on the system. These permissions are known as privileges and are determined in the *authorization* stage. The system must look up the relevant privileges for the authenticated identity. Therefore there must be a binding of identity to privileges. Finally, *accounting* is a recording of what happened once authorization has been granted. It consists of a set of records, each one linking an established identity to an action.

In a general IdM system, we can identify three entities involved: user, identity provider and service provider [Cao and Yang 2010]. A *User (U)* is an entity that uses a service supplied by a service provider. Users use IdM systems to access services that require certification of their attributes by a third party. This is common in Internet access because users do not trust the security of data transmission. *Identity Provider (IdP)* is an entity that controls user's credentials and provides authentication services. *Service Provider (SP)* is an entity that offers one or more services from being accessed by potential users and uses the IdP services to authenticate a user.

## 4. Identity Management System Requirements in Future Internet

This section defines the requirements for IdM systems on the context of FI. Poorly designed IdM systems can aggravate existing security problems and create opportunities to extract personal information from users [Dhamija and Dusseault 2008]. We classify the requirements for IdM on the FI context as: *FI characteristics aware* and *IdM characteristics aware*, as illustrated in Figure 1. In the figure, we can see on the left side the requirements aware of FI characteristics and on the right side the requirements aware of IdM characteristics. The combination of these two kinds of requirements results in the requirements for IdM on the FI context.

In Figure 1, *privacy* is an important requirement in terms of law enforcement and user trust. The Internet and all technical communication have to comply with laws and regulations concerning the respective rights and privileges of the user and the provider. As FI involves the transfer of sensitive information between parties, protecting privacy can prevent personal information to be used improperly causing loose of autonomy and freedom. To maintain privacy, it should be possible for users to be anonymous, to use pseudonyms, to choose IdPs that do not link all user transactions at all SPs together and not to keep records of everything the user was doing. To ensure *security*, an IdM system has to be as robust as possible against attacks on the availability, integrity and confidentiality of its services and information. This is particularly important because of the concentrated amount of information about the user it stores and represents. Nevertheless, there are always risks of spying, manipulation and identity theft. Because of the amount of identity information stored and managed by organizations, security is an essential re-

 ©2012 SBC — Soc. Bras. de Computação

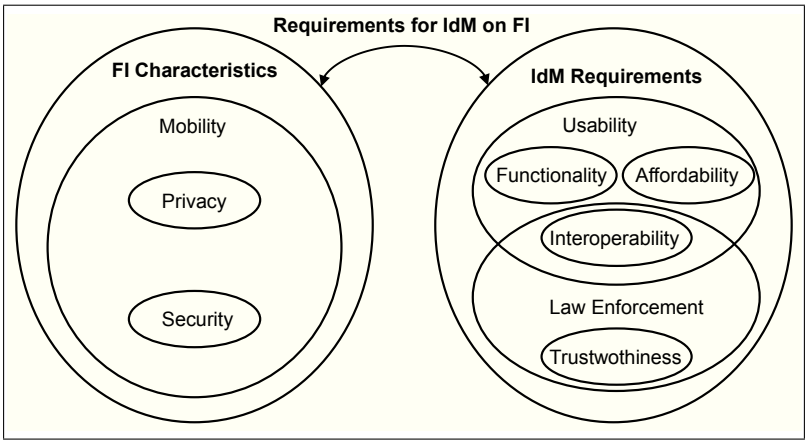



**Figure 1. Requirements for IdMs on Future Internet**

quirement for the SP. IdM systems require the user and the SP to place a large amount of trust in the IdP. All the identity information is stored at IdPs, and users can do nothing but trust them to preserve their privacy and secure their identity information.

In general, making IdM systems simple and easy to use reduces barriers to adoption. *Usability* refers to the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments [Levin et al. 2009]. A lack of usability can have a negative impact on functionality, security and privacy. Although many IdM systems claim to be designed with user in mind, most still have important usability issues [Alpár et al. 2011]. Considering the heterogeneous environment of FI and the complexity that can cause the lack of usability, IdM systems must be easy to use and its operations should be transparent for the user. Among some aspects missing in actual IdM systems there is the location independence, which means to create, manage, and use the identities independently of their current location and current device in use. Any enhancement of usability is more a question of *affordability*. Every technology needs to be affordable to become widely accepted. This applies to all kinds of users but can be rephrased to the question if the IdM system only adds overhead or enhances the functionality or quality of a given transaction. Organizations will look at IdM systems less from a cost but from a cost-effectiveness angle. In other words, the benefits of an IdM system need to outweigh its direct and indirect cost.

*Mobility* is very important in the growing use of portable devices. In a mobile environment, a user has several devices such as telephones, smart cards or RFID (Radio Frequency ID). As these devices have fixed identifiers, they are essentially providing a mobile identity. The mobility in IdM takes into account location data of users in addition to their personal data. Mobile IdM empowers users to manage their identities to enforce their security and privacy interests. Privacy and the protection against identity theft are important criteria for services that use mobile identities. However, in addition to privacy, also usability is important for the success of mobile IdM. Usability influences the correctness of security mechanisms. In mobile IdM, users must be able to control the disclosure of their identity and also their location.

The main *functionality* of IdM systems is to help the user to manage their identities. An IdM system has to provide the possibility of managing partial identities and

 ©2012 SBC — Soc. Bras. de Computação

identity data. It is also necessary for IdM systems to have interfaces to the communication partners, especially to digital networks. An IdM system can act as gateway for digital communication. This gateway functionality lets it to manage data exchange with all communication partners. IdM systems incorporate the gateway functionality by definition, because IdM is always a process between the user and another party. Typically, organizations will have to manage members' and associates' identity information, thus different kinds of identities. Functions for controlling this complexity and keeping it up-to-date are part of the main basic requirements of functionality that must be considered when proposing an IdM system.

*Trustworthiness* is a prerequisite for all transactions, which defines if a user trusts the SP or the system. Even in systems where the user has complete control over hardware, software and data flow, a certain amount of trust is still required because the complexity of the system demands transparency. Therefore, the reputation of software and hardware suppliers and SPs becomes an asset in the market. Although the notion of trust may depend on many factors, it is clear that privacy, security and usability are preconditions for trustworthiness. Also the law enforcement influences the perception of trust. Agencies responsible for control and *law enforcement*, such as police departments or criminal investigative practices are typically interested in collecting as much information as possible for giving evidence and make criminal proceedings easier and more effective. Any IdM system has to take care about the legal requirements for law enforcement of the countries where it should be used. However, these requirements are sometimes contradictory in different countries and even regions within a country, as the result of different cultures and realities.

*Interoperability* among existing systems is a basic requirement for an IdM system. IdM systems should implement interfaces compatible with international standards. It is possible that certain players will resist compatible interfaces in order to protect their market position. In such a case, the acquisition of critical mass for IdM systems as a product may be more difficult. Trust regulations may be able to regulate conventional trends in the market. Achieving interoperability across different contexts is impossible without a coherent semantic foundation in the culture and society where the IdM systems intend to be used.

## 5. Discussion

This section discusses how the requirements described in Section 4 can be observed in a realistic context of FI. The scenario is inspired by the Pervasive Healthcare example presented in [Akinyele et al. 2010]. We have chosen a pervasive healthcare context to search solutions that enable continuous monitoring of the health on chronic patients through the use of embedded wireless sensors. With this technology, it is also possible to obtain a larger amount of information about patient's health, including the creation of medications produced according to the specific needs of a person. In this scenario, it is necessary to make a communication between the sensors that will monitor patients and devices for storage of medical information. Therefore, the efforts of non-disclosure of data collected about patient health to unauthorized persons highlight the necessity for data *privacy*. Consider an attacker who removes the information stored about a patient or makes it unavailable. For instance, suppose that he is allergic to penicillin. In this case, this patient can receive a medication based on penicillin and as a result may manifest symptoms of

 ©2012 SBC — Soc. Bras. de Computação

faints. Thus, the need to ensure that data collected is exactly the same than that stored and to guarantee that it is always available when required show the necessity for *security*. *Usability* requirement is related to security too. Users must understand which security actions are required of them, to not disclose their personal data. While to make the use of data collected feasible for any type of user, it requires *affordability*.

In conjunction with the computerization of healthcare, there is the concept of Electronic Medical Records (EMR) [Benaloh et al. 2009]. The EMR is the on-line *availability* of information about a patient with restrictions access. Since this information may only be disclosed to the patient, your doctor or in an emergency, this information may be disclosed to other doctors. Based on the location of the patient in an emergency procedure could be created to make available information about the patient to the doctor that is closer to employ the requirement for *mobility*. The *functionality* can be achieved by considering the creation of several partial identities in order to use a specific identity for medical information, isolating this information from other identities. The requirement *trustworthiness* can be used by trust between the patient's body sensors and data storage service. Finally, consider two cities $A$ and $B$ each one has a unique IdM for health monitoring of patients using different technologies. The requirement for *interoperability* exists when a chronic patient travels from city $A$ to city B, for example.

## 6. Conclusions and Future Work

FI asserts that the digital world is becoming more flexible, interconnected and open. Boundaries between enterprises, organizations and government agencies are getting indistinct as people cover multiple roles and are involved in different activities across heterogeneous environments, creating new threats and issues. IdM need has a strategic role achieving this new world and addressing these new issues. A new generation of IdM solutions is needed to provide mechanisms to rapidly adapt and affront changing environments, in business, personal and social contexts.

In this paper, we highlighted the requirements of IdM for FI. IdM plays a key role in enabling personal and business activities along with interactions and transactions in the digital world. Although IdM has strong links with the management of security and privacy, the security component is only a small area, due to there are much more requirements and technologies needed to achieve IdM systems. IdM systems still have challenges today. Among these challenges, there are those related to security, privacy and usability that must be treated in order to become more suitable for FI. Beyond these issues, the IdM systems developed for FI need to be interoperable, since heterogeneous technologies are expected to work together. The network infrastructure needs to achieve these requirements in order to not compromise services. Different IdM systems will have different trust requirements, since there are costs associated with establishing trust.

## References

Akinyele, J. A., Lehmann, C. U., Green, M. D., Pagano, M. W., Peterson, Z. N. J., and Rubin, A. D. (2010). Self-protecting electronic medical records using attribute-based encryption. *IACR Cryptology ePrint Archive*, 2010.

Alpár, G., Hoepman, J., and Siljee, J. (2011). The identity crisis security, privacy and usability issues in identity management. *Identity Management on Mobile Devices*, abs/1101.0427.

 ©2012 SBC — Soc. Bras. de Computação

Benaloh, J., Chase, M., Horvitz, E., and Lauter, K. (2009). Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 103–114, New York, NY, USA. ACM.

Bosworth, K., Lee, M. G. G., Jaweed, S., and Wright, T. (2005). Entities, identities, identifiers and credentials - what does it all mean? *BT Technology Journal*, 23(4).

Cao, Y. and Yang, L. (2010). A survey of identity management technology. In *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, pages 287–293.

Chim, T., Yiu, S., Hui, L., and Li, V. (2011). SPECS: secure and privacy enhancing communications schemes for VANETs. *Ad Hoc Network*, 9:189 – 203.

Dhamija, R. and Dusseault, L. (2008). The seven flaws of dentity management: usability and security challenges. *IEEE Security and Privacy*, 6:24–29.

Gomez-Skarmeta, A. F., Martinez-Julia, P., Girao, J., and Sarma, A. (2010). Identity based architecture for secure communication in Future Internet. In *The 6th ACM Workshop on Digital Identity Management*, pages 45–48, Chicago, Illinois, USA.

ITU-T (2009). NGN Identity Management framework. Recommendation ITU-T Y.2720. December.

Levin, A., Sutherland, E., and Choe, Y. (2009). The Future Internet. Technical report, International Communication Union - ITU.

Paul, S., Pan, J., and Jain, R. (2011). Architectures for the future networks and the next generation internet: a survey. *Computer Communications*, 34:2–42.

Sabena, F., Dehghantanha, A., and Seddon, A. P. (2010). A review of vulnerabilities in identity management using biometrics. In *International Conference on Future Networks*, pages 42–49.

Salsano, S., Polidoro, A., Mingardi, C., Niccolini, S., and Veltri, L. (2008). Sip-based mobility management in next generation networks. *IEEE Wireless Communications*, 15(2):92–99.

Sarma, A. and Girao, J. (2009). Identities in the Future Internet of Things. *Wireless Personal Communication*, 49:353–363.

Van Rooy, D. and Bus, J. (2010). Trust and privacy in the future internet, a research perspective. *Identity in the Information Society*, 3:397–404. 10.1007/s12394-010-0058-7.

Wan, M., Liu, Y., Zhou, H., and Zhang, H. (2010). A chord-based handoff authentication scheme under id/locator separation architecture. In *Advanced Intelligence and Awarenss Internet (AIAI 2010), 2010 International Conference on*, pages 309–314.

Weber, S., Martucci, L., Ries, S., and Mühlhäuser, M. (2010). Towards trustworthy identity and access management for the Future Internet. In *Trustworthy IoPTS: The 4th International Workshop on Trustworthy Internet of People Things and Services*, Tokyo, Japan.

 ©2012 SBC — Soc. Bras. de Computação

# IPSFlow – Uma Proposta de IPS Distribuído para Captura e Bloqueio Seletivo de Tráfego Malicioso em Redes Definidas por Software

**Fábio Yu Nagahama[1], Fernando Farias[1], Elisângela Aguiar[1], Luciano Gaspary[2], Lisandro Granville[2], Eduardo Cerqueira[1], Antônio Abelém[1]**

[1]Instituto de Tecnologia - Universidade Federal do Pará (UFPA)
[2]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
{nagahama, fernnf, eaguiar, cerqueira, abelem}@ufpa.br,
{paschoal, granville}@inf.ufrgs.br

***Abstract.*** *Traditional Intrusion Prevention Systems (IPS) have limitations in their operations. When running in active mode, IPSes do not have a wide coverage on the network and when capturing mirrored traffic they only block malicious one when working together with network devices from the same vendor or same solution. In this scenario, we introduce in this paper the IPSFlow, an IPS framework for Software Defined Networks (SDN) that, through the OpenFlow protocol, allows the creation of an IPS with wide coverage on the network, allowing the selective capture and automated blocking of malicious traffic near its source by combining the results of different techniques of traffic analysis.*

***Resumo.*** *Os tradicionais sistemas de prevenção de intrusão (Intrusion Prevention Systems – IPS) possuem limitações em sua atuação. Quando operam no modo ativo, não possuem uma ampla cobertura na rede, e quando capturam tráfego espelhado, só bloqueiam o tráfego malicioso se atuarem em conjunto com equipamentos de rede do mesmo fabricante ou solução. Neste contexto, propomos neste artigo o IPSFlow, um framework de IPS para Redes Definidas por Software (Software Defined Networks - SDN) que, através do protocolo Openflow, possibilita a criação de um IPS com ampla cobertura na rede, permitindo a captura seletiva e o bloqueio automatizado de tráfego malicioso o mais próximo de sua origem, através da combinação dos resultados de diferentes técnicas de análise de tráfego.*

## 1. Introdução

A cada dia, novos ataques ou mesmo variações de ataques conhecidos surgem e são lançados na rede em busca de vulnerabilidades. De acordo com o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br), só em 2011, foram quase 400.000 incidentes reportados [CERT 2012]. Desta forma, ferramentas que permitam a identificação e o tratamento automatizado de incidentes de segurança na rede tornam-se cada vez mais importantes na atividade realizada por um administrador.

Sistemas de Detecção de Intrusão de Rede (*Network Intrusion Detection Systems* – NIDSs) são ferramentas que capturam tráfego de rede, analisam e notificam o administrador quando da detecção de um possível tráfego malicioso. Já um Sistema de Prevenção de Intrusão (*Intrusion Prevention System* – IPS) estende a funcionalidade de

 ©2012 SBC — Soc. Bras. de Computação

um IDS e, além de notificar, tem a capacidade de bloquear automaticamente o tráfego malicioso [Mukhopadhyay 2011].

Os IPSs e IDSs convencionais possuem limitações em sua atuação e necessitam de alguns cuidados em suas instalações. Dependendo do local de instalação, eles podem ter uma cobertura restrita, passando a falsa sensação de proteção. Aspectos de desempenho também devem ser considerados no dimensionamento do *hardware* utilizado nestas ferramentas para que não impactem no funcionamento da rede, permitindo, desta forma, uma análise mais apurada do tráfego capturado [Snyder 2008].

No geral, um IDS ou IPS utiliza apenas uma técnica de classificação e análise de tráfego, o que pode acarretar em uma atuação ineficiente ao considerar as taxas de detecção e falsos alarmes. A combinação de diferentes classificadores para a construção de um IDS híbrido vem sendo estudada, apresentando bons resultados [Panda, 2012].

Em uma Rede Definida por *Software* (*Sofware Defined Network* – SDN), o plano de controle é separado do plano de dados e é delegado a um elemento externo, permitindo que pesquisadores, administradores e usuários programem o comportamento da rede. Nesta arquitetura, a definição do funcionamento interno dos equipamentos de rede como *switches* e roteadores continua sob a responsabilidade de seus fabricantes.

O OpenFlow, proposto por McKeown *et al*. [McKeown 2008], é um *framework* aberto nos moldes das SDNs, que através do protocolo OpenFlow disponibiliza interfaces de programação para construção de controladores de rede. Em uma rede OpenFlow, o controlador tem uma visão completa da rede e, assim, tem a capacidade de manipular os fluxos de dados logo que estes são recebidos em qualquer *switch* que implemente as funcionalidades contidas na especificação [Openflow 2011]. O uso do OpenFlow, na área de segurança e monitoramento de redes, já foi abordado em pesquisas anteriores, como, por exemplo, nas pesquisas de Ballard *et al.* [Ballard 2010], Braga *et al.* [Braga 2010] e Mehdi *et al.* [Mehdi 2011].

Em Ballard *et al.* [Ballard 2010], o OpenSAFE foi proposto como uma solução para redirecionamento de tráfego com propósitos de monitoramento em conjunto com uma linguagem de alto nível que especifica fluxos, chamada de ALARMS (*A Language for Arbitrary Redirection for Measuring and Security*). No trabalho de Braga *et al.* [Braga 2010], o OpenFlow foi usado em conjunto a uma rede neural artificial do tipo SOM (*Self Organizing Maps*) para a detecção de ataques DDoS (*Distributed Denial of Service*), obtendo bons resultados. Mehdi *et al.* [Mehdi 2011] propuseram e avaliaram o uso do OpenFlow na captura de tráfegos anômalos nas redes de usuários domésticos ou de pequenas empresas conectadas a um provedor de serviço de Internet.

Apesar da existência dos trabalhos mencionados, ainda não se tem notícias de um ambiente que permita a captura seletiva do tráfego e o uso do resultado do IDS para bloqueio automatizado do tráfego malicioso, tal como é proposto neste artigo. Desta forma, devido à flexibilidade permitida pelas SDNs, apresentamos neste artigo o IPSFlow, um *framework* que utiliza o OpenFlow para a construção de um IPS com captura seletiva e distribuída de tráfego nos *switches* para análise em um ou mais IDSs. No IPSFlow, de acordo com o resultado da análise, o controlador OpenFlow poderá bloquear o fluxo de forma automática no *switch* mais próximo da origem do tráfego.

O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta algumas dificuldades envolvidas no uso de um IPS na rede. Na seção 3, a arquitetura

 ©2012 SBC — Soc. Bras. de Computação

das SDNs e o OpenFlow são abordados. A proposta da solução IPSFlow é explicada na seção 4. Por fim, as considerações finais e trabalhos futuros são discutidos na seção 5.

## 2. Desafios em Sistemas de Detecção e Prevenção de Intrusão na Rede

De acordo com o local de instalação de um sensor, este pode atuar como IDS ou IPS e possuir coberturas distintas. A Figura 1 representa uma visão simplificada de uma rede típica composta por um *switch* de núcleo interligando as demais redes através de *switches* de distribuição ou acesso. As posições identificadas com as letras A, B e C indicam alguns locais onde os sensores IPS ou IDS podem ser instalados. As linhas tracejadas identificadas com os números 1, 2 e 3 exemplificam alguns tipos de tráfego que uma estação interna à rede pode iniciar.

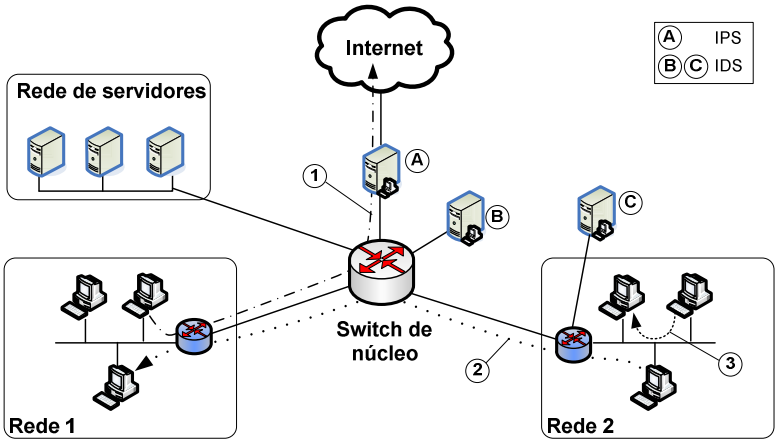



**Figura 1. Visão geral de uma rede com IPS e IDSs.**

Na posição A, o IPS atua no modo ativo (*inline)* e só consegue capturar e analisar o tráfego passante (*i.e.*, fluxo 1), não tendo ações sobre os demais tráfegos confinados dentro da rede (*i.e.*, fluxos 2 e 3). Para capturar os fluxos do tipo 2 e 3, o IPS pode ser conectado ao *switch* central (posição B) ou nos *switches* de distribuição (posição C) e receber uma cópia do tráfego através do espelhamento de portas (modo passivo). Porém, uma vez que apenas uma cópia do tráfego é recebida, o bloqueio de tráfego malicioso torna-se inviável e sua atuação fica restrita a apenas gerar alertas de forma semelhante a um IDS. Para que um tráfego seja bloqueado nas posições B e C, o IPS necessitaria atuar em conjunto com um equipamento capaz de bloquear o tráfego malicioso e esta é uma situação rara de acontecer devido à variedade de fabricantes e modelos de equipamentos utilizados nas redes.

Em quaisquer das posições (A, B ou C), não é comum que o tipo de tráfego a ser encaminhado aos IDSs e IPSs seja filtrado previamente. Assim, o *hardware* destas ferramentas deve ser devidamente dimensionado para que as etapas de captura, análise e eventual reencaminhamento do tráfego não adicionem retardos excessivos aos pacotes (no caso do IPS) ou para que todos os pacotes copiados sejam recebidos e analisados adequadamente para uma avaliação mais apurada (no caso do IDS).

Balancear a carga adicionando mais IPSs e/ou IDSs pode evitar um eventual superdimensionamento do *hardware*. Porém, a adição de IPSs *inline* pode acarretar na adição de mais retardo nos pacotes. Para o caso de IDSs passivos, a limitação passa a ser dos *switches*, visto que nem todos são capazes de realizar o espelhamento de um

 ©2012 SBC — Soc. Bras. de Computação

número elevado de portas [Ballard 2010]. Além disso, pelos mesmos motivos expostos, o uso de diferentes sensores com diversas técnicas de análise torna-se inviável.

A atuação limitada dos IDSs leva a crer que a distribuição de IPSs ao longo da rede tende a ser uma solução inevitável. Porém, a sincronização e a administração de diversos sensores tornam-se tarefas onerosas e inviáveis para o administrador da rede. As soluções de IPSs *inline* distribuídos de grandes fabricantes facilitam a administração, mas implicam na dependência de seus clientes com os fornecedores. Nesses casos, a adição de módulos ou mesmo a atualização do ambiente tendem a ser restritos apenas ao fornecedor, inviabilizando qualquer tipo de personalização por parte do administrador da rede. Mesmo em soluções com *softwares* livres, questões como desempenho, administração onerosa e sincronização de informação tornam-se um problema.

## 3. Redes Definidas por *Software*

Os equipamentos de redes como *switches* e roteadores convencionais implementam os planos de controle e de dados acoplados dentro do mesmo *hardware*. Desta forma, qualquer funcionalidade nova só pode ser definida pelo fabricante do equipamento, tornando-se uma arquitetura engessada do ponto de vista de novas funcionalidades. Já na arquitetura das Redes Definidas por *Software*, os planos de controle e de dados são separados, e com isso a definição do funcionamento da rede fica a cargo do administrador, enquanto que a implementação do mecanismo de comutação dos dados permanece sob o sigilo dos fabricantes. Portanto, a inteligência e o estado da rede são logicamente centralizados e, como resultado, as redes têm potencial para se tornarem mais flexíveis e facilmente adaptáveis [OpenFlow 2011].

Com as SDNs, o administrador passa a ter o controle da rede através de um ponto lógico central independentemente do fabricante do equipamento, o que simplifica bastante sua operação. Através da centralização do plano de controle, as SDNs permitem que a rede seja dinamicamente configurada por aplicações desenvolvidas pelos próprios administradores ou usuários da rede.

O OpenFlow é o primeiro padrão de interface de comunicação que especifica um protocolo com o propósito de interconectar os planos de controle e de dados definidos pela arquitetura de SDN, e vem recebendo o reconhecimento e apoio de grandes fabricantes e empresas de Tecnologia da Informação. Ele trabalha com o conceito de fluxos para definir tráfegos de rede baseado em regras predefinidas, que podem ser configuradas de forma estática ou dinâmica por um controlador SDN.

O *framework* Openflow é composto basicamente por 5 componentes: o controlador, que define o plano de controle; o *switch* (também conhecido como *datapath*), que continua sendo o responsável pelo encaminhamento dos pacotes no plano de dados; o canal seguro para a comunicação entre o controlador e o *datapath*; o protocolo OpenFlow, que define como a comunicação entre o controlador e o *datapath* é realizada; e a tabela de fluxos, que é o local onde as ações para os fluxos são definidas dentro do *datapath*.

O funcionamento básico do OpenFlow consiste em, ao receber o primeiro pacote do fluxo no primeiro *datapath* da rede, verificar se há algum tratamento definido na tabela de fluxos. Havendo uma entrada na tabela de fluxos, a ação definida é executada. Caso contrário, os dados do fluxo são extraídos, encapsulados e encaminhados ao controlador para tomada de decisão. O controlador consulta a aplicação utilizada na

 ©2012 SBC — Soc. Bras. de Computação

rede e decide como o tráfego deve ser tratado. Uma resposta é enviada ao *datapath* que atualiza a tabela de fluxos com as instruções recebidas e os pacotes subsequentes do mesmo fluxo receberão o mesmo tratamento, até que a entrada na tabela de fluxos seja removida [Openflow 2011].

Diante do exposto, a arquitetura de SDNs torna-se uma grande aliada na administração de IPSs, pois combina a rápida comutação de pacotes no plano de dados com a flexibilidade e a visão única do plano de controle [Lantz 2010]. Desta forma, o uso da arquitetura SDN favorece o aumento da área de abrangência do IPS, permitindo este atuar de uma maneira mais proativa na origem do problema.

## 4. IPSFlow

Diante do apresentado, este artigo propõe um *framework* de IPS denominado de IPSFlow, que utiliza uma rede baseada no modelo de SDN em conjunto com o protocolo OpenFlow e possibilita a criação de um IPS com ampla cobertura na rede e a capacidade de capturar seletivamente um fluxo para posterior análise em um ou mais IDSs. Além disso, o IPSFlow prevê mecanismos para que os resultados de diferentes IDSs sejam combinados para o bloqueio automático de um tráfego malicioso, o mais próximo de sua origem. O IPSFlow possui dois elementos fundamentais: uma aplicação denominada de IPSFlowApp, que define como a SDN baseada no OpenFlow deve funcionar, e um ou mais IDSs para análise do tráfego.

### 4.1. Arquitetura e funcionamento do IPSFlow

A Figura 2(i) apresenta uma visão geral da composição do IPSFlow em uma SDN. Quando o primeiro pacote de um fluxo é recebido no primeiro *datapath* (passo 1), o processo de consulta à tabela de fluxos e o controlador seguem conforme especificado no OpenFlow [Openflow 2011] (passo 2). Ao receber uma consulta de um *datapath*, o controlador verifica se no IPSFlowApp há uma regra definida para captura e análise do tráfego recebido (passo 3).

Caso o fluxo esteja definido para ser analisado, este pode ser tratado de duas formas no *datapath.* Na primeira, o fluxo pode ser encaminhado para o destinatário (passo 4) e uma cópia enviada para análise em um conjunto de IDSs (passo 5). Ao se concluir que o fluxo se trata de um tráfego malicioso, os IDSs enviam o resultado ao controlador (passo 6) para que o tráfego passe a ser bloqueado no *datapath*. Esta decisão pode ser armazenada no controlador para ser reutilizada em futuras consultas.

Na segunda forma, o fluxo pode ser retido no *datapath* e uma cópia enviada para análise em um conjunto de IDSs. Neste momento, um contador decrescente definido pelo administrador é inicializado. Caso a análise conclua que o tráfego seja malicioso antes do contador chegar a zero, o resultado é enviado ao controlador para instruir o *datapath* a descartar os pacotes retidos. Caso o contador chegue a zero, o tráfego é liberado para o seu destino. Se após a liberação do tráfego, for constatado que este se tratava de um tráfego malicioso, a tabela de fluxo é alimentada para descartar pacotes subsequentes daquele fluxo.

### 4.2. A aplicação IPSFlowApp

Os elementos que compõem o IPSFlowApp são apresentados na Figura 2(ii): módulo de configuração (a), base de regras (b), agente (c) e módulo de alertas (d). O módulo de

 ©2012 SBC — Soc. Bras. de Computação

configuração é o ponto de comunicação entre a aplicação e o administrador e é neste módulo que ocorre toda a definição de quais ações devem ser tomadas nos fluxos. Estas decisões são armazenadas na base de regras para serem consultadas pelo agente. Ele também é responsável por receber e tratar as notificações de alertas, podendo reconfigurar automaticamente a base de regras de acordo com as definições do administrador e acionar o agente para realizar o descarte do tráfego malicioso.

A base de regras é o local onde todas as ações a serem tomadas com os fluxos são armazenadas, por exemplo, bloqueio de fluxo, encaminhamento de fluxo por uma porta com cópia para análise ou retenção com cópia para análise (como apresentadas anteriormente).

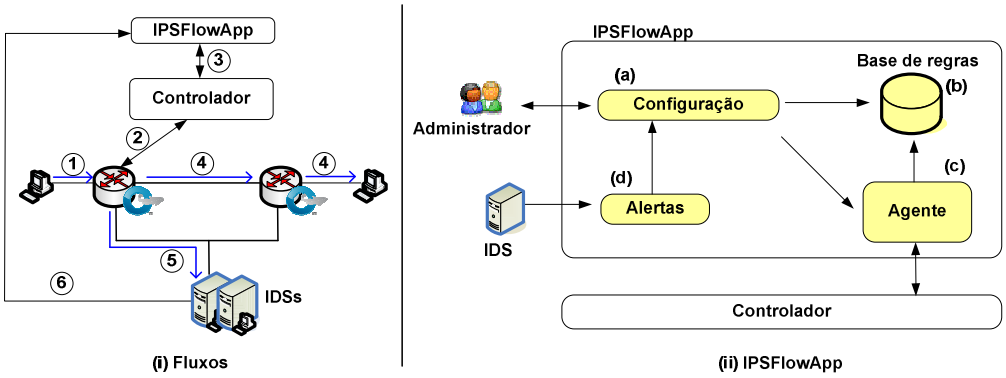



**Figura 2. *Framework* IPSFlow.**

O agente é o elemento responsável pela comunicação entre o controlador e a aplicação. É ele que recebe a consulta do controlador, extrai as informações necessárias e pesquisa na base de regras à procura de ações que devam ser tomadas para o fluxo recebido. Havendo uma regra definida, a ação é encaminhada ao controlador para as devidas configurações nos *datapaths*. Caso não haja regra alguma, uma regra padrão de encaminhamento adiante pode ser dada como resposta. Ele também é responsável por instruir o controlador a descartar fluxos detectados como maliciosos.

O módulo de alertas é o responsável por receber, tratar o resultado da análise dos IDSs e notificar o módulo de configuração. Para permitir a utilização de diversos IDSs, o módulo de alerta define um formato padrão esperado para receber os resultados de forma semelhante ao documento experimental 4765 do IETF [Debar, 2007].

É válido ressaltar que a utilização de SDNs para propósitos de segurança mostrou-se bastante promissora em trabalhos anteriores [Ballard 2010], [Braga 2010] e [Mehdi 2011]. Porém, estes trabalhos se limitaram a apenas detectar e alertar a ocorrência de tráfegos maliciosos na rede. A proposta do IPSFlow é estender estes trabalhos, uma vez que passa a ter uma visão centralizada e completa da rede e assim permitir a captura do tráfego em qualquer local da rede, encaminhar o tráfego para análise em um ou mais IDSs e utilizar o resultado das análises para a reconfiguração automática das tabelas de fluxos dos *datapaths*.

## 5. Considerações finais e trabalhos futuros

Com o crescimento das redes, a diversificação dos dispositivos de rede utilizados e o aumento de usuários conectados em rede, o aumento de incidentes de segurança torna-

 ©2012 SBC — Soc. Bras. de Computação

se inevitável. Desta forma, cada vez mais ferramentas como IDSs e IPSs se fazem necessários na administração da segurança de uma rede.

O IPSFlow proposto neste artigo utiliza o protocolo OpenFlow e viabiliza uma administração simplificada e centralizada, com solução aberta e flexível permitindo a análise do tráfego por um ou IDSs. Além disso, uma vez que o IPSFlow prevê a captura seletiva, esta não exige o uso de *hardwares* superdimensionados nos IDSs, possibilitando inclusive o uso de diversos tipos de IDSs com especialidades distintas. Com atuação em todos os *switches* da rede, o IPSFlow potencialmente apresenta uma cobertura mais ampla e com a captura de tráfego sendo feita sem a necessidade de espelhamento de portas, permite o bloqueio do tráfego malicioso o mais próximo de sua origem.

Para trabalhos futuros, tem-se o desenvolvimento da aplicação IPSFlowApp e a validação da proposta através da criação de um *testbed* utilizando tráfego capturado em uma rede real. Testes de desempenho da solução variando e/ou combinando diversos tipos de IDSs também deverão ser executados para coleta e comparação de resultados.

## Referências


Ballard, J., Rae, I., Akella, A. **Extensible and Scalable Network Monitoring Using OpenSAFE**. Internet Network Management Workshop / Workshop on Research on Entreprise Networking. San Jose, CA. 27 Abr. 2010.

Braga, R. S., Mota, E., Passito, A. **Lightweight ddos floding attack detection using nox/openfow**. IEEE Conference on Local Computer Networks (2010), IEEE.

CERT, **Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil**. Disponível em: <http://www.cert.br>. Acesso em 25 out. 2011.

Debar, H., Curry, D., Feinstein, B. **The Intrusion Detection Message Exchange Format**. 2007. Disponível em: <http://tools.ietf.org/html/rfc4765>. Acesso em 20 mai. 2012.

Lantz, B., Heller, B., and McKeown, N. (2010). **A network in a laptop: rapid prototyping for software-defined networks**. ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets '10, pages 19:1–19:6, New York, NY, USA. ACM

McKeown, N., Anderson, et al. **OpenFlow: Enabling Innovation in Campus Networks**. Computer Communication Review. 2008. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.2269&rep=rep1&type=pdf >. Acesso em 20 ago. 2011.

Mehdi, S. Khalid, J., Khayam, S. **Revisiting Traffic Anomaly Detection using Software Defined Networking.** Recent Advances in Intrusion Detection (RAID), 2011. Disponível em: < http://www.wisnet.seecs.nust.edu.pk/publications/2011/raid2011_paper.pdf>. Acesso em 21 jan. 2012.

Mukhopadhyay, I., Chakraborty, M., Chakrabarti, S. **A Comparative Study of Related Technologies of Intrusion Detection & Prevention Systems**. Journal of Information Security, 2011, v. 2, p. 28-38.

OpenFlow. **The OpenFlow Switch Specification**. Disponível em: <http://OpenFlowSwitch.org>. Fevereiro 2011. Acesso em 27 set. de 2011.

Panda, M., Abraham, A., Patra, M. R. **A Hybrid Intelligent Approach for Network Intrusion Detection**. Procedia Engineering, 2012, v. 30, p. 1-9.

Snyder, J. **Guide to Network Intrusion Prevention Systems.** Disponível em: <http://www.pcworld.com/businesscenter/article/144634/guide_to_network_intrusion_prevention_systems.html>. Outubro 2008. Acesso em 20 fev. 2012.


 ©2012 SBC — Soc. Bras. de Computação

# Uma Arquitetura para Mitigar Ataques DDoS em Serviços Web sob Nuvem

**Fernando Gielow, Fernando Bernardelli, Cinara Menegazzo, Nadine Pari, Aldri Santos**

[1] Departamento de Informática – Universidade Federal do Paraná
NR2 - Núcleo de Redes Sem Fio e Redes Avançadas – Curitiba – Brasil

{fhgielow,fcbernardelli,cmenegazzo,nelpari,aldri}@inf.ufpr.br

***Abstract.*** *Distributed Denial of Service (DDoS) attacks are often neglected because they cause just a temporary interruption of the system normal functioning. With the advent of paradigms like Cloud Computing, mitigating DDoS attacks giving more resources to the applications has become a feasible alternative, but entails the Economic DDoS problem. This paper presents an architecture to mitigate DDoS attacks against a Cloud hosted application. Such architecture is based on the idea of instantiating a replica of the application - simple operation for a Cloud - and redirecting only authentic queries to the new replica. The proposed architecture does not need to identify the attackers and, even so, it filters only authentic traffic, without extra overhead and potential categorization errors that could arise when trying to identify the clients.*

***Resumo.*** *Ataques de Distributed Denial of Service (DDoS) frequentemente são negligenciados por representarem apenas uma interrupção temporária no funcionamento normal de um sistema. Com o advento de paradigmas como a cloud, a mitigação deste tipo de ameaça com o acréscimo de recursos para as aplicações se torna viável, mas acarreta em um problema denominado economic DDoS. Este artigo apresenta uma proposta de arquitetura para a mitigação de ataques DDoS direcionados a uma aplicação hospedada em uma cloud. Tal arquitetura é baseada na instanciação de uma réplica da aplicação - operação simples em uma cloud - e no redirecionamento apenas de requisições legítimas a esta réplica. A arquitetura proposta não precisa identificar os clientes atacantes e, ainda assim, consegue filtrar apenas o tráfego legítimo sem a carga e possíveis erros decorrentes da necessidade de identificação.*

## 1. Introdução

Diversas pesquisas têm sido desenvolvidas para tratar questões da Internet atual, que podem se propagar para a Internet do Futuro (IF). Tais problemas podem ser amplamente categorizados nas áreas de mobilidade, qualidade de serviço e segurança, os quais ainda caminham para soluções aceitáveis, agravados pelo surgimento de novas arquiteturas. Hoje tanto os dados quanto as aplicações são oferecidos em localizações físicas distintas e desconhecidas. Outra grande mudança ocorreu na forma de administrar um sistema, que antes era de âmbito mais local, com seus usuários e servidores característicos, e agora esses sistemas são hospedados em ambientes construídos pelo compartilhamento de recursos de diversos sistemas autônomos (AS) e heterogêneos [Pianese et al. 2010].

Apesar de muitos esforços em pesquisas, os ataques de *Denial of Service* (DoS) ainda representam sérias ameaças a muitos servidores na Internet e se configuram como

 ©2012 SBC — Soc. Bras. de Computação

um dos principais desafios de segurança atualmente propagado para a IF, que interconectará muito mais dispositivos e indivíduos. Um ataque DoS não visa invadir um computador para obter informações confidenciais, nem tão pouco alterar informações armazenadas nele. Seu objetivo é a indisponibilização de um serviço fornecido, utilizando-se do encaminhamento de grandes quantidades de tráfego ao hospedeiro do serviço. Essa questão torna-se ainda mais severa quando diversos geradores de tráfego intensificam o encaminhamento de tráfego de maneira distribuída, caracterizando um ataque de *Distributed Denial of Service* (DDoS) [Sachdeva et al. 2008]. Embora tal carga seja um problema apenas momentâneo, em se tratando de aplicações destinadas ao comércio eletrônico, por exemplo, uma parada do serviço representa grandes perdas financeiras.

Com as novas arquiteturas de rede e de aplicações que configuram a Internet, têm surgido sistemas complexos e robustos como *clouds* (nuvens), onde o desafio de mitigar ataques DoS torna-se ainda mais necessário. Embora a maioria das soluções comumente oferecidas para mitigar DDoS em *cloud* se baseie na maior alocação de recursos [Peng et al. 2007], essas abordagens tornam-se inadequadas pois a premissa da possibilidade de maior alocação de recursos nem sempre é viável por ser custosa demais [Bakshi and Yogesh1 2010], [Liu 2010]. Este comportamento carateriza o *economic DDoS* (eDDoS) [Khor and Nakao 2009].

Este trabalho propõe uma arquitetura reativa e tolerante a falhas para a mitigação de ataques de DDoS executados contra aplicações hospedadas em uma *cloud*. Tal arquitetura é baseada na instanciação de uma réplica da aplicação e no redirecionamento apenas de requisições legítimas a esta réplica. A arquitetura monitora o tráfego de uma aplicação e ao detectar uma possível anomalia, isto é, a ocorrência de um ataque de DDoS, ela estabelece uma nova instância desta aplicação, garantindo que nenhum tráfego malicioso a alcance. As diferenças desta solução para outras propostas são que a aplicação hospedada não precisa prover acurácia na filtragem de tráfego legítimo, o uso dos recursos não é onerado financeiramente, e a intervenção humana é desnecessária. Uma avaliação experimental considerando o tempo de resposta aos clientes mostra a eficácia de uma implementação da arquitetura diante de ataques DDoS a um serviço Web.

O restante do artigo está organizado da seguinte maneira: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha a arquitetura proposta para a mitigação de ataques *DDoS*. A Seção 4 apresenta uma descrição da implementação realizada da arquitetura. A Seção 5 apresenta uma avaliação, juntamente com o cenário e os resultados. Por fim, a conclusão e trabalhos futuros são apresentados na Seção 6.

## 2. Trabalhos Relacionados

As pesquisas que envolvem propostas de mitigação de DDoS em arquiteturas de *cloud*, ainda são consideradas incipientes e distantes de uma convergência. Dentre as poucas propostas para estes ambientes, destaca-se o *framework* pró-ativo CluB, apresentado em [Hazelhurst 2008], que sugere que sejam selecionados determinados roteadores, dispostos de forma distribuída, para a análise de tráfego e consequente prevenção de atividade maliciosa. Neste *framework*, todo pacote deve ser verificado para entrar, sair ou transitar na arquitetura. Cada roteador alocado realiza a verificação, o que é custoso devido ao *overhead* causado pela autenticação de cada pacote e pela necessidade inviável de alterar o comportamento dos roteadores.

 ©2012 SBC — Soc. Bras. de Computação

Em [Verkaik et al. 2006], é apresentado um esquema pró-ativo que emprega Comunidades de Interesse (COIs) para capturar dados sobre o comportamento coletivo das entidades remotas, utilizando-os para predizer o comportamento futuro. Tal esquema assume que os clientes que tiveram relações legítimas anteriormente possuem bons indícios e podem ser considerados novamente legítimos. Entretanto, a identificação dos clientes antigos não é tão trivial. Além do *overhead* gerado pela verificação, os endereços IPs são normalmente dinâmicos e a exigência da realização de *login* para a identificação não é possível, dado que o ataque de DDoS pode impossibilitar uma operação de identificação. Em [Bakshi and Yogesh1 2010], os ataques são tratados através da criação de uma nova instância da aplicação. Uma vez que um ataque DDoS é detectado, o mecanismo proposto busca identificar os atacantes através de PINGs: caso um cliente suspeito de ser atacante não responda ao PING, ele é considerado como um atacante, não sendo redirecionado para a nova instância da aplicação. Entretanto, essa solução assume que sempre e apenas clientes genuínos responderão a PINGs, o que as vezes não condiz com a realidade.

A eficácia desses esquemas de mitigação depende diretamente da capacidade de identificação ou filtragem dos clientes legítimos. A solução WebSoS [Stavrou et al. 2005] oferece uma filtragem robusta de tráfego atacante e bloqueio de requisições não aprovadas, formando assim um *overlay* seguro que mitiga DDoS em servidores web. O servidor utiliza mecanismos de autenticação criptográfica e um teste gráfico de Turing [Dietrich et al. 2000] para diferenciar clientes humanos de *scripts* de ataque. Estes procedimentos, segundo os testes dos autores, não sobrecarregam o funcionamento do serviço, porém exigem que os roteadores localizados no perímetro do servidor sejam reconfigurados, o que é inviável para arquiteturas de *cloud*.

## 3. Arquitetura para Mitigação de DDoS em *Cloud*

Esta Seção descreve uma arquitetura para mitigar ataques de DDoS em *clouds* de forma autônoma e independente. A arquitetura proposta pode ser utilizada por qualquer aplicação web hospedada em uma *cloud* que, ao sofrer indícios de um ataque DDoS, filtra o tráfego legítimo e encaminha apenas este para uma nova instância da mesma aplicação.

Esta arquitetura, ilustrada na Figura 1, é composta por um módulo geral chamado de Gerenciador de Tráfego (GT), que não se comunica diretamente com a aplicação. Esse módulo possui os submódulos INA, GB, AT e RT. Além disso, a instância do banco de dados (BD) é exterior às demais instâncias da *cloud*, visto que o banco de dados também está nas nuvens e pode ser acessado de qualquer outra instância *cloud*.

O submódulo AT observa o comportamento do tráfego de entrada para a aplicação de forma pró-ativa. Ele foca na estimativa da quantidade de tráfego e de processamento no servidor, e realiza medição para detectar a existência de um possível ataque DDoS. Caso um ataque seja detectado, o submódulo INA é ativado. O INA criará uma nova instância da aplicação em outro servidor na *cloud*, consequentemente com um endereço IP diferente. O submódulo RT trata todo o tráfego de entrada, respondendo com um redirecionamento para a nova instância da aplicação, como visto na Figura 2. O RT parte do princípio que os atacantes DDoS não interpretam as respostas obtidas do servidor, pois se eles interpretarem, sua eficiência é reduzida. Desta maneira, apenas os clientes legítimos (Nós *C* na figura) serão, de fato, redirecionados à nova aplicação, enquanto os nós atacantes não passarão pelo RT.

 ©2012 SBC — Soc. Bras. de Computação

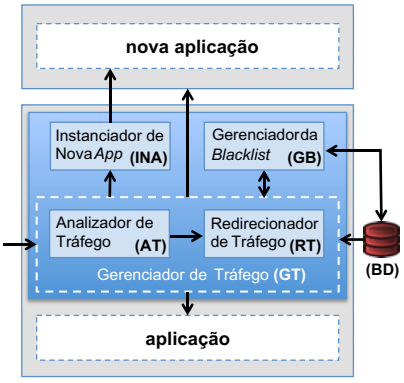



**Figura 1. Arquitetura de mitigação de DDoS**

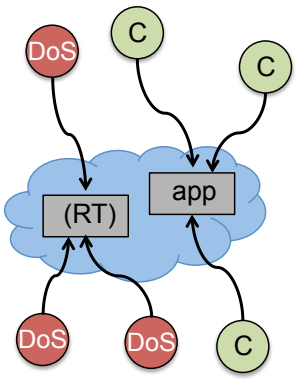



**Figura 2. Fluxo de tráfego**

Ao tentar redirecionar os clientes para a nova instância, o endereço do cliente, seja ele legítimo ou não, será adicionado em uma *blacklist*. Os clientes presentes nesta lista têm suas requisições descartadas, a fim de reduzir o custo de processamento de respostas no servidor. Entretanto, como o cliente legítimo é informado do redirecionamento antes de seu endereço entrar na *blacklist*, ele terá acesso à esta nova instância replicada e poderá enviar uma nova requisição. Registros com tempo de validade são empregadas nesta *blacklist*, dado que as respostas podem ser perdidas. O tempo de validade na lista aumenta exponencialmente, para diminuir ainda mais a sobrecarga. Cabe ao GB, o papel de adicionar e gerenciar a saída de endereços de clientes à *blacklist*, assim como o tempo de validade da entrada que aumenta exponencialmente.

## 4. Implementação

Para a implementação da arquitetura, a solução em *cloud* [Heroku 2012] foi utilizada. Ela oferece infra-estrutura como serviço de hospedagem, possibilitando o desenvolvimento em *Ruby on Rails*. A arquitetura do *framework* [RubyOnRails 2012] é completamente baseada no paradigma *Model View Controler* (MVC), facilitando a organização dos módulos de nossa arquitetura. Assim, a estrutura do código escrito em RoR é composta de componentes de Modelo, de Visão e de Controle. Os componentes de **modelo** correspondem aos dados - como eles são armazenados, obtidos, correlacionados. A parte de **visão** corresponde à parte gráfica da aplicação. Finalmente, os **controladores** realizam a manipulação de dados como um todo, e correspondem à parte lógica e funcional do código. Eles funcionam também como uma ponte entre modelo e visão, para que os dados transitem em ambos os sentidos.

Considerando o RoR, o submódulo analizador de tráfego (AT) da arquitetura corresponde a um controlador. Assim, uma requisição à aplicação será interceptada por esse componente de controle, que realizará a medição de estatísticas, e imediatamente acionará o controlador que corresponde ao funcionamento da aplicação em si. Deve-se notar, contudo, que o tempo despendido neste controlador é ínfimo. A Figura 3 apresenta um fluxograma da implementação realizada. Em outras implementações, caso se perceba que o tempo afeta o funcionamento do mecanismo de mitigação, este processamento poderia ainda ser realizado em segundo plano.

Quando o AT detectar a existência de um possível ataque, uma nova instância *cloud* é criada pelo submódulo INA e a aplicação é replicada para esta instância, parali-

 ©2012 SBC — Soc. Bras. de Computação

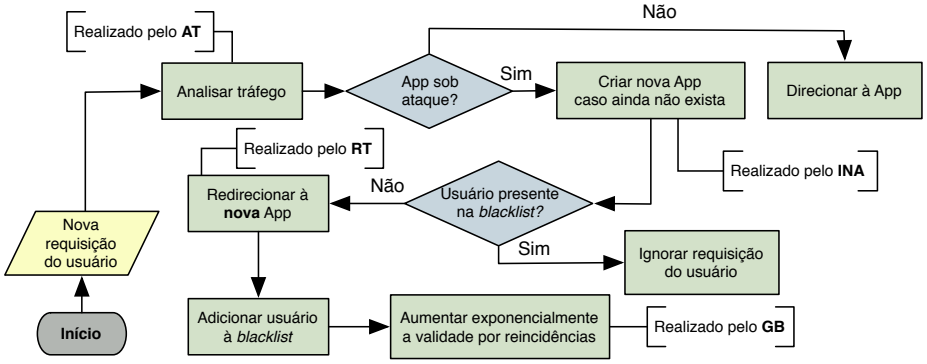



**Figura 3. Operações da implementação para a mitigação**

sando a aplicação original, que passa a responder apenas como redirecionador. O processo de reinstanciação da aplicação na implementação realizada consiste da existência prévia de uma segunda aplicação, inicialmente sem nenhum recurso alocado.

Uma particularidade interessante do *framework* RoR é a existência de um arquivo de rotas. A implementação do submódulo redirecionador de tráfego (RT) é realizada em cima deste arquivo, chamado *routes.rb*. Para a exibição de qualquer página dinâmica da aplicação, o arquivo de rotas é inevitavelmente chamado. Desta forma, ele é utilizado para a adição de clientes na *blacklist* e respectiva filtragem dos clientes bloqueados pelo gerenciador da *blacklist* (GB). No redirecionamento do tráfego para uma nova instância, uma entrada será adicionada, bloqueando o cliente em questão por determinado tempo.

A *blacklist* em si e as diversas outras variáveis de controle são gerenciadas pela base de dados em *cloud* [Redis 2012]. Esta base de dados é conhecida por sua simplicidade e eficiência. Ela basicamente mapeia **chave** e **dado**, oferecendo tempos de escrita e de leitura correspondentes à *hashing*. A implementação da *blacklist* foi feita utilizando o endereço IP de um cliente como chave, e o tempo que este cliente permanecerá bloqueado como dado. Por ser, indiretamente, um mecanismo de *hashing*, o tempo de busca por um cliente será O(1), o que é excelente para um mecanismo que filtrará todo tráfego que chega à aplicação.

## 5. Avaliação

A avaliação da arquitetura proposta consiste na análise da capacidade do servidor em atender novas requisições, sendo que o atendimento pode ser apenas o redirecionamento. Se o ataque de DDoS for devidamente mitigado, as requisições dos atacantes serão ignoradas, após a sua inclusão na *blacklist*. Logo, o servidor na *cloud* deverá ser capaz de redirecionar apenas clientes legítimos para a nova instância e garantir que eles terão acesso direto nas próximas requisições.

Para a experimentação, como nós atacantes, foram utilizadas oito máquinas do laboratório de pesquisa para processar os ataques, observando-se uma latência de rede no intervalo de $3ms$ a $7ms$. Cada uma destas máquinas operou com 25, 50, 75 ou 100 instâncias de um *script* atacante, que utiliza o comando *curl* para bombardear o servidor com requisições HTTP do tipo *GET*. Tais experimentos foram realizados diversas vezes, obtendo resultados de comportamento similar. Quanto à hospedagem, foi utilizado um

 ©2012 SBC — Soc. Bras. de Computação

*dyno* para cada aplicação. Para a *cloud* Heroku, um *dyno* é uma instância de servidor virtual isolado com 512MBs de RAM e 4 cores Intel Xeon X5550 @ 2.67GHz.

### 5.1. Resultados

Para explorar e validar a nossa proposta, foram realizados experimentos de acordo com as métricas especificadas em [Bellaiche and Gregoire 2008]. Primeiro, foi avaliado o impacto da mitigação de DDoS em tempo de resposta e, em seguida, em relação à taxa de resposta e *overhead*. Como observado na Figura 4, com um intervalo de confiança de 95%, o uso da arquitetura proposta, gráfico à esquerda, reduziu o tempo de resposta às requisições legítimas em comparação ao gráfico à direita, que mostra o tempo gasto para atender o mesmo número de requisições originadas sem o nosso mecanismo. Tal comportamento ocorre porque com o uso da arquitetura, a *blacklist* impedirá uma aplicação de responder ao mesmo cliente múltiplas vezes, garantindo ainda assim que o cliente legítimo seja capaz de atingir a nova instância.

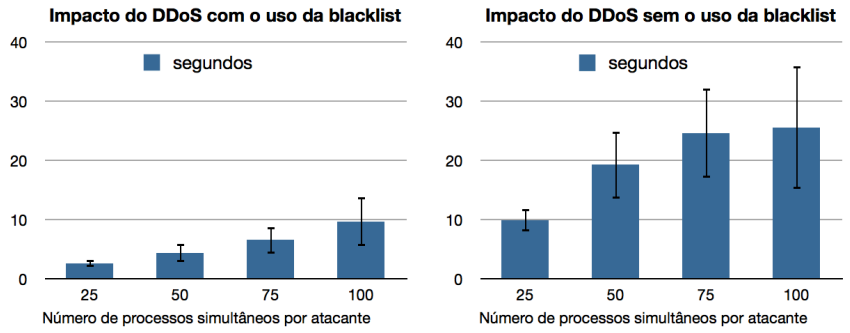



**Figura 4. Tempo de resposta para clientes legítimos**

Outra métrica avaliada é a taxa de páginas solicitadas recebidas com sucesso, mostrada na Figura 5. Há uma queda no número de respostas apenas quando não foram utilizadas a filtragem pela *blacklist* e o conseqüente redirecionamento. Nestes casos, a aplicação envia uma resposta ao atacante, que descarta esta resposta de imediato, dando continuidade ao ataque. Nota-se que a taxa de entrega sem o uso da *blacklist* é afetada pela ocorrência de *timeouts* de requisições HTTP não respondidas, pois a aplicação permanece ocupada com as requisições atacantes.

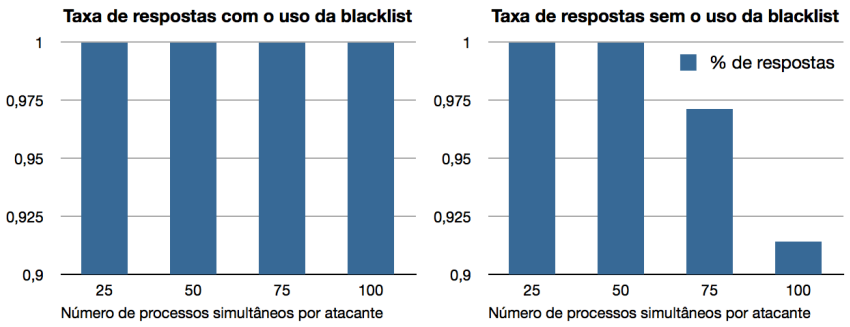



**Figura 5. Taxa de respostas do servidor a clientes legítimos**

 ©2012 SBC — Soc. Bras. de Computação

## 6. Conclusão

Este trabalho apresentou uma arquitetura de mitigação para ataques de DDoS direcionados à aplicações web hospedadas em *cloud*. A arquitetura é dependente apenas da existência do ambiente na *cloud* e permite livre acesso aos clientes legítimos. O uso de uma *blacklist* é eficiente para filtragem devido ao tempo de validade dos registros, que no caso de atacantes, terá aumento exponencial para reincidências. Os resultados alcançados nas experimentações demonstram a validade da solução proposta, pois conseguem direcionar o tráfego legítimo de modo satisfatório, impossibilitando os atacantes de acessarem a nova instância criada. Mecanismos mais robustos para a checagem da *blacklist* em níveis mais baixos e otimizados serão desenvolvidos como trabalhos futuros, complementando a solução atual. Além disso, experimentos em maiores escalas tanto no ataque DDoS como também nos recursos alocados às instâncias da aplicação serão realizados.

## Referências

Bakshi, A. and Yogesh1, B. (2010). Securing cloud from ddos attacks using intrusion detection system in virtual machine. In *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, pages 260 – 264.

Bellaiche, M. and Gregoire, J.-C. (2008). Measuring defense systems against flooding attacks. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*, pages 600 –605.

Dietrich, S., Goddard, N., and Long, N. (2000). Analyzing distributed denial of service tools: The shaft case. In *In Proceedings of USENIX LISA'2000*, pages 329–339.

Hazelhurst, S. (2008). Scientific computing using virtual high-performance computing: a case study using the amazon elastic computing cloud. In *Proceedings of the 2008 SAICSIT'08*, pages 94–103, New York. ACM.

Heroku (2012). http://www.heroku.com/.

Khor, S. H. and Nakao, A. (2009). spow: On-demand cloud-based eddos mitigation mechanism. *HotDep (Fifth Workshop on Hot Topics in System Dependability)*.

Liu, H. (2010). A new form of dos attack in a cloud and its avoidance mechanism. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, pages 65–76, New York.

Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Comput. Surv.*, 39.

Pianese, F., Bosch, P., Duminuco, A., Janssens, N., Stathopoulos, T., and Steiner, M. (2010). Toward a cloud operating system. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 335 –342.

Redis (2012). http://redis.io/.

RubyOnRails (2012). http://rubyonrails.org/.

Sachdeva, M., Singh, G., Kumar, K., and Singh, K. (2008). Ddos incidents and their impact: A review.

Stavrou, A., Cook, D. L., Morein, W. G., Keromytis, A. D., Misra, V., and Rubenstein, D. (2005). Websos: an overlay-based system for protecting web servers from denial of service attacks. *Computer Networks*, 48:781–807.

Verkaik, P., Spatscheck, O., Van der Merwe, J., and Snoeren, A. C. (2006). Primed: community-of-interest-based ddos mitigation. In *Proceedings of the 2006 SIGCOMM*, LSAD '06, pages 147–154, New York. ACM.

 ©2012 SBC — Soc. Bras. de Computação

# Um Modelo de Segurança e Privacidade para Redes Sociais Móveis Aplicadas à Área da Saúde

**Jesseildo Gonçalves**[1], **Ariel Teles**[1], **Francisco José Silva**[1]

[1]Laboratório de Sistemas Distribuídos - LSD
Universidade Federal do Maranhão - UFMA
São Luís - MA - Brasil

jesseildo@gmail.com, ariel@dee.ufma.br, fssilva@deinf.ufma.br

***Abstract.*** *Mobile Social Networks (MSN) comprise a social structure whose members are related in groups and the interaction is done through portable computing devices with access to wireless technologies. In health care it is possible to apply MSN concepts for conducting collaborative actions related to health care and its education. Recently many middleware for MSN have been proposed. However, the current MSN middleware are in a preliminary stage in relation to security and privacy requirements. This requirements becomes even more essentials when sensitive data are shared, such as in health applications where patient's profiles and their medical records are manipulated. This paper presents a security and privacy model designed for MSN applications with focus in the health care domain.*

***Resumo.*** *Redes Sociais Móveis (RSM) consistem de uma estrutura social cujo os membros se relacionam em grupos e a interação é feita através de dispositivos de computação portáteis com acesso a tecnologias de comunicação sem fio. Na área da saúde é possível aplicar o conceito de RSM para conduzir ações colaborativas relacionadas ao tratamento de pacientes e sua educação. Recentemente muitos middleware para RSM foram propostos. Entretanto, os atuais middleware de RSM estão em um estágio preliminar em relação a atender requisitos de segurança e privacidade. Esses últimos se tornam indispensáveis quando dados sensíveis são compartilhados, tais como em aplicações para saúde, onde os perfis dos pacientes e suas informações médicas são manipuladas. Este artigo apresenta um modelo de segurança e privacidade desenvolvido para aplicações de RSM focadas no domínio da saúde.*

## 1. Introdução

Uma Rede Social é uma estrutura social cujos membros se relacionam em grupos e a interação é realizada através de tecnologias da informação e comunicação. As Redes Sociais Móveis (RSM) são uma extensão das redes sociais, onde os usuários utilizam dispositivos portáteis com acesso a tecnologias de comunicação sem fio, agregando a capacidade de realizar interações sociais a qualquer hora e em qualquer lugar [Kayastha et al. 2011]. Uma rede social na área da saúde pode ser definida como um grupo de pessoas (e a estrutura social que elas coletivamente constroem) que utilizam tecnologias da informação e comunicação com o propósito de conduzir coletivamente ações relacionadas à assistência médica e sua educação [Demiris 2006]. Redes sociais

 ©2012 SBC — Soc. Bras. de Computação

na área da saúde podem ter como usuários diversos agentes envolvidos no processo de atenção a saúde, incluindo profissionais e pesquisadores da saúde, pacientes e seus familiares, como também membros da comunidade em geral.

No entanto, o desenvolvimento de RSM é complexo. O desenvolvedores precisam atentar para questões relacionadas ao compartilhamento de dados na rede, a mobilidade dos usuários, escalabilidade, a disponibilização de mecanismos de interação síncrona e assíncrona entre pessoas e questões relacionadas à privacidade e segurança dos dados, entre outras. Em aplicações de RSM para a área da saúde muitas informações compartilhadas são sensíveis e exigem um rigoroso nível de privacidade como, por exemplo, perfis de usuários e informações médicas de pacientes. Portanto, uma RSM segura deve garantir, sobretudo, a privacidade dos dados compartilhados na rede, buscando também manter a integridade e confidencialidade destas informações. Aaron Beach et al. [Beach et al. 2009] chamam atenção para um conjunto de problemas de segurança que uma RSM pode apresentar, enquanto que Hongyu Gao et al. [Gao et al. 2011] apresentam problemas comuns em redes sociais *on-line*, que também se manifestam em RSM. A especificidade do domínio pode acrescentar diversos requisitos de segurança ao software. Por exemplo, no domínio da saúde deve-se seguir um conjunto de requisitos e normas legais definidas por entidades médicas e/ou governamentais que variam de acordo com o país em que o sistema será usado. Assim, a construção de uma RSM segura requer mecanismos de segurança destinados a lidar com problemas e peculiaridades de ambientes de RSM, além do domínio de suas aplicações.

Este trabalho tem como objetivo apresentar um modelo de privacidade e segurança para RSM considerando os requisitos específicos da área da saúde. O desenvolvimento deste modelo está inserido no contexto do projeto MobileHealthNet, desenvolvido em parceria pelo Laboratório de Sistemas Distribuídos da Universidade Federal do Maranhão e o *Laboratory for Advanced Collaboration* da Pontifícia Universidade Católica do Rio de Janeiro. Este projeto tem por objetivo geral avançar o estado da arte em sistemas de *middleware* para redes sociais móveis, e usar o middleware para a criação de serviços de redes sociais móveis para a área da saúde. O projeto MobileHealthNet conta com apoio institucional do Hospital Universitário da UFMA (HU-UFMA). Este artigo está organizado como segue. A Seção 2 descreve o projeto o qual este trabalho está inserido. Na Seção 3 são descritos os componentes que compõem o modelo proposto. Os principais aspectos de implementação são mostrados na Seção 4, enquanto a Seção 5 aborda os principais trabalhos relacionados a esta iniciativa. Por fim, na Seção 6 são descritas as conclusões e os próximos passos do trabalho.

## 2. O projeto MobileHealthNet

O projeto MobileHealthNet visa permitir serviços para colaboração entre os profissionais da saúde, entre os pacientes e entre profissionais da saúde e pacientes, onde a colaboração ocorrerá em ambientes móveis. Em especial, este projeto foi concebido para ser aplicado a comunidades carentes e remotas. As aplicações previstas no âmbito deste projeto tem por objetivo: (i) encurtar a distância entre profissionais da saúde e pacientes; (ii) facilitar a colaboração entre profissionais da saúde de diversas especialidades; (iii) promover a educação em saúde tanto para pacientes como profissionais da saúde; e (iv) promover um meio de comunicação entre profissionais dos diversos níveis da atenção a saúde de forma a facilitar o intercâmbio e colaboração entre os mesmos.

 ©2012 SBC — Soc. Bras. de Computação

Após diversas sessões de interação entre profissionais da computação e da saúde, foram definidos os requisitos principais para construção do *middleware* e aplicações. A arquitetura do software foi organizada em cinco camadas, descritas a seguir.

A **Camada de Comunicação** é responsável por gerenciar o envio e recebimento de mensagens entre os dispositivos móveis, definindo os modelos de interação entre os usuários. Ela está sendo construída a partir da especificação *Data Distribution Service* (DDS) da OMG (*Object Management Group*), um padrão para comunicação *publish/subscribe* com qualidade de serviço que visa a distribuição crítica de informações em sistemas distribuídos de tempo real. A **Camada de Serviços Básicos** disponibiliza serviços básicos do *middleware*, como o serviço de gerenciamento de informações de contexto, o serviço de armazenamento de conteúdo e seus meta-dados, e o serviço de gerenciamento de usuários e grupos. A **Camada de Serviços de Aplicação** disponibiliza serviços típicos de redes sociais, como Serviço de Fórum, Serviço de Chat, Serviço de Mural e Serviço de Alertas, sendo este último para notificação de mensagens que exigem atenção prioritára por parte de seus destinatários. **A camada Aplicações**, compreende as aplicações previstas no projeto.

## 3. Modelo de Segurança para Redes Sociais Móveis

A camada **Serviços de Segurança** é transversal a todas as camadas da arquitetura do *middleware* e disponibiliza os mecanismos que implementam o modelo de privacidade e segurança proposto neste trabalho. Seu desenvolvimento está sendo conduzido através de um processo de desenvolvimento de software seguro, o CLASP[1].

Como primeiro passo para o desenvolvimento deste modelo, foi realizado o processo de elicitação de requisitos funcionais e não funcionais das aplicações do projeto junto aos profissionais da saúde. A seguir, foram identificados os requisitos específicos de segurança e privacidade, onde verificou-se que os mesmos deveriam seguir um conjunto de restrições legais e éticas definidas pelo Conselho Nacional de Saúde[2] e um conjunto de padrões e requisitos presentes no Manual de Certificação para Sistemas de Registro Eletrônico em Saúde [SBIS and CFM 2011], editado pela Sociedade Brasileira de Informática na Saúde (SBIS).

Os requisitos especificados englobam privacidade e confidencialidade das informações compartilhadas na rede social, mecanismos de controle de informações de contexto, notificação de presença, criação e customização de regras de acesso de granularidade fina, uso de processos de encriptação de dados, persistência das interações relevantes entre os usuários e construção de um canal seguro de comunicação para transferência de dados.

Após a fase de elicitação de requisitos, foram iniciadas as fases de identificação dos papéis e recursos do sistema, onde foram identificados os donos (*owners*) e usuários (*users*) de cada recurso do sistema, bem como especificado o controle de acesso a tais recursos. Além disso, foi realizado um levantamento sobre os possíveis ataques que uma RSM pode vir a sofrer, bem como medidas de defesa a estas ameaças.

[1] `https://www.owasp.org/index.php/Category:OWASP_CLASP_Project`
[2] `http://conselho.saude.gov.br/`

 ©2012 SBC — Soc. Bras. de Computação

Uma visão geral dos componentes que compõem o modelo de segurança proposto pode ser vista na Figura 1 e cada um desse componentes é apresentado a seguir.

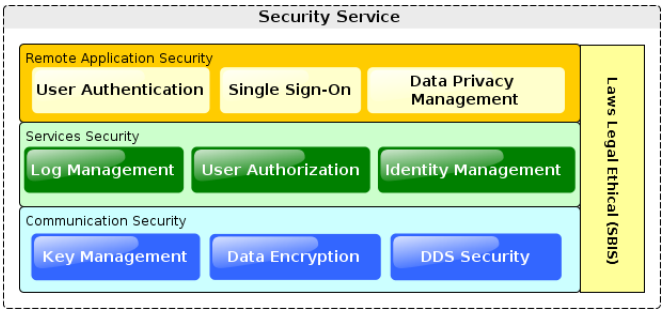



**Figura 1. Modelo de Segurança**

Em RSM para saúde, todas as operações devem ser realizadas por usuários autenticados, devendo ser garantido também o não repúdio de operações sobre os dados armazenados, uma vez que estas informações estão diretamente relacionadas ao processo de atendimento à saúde das pessoas. Este processo é realizado pelo componente de autenticação (***User Authentication***). Para que o usuário não seja obrigado a autenticar-se a cada serviço que ele necessite utilizar, o modelo dispõe de um mecanismo de autenticação ***Single Sign-On*** (SSO), onde uma única ação de autenticação permite que um usuário acesse todos os serviços aos quais possui permissão de acesso.

O modelo de segurança proposto provê um mecanismo de gerenciamento de privacidade (***Data Privacy Management***), o qual deve disponibilizar aos usuários meios para que sejam definidos as políticas de privacidade a serem aplicadas sobre seus próprios dados e informações de contexto. A implementação das regras definidas na política de privacidade será realizada através de diversos componentes, como o componente de autorização (***User Authorization***), responsável por gerenciar o controle de acesso às informações. Assim, qualquer operação sobre os dados deve passar primeiramente por este componente antes de ser efetivamente realizada.

O modelo também dispõe de um componente que permite registrar o acesso e manipulação de dados e informações de contexto sensíveis, o ***Log Management***. Através deste componente será possível auditar as ações realizadas pelos usuários do sistema. Este componente é também responsável pelo processo de monitoramento da RSM, armazenando as exceções que podem ocorrer no sistema, notificando-as ao administrador do sistema com o objetivo de se identificar tentativas de acesso indevido aos dados ou até mesmo tentativas de intrusão no sistema.

Atualmente o projeto MobileHealthNet possui quatro aplicações sendo desenvolvidas a partir do *middleware*. Para evitar que os mesmos usuários sejam cadastrados em cada uma dessas aplicações, no modelo de segurança é proposto um gerenciamento de identidades (***Identity Management***). Assim, uma vez cadastrado no sistema, o usuário pode acessar qualquer aplicação e serviço dentro do mesmo ambiente utilizando as mesmas credenciais.

Na comunicação DDS, o acesso aos dados é permitido somente aos usuários que estejam participando de um mesmo domínio. O domínio é uma abstração do DDS

 ©2012 SBC — Soc. Bras. de Computação

usada para isolar e otimizar a comunicação dentro de uma comunidade que compartilha interesses em comum. Porém, todos os usuários participantes de um domínio possuem acesso a qualquer informação nele publicada, através de um processo de subscrição em tópicos. Isto pode representar uma vulnerabilidade que pode levar a uma quebra de privacidade ou a um ataque do tipo *man-in-the-middle*. Portanto, faz-se necessário um mecanismo que torne a comunicação em um domínio DDS mais segura. Este é objetivo do ***DDS Security***, responsável por garantir que somente terão acesso aos dados publicados e a publicar no domínio os usuários que estejam devidamente autenticados e autorizados. Deve-se prover mecanismos para o estabelecimento de canais seguros de comunicação no DDS, garantindo-se ainda as propriedades básicas de autenticidade, integridade e confidencialidade na distribuição dos dados. O componente ***Data Encryption*** é responsável pelos recursos de criptografia, enquanto o componente ***Key Management*** tem por atribuição o gerenciamento das chaves criptográficas necessárias ao processo de encriptação.

## 4. Aspectos de Implementação

O desenvolvimento dos mecanismos de segurança e privacidade para o *middleware* MobileHealthNet faz uso de algoritmos, protocolos e tecnologias presentes no meio industrial e acadêmico. Os serviços do *middleware* estão sendo desenvolvidos utilizando a tecnologia *Java Enterprise Edition* (J2EE) que, por sua vez, traz um conjunto de recursos destinados a autenticação e autorização, implementados pelo servidor de aplicações. Os mecanismo de autorização será baseado em papéis, por ser facilmente ajustável a mudanças nos requisitos das aplicações. Por outro lado, como as aplicações são destinadas a dispositivos móveis (em particular o Google Android neste primeiro protótipo), os componentes de autenticação e SSO estão sendo desenvolvidos reutilizando os recursos disponíveis na plataforma dos dispositivos.

Para prover o gerenciamento de *logs*, são realizadas interceptações às chamadas dos serviços, utilizando os recursos de interceptações disponibilizados pelo J2EE. Através destes, o desenvolvedor consegue interceptar uma chamada a um método de um objeto instanciado no contexto do servidor de aplicação, possibilitando a execução de instruções antes e/ou depois da execução do método. As informações atualmente armazenadas pelo gerenciador de *logs* correspondem ao nome do serviço, nome do método chamado, nome do usuário responsável pela realização da chamada e parâmetros de entrada passados ao método.

O gerenciamento de identidades está sendo implementado através de um modelo centralizado, onde uma única entidade é responsável por gerenciar as autenticações e atribuições de credenciais, as quais serão utilizadas para acessar os serviços. Este modelo torna o gerenciamento mais eficiente e mais flexível em relação a inclusão de novos serviços.

Para a segurança da comunicação no DDS, estão sendo analisados diversos protocolos: IPSec (*Security Architecture for the Internet Protocol*), TLS (*Transport Layer Security*), SSL (*Secure Sockets Layer*) e SRTP (*Secure Real-time Transport Protocol*). O SRTP provavelmente será o protocolo a ser utilizado, devido ao seu suporte *UDP multicast*, também utilizado pelo DDS.

 ©2012 SBC — Soc. Bras. de Computação

Para o gerenciamento de chaves, estamos avaliando o uso do protocolo MIKEY (Multimedia Internet KEYing). O MIKEY é um protocolo para gerenciamento de chaves destinado a uso de aplicações de tempo real, com suporte a diversos algoritmos de compartilhamento de chaves. A escolha dos algoritmos a serem adotados, no entanto, ainda não realizada, deve levar em consideração aspectos relacionados à escalabilidade e ao desempenho causado pela execução do algoritmo em dispositivos móveis.

## 5. Trabalhos Relacionados

Vários *middleware* para RSM foram recentemente propostos [Karam and Mohamed 2012]. No entanto, uma pequena parcela deles fornecem mecanismos de segurança e privacidade, entre os quais destacamos os seguintes.

O Mobilis [Lübke 2011] disponibiliza serviços para o compartilhamento de informações de contexto dos usuários, gerenciamento de grupos, serviços de armazenamento de arquivos e meta-dados a eles associados, como também edição colaborativa de textos e imagens. Ele tem sua comunicação baseada no protocolo *Extensible Messaging and Presence Protocol* (XMPP) e parte dos mecanismos de segurança são obtidos a partir desse protocolo. Isto inclui segurança da comunicação, realizada através do protocolo *Secure Sockets Layer* (SSL), especificado como extensão do XMPP. Além disso, o Mobilis provê um gerenciamento de privacidade e segurança das informações de contexto compartilhadas pelos usuários.

O MobiSoc [Gupta et al. 2009] garante a privacidade dos usuários através de um gerenciamento com granularidade fina, baseado em regras de controle de acesso às entidades no sistema (usuários e aplicações). Cada usuário registra suas preferências de privacidade sobre seus próprios conteúdos. O MobiSoc também provê mecanismo de autenticação e comunicação segura através do RSA em conjunto com o AES.

O MyNet [Kalofonos et al. 2008] disponibiliza um mecanismo de controle de acesso no qual o usuário determina quem terá acesso a suas informações. Para isso ele utiliza uma estrutura chamada de *Passlet*. Os *Passlets* incluem a quem e quais permissões estão sendo concedidas, sobre quais informações elas se aplicam e por quanto tempo as permissões serão válidas. O MyNet também provê mecanismo de autenticação, autorização e comunicação segura dos dados através do protocolo SSL.

Embora os *middleware* analisados reconheçam a importância da segurança, e também provêem alguns mecanismos, eles não atendem as necessidades do domínio da saúde. Por outro lado, o MobileHealthNet está focando neste domínio, portanto deve fornecer mecanismos de segurança e privacidade que atendam suas exigências, resultando em mecanismos de autorização mais rígidos, além dos não encontratos em outros *middleware*, como gerenciamento de identidade e gerenciamento de *logs*, utilizado para auditoria. Além disso, nosso *middleware*, através deste trabalho, inclui como diferencial a concepção de mecanismos de segurança específicos para uma abordagem de desenvolvimento centrado nos dados baseados na especificação OMG-DDS, dado que ainda não foram definidos padrões de segurança para esta especificação.

## 6. Conclusão e Trabalhos Futuros

Este trabalho propõe um modelo de privacidade e segurança projetado para suprir as necessidades e exigências no contexto do *middleware* MobileHealthNet, destinado a

 ©2012 SBC — Soc. Bras. de Computação

prover a segurança necessária para RSM voltadas ao domínio da saúde. Este modelo contribui também no desenvolvimento de mecanismos de segurança para o DDS, agregando a ele autenticidade, integridade e confidencialidade das mensagens.

Próximas etapas do desenvolvimento deste trabalho incluem a conclusão da implementação do modelo, começando pela autenticação e autorização, seguido pelo gerenciamento de identidades e *logs* e finalmente a segurança na comunicação. Após a implementação será realizada a avaliação do modelo, esta envolve aspectos quantitativos, como a análise do impacto dos mecanismos de privacidade e segurança no desempenho dos serviços disponibilizados pelo *middleware* e uma avaliação do atendimento aos requisitos especificados e a ameças identificadas para o cenário de RSM. Será ainda avaliado aspectos qualitativos relativas a usabilidade dos mecanismos de privacidade e segurança considerando a perspectiva dos usuários. Para tanto, serão utilizados questionários e técnicas de entrevistas.

## Agradecimentos

Os autores gostariam de agradecer a FAPEMA por bolsa de estudo e pelo financiamento deste projeto, processo APP-00932/10, e a CAPES por bolsa de estudo.

## Referências

Beach, A., Gartrell, M., and Han, R. (2009). Solutions to security and privacy issues in mobile social networking. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, pages 1036–1042, Washington, DC, USA. IEEE Computer Society.

Demiris, G. (2006). The diffusion of virtual communities in health care: concepts and challenges. *Patient Education and Counseling*, 62(2):178–188.

Gao, H., Hu, J., Huang, T., Wang, J., and Chen, Y. (2011). Security issues in online social networks. *Internet Computing, IEEE*, 15(4):56 –63.

Gupta, A., Kalra, A., Boston, D., and Borcea, C. (2009). Mobisoc: a middleware for mobile social computing applications. *Mobile Networks and Applications*, 14(1):35–52.

Kalofonos, D. N., Antoniou, Z., Reynolds, F. D., Van-Kleek, M., Strauss, J., and Wisner, P. (2008). Mynet: A platform for secure p2p personal and social networking services. In *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, PerCom '08, pages 135–146.

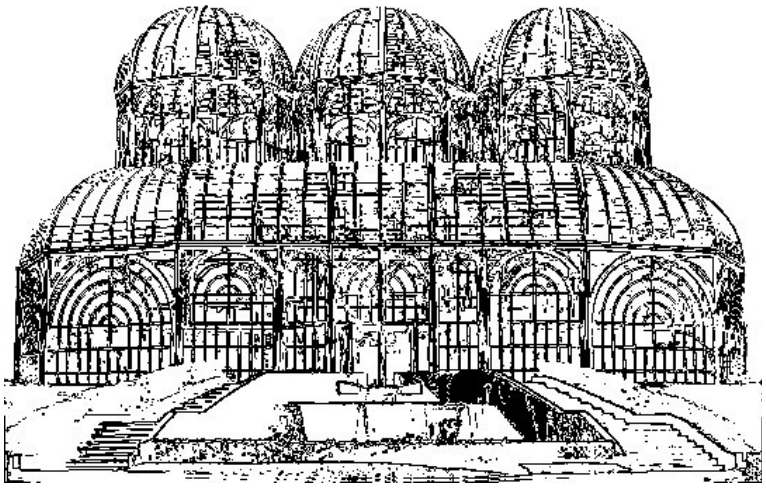
Karam, A. and Mohamed, N. (2012). Middleware for mobile social networks: A survey. *Proceedings of the 45th Hawaii International Conference on System Sciences*, pages 1482–1490.

Kayastha, N., Niyato, D., Wang, P., and Hossain, E. (2011). Applications, architectures, and protocol design issues for mobile social networks: A survey. *Proceedings of the IEEE*, 99(12):2130–2158.

Lübke, R. (2011). *Ein Framework zur Entwicklung mobiler Social Software auf Basis von Android*. PhD thesis, Dresden, Germany.

SBIS and CFM (2011). Manual de Certificação para Sistemas de Registro Eletrônico em Saúde (S-RES).

 ©2012 SBC — Soc. Bras. de Computação



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# CTDSeg – Concurso de Teses e Dissertações em Segurança

# Implementação eficiente em *software* de curvas elípticas e emparelhamentos bilineares

**Diego de Freitas Aranha**[1], **Julio López (Orientador)**[1]

[1] Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
CEP 13083-970 – Campinas – SP - Brazil
{dfaranha,jlopez}@ic.unicamp.br

***Abstract.*** *The development of asymmetric or public key cryptography made possible new applications of cryptography such as digital signatures and electronic commerce. Elliptic Curve Cryptography is among the most efficient public-key methods because of its low storage and computational requirements. The relatively recent advent of Pairing-Based Cryptography allowed further constructions of flexible and innovative cryptographic solutions. However, the computational cost of pairing-based cryptosystems remains significantly higher than traditional public key cryptosystems and thus an important obstacle for adoption, specially in resource-constrained devices. The main contributions of this work aimed to improve the performance of curve-based cryptosystems, consisting of efficient formulation and implementation of finite field arithmetic in different computing platforms; and serial and parallel algorithmic techniques for arithmetic in elliptic curves and computation of cryptographic pairings. These contributions produced important performance improvements and several speed records for computing relevant cryptographic algorithms in modern computer architectures ranging from embedded 8-bit microcontrollers to 8-core processors.*

***Resumo.*** *O advento da criptografia assimétrica ou de chave pública possibilitou a aplicação de criptografia na forma de assinaturas digitais e comércio eletrônico. Dentre os vários métodos de criptografia assimétrica, a Criptografia de Curvas Elípticas destaca-se pelos baixos requisitos de armazenamento e custo computacional para execução. A descoberta relativamente recente da criptografia baseada em emparelhamentos bilineares permitiu ainda sua flexibilização e a construção de sistemas criptográficos com propriedades inovadoras. Porém, o custo computacional de sistemas baseados em emparelhamentos ainda permanece significativamente superior aos tradicionais, representando um obstáculo para sua adoção, especialmente em dispositivos com recursos limitados. As contribuições deste trabalho objetivaram aprimorar o desempenho de sistemas baseados em curvas elípticas e emparelhamentos bilineares e consistem em formulação e implementação eficientes de aritmética em corpos finitos em diversas plataformas computacionais; técnicas algorítmicas seriais e paralelas para aritmética em curvas elípticas e cálculo de emparelhamentos de interesse criptográfico. Estas contribuições permitiram obter significativos ganhos de desempenho e uma série de recordes de velocidade para o cálculo de diversos algoritmos criptográficos relevantes em arquiteturas modernas que vão de sistemas embarcados de 8* bits *a processadores com 8* cores.

[1]Financiado por FAPESP, Processo No. 2007/06950-0.

 ©2012 SBC — Soc. Bras. de Computação

## 1. Introdução

A descoberta da criptografia de chave pública [1] revolucionou a forma de se construir sistemas criptográficos e possibilitou, de forma definitiva, a integração entre teoria criptográfica e implementação em aplicações reais. Em particular, trouxe a possibilidade de se estabelecer serviços criptográficos como sigilo e assinatura irretratável em ambientes onde não existe qualquer relação de confiança entre os envolvidos ou canal seguro para distribuição de chaves. O surgimento de infra-estruturas de chaves públicas solucionou o problema de tituaridade de chaves públicas e impulsionou o comércio eletrônico. Por outro lado, criou diversos problemas adicionais, durante operações de validação de certificados e revogação de chaves públicas. A descoberta de sistemas criptográficos baseados no problema do logaritmo discreto em curvas elípticas [2, 3] produziu uma nova revolução na área. Ao apresentarem desempenho superior e exigirem chaves mais curtas para um mesmo nível de segurança que os métodos tradicionais de criptografia assimétrica, especialmente o algoritmo RSA [4], alguns dos problemas inerentes às infra-estruturas de chaves públicas foram minimizados. Contudo, a dificuldade de gerência e a sobrecarga de desempenho decorrentes da utilização de certificados ainda impõe obstáculos na adoção de criptografia assimétrica em ambientes restritos.

A busca de alternativas ao paradigma tradicional de infra-estruturas de chave pública resultou na descoberta de sistemas baseados em identidade. Foram concebidos inicialmente por Shamir [5], em 1984, para assinaturas digitais, mas as primeiras realizações funcionais e eficientes para cifração só foram apresentadas em 2001, por Boneh e Franklin [6], e Sakai, Ohgishi e Kasahara [7], a partir de emparelhamentos bilineares sobre curvas elípticas. A motivação original para sistemas baseados em identidade era aproveitar a autenticidade de informação publicamente conhecida para simplificar a autenticação de chaves públicas. Até então, a única aplicação de emparelhamentos bilineares em criptografia era atacar sistemas criptográficos de curvas elípticas [8]. Após a aplicação de emparelhamentos para concretizar cifração baseada em identidade, uma gama de novos protocolos com propriedades inovadoras e especiais foi desenvolvida. Isto levou a uma flexibilização enorme das primitivas criptográficas conhecidas e ampliou os cenários de aplicação de criptografia assimétrica de forma considerável. Entre os protocolos baseados em problemas sobre grupos bilineares, destacam-se: acordo de chaves eficientes para múltiplas entidades [9], assinaturas curtas e agregadas [10] e paradigmas alternativos de certificação implícita [11].

## 2. Justificativa e objetivo

Apesar das propriedades inovadoras, o desempenho de sistemas criptográficos baseados em emparelhamentos ainda representa um obstáculo. Tipicamente, o cálculo de um emparelhamento bilinear ainda é comparável a uma decifração/assinatura RSA, e uma ordem de magnitude menos eficiente que uma multiplicação de ponto em curvas elípticas. Isto é natural, visto que os métodos mais estabelecidos de criptografia assimétrica puderam receber maior esforço de pesquisa, produzindo algoritmos cada vez mais eficientes. Esforço similar já é consistente em criptografia baseada em emparelhamentos [12], resultando em novos algoritmos [13, 14, 15, 16] e novas curvas adequadas à sua instanciação [17].

Este projeto teve como finalidade desenvolver algoritmos eficientes (seqüenciais e

 ©2012 SBC — Soc. Bras. de Computação

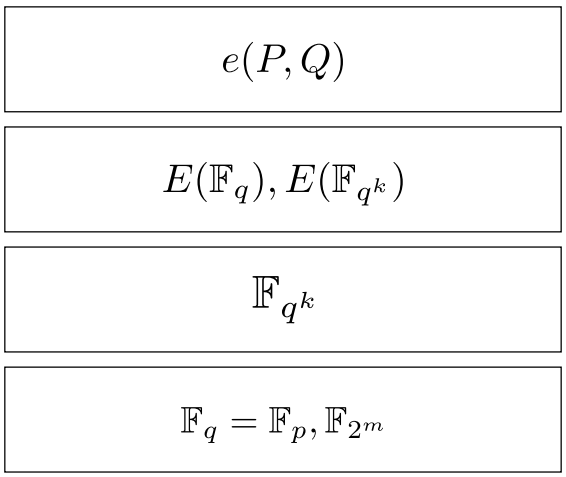



**Figura 1. Níveis de aritmética envolvidos no cálculo de um emparelhamento bilinear.**

paralelos) e implementações em *software* otimizadas para criptografia de curvas elípticas e criptografia baseada em emparelhamentos. Vários níveis de aritmética são empregados [18]: algoritmos para o cálculo do emparelhamento $e(P, Q)$ propriamente dito [12, 14], aritmética na curva elíptica onde o emparelhamento bilinear encontra-se definido [19], aritmética no corpo finito onde a curva elíptica está definida [20] e aritmética nas extensões deste corpo finito [21]. Estes níveis de aritmética encontram-se visualmente representados na Figura 1 em uma organização descendente partindo do cálculo do emparelhamento.

O objetivo principal deste trabalho consistiu em tornar estes métodos de criptografia mais eficientes nas arquiteturas modernas, abrangendo tanto pesquisa algorítmica quanto pesquisa aplicada de implementação. A implementação exigiu o projeto de técnicas de otimização de algoritmos em arquiteturas modernas (embarcadas, multiprocessadas) e concretizou os algoritmos em código funcional eficiente, fazendo o melhor uso possível dos recursos disponibilizados pelo *hardware*. Para isso, paralelismo em nível de tarefas e em nível de dados foram extensamente utilizados, incluindo a aplicação de multiprocessamento e conjuntos de instruções vetoriais como SSE e AVX.

## 3. Contribuições

Esta seção apresenta um relato em duas partes dos resultados obtidos: implementação de criptografia de curvas elípticas e de criptografia baseada em emparelhamentos bilineares.

### 3.1. Implementação de criptografia de curvas elípticas

O primeiro conjunto de resultados foi derivado da implementação de corpos e curvas elípticas binárias em sensores sem fio. O baixo poder computacional dos sensores torna inviável a utilização de algoritmos convencionais de criptografia de chave pública (RSA, por exemplo) e, até recentemente, primitivas de segurança como sigilo, autenticação e integridade eram alcançadas apenas através de técnicas de criptografia simétrica. Nos artigos [22, 23, 24] publicados como resultados da tese, são apresentadas técnicas para implementação de

 ©2012 SBC — Soc. Bras. de Computação

aritmética em corpos binários em microcontroladores AVR ATmega de 8 *bits* que aproveitam ao máximo os recursos da plataforma subjacente em busca de desempenho. A implementação de curvas elípticas no nível de segurança de 80 *bits* sobre estes corpos permite calcular uma multiplicação de ponto aleatório, operação fundamental de protocolos baseados em curvas elípticas, em menos de $\frac{1}{3}$ segundo. Este resultado contraria diversas observações levantadas em trabalhos anteriores sobre a inviabilidade de curvas binárias para dispositivos limitados, e aperfeiçoa o estado-da-arte entre 57% e 61%. As otimizações desenvolvidas podem também ser empregadas em outras arquiteturas com conjunto de instruções limitado e latência proibitiva em operações de acesso à memória, típicas do segmento de microcontroladores.

O segundo conjunto de resultados trata da implementação de corpos binários em conjuntos de instruções vetoriais. Em particular, havia interesse em se aproveitar explicitamente instruções de permutação de *bytes* que codificam implicitamente acessos simultâneos a tabelas de constantes. Foi desenvolvida uma formulação da aritmética com granularidade de 4 *bits* que emprega vastamente as instruções de permutação, produzindo inclusive um novo ainda que ineficiente algoritmo de multiplicação. Esta formulação foi complementada com um estudo detalhado dos impactos algorítmicos da disponibilidade repentina de suporte nativo à multiplicação em um corpo binário recentemente introduzida na arquitetura Intel. A aceleração significativa da operação de multiplicação forçou uma reavaliação de estratégias de implementação para a operação de multiplicação de ponto em curvas elípticas binárias e sua paralelização em múltiplas unidades de processamento. Estes resultados são apresentados nos artigos [25, 26, 27]. O primeiro trabalho obtém ganhos de desempenho entre 8% e 84% para diversas operações em um corpo binário. Esta formulação eficiente da aritmética permitiu aprimorar o estado da arte para multiplicação de ponto em 27%-30%, desconsiderando o modo de operação em lote. O custo médio de uma multiplicação de ponto em modo lote no nível de segurança de 128 *bits* foi superado em 10% apenas nos dois últimos trabalhos, após a introdução de suporte nativo à multiplicação em um corpo binário. Estes últimos apresentam ainda resultados experimentais para implementações seriais e paralelas da operação de multiplicaçao de ponto nos níveis de segurança de 112, 128 e 192 *bits*.

### 3.2. Implementação de criptografia baseada em emparelhamentos bilineares

Em seguida, buscou-se estudar protocolos úteis para o fornecimento de serviços de segurança em redes de sensores sem fio. Um exemplos com estas características é um protocolo de acordo de chaves não-interativo, providenciando o acordo autenticado de chaves simétricas entre nós sensores sem exigir qualquer comunicação, o que permite importante economia de energia. Outro exemplo é o emprego de assinaturas digitais para transmissão autenticada de mensagens entre um nó específico da rede de sensores e um usuário ou aplicação final. Esta linha de pesquisa produziu os resultados [28, 29, 30]. Os dois primeiros trabalhos obtém uma aceleração de até 61% no cálculo do emparelhamento $\eta_T$ sobre curvas supersingulares binárias a partir de uma generalização das técnicas previamente desenvolvidas. O terceiro trabalho compara a implementação de diversos esquemas de assinatura digital sobre curvas elípticas e emparelhamentos bilineares, fornecendo um panorama completo da relação entre tempo de execução, consumo de memória e consumo de energia. Surpreendentemente, foi detectado que assinaturas curtas não fornecem a economia de energia suficiente em sensores

 ©2012 SBC — Soc. Bras. de Computação

para justificar sua escolha em detrimento de esquemas convencionais.

Dada a tendência tecnológica recente da indústria de computação em migrar as arquiteturas computacionais para arquiteturas paralelas, algoritmos paralelos para cálculo de emparelhamentos em arquiteturas multiprocessadas são desejáveis. Desta forma, foi derivada uma formulação paralela do Algoritmo de Miller [31] empregado para o cálculo de emparelhamentos bilineares. Esta formulação fornece um algoritmo genérico independente da instanciação do emparelhamento e com ótima escalabilidade em curvas sobre corpos de característica pequena [32]. Essa construção foi então acelerada com técnicas para implementação de corpos binários em processadores equipados com conjuntos de instruções vetoriais para reduzir sobrecargas da paralelização e produziu aprimoramentos em relação ao estado-da-arte de 28%, 44% e 66% empregando 2, 4 e 8 processadores, respectivamente. A aceleração do cálculo serial foi também significativa e situou-se em pouco mais de 24% [33, 34]. As mesmas técnicas permitiram ainda o estabelecimento de um novo recorde de velocidade para emparelhamentos simétricos com a derivação e implementação de um novo emparelhamento *eta* otimal sobre curvas supersingulares binárias de genus 2 [35]. Para o cálculo de emparelhamentos assimétricos, propomos várias técnicas para sua implementação serial no nível de segurança de 128 *bits* [36]: (i) a generalização da noção de *redução modular preguiçosa* para aritmética em corpos de extensão e em *twists* de curvas elípticas; (ii) o desenvolvimento de novas fórmulas para o cálculo de quadrados comprimidos sucessivos em subgrupos ciclotômicos de corpos de extensão; (iii) a eliminação de penalidades de desempenho em parametrizações negativas de curvas Barreto-Naehrig [37]. Estas novas técnicas estabelecem um novo recorde de velocidade para o cálculo de qualquer emparelhamento, aprimorando o melhor resultado anterior em até 34%. Finalmente, revisitamos o problema do cálculo paralelo de emparelhamentos, aprimorando significativamente os resultados obtidos anteriormente com o emprego dos quadrados comprimidos em grupos ciclotômicos, do suporte nativo à multiplicação binária e de um compromisso entre tempo de execução e espaço de armazenamento para acelerar a exponenciação final do emparelhamento $\eta_T$. Além disso, como tentativa de se contornar os obstáculos à paralelização do Algoritmo de Miller, foi proposto um novo emparelhamento baseado no emparelhamento de Weil. Esta nova construção permitiu incrementar a escalabilidade do cálculo do emparelhamento *ate* otimal primo em máquinas com até 8 unidades de processamento [38].

A tese consolidou 15 resultados submetidos ao processo de revisão por pares: 1 trabalho aceito em revista não-indexada por ter sido recentemente fundada na área, 3 trabalhos aceitos em revistas indexadas (Qualis A2, B1 e B5), 2 publicações em anais de conferências nacionais organizadas pela Sociedade Brasileira de Computação (SBC), 3 publicações em *workshops* de especialistas e 6 publicações em conferências internacionais, sendo 4 destas de bastante destaque na área de Computação (Qualis A1/A2) e 2 delas de maior prestígio por serem diretamente organizadas pela Associação Internacional de Pesquisa em Criptologia (*International Association for Cryptologic Research – IACR*).

 ©2012 SBC — Soc. Bras. de Computação

### 3.3. Biblioteca criptográfica

Um efeito colateral direto deste trabalho foi a fundação do projeto *RELIC* (*RELIC is an Efficient LIbrary for Cryptography*) [39]. O projeto teve como finalidade inicial a produção de uma biblioteca criptográfica para dispositivos embutidos que tivesse um menor consumo de memória que as alternativas. Atualmente, o projeto já constitui um *framework* completo para experimentação com implementação eficiente de algoritmos criptográficos, fornecendo ampla portabilidade e arquitetura modular especialmente projetada para permitir acelerações dependentes de arquitetura. A versão em repositório da biblioteca já atingiu a marca de 70 mil linhas de código, contando com mais de 8500 acessos de 1300 visitantes únicos provenientes de 74 países, e mais de 1500 *downloads* distribuídos nas 9 versões já disponibilizadas.

## 4. Conclusões

Dentre os vários recordes de velocidade para implementações em *software* no nível de segurança de 128 *bits* apresentados neste trabalho, nenhum foi superado até então. Desta forma, todos os resultados aqui apresentados permanecem como estado-da-arte em suas áreas correspondentes. São estes: recorde de velocidade para a multiplicação em curvas binárias genéricas resistente a canais laterais e para o cálculo serial ou paralelo de emparelhamentos bilineares nos cenários simétrico e assimétrico. Estas contribuições foram disponibilizadas na forma de uma biblioteca criptográfica para microcontroladores e máquinas multiprocessadas.

## Referências

[1] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, November 1976.

[2] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of computation*, 48:203–9, 1987.

[3] V. Miller. Uses of Elliptic Curves in Cryptography. In H. C. Williams, editor, *CRYPTO 1985*, volume 218 of *LNCS*, pages 417–426. Springer, 1985.

[4] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[5] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

[6] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[7] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing over Elliptic Curve (in Japanese). In *The 2001 Symposium on Cryptography and Information Security*, 2001.

[8] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC 91*, pages 80–89. ACM, 1992.

[9] A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–276, 2004.

 ©2012 SBC — Soc. Bras. de Computação

[10] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[11] Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless Public Key Cryptography. In C.-S. Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 452–473. Springer, 2003.

[12] Paulo S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 354–368. Springer, 2002.

[13] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Efficient Implementation of Pairing-Based Cryptosystems. *Journal of Cryptology*, 17(4):321–334, 2004.

[14] P. S. L. M. Barreto, S. Galbraith, C. Ó hÉigeartaigh, and M. Scott. Efficient Pairing Computation on Supersingular Abelian Varieties. *Designs, Codes and Cryptography*, 42(3):239–271, 2007.

[15] H. Lee E. Lee and C. Park. Efficient and Generalized Pairing Computation on Abelian Varieties. *IEEE Transactions on Information Theory*, 55(4):1793–1803, 2009.

[16] F. Vercauteren. Optimal pairings. *IEEE Trans. Inf. Theor.*, 56(1):455–461, 2010.

[17] D. Freeman, M. Scott, and E. Teske. A Taxonomy of Pairing-Friendly Elliptic Curves. *Journal of Cryptology*, 23(2):224–280, 2010.

[18] M. Scott. Implementing Cryptographic Pairings. In *PAIRING 2007*, volume 4575 of *LNCS*, pages 177–196. Springer, 2007.

[19] C. Costello, T. Lange, and M. Naehrig. Faster Pairing Computations on Curves with High-Degree Twists. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 224–242. Springer, 2010.

[20] J. López and R. Dahab. High-Speed Software Multiplication in GF($2^m$). In B. K. Roy and E. Okamoto, editors, *INDOCRYPT 2000*, volume 1977 of *LNCS*, pages 203–212. Springer, 2000.

[21] A. J. Devegili, C. Ó hÉigeartaigh, M. Scott, and R. Dahab. Multiplication and Squaring on Pairing-Friendly Fields. Cryptology ePrint Archive, Report 2006/471, 2006. `http://eprint.iacr.org/`.

[22] D. F. Aranha, D. Câmara, J. López, L. B. Oliveira, and R. Dahab. Implementação eficiente de criptografia de curvas elípticas em sensores sem fio. In *SBSEG 2008*, pages 173–186.

[23] D. F. Aranha, J. López, L. B. Oliveira, and R. Dahab. Efficient implementation of elliptic curves on sensor nodes. In *CHiLE 2009*, 2009.

[24] D. F. Aranha, L. B. Oliveira, J. López, and R. Dahab. Efficient implementation of elliptic curve cryptography in wireless sensors. *Adv. Math. Comm.*, 4(2):169–187, 2010.

[25] D. F. Aranha, J. López, and D. Hankerson. Efficient Software Implementation of Binary Field Arithmetic Using Vector Instruction Sets. In *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 144–161, 2010.

 ©2012 SBC — Soc. Bras. de Computação

[26] J. Taverne, A. Faz-Hernández, D. F. Aranha, F. Rodríguez-Henríquez, D. Hankerson, and J. López. Software implementation of binary elliptic curves: Impact of the carry-less multiplier on scalar multiplication. In *CHES 2011*, volume 6917 of *LNCS*, pages 108–123. Springer, 2011.

[27] J. Taverne, A. Faz-Hernández, D. F. Aranha, F. Rodríguez-Henríquez, D. Hankerson, and J. López. Speeding scalar multiplication over binary elliptic curves using the new carry-less multiplication instruction. *J. Cryptographic Engineering*, 1(3):187–199, 2011.

[28] D. F. Aranha, L. B. Oliveira, J. López, and R. Dahab. NanoPBC: implementing cryptographic pairings on an 8-bit platform. In *CHiLE 2009*, 2009.

[29] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab. TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks. *Computer Communications*, 4(2):169–187, 2011.

[30] C. P. L. Gouvêa D. F. Aranha J. López B. Priyantha M. Goraczko F. Zhao L. B. Oliveira, A. Kansal. Secure-TWS: Authenticating Node to Multi-user Communication in Shared Sensor Networks. *The Computer Journal*, 55(4):384–396, 2012.

[31] V. S. Miller. The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[32] D. F. Aranha and J. López. Paralelização em software do Algoritmo de Miller. In *SBSEG 2009*, pages 27–40, 2009.

[33] D. F. Aranha, J. López, and D. Hankerson. High-Speed Parallel Software Implementation of the $\eta_T$ Pairing. In *SPEED-CC 2009*, pages 73–88, 2009.

[34] D. F. Aranha, J. López, and D. Hankerson. High-Speed Parallel Software Implementation of the $\eta_T$ Pairing. In *CT-RSA 2010*, volume 5985 of *LNCS*, pages 89–105. Springer, 2010.

[35] D. F. Aranha, J.-L. Beuchat, J. Detrey, and N. Estibals. Optimal eta pairing on supersingular genus-2 binary hyperelliptic curves. In O. Dunkelman, editor, *CT-RSA 2012*, volume 7178 of *LNCS*, pages 98–115. Springer, 2012.

[36] D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. López. Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 48–68. Springer, 2010.

[37] P. S. L. M. Barreto and M. Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In B. Preneel and S. E. Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, 2005.

[38] D. F. Aranha, A. Menezes, E. Knapp, and F. Rodríguez-Henríquez. Parallelizing the Weil and Tate Pairings. In *IMACC 2011*, volume 7089 of *LNCS*, pages 275–295, 2011.

[39] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. `http://code.google.com/p/relic-toolkit/`.

 ©2012 SBC — Soc. Bras. de Computação

# Criptografia de Chave Pública sem Certificado

**Denise Goya**[1], **Routo Terada**[1]

[1]Departamento de Ciência da Computação – IME – Universidade de São Paulo

{dhgoya, rt}@ime.usp.br

***Abstract.*** *Certificateless public key cryptography is an alternative model to traditional asymmetric key cryptography, because the public key authentication occurs implicitly during a protocol run, with no need of digital certificates management and distribution. In this PhD thesis summary, formal security models for certificateless authenticated key agreement are improved with two independent objectives: (1) to increase the trust level for the partial secret key generating authority on which users rely, and (2) to enable computationally efficient protocols, with significant security properties. In order to demonstrate that these improvements made are feasible and achieve the objectives, new protocols are proposed and are proved secure.*

***Resumo.*** *A criptografia de chave pública sem certificado (certificateless) é uma alternativa ao modelo convencional de criptografia assimétrica, pois a autenticação da chave pública ocorre implicitamente durante a execução dos protocolos, sem a necessidade de gerenciamento e distribuição de certificados digitais. Neste resumo de tese de doutorado, modelos formais de segurança para acordo de chave com autenticação sem certificado são aprimorados visando dois objetivos paralelos: (1) aumentar o nível de confiança que usuários podem depositar na autoridade geradora de chaves secretas parciais e (2) viabilizar protocolos que sejam eficientes computacionalmente e com propriedades de segurança relevantes. Para atestar que as melhorias efetuadas são praticáveis e possibilitam que os objetivos sejam alcançados, novos protocolos são propostos e provados seguros.*

## 1. Introdução

A criptografia de chave pública surgiu com a busca por soluções para dois problemas intrinsecamente relacionados com o modelo de criptografia simétrica: o de distribuir uma chave secreta e o de autenticar uma entidade, com garantia de irretratabilidade. Ambos problemas foram tratados por Diffie e Hellman, quando nasceu um novo paradigma para a criptografia, a de chave pública [Diffie e Hellman 1976]. Nesse modelo, também chamado de assimétrico, todo usuário possui um par de chaves, uma pública e outra secreta. A chave pública pode ser divulgada por meio de um canal público e seu valor é matematicamente relacionado com o valor da chave secreta correspondente, que convencionalmente fica sob guarda exclusiva de seu dono. Enquanto uma chave é usada para realizar uma operação criptográfica, a outra é requerida para que tal operação possa ser invertida.

É natural que um novo paradigma traga consigo novos problemas. Com a criptografia de chave pública, não foi diferente. Um problema central nesse modelo é o de

 ©2012 SBC — Soc. Bras. de Computação

legitimação da chave pública. Como garantir que uma chave pública pertence, de fato, a uma entidade? No modelo convencional, o valor da chave pública não fornece indícios de quem seja seu proprietário. No entanto, os criptossistemas assimétricos funcionam sob a premissa de que a chave pública é certificada, ou seja, é vinculada com seu dono. Uma solução que se tornou prática comum é a de implantação de uma infraestrutura de chaves públicas (ICP, ou PKI – *Public Key Infrastructure*), cuja característica principal são os certificados digitais de chave pública. Tal solução, entretanto, embute dificuldades, como as citadas abaixo:

- processos complexos de implantação e manutenção da infraestrutura;
- custos de emissão, distribuição e armazenamento de certificados;
- custos para recuperar e validar certificados;
- dificuldades com revogação de certificados.

Há pelo menos dois caminhos para minimizar essas dificuldades: modificar o modelo de funcionamento da infraestrutura, que não é escopo deste trabalho, e modificar o próprio modelo de criptografia de chave pública. Esta última abordagem dá origem ao que chamamos de modelos alternativos.

O modelo de criptografia de chave pública baseado em identidade, proposto por Shamir em 1984, dispensa a distribuição de certificados digitais e de uma ICP que os gerencie, uma vez que a chave pública de um usuário é sua própria identificação no sistema (ou seja, dados pessoais, como nome, número do CPF, telefone celular, endereço IP ou de correio eletrônico) [Shamir 1984]. Uma característica inerente ao modelo de criptografia de chave pública baseado em identidade é a de custódia de chaves, em que a autoridade do sistema conhece as chaves secretas de todos seus usuários. A custódia de chaves é indesejável em vários tipos de aplicações; em particular, impossibilita a garantia de irretratabilidade, pois o usuário não é o único conhecedor de sua chave secreta.

Em 2003, Al Riyami e Paterson introduziram o paradigma de criptografia de chave pública sem certificado, que une propriedades do modelo convencional com o baseado em identidade [Al-Riyami e Paterson 2003]. Nesse paradigma, também conhecido por *certificateless*, a ICP convencional é dispensada, pois a identidade de um usuário fica parcialmente relacionada com a chave secreta. E não há custódia de chaves, pois o usuário é o único conhecedor de sua chave criptográfica secreta. Ademais, para uso interno a uma dada instituição (uma empresa privada ou órgão do governo) este paradigma pode ser apropriado, uma vez que a entidade geradora das chaves (chamada *Key Generating Centre*, KGC) pode ser um membro idôneo desta instituição. Entretanto, justamente pelo fato da chave pública não precisar de certificados, há que se levar em consideração que as chaves públicas podem ser arbitrariamente substituídas por um adversário.

### 1.1. Segurança de Protocolos Criptográficos

O desenvolvimento de protocolos criptográficos de chave pública é baseado em modelos de segurança que descrevem ações de um adversário contra um criptossistema e um objetivo de comprometimento de propriedades do protocolo que, em suma, define o que vem a ser segurança formal para aquele tipo de protocolo. Denomina-se segurança demonstrável a metodologia para o estabelecimento de modelos de segurança e respectiva prova para protocolos criptográficos [Castro et al. 2007]. De forma sucinta, uma prova de segurança formal parte da existência de um problema computacional difícil (para o qual

 ©2012 SBC — Soc. Bras. de Computação

não se conhece solução de tempo polinomial) e mostra que a existência de um adversário com poderes descritos no modelo de segurança e que seja capaz de atingir o objetivo em tempo polinomial implica na existência de um algoritmo de tempo polinomial para a solução do problema suposto difícil.

No modelo de chave pública sem certificado, a definição dos modelos de segurança tem se mostrado complexa, dada a necessidade de se lidar com um adversário que altera o valor da chave pública de usuários à sua escolha. Dent, em 2008, apresenta uma análise dos modelos de segurança para esquemas de cifragem no modelo de chave pública sem certificado e relaciona uma grande diversidade de modelos e adversários com variados graus de poder [Dent 2008]. Considerando-se que esquemas de cifragem garantem autenticidade apenas do usuário de destino, eleva-se consideravelmente a complexidade dos modelos de segurança para protocolos que visam garantir a autenticidade mútua dos usuários (autenticação na origem e no destino das mensagens). Uma classe particular de protocolos com autenticidade mútua é a de estabelecimento de chave secreta com autenticação. No trabalho de Swanson e Jao, relata-se que todos os protocolos para acordo de chave secreta com autenticação sob chave pública sem certificado (*Certificateless Authenticated Key Agreement*, ou CL-AKA) eram inseguros até então e se propõe novo modelo de segurança [Swanson e Jao 2009].

Vale notar que primitivas de estabelecimento de chave secreta são peças fundamentais no desenvolvimento de soluções de segurança de sistemas computacionais, pois quase sempre são a base dessas soluções. No entanto, os protocolos CL-AKA conhecidos que foram mostrados seguros são pouco eficientes e seguem modelos de segurança mais complexos que o de Swanson e Jao, como em [Lippold et al. 2009] e [Yang e Tan 2011], ou satisfazem modelos de adversários mais fracos, como em [Zhang et al. 2010] e [Lippold e González Nieto 2010].

### 1.2. Objetivos

O objetivo primário desta pesquisa é o desenvolvimento de protocolos em que autenticidade é um requisito, dentro do modelo de chave pública sem certificado. Algumas questões que buscamos responder são:

(Q.1) Como se caracterizam os modelos alternativos de criptografia de chave pública, quais são suas limitações e como eles se relacionam entre si?
(Q.2) Quais são as propriedades e restrições dos diferentes modelos de segurança para protocolos de acordo de chave secreta com autenticação sob chave pública sem certificado (CL-AKA)? Em que condições cada um é indicado?
(Q.3) Até que ponto os modelos mais fortes para CL-AKA podem ser enfraquecidos para possibilitarem melhor desempenho computacional, preservando-se importantes propriedades de segurança?
(Q.4) Existem ataques de segurança previstos em esquemas de cifragem ou assinatura que não foram estudados em CL-AKA? Em caso positivo, há como evitá-los?
(Q.5) Os protocolos mostrados seguros são viáveis na prática?

Não havia respostas completas registradas na literatura para as questões acima. Ao respondê-las ao longo do desenvolvimento da tese, produzimos resultados inéditos.

 ©2012 SBC — Soc. Bras. de Computação

### 1.3. Contribuições

Durante a busca por respostas às questões citadas, originamos os itens a seguir, que representam as principais contribuições deste trabalho:

- Levantamento da literatura sobre os modelos alternativos em criptografia de chave pública, respondendo a questão (Q.1) [Goya et al. 2009];
- Protocolos de acordo de chave com autenticação sem certificado, mostrados seguros em modelos de segurança variados, objetivando otimização em tempo computacional, tratando as questões (Q.2) e (Q.3) [Goya et al. 2010b], [Goya et al. 2010a], [Goya et al. 2012b];
- Análise de modelos de segurança para acordo de chave com autenticação sem certificado, extensão deles para elevação do nível de confiança que usuários podem depositar na autoridade geradora de chaves parciais, e proposta de protocolo, para as questões (Q.2) e (Q.4) [Goya et al. 2011];
- Fortalecimento de modelo de segurança para acordo de chave com autenticação sem certificado, visando adversário com poder mais realista e proposta de protocolo mais eficiente, para a questão (Q.3) [Goya et al. 2012b];
- Protocolo de acordo de chave com autenticação sem certificado seguro no modelo fortalecido e com extensão para caso da autoridade mal intencionada, refinando soluções para questões (Q.3) e (Q.4) [Goya et al. 2012a];
- Prova de conceito para estabelecimento de chave de sessão entre servidor e dispositivos móveis, no modelo sem certificado, respondendo afirmativamente a questão (Q.5) [Okida et al. 2012], [Goya et al. 2010a].

Apresentamos nas seções a seguir um resumo da tese, cuja versão integral encontra-se disponível em http://www.teses.usp.br/teses/disponiveis/45/45134/tde-02082012-191831.

## 2. Modelos Alternativos de Chave Pública

Os modelos alternativos de chave pública estudados são: baseado em identidade (*identity-based*), sem certificados (*certificateless*), autocertificado (*self-certified*) e baseado em certificado (*certificate-based*). Esses dois últimos são originalmente apresentados em [Girault 1991] e [Gentry 2003]. Na Tabela 1, algumas das propriedades são colocadas lado a lado para comparação. A segunda coluna refere-se ao modelo convencional sobre ICP. Em geral, um "sim" é considerado um ponto positivo; um "não", ponto negativo. A primeira propriedade, "Dispensa ICP", diz respeito à infraestrutura tradicional de gerenciamento de chaves públicas. Naturalmente, todos os modelos necessitam de algum tipo de infraestrutura, cujos requisitos e características são retratados pelas demais linhas da tabela; em todos os casos a autoridade central e os parâmetros do sistema são por premissa previamente autenticados. A característica "Dispensa certificados" é relacionada a certificados digitais de chave pública no sentido convencional, como no padrão X.509. A propriedade "Fluxo criptográfico" se refere ao uso da chave pública antes da emissão da chave secreta. A análise dos modelos alternativos leva-nos a concluir que as diferenças básicas entre eles decorrem do fato de ser permitido ao adversário corromper o registro da chave pública (não permitido no modelo convencional nem no baseado em identidade) e de um processo intrínseco de prova de posse de segredo (existente no convencional e baseado em certificado). As propriedades listadas na Tabela 1 são fundamentalmente decorrentes desses fatos.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Comparação entre os modelos de criptografia de chave pública**

| Propriedades | Modelos | | | | |
|---|---|---|---|---|---|
| | Com ICP | Baseado em Identidade | Auto-certificado | Sem Certificado | Baseado em Certificado |
| Dispensa ICP | não | sim | sim | sim | não |
| Dispensa certificados | não | sim | sim | sim | não |
| Dispensa diretório de chaves ou certificados | não | sim | não | não | não |
| Certificação explícita | sim | não | não | não | não |
| Nível de confiança | 3 | 1 | 2 | 2 | 3 |
| Sem custódia de chaves | sim | não | sim | sim | sim |
| Chave secreta criada integralmente pelo usuário | sim | não | sim | não | não |
| Irretratabilidade | sim | não | sim | sim | sim |
| Risco da chave mestra | alto | altíssimo | alto | alto | alto |
| Dispensa canal seguro para distribuir chaves | sim | não | sim | não | sim |
| Renovação de chaves controlada pelo usuário | não | não | não | sim | não |
| Fluxo criptográfico | não | sim | não | sim | não |
| Principais protocolos demonstrados seguros | sim | sim | não | sim | sim |

## 3. Modelos de Segurança para CL-AKA e Protocolos

Realizamos uma classificação dos modelos de segurança existentes para protocolos CL-AKA em função do poder que o adversário tem sob eles e propusemos algumas extensões. Um adversário dito *forte* possui os poderes de um adversário dito *moderado*, que, por sua vez, abrange os de um *fraco*. Citamos os modelos descritos na literatura:

- LBG: modelo contra adversário forte de [Lippold et al. 2009];
- SJ: modelo contra adversário moderado de [Swanson e Jao 2009];
- LG: modelo contra adversário fraco de [Lippold e González Nieto 2010].

As extensões que desenvolvemos foram as abaixo relacionadas, com base nos fundamentos em [Bellare e Rogaway 1994], no qual se captura a noção de autenticidade dos participantes perante ataques de adversários que têm total controle do canal de comunicação:

- $SJ^+$: melhorias no modelo SJ contra adversário moderado, respondendo a nossa questão (Q.3);
- Mal-LBG: extensão sobre LBG contra autoridade mal intencionada que gera parâmetros do sistema desonestamente, respondendo a questão (Q.4);
- Mal-$SJ^+$: extensão sobre $SJ^+$ contra autoridade mal intencionada.

Esses modelos se relacionam entre si, como se vê representado na Figura 1. Pode-se afirmar que o modelo LG é o mais fraco de todos, enquanto o modelo Mal-LBG é o mais forte. O modelo Mal-$SJ^+$ não é comparável a LBG, mas é mais fraco que Mal-LBG. Propusemos os protocolos listados na Figura 2 e demonstramos sua

 ©2012 SBC — Soc. Bras. de Computação

segurança. Quando o protocolo foi mostrado seguro em modelo já existente, o objetivo foi otimizar eficiência computacional de protocolos de outros autores e responder a questão (Q.5). O problema computacional Diffie-Hellman Bilinear (BDH) foi definido em [Boneh e Franklin 2003]; a família de problemas lacunares (Gap) foi apresentada em [Okamoto e Pointcheval 2001]. Um protocolo que tenha sido provado seguro sob a hipótese de dificuldade do problema BDH é, do ponto de vista da teoria da complexidade, pelo menos tão seguro quanto outro que tenha sido provado seguro sob Gap-BDH, no mesmo modelo de segurança. O problema Gap-BDH foi usado para oferecer opções de protocolos mais rápidos.

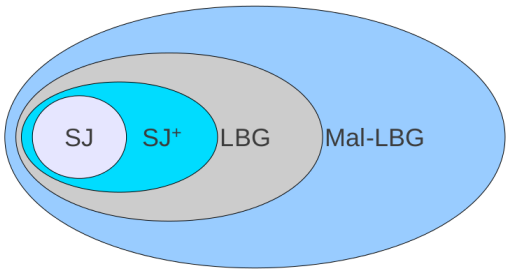



**Figura 1. Hierarquia dos modelos para CL-AKA**

| Protocolo | Modelo de segurança | Problema computacional | Poder do adversário |
|---|---|---|---|
| GNT2 | SJ+ | Gap-BDH | moderado |
| GNT4 | | BDH | |
| GOT1-Gap | LBG | Gap-BDH | forte |
| GOT1-BDH | | BDH | |
| GNT3 | | BDH | |
| GNT2 | Mal-SJ+ | Gap-BDH | autoridade mal intencionada |
| GNT1 | Mal-LBG | Gap-BDH | |
| GOT2 | LG | Gap-BDH | fraco |

**Figura 2. Nossos protocolos e sua segurança**

A análise dos modelos de segurança nos permitiu concluir que a diferença entre eles recai essencialmente sobre quais aspectos se quer assegurar para o registro dos usuários e de suas chaves. Quanto mais fortes são os modelos, mais podemos relaxar na confiança a ser depositada na infraestrutura de gerenciamento de chaves (e na autoridade de confiança). Quanto mais fraco for o modelo de adversário, mais nos aproximamos da prática comum em criptografia de chave pública: usar protocolos criptográficos mais simples que delegam complexidade para os procedimentos extra-criptográficos de gerenciamento da infraestrutura. Portanto, as aplicações de cada modelo e protocolo depende principalmente do quanto se deseja permitir que um adversário interfira no processo de registro das chaves, respondendo a questão (Q.2).

## 4. Considerações Finais

Apresentamos um estudo sobre diferentes paradigmas de criptografia de chave pública que dispensam a distribuição de certificados de chave pública. Exploramos modelos formais de segurança para acordo de chave com autenticação sem certificado e propusemos aprimoramentos para:

1. elevar o nível de confiança que usuários podem depositar na autoridade geradora de chaves secretas parciais e
2. viabilizar protocolos eficientes computacionalmente e com propriedades de segurança relevantes, como não violação de chaves de sessão mesmo que no futuro as chaves secretas venham a ser corrompidas (*forward secrecy*) e resistência a ataques de adversários que têm total controle do canal de comunicação e substituem chaves públicas de usuários por valores arbitrários.

 ©2012 SBC — Soc. Bras. de Computação

Mostramos que as melhorias efetuadas nos modelos de segurança são realizáveis, apresentando novos protocolos sob esses modelos, que foram mostrados seguros, sob a técnica de redução de problemas computacionais (segurança demonstrável). Também efetuamos otimizações sobre protocolos previamente existentes e, por meio de testes de desempenho sobre codificações dos protocolos, mostramos que eles podem ser adotados em aplicações reais. Algumas questões de pesquisa que ficaram por serem respondidas e são objetos de trabalhos futuros:

- Nossos protocolos LBG2-Gap e GOT1-Gap podem ser mostrados seguros contra ataque da autoridade mal intencionada?
- Como se comparam implementações de uma ICP com protocolos de acordo de chave com autenticação com os propostos?
- Em [Yang e Tan 2011] é proposto CL-AKA sem uso de emparelhamentos bilineares, no entanto é embutida a verificação de assinaturas digitais (no modelo sobre ICPs). É possível a remoção de assinaturas digitais nesse contexto?

Este trabalho recebeu apoio financeiro de Fapesp, sob processo n° 2008/06189-0.

## Referências

[Al-Riyami e Paterson 2003] Al-Riyami, S. S. e Paterson, K. G. (2003). Certificateless public key cryptography. In *ASIACRYPT 2003*, volume 2894 of *LNCS*. Springer.

[Bellare e Rogaway 1994] Bellare, M. e Rogaway, P. (1994). Entity authentication and key distribution. In *Proceedings of CRYPTO '93*, pages 232–249, New York, NY, USA. Springer-Verlag New York, Inc.

[Boneh e Franklin 2003] Boneh, D. e Franklin, M. (2003). Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615.

[Castro et al. 2007] Castro, R., Dahab, R. e Devegili, A. J. (2007). Introdução à segurança demonstrável. In *Minicursos SBSeg 2007 VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 103–152. SBC.

[Dent 2008] Dent, A. W. (2008). A survey of certificateless encryption schemes and security models. *Int. J. Inf. Secur.*, 7(5):349–377.

[Diffie e Hellman 1976] Diffie, W. e Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654.

[Gentry 2003] Gentry, C. (2003). Certificate-based encryption and the certificate revocation problem. In *EUROCRYPT'03*, pages 272–293, Berlin, Heidelberg. Springer-Verlag.

[Girault 1991] Girault, M. (1991). Self-certified public keys. In *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer.

[Goya et al. 2009] Goya, D., Misaghi, M., Rufino, V. e Terada, R. (2009). Modelos de criptografia de chave pública alternativos. In Santin, A., Nunes, R. C. e Dahab, R., editors, *Minicursos SBSeg 2009 IX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 49–97. SBC.

[Goya et al. 2011] Goya, D., Nakamura, D. e Terada, R. (2011). Acordo de chave seguro contra autoridade mal intencionada. In *SBSeg 2011 XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 265–278. Sociedade Brasileira de Computação.

 ©2012 SBC — Soc. Bras. de Computação

[Goya et al. 2012a] Goya, D., Nakamura, D. e Terada, R. (2012a). A certificateless authenticated key agreement protocol secure against malicious KGC. artigo submetido.

[Goya et al. 2012b] Goya, D., Nakamura, D. e Terada, R. (2012b). Certificateless key agreement protocols under strong and realistic models. artigo submetido.

[Goya et al. 2010a] Goya, D., Okida, C. e Terada, R. (2010a). Aplicação de acordo de chave com autenticação sem certificado digital. In *I2TS 2010, 9th International Information and Telecommunication Technologies Symposium*.

[Goya et al. 2010b] Goya, D., Okida, C. e Terada, R. (2010b). A two-party certificateless authenticated key agreement protocol. In *SBSeg 2010 X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação.

[Lippold et al. 2009] Lippold, G., Boyd, C. e González Nieto, J. (2009). Strongly secure certificateless key agreement. In *Pairing '09: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, volume 5671 of *Lecture Notes in Computer Science*, pages 206–230, Berlin, Heidelberg. Springer-Verlag.

[Lippold e González Nieto 2010] Lippold, G. e González Nieto, J. (2010). Certificateless key agreement in the standard model. In *Proceedings of the Eighth Australasian Conference on Information Security - Volume 105*, AISC '10, pages 75–85, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[Okamoto e Pointcheval 2001] Okamoto, T. e Pointcheval, D. (2001). The gap-problems: A new class of problems for the security of cryptographic schemes. In *PKC '01: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography*, pages 104–118, London, UK. Springer-Verlag.

[Okida et al. 2012] Okida, C., Goya, D. e Terada, R. (2012). Java cryptographic library for smartphones. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 10(1):1377 –1384.

[Shamir 1984] Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, volume 196/1985 of *LNCS*, pages 47–53, New York, NY, USA. Springer-Verlag New York, Inc.

[Swanson e Jao 2009] Swanson, C. e Jao, D. (2009). A study of two-party certificateless authenticated key-agreement protocols. In *INDOCRYPT '09: Proceedings of the 10th International Conference on Cryptology in India*, volume 5922 of *LNCS*, pages 57–71, Berlin, Heidelberg. Springer-Verlag.

[Yang e Tan 2011] Yang, G. e Tan, C.-H. (2011). Strongly secure certificateless key exchange without pairing. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 71–79, New York, NY, USA. ACM.

[Zhang et al. 2010] Zhang, L., Zhang, F., Wu, Q. e Domingo-Ferrer, J. (2010). Simulatable certificateless two-party authenticated key agreement protocol. *Inf. Sci.*, 180:1020–1030.

 ©2012 SBC — Soc. Bras. de Computação

# An Orchestration Approach for Unwanted Internet Traffic Identification

**Eduardo Luzeiro Feitosa[1], Djamel Sadok[2]**

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
69077-000 – Manaus – AM – Brasil

[2]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
50740-540 - Recife – PE - Brasil

efeitosa@icomp.ufam.edu.br, jamel@cin.ufpe.br

***Abstract.*** *Current Internet traffic shows a varying mix of relevant and unwanted traffic. The latter is increasingly becoming harmful to network performance and service availability, while often taking up scarce precious network and processing resources. This paper presents some definitions of definitions of unwanted traffic and an approach to identify unwanted traffic based on orchestration. There are basically two main contributions to our work. First, we show that multiple specialized security modules benefit from an integrated and coordinated flow process. We also present a solution that generates new rules and produces inferences with a greater degree of certainty than the uncertainty generated by existing anomaly detectors. As proof of concept, we implement a prototype showing the effectiveness and accuracy of our proposal.*

***Resumo.*** *O atual tráfego Internet apresenta uma gama variada de tráfego relevante e indesejado. Este último é cada vez mais prejudicial para o desempenho da rede e disponibilidade de serviços, muitas vezes ocupando a redes e o processamento de recursos. Este artigo apresenta algumas definições de definições de tráfego indesejado e uma abordagem para identificar o tráfego indesejado com base na orquestração. Existem basicamente duas contribuições principais nesse trabalho. Primeiro, mostrar que vários módulos especializados de segurança beneficiam-se de um integrado e coordenado processo de fluxo. Também apresenta uma solução que gera novas regras e produz inferências com um maior grau de certeza que a incerteza gerada pelos detectores de anomalias existentes. Como prova de conceito, um protótipo foi implementado para demonstrar a eficácia e precisão da proposta..*

## 1. Introdução

Uma breve análise do tráfego Internet comprova o crescente aumento no transporte do tráfego considerado desconhecido, não solicitado, improdutivo e muitas vezes ilegítimo, em outras palavras, tráfego não desejado. Originado através de atividades como, por exemplo, mensagens eletrônicas não solicitadas (spam); atividades fraudulentas como

 ©2012 SBC — Soc. Bras. de Computação

phishing[1] e pharming[2]; ataques de negação de serviço (do inglês *Distributed Denial of Service* - DDoS); proliferação de vírus e worms; backscatter[3], entre outros, o tráfego não desejado pode ser considerado uma pandemia cujas consequências refletem-se no crescimento dos prejuízos financeiros dos usuários da Internet.

Exemplos de perdas financeiras podem ser encontrados em todo o mundo. Em 2006, os prejuízos ocasionados por worms foram de aproximadamente US$ 245 milhões, somente entre provedores de acesso norte-americanos [Morin, 2006]. O instituto de segurança americano CSI (*Computer Security Institute*) [Richardson, 2007], depois de entrevistar 194 empresas nos Estados Unidos, contabilizou perdas superiores a US$ 66 milhões ocasionadas pelo tráfego não produtivo gerado principalmente por fraudes, vírus, worms, spyware e intrusões em 2007. No Brasil, o CERT.br (Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil) contabilizou o número de incidentes relacionados às tentativas de fraude em 45.298 em 2007 [CERT.br, 2008] enquanto que, no mesmo período, o CAIS (Centro de Atendimento a Incidentes de Segurança) da RNP (Rede Nacional de Pesquisa) registrou cerca de 4000 tentativas de fraudes através de spam e phishing.

Apesar dos diferentes esforços, isolados e/ou coordenados, o tráfego Internet não desejado continua a crescer. Primeiro, porque o tráfego indesejado, basicamente, representa uma ampla gama de aplicações de usuário, rede de dados e informações nocivas, com objetivos diferentes para a sua existência. Algumas delas podem ser incômodas, como mensagens de spam. Outras consistem em volumosos conteúdos multimídia a partir de tendências tecnológicas em aplicativos, redes e hábitos dos usuários, incluindo compartilhamento de arquivos P2P (por exemplo, Emule, Bit Torrent), de vídeo (por exemplo, Justin.tv, Joost, YouTube), e do tráfego recreativo (por exemplo, downloads de MP3, mensagens instantâneas, Skype, MSN). Por fim, encontra-se o trafego especialmente concebido para ser intrusivo e malicioso tais como worms, vírus e ataques de negação de serviço .

Em segundo lugar, a ineficiência das atuais soluções em identificar, reduzir e interromper o tráfego não desejado é notório, como descrito nos trabalhos recentes em [Anderson et al., 2007] [Krishnamurthy, 2006] [Davis, 2007]. Tipicamente, a efetividade fornecida pelas soluções existentes só é percebida após a ocorrência de algum dano. Além disso, a alta taxa de alarmes falsos e a falta de cooperação com outras soluções ou mesmo com a infraestrutura de rede são fatores considerados incentivadores do aumento do tráfego não desejado. Como mencionado em [Oliveira et al., 2007], as soluções usadas para detectar e reduzir os efeitos de ataques DDoS tais como filtragem, limitação de banda, IP *traceback* e esquemas de marcação de pacotes são difíceis de implementar porque necessitam de mudanças na infraestrutura da Internet. Ao mesmo tempo, soluções tradicionais como firewall e VPN (*Virtual Private Network*) são ineficazes contra códigos maliciosos e spam.

Por fim está a definição do que é tráfego não desejado. Uma vez que normalmente parte deste tipo de tráfego é o resultado de atividades recreativas como o

[1] *Phishing* é um tipo de fraude eletrônica caracterizada pela tentativa de obter informações pessoais privilegiadas através de sites falsos ou mensagens eletrônicas forjadas.
[2] *Pharming* refere-se ao ataque de envenenamento de cache DNS cujo objetivo é preparar terreno para atividades de *phishing*.
[3] *Backscatter* é o tráfego recebido de vítimas que estão respondendo a ataques de negação de serviço.

 ©2012 SBC — Soc. Bras. de Computação

download de músicas e vídeos e jogos on-line. Como exemplos de aplicações utilizadas em atividades recreativas podem-se citar: Emule, BitTorrent e Kazaa na categoria de aplicações P2P; Skype, MSN e Google Talk como aplicações utilizadas na comunicação instantânea; e Joost e Justin.TV como aplicações de TV via Internet (em tempo real). Muitas das soluções existentes não reconhecem ou são configuradas para não detectar o tráfego gerado por essas atividades.

Em suma, esta tese é motivada pela situação alarmante na qual o tráfego indesejado chegou atualmente. Torna-se claro que este tipo de tráfego é um dos principais problemas de segurança e que precisa urgente de identificação e mitigação, embora ainda não é trivial como fazê-lo. Questões como tipos, origens e objetivos precisam ser cuidadosamente estudadas e respondidas para que ações efetivas podem ser tomadas para atenuar o efeito do tráfego indesejado.

Esta tese compartilha da opinião de que:

- Limitar as defesas apenas para mecanismos periféricos é ineficiente e fornece uma pequena visão de um cenário bem amplo.
- A utilização de sistemas altamente especializados para combater os problemas específicos podem trazer poucos e ineficientes benefícios no tratamento do tráfego indesejado.
- Soluções distribuídos e colaborativos são primordiais para se obter uma identificação precoce e a filtragem do tráfego suspeito, garantindo assim danos limitados para esse tipo de tráfego.

## 1.1 Objetivos

Atualmente, as estratégias para lidar com o tráfego indesejado baseiam-se em três etapas: (i) ganhar conhecimento sobre os diferentes tipos e fontes de tráfego indesejado, (ii) avaliar seu impacto e a eficácia das soluções existentes contra ele, e (iii) desenvolver e testar novas contramedidas eficazes contra o tráfego indesejado.

O objetivo principal desta tese é realizar estes passos com foco no estudo, definição e descrição do problema de identificação do tráfego Internet não desejado e, consequentemente, apresentar uma solução que é capaz de identificar com precisão este tráfego.

De modo geral, esta tese apresenta:

1. Um estudo sobre o universo do tráfego Internet não desejado, apresentado definições, discussões sobre contexto e classificação e uma série de existentes e potencias soluções.

2. Uma metodologia para identificar tráfego não desejado baseada em orquestração. OADS (*Orchestration Anomaly Detection System*) é uma plataforma única para a identificação de tráfego não desejado que permite um gerenciamento cooperativa e integrado de métodos, ferramentas e soluções voltadas a identificação de tráfego não desejado.

3. O projeto e implementação de soluções modulares integráveis a metodologia proposta. A primeira delas é um sistema de suporte a recuperação de informações na Web (WIRSS), chamado *OADS Miner* ou simplesmente ARAPONGA, cuja função é reunir informações de segurança sobre

 ©2012 SBC — Soc. Bras. de Computação

vulnerabilidades, ataques, intrusões e anomalias de tráfego disponíveis na Web, indexá-las eficientemente e fornecer uma máquina de busca focada neste tipo de informação. A segunda, chamada *Alert Pre-Processor*, é um esquema que utilize uma técnica de cluster para receber múltiplas fontes de alertas, agregá-los e extrair aqueles mais relevantes, permitindo correlações e possivelmente a percepção das estratégias usadas em ataques. A terceira e última é um mecanismo de correlação e fusão de alertas, *FER Analyzer*, que utilize a técnica de descoberta de episódios frequentes (FED) para encontrar sequências de alertas usadas para confirmar ataques e possivelmente predizer futuros eventos.

De modo a avaliar a proposta e suas implementações, uma série de experimentos foram conduzidos com o objetivo de comprovar a eficácia e precisão das soluções.

## 2. Orchestration oriented Anomaly Detection System

Os serviços Internet se basearam, por um longo tempo, em acordos informais e na boa vontade das partes em proteger e encaminhar corretamente os tráfegos. Apesar da falta de um controle centralizado ou proprietário, a Internet é um dos poucas, senão a única, infraestrutura autogovernada que consegue operar razoavelmente bem sob tal paradigma.

Hoje, esse modelo de confiança está sob ataque intenso. Pode este "modo de vida" ser mantido? A que preço? E o que pode ser feito para tornar o trabalho dos administradores de rede mais fácil ao lidar com tais problemas? Não se pode simplesmente barrar os domínios onde o tráfego indesejado vem, uma vez que isso beneficiaria apenas aqueles que estão explorando as fraquezas existentes na Internet.

Na tentativa de responder a essas preocupações, foi feita a especificação de uma nova abordagem baseada em orquestração para detectar e, na medida do possível, limitar o tráfego (indesejado) anômalo. O núcleo dessa proposta é uma estrutura que coordena o recebimento de múltiplos alertas e eventos oriundo de vários detectores, avalia esta entrada para detectar ou provar a existência de anomalias e, consequentemente, escolhe a melhor ação a ser tomada. É abordagem é chamada de OADS (*Orchestration-oriented Anomaly Detection System*).

### 2.1. Arquitetura OADS

Para melhor esclarecer a abordagem proposta, a Figura 1 ilustra sua organização e mostra seus componentes.

A arquitetura OADS consiste de quarto (04) elementos básicos:

- **Alert Pre-Processor**: Tem a função de receber informações (alertas) dos detectores de anomalias e prepara-los para análise. Basicamente, ele realiza duas atividades. A primeiro é a adequação dos alertas para o formato padrão IDMEF (*Intrusion Detection Message Format*). A segunda é agregar os múltiplos alertas de entrada dos vários detectores, tornando mais fácil construir hipóteses sobre as anomalias e estratégias de defesa possíveis, além de reduzir o volume de dados. Este componente emprega uma técnica de cluster proposta por [Xu et al., 2005] para criar perfis de tráfego no backbone Internet e descobrir padrões de comportamento significativos de interesse.

 ©2012 SBC — Soc. Bras. de Computação

Entropia, uma abordagem de informação teórica, é usada para classificar o tráfego em clusters significativos e, consequentemente, para medir a quantidade de incerteza relativa (RU) contido em cada um dos dados.

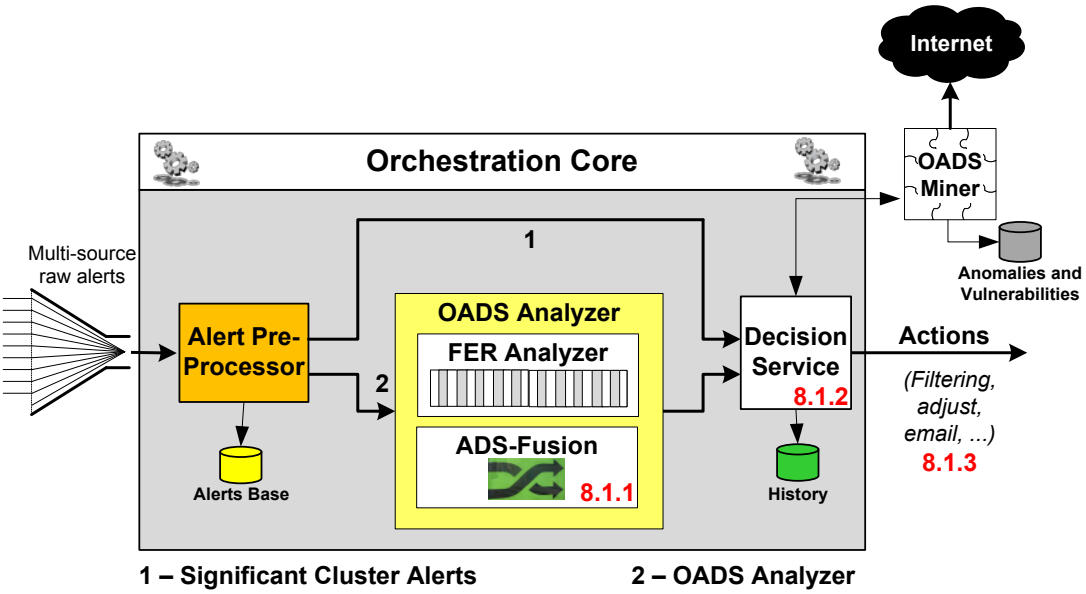



**Figura 1. Visão da Arquitetura OADS.**

- **OADS Analyzer:** Correlaciona os alertas recebidos, tentando confirmar a existência ou não de ataques e anomalias. Além disso, é também capaz de prever futuras ameaças e alvos. Em linhas gerais, recebe alertas agregados construídos pelo Alert Pre-Processor, correlaciona-os para aumentar a sua precisão e, consequentemente, valida os afirmações contidos em cada um deles e, possivelmente, até prevê a sua ocorrência no futuro com algum nível de confiança. A fim de apoiar as funcionalidades do OADS, foram desenvolvidos dois módulos distintos: um para lidar com problemas de correlação de alerta (Analyzer FER) e outro para lidar com a incerteza ou imprecisão dos resultados de detecção de anomalias (ADS-Fusion). O primeiro usa o conceito de Regras de Episódios Frequentes (FER) para realizar análises de sequência e, consequentemente, detectar anomalias, incluindo padrões de ataque também desconhecidos. O segundo é baseado na teoria da evidência de Dempster-Shafer para lidar com incertezas e imprecisões nos resultados da análise de detecção de anomalias.

- **Decision Service**: É responsável pelo processo de tomada de decisão relacionada ao tráfego analisado. De acordo com a informação recebida (ou coletada), decide se existe ou não anomalias e o que fazer. Recebe duas entradas padrão. A primeira são conjuntos de alertas reduzidos (agregados e significantes) enviados pelo Alert Pre-Processor. O segundo são regras enviadas pelo OADS Analyzer. Desta forma, o Decision Service adota duas estratégias diferentes. Os alertas reduzidos alimentam regras de básicas firewall enquanto as regras alimentar uma máquina de estado finito simples, que correlaciona a informação processada pelos diferentes analisadores (OADS Analyzer) a diferentes possíveis estados que o Decision Service poderia tomar.

 ©2012 SBC — Soc. Bras. de Computação

- **OADS Miner**: É um sistema de busca e recuperação de informações baseada nos conceitos de WIRSS (Sistemas de Suporte a Recuperação de Informação na Web) e ISSS (Sistemas de Suporte a Busca de Informação). É capaz de fornecer: (i) integração de dados Web em um único lugar, (ii) um motor poderoso de suporte a busca focada em segurança, (iii) acesso simples e unificado com suporte para expressões lógicas, e (iv) interfaces para lidar com os usuários humanos e com sistemas externos, como motores de busca e ferramentas de tomada de decisões.

Além desses componentes, OADS usa detectores de anomalia como elementos de entrada externos, ou seja, que geram os alertas.

### 2.2. Heurística para Orquestração

Este trabalho utiliza o conceito de orquestração para gerenciar automaticamente entradas de diferentes detectores de anomalias, harmoniza-las e, consequentemente, ser capaz de tomar uma decisão bem informada, correta e eficiente no que diz respeito à existência de anomalias de tráfego.

Formalmente, a orquestração se refere a um processo de negócio executável que pode interagir com sistemas complexos, middleware e serviços. Atualmente, a orquestração é relacionada principalmente à conexão de serviços da Web de forma colaborativa. Orquestração estabelece a sequência de passos dentro de um processo, incluindo as condições e exceções, e cria um controlador central para implementar a sequência.

Apesar de OADS poder adoptar Web Services como seu mecanismo de coordenação, foi utilizada uma heurística simples, parametrizada e eficaz (representado como um algoritmo) para orquestração. A heurística é baseada em regras comuns e conhecimentos adquiridos que regem a forma como a informação recebida (alertas e regras) deve ser tratada pelo núcleo do OADS. Apesar de sua simplicidade, este módulo inteligente obtém bons resultados, como mostrado na últimas sessões deste artigo.

Os detalhes da heurística de orquestração são descritos no Algoritmo abaixo.

**Algorithm 1** Simplified algorithm for orchestration

**Step 0:** Initialization
  **R**ead alerts every x time
  ***While*** TRUE
    **Step 1:** *Alert Pre-Processor* receive all multi-source alerts
      All received alert are prepared for possible extraction of significant cluster process.
    **Step 2:** Significant cluster extraction
      Execute significant cluster extraction process.
      All alerts classified as significant are send to the *Decision Service* (**Step 6**).
    **Step 3:** ***If*** there are alerts to evaluate, go to **step 3a**, otherwise go to **Step 6**
      **Step 3a:**
        ***If*** number of remain alerts > *fer_threshold*, go to **step 4**;
        ***Otherwise*** go to **step 5**
    **Step 4:** FER Analyzer examines received alerts
      ***For*** all alerts

 ©2012 SBC — Soc. Bras. de Computação

Calculate Frequent Episodes.
Generate Rule Episodes.
***End***
Send all rules to *Decision Service* (**Step 6**).
**Step 5:** ADS-Fusion examines received alerts
***If*** alerts have two or more sources, go to **step 5a**, otherwise go to **Step 6**
**Step 5a:**
Execute Dempster-Shafer analysis in alerts.
Send all inferences (rules) to *Decision Service* (**Step 6**)
**Step 6:** Decision Service receives alerts or analysis results
Evaluate the received information
***If*** necessary use OADS Miner to discover extra information
Make decisions
***End***

## 3. Conclusões

A principal contribuição desta tese é permitir que operadores de rede e gerentes de TI minimizem o tempo gasto na avaliação de eventos de segurança, alertas e incidentes fornecidos por diferentes software de segurança de rede e serviços. Para este objetivo, uma abordagem genérica, aberta e flexível para lidar com o tráfego Internet não desejado foi projetada e implementada. Genérica, pois seus componentes podem ser organizados para detectar uma gama ampla e variada de invasões, ataques e anomalias, abrangendo desde o tradicional serviço de controle de acesso de rede até backbones de alta velocidade. A abertura decorre do emprego de linguagens e ferramentas de código aberto e protocolos padrão. É flexível na medida em que permite a atualização ou inserção contínua de novas técnicas de detecção.

A abordagem desenvolvida é uma visão holística do conceito de orquestração aplicado a intrusão e detecção de anomalias. OADS oferece suporte à colaboração e harmonização de diferentes detectores, aumentando o poder de percepção (detecção) de anomalias e, consequentemente, tornando a rede mais segura. Em termos práticos, OADS é uma estrutura completa e unificada para a identificação de tráfego indesejado capaz de lidar com diferentes detectores e sua infinidade de alertas, empregando diferentes técnicas e métodos de análise para confirmar ou negar a presença de invasões, ataques e anomalias e indicar algum tipo de ação de execução.

### 3.1. Sumário das Contribuições

Nesta Tese, foi defendido que a colaboração e harmonização entre diferentes detectores de anomalias é um bom passo para alcançar um resultado desejado na área de segurança. Motivado por este ponto de vista, projetou-se e desenvolveu-se um arcabouço completo e unificado para lidar com o tráfego indesejado da Internet. Em seguida, aplicou-se este arcabouço em diferentes cenários, com o objetivo de validá-lo e, consequentemente, apontar onde e como ele poderia ser melhorado.

Em particular, esta tese fez as seguintes contribuições:

- **Survey sobre tráfego Internet não desejado**: A fim de melhor compreender as questões que envolvem este tráfego, foi feita uma ampla revisão do assunto. Foram levantadas diferentes definições sobre tráfego indesejado,

 ©2012 SBC — Soc. Bras. de Computação

além da formulação de uma nova definição que inclui aspectos como a caracterização do tráfego legítimo e ilegítimo como indesejável. Também se discutiu o contexto, classificações e as possíveis razões para explicar o seu aumento nos últimos anos. O resultado deste survey foi publicado em 2008, como minicurso, no VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'08).

- **Nova Abordagem para identificação de Tráfego Internet não Desejado**: A segunda contribuição desta tese foi a proposta de uma nova abordagem baseada na orquestração de diferentes componentes de segurança para identificação tráfego indesejado. Esta nova abordagem foi publicada este ano no periódico Computer Networks.
- **OADS Miner (ARAPONGA)**: ARAPONGA é uma ferramenta capaz de reunir informações de segurança sobre relatórios de vulnerabilidades, eventos e estatísticas do tráfego Internet, consolida-las e indexá-las em um único lugar. O resultado é um prática, simples e útil fonte de informação rápida para consulta dos usuários. Araponga aplica os conceitos de WIRSS e ISSS para fornecer funcionalidades de suporte que vão além de um sistema tradicional de busca e indexação. Além disso, usa técnicas de mineração de dados e modelos para expandir a capacidade de indexação de informações sobre segurança e, portanto, permitir consultas diferentes e mais focada. O ARAPONGA teve seus resultados publicados X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'10) e no WWW/Internet, em 2011. Além disso, rendeu uma dissertação de Mestrado na UFPE em 2012.
- **OADS Alert Pre-Processor:** A fim de lidar com múltiplos alertas de detectores diferentes, outra contribuição foi uma solução para agregação e extração de alertas significativos. Com base na abordagem de agrupamento, Alert Pre-Processor recebe múltiplos alertas de múltiplas fontes, os agrega e extrai os mais relevantes, além de reduzir a carga computacional no servidor (centralizado), diminuir a taxa de falsos negativos e também poder ser utilizado no aprendizado sobre estratégias de ataque. O Alert Pre-Processor teve seus resultados publicados no XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'11), além uma dissertação de Mestrado na UFPE em 2011. Atualmente, um journal está em submissão.
- **OADS Analyzer**: A quinta contribuição da tese são módulos capazes de correlacionar alertas, descobrir sequências de eventos que representam estratégias ou fases de ataques e permitindo a previsão de alertas futuros. Para tanto foi utilizada a técnica de episódios frequentes e a teoria da incerteza de Dempster-Shafer. Os resultados foram no XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'09) e no X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'10). Além disso, renderam dois trabalhos finais de graduação e uma dissertação de Mestrado na UFPE em 2012. Atualmente, um journal está em submissão.

 ©2012 SBC — Soc. Bras. de Computação

### 3.2. Publicações

Todos os resultados apresentados na tese foram publicados em anais de conferências nacionais e internacionais e em periódicos internacionais. A lista de publicações é mostrado a seguir:

- L. E. Oliveira, R. Aschoff, B. Lins, E. L. Feitosa, D. Sadok. Avaliação de Proteção contra Ataques de Negação de Serviço Distribuídos (DDoS) utilizando Lista de IPs Confiáveis. Anais do VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2007). Rio de Janeiro, Brasil: SBC, 2007.
- E. L. Feitosa, L. E. Oliveira, B. Lins, A. Carvalho Junior, R. Amorim, D. Sadok, U. Carmo. "Security Information Architecture for Automation and Control Networks". Anais do VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'08). Porto Alegre, Brasil: SBC, pp. 17-30, 2008.
- E. L. Feitosa, E. Souto, and D. Sadok, "Tráfego Internet não Desejado: Conceitos, Caracterização e Soluções,". Livro texto dos Minicursos do VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'08). Porto Alegre, Brasil: SBC, 2008, ch. 3.
- D. Sadok, E. J. Souto, E. L. Feitosa, J. Kelner, L. Westberg. "RIP - A Robust IP Access Architecture". Computers & Security, vol. 38, pp. 359-380, 2009.
- B. Lins, E. L. Feitosa, D. Sadok. "Aplicando a Teoria da Evidência na Detecção de Anomalias". Anais do XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'09). Recife, Brasil: SBC, pp. 583-596, 2009.
- T. G. Rodrigues, E. L. Feitosa, D. Sadok, J. Kelner. "Uma Ferramenta de Suporte a Recuperação de Informação na Web focada em Vulnerabilidades e Anomalias Internet". Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG'10). Porto Alegre, Brasil: SBC, pp. 227-240, 2010.
- L. Vilaça, E. L. Feitosa, D. Sadok, J. Kelner. "Aplicando a Frequência de Episódios na Correlação de Alertas". Anais do X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG'10). Porto Alegre, Brasil: SBC, pp. 241-254, 2010.
- CARMO, U. ; FEITOSA, E. L. ; SADOK, D. Security Architecture for Substation Automation System and Control Center. In: 7th International Conference on Electrical and Electronics Engineering Research, 2010, Aguascalientes. 7th International Conference on Electrical and Electronics Engineering Research, 2010.
- FEITOSA, E. L. ; Rodrigues, T. G. ; KELNER, J. ; SADOK, D. . ARAPONGA: A WIRSS for Internet Anomalies and Vulnerabilities. In: IADIS International Conference on WWW/Internet, 2011, Rio de Janeiro. IADIS International Conference on WWW/Internet. Lisboa : IADIS, 2011. v. 1. p. 396-403.

 ©2012 SBC — Soc. Bras. de Computação

- LINS, B. ; FEITOSA, E. L. ; SADOK, D. . Uma Ferramenta de Agregação e Extração de Alertas para Soluções Colaborativas. In: XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2011), 2011, Campo Grande. Porto Alegre : SBC, 2011. v. 1. p. 469-483.
- Feitosa, Eduardo; Souto, Eduardo; Sadok, Djamel H. An Orchestration Approach for Unwanted Internet Traffic Identification. Computer Networks (1999), v. 56, p. 2805-2831, 2012.

## Referências

Anderson, L., Davies, E. and Zhang, L. (2007) "Report from the IAB workshop on Unwanted Traffic", IETF, Internet informational RFC 4948.

CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. (2008) "Estatísticas do CERT.br", http://www.cert.br/stats/incidentes/.

Davies, E. (2007) "Unwanted Traffic", IETF Journal, December.

Dyer, S., Martin, J. and Zulauf, J. (1995) "Motion Capture White Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.

Krishnamurthy, B. (2006) "Unwanted traffic: Important problems, research approaches", Internet Architecture Board Workshop.

Morin, M. (2006) "The Financial Impact of Attack Traffic on Broadband Networks", IEC Annual Review of Broadband Communications, p. 11–14.

Oliveira, E. L., Aschoff, R., Lins, B., Feitosa, E., Sadok, D. (2007) "Avaliação de Proteção contra Ataques de Negação de Serviço Distribuídos (DDoS) utilizando Lista de IPs Confiáveis", VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (VII SBSEG). Rio de Janeiro, Brasil.

Richardson, R. (2007) "CSI/FBI Computer Crime Survey." 12th Annual Computer Crime and Security, p. 1-30.

Xu, K., Zhang, Z. and Bhattacharyya, S. (2005) "Profiling internet backbone traffic: behavior models and applications", 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '05), Philadelphia, Pennsylvania, USA, p. 169-180.

 ©2012 SBC — Soc. Bras. de Computação

# Análise de Políticas de Controle de Acesso Baseado em Papéis com Rede de Petri Colorida

**Eduardo Takeo Ueda**[1], **Wilson Vicente Ruggiero**[1,2]

[1]Escola Politécnica da Universidade de São Paulo
Laboratório de Arquitetura e Redes de Computadores
São Paulo – SP – Brazil

[2]Scopus Tecnologia S.A.
São Paulo – SP – Brazil

edutakeo@larc.usp.br, wilson@larc.usp.br

***Abstract.*** *Access control is an important research topic both for academia and industry. Role Based Access Control (RBAC) was developed in the early 1990s, becoming a generalized standard of access control for many products and computing solutions. Although RBAC models have been widely accepted and adopted, there are issues to answer. One of the key challenges for role-based security research is to characterize whether an access control policy is consistent in a highly dynamic environment. Our research aims filling this gap providing a method to analyze RBAC policies with respect to two significant aspects: security and dynamics involving roles and objects. For this purpose, we developed a policy description and simulation model using colored Petri net and the CPN Tools. The model describes and is able to simulate many RBAC states in a typical online education context. Using this model it was possible to analyze the state space provided by colored Petri net that simulates a dynamic environment and the creation of new roles, actions and objects. The result of the reachability analysis of Petri net policy showed that it is possible to check the consistency of access control policies considering dynamic of roles, actions and objects, and point out the advantages and applicability of modeling security policies in distributed environments using colored Petri net.*

***Resumo.*** *Controle de acesso é um tópico de pesquisa importante tanto para a academia quanto para a indústria. Controle de Acesso Baseado em Papéis (CABP) foi desenvolvido no início dos anos* 1990*, tornando-se um padrão generalizado para controle de acesso em vários produtos e soluções computacionais. Embora modelos CABP sejam largamente aceitos e adotados, ainda existem questões para responder. Um dos principais desafios de pesquisa em segurança baseada em papéis é determinar se uma política de controle de acesso é consistente em um ambiente altamente dinâmico. Nossa pesquisa visa preencher essa lacuna fornecendo um método para analisar políticas CABP com respeito a dois aspectos significativos: segurança e dinamismo envolvendo papéis, ações e objetos. Para este propósito, desenvolvemos um modelo de descrição e simulação de política usando rede de Petri colorida e CPN Tools. O modelo descreve e é capaz de simular vários estados CABP em um contexto de educação online típico. Usando este modelo, foi possível analisar o espaço de estados produzido*

 ©2012 SBC — Soc. Bras. de Computação

*pela rede de Petri colorida em um cenário dinâmico envolvendo a criação de novos papéis, ações e objetos. O resultado da análise de alcançabilidade da rede de Petri da política demonstrou que é possível verificar a consistência de políticas de controle de acesso considerando a dinamicidade de papéis, ações e objetos, e apontou vantagens de aplicabilidade da modelagem de políticas de segurança em ambientes distribuídos utilizando rede de Petri colorida.*

## 1. Introdução

Controle de acesso é um tema de pesquisa importante dentro do contexto de segurança de sistemas, pois é um dos componentes de segurança fundamentais em qualquer sistema de computação. No início da década de 90, o modelo de Controle de Acesso Baseado em Papéis (*Role-Based Access Control – RBAC*) começou a ganhar destaque, devido, principalmente, à sua flexibilidade e facilidade de gerenciamento [Ferraiolo e Kuhn 1992].

Um dos maiores desafios em grandes sistemas de informação é a complexidade de administração de segurança. O modelo RBAC tem contribuído para reduzir esse problema, principalmente diminuindo custos. Hoje em dia, a maioria das empresas fornecedoras de tecnologia da informação tem incorporado RBAC em suas soluções e produtos. Assim, é possível encontrar implementações de modelos RBAC em áreas que vão desde saúde e medicina até defesa militar.

Com a enorme aplicação do modelo RBAC, muitas pesquisas foram direcionadas para estudar maneiras de verificar a consistência desta política de controle de acesso. Verificar a consistência de uma política de segurança corresponde a analisar a aderência das regras desta política ao cenário que ela é aplicada. Com isso, uma inconsistência na política é representada por uma violação de uma regra [Cholvy e Cuppens 1997, Ribeiro et al. 2000, Changhua et al. 2009].

Uma iniciativa relativamente recente para análise de consistência de políticas de segurança diz respeito à modelagem de RBAC usando rede de Petri colorida [Kahloul et al. 2010]. Por meio desse tipo de abordagem é possível simular o comportamento das relações de uma especificação RBAC e analisar o espaço de estados da rede para tentar descobrir se existe a possibilidade do sistema assumir algum estado considerado inseguro.

O principal objetivo da nossa pesquisa é fornecer um método que demonstre a aplicabilidade da modelagem e análise de políticas de segurança baseada em papéis, a partir de um modelo de dados, utilizando rede de Petri colorida, considerando o aspecto de dinamicidade de papéis, ações e objetos. Visando atingir tal objetivo geral, devemos considerar a formulação de objetivos específicos, que são explicados a seguir.

**Modelagem e Simulação.** Mostrar a eficácia de modelagem e simulação de um sistema que adota uma política de controle de acesso baseada em papéis com rede de Petri colorida. Desenvolver o modelo a partir do modelo de dados, ou seja, elaborar um metamodelo. Ressaltar a facilidade de incrementar o modelo quando surge um novo papel, ação ou objeto, devido às características expansíveis que apresenta uma rede de Petri colorida.

**Análise de Consistência.** Desenvolver um procedimento de análise de consistência da política de segurança do metamodelo com rede de Petri colorida. Em tal análise

 ©2012 SBC — Soc. Bras. de Computação

deve ser possível avaliar a segurança da política de controle de acesso, bem como verificar o efeito da dinamicidade do sistema caracterizado pelo surgimento de um novo papel, ação ou objeto, que afeta o ciclo de vida da política adotada.

A abordagem com rede de Petri colorida que adotamos para modelar, simular e analisar políticas de controle de acesso baseado papéis se destaca pelo fato de possibilitar a análise determinística que foi conduzida, porém uma rede de Petri colorida também permite a realização de análises probabilísticas.

## 2. Controle de Acesso Baseado em Papéis

A Figura 1 apresenta os diversos elementos do modelo RBAC, incluindo usuários, papéis, permissões, restrições e sessões. Um usuário pode ter várias sessões abertas e cada sessão pode ter vários papéis ativados. Um papel pode estar relacionado a muitas permissões e uma mesma permissão pode estar relacionada a muitos papéis. Um usuário pode estar relacionado a muitos papéis e um papel pode estar relacionado a vários usuários. Pode haver ainda uma relação de herança entre múltiplos papéis. Por fim, as restrições podem ser aplicadas a diversas partes do modelo.

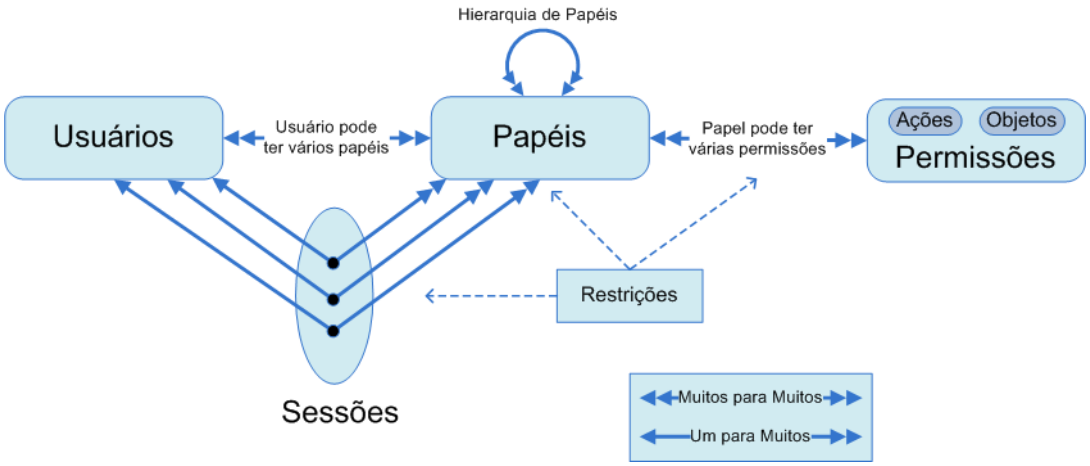



**Figura 1. Elementos do Modelo RBAC**

A flexibilidade do RBAC reside no fato de que o gerenciamento de permissões não precisa mais ser individualizado. Ou seja, quando um usuário deixa de ser responsável por uma função no sistema, basta desassociá-lo do papel. Caso seja necessário adicionar uma permissão ao papel, basta conceder o direito ao papel, ao invés de ter que conceder a permissão individualmente para todos os participantes que estão associados ao papel.

Um outro aspecto importante do Controle de Acesso Baseado em Papéis (CABP) é que ele permite a implementação do **princípio do privilégio mínimo** – um usuário só deverá ter as permissões que precisa para realizar sua tarefa.

## 3. Rede de Petri Colorida

A estrutura de uma rede de Petri [Petri 1962] pode ser definida formalmente como uma quádrupla $(P, T, F, W)$, sendo que:

- $P = \{p_1, p_2, \ldots, p_m\}$ é o conjunto de lugares, no qual $m \in \mathbb{N}$ é a quantidade total de lugares da rede de Petri;

 ©2012 SBC — Soc. Bras. de Computação

- $T = \{t_1, t_2, \ldots, t_n\}$ é o conjunto transições, no qual $n \in \mathbb{N}$ é a quantidade total de transições da rede de Petri;
- $F \subseteq ((P \times T) \cup (T \times P))$. No qual $F$ é o conjunto de arcos da rede. Os arcos podem relacionar lugares a transições ou vice-versa. $(P \times T)$ é o conjunto de arcos de entrada das transições. $(T \times P)$ é o conjunto de arcos de saída das transições;
- $W \subseteq ((P \times T) \cup (T \times P)) \mapsto \mathbb{N}$. No qual $W$ é uma função que indica o peso de cada arco possível da rede de Petri.

Uma rede de Petri colorida (RPC) é uma classe de rede de Petri que adiciona recursos para definição e manipulação de tipos de *tokens* [Jensen e Kristensen 2009]. Convencionou-se designar os tipos em uma RPC como **"cores"**, em contraste com as redes de Petri originalmente propostas, nas quais todos os *tokens* (marcas) são iguais, ou seja, sem cor. A Figura 2 apresenta em forma gráfica o que acabamos de explicar, circunferências ou elipses representam lugares, retângulos simbolizam transições e linhas com setas nas extremidades denotam arcos.

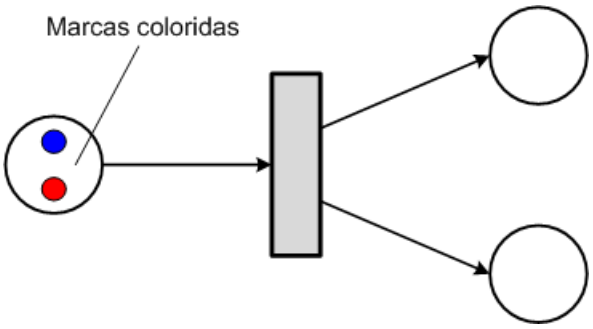



**Figura 2. Rede de Petri colorida**

## 4. Modelagem de uma Política de Controle de Acesso em um Ambiente Educacional

Depois de descrever os detalhes de um cenário educacional e uma política de controle de acesso baseado em papéis aplicada a esse ambiente, foi possível desenvolver a modelagem de um *workflow* operacional utilizando rede de Petri colorida. A Figura 3 mostra a representação do módulo principal do cenário educacional modelado com a ferramenta CPN Tools [Jensen e Kristensen 2009].

Por questões de simplificação do modelo, foram considerados como usuários no sistema apenas 2 professores e 4 alunos, sendo que um destes alunos também é monitor. Para representar um papel na RPC foi definida uma **cor** do tipo enumerável que pode assumir os valores: Professor, Aluno, Monitor.

As regras da política de controle de acesso são representadas nas **inscrições** presentes nos arcos e nas **funções de guarda** das transições. Devido a restrição de espaço desse artigo não é possível apresentar uma descrição detalhada de todos os módulos do *workflow* que foram desenvolvidos com a ferramenta CPN Tools.

O processo de modelagem de um cenário educacional genérico a partir do seu modelo de dados nos permitiu desenvolver regras de mapeamento que podem ser aplicadas em qualquer tipo de ambiente, independente de ser educacional. Essas regras caracterizam um metamodelo e são apresentadas na Tabela 1.

 ©2012 SBC — Soc. Bras. de Computação

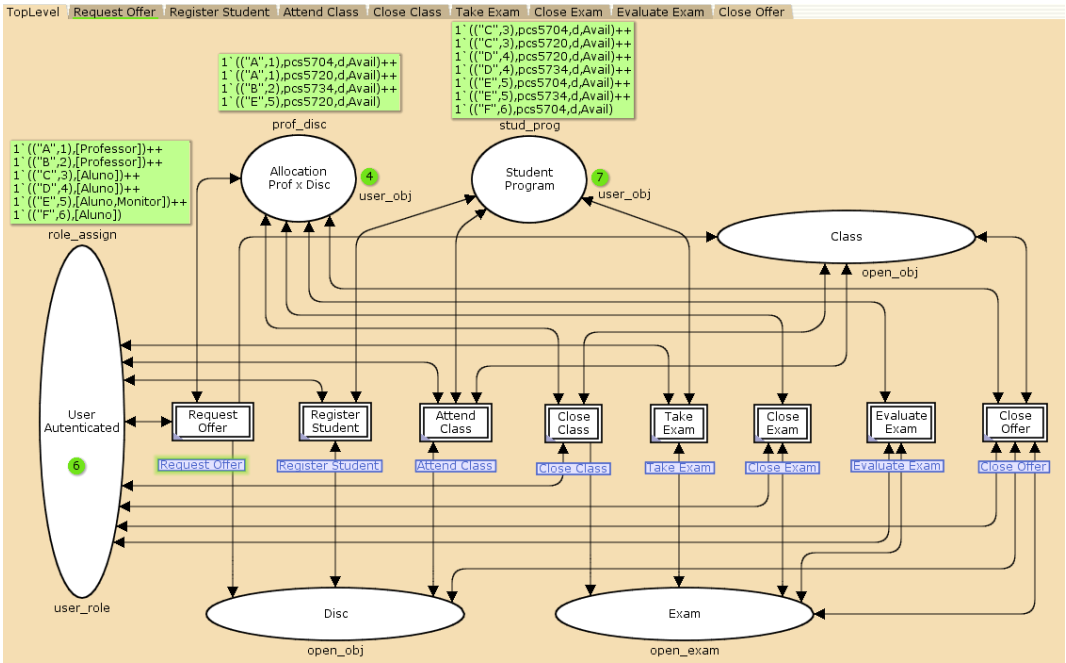



**Figura 3. Workflow do Cenário Educacional**

Dessa forma, através da modelagem de um caso de estudo foi possível desenvolver um metamodelo que pode ser aplicado para modelar qualquer tipo de ambiente, indicando que os resultados de nossa pesquisa não se aplicam apenas a ambientes educacionais.

**Tabela 1. Metamodelo**

| Entidades | Representação |
|---|---|
| Usuários | tokens |
| Papéis | tokens |
| Objetos | lugares |
| Ações | transições |
| Permissões | transições e lugares |
| Sessões | tokens |
| Restrições | funções de guarda e expressões em arcos |
| Estados | atributos |
| Relacionamentos | lugares |

## 5. Análise de Consistência da Política

Em uma rede de Petri colorida o espaço de estados é representado por um grafo de alcançabilidade que descreve o comportamento dinâmico da rede. Assim, é necessário ter à disposição mecanismos que permitam representar e manipular eficientemente este espaço de estados que denota todos os estados possíveis de um sistema modelado com rede de Petri colorida [Hadjidj e Boucheneb 2011].

Com a ferramenta CPN Tools é possível realizar a análise de alcançabilidade no espaço de estados de uma rede de Petri colorida [Jensen e Kristensen 2009]. Dessa forma,

 ©2012 SBC — Soc. Bras. de Computação

é possível fazer consultas descritas em linguagem de programação funcional ML para determinar se estados específicos considerados inconsistentes podem ocorrer.

Para ilustramos a verificação de consistência de uma regra da política incorporada ao *workflow* do cenário educacional consideremos uma proposição que estabelece que *"Um usuário **não pode** ser professor e aluno de uma mesma disciplina"*.

A violação da proposição que acabamos de enunciar acontece apenas se no lugar *Disc* da rede de Petri colorida um usuário no papel de professor de uma disciplina também for aluno da mesma disciplina. Para verificar essa possível inconsistência escrevemos a consulta em linguagem ML descrita na Figura 4.

```
fun match ( _,d,1,(prof, _),_, [(aluno1, _),(aluno2, _),(aluno3, _)]) =
      (prof = aluno1) orelse (prof = aluno2) orelse (prof = aluno3)
 | match _ = false

fun Pred n = List.exists match(Mark.TopLevel'Disc 1 n);

PredNodes(EntireGraph, Pred, NoLimit);
```

```
val match = fn
  : 'a * objtype * int * ("b * 'c) * 'd * ("b * 'e) list -> bool
val Pred = fn : Node -> bool
val it = [] : Node list
```

**Figura 4. Consulta para determinar conflito entre Professor e Aluno**

Quando o código da consulta é executado, a ferramenta CPN Tools retorna como resposta um quadro verde onde destacamos, com uma elipse vermelha, uma lista vazia. A interpretação para essa lista vazia é que não existe nenhum nó no espaço de estados da rede de Petri colorida do *workflow* onde a proposição é violada.

## 6. Discussão

A análise de consistência da política de controle de acesso que considerou inclusive casos de dinamicidade quando surgem novos papéis, ações ou objetos, é bem detalhada e representa um cenário minimalista. Porém, podemos inferir os resultados da análise do modelo para cenários com alta cardinalidade de usuários, papéis, permissões.

Apesar da cardinalidade de elementos do modelo da rede de Petri colorida ser pequena, esse fator não inviabiliza a aplicação do estudo realizado por nós para analisar cenários com cardinalidades elevadas.

Para analisar um cenário complexo seria necessário seguir uma estratégia semelhante a divisão e conquista. Apesar da política de uma determinada organização apresentar muitas regras é possível separar elas em categorias de tal maneira que poderia ser utilizado modelagem e análise de alcançabilidade em espaço de estados para analisar cada um dos subconjuntos de regras separadamente.

Portanto, a propriedade de separabilidade das regras de uma política de controle de acesso de um sistema permite que sejam realizadas análise de submodelos menores que podem aferir a consistência desta política através da abordagem que apresentamos.

## 7. Conclusão

A primeira contribuição de nossa pesquisa foi a publicação de um artigo [Ueda e Ruggiero 2012c], em periódico, que apresenta o resultado de um mapeamento sistemático. Nesse artigo foram identificadas abordagens propostas para tratar o estabelecimento da relação papel-permissão em modelos baseados em papéis. Não encontramos

 ©2012 SBC — Soc. Bras. de Computação

na literatura da área nenhum estudo sistemático com resultados similares ao mapeamento que realizamos. Portanto, consideramos esta contribuição bastante relevante para a comunidade científica. Outros pesquisadores interessados em estudar controle de acesso baseado em papéis podem se beneficiar desta publicação com o intuito de conhecer possíveis direções de investigação.

A segunda contribuição relevante dessa pesquisa consiste no processo de modelagem e simulação do sistema educacional que adota uma política de controle de acesso, utilizando rede de Petri colorida e CPN Tools como suporte. A representação da política com esse tipo de rede de Petri possibilita gerar um método de avaliação de segurança e também dinamicidade de papéis, ações e objetos. Linguagens de modelagem como UML são populares e úteis para modelar sistemas, porém não permitem realizar simulações e análise exaustiva de espaço de estados. Nossa abordagem se destaca por considerar a interação do *workflow* operacional com a política de controle de acesso baseado em papéis. A modelagem e análise realizada em [Kahloul et al. 2010] e [Laid et al. 2010] não considerou tal interação. Além disso, nossa contribuição também constitui um processo de metamodelagem que pode ser utilizada para representar qualquer sistema através de rede de Petri colorida.

Como terceira contribuição produzimos um método de verificação de consistência de políticas de controle de acesso através da análise de espaço de estados da rede de Petri colorida. Nossos resultados demonstram que é possível analisar a consistência de regras de segurança de uma política e também o impacto do aspecto de dinamicidade de papéis envolvendo ações e objetos, utilizando a propriedade de alcançabilidade de estados na rede de Petri colorida com a ferramenta CPN Tools. O aspecto de dinamicidade de papéis, ações ou objetos não foi bem equacionado e tratado por [Kahloul et al. 2010] e [Laid et al. 2010]. Além disso, também identificamos que é possível analisar uma política de controle de acesso baseado em papéis extensa aplicando uma técnica de separabilidade nas regras da política para que elas sejam agrupadas em categorias que podem ser analisadas independentemente.

No momento, dois artigos estão sendo elaborados para submissão em periódicos que tenham conceito qualis além de considerável fator de impacto. O primeiro artigo cujo título é *"A Meta-Modeling Approach of RBAC Policies using Colored Petri Net"* [Ueda e Ruggiero 2012b] é constituído pela segunda contribuição de nossa pesquisa. O segundo artigo em fase de elaboração que tem título *"Analyzing Dynamicity of Policies in Role Based Access Control Models with Colored Petri Net"* [Ueda e Ruggiero 2012a] é o resultado da terceira contribuição da pesquisa.

Acreditamos que as possibilidades de continuidade desta pesquisa de doutorado são muitas. A seguir, descrevemos de forma sucinta apenas duas delas.

**Análise de Desempenho** – Investigar estratégias para análise de desempenho utilizando rede de Petri colorida em modelos de controle de acesso baseados em papéis, ações ou objetos. Devem ser considerados métodos para mensurar o impacto do efeito de dinamicidade de papéis em uma política de segurança baseada em papéis e sua correlação com o desempenho de um sistema.

**Ferramenta para Análise de Política** – Desenvolver uma ferramenta computacional que possa ser utilizada para automatizar a análise de políticas a partir do processo

 ©2012 SBC — Soc. Bras. de Computação

de metamodelagem, simulação e análise de estados de rede de Petri colorida. Essa ferramenta poderia auxiliar administradores de sistemas no gerenciamento da robustez das políticas de controle de acesso que estão sendo especificadas por eles.

Os resultados produzimos por nossa pesquisa indicam que é possível validar implementações de políticas de controle de acesso desenvolvidas em uma linguagem de programação, como Java ou C++, através da validação de especificações de políticas promovidos por meio da modelagem, simulação e análise de consistência com rede de Petri colorida.

## Referências

Changhua, L., Yaqin, L. e Zhensong, L. (2009). A novelty approach for checking consistency of security policy. In *Proceedings of the 2009 International Forum on Information Technology and Applications - Volume 03*, IFITA'09, pages 319–324, Washington, DC, USA. IEEE Computer Society.

Cholvy, L. e Cuppens, F. (1997). Analyzing consistency of security policies. In *IEEE Symposium on Security and Privacy*, pages 103–112.

Ferraiolo, D. e Kuhn, R. (1992). Role-based access control. In *In 15th NIST-NCSC National Computer Security Conference*, pages 554–563.

Hadjidj, R. e Boucheneb, H. (2011). Efficient reachability analysis for time petri nets. *IEEE Trans. Comput.*, 60(8):1085–1099.

Jensen, K. e Kristensen, L. M. (2009). *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. Springer.

Kahloul, L., Djouani, K., Tfaili, W., Allaoua, C. e Amirat, Y. (2010). Modeling and verification of rbac security policies using colored petri nets and cpn-tool. In Zavoral, F., Yaghob, J., Pichappan, P. e El-Qawasmeh, E., editors, *Networked Digital Technologies - Second International Conference, NDT 2010, Prague, Czech Republic, July 7-9, 2010. Proceedings, Part II*, volume 88 of *Communications in Computer and Information Science*, pages 604–618. Springer.

Laid, K., Djouani, K. e Tfaili, W. (2010). Using timed colored petri nets and cpn-tool to model and verify trbac security policies. In *Fourth International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS 2010)*.

Petri, C. A. (1962). *Kommunikation mit Automaten*. PhD thesis, Institut für instrumentelle Mathematik, Bonn.

Ribeiro, C., Zuquete, A., Ferreira, P. e Guedes, P. (2000). Security policy consistency. *CoRR*, cs.LO/0006045.

Ueda, E. T. e Ruggiero, W. V. (2012a). Analyzing dynamicity of policies in role based access control models with colored petri net.

Ueda, E. T. e Ruggiero, W. V. (2012b). A meta-modeling approach of rbac policies using colored petri net.

Ueda, E. T. e Ruggiero, W. V. (2012c). A systematic mapping on the role-permission relationship in role based access control models. *IEEE Latin America Transactions*, 10(1):1243–1250.

 ©2012 SBC — Soc. Bras. de Computação

# Segurança do bit menos significativo no RSA e em curvas elípticas

**Dionathan Nakamura**[1], **Routo Terada**[1]

[1]Departamento de Ciência da Computação – Instituto de Matemática e Estatística
Universidade de São Paulo (USP)

{nakamura,rt}@ime.usp.br

***Abstract.*** *In the master's thesis, algorithms are presented to invert cryptographic systems like RSA making use of oracles that predict the LSB. We implemented two of them, critical parameters are identified and the original sampling is changed. With the modified sampling we achieve an improvement in the execution times.*

***Resumo.*** *Na dissertação são apresentados algoritmos que conseguem inverter criptossistemas como o RSA fazendo uso de oráculos que predizem o LSB. Fizemos a implementação de dois desses algoritmos, identificamos parâmetros críticos e mudamos a amostragem do formato original. Com a modificação na amostragem conseguimos uma melhora nos tempos de execução.*

## 1. Introdução

Na dissertação foi investigada a relação da segurança de algoritmos criptográficos com os bits que compõem seus segredos como em [Goldwasser et al. 1982, Ben-Or et al. 1983, Alexi et al. 1988, Fischlin e Schnorr 2000, Boneh e Shparlinski 2001, Jetchev e Venkatesan 2008, Chevalier et al. 2009, Hofheinz e Kiltz 2009, Roh e Hahn 2010], em especial com o LSB (*least significant bit*).

Esses estudos analisam o algoritmo RSA e o protocolo Diffie-Hellman sobre Curvas Elípticas (DHCE) — enquanto no RSA o interesse está no LSB da mensagem original, no DHCE está no LSB da chave secreta combinada. Nos estudos com RSA, o algoritmo Rabin também é abordado por sua similaridade. Então, verificar a segurança do LSB é o mesmo que verificar a segurança de geradores de números pseudoaleatórios como em [Blum e Micali 1984, Blum et al. 1986].

Dos trabalhos relacionados, destacamos para o RSA o artigo de [Alexi et al. 1988] onde mostrou ser possível inverter o RSA com um oráculo com probabilidade de acerto de $(50 + \varepsilon)\%$, onde $\varepsilon$ é bem pequeno, todavia não negligenciável. Vamos chamar esse trabalho (e o respectivo algoritmo) de ACGS por brevidade.

Para o DHCE, destacamos o artigo de [Boneh e Shparlinski 2001] onde mostrou que o LSB da abscissa do ponto combinado é imprevisível, ou seja, a existência de um oráculo para o LSB implicaria a quebra do DHCE. Vamos chamar esse trabalho (e o respectivo algoritmo) de BS.

Em [Fischlin e Schnorr 2000], os autores mostraram como esses algoritmos podem ter seu tempo de execução demorado, a ponto de ultrapassar o tempo de execução de soluções para problemas difíceis, no caso, o problema da fatoração de inteiros (PFI). Aqui

 ©2012 SBC — Soc. Bras. de Computação

surgiu nossa motivação. Os trabalhos são apresentados apenas como artigos científicos, imaginávamos como eles se comportariam depois de implementados. Apesar de apresentarem convergência teórica, queremos verificar se ela se traduz na prática com o uso dos geradores de aleatoriedade que dispomos. Ainda, se o tempo real de computação é viável com recursos computacionais modestos.

Assim, neste trabalho nós implementamos o ACGS para o RSA e em seguida o BS para DHCE. Como parte secundária dos objetivos, queríamos identificar os parâmetros que mais influenciavam no tempo de execução, procurando encontrar onde erros eram superestimados e promover melhorias no modo como os oráculos eram utilizados.

As principais contribuições deste trabalho são as seguintes:

- provê uma implementação da segurança do LSB no algoritmo RSA;
- provê uma implementação da segurança do LSB no protocolo DHCE;
- identificação dos parâmetros relevantes para o tempo de execução;
- conseguimos uma redução no limitante da função de paridade;
- encontramos uma melhoria significativa no número de acessos ao oráculo;
- das melhorias, o ACGS passou a ser mais rápido que o PFI para valores práticos.

## 2. Conceitos preliminares

Consideramos nesse trabalho curvas elípticas $E$ sobre um corpo finito $\mathbb{F}$ de característica prima $\mathrm{char}(\mathbb{F}) = p$, $p \neq 2$, $p \neq 3$ e sendo $p$ grande. Apresentamos uma curva elíptica $E/\mathbb{F}$ na forma da equação simplificada de Weierstrass $E : y^2 = x^3+ax+b$, onde $a, b \in \mathbb{F}$, $\infty$ é o ponto no infinito e o discriminante $\Delta = 4a^3 + 27b^2 \neq 0$.

Adotamos notar o módulo $N$ com letra maiúscula e seu tamanho em bits $n = \lfloor \lg N \rfloor + 1$ com letra minúscula. Para facilitar a escrita, seja $x$ um inteiro, denotamos $[x]_N$ como o resto de $x$ módulo N, ou seja, $[x]_N = x \bmod N$.

O que chamamos de oráculo neste trabalho trata-se de um algoritmo que, ao receber como entrada parâmetros públicos de um criptossistema, retorna o LSB da parte secreta desse sistema em tempo polinomial.

Dizemos que um oráculo tem **vantagem** $\varepsilon$ se ele acerta o LSB com uma probabilidade igual ou superior a $\frac{1}{2}+\varepsilon$. Por exemplo, para o DHCE em uma curva $E$, gerador $G$ de ordem $q$ e $a, b$ distribuídos uniformemente em $[1, q-1]$, o algoritmo $\mathcal{A}$ é um oráculo com vantagem $\varepsilon$ em predizer o LSB da coordenada $x$ da função Diffie-Hellman e denotamos $\mathrm{Adv}^{X}_{E,G}(\mathcal{A}) = \left|\Pr_{a,b}[\mathcal{A}(E, G, aG, bG) = \mathrm{LSB}(x)] - \frac{1}{2}]\right| > \varepsilon$.

A Desigualdade de Chebyshev é uma ferramenta estatística que nos permite estabelecer limitantes para uma distribuição de probabilidade que não conhecemos. Por outro lado, devemos levar em conta que ela pode não definir aproximações muito exatas de probabilidade. Citamos dois exemplos do livro de Ross [Ross 2006, Example 8.8b].

## 3. Métodos e Implementação

Suponha a existência de um oráculo $\mathcal{O}_N$ com vantagem $\varepsilon$ em descobrir o LSB do RSA com módulo $N$. Onde a vantagem $0 < \varepsilon \leq \frac{1}{2}$ é pequena, porém não negligenciável. Vamos resumidamente explicar como o ACGS faz uso de $\mathcal{O}_N$ para inverter o RSA como no algoritmo da pág. 197 de [Alexi et al. 1988];

 ©2012 SBC — Soc. Bras. de Computação

Para a mensagem secreta $x$, o ACGS computa dois valores aleatórios $[ax]_N$ e $[bx]_N$, assegura que $[ax]_N$ seja ímpar e depois calcula o MDC com ajuda do Brent Kung GCD [Brent e Kung 1984], por brevidade chamaremos esse MDC binário de BKGCD.

Ao final do algoritmo, a variável $a$ contém um valor cujo produto com $x$ é o MDC de $[ax]_N$ e $[bx]_N$. É neste momento que esperamos que o $[ax]_N$ e $[bx]_N$ escolhidos sejam coprimos, pois teremos $\mathrm{RSA}(ax) = \pm 1$ e assim conseguimos recuperar $x$ através do cálculo do inverso multiplicativo $[\pm a^{-1}]_N$. Caso não sejam, o algoritmo, repetidamente, sorteia novamente $a$ e $b$ na esperança que esses resultem em $[ax]_N$ e $[bx]_N$ coprimos.

Por um famoso teorema de Dirichlet [Knuth 1981, Sec.4.5.2 Theo.D p.324], a probabilidade de dois inteiros aleatórios no intervalo $[-K, K]$ serem relativamente primos converge para $\frac{6}{\pi^2}$ conforme $K$ tende a $\infty$. Isso é algo em torno de $60{,}8\%$. Assim, com duas tentativas em $a$ e $b$ já é esperado encontrar coprimos em $[ax]_N$, $[bx]_N$.

Outra condição que faz o algoritmo repetir se refere à variável $limiteGCD$, que chamaremos limitante do BKGCD. Esse limitante é para o caso de a função de paridade PAR ser falha, assim o algoritmo não ficará executando indefinidamente. Um exemplo dessa função é, dado $\mathrm{PAR}(d, y)$, retorna a paridade de $[dx]_N$, onde $y$ é criptograma de $x$.

A função PAR gera uma quantidade $m$ de amostras entorno da paridade de $dx$, denominadas **medidas-dx**. São somados os votos para a resposta, par ou ímpar, e o resultado é a maioria dos votos.

Para $r$'s aleatórios, cada voto de paridade é obtido verificando se $\mathrm{LSB}(rx) = \mathrm{LSB}(rx + dx)$. A igualdade mantém se, e somente se, $dx$ for par, isso porque um número par não altera a paridade de outro número na operação de adição.

Com esse esquema de votos, conseguimos construir uma função de paridade quase perfeita, desde que a quantidade de medidas $m$ seja grande o suficiente. Isso é garantido pela lei dos grandes números.

Na comparação $\mathrm{LSB}(rx) = \mathrm{LSB}(rx + dx)$ são feitas duas consultas aos oráculos, ou seja, $\mathcal{O}_N(rx)$ e $\mathcal{O}_N(rx + dx)$. Isso gera um problema conhecido como **duplicação de erro** [Ben-Or et al. 1983]. Para contorná-lo, o ACGS usa a função PAR de modo que o oráculo só é consultado uma vez a cada voto. Isso é possível dividindo o conjunto dos inteiros módulo N em intervalos e aplicado uma técnica de suposição sobre eles.

Note que das suposições feitas em [Alexi et al. 1988], temos ao todo $2 \cdot 4\varepsilon^{-1} \cdot 2 \cdot 4m\varepsilon^{-1} = 2^6 m\varepsilon^{-2}$ possibilidades. Rodamos o ACGS com cada uma dessas alternativas. Em apenas uma delas estaremos computando corretamente $\mathrm{LSB}(r_i x)$ e chamaremos essa instância de **alternativa correta**. Então, na alternativa correta o ACGS recupera a mensagem original.

Os autores calcularam utilizando Desigualdade de Chebyshev a probabilidade de erro da função PAR igual a $\Pr(\text{PAR errar}) = \frac{4}{m\varepsilon^2}$. Então, temos que escolher uma amostragem $m$ suficientemente grande para que tenhamos uma função de paridade quase perfeita. Os autores escolheram $m \stackrel{def}{=} 64n\varepsilon^{-2}$, assim $\Pr(\text{PAR errar}) \leq \frac{1}{16n}$.

Com probabilidade de erro $\frac{1}{16n}$, a função PAR, em média, consegue passar sem erro em todas as requisições da função BKGCD (no máximo $limiteGCD = 6n + 3$).

De acordo com a nossa discussão sobre a Desigualdade de Chebyshev na Seção 2,

 ©2012 SBC — Soc. Bras. de Computação

acreditamos que a probabilidade de erro da função PAR é superestimada aqui. Verificamos também, empiricamente, que na média o limitante do BKGCD pode ser menor do que $6n + 3$, pelo menos metade.

O tempo de execução do ACGS corresponde a $\varepsilon^{-2}$ tentativas para se obter $ax$ e $bx$ pequenos, mais duas tentativas para que sejam coprimos $\left(\frac{\pi^2}{6}\right)$, mais a execução das $2^6 m\varepsilon^{-2}$ alternativas, $6n + 3$ chamadas à função PAR e $m$ medidas-dx. Escrevendo tudo:

$$\varepsilon^{-2}\cdot 2\cdot 2^6 m\varepsilon^{-2}\cdot(6n+3)\cdot m \approx 3\cdot 2^8\varepsilon^{-4}nm^2 \approx 3\cdot 2^8\varepsilon^{-4}n\left(\frac{64n}{\varepsilon^2}\right)^2 \approx 3\cdot 2^{20}\varepsilon^{-8}n^3 = O(\varepsilon^{-8}n^3)$$

O algoritmo BS faz uma adaptação do algoritmo ACGS para uso no DHCE. Considere um grupo cíclico de curva prima $E(\mathbb{F}_p)$ com um gerador $G$ de ordem prima $q$ e as chaves privadas de Alice e Beto $a, b \in [1, q-1]$, definimos o segredo da função Diffie-Hellman $DH_{E,G}(aG, bG) = abG$ como a coordenada $x$ do ponto $abG$. Nessa seção, supomos a existência de um oráculo $\mathcal{O}_p$ para $[abG]_x$, tal que $\mathrm{Adv}^X_{E,G}(\mathcal{O}_p) = \varepsilon$.

Para um $\lambda \in \mathbb{F}_p^*$, $\phi_\lambda(E)$ define a curva elíptica $Y^2 = X^3 + A\lambda^4 X + B\lambda^6$, os autores chamam-na *twisted elliptic curve*. É fácil identificar que esses *twists* são na verdade isomorfismos. Assim, para pontos $Q, R, S, \in E$, com $Q = (x, y)$, temos $(Q)_\lambda = Q_\lambda = (\lambda^2 x, \lambda^3 y) \in \phi_\lambda(E)$ e para $Q + R = S$, temos $Q_\lambda + R_\lambda = S_\lambda$ (propriedade homomórfica). Ainda, com tais *twists*, definimos para uma curva inicial $E_0$, uma família de curvas isomorfas $\phi_\lambda(E_0)_{\lambda\in\mathbb{F}_p^*}$.

Seja $K_{ab}$ o ponto combinado no DHCE sobre a curva $E_0$ com as chaves públicas $PK_a$ e $PK_b$. A coordenada $x$ de $K_{ab}$ é a chave secreta. Da mesma forma como no ACGS, sorteamos $a'$ e $b'$ (notamos $a/b$ linha para não confundir com as chaves privadas $a/b$) esperando que $[a'x]_p$ e $[b'x]_p$ sejam pequenos. Similarmente ao BKGCD, $a'$ e $b'$ geram valores $d$, aqui chamamos $\lambda$, e uma consulta para a paridade é feita com a chamada de $\mathrm{PAR}(\lambda, PK_a, PK_b)$. Porém, a diferença fica por conta da requisição ao LSB. A função PAR é perguntada pelo LSB de $\lambda^2 x$, pois $(K_{ab})_\lambda = (\lambda^2 x, \lambda^3 y)$. Então, a requisição é feita para um ponto que fica em outra curva, diferente de $E_0$.

Os autores consideram um oráculo $\mathcal{O}_p$ com vantagem $\varepsilon$ em predizer o LSB na curva $E_0$. Isso quer dizer que $\mathcal{O}_p$ não tem vantagem em todas as curvas na família $\phi_\lambda(E_0)_{\lambda\in\mathbb{F}_p^*}$. Por definição, é considerado que $\mathcal{O}_p$ mantém essa vantagem para pelo menos uma fração $\delta$ das curvas na família de isomorfismo.

O problema é que o algoritmo BS precisa consultar a paridade para vários valores de $\lambda$. Então, os autores mostram como construir um novo oráculo $\mathcal{B}_p$ cuja probabilidade de sucesso dentro ou fora da fração $\delta$ das curvas seja conhecida.

A seguir, resumidamente apresentamos os passos do algoritmo BS:

1. com entrada $\langle E, G, PK_a, PK_b\rangle$, onde $PK_a = aG$ e $PK_b = bG$ e $G$ de ordem prima $q$, queremos calcular o ponto $C = abG$;
2. como $a$ e $b$ são fixos e desconhecidos, aleatorizamos o processo definindo $PK_{ra} = a_r aG$ e $PK_{rb} = b_r bG$, para $a_r, b_r \in [1, q-1]$, e esperamos que os valores $a_r a$ e $b_r b$ levem a um caso onde o oráculo $\mathcal{B}_p$ tenha vantagem não negligenciável;
3. sendo $DH_{E,G}(PK_{ra}, PK_{rb}) = D$, basta calcular $D$ para obtermos $C$, pois $C = c_r \cdot D$, onde $c_r \equiv (a_r b_r)^{-1} \bmod q$;

 ©2012 SBC — Soc. Bras. de Computação

4. agora, executamos um algoritmo similar ao ACGS com o oráculo $\mathcal{B}_p$;
5. a fim de assegurar que encontremos os valores $a_r a$ e $b_r b$ desejados pode ser necessário repetir todo o processo $\frac{8}{\varepsilon\delta}$ vezes.

Ao final do algoritmo BS, teremos uma lista de candidatos $C$ para o ponto $abG$. Porém, ao contrário do ACGS, não temos condições de identificar automaticamente a alternativa correta, o que nos obriga a executar todas as alternativas e apenas ao final tentar identificar a resposta correta. A identificação é possível com um algoritmo devido a Shoup [Shoup 1997, Theo.7].

### 3.1. Biblioteca criptográfica Relic

Para implementar os algoritmos ACGS e BS, utilizamos a linguagem de programação C e a biblioteca criptográfica ***Relic Toolkit*** versão 0.3.0 [Aranha e Gouvêa ].

Utilizamos as curvas SECG_P160 e NIST_P224 da biblioteca para o DHCE. Para o RSA, utilizamos módulos de 1024, 2048 e 5000 bits, além de um módulo teórico de 128 bits para registrar tempos mais demorados. Escolhemos 1024 e 5000 bits devido a [Fischlin e Schnorr 2000] e 2048 bits por ser a recomendação atual do NIST[1]. E por ter equivalência no nível de segurança a 1024 e 2048 bits do RSA, escolhemos 160 e 224 bits para as curvas elípticas.

A respeito da implementação dos oráculos, a resposta para a consulta no DHCE é imediata, com a execução de apenas alguns poucos cálculos. Já no RSA, é feita uma decriptação para cada consulta, resultando em um tempo para a resposta bem maior.

O gerador pseudoaleatório construído dentro da Relic é o FIPS 186-2 baseado em SHA1. A Relic já possui ferramentas de cronômetro (*benchmark*) para registrar os tempos de execução, nós utilizamos a versão mais precisa desses cronômetros (HPROC). Para simulação dos algoritmos e coleta dos tempos de execução, utilizamos um computador com processador Intel Core 2 Duo T5450 de 1,66 Ghz com 2 GB de memória RAM.

Os códigos fontes das implementações estão disponíveis no site do Laboratório de Segurança de Dados do IME-USP[2]. A dissertação se encontra no site de teses da USP [3].

## 4. Resultados

O número $m$ de amostragens é um parâmetro crítico do sistema. No modelo original do ACGS ele é por definição $m \stackrel{def}{=} \frac{64n}{\varepsilon^2}$. Dessa forma, o valor de $m$ depende apenas do tamanho do módulo $N$ em bits e da vantagem $\varepsilon$ do oráculo. Na Tabela 1 apresentamos alguns exemplos de amostragem em relação ao par $n, \varepsilon$.

Procuramos tentar reduzir a amostragem. Percebemos que para valores menores do que a do modelo original, o algoritmo ACGS ainda obtinha sucesso em alguns casos. Após tentativas, obtivemos empiricamente a seguinte definição do parâmetro $m \stackrel{def}{=} \frac{4}{\varepsilon^2}$. Outrossim, o limitante do BKGCD pode ser reduzido pela metade $limiteGCD = 3n$.

Na Figura 1, apresentamos exemplos para o novo padrão de amostragem. Note que no gráfico incluímos apenas a vantagem, já que no novo padrão o tamanho do módulo não influencia o número de amostras.

[1] `http://csrc.nist.gov/groups/ST/toolkit/key_management.html`
[2] `http://lsd.ime.usp.br/`
[3] `http://www.teses.usp.br/teses/disponiveis/45/45134/tde-14032012-213011/en.php`

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Amostragem em relação aos bits e à vantagem do oráculo.**

| | 128 bits | 1024 bits | 5000 bits |
|---|---|---|---|
| 0,4 | 51.200 | 409.600 | 2.000.000 |
| 0,3 | 91.023 | 728.178 | 3.555.556 |
| 0,2 | 204.800 | 1.638.400 | 8.000.000 |
| 0,1 | 819.200 | 6.553.600 | 32.000.000 |
| 0,05 | 3.276.800 | 26.214.400 | 128.000.000 |
| 0,01 | 81.920.000 | 655.360.000 | 3.200.000.000 |

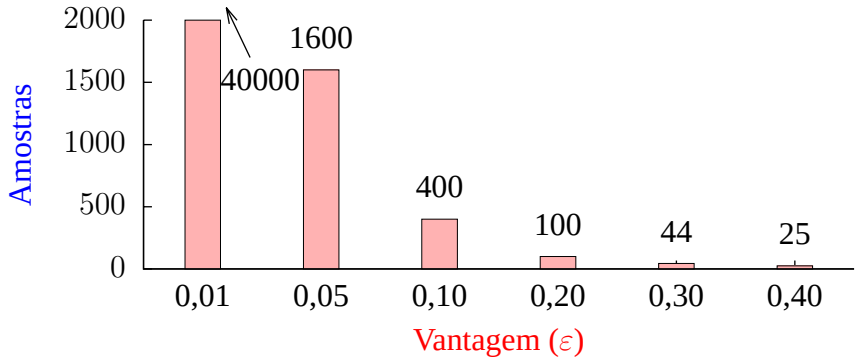



**Figura 1. Número de amostras em relação à vantagem do oráculo.**

A Tabela 2 traz estimativas de complexidade. Adicionamos também o tempo do algoritmo de solução para o PFI ([Buhler et al. 1993]) como o utilizado em [Fischlin e Schnorr 2000]. Perceba que o ACGS passou a ser mais rápido que o PFI.

**Tabela 2. Comparação entre as complexidades existentes.**

| | Estimativa | 1024 bits | 2048 bits | 5000 bits |
|---|---|---|---|---|
| PFI | $\exp(1{,}9(\ln N)^{\frac{1}{3}}(\ln\ln N)^{\frac{2}{3}})$ | $6{,}409 \cdot 10^{25}$ | $5{,}817 \cdot 10^{34}$ | $3{,}755 \cdot 10^{50}$ |
| ACGS orig. | $3 \cdot 2^{20}\varepsilon^{-8}n^3$ | $3{,}378 \cdot 10^{31}$ | $2{,}702 \cdot 10^{32}$ | $3{,}932 \cdot 10^{33}$ |
| ACGS novo | $3 \cdot 2^{11}\varepsilon^{-8}n$ | $6{,}292 \cdot 10^{22}$ | $1{,}258 \cdot 10^{23}$ | $3{,}072 \cdot 10^{23}$ |

A Tabela 3 traz os tempos estimados para se tentar inverter o RSA com $\varepsilon = 0{,}1$. Temos uma coluna para a alternativa correta e uma para o tempo total do algoritmo.

Quanto ao algoritmo BS, seguimos a mesma linha de discussão do RSA: nova amostragem e $limiteGCD = 3 \lg p$. Apenas uma execução do BS é considerada, assim, não entra aqui as execuções adicionais requeridas pelo algoritmo de Shoup.

Quanto à amostragem, deve-se levar em conta que vamos utilizar dois algoritmos, o oráculo $\mathcal{O}_p$ e o oráculo $\mathcal{B}_p$. O oráculo $\mathcal{B}_p$ é regido pelo parâmetro $\delta$ que define a proporção de curvas onde o oráculo $\mathcal{O}_p$ possui vantagem $\varepsilon$. O caso ideal é quando $\delta = 1$, o que define o oráculo $\mathcal{O}_p$ como aplicável a todas as curvas da família.

No melhor caso, $\delta = 1$, temos um algoritmo BS semelhante ao ACGS. Com amostragem $m \stackrel{def}{=} \frac{8}{\varepsilon^2}$ e $\varepsilon = 0{,}1$ montamos a Tabela 4 com os tempos estimados.

Agora analisaremos o pior caso, onde $\delta < 1$ e próximo de zero. Fixamos $\delta = 0{,}1$ e

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 3. Tempos estimados para execução do algoritmo ACGS.**

| n (bits) | Alt. correta | Tempo total |
|---|---|---|
| 1024 | 19,7 dias | $138{,}4 \cdot 10^3$ anos |
| 2048 | 243,5 dias | $1{,}7 \cdot 10^6$ anos |
| 5000 | 20,3 anos | $52{,}1 \cdot 10^6$ anos |

**Tabela 4. Tempos estimados para execução do algoritmo BS.**

| $\lg p$ (bits) | Alt. correta | Tempo total |
|---|---|---|
| 160 | 5,6 minutos | 54,6 anos |
| 224 | 9,9 minutos | 97,3 anos |

então usamos o oráculo $\mathcal{B}_p$ que implica em uma nova amostragem $m2 \stackrel{def}{=} \frac{8}{\varepsilon\delta^2}$ relacionada com os acessos ao oráculo $\mathcal{O}_p$ pelo oráculo $\mathcal{B}_p$.

Diretamente, apresentamos a Tabela 5 com os tempos estimados para se tentar descobrir a chave combinada pelo DHCE com uso do oráculo $\mathcal{B}_p$.

**Tabela 5. Tempos estimados para execução do algoritmo BS com o oráculo $\mathcal{B}_p$.**

| $\lg p$ (bits) | Alt. correta | Tempo total |
|---|---|---|
| 160 | 15,6 anos | $80 \cdot 10^6$ anos |
| 224 | 44,7 anos | $229 \cdot 10^6$ anos |

## 5. Conclusões

Com base nos algoritmos propostos, conseguimos construir implementações dos artigos com êxito na recuperação dos segredos. Alcançamos sucesso com um nível de convergência bem maior do que o do cálculo aproximado.

Nossos testes conseguiram, de maneira experimental, ajustar os parâmetros críticos do sistema para diminuir significativamente o tempo de execução. E com os tempos diminuídos, os métodos ainda alcançavam taxas de sucesso consideradas aceitáveis. Isto é garantido pela lei dos grandes números.

O resultados provêem medidas de quão rápido é inverter os criptossistemas citados a partir de um mínimo comprometimento do LSB. Em particular, no caso do ACGS com RSA-1024 tivemos um avanço significativo no tempo de inversão a ponto de ficar abaixo do tempo de solução para o PFI, anteriormente não alcançado.

Adicionalmente, incluímos a análise de um BS com um $\delta$ ideal. Tivemos a oportunidade de contribuir com sugestões de melhorias na biblioteca brasileira Relic, alguns exemplos: correção dos tipos na função de *benchmark*, correção com fp_cmp_dig, adição do gt_exp $\rightarrow$ fb4_exp, correção de bilinearidade do emparelhamento simétrico, reconstrução de pontos para NIST_P224 e problemas com linker.

Uma ideia para prova de conceito de nossos métodos é criar uma implementação falha (revelando o LSB através de um ataque de canal secundário) do RSA e então adaptá-la como oráculo para o ACGS. Um trabalho como esse é capaz de dar dimensões práticas da aplicabilidade do método.

 ©2012 SBC — Soc. Bras. de Computação

## Referências

[Alexi et al. 1988] Alexi, W., Chor, B., Goldreich, O. e Schnorr, C.-P. (1988). RSA and Rabin functions: Certain parts are as hard as the whole. *SICOMP*, 17(2):194–209.

[Aranha e Gouvêa ] Aranha, D. F. e Gouvêa, C. P. L. RELIC is an Efficient LIbrary for Cryptography. `http://code.google.com/p/relic-toolkit/`.

[Ben-Or et al. 1983] Ben-Or, M., Chor, B. e Shamir, A. (1983). On the cryptographic security of single RSA bits. In *ACM Symposium on Theory of Computing (STOC '83)*, pages 421–430, Baltimore, USA. ACM Press.

[Blum et al. 1986] Blum, Blum e Shub (1986). A simple unpredictable pseudo-random number generator. *SICOMP: SIAM Journal on Computing*, 15.

[Blum e Micali 1984] Blum, M. e Micali, S. (1984). How to generate cryptographically strong sequence of pseudo-random bits. *SIAM Journal Computing*, 13:850–864.

[Boneh e Shparlinski 2001] Boneh, D. e Shparlinski, I. (2001). On the unpredictability of bits of the elliptic curve Diffie-Hellman scheme. In Kilian, J., editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 201–212. Springer.

[Brent e Kung 1984] Brent e Kung (1984). Systolic VLSI arrays for polynomial GCD computation. *IEEE Transactions on Computers*, 33.

[Buhler et al. 1993] Buhler, J. P., Lenstra, H. W. e Pomerance, C. (1993). Factoring integers with the number field sieve. In *The development of the number field sieve*, volume 1554 of *LNM*, pages 50–94. Springer-Verlag, Berlin.

[Chevalier et al. 2009] Chevalier, C., Fouque, P.-A., Pointcheval, D. e Zimmer, S. (2009). Optimal randomness extraction from a Diffie-Hellman element. In Joux, A., editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 572–589. Springer-Verlag.

[Fischlin e Schnorr 2000] Fischlin, R. e Schnorr, C.-P. (2000). Stronger security proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244.

[Goldwasser et al. 1982] Goldwasser, S., Micali, S. e Tong, P. (1982). Why and how to establish a private code on a public network (extended abstract). In *FOCS*, pages 134–144, Chicago, Illinois. IEEE.

[Hofheinz e Kiltz 2009] Hofheinz, D. e Kiltz, E. (2009). Practical chosen ciphertext secure encryption from factoring. In Joux, A., editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 313–332. Springer.

[Jetchev e Venkatesan 2008] Jetchev, D. e Venkatesan, R. (2008). Bits security of the elliptic curve Diffie-Hellman secret keys. In Wagner, D., editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 75–92. Springer.

[Knuth 1981] Knuth, D. E. (1981). *The Art of Computer Programming*, volume 2, Seminumerical Algorithms. Addison-Wesley, Reading, MA, 2 edition.

[Roh e Hahn 2010] Roh, D. e Hahn, S. G. (2010). On the bit security of the weak Diffie–Hellman problem. *Information Processing Letters*, 110:799–802.

[Ross 2006] Ross, S. M. (2006). *A First Course in Probability*. Prentice Hall, 7 edition.

[Shoup 1997] Shoup, V. (1997). Lower bounds for discrete logarithms and related problems. In *EUROCRYPT '97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag.

 ©2012 SBC — Soc. Bras. de Computação

# Um esquema bio-inspirado para tolerância à má-conduta em sistemas de quórum apoiando serviços em MANETs

**Elisa Mannes***
**Co-orientadora: Michele Nogueira - Orientador: Aldri Santos**

[1]NR2 - Departamento de Informática - Universidade Federal do Paraná
Caixa Postal 19.081 - 81.531-980 – Curitiba – PR – Brasil

{elisam, michele, aldri}@inf.ufpr.br

***Abstract.*** *Network operation services in MANETs, such as resource location and distribution of connectivity information, must be reliable to support the use of applications. The reliability and availability of these services can be obtained by data management techniques, such as quorum based replication system. However, quorum systems are vulnerable to misbehavior nodes, that intentionally do not cooperate with replication operations or disseminate false information in order to disable the network. To deal with these issues, we propose and evaluate a bio-inspired scheme for tolerance of selfish and malicious nodes in replication operations on quorum systems. In contrast to existing approaches,* $QS^2$ *is autonomous and distributed, and does not require the use of extra messages. The* $QS^2$ *was applied on realistic scenarios of MANETs and the results show that* $QS^2$ *provides a reliability higher than 90% for replicated data even facing malicious or selfish nodes actions.*

***Resumo.*** *Os serviços de operação de rede em MANETs, como localização de recursos e distribuição de informações de conectividade, devem ser confiáveis para oferecer apoio às aplicações. A confiabilidade e a disponibilidade desses serviços podem ser obtidas por meio de técnicas de gerência de dados, como a replicação por sistemas de quórum. Entretanto, os sistemas de quórum são vulneráveis a nós de má-conduta que intencionalmente não colaboram com as operações de replicação ou divulgam dados falsos a fim de prejudicar o funcionamento da rede. Para lidar com esses problemas, nós propomos e avaliamos um esquema bio-inspirado para a tolerância de nós egoístas e maliciosos em operações de replicação nos sistemas de quórum. Diferente de abordagens existentes, este esquema, chamado de* $QS^2$*, é autônomo, distribuído e não requer o uso de mensagens extras. O* $QS^2$ *foi aplicado diante de cenários realísticos de MANETs, e os resultados mostram que o* $QS^2$ *provê uma confiabilidade dos dados acima de 90% mesmo diante de nós egoístas e maliciosos.*

## 1. Introdução

Os recentes esforços para o desenvolvimento e aprimoramento das redes sem fio reforçam a intenção do uso dessas redes para suportar a Internet do Futuro no provimento de aplicações em diversas áreas, como saúde, transporte e entretenimento [Balasubramaniam et al. 2011]. As MANETs compõem o conjunto de redes sem fio que tendem a convergir para suportar essas aplicações, sendo compostas de dispositivos (nós) móveis, que possuem recursos escassos. Essas características podem facilmente ocasionar

*Este trabalho foi financiado pela CAPES

 ©2012 SBC — Soc. Bras. de Computação

a partição da rede, tornando os serviços indisponíveis ou sustentando informações desatualizadas. Os serviços de operação de rede, como localização de recursos e distribuição de informações de conectividade, devem auxiliar a rede a gerenciar essas características, visto que ambas as situações influenciam no funcionamento da rede, dos serviços e das aplicações, comprometendo o bom desempenho da rede.

A replicação de dados é uma das técnicas de gerenciamento utilizada para lidar com problemas originados pela partição da rede e pela mobilidade, fornecendo disponibilidade e tolerância a falhas para os dados dos serviços de operação de rede. Dentre as técnicas de replicação propostas para as MANETs, esse trabalho destaca os sistemas de quórum como uma solução efetiva para a replicação. Os sistemas de quórum consistem em conjuntos de nós, em que cada subconjunto formado por operações de leitura e de escrita devem se intersectar. Os sistemas de quórum probabilísticos possuem características ainda mais relevantes para as MANETs, tornando a replicação por quórum mais dinâmica e apropriada para esse tipo de rede [Luo et al. 2003].

Entretanto, os sistemas de quórum para MANETs apresentam vulnerabilidades que podem ser exploradas por nós de má-conduta, como nós egoístas e maliciosos. Tais vulnerabilidades estão presentes pela necessidade de colaboração na replicação e pelas próprias características das MANETs, que permite a entrada e a saída de nós na rede e não é apoiada por uma infraestrutura fixa de controle e gerência. Um sistema de replicação por quórum com a presença de nós maliciosos em sua operação é capaz de oferecer aproximadamente apenas 10% de confiabilidade para os dados replicados [Mannes et al. 2009]. Nós egoístas também influenciam na confiabilidade dos dados, porém com um menor impacto [Mannes et al. 2010]. Portanto, os sistemas de quórum precisam tolerar a ação dos nós de má-conduta que interferem em seu desempenho, e assim suportar corretamente a disponibilidade dos dados dos serviços de operação de rede.

Em geral, as soluções existentes para lidar com nós de má-conduta assumem a existência de uma infraestrutura fixa, canais confiáveis ou o compartilhamento de recomendações, implicando em um custo extra de comunicação. Contudo, o emprego de uma solução para a tolerância aos nós de má-conduta em MANETs deve ser preferencialmente autônoma, de baixo custo e auto-organizada, de forma a complementar e ressaltar a infraestrutura dinâmica das MANETs. Essas características são encontradas naturalmente em sistemas biológicos, fornecendo inspiração para o desenvolvimento eficiente de soluções descentralizadas e autônomas para MANETs [Balasubramaniam et al. 2011].

Este trabalho apresenta um esquema bio-inspirado que provê tolerância aos sistemas de quórum diante da ação de nós de má-conduta nas operações de replicação[1]. Este esquema, chamado $QS^2$ (*quorum systems, quorum sensing*), é inspirado nos mecanismos biológicos de sensoriamento em quórum e de seleção por parentesco, ambos presentes em bactérias. Diferente das soluções atuais, os nós confiáveis são capazes de detectar nós egoístas e maliciosos baseando-se em uma análise independente do comportamento dos nós. Além disso, o esquema evita a interação entre nós confiáveis e nós de má-conduta, e assim aumenta a confiabilidade dos dados dos serviços de operação de rede.

As contribuições decorrentes desta dissertação de mestrado são: a análise das vulnerabilidades de segurança um sistema de quórum probabilístico; a definição de métricas

[1]Dissertação de mestrado disponível em http://www.nr2.ufpr.br/∼elisa/publications

 ©2012 SBC — Soc. Bras. de Computação

para aferir o impacto desse sistema diante de ataques; a proposta e a especificação do esquema $QS^2$; a aplicação do $QS^2$ e a sua análise no apoio aos serviços de operação de rede em cenários realísticos de MANETs sob ataques.

O restante do artigo está organizado como descrito a seguir. A Seção 2 relata as vulnerabilidades e o impacto de ataques de nós de má-conduta em um sistema de quórum para MANETs. A Seção 3 descreve o funcionamento do $QS^2$. A Seção 4 mostra os resultados da avaliação do esquema e a Seção 5 apresenta a avaliação em cenários realísticos. A Seção 6 conclui o artigo e resume as contribuições deste trabalho.

## 2. Vulnerabilidades de segurança nos sistemas de quórum para MANETs

Os sistemas de quórum são um método de replicação de dados apropriados para as MANETs pela dinamicidade e utilização de menos servidores comparado a estratégias clássicas. Eles proveem operações de leitura e de escrita de dados na rede, formando quóruns de leitura e quóruns de escrita [Malkhi and Reiter 1997]. Contudo, os mecanismos utilizados pelos sistemas de quórum para a construção dos quóruns de leitura e de escrita apresentam vulnerabilidades e podem ser suscetíveis a alguns ataques por parte de nós maliciosos. O *comportamento egoísta* de alguns nós, por exemplo, pode explorar a natureza colaborativa dos protocolos epidêmicos que são utilizados frequentemente pelos sistemas de quórum. Isso acontece quando nós egoístas se recusam a executar a sua parte no andamento do sistema. Outra vulnerabilidade é a *modificação do intervalo de tempo* entre a propagação das escritas nos quóruns. O intervalo de tempo determina a rapidez da propagação de uma escrita e caso um nó atrase o envio, o dado será escrito de forma mais demorada nos quóruns. Uma outra vulnerabilidade ligada a colaboração na rede é a disseminação dos dados nos quóruns. Os nós não têm mecanismos de verificação da veracidade dos dados encaminhados, e dessa forma, nós de má-conduta podem explorar tal vulnerabilidade *propagando dados inúteis* aos outros nós do sistema de quórum.

Um estudo do funcionamento de um sistema de quórum para MANETs diante de ataques de falta de cooperação e de temporização, desempenhados por nós egoístas, e ataques de injeção de dados, desempenhados por nós maliciosos, pode ser obtido em [Mannes et al. 2009]. A Figura 1 apresenta um resumo do impacto desses ataques na confiabilidade de dados replicados por sistemas de quórum em MANETs. Observa-se que os ataques têm um impacto maior quando são realizados nas escritas, independente do ataque utilizado. O ataque de cooperação e injeção de dados apresentam as maiores ameaças à confiabilidade dos dados, sendo que a perda na confiabilidade com ataques de injeção de dados pode chegar até 95%. Com essa confiabilidade, o sistema se torna incapaz de apoiar a distribuição de dados para os serviços de gerência de redes. Em contrapartida, o ataque de temporização não apresenta um grande impacto ao sistema, revelando uma resiliência inerente do sistema. Contudo, conforme o intervalo de atraso das propagações $T$ aumenta, os nós começam a se comportar como um nó egoísta, e dessa forma o impacto do ataque se assemelha ao ataque de falta de cooperação.

## 3. O esquema $QS^2$

O esquema $QS^2$ tem como objetivo criar quóruns de escrita e de leitura com participantes colaborativos, negando a participação de nós de má-conduta nas operações de um sistema de quórum na camada de gerência de dados, conforme a Figura 2. Características como

 ©2012 SBC — Soc. Bras. de Computação

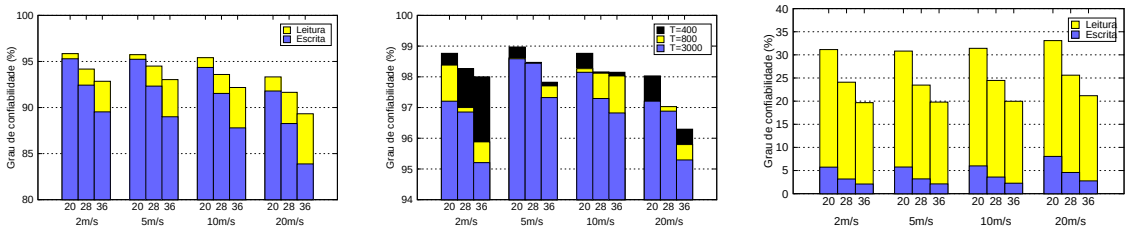



**Figura 1. Impacto dos ataques de falta de cooperação, temporização e injeção de dados na confiabilidade dos dados em um sistema de quórum para MANETs**

a auto-organização, a autonomia e o uso de poucos recursos são comumente encontradas em soluções bio-inspiradas, e desta forma, o $QS^2$ é inspirado nos sistemas biológicos de sensoriamento em quórum e de seleção por parentesco, presentes em bactérias. O *sensoriamento em quórum* é fundamentado na produção e na detecção de produtos químicos chamados autoindutores, que agem como um sinalizador da quantidade de bactérias no ambiente [Ng and Bassler 2009]. No entanto, esse mecanismo é vulnerável a bactérias egoístas e maliciosas, que não desejam ter o custo metabólico da produção de autoindutores. Para tolerar a ação de tais bactérias, uma das teorias aceitas é a *seleção por parentesco*, que define a interação entre bactérias compartilhando o mesmo material genético. Assim, as bactérias egoístas e maliciosas são excluídas do processo de sensoriamento.

Para indicar a qualidade da interação dos nós na rede, o $QS^2$ contabiliza autoindutores, que são as escritas *AI-W* e os encaminhamentos *AI-F* de dados através de um sistema de quórum, assim como é observado no mecanismo de sensoriamento em quórum encontrado nas bactérias [Mannes et al. 2012a]. A identificação e a contabilização dos autoindutores no esquema $QS^2$ acontecem por meio de informações enviadas juntamente com as mensagens de replicação dos dados em operações de leitura ou escrita. A quantidade de escritas ($qty_W$) e de encaminhamentos ($qty_F$) observados pelos nós da rede possibilitam a verificação de um comportamento anormal para o sistema.

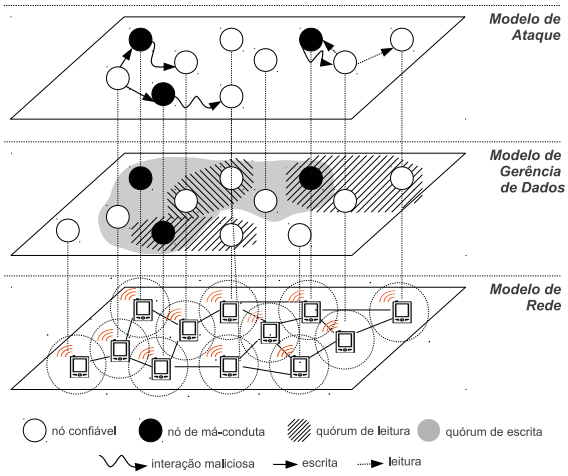



**Figura 2. Modelo em camadas**

O $QS^2$ é composto por dois módulos: o módulo de monitoramento dos nós e o módulo de decisão de cooperação, ilustrados na Figura 3. O *módulo de monitoramento dos nós* é responsável pela classificação dos nós em confiáveis ou de má-conduta. Ele é subdividido em dois componentes: o monitoramento do comportamento e a classificação dos nós. Já o *módulo de decisão de cooperação* determina a relação de cooperação entre

 ©2012 SBC — Soc. Bras. de Computação

dois nós, selecionando para interação somente aqueles que têm maior probabilidade de colaborar, e é composto pelo componente de seleção de nós. O $QS^2$ está presente em cada nó de uma MANET, e é utilizado sempre que um nó for requisitado para participar de um quórum. A Tabela 1 resume a notação[2] dos parâmetros do $QS^2$.

**Tabela 1. Notação do $QS^2$**

| Notação | Representação | Descrição |
|---|---|---|
| $k_{env}$ | $\sum_{x=0}^{\infty} x \frac{\lambda^x \times e^{-\lambda}}{x!} x$ | taxa de escritas esperada |
| $k_{enc}$ | $\sum_{i=0}^{n} k_{env_i}$ | taxa de encaminhamento esperada |
| $k_{env^{max}}$ | $\sum_{i=0}^{k_{env}} \frac{\lambda^{k_{env^{max}}} \times e^{-\lambda}}{k_{env^{max}}!} \leqslant \delta$ | $k_{env}$ máximo permitido, $\delta$ = prob. de $k_{env} < k_{env^{max}}$ |
| $k_{enc^{min}}$ | $\sum_{i=0}^{k_{enc}} \frac{\lambda^{k_{enc^{min}}} \times e^{-\lambda}}{k_{enc^{min}}!} \geqslant \gamma$ | $k_{enc}$ mínimo permitido, $\gamma$ = prob. de $k_{enc} > k_{enc^{min}}$ |

A Figura 4 ilustra a execução do $QS^2$ em uma operação de escrita em um sistema de quórum. Nesse exemplo, a rede é composta por 13 nós, sendo que o nó $A$ é a origem do dado a ser replicado. O nó $G$ é um nó de má-conduta, que pode não colaborar e perturbar a correta disseminação do dado. Cada nó possui uma lista de autoindutores, relacionada a cada um dos demais nós na rede que já utilizou esse nó em alguma operação no sistema de quórum. O funcionamento das etapas do $QS^2$ são descritos a seguir.

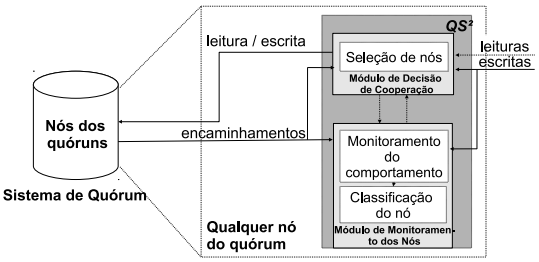
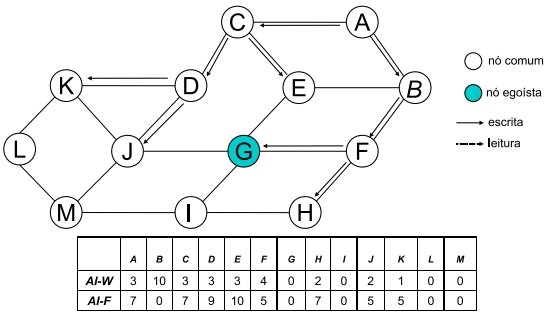



**Figura 3. Arquitetura do $QS^2$**




| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *AI-W* | 3 | 10 | 3 | 3 | 3 | 4 | 0 | 2 | 0 | 2 | 1 | 0 | 0 |
| *AI-F* | 7 | 0 | 7 | 9 | 10 | 5 | 0 | 7 | 0 | 5 | 5 | 0 | 0 |

**Figura 4. Funcionamento do $QS^2$**

**Monitoramento do comportamento** - ao iniciar uma operação de escrita no sistema de quórum, o nó $A$ anexa ao dado o seu identificador, iniciando uma rota de identificação do caminho percorrido pelo dado. A medida em que o dado é disseminado, os nós acrescentam seus próprios identificadores ao caminho percorrido pelo dado. Ao receberem o dado enviado por $A$, cada nó atualiza a contagem de *AI-W* para o nó $A$, a origem do dado, e de *AI-F* para cada nó presente no caminho percorrido.

**Classificação dos nós** - para classificar um nó em confiável, egoísta ou malicioso, o esquema $QS^2$ verifica a $qty_F$ e $qty_W$ enviada pelos nós e compara com o $k_{env^{max}}$ e $k_{enc^{min}}$ estabelecidos. Um nó é considerado egoísta se $qty_F < k_{enc^{min}}$ e malicioso se $qty_W > k_{env^{max}}$. Estando $qty_W$ e $qty_F$ dentro dos limites estabelecidos, o nó é assumido confiável. Por exemplo, supondo que a tabela apresentada seja do nó $K$, o nó $G$ apresenta valores de *AI-F* e *AI-W* diferente do esperado, sendo classificado como um nó de má-conduta.

**Seleção dos nós** - para iniciar uma operação no sistema, os nós dão preferência para aqueles que tenham sido classificados como confiáveis por eles. Apesar de não ser recomendável a escolha de nós classificados como egoístas ou maliciosos, a escolha do nó é independente e flexível para escolher quais nós quiser.

[2]Por razão de espaço, apenas as principais notações são apresentadas

 ©2012 SBC — Soc. Bras. de Computação

## 4. Avaliação do $QS^2$

O $QS^2$ foi implementado e simulado utilizando o simulador de redes *Network Simulator* (NS-2) versão 2.33[3]. O esquema foi desenvolvido e adicionado ao código do PAN [Luo et al. 2003], um sistema de quórum para MANETs, sendo chamado de $PAN+QS^2$. Todos os resultados são a média de 35 simulações, com um intervalo de confiança de 95% e são apresentados em pontos percentuais. A Tabela 2 resume os principais parâmetros. Foram empregadas quatro métricas na avaliação do $QS^2$ diante de nós de má-conduta:

**Tabela 2. Principais parâmetros de simulação dos cenários de validação**

| Parâmetros | Valor | Parâmetros | Valor |
|---|---|---|---|
| Quantidade de nós | 50 | Raio de transmissão | 250m |
| Quantidade de nós no StS | 25 | *Fanout (F)* | 2 nós |
| Tempo de vida da rede | 1500s | Intervalo de propagação (*T*) | 200, 400, 800, 3000ms |
| Área de movimentação | 1000x1000m | Nós de má-conduta (*f*) | 20%, 28%, 36% |
| Velocidades máximas | 2, 5, 10, 20m/s | Escrita ($k_{env^{max}}$) | 0,018 escritas/s |
| Tempo de pausa | 10, 20, 40, 80s | Encaminhamento ($k_{enc^{min}}$) | 0,15 encaminhamentos/s |

uma métrica de confiabilidade - *Grau de Confiabilidade* ($G_c$) e três métricas de eficiência - *Taxa de detecção* $Tx_{det}$, *Taxa de falsos negativos* $Tx_{fn}$ e *Taxa de falsos positivos* $Tx_{fp}$. A Tabela 3 descreve as fórmulas das métricas utilizadas.

**Tabela 3. Métricas utilizadas na validação do $QS^2$**

| Métricas | Representação | Descrição |
|---|---|---|
| $G_c$ | $\frac{\sum C_r}{\lvert R \rvert}$ | probabilidade de intersecção entre os quóruns de escrita e de leitura |
| $Tx_{det}$ | $\frac{\sum D_i}{\lvert A \rvert} \forall i \in A$ | porcentagem de interações de nós de má-conduta que foram detectadas |
| $Tx_{fn}$ | $\lvert A \rvert - Tx_{det}$ | falsos negativos obtidos na seleção de nós |
| $Tx_{fp}$ | $\frac{\sum D_i}{\lvert B \rvert} \forall i \in B$ | falsos positivos obtidos na seleção de nós |

Em cenários sem ataques, o desempenho do $PAN + QS^2$ mantém-se equivalente ao PAN, com o $G_c$ acima de 98%. Além disso, o $QS^2$ ainda proporcionou uma melhora no desempenho do PAN em velocidades mais altas [Mannes et al. 2012a]. Já na presença de 5, 7 e 9 nós desempenhando ataques de falta de cooperação (Figura 5(a)), temporização (Figura 5(b)) e injeção de dados (Figura 5(c)), o $PAN + QS^2$ sobrepõe a confiabilidade dos dados com relação ao PAN. O $PAN + QS^2$ provê uma confiabilidade acima de 90% diante desses ataques, enquanto que a confiabilidade do PAN é inferior a 30% diante de ataques de injeção de dados, por exemplo [Mannes et al. 2011]. O $PAN + QS^2$ apre-

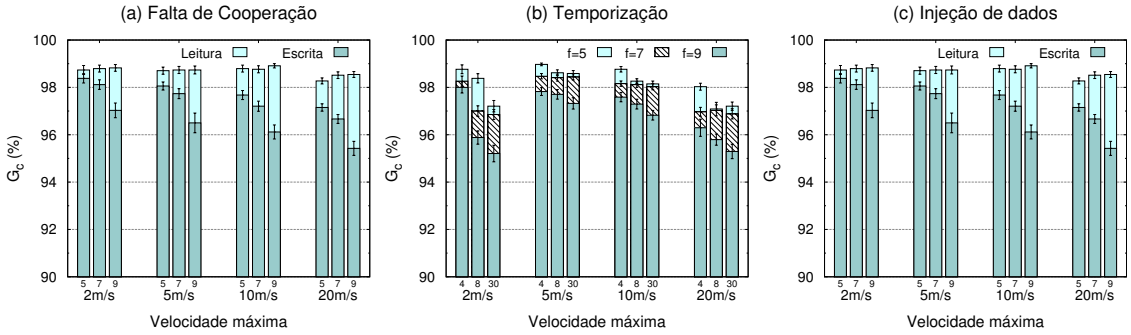



**Figura 5. Resultados da validação do $QS^2$ diante de nós de má-conduta**

senta uma variação pequena no $G_c$ em relação aos diferentes valores considerados para as velocidades. Isso acontece devido ao esquema de identificação e exclusão de nós de má-conduta, que não permite a participação desses nós nas operações de leitura e escrita, e

[3]O código do $QS^2$ e a sua documentação estão em http://www.nr2.ufpr.br/∼elisa/qs2_installation_pack.tar.gz

 ©2012 SBC — Soc. Bras. de Computação

indica uma vantagem do $PAN+QS^2$, pois mostra que a velocidade dos nós não interfere na execução das operações. Os resultados das métricas de eficiência mostram uma maior taxa de detecção dos nós egoístas (ataques de falta de cooperação e temporização), 98%, em comparação com os nós maliciosos (ataque de injeção de dados), 86%. Isso ocorre principalmente devido ao modo que o $QS^2$ age ao detectar nós de má-conduta, permitindo que eles recuperem seu bom comportamento ao passar do tempo. Apesar disso, a $Tx_{fp}$ se mantém baixa para ambos os ataques, 1,2% e 0,8% respectivamente.

## 5. Serviços de operação de rede confiáveis com o $QS^2$

Esta seção apresenta uma avaliação do $QS^2$ no suporte a cenários realísticos de uso das MANETs, para a disseminação de informações entre transeuntes em uma cidade e entre usuários de linhas de ônibus. Esses cenários representam o uso das MANETs para o apoio às atividades cotidianas. Maiores informações sobre os cenários, os parâmetros e os resultados estão em [Mannes et al. 2012b] e [Mannes et al. 2012c].

O cenário urbano corresponde ao centro de uma cidade, inspirado em [Becker et al. 2002]. Essa rede possibilita a disseminação de mensagens informativas entre os transeuntes e o comércio local, a fim de divulgar promoções, cardápios de restaurantes e novidades. Os resultados apresentados na Figura 6 são de uma rede de 150 nós que se movem entre 3 e 5 km/h, com um tempo de pausa entre 12 e 20 minutos. As simulações formam dois conjuntos: ataques de injeção de dados nas operações de escrita (5, 7 e 9 nós maliciosos) e ataques de falta de cooperação, temporização e injeção de dados agindo em conjunto (5, 10 e 15 nós de má-conduta). Os demais parâmetros e métricas são similares aos considerados na seção anterior. O uso do $QS^2$ permitiu um aumento

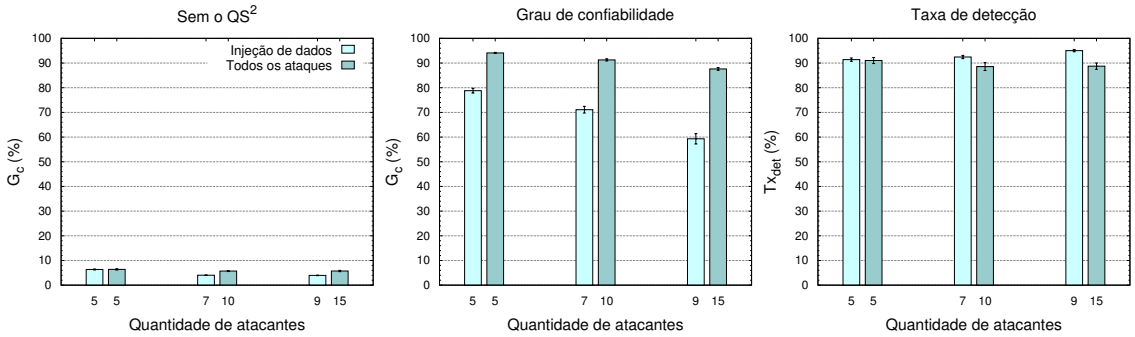



**Figura 6. Resultados do $QS^2$ em cenários de pedestres**

entre 70% e 80% na confiabilidade dos dados, que antes era de 10%. A $Tx_{det}$ foi maior do que a obtida com os cenários de validação anteriores, sendo superior à 90%. Devido à discrepância entre o $G_c$ mais baixo e a $Tx_{det}$ maior, suspeitou-se que o motivo para a queda do $G_c$ em tais cenários seria devido a dados desatualizados, e não a dados comprometidos. Para verificar essa suposição, foram quantificadas as taxas de leituras que obtiveram dados falsos ($Tx_{mis}$) e dados desatualizados ($Tx_{out}$). Para ambos os ataques, a suposição se mostrou verdadeira, sendo que a $Tx_{mis}$ é de apenas 12% contra 39% de $Tx_{out}$, em média.

Nos cenários de MANETs criadas em linhas de ônibus, o $G_c$ sem o uso do $QS^2$ diante de ataques também é inferior a 10%. Nesses ambientes, a confiabilidade alcançada com o $PAN+QS^2$ foi acima de 60% e a taxa de detecção do $QS^2$ reflete os resultados de $G_c$ obtidos, sendo cerca de 65%. Além disso, como no cenário anterior, cerca de 50%

 ©2012 SBC — Soc. Bras. de Computação

dos dados obtidos pelos nós são descartados porque não estão atualizados, e apenas cerca de 10% são dados falsos, injetados por nós maliciosos.

## 6. Conclusão

Este trabalho propôs um esquema bio-inspirado, chamado $QS^2$, para identificar a participação de nós de má-conduta na replicação por sistemas de quórum, que apoiam os serviços de operação de rede em MANETs. O esquema evita a escolha desses nós, enfatizando características de autonomia, auto-organização e utilização de poucos recursos. O $QS^2$ foi avaliado na presença de nós de má-conduta e os resultados obtidos mostram que a confiabilidade aumentou em até 87%. O $QS^2$ também foi aplicado no apoio aos sistemas de quórum em cenários urbanos realísticos, e proporcionam um aumento entre 50% e 85% na confiabilidade dos dados replicados. As contribuições deste trabalho resultaram nas publicações: [Mannes et al. 2009, Mannes et al. 2010, Mannes et al. 2011, Mannes et al. 2012a, Mannes et al. 2012b, Mannes et al. 2012c].

## Referências

Balasubramaniam, S., Leibnitz, K., Lio, P., Botvich, D., and Murata, M. (2011). Biological principles for future internet architecture design. *IEEE Communications Magazine*, 49(7):44 –52.

Becker, C., Bauer, M., and Hohner, J. (2002). Usenet-on-the-fly: supporting locality of information in spontaneous networking environments. In *Workshop on Ad Hoc Communications and Collaboration in Ubiquitous Computing Environments*. ACM Press.

Luo, J., Hubaux, J.-P., and Eugster, P. T. (2003). PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pages 1–12.

Malkhi, D. and Reiter, M. (1997). Byzantine quorum systems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pages 569–578.

Mannes, E., da Silva, E., and dos Santos, A. L. (2009). Analisando o desempenho de um sistema de quóruns probabilístico para MANETs diante de ataques maliciosos. In *Anais do IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg '09)*, pages 71–84.

Mannes, E., da Silva, E., Nogueira, M., and dos Santos, A. L. (2010). Implications of misbehaving attacks on probabilistic quorum systems for MANETs. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT)*, pages 189–195, Atenas, Grécia.

Mannes, E., Nogueira, M., and dos Santos, A. (2011). Um esquema bio-inspirado para tolerância á má-conduta em sistemas de quórum para MANETs. In *Anais do XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg '11)*, pages 239–252, Brasília, DF.

Mannes, E., Nogueira, M., and dos Santos, A. (2012a). A bio-inspired scheme on quorum systems for reliable services data management in MANETs. In *IEEE/IFIP Network Operations and Management Symposium (NOMS '12)*, pages 278 – 285, Maui, HI.

Mannes, E., Nogueira, M., and dos Santos, A. (2012b). Reliable operational services in MANETs by misbehavior-tolerant quorum systems. In *Proceedings of the 8th Conference on Network and Service Management (CNSM '12)*, pages 1–8, Las Vegas, EUA.

Mannes, E., Nogueira, M., and dos Santos, A. (2012c). Serviços confiáveis em MANETs baseado em sistema de quórum tolerante a má-conduta. In *Anais do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC '12)*, pages 305–318, Ouro Preto, MG.

Ng, W.-L. L. and Bassler, B. L. (2009). Bacterial quorum-sensing network architectures. *Annual Review of Genetics*, 43(1):197–222.

 ©2012 SBC — Soc. Bras. de Computação

# Assinatura digital Rabin-Williams sem randomização e com prova eficiente de segurança

Bernardo Caraponale Magri[1], Routo Terada[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo
Rua do Matão, 1010 - Butantã - São Paulo - SP

{bemagri,rt}@ime.usp.br

***Abstract.*** *With the development of public-key cryptography, many efforts were made to build encryption and signature protocols that were secure against malicious adversaries. The* Rabin *family of signature protocols has the speed records for verification of signature, being up to 100 times faster than* RSA. *This work presents a tight security proof (in the random oracle model) for a variant of the Rabin signature protocol presented by Bernstein, that does not require the use of* pseudo-random *bits, making the protocol more robust. The proof presented here is a polynomially tight reduction for the problem of integer factorization to the problem of* breaking *the* Principal Rabin-Williams $B = 0$ *protocol.*

***Resumo.*** *Com o surgimento da criptografia de chave pública, muito esforço foi feito para a criação de protocolos de assinatura que fossem seguros contra indivíduos maliciosos. A família de protocolos de assinatura* Rabin *possui os recordes de velocidade de verificação da assinatura, chegando a ser até 100 vezes mais rápida do que o* RSA. *Este trabalho apresenta uma redução eficiente de segurança para uma variante do protocolo de assinatura Rabin descrito por Bernstein, onde não é necessário o uso de nenhuma função para geração de bits* pseudo-aleatórios*, o que torna o protocolo mais robusto. A redução apresentada é uma redução polinomial e eficiente do problema da fatoração de inteiros para o problema de* quebrar *o protocolo* Principal Rabin-Williams $B = 0$.

## 1. Introdução

Os objetivos das assinaturas digitais são a autenticação do remetente, não-repudiação, e garantia da integridade da mensagem.

Para um protocolo de assinatura ser utilizado na prática ele precisa ser eficiente em seu tempo de execução (tanto para realizar uma assinatura quanto para verificar a autenticidade da mesma), e precisa garantir um certo nível de segurança contra possíveis falsificações. Para alcançar estes requisitos, os projetistas dos protocolos estabelecem uma escolha (*trade-off*) entre eficiência e segurança, o que depende do cenário em que o protocolo será utilizado.

Depois do início da era da criptografia de chave pública, surgiram vários novos problemas criptográficos, e vários protocolos de criptografia e assinatura. Inicialmente, esses protocolos eram considerados seguros quando resistiam a vários ataques durante alguns anos. O que normalmente ocorria é que esses protocolos eram quebrados depois de alguns anos

 ©2012 SBC — Soc. Bras. de Computação

de tentativas [Chor and Rivest 1988].
Logo depois começaram a aparecer algumas *provas de segurança*, que utilizavam técnicas da *teoria da complexidade computacional*, onde um problema reconhecidamente difícil era *reduzido* polinomialmente, através de algoritmos de redução, para o problema de se quebrar o protocolo. Se essa redução fosse possível (e polinomial) então poderíamos afirmar que o problema de se quebrar o protocolo é pelo menos tão difícil quanto o problema reconhecidamente difícil (por um fator polinomial).
Bellare e Rogaway popularizaram em [Bellare and Rogaway 1993] um modelo para *facilitar* a criação de reduções de segurança para protocolos, que ficou conhecido como *modelo do oráculo aleatório*. Esse modelo assume que a função de hash utilizada pelo protocolo possui uma distribuição uniforme.
Em [Bellare and Rogaway 1996] foi iniciado o estudo de reduções *eficientes* de segurança, onde o fator polinomial que relaciona os dois problemas é *pequeno*, e a probabilidade de se quebrar o problema reconhecidamente difícil é *quase* igual a probabilidade de se quebrar o protocolo, em *quase* o mesmo tempo de execução. Essa linha de pesquisa ficou conhecida como *segurança exata* ou *segurança concreta*.
No mesmo artigo [Bellare and Rogaway 1996] foi apresentado o primeiro protocolo de assinatura com uma redução *eficiente* de segurança, chamado PSS.
Michael Rabin introduziu em [Rabin 1979], inspirado pelo RSA, um novo problema criptográfico, que consistia na dificuldade de encontrar raízes quadradas módulo um número composto. A carecterística mais importante do problema criptográfico *Rabin* é que ele é equivalente ao problema da fatoração, isto é, se for possível obter raízes quadradas módulo um número composto, então é possível obter a fatoração do módulo composto.
O protocolo de assinatura descrito por Rabin [Rabin 1979] é mais rápido do que o protocolo de assinatura RSA (podendo chegar a ser até 100 vezes mais rápido), pois no caso do *Rabin*, é necessário apenas uma multiplicação modular para a verificação de uma assinatura.
Dentre os protocolos de assinatura que apresentam provas de segurança, os protocolos baseados em *Rabin* são os mais eficientes. A exceção é apenas em casos onde o tamanho (em bits) de uma assinatura é mais importante do que a velocidade de verificação da assinatura, e nesse caso, outros protocolos seriam mais eficientes, como por exemplo o protocolo ElGamal, que é baseado no problema do logaritmo discreto.
Williams [Williams 1980] utilizou o problema criptográfico proposto por Rabin e criou uma variação do protocolo de assinatura *Rabin*, que ficou conhecido por *Rabin-Williams*. O que Williams fez foi estender o protocolo proposto por Rabin, eliminando a limitação de ser possível assinar apenas $1/4$ das mensagens existentes.
Bernstein [Bernstein 2003] apresentou uma prova de segurança eficiente para uma variação do protocolo de assinatura *Rabin-Williams*, e em seguida [Bernstein 2008] estendeu seu próprio resultado, apresentando uma prova de segurança eficiente para uma outra variação do protocolo de assinatura *Rabin-Williams*, que não necessita de bits aleatórios concatenados à mensagem. O protocolo descrito em [Bernstein 2008] é considerado por Bernstein o estado da arte em protocolos eficientes de assinatura.

## 2. Motivação e Contribuição

O protocolo *Principal Rabin-Williams B = 0* é mais eficiente do que o protocolo analisado em [Bernstein 2008], portanto, a motivação para este trabalho foi encontrar uma redução

 ©2012 SBC — Soc. Bras. de Computação

eficiente de segurança para o protocolo de assinatura *Principal Rabin-Williams B = 0*, podendo assim comprovar sua segurança, relacionando o problema de *quebrar* o protocolo com o problema da fatoração.
A contribuição deste trabalho é a prova de segurança eficiente (no modelo do oráculo aleatório) para o protocolo de assinatura *Principal Rabin-Williams B = 0*.

## 3. O protocolo de Assinatura *Rabin-Williams*

A idéia por trás do protocolo de assinatura *Rabin-Williams* é estender o domínio das mensagens que podem ser assinadas. Isso é possível quando selecionamos os valores $p$ e $q$ de uma forma especial.
Uma assinatura *Rabin-Williams* de uma mensagem $m$ é a tripla $(e, f, s)$, chamada de *raiz quadrada modificada* [1] de $H(m) \bmod pq$, tal que $efs^2 \equiv H(m) \pmod{pq}$. Diferentemente da assinatura *Rabin*, todas as mensagens possuem uma assinatura válida. O protocolo *Rabin-Williams* é o protocolo mais eficiente conhecido, em termos de velocidade de verificação de assinaturas. Vamos citar algumas características que o tornam tão eficiente.

**Extensão do domínio de mensagens assináveis.** Em contraste à assinatura *Rabin* [Rabin 1979], que permite assinar apenas $1/4$ das mensagens do domínio, a assinatura *Rabin-Williams* permite que todas as mensagens do domínio sejam assináveis, mesmo sem precisar concatenar qualquer bit aleatório à mensagem, o que a torna mais indicada ao uso em aplicações *práticas*.

**Função de hash.** O protocolo *Rabin-Williams* não assina diretamente a mensagem $m$, o que é assinado é um valor $h = H(m)$, que é o resultado da mensagem $m$ aplicada a uma função de hash $H : \{0,1\}^* \rightarrow \{0,1\}^K$. Basicamente, o papel dessa função de hash é reduzir o tamanho da mensagem $m$ para um número de bits $K$ especificado. A função de hash é essencial para a segurança do protocolo, e se não utilizarmos uma função de hash, o protocolo pode ser facilmente quebrado por um ataque de mensagem escolhida, o que resultaria em uma falsificação existencial [Goldwasser et al. 1988].

**Expoente 2.** O protocolo *Rabin* foi o primeiro a usar o expoente $2$ como chave pública. Isso torna o processo de verificação de assinatura muito mais rápido do que se utilizado um expoente público *grande* (como no caso do *RSA*), pelo fato de a verificação necessitar de uma simples multiplicação modular.

**Assinaturas determinísticas.** Foi apresentado em [Katz and Wang 2003] o primeiro protocolo de assinatura determinística com uma prova eficiente de segurança. Antes desse resultado, acreditava-se que era necessário utilizar bits *pseudo-aleatórios* concatenados à mensagem na entrada da função de *hash* para obter uma prova eficiente de segurança [Coron 2002]. A vantagem das assinaturas determinísticas é que a mesma mensagem irá gerar sempre a mesma assinatura, eliminando totalmente a possibilidade de uma colisão

[1] do termo em inglês *tweaked square root*

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Relação entre os parâmetros $e, f$ e o símbolo de *Legendre***

| | $\left(\frac{h}{q}\right) = 1$ | $\left(\frac{h}{q}\right) = -1$ |
|---|---|---|
| $\left(\frac{h}{p}\right) = 1$ | $e = 1$<br>$f = 1$ | $e = -1$<br>$f = 2$ |
| $\left(\frac{h}{p}\right) = -1$ | $e = 1$<br>$f = 2$ | $e = -1$<br>$f = 1$ |

entre assinaturas de mensagens distintas.
Esta característica torna o protocolo ainda mais robusto e eficiente, pois não é necessária a geração de bits *pseudo-aleatórios* para computar a assinatura das mensagens.

### Parâmetros do Protocolo

Os usuários do protocolo compartilham três parâmetros, que serão descritos abaixo.

(i) um inteiro $K \geq 1024$.
(ii) uma função de hash criptográfica $H : \{0,1\}^* \to \{0,1\}^K$.
(iii) uma distribuição $D$ de pares de primos $p$ e $q$, sendo $p \equiv 3 \bmod 8$, $q \equiv 7 \bmod 8$, $p \approx q$, tal que $2^K < pq < 2^{K+1}$.

**Geração de chaves.** A geração das chaves do protocolo baseia-se em encontrar dois números primos aleatórios (com uma distribuição $D$) $p$ e $q$, tal que $p \equiv 3 \bmod 8$ e $q \equiv 7 \bmod 8$. A chave pública será $n = pq$, e a chave privada será o par $(p, q)$.

### Distribuição de assinaturas

Sabemos que todo $h = H(m)$ possui quatro raízes quadradas módulo $n = pq$. O protocolo precisa determinar qual das quatro raízes será devolvida como sendo a assinatura da mensagem $m$. Vamos considerar 2 formas utilizadas na literatura.

(i) **Não-estruturado:** O assinante escolhe uma *raiz quadrada modificada* de $h = H(m)$ uniforme e aleatória, independente de qualquer escolha anterior.
(ii) **Principal:** O assinante escolhe a *raiz quadrada principal modificada* de $h = H(m)$.

### Assinatura Principal

A assinatura Principal *Rabin-Williams* de uma mensagem $m$ é a tripla $(e, f, s)$, sendo que:

(i) $h = H(m)$.
(ii) $efs^2 \equiv h \pmod{pq}$.
(iii) $e = 1$ se $h$ é um resíduo quadrático módulo $q$, caso contrário $e = -1$.
(iv) $f = 1$ se $eh$ é um resíduo quadrático módulo $p$, caso contrário $f = 2$.
(v) $s$ é um resíduo quadrático módulo $n = pq$.

 ©2012 SBC — Soc. Bras. de Computação

A tabela 1 mostra a relação entre o símbolo de Legendre e os parâmetros $e$ e $f$ da assinatura.
A vantagem da assinatura *Principal*, é que não é necessário guardar nenhuma informação sobre as mensagens previamente assinadas, e também não é necessário nenhum tipo de randomização (para $B = 0$), o que torna o protocolo mais eficiente (não precisa gerar números aleatórios) e mais seguro contra ataques específicos (i.e ataques ao gerador de números *aleatórios*).
O parâmetro $B$ representa a quantidade de bits aleatórios (no nosso caso $0$) anexados à mensagem antes de obter o valor da função de *hash* $H$.

**Algoritmo 3.1.** [Bernstein 2008, p. 19] Assinatura Principal *Rabin-Williams* da mensagem $m$ sob a chave privada $(p, q)$:

1. Calcular $h \leftarrow H(m)$;
2. Calcular $u \leftarrow h^{(q+1)/8} \bmod q$;
3. Se $u^4 - h \bmod q = 0$ então $e \leftarrow 1$, senão $e \leftarrow -1$;
4. Calcular $v \leftarrow (eh)^{(p-3)/8} \bmod p$;
5. Se $v^4(eh)^2 - (eh) \bmod p = 0$ então $f \leftarrow 1$, senão $f \leftarrow 2$;
6. Calcular $w \leftarrow f^{(3q-5)/8} \cdot u \bmod q$;
7. Calcular $x \leftarrow f^{(9p-11)/8} \cdot v^3(eh) \bmod p$;
8. Calcular $y \leftarrow w + q(q^{p-2}(x - w) \bmod p)$;
9. Calcular $s \leftarrow y^2 \bmod pq$;
10. Retorna $(e, f, s)$;

A saída esperada do algoritmo 3.1 é a assinatura *principal* da mensagem $m$.

## 4. Prova eficiente de segurança

Nesta seção, iremos apresentar a nossa nova idéia, que possibilita uma prova eficiente de segurança para o protocolo *Principal Rabin-Williams B = 0*. Essa idéia é uma variação do que foi apresentado em [Bernstein 2008].

**Definição 4.1** (*Inversão Existencial Principal*)**.** Suponha que temos uma chave pública $n = pq$ com uma distribuição uniforme, e $Q+1$ elementos $h_1, h_2, ..., h_{Q+1} \in \{1, 2, 3, ...2^K\}$ aleatórios, com uma distribuição uniforme, e independentes uns dos outros. Podemos fazer $Q_i$ consultas a um oráculo de assinatura, que irá nos responder com a *raiz quadrada principal modificada* $(e_i, f_i, s_i)$ de $h_i \bmod n$. Nossa tarefa é gerar uma saída $(e_i, f_i, s_i)$ tal que $e_i f_i s_i^2 \equiv h_i \pmod n$, e $(e_i, f_i, s_i)$ seja a *raiz quadrada principal modificada* de $h_i$, e $h_i$ não foi uma das consultas feitas ao oráculo de assinatura.

Vamos assumir que existe um algoritmo eficiente $A(n, h_1, h_2, ..., h_{Q+1})$ que resolve o problema da *Inversão Existencial Principal* com uma probabilidade $Pr(A)$. Abaixo vamos descrever um outro problema semelhante.

**Definição 4.2** (*Inversão Existencial*)**.** Suponha que temos uma chave pública $n = pq$ com uma distribuição uniforme, e $Q + 1$ elementos $h_1, h_2, ..., h_{Q+1} \in \{1, 2, 3, ...2^K\}$ aleatórios, com uma distribuição uniforme, e independentes uns dos outros. Podemos fazer $Q_i$ consultas a um oráculo de assinatura, que irá nos responder com uma *raiz quadrada principal modificada* $(e_i, f_i, s_i)$ de $h_i \bmod n$. Nossa tarefa é gerar uma saída $(e_i, f_i, s_i)$ tal que $e_i f_i s_i^2 \equiv h_i \pmod n$, e $h_i$ não foi uma das consultas feitas ao oráculo de assinatura.

 ©2012 SBC — Soc. Bras. de Computação

Vamos assumir que existe um algoritmo eficiente $B(n, h_1, h_2, ..., h_{Q+1})$ que resolve o problema da *Inversão Existencial* com uma probabilidade $Pr(B)$.

Esse problema difere do problema da*Inversão Existencial Principal* apenas na saída esperada, que agora é uma *raiz quadrada modificada* aleatória de algum $h_i$ que não foi consultado ao oráculo de assinaturas. Usando o algoritmo $A$ com $Pr(A) = 1$ podemos construir um algoritmo $B$ com $Pr(B) = 1$ da maneira descrita a seguir.

**Redução 4.3.** Execute o algoritmo $A$ respondendo as consultas de assinatura de cada mensagem $m_i$ com a *raiz quadrada principal modificada* de $h_i = H(m_i) \bmod n$. Eventualmente $A$ irá retornar $(e, f, s)$ que é a *raiz quadrada principal modificada* de algum $h_i$ que não tenha sido assinado.

Usando o algoritmo $B$ com $Pr(B) = 1$ podemos construir um algoritmo $A$ com $Pr(A) = 1/2$ da maneira descrita a seguir.

**Redução 4.4.** Execute o algoritmo $B$ respondendo as consultas de assinatura de cada mensagem $m_i$ com a *raiz quadrada principal modificada* de $h_i = H(m_i) \bmod n$. Eventualmente $B$ irá retornar $(e, f, s)$ que é alguma *raiz quadrada modificada* aleatória de algum $h_i$ que não tenha sido assinado. Como sabemos que cada $h_i$ possui exatamente *quatro* raízes quadradas modificadas módulo $n$, a probabilidade de $(e, f, s)$ ser a *raiz quadrada principal modificada* de $h_i$ é *apenas* uma em quatro, mas também podemos obter a *raiz quadrada principal modificada* no caso em que for retornado sua inversa, portanto a probabilidade é $1/2$.

Abaixo é apresentado um diagrama mostrando as reduções para a prova de segurança. A flecha à direita significa que o problema da esquerda é reduzido para o problema da direita, utilizando a redução especificada entre parênteses.

InversãoExistencialPrincipal(A) $\longrightarrow$ InversãoExistencial(B)

InversãoExistencial(B) $\longrightarrow$ InversãoExistencialPrincipal(A)

Fatoração(B) $\longrightarrow$ InversãoExistencial(B) $\longrightarrow$ InversãoExistencialPrincipal(A)

Com essas reduções, mostramos que a existência de um algoritmo $A$ que resolve o problema da *Inversão Existencial Principal* implica a existência de um algoritmo $B$ que resolve o problema da *Inversão Existencial*, da mesma forma que a existência de um algoritmo $B$ que resolve o problema da *Inversão Existencial* implica a existência de um algoritmo $A$ que resolve o problema da *Inversão Existencial Principal*.
Reduzindo o problema da fatoração para qualquer um dos problemas acima, de uma forma eficiente, estaremos demonstrando que o problema de gerar uma falsificação de assinatura no protocolo *Principal Rabin-Williams B = 0* está fortemente relacionado ao problema da fatoração.

Vamos construir abaixo uma redução chamada *Fatorar*, que utiliza o algoritmo $B$ para obter uma fatoração do módulo $n$ com uma probabilidade $Pr(\textit{Fatorar}(B)) \geq PrInversaoExistencial(B)/2$.

 ©2012 SBC — Soc. Bras. de Computação

O parâmetro $Q$ representa a quantidade de assinaturas que o adversário poderá obter, que na literatura [Bellare and Rogaway 1996] é considerado aproximadamente $2^{30}$.

**Redução 4.5** (*Fatorar($B$)*)**.** Recebe com entrada o módulo $n$.

1. Para cada $i \in \{1, 2, ..., Q+1\}$
2. Gerar aleatoriamente com uma distribuição uniforme a tripla $(e'_i, f'_i, x_i) \in \{-1, 1\} \times \{1, 2\} \times \{0, 1, 2, ..., n-1\}$.
3. Calcular $g_i = mdc(x_i, n)$.
4. Se $g_i \equiv 7 \bmod 8$ então $e_i = 1$, senão $e_i = e'_i$.
5. Se $g_i \equiv 3 \bmod 8$ então $f_i = 1$, senão $f_i = f'_i$.
6. Calcular $s_i = x_i^2 \bmod n$.
7. Calcular $h_i = e_i f_i s_i^2 \bmod n$.
8. Volte ao passo 2 se $h_i \notin \{1, 2, ..., 2^K\}$.
9. Calcular $(j, e', f', s') = B(n, h_1, h_2, ..., h_{Q+1})$ usando $(e_i, f_i, s_i)$ para responder as consultas de assinatura do algoritmo $B$
10. Se $mdc(n, s' - s_j) \notin \{1, n\}$ imprima e pare.
11. Se $mdc(n, s') \notin \{1, n\}$ imprima e pare.

O tempo de execução da redução *Fatorar($B$)* é quase igual ao tempo de execução do algoritmo $B$, sendo necessário algumas operações módulo $n$ para gerar cada $h_i$, que são repetidas em média 2 vezes a cada iteração do laço, portanto o tempo da redução *Fatorar($B$)* é $< 2(Q+1) \cdot t(B)$.
O que ocorre no passo 9 é que o algoritmo $B$ irá retornar $(j, e', f', s')$ que é uma *raiz quadrada modificada* aleatória de $h_j$. Como o algoritmo já havia gerado a *raiz quadrada principal modificada* de $h_j$ temos duas *raízes quadradas modificadas* de $h_j$, sendo possível obter uma fatoração no frequente caso em que $s' \not\equiv s_j$ e $s' \not\equiv -s_j$, o que ocorre com uma probabilidade $1/2$.
Com a redução apresentada, podemos afirmar que uma redução do problema da fatoração para o problema da *Inversão Existencial Principal* ocorre com probabilidade $1/4$, pois temos que o algoritmo *Fatorar($B$)* obtém uma fatoração com probabilidade $1/2$, e temos que uma redução do problema da *Inversão Existencial* para o problema da *Inversão Existencial Principal* tem uma probabilidade de $1/2$.

Este trabalho apresenta uma prova eficiente (*tight*) de segurança no modelo do oráculo aleatório para o protocolo *Principal Rabin-Williams B = 0*.
Obtivemos essa prova eficiente de segurança demonstrando que o problema de encontrar a *raiz quadrada principal modificada* de um $h$ aleatório $\bmod n$ é equivalente ao problema de encontrar qualquer *raiz quadrada modificada* de um $h$ aleatório $\bmod n$. Portanto, o problema de se obter uma *raiz quadrada principal modificada* (em outras palavras, obter uma falsificação de assinatura) é pelo menos tão difícil quanto obter a fatoração do módulo $n$.
O protocolo *Principal Rabin-Williams B = 0* é considerado o *estado-da-arte* em protocolos de assinatura eficientes na velocidade de verificação [Bernstein 2008]. Esse resultado fortalece a escolha do protocolo *Principal Rabin-Williams B = 0* para o uso em aplicações práticas que necessitam de velocidade para a verificação de assinaturas.

Portanto, devido ao resultado obtido, podemos afirmar que o protocolo *Principal Rabin-Williams B = 0* é existencialmente seguro contra ataques adaptativos de mensagem escolhida.

 ©2012 SBC — Soc. Bras. de Computação

## Referências

Bellare, M. and Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73.

Bellare, M. and Rogaway, P. (1996). The exact security of digital signatures - how to sign with rsa and rabin. In *EUROCRYPT*, pages 399–416.

Bernstein, D. J. (2003). Proving tight security for standard rabin-williams signatures.

Bernstein, D. J. (2008). Proving tight security for rabin-williams signatures. In *EUROCRYPT*, pages 70–87.

Chor, B. and Rivest, R. L. (1988). A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5):901–909.

Coron, J.-S. (2002). Optimal security proofs for pss and other signature schemes. In *EUROCRYPT*, pages 272–287.

Coron, J.-S., Naccache, D., and Stern, J. P. (1999). On the security of rsa padding. In *CRYPTO*, pages 1–18.

Goldwasser, S., Micali, S., and Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308.

Katz, J. and Wang, N. (2003). Efficiency improvements for signature schemes with tight security reductions. In *ACM Conference on Computer and Communications Security*, pages 155–164.

Rabin, M. O. (1979). *Digitalized signatures and public-key functions as intractable as factorization*. MIT Laboratory for Computer Science.

Rivest, R. L., Shamir, A., and Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126.

Williams, H. C. (1980). A modification of the rsa public-key encryption procedure (corresp.). *IEEE Transactions on Information Theory*, 26(6):726–729.

 ©2012 SBC — Soc. Bras. de Computação

# Protocolos de acordo de chaves baseados em emparelhamentos, para dispositivos móveis

**Cleber M. Okida**[1], **Orientador: Routo Terada**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
Rua do Matão 1010 – Cidade Universitária – 05.508-090 – São Paulo – SP – Brazil

{cleberok,rt}@ime.usp.br

***Abstract.*** *This article describes a software implementation, using Elliptic Curves Cryptography and Pairings-based Cryptography for mobile devices. This work have focus on building a system providing privacy and security of information and integrity of data collection from medical records that travels in an insecure channel. We used a certificateless authenticated key agreement protocol, with BN curves to reach the desired security level. The main contributions of this work are a library in Java to build this cryptosystem and implementation of protocols and security system for mobile devices.*

***Resumo.*** *Neste trabalho é descrito uma implementação em software, utilizando Criptografia de curvas elípticas e baseada em emparelhamentos bilineares para dispositivos móveis. Este trabalho possui foco na construção de um mecanismo provedor de segurança e integridade das informações sigilosas dos dados dos prontuários médicos trafegados em um canal inseguro. Utilizou-se um protocolo de acordo de chaves sem certificado mutuamente autenticado, sobre as curvas BN para atingir o nível de segurança desejado. Dentre as contribuições deste trabalho estão uma biblioteca escrita na linguagem Java para a implementação deste sistema e a implementação dos protocolos e do próprio sistema de segurança para os dispositivos móveis.*

## 1. Introdução

No Sistema Brasileiro de Saúde, cabe aos Centros de Saúde o papel de órgão provedor de assistência médica primária. Para que este papel seja cumprido com responsabilidade e eficácia, se mostrou fundamental a condução de programas públicos de atenção domiciliar.

Em 2008 foi iniciado o desenvolvimento de um sistema de captura de dados por dispositivos móveis, por exemplo "*smartphones*" e PDAs, capaz de auxiliar o trabalho dos profissionais de saúde em campo, para disponibilizar melhores ferramentas de TI para os profissionais de saúde. A estratégia permite uma mudança do paradigma de tratamento de doenças para o de promoção da saúde melhorando a qualidade do serviço de saúde prestado à população por meio da aproximação entre equipes de saúde e a comunidade.

O Projeto Borboleta[1] inicialmente utilizou dispositivos móveis para auxiliar os profissionais da saúde nos atendimentos domiciliares, como uma ferramenta móvel de apoio à coleta de informações para um Sistema Móvel de Prontuário Eletrônico, que

[1]Borboleta: A Mobile Telehealth System – `http://ccsl.ime.usp.br/borboleta`

 ©2012 SBC — Soc. Bras. de Computação

integra novas tecnologias e está fortemente baseado no conceito de acompanhamento da situação clínica dos pacientes. Posteriormente, o projeto teve o seu escopo ampliado para atender outras áreas de um centro de saúde, para agregar valor ao serviço de saúde e, em larga escala, permitir a redução de custos [Wang et al. 2003].

### 1.1. Motivação

Ao analisar este contexto, observou-se que o ambiente onde estes Protuários Eletrônicos estariam disponíveis seria bem definido e controlado. Dessa forma, surgiu a ideia de utilizar o modelo de Criptografia de Chaves Públicas Sem Certificados (*Certificateless*) [Al-Riyami e Paterson 2003].

Este modelo é composto pelo paradigma da infraestrutura de chaves públicas (ICP) e pela criptografia baseada em identidade (*ID-based*); propõe uma estrutura mais simples comparada às implementações tradicionais de chaves públicas que usam certificados digitais para gerenciar as chaves públicas e resolve o problema de custódia de chaves gerada na criptografia baseada em identidade.

Em 1999, foi apresentado a função emparelhamento bilinear utilizando um par de pontos $P$ e $Q$ de uma curva elíptica para construção de sistemas criptográficos. A partir desse estudo, surgiu a nova área de pesquisa chamada criptografia baseada em emparelhamentos (*Pairing based Cryptography* – PBC) [Sakai et al. 2000].

No Projeto Borboleta a segurança da informação que trafega entre dispositivos móveis torna-se inerentemente necessária. Como os PDAs possuem baixo poder de processamento e conexões de pequena largura de banda, ainda exigem algoritmos de alta velocidade e consumo de energia reduzida, com nível de segurança adequado.

Por isso estudamos métodos baseados em curvas elípticas, em especial protocolos de acordo de chaves que possibilitam a combinação de chaves para utilização em algoritmos simétricos, que tipicamente são mais eficientes que algoritmos assimétricos para cifragem, garantindo o sigilo dos dados trafegados [Terada 2008].

A garantia de segurança e privacidade na telemedicina móvel é especialmente importante devido a:

- A natureza dos dados sendo trocados é fundamentalmente diferente. Se não tratada adequadamente, alguns dos dados poderiam até mesmo ameaçar a vida, porque dados errados podem facilmente levar a um diagnóstico fatal.
- Embora não seja fatal, o roubo ou a revelação de informações médicas podem ser usados para cometer crimes e discriminar uma pessoa para fins de vendas de produtos e serviços. [Bates et al. 2003]

## 2. Objetivo

Neste cenário, definimos o objetivo principal do trabalho pela construção de um mecanismo provedor de segurança e integridade das informações sigilosas dos dados dos prontuários médicos.

Em vista que estes dados serão trafegados entre os dispositivos móveis e um servidor de banco de dados, gerenciador dos prontuários eletrônicos, este sistema necessita obter o equilíbrio entre uma segurança eficiente por parte dos dispositivos móveis e obter um alto nível de segurança por parte do servidor.

 ©2012 SBC — Soc. Bras. de Computação

## 3. Metodologia

Após analisar os vários tipos de esquemas (protocolos de cifragem, assinatura e cifrassinatura) e de criptografia de chave pública (ICP, *ID-based* e sem certificados), foi escolhido os esquemas de acordos de chaves sem certificados.

Pois, desejou-se evitar o custo da emissão de certificados por uma autoridade certificadora. Prevendo a expansão da aplicação para os outros centros de saúde gerenciados pela secretaria municipal de saúde, e sabendo que a estrutura autônoma destes centros de saúde requer o alto nível de confiança, na criação e no gerenciamento das chaves secretas, atendido pelo paradigma Sem Certificados.

Os acordos de chaves permitem a adaptação para armazenar menos segredos para poupar consumo de memória do dispositivo. As chaves geradas em cada sessão foram usadas como chave secreta de algoritmos simétricos para cifrar os dados que trafegam pelo canal inseguro.

De acordo com as normas [2] , este tipo de mecanismo requer autenticação mútua, isto é, autenticação do remetente e do destinatário. Assim utilizamos o acordo de chaves baseado em criptografia sem certificados de Lippold et al. (2009), demonstrado seguro para o problema computacional Diffie-Hellman bilinear [Lippold et al. 2009].

Desse modo, a segurança do sistema tinha como requisitos:

**Eficiência** - atender os requisitos de eficiência dentro das possibilidades dos dispositivos de baixo poder de processamento;

**Segurança** - ser capaz de prover um alto nível de segurança para manter os dados dos prontuários médicos sob sigilo;

**Portabilidade** - atender aos vários tipos de dispositivos disponíveis no mercado;

Porém, para possibilitar a construção deste sistema, foi necessário um levantamento do estado da arte de implementações criptográficas para esta finalidade.

No início da pesquisa, havia uma única biblioteca criptográfica em Java com suporte a curvas elípticas disponível, Bouncy Castle [3] que permitia a utilização em dispositivos móveis, porém não suportavam emparelhamentos.

Portanto, iniciou-se a confecção de uma biblioteca provedora de mecanismos para implementação de protocolos criptográficos que utilizassem curvas elípticas, emparelhamentos bilineares em Java para dispositivos móveis.

As operações de pré-requisitos para construção dessa ferramenta foram divididos em camadas como apresentamos a seguir:

- Operações com precisão arbitrária: inclui aritmética sobre números inteiros de tamanho arbitrário; nesta camada está localizada a classe `BigInteger` e `SecureRandom`, codificado para a plataforma JME;
- Operações sobre corpos finitos: definem-se as operações algébricas sobre corpos finitos de característica prima;

[2]Manual de Certificação para Sistemas de Registro Eletrônico em Saúde - disponível em `http://www.sbis.org.br/certificacao/Manual_Certificacao_SBIS-CFM_2009_v3-3.pdf`

[3]Bouncy Castle – `http://www.bouncycastle.org`

 ©2012 SBC — Soc. Bras. de Computação

- Curvas elípticas: definem-se as curvas elípticas supersingulares e ordinárias e suas operações;
- Emparelhamentos bilineares: cálculo dos emparelhamentos de Tate, Weil e Ate;
- Aplicações: é nessa camada onde são implementadas as aplicações; como exemplos são fornecidas duas codificações de protocolos de acordo de chaves.

Esta biblioteca foi chamada de **jMobilePBC**[4], e alguns dos seus aspectos de implementação serão discutidos a seguir.

### 3.1. Implementação

Até sua versão 2.5.2, a WTK para JME não contempla a implementação de uma classe para números inteiros de tamanho arbitrário. Por esse motivo, houve necessidade de gerar um gerenciador eficiente para esse tipo de estrutura.

Dentre as operações de números inteiros, a mais crítica é a multiplicação. Desse modo, foi implementado o algoritmo de Karatsuba otimizado para sistemas de 32 bits de tamanho da palavra [Chung e Hasan 2007].

A BigInteger aceita números de tamanho no mínimo 56 bits até 1024 bits, observa-se o acréscimo de 8 bits no intervalo. A classe `SecureRandom` foi criada com intuito de melhorar a eficiência dos sorteios dos valores aleatórios.

### 3.2. Corpos Finitos

Da mesma forma que no algoritmo RSA, os cálculos envolvidos em curvas elípticas necessitam da estrutura matemática de corpos finitos, representado como $\mathbb{F}_p$, onde $p$ é a característica do corpo.

Essa estrutura de corpo finito, assim como a estrutura de grupo, permite que trabalhemos a abstração de pontos sobre uma curva elíptica, como sendo uma estrutura de conjuntos [Martin 2010].

A partir da ideia didática e por simplificação, preferimos iniciar a implementação pelos corpos finitos de característica prima.

Assim, foi possível a implementação dos corpos e das estruturas de extensões sobre esses corpos, representadas pelo o expoente de cada corpo, como a extensão de grau 2 para o caso $\mathbb{F}_{p^2}$, por exemplo.

Os corpos finitos implementados são descritos a seguir:

- $\mathbb{F}_p$: representa o corpo finito de característica prima, onde $p$ é um primo ou uma potência de primo, a representação desse conjunto é um número inteiro;
- $\mathbb{F}_{p^2}$: caracteriza a extensão de grau 2 do corpo finito $\mathbb{F}_p$, sua representação é dada como um número complexo com parte real e imaginária, para tirar proveito da propriedade da multiplicação complexa;
- $\mathbb{F}_{p^{12}}$: é representa a extensão de grau 12 do corpo finito, que consiste em 6 elementos de $\mathbb{F}_{p^2}$.

Pode-se visar dois conjuntos a serem explorados nos criptossistemas baseados em emparelhamentos, pois o corpo de $\mathbb{F}_p$ em conjunto com a extensão $\mathbb{F}_{p^2}$ pode ser usada para

[4]Disponível em `svn://ccsl.ime.usp.br/lsd/trunk/jmobilepbc`

 ©2012 SBC — Soc. Bras. de Computação

definir os elementos das curvas supersingulares e as extensões $\mathbb{F}_{p^2}$ e $\mathbb{F}_{p^{12}}$ são usadas para definir os elementos das curvas ordinárias do tipo BN [Barreto e Naehrig 2005]. Porém, um desses conjuntos passa a não ser interessante quando analisado sob o aspecto de nível de segurança.

Os corpos de característica binária não foram implementados porque, apesar de serem eficientes para as operações aritméticas, são mais rápidas para serem atacadas, quando comparadas às de característica prima, além de viabilizarem apenas curvas supersingulares [Wiener e Zuccherato 1999].

### 3.3. Curvas Elípticas

As curvas elípticas podem ser de dois tipos:

- curvas subjacentes: são as curvas definidas sobre um corpo finito, e são da forma $E(\mathbb{F}_p) : y^2 = x^3+ax+b$, onde $a, b \in \mathbb{Z}_p$; tais curvas são necessárias para todo tipo de aplicação envolvendo curvas elípticas, elas estão definidas na classe `Curva`;
- curvas torcidas: são as curvas definidas sobre uma extensão de um corpo finito, e são da forma $E'(\mathbb{F}_{p^n}) : y^2 = x^3 + ax + b$, onde $a, b \in \mathbb{F}_{p^n}$ e $n$ é o grau da extensão do corpo, tais curvas são necessárias quando são empregados emparelhamentos bilineares assimétricos, e estão definidas na classe `Curva2`;

A partir do problema do logaritmo discreto (PLD), que diz: dado um ponto $P \in \mathbb{G}$ gerador do grupo $\mathbb{G}$ e um valor aleatório $a \in \mathbb{Z}_p$, e dado $aP \in G$, é difícil calcular $a$.

Abstraindo o conceito de grupo como um conjunto e considerando a característica discreta do sistema, as aplicações criptográficas têm se apoiado na dificuldade desse problema para esconder o segredo $a$.

Esta biblioteca permite, ao contrário da Bouncy Castle, flexibilidade ao implementar tipos diferentes de curvas subjacentes definida como supersingular ou ordinária. Ambas podem ser implementadas com a classe `Curva` da jMobilePBC, por estarem definidas sobre corpos de característica prima.

As curvas supersingulares são as curvas mais simples existentes, definidas sobre corpos de característica prima, binária ou ternária com grau de mergulho $k \leq 6$ e são aplicadas no emparelhamento de Weil e Tate [Menezes et al. 1991]. Entretanto, no universo das curvas elípticas, são poucas as curvas supersingulares que atendem o nível de segurança necessário.

A alternativa para as curvas supersingulares são as famílias especiais de curvas ordinárias ou também chamadas de não-supersingulares. Neste caso, o emparelhamento (Tate, por exemplo) recai sobre o corpo estendido $\mathbb{F}_{p^k}$ , onde $k$ é o grau de mergulho da curva.

Dentre as curvas ordinárias, as curvas BN constituem uma das classes mais versáteis de curvas elípticas amigáveis a emparelhamentos. Elas são conhecidas por facilitar o desenvolvimento de emparelhamentos para o nível de segurança de 128 bits.

Nossa implementação contempla as curvas supersingulares de grau de mergulho 2 e curvas BN de grau de mergulho 12.

 ©2012 SBC — Soc. Bras. de Computação

### 3.4. Emparelhamentos Bilineares

Um emparelhamento bilinear é uma função definida sobre curvas elípticas, onde temos três grupos cíclicos $\mathbb{G}_1$, $\mathbb{G}_2$ e $\mathbb{G}_T$ de ordem prima $q$. Um emparelhamento bilinear admissível, é uma aplicação bilinear $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ [Boneh e Franklin 2001].

Os emparelhamentos bilineares podem ser de dois tipos:

- Simétricos: onde $\mathbb{G}_1 = \mathbb{G}_2$, consequentemente temos a propriedade de $e(P, Q) = e(Q, P)$;
- Assimétricos: onde $\mathbb{G}_1 \neq \mathbb{G}_2$, que implica em $e(P, Q) \neq e(Q, P)$

Os emparelhamentos escolhidos para serem implementados foram os emparelhamentos de Weil, Tate e Ate, pois são os principais tipos existentes e são facilmente adaptáveis para os outros tipos, afim de obter maior eficiência.

Os emparelhamentos de Tate e Weil utilizam curvas supersingulares, sendo que podemos obter o emparelhamento de Weil a partir de duas iterações do emparelhamento de Tate, e este é obtido por meio de dois pontos $P, Q \in \mathbb{G}$, mapeado para um grupo $\mathbb{G}_T$, que é um `Elemento` de $\mathbb{F}_{p^2}$ para a curva subjacente.

Já o emparelhamento Ate, é uma generalização do emparelhamento de Tate, porém utiliza-se as curvas ordinárias.

O emparelhamento de Ate foi o único que permitiu atingir o nível de segurança necessário, pois permite que tamanho do contradomínio tenha 3072 bits, que equivale ao RSA de 3248 bits, quando utilizamos uma curva BN de 128 bits [Pereira et al. 2011].

Portanto, para a implementação do criptossistema do Projeto Borboleta foram escolhidas uma curva ordinária BN $E_3/\mathbb{F}_p : y^2 = x^3 + 3$, para permitir o nível de segurança de 128 bits, utilizando emparelhamentos Ate [Devegili et al. 2007].

### 3.5. Protocolos

Os sistemas "*Telehealth*" possuem como característica serem ambientes controlados e limitados. Uma vantagem observada na existência da comunicação entre usuários que acessam informações sigilosas de um servidor de banco de dados é a associação de um usuário a cada cliente e um associado ao servidor.

Muitos protocolos de acordos de chaves sem certificados foram propostos, mas nenhum apresentava um modelo de segurança adequado [Lippold et al. 2009].

Este protocolo é baseado em dois protocolos principais, o protocolo de acordo de chaves não interativo de Sakai et al. (2000) (SOK), por isso necessita de emparelhamentos simétricos e o protocolo de Krawczyk (2005) para prova da autenticação [Sakai et al. 2000] [Krawczyk 2005].

O artigo de Dupont e Enge (2002) generalizou o protocolo de SOK, fornecendo um método para converter um protocolo que utiliza emparelhamentos simétricos para um com emparelhamento assimétrico [Dupont e Enge 2002].

## 4. Resultados

Além desta modificação do protocolo de Lippold et al. (2009), outro protocolo foi proposto em 2009 com um número reduzido de operações, sendo que este obteve um desempenho médio em 22,3% mais eficiente que o original [Goya et al. 2010].

 ©2012 SBC — Soc. Bras. de Computação

Quando executamos o protocolo com as curvas BN com 128 bits, esta implementação executa emparelhamentos em $356,69$ ms que sobrepuja o tempo de execução completa do protocolo em $844,2$ ms em um processador Dual Core de 1,73GHz e 1GB de memória RAM. No PDA com um processador de 201MHz, 64MB de RAM e máquina vitual original fornecida pelo fabricante, executou em $1,947$ segundos.

## 5. Publicações

As publicações frutos deste trabalho são:

1. **Okida, C.**, Goya, D. e Terada, R. *Java Cryptographic Library for Smartphones* In *Revista IEEE América Latina*, Jan. 2012, Vol. 10, Issue 1, ISSN 1548-0992.
2. Goya, D., **Okida, C.** e Terada, R. *A two-party certificateless authenticated key agreement protocol*. In *SBSeg 2010 X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação, Fortaleza 2010.
3. Goya, D., **Okida, C.** e Terada, R. *Acordo de Chave com Autenticação sem Certificado Digital*. In *9th. International Information and Telecommunication Technologies Symposium, 2010*, Rio de Janeiro. Proceedings of the 9th I2TS, 2010.

O trabalho completo: Okida, C. M. (2011). *Protocolos de acordo de chaves baseados em emparelhamentos, para dispositivos móveis*. Master's thesis, Universidade de São Paulo, pode ser obtido em `www.ime.usp.br/~cleberok/mestrado/dissCleber.pdf`

## 6. Contribuições

As principais contribuições deste trabalho, foram:

1. Estrutura provedora de segurança para transmissão de prontuários médicos eletrônicos;
2. Protocolo de acordo de chaves mutuamente autenticado utilizando emparelhamentos assimétricos, demonstrado seguro baseado no protocolo de Lippold [Goya et al. 2010];
3. Biblioteca "*open source*" provedora de recursos para implementação de protocolos criptográficos.

## 7. Trabalhos Futuros

A biblioteca JMobilePBC é modesta e ainda está em fase de construção para atender os mais diversos tipos de curvas e emparelhamentos encontrados na literatura.

Existe um grande interesse de incluir implementações de assinatura sobre curvas elípticas e sobre emparelhamento bilineares, como por exemplo ECDSA e BLS, e outros tipos de protocolos como os baseados em identidade e sem certificados [Joye e Neven 2008].

Deseja-se aproveitar a versatilidade da biblioteca implementando corpos de características 3, para realizar um estudo comparativo com o emparelhamento de Tate para curvas destas características [Kawahara et al. 2007].

Como novas pesquisas, nós gostariamos de realizar testes com outras abordagens de coordenadas para curvas elípticas, por exemplo coordenadas de Edwards, com a finalidade de verificar a eficiência da estratégia adotada [Lauter et al. 2010].

 ©2012 SBC — Soc. Bras. de Computação

## 8. Agradecimentos

Esta pesquisa foi apoiada pela Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP – e Microsoft Research Institute, Brasil.

## Referências

Al-Riyami, S. S. e Paterson, K. G. (2003). Certificateless public key cryptography. In *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*. Springer. Cryptology ePrint Archive, Report 2003/126, `http://eprint.iacr.org/2003/126`.

Barreto, P. S. L. M. e Naehrig, M. (2005). Pairing-friendly elliptic curves of prime order. In *Proceedings of SAC 2005, volume 3897 of LNCS*, pages 319–331. Springer-Verlag.

Bates, D. W., Ebell, M., Gotlieb, E., Zapp, J. e Mullins, H. (2003). A proposal for electronic medical records in u.s. primary care. *Journal of the American Medical Informatics Association*, 10(1):1–10.

Boneh, D. e Franklin, M. K. (2001). Identity-based encryption from the weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229, London, UK. Springer-Verlag.

Chung, J. e Hasan, M. A. (2007). Asymmetric squaring formulae. In *Proceedings of the 18th IEEE Symposium on Computer Arithmetic*, pages 113–122, Washington, DC, USA. IEEE Computer Society.

Devegili, A. J., Scott, M. e Dahab, R. (2007). Implementing cryptographic pairings over barreto-naehrig curves, in: Pairing-based cryptography pairing 2007. *LNCS*, 4575:197–207.

Dupont, R. e Enge, A. (2002). Practical non-interactive key distribution based on pairings. Cryptology ePrint Archive, Report 2002/136. `http://eprint.iacr.org/2002/136`.

Goya, D., Okida, C. e Terada, R. (2010). A two-party certificateless authenticated key agreement protocol. In *SBSeg 2010 X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação.

Joye, M. e Neven, G. (2008). *Identity-Based Cryptography*. Volume 2 Cryptology and Information Security Series. IOS Press.

Kawahara, Y., Takagi, T. e Okamoto, E. (2007). Efficient implementation of tate pairing on a mobile phone using java. pages 396–405.

Krawczyk, H. (2005). Hmqv: A high-performance secure diffie-hellman protocol. In *CRYPTO*, pages 546–566. `http://eprint.iacr.org/2005/176`.

Lauter, K., Montgomery, P. L. e Naehrig, M. (2010). An analysis of affine coordinates for pairing computation. Cryptology ePrint Archive, Report 2010/363. `http://eprint.iacr.org/`.

Lippold, G., Boyd, C. e Nieto, J. (2009). Strongly secure certificateless key agreement. In *Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*. Springer. Cryptology ePrint Archive, Report 2009/219, `http://eprint.iacr.org/2009/219`.

Martin, P. A. (2010). *Grupos, Corpos e Teoria de Galois*. editora livraria da fisica.

 ©2012 SBC — Soc. Bras. de Computação

Menezes, A., Vanstone, S. e Okamoto, T. (1991). Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 80–89, New York, NY, USA. ACM.

Pereira, G. C. C. F., Simplício, Jr., M. A., Naehrig, M. e Barreto, P. S. L. M. (2011). A family of implementation-friendly bn elliptic curves. *J. Syst. Softw.*, 84:1319–1326.

Sakai, R., Ohgishi, K. e Kasahara, M. (2000). Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security (SCIS2000)*, pages 26–28, Okinawa, Japan. Inst. of Electronics, Information and Communication Engineers.

Terada, R. (2008). *Segurança de Dados - Criptografia em Redes de Computador*. Editora Edgard Blücher, São Paulo, SP, 2 edition.

Wang, S. J., Middleton, B., Prosser, L. A., Bardon, C. G., Spurr, C. D., Carchidi, P. J., Kittler, A. F., andDavid G. Fairchild, R. C. G., Sussman, A. J., Kuperman, G. J. e Bates, D. W. (2003). A cost-benefit analysis of eletronic medical records in primary care. *The American Journal of Medicine*, 114(5):397–403.

Wiener, M. e Zuccherato, R. (1999). Faster attacks on elliptic curve cryptosystems. In Tavares, S. e Meijer, H., editors, *Selected Areas in Cryptography*, volume 1556 of *Lecture Notes in Computer Science*, pages 631–631. Springer Berlin / Heidelberg.

 ©2012 SBC — Soc. Bras. de Computação

# Mineração de Dados para Detecção de Fraudes em Transações Eletrônicas

**Aluno Mestrado: José Felipe Jr.[1], Orientador: Wagner Meira Jr.[1], Co-Orientador: Adriano Pereira [2]**

[1]Univ. Federal de Minas Gerais (UFMG)
Dpto de Ciência da Computação - Belo Horizonte, MG, Brazil

[2]Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Dpto de Computação - Belo Horizonte, MG, Brazil

{felipe.junior, meira}@dcc.ufmg.br, adriano@decom.cefetmg.br

***Abstract.*** *Due to the growing popularity of the Web, there is an increasing number of people who perform e-business transactions. On the other hand, this popularity has also attracted the attention of criminals, as the growing number of fraud cases in Web and financial losses that reach billions of dollars per year demonstrate. This paper proposes a methodology, based on the knowledge discovery process, to detect fraud in online payment systems. In order to evaluate this methodology we define the concept of economical efficiency and applied it to an actual dataset of one of the largest Latin American electronic payment systems. The results show a very good performance for our proposal, providing gains of up to 46.5% in comparison with the strategy currently employed.*

***Resumo.*** *Com a popularização da Web, cresce o número de pessoas que a utilizam para realizar negócios e transações financeiras. Entretanto, essa popularização atrai a atenção de criminosos, aumentando o número de fraudes nesse cenário. As perdas financeiras chegam a bilhões de dólares por ano. Este trabalho propõe uma metodologia para detecção de fraude em pagamentos online baseado no processo de descoberta do conhecimento. Para avaliação, foi definido o conceito de eficiência econômica e aplicado em um conjunto de dados real de uma das maiores empresas latino-americanas de serviço de pagamentos eletrônicos. Os resultados mostram excelente desempenho na detecção de fraudes, com ganhos de até 46,46% em relação ao cenário atual da empresa.*

## 1. Introdução

Durante as últimas décadas tem-se observado uma popularização do *e-commerce*, o que tem atraído a atenção de criminosos devido ao volume financeiro associado. Segundo [Bhatla et al. 2003], a taxa de fraude na Internet é 12 a 15 vezes maior do que no "mundo físico", tornando as vendas pela *Web* uma maior ameaça para os comerciantes. De acordo com o [Mindware Research Group 2011], nas vendas *online* da América do Norte estima-se que a perda em 2011 foi de aproximadamente US$ 3,4 bilhões, um aumento de US$ 700 milhões em relação a 2010.

Portanto, a prevenção e a detecção de fraudes têm se mostrado essenciais. Nesse sentido, várias técnicas de mineração de dados têm sido utilizadas para o combate à fraude, mas, previamente à sua utilização é necessária a adequação dos dados às técnicas em termos de formato e natureza. Dessa forma, são necessárias metodologias que auxiliem na seleção, limpeza e preparação dos dados para que seja possível a utilização das técnicas de mineração de dados.

 ©2012 SBC — Soc. Bras. de Computação

O objetivo deste trabalho é desenvolver uma metodologia que possa auxiliar de forma eficaz o processo de detecção de fraude. Essa metodologia é uma instância do processo de *KDD (Knowledge Discovery in Database)* [Fayyad 1996] com inclusões, modificações e orientações que visam auxiliar os usuários a avaliar cenários de fraude na *Web* ou outros que apresentem características semelhantes. No passo de seleção de dados, é proposto um método para auxiliar na extração de dados de um banco de dados, tornando mais fácil esta tarefa e também é apresentado um método de auditoria para garantir a integridade e a coerência dos dados extraídos. No pré-processamento, algumas manipulações nos dados e testes são empregados para auxiliar na identificação de padrões de fraude e não fraude. Na transformação, a metodologia ajuda a encontrar algumas evidências de dados que possam identificar os melhores atributos para mineração de dados. Na etapa de mineração de dados, técnicas mais promissoras para detecção e entendimento da fraude são apresentadas. Finalmente, o último passo proporciona um método para avaliar a fraude e o seu impacto nas transações eletrônicas.

O restante deste trabalho é organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados e a Seção 3 realiza uma descrição da metodologia criada. A Seção 4 apresenta o estudo de caso e os resultados avaliados. A Seção 5 apresenta as conclusões e por fim, a Seção 6 apresenta as publicações obtidas com a pesquisa.

## 2. Trabalhos Relacionados

Devido à importância do problema de detecção de fraude, vários são os trabalhos associados a essa área. Existem pesquisas que buscam identificar as classes de fraudes e criam metodologias que permitam a sua classificação. O objetivo dessas pesquisas é compreender melhor o fenômeno das fraudes e identificar aquelas que compartilham alguma similaridade [Vasiu and Vasiu 2004, Chau et al. 2006, Fawcett and Provost 1997]. Pode-se perceber que devido às especificidades das fraudes, essas metodologias podem apresentar diferenças significativas em razão das particularidades de cada tipo de fraude. Entretanto, as técnicas de mineração de dados têm sido amplamente utilizadas na detecção independente da metodologia adotada. Isso se deve ao fato dessas técnicas permitirem a extração de informações úteis em bases com grande volume de dados.

[Phua et al. 2005] realiza uma pesquisa exploratória dos inúmeros artigos associados à detecção de fraude utilizando mineração de dados e apresenta métodos e técnicas juntamente com os seus problemas. Segundo o autor, esses algoritmos são baseados em algumas abordagens como: *(1) estratégia supervisionada* [Maes et al. 1993], em que algoritmos de aprendizagem examinam todas as transações, rotuladas previamente, para matematicamente determinar o perfil de uma transação fraudulenta e estimar o seu risco; *(2) estratégia não supervisionada* [Bolton and Hand 2002], em que os métodos dispensam o conhecimento prévio das transações fraudulentas e não fraudulentas detectando alterações no comportamento ou identificando transações não usuais; *(3) estratégia híbrida* [Williams and Huang 1997], onde estratégias não-supervisionadas têm sido utilizadas para segmentar os dados em grupos para as abordagens supervisionadas.

## 3. Metodologia

A metodologia para detecção de fraudes tem como objetivo criar modelos de classificação que auxiliem na identificação de registros anômalos. Essa metodologia tem como diferencial sua abrangência, pois aborda questões que vão desde a extração de dados para a formação de um *dataset*, passando pela avaliação de técnicas mais promissoras para identificação de fraude, até questões relacionadas ao retorno financeiro obtido com as

 ©2012 SBC — Soc. Bras. de Computação

técnicas. Ela tem como base o *KDD*, composto de 5 etapas que serão utilizadas para nortear a tarefa de detecção de fraude. Entretanto, todas as etapas sofreram modificações para auxiliar na instanciação para o problema da fraude. A seguir serão apresentadas as modificações feitas em cada etapa do processo.

A **Seleção de Dados** envolve a identificação de quais informações deverão ser efetivamente consideradas. Devido à complexidade e riscos associados na extração dos dados propõe-se um método de extração em 4 passos. *Passo 1:* Análise do banco e seleção de tabelas relevantes para detecção de fraude, feita em conjunto com os especialistas da área. *Passo 2:* Agrupamento (deve-se definir uma *tabela base* para cada grupo) e junções das tabelas selecionadas considerando as informações existentes. Como processo de auditoria deve-se contabilizar o número de registros existentes na *tabela base* e cada junção de uma tabela avalia-se a ocorrência de alteração no número de registros. Deve-se fazer também uma verificação amostral de alguns registros para analisar se as informações de relacionamento entre os dados se mantêm corretas. Os erros devem ser corrigidos. *Passo 3:* União dos grupos formados para criação de um *dataset* único, deve-se utilizar o processo de auditoria proposto. *Passo 4:* Extração do *dataset* gerado para o formato de arquivo texto.

No **Pré-Processamento** são aplicadas estratégias para tratamento dos dados de forma a eliminar inconsistências. Como diferencial propõe-se a utilização de análises de dispersão e histogramas para identificação de padrões de fraude. Se algum padrão é encontrado, isso pode fornecer indícios de que os atributos selecionados são significativos para o processo de detecção de fraude. Assim, permite-se verificar com antecedência se os atributos selecionados são relevantes para a detecção de fraude evitando a execução de toda a metodologia para que ocorra essa validação.

Na **Transformação** novas informações ou informações em formatos mais apropriados são geradas a partir das características originais. Para auxiliar nesta etapa alguns exemplos de variáveis para detecção de fraude são: categoria do estabelecimento, lista de estabelecimentos mais utilizados pelo cliente, valor médio das compras, situação do CPF, tempo de registro, número de casos de fraude e tempo desde a última transação [Gadi 2008].

Na **Mineração de Dados** são extraídos padrões ou modelos em grandes volumes de dados. No contexto de fraude, técnicas que sejam descritivas e preditivas permitem tanto a identificação das fraudes quanto o entendimento dos padrões adotados para identificá-las. As técnicas selecionadas que apresentam essas características são: *Árvore de Decisão*[Quinlan 1993], *RIPPER*[Cohen 1995], *Lazy Associative Classification (LAC)*[Veloso et al. 2006]. A técnica *Naive Bayes* foi escolhida devido a bons resultados obtidos em pesquisas na área [Phua et al. 2004]. *Boosting*[Freund and Schapire 1995], *Oversampling* e *Stacking*[Phua et al. 2004] foram técnicas escolhidas com o intuito de melhorar a qualidade da previsão dos classificadores definidos anteriormente.

Na **Interpretação e Avaliação dos Resultados**, os padrões obtidos na mineração de dados são analisados e avaliados. Algumas medidas comuns, tais como precisão e revocação, apesar de relevantes, não representam as regras de negócio que devem ser consideradas para modelar o cenário real de perdas e ganhos. Por isso, é proposta uma nova medida que indica os ganhos financeiros obtidos considerando as regras de negócio, denominada como Eficiência Econômica *(EE)*, Equação 1. Onde $n$ representa o número de transações, $G$ representa o valor financeiro das transações verdadeiras positivas, $gr$ é o percentual de ganho da empresa em uma transação bem sucedida, $L$ é o valor financeiro

 ©2012 SBC — Soc. Bras. de Computação

das transações falsas negativas e $lr$ é o percentual de perda em uma transação de fraude classificada como não fraude. São definidos três limites de referência: (1) Eficiência Econômica Máxima *($EE_{Max}$)*, que representa o ganho máximo possível, onde o termo $(L * lr)$ da Equação 1 é igual a $0$; (2) Eficiência Econômica Real *($EE_{Real}$)*, que consiste no cálculo da *(EE)* sem a utilização de nenhuma das técnicas propostas, os resultados devem ficar acima desse limite para que tragam algum benefício econômico; (3) Eficiência Econômica Mínima *($EE_{Min}$)* que representa 100% de perda, onde o termo $(G * gr)$ da Equação 1 é igual a $0$.

$$EE = \sum_{j=1}^{n} G * gr - L * lr \tag{1}$$

A Equação 1 foi adaptada para considerar uma penalização nos casos em que transações legítimas são erroneamente classificadas como ilegítimas [Júnior et al. 2011]. A Equação 2 define a Eficiência Econômica com Penalização *($EE_{CP}$)*. Onde $NG$ representa o valor dos registros legítimos classificados como fraude. $p$ representa o percentual desse valor que será considerado como perda. Para avaliar o percentual de ganho obtido em relação ao ganho máximo possível deve-se utilizar Equação 3. Ela fornecem o ganho relativo em que 100% representa o ganho máximo possível *($EE_{Max}$)* e 0% o cenário real *($EE_{Real}$)*. A Equação 3 representa o ganho alcançado considerando a Eficiência Econômica sem penalidade *(EE)* e para obter o ganho alcançado considerando a penalidade deve-se substituir na Equação 3 o termo *(EE)* pela *($EE_{CP}$)*.

$$EE_{CP} = \sum_{j=1}^{n} G * gr - L * lr + NG * p \tag{2}$$

$$Ganho_{SP} = \frac{EE - EE_{Real}}{EE_{Max} - EE_{Real}} \tag{3}$$

Os classificadores selecionados tem como resultado a probabilidade de um registro ser fraudulento ou não. Sendo assim, como forma de maximizar os resultados obtidos com o classificador, é proposta a geração de um *ranking* considerando a probabilidade do registro ser fraudulento. Dessa forma, a *EE* deverá ser analisada ao longo do *ranking* de forma a encontrar um ponto onde os resultados são maximizados.

## 4. Estudo de Caso

Para avaliar a eficiência da metodologia proposta, este trabalho utiliza como cenário os dados da empresa *UOL PagSeguro*, que possui soluções em pagamentos *online*. A seguir será instanciada a metodologia seguindo suas etapas. Para **Seleção dos Dados** o *UOL PagSeguro* disponibilizou uma base *Oracle* compreendendo um período de agosto de 2010 a junho de 2011. Após o entendimento dos dados, foi utilizado o método de extração definido na seção 3. Os grupos criados seguindo o *passo 2* foram: *Transação*, *Vendedores*, *Compradores* e *Crédito*. O passo final consistiu em extrair o *dataset* criado para um arquivo texto no formato *CSV*. No **Pré-Processamento** dentre as inconsistências encontradas e eliminadas pode-se citar: valores negativos para a idade; campos que possuíam muitos valores nulos e siglas de estados não existentes. Foram retirados, também, campos de IDs. Alguns dos padrões encontrados foram: vendedores registrados mais recentemente são mais propensos a fraude; as fraudes se concentram em vendas feitas com um número

 ©2012 SBC — Soc. Bras. de Computação

pequeno de parcelas; as fraudes são maiores a partir de idades superiores a 40 anos. Essa existência de padrões fornece bons indícios de resultados relevantes na mineração de dados. Na **Transformação** os atributos derivados criados foram: *Período de registro do comprador*, *Período de registro do vendedor*, *Flag CPF* indica diferença entre CPF do usuário e titular do cartão e *Flag DDD* indica diferença entre DDD do usuário e titular do cartão. Os atributos de data foram separados gerando 4 atributos: dia da semana, semana, mês e hora. O campo *data de nascimento do titular do cartão* foi transformado em idade, calculada baseada na *data de criação da transação*. Métricas como *Entropia* e *Chi-Quadrado* foram utilizadas para avaliação dos atributos onde foram retirados aqueles que apresentaram os valores mais baixos. Na **Mineração de Dados** os algoritmos escolhidos são aqueles definidos na seção 3. Para *Árvore de Decisão:* o algoritmo utilizado foi o *C4.5* que consegue lidar também com atributos contínuos. Para *boosting* o algoritmo utilizado foi o *AdaBoost*. O *oversampling* consistiu em realizar sobreamostragens da classe minoritária com distribuição de classe de 10% até 50% variando de 10 em 10. A combinação de algoritmos para *Stacking* foi *C4.5*, *RIPPER* e *Naive Bayes* tendo como *meta learner* o classificador *Naive Bayes*. Na **Interpretação e Avaliação dos Resultados** foi utilizada a validação cruzada e cada um dos meses foi dividido em 5 partições, sendo 4 utilizadas para treino e 1 para teste. As métricas são calculadas considerando uma média dos resultados de cada partição de teste. As tabelas a seguir representam o ponto no *ranking* onde os resultados são maximizados. São mostradas as seguintes medidas: Precisão *(Prec.)*, Revocação *(Rev.)*, *($EE_{SP}$)*, *($EE_{CP}$)*, *($Ganho_{SP}$)*, *($Ganho_{CP}$)* e Percentual do *Ranking* onde o resultado é maximizado *(Perc. Rank.)*. O melhor resultado de cada medida é destacado em **negrito** e assinalado com um (†).

**Tabela 1. Resultado (%) - Novembro.**

| Medida | C4.5 | RIP. | LAC | NB |
|---|---|---|---|---|
| **Prec.** | 15,93 | **77,00**† | 23,00 | 57,50 |
| **Rev.** | **19,88**† | 15,00 | 6,27 | 4,45 |
| $EE_{SP}$ | 5,00 | 5,00 | 2,00 | **6,00**† |
| $EE_{CP}$ | 5,00 | 5,00 | 2,00 | **6,00**† |
| $Ganho_{SP}$ | 15,83 | 14,85 | 7,26 | **18,06**† |
| $Ganho_{CP}$ | 14,98 | 14,78 | 6,65 | **17,90**† |
| **Perc. Rank.** | 0,40 | 0,06 | 0,09 | **0,03**† |

**Tabela 2. Resultado Meta Aprendizagem (%) - Dezembro.**

| Medida | B. C4.5 | B. RIP. | B. NB | Stack. |
|---|---|---|---|---|
| **Prec.** | 58,64 | 11,71 | 7,60 | **97,71**† |
| **Rev.** | **30,48**† | 30,44 | 11,25 | 25,39 |
| $EE_{SP}$ | 13,00 | 12,00 | 6,00 | **20,00**† |
| $EE_{CP}$ | 13,00 | 11,00 | 5,00 | **20,00**† |
| $Ganho_{SP}$ | 30,78 | 28,48 | 12,84 | **46,46**† |
| $Ganho_{CP}$ | 30,56 | 26,08 | 12,14 | **46,44**† |
| **Perc. Rank.** | 0,19 | 0,94 | 0,54 | **0,09**† |

A Tabela 1 mostra o resultado em percentual do *C4.5*, *RIPPER (RIP.)*, *LAC*, *Naive Bayes (NB)* para novembro. O *RIP.* obteve uma alta precisão, 77%, e uma boa cobertura de fraudes, 15%. O *NB*, apresentou uma melhor *EE*, porque as fraudes detectadas apresentavam valores maiores do que as fraudes obtidas pelo *RIP.*. O *C4.5* apresentou uma cobertura maior de fraude entretanto a baixa precisão fez com que a sua *EE* não superasse o *NB*. O resultado não tão bom do *LAC* está associado à necessidade de discretização de algumas variáveis para utilizar o algoritmo, o que representa perda de informação. O *NB* apresentou os menores percentuais de utilização do *ranking* para maximizar os resultados. A Tabela 2 mostra o resultado da meta aprendizagem, o objetivo é verificar se a técnica permite melhorar a qualidade dos resultados dos classificadores do primeiro ex-

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 3. Resultado *oversampling* (%) - Dezembro.**

| Medida | 10% | 20% | 30% | 40% | 50% |
|---|---|---|---|---|---|
| | | **C4.5** | | | |
| **Prec.** | **9,06**† | 8,11 | 8,07 | 7,80 | 5,89 |
| **Rev.** | 21,86 | **27,09**† | 22,75 | 20,23 | 25,13 |
| $EE_{SP}$ | 9,00 | **10,00**† | 9,00 | 9,00 | 8,00 |
| $EE_{CP}$ | 8,00 | **9,00**† | 8,00 | 8,00 | 7,00 |
| $Ganho_{SP}$ | 21,20 | **22,68**† | 20,58 | 19,58 | 18,89 |
| $Ganho_{CP}$ | 19,21 | **20,09**† | 18,28 | 17,45 | 15,41 |
| **Perc. Rank.** | **0,88**† | 1,20 | 1,02 | 0,94 | 1,54 |
| | | **NB** | | | |
| **Prec.** | 84,33 | **100,00**† | 55,70 | 71,40 | 99,00 |
| **Rev.** | 9,39 | 11,14 | 12,40 | 13,25 | **14,70**† |
| $EE_{SP}$ | 12,00 | 13,00 | 14,00 | 15,00 | **16,00**† |
| $EE_{CP}$ | 12,00 | 13,00 | 14,00 | 15,00 | **16,00**† |
| $Ganho_{SP}$ | 26,52 | 30,65 | 31,90 | 33,64 | **36,42**† |
| $Ganho_{CP}$ | 26,43 | 30,65 | 31,57 | 33,40 | **36,41**† |
| **Perc. Rank.** | **0,04**† | **0,04**† | 0,08 | 0,07 | **0,04**† |
| | | **LAC** | | | |
| **Prec.** | **6,18**† | 5,26 | 5,42 | 4,69 | 4,30 |
| **Rev.** | 24,35 | **24,61**† | 21,16 | 22,83 | 22,01 |
| $EE_{SP}$ | **8,00**† | **8,00**† | 7,00 | 7,00 | 7,00 |
| $EE_{CP}$ | **7,00**† | 6,00 | 6,00 | 5,00 | 5,00 |
| $Ganho_{SP}$ | **19,09**† | 17,88 | 16,83 | 15,72 | 14,89 |
| $Ganho_{CP}$ | **15,12**† | 13,46 | 13,12 | 11,57 | 10,70 |
| **Perc. Rank.** | 1,42 | 1,68 | 1,47 | 1,75 | **1,84**† |

perimento. A técnica de *boosting* permitiu uma melhora significativa no algoritmo *C4.5*, apresentando uma excelente Prec. e Rev. e uma boa *EE*. Para o *RIP.* isso representou uma piora na Prec., entretanto foi obtida uma melhora considerável na Rev. e na *EE*. Isso ocorreu devido ao reforço nos registros das classes de fraude promovido pelo *boosting*. No entanto, essa técnica piorou consideravelmente o resultado do algoritmo *NB*, uma vez que a piora na Prec. não refletiu em uma melhora da Rev., porque as amostras criadas pela técnica não reforçaram significativamente as classes de fraude. O *Stacking* apresenta o melhor resultado, sendo o mais importante, a obtenção de um ganho em eficiência de 46,46% sem penalidade e 46,44% considerando a penalidade no erro, ou seja, resultado praticamente igual devido à excelente precisão de 97,71% em um pequeno número de ocorrências próximas ao topo do *ranking*. A Tabela 3 mostra os resultados das técnicas com *oversampling*. O objetivo é avaliar se uma melhora na distribuição das classes de fraude permitirá uma melhora nos resultados. Não foi possível realizar teste com o *RIP.* devido a problemas de alocação de memória. No geral, tanto para o algoritmo *C4.5* quanto para o *NB* ocorreu uma melhora da cobertura de fraude, mostrando que a alteração na distribuição contribui para uma melhora na identificação da classe minoritária. Entretanto, tanto o *C4.5* quanto o *LAC* apresentam uma piora na Prec. devido ao aumento de registros legítimos sendo classificados como fraude. No caso do *C4.5* essa piora na Prec. não afetou negativamente a *EE*, pois na maioria dos casos ela foi superior ao resultado da técnica onde não houve mudança da distribuição. Para o *LAC* a alteração da distribuição representou uma piora na *EE* pois houve queda significativa da Prec. já que muitos registro legítimos que antes não eram classificados como fraude passaram a ser considerados como tal. O *NB* mostrou os melhores resultados apresentando melhoras em todas as métricas e um valor de *EE* de 36,42% sem penalidade e 36,41% com penalidade.

Algumas hipóteses realizadas em relação à metodologia aplicada e a sua eficácia na resolução do problema foram: **H1 - Os atributos selecionados são relevantes para a detecção de fraude.** Todas as tabelas de resultados apresentadas mostram ganhos em relação à eficiência econômica do *UOL PagSeguro*. Sendo assim, pode-se considerar que os atributos selecionados são relevantes para resolução do problema. No entanto, exis-

 ©2012 SBC — Soc. Bras. de Computação

tem vários outros atributos que podem representar indícios de fraude, sendo que novos *datasets* podem ser gerados e propiciar uma melhora dos resultados obtidos. **H2 - A eficiência econômica permite avaliar a qualidade dos resultados.** Geralmente as melhores eficiências econômicas estarão associadas aos melhores modelos. No entanto, ela pode tender para modelos que identificam casos de fraude de maiores valores. Sendo assim, é interessante que ela seja utilizada em conjunto com outras medidas de avaliação do modelo como a precisão e a revocação. Isso irá garantir que os melhores modelos com os melhores ganhos financeiros sejam selecionados. **Q1 - Qual técnica apresentou o melhor resultado?** Conforme apresentado na Tabela 2 a técnica *Stacking* apresentou o melhor resultado, atingindo ganhos de eficiência de até 46,46%. Uma das vantagens dessa técnica é que ela combina vários classificadores extraindo o melhor resultado deles. Sendo assim, esse resultado pode ser melhorado, combinando outras técnicas que se mostrem mais eficazes na detecção de fraude.

## 5. Conclusão

Devido aos prejuízos financeiros causados pelas fraudes em pagamentos *online* com cartão de crédito este trabalho propôs uma metodologia, baseada no KDD, para sua detecção considerando o cenário de comércio eletrônico. Para sua avaliação, foi feito um estudo de caso em um conjunto de dados real da empresa *UOL PagSeguro* de serviços de pagamentos eletrônicos. Os passos da metodologia foram seguidos e os resultados alcançados comprovam sua eficácia, onde todas as técnicas apresentam ganhos em eficiência econômica superiores aos alcançados pelo *UOL PagSeguro*. Dentre os melhores resultados pode-se citar o ganho de 46,46% obtido pelo *Stacking*, o ganho de 36,42% obtido com o *oversampling* para o *Naive Bayes* e o ganho de 18,08% obtido com o *Naive Bayes*. Esses valores representam a porcentagem em relação ao ganho máximo possível, sendo o máximo representado por 100%. É possível perceber que são valores expressivos, que podem representar uma economia significativa para o *UOL PagSeguro*. Um ponto importante é a ótima precisão obtida com a técnica *Stacking* com um valor de 97,71%, mostrando que essa combinação de classificadores é uma boa opção para o processo de detecção de fraude.

## 6. Publicações

Durante a realização desta pesquisa, os seguinte artigos foram publicados:

1. Artigo intitulado *Fraud Detection in Electronic Transactions* publicado em *Iadis International Conference WWW/Internet*, realizada em 2011 no Rio de Janeiro. Nesse artigo são apresentados os resultados preliminares deste trabalho.
2. Artigo intitulado *Slice and Aggregate: New Evolutionary Approaches to High-Dimensional Data* aceito como resumo estendido (poster) em *GECCO'12 - Genetic and Evolutionary Computation Conference*, realizada em 2012 na Filadélfia, EUA. Nesse artigo é apresentado um método utilizando algoritmos genéticos para lidar com dados de alta dimensionalidade.
3. Artigo curto intitulado *Methodology for Fraud Detection in Electronic Transactions* publicado em *WebMedia 2012 XVIII Simpósio Brasileiro de Sistemas Multimídia e Web*, a ser realizado em 2012 em São Paulo. Nesse artigo são apresentados os resultados finais deste trabalho.

## 7. Agradecimentos

Este trabalho foi parcialmente financiado pelo Universo Online SA - UOL (www.uol.com.br) e parcialmente financiado pelo Instituto Nacional de Ciência e Tecnologia para a Web (CNPq processo $n^o$. 573871/2008-6 ), CAPES, CNPq , Finep e Fapemig.

 ©2012 SBC — Soc. Bras. de Computação

## Referências


Bhatla, T. P., Prabhu, V., and Dua, A. (2003). *Understanding Credit Card Frauds*. USA.

Bolton, R. J. and Hand, D. J. (2002). Unsupervised Profiling Methods for Fraud Detection. *Statistical Science*, 17(3):235–255.

Chau, D. H., P, S., and Faloutsos, C. (2006). Detecting fraudulent personalities in networks of online auctioneers. In *In Proc. ECML/PKDD*, pages 103–114.

Cohen, W. W. (1995). Fast effective rule induction. In *In Proceedings of the Twelfth Intl. Conf. on Machine Learning*, pages 115–123. Morgan Kaufmann.

Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. data mining and knowledge discovery.

Fayyad, U. M. (1996). Data mining and knowledge discovery: Making sense out of data. *IEEE Expert: Intelligent Systems and Their Applications*, 11:20–25.

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. of the Second European Conf. on Computational Learning Theory*, EuroCOLT '95, pages 23–37, London, UK, UK. Springer-Verlag.

Gadi, M. F. A. (2008). *Uma Comparação de Métodos de Classificação Aplicados à Detecção de Fraude em Cartões de Crédito*. São Paulo.

Júnior, J. F., Veloso, A., Jr., W. M., and Pereira, A. (2011). Fraud detection in electronic transactions. *Iadis International Conference WWW/Internet*, 1:333–339.

Maes, S., Tuyls, K., Vanschoenwinkel, B., and Manderick, B. (1993). Credit card fraud detection using bayesian and neural networks. In *In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association Neurological Surgeons*, pages 261–270.

Mindware Research Group, C. (2011). *2011 Online Fraud Report*. California, USA, 12th annual edition.

Phua, C., Alahakoon, D., and Lee, V. (2004). Minority report in fraud detection: classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59.

Phua, C., Lee, V., Smith-Miles, K., and Gayler, R. (2005). A comprehensive survey of data mining-based fraud detection research.

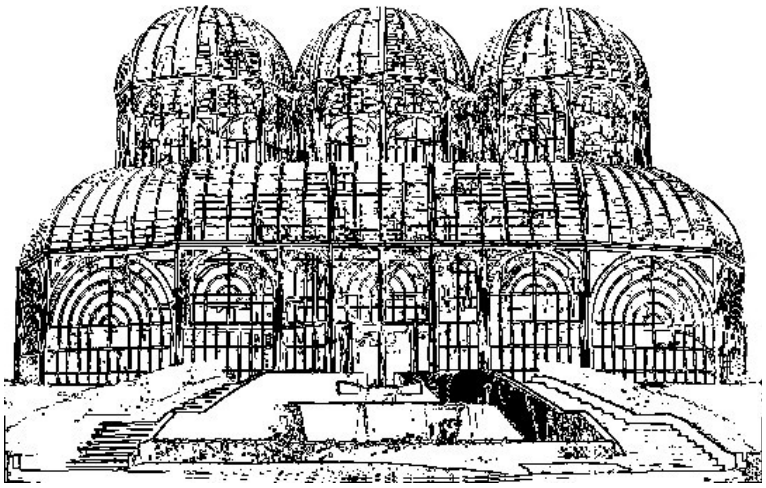
Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Vasiu, L. and Vasiu, I. (2004). Dissecting computer fraud: From definitional issues to a taxonomy. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7 - Volume 7*, HICSS '04, pages 70170.3–, Washington, DC, USA. IEEE CS.

Veloso, A., Meira, W., and Zaki, M. J. (2006). Lazy associative classification. In *Int. Conf. on Data Mining*, pages 645–654.

Williams, G. J. and Huang, Z. (1997). Mining the knowledge mine: The hot spots methodology for mining large real world databases.


 ©2012 SBC — Soc. Bras. de Computação



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# WTICG – Workshop de Trabalhos de Iniciação Científica e de Graduação

# Uma Estratégia para Gerenciar o Compartilhamento de Recursos entre Dispositivos Móveis

**Tiago H. Bono**[1], **Luciana A. F. Martimiano**[1]

[1]Departamento de Informática - Universidade Estadual de Maringá (UEM)
Avenida Colombo, 5790 – 87020-900 – Maringá – PR – Brazil

thbono@gmail.com, luciana@din.uem.br

***Abstract.*** *The use of wireless networks has increased the amount of resources that can be shared amongst mobile devices and consequently caused an increase in consumption of these devices. Thus, it becomes important to develop a strategy that defines which resources can be shared among users (devices), considering restrictions on access to confidential information and on devices capacities, keeping it available to the user. One proposed solution for this scenario is the ontology named PrOHand (Privacy Ontology for Handovers). This paper describes PrOHand, its reputation system and the application developed to validate its rules.*

***Resumo.*** *A intensificação do uso das redes sem fio tem elevado a quantidade de recursos que podem ser compartilhados entre os dispositivos móveis e, consequentemente, provocado um aumento do consumo dos recursos desses dispositivos. Assim, torna-se importante desenvolver uma estratégia que defina quais recursos devem ser compartilhados entre os usuários, considerando restrições de acesso às informações confidenciais e à capacidade do dispositivo, mantendo-o disponível para o usuário. Uma solução proposta para este cenário é a ontologia denominada PrOHand (Privacy Ontology for Handovers). Este artigo descreve a PrOHand, seu sistema de reputação e a aplicação desenvolvida para validar suas regras.*

## 1. Introdução

Diversos fatores têm influenciado na intensificação do uso das redes sem fio. Dentre esses fatores podemos citar o aumento da área de abrangência e da velocidade de transmissão dessas redes. Adendo ao crescimento do número de usuários, está a expansão dos tipos de mídias transmitidas entre os dispositivos móveis diretamente.

Torna-se então interessante desenvolver uma estratégia que possibilite a esses dispositivos selecionar os recursos e serviços (como *cache* de Web, sinal WiFi, lista de contatos, entre outros) que podem ser compartilhados entre seus usuários. Com tal estratégia, seria possível filtrar as informações trocadas, e estabelecer padrões de segurança com o intuito de bloquear o acesso e a manipulação de informações confidenciais por meio de ataques de intrusos na rede.

Uma proposta de estratégia foi desenvolvida por [Gonçalves 2008] e é denominada *PrOHand* (*Privacy Ontology for Handovers*). A *PrOHand* é uma ontologia de privacidade que propõe o estabelecimento de níveis de confiança entre os dispositivos móveis por meio do armazenamento e compartilhamento de informações de privacidade entre os

 ©2012 SBC — Soc. Bras. de Computação

usuários. A ontologia foi criada para gerenciar o compartilhamento de recursos que são disponibilizados de acordo com o nível de confiança que um usuário tem com relação a outro em uma rede ad-hoc, na qual a comunicação entre os dispositivos é efetuada de maneira direta, ou seja, sem a necessidade de uma estação-base (rede infraestruturada).

Para validar as regras da *PrOHand*, uma aplicação foi desenvolvida e testada. Nessa aplicação é possível simular o compartilhamento de recursos e serviços entre usuários de dispositivos móveis a partir dos níveis de confiança previamente definidos pelo usuário para o acesso ou uso do recurso.

Este artigo está organizado da seguinte forma: a Seção II descreve a ontologia *PrOHand*; a Seção III descreve a aplicação desenvolvida; a Seção IV apresenta os resultados alcançados; e a Seção V apresenta as considerações finais e os trabalhos futuros.

## 2. A Ontologia *PrOHand*

Segundo [Misztal 1996], confiança é um grau subjetivo de crença sobre determinados agentes. O nível de confiança que se tem sobre uma pessoa reflete o quão bem-intencionada esta pessoa é julgada. Segundo [Gambetta 2000], quando confia-se em uma pessoa, pressupõe-se muito provavelmente que ela efetuará ações benéficas, ou pelo menos não prejudiciais, de forma que uma cooperação seja possível. Para o meio virtual, as definições de confiança tradicionais podem ser traduzidas por meio do desenvolvimento de um modelo de confiança que regulamente as relações de confiabilidade neste meio.

De acordo com [Westin 1967], privacidade é "a necessidade que indivíduos, grupos ou instituições possuem em determinar por si próprios, com quem, quando, e como compartilhar informações no sistema nos quais atuam".

A *PrOHand*, ilustrada na Figura 1, fornece um mecanismo para controlar o compartilhamento de recursos e serviços entre dispositivos móveis em redes *ad-hoc*. O acesso a algum destes recursos e/ou serviços de um dispositivo é permitido somente mediante certo nível de confiança.

O nível de confiança entre os dispositivos (ou usuários) é calculado segundo um sistema de reputação, e pode variar de pouco confiável a muito confiável. As reputações utilizadas para o cálculo são a direta (*direct reputation*) e a indireta (*indirect reputation*). A reputação direta é baseada no histórico de conexões diretas com o dispositivo solicitante, ou seja, de experiências passadas. A reputação indireta é obtida a partir de terceiros que já tiveram experiências diretas com o solicitante, e atuam como recomendantes sobre o mesmo.

Na *PrOHand* são definidos três atores: *user*, *requester* e *recommender*. O *requester* tem intenção de solicitar acesso a algum recurso disponibilizado pelo *user*. Para isso, o *user* deve verificar o nível de confiança, ou *trust level*, do *requester*. Caso este nível esteja de acordo com o valor pré-estabelecido pelo *user*, o *requester* terá acesso ao recurso. Caso haja um *recommender*, as suas recomendações também são utilizadas para definir o nível de confiança do *requester* [Gonçalves et al. 2009].

Os usuários são capazes de se comunicar sem uma infraestrutura de rede tradicional. Esta comunicação é possível devido aos encontros oportunísticos (ou *haggle* [Su et al. 2007]), típicos de redes móveis. Nesses encontros os usuários podem compartilhar informações e serviços, justificando o caráter colaborativo em **acordos usuário-**

 ©2012 SBC — Soc. Bras. de Computação

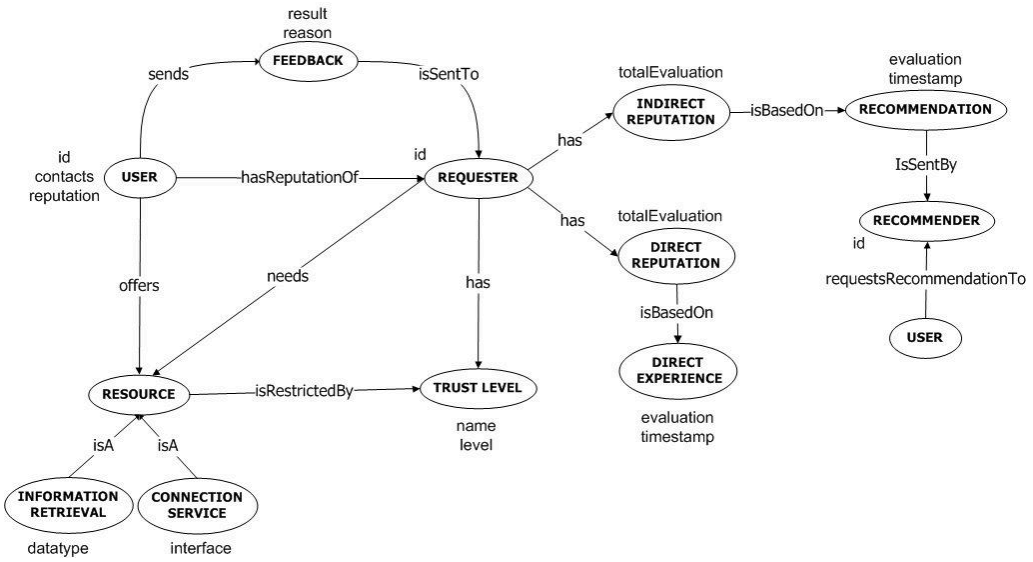



**Figura 1. A Ontologia *PrOHand* [Gonçalves et al. 2009]**

**usuário**. Esse compartilhamento se dá por meio do tráfego de dados entre dispositivos vizinhos. Se tal compartilhamento for ilimitado, há um espaço de atuação para dispositivos causadores de incidentes maliciosos. Há também a possibidade de inundar o dispositivo, incapacidando-o de efetuar tarefas essenciais do usuário.

A *PrOHand*, considerando este cenário, permite ao usuário manter níveis de controle sobre seus recursos. Esse controle é gerenciado com base no modo como ele e seus contatos confiam no requisitante do recurso. Para tal, é utilizado um sistema de reputação entre usuários baseado no sistema proposto por [Abdul-Rahman and Hailes 2000].

Em um cenário de sistema colaborativo, um usuário pode efetuar requisições por recursos a outro usuário. O usuário que recebeu a requisição pode optar por aceitá-la ou rejeitá-la, de acordo com o nível de confiança necessário para acessar o recurso. Esse nível, definido pelo detentor do recurso, é confrontado com nível de confiança que o requisitante possui. Se tal nível for suficiente, a requisição é atendida, caso contrário, um *feedback* negativo é enviado ao requisitante.

Esse nível de confiança sobre um dispositivo, ao contrário do nível de confiança necessário para acessar determinado recurso que é definido previamente, é calculado dinamicamente. Tendo como exemplo os usuários A (*user*) e B (*requester*), o nível de confiança que A mantém sobre B é baseado na reputação de B no sistema. A reputação de B é formada por sua reputação direta e também por sua reputação indireta. A reputação direta é definida pelo histórico de experiências diretas, chamadas de comunicações, que A já teve com B. A reputação indireta é definida por opiniões, chamadas de recomendações, de outros usuários sobre B. As informações necessárias para o cálculo da reputação e da reputação em si são armazenadas pelo *user*.

Os recursos do usuário A podem ser associados a um determinado nível de confiança (*Trust Level*), com base na importância que cada recurso representa para ele. Aspectos como confidencialidade e disponibilidade são considerados para determinar o nível de confiança de cada recurso. A confidencialidade se refere à proteção

 ©2012 SBC — Soc. Bras. de Computação

de informações confidenciais (como dados pessoais) e a disponibilidade se refere à manutenção da operabilidade do dispositivo (como banda de conexão).

A *PrOHand* prevê cinco níveis de confiança: *Very Untrustworthy, Untrustworthy, No Opinion, Trustworthy, Very Trustworthy* (definidas de acordo com o trabalho de [Abdul-Rahman and Hailes 2000]). Cada recurso disponibilizado pelo dispositivo pode ter seu nível configurado pelo usuário, ficando disponível somente para requisitantes de nível de confiança compatível. Por exemplo, o usuário pode configurar sua lista de contatos como um recurso de categoria *Very Trustworthy*, seu *cache* de navegação Web como *Trustworthy*, e assim por diante. Assim, somente usuários classificados com no mínimo estes níveis poderão acessar o recurso.

Com os valores das reputações direta e indireta, é possível calcular o valor final da reputação. Esse valor determinará o nível de confiança que, por sua vez, determinará se o recurso poderá ser acessado pelo requisitante. Seguindo a ordem semântica das categorias de confiança, se a confiança calculada for maior ou igual à definida pelo detentor do recurso solicitado, o acesso será permitido. Ao final da comunicação, ambos os usuários adicionam uma nova entrada no histórico de interações, de modo que possam calcular a reputação direta e enviar recomendações sobre o outro. Este processo ocorre mesmo que o acesso não seja permitido, pois um *feedback* é enviado.

## 3. O Sistema de Reputação

De acordo com [Yu et al. 2004], sistemas de reputação são mecanismos que permitem a usuários (*peers*) avaliar comportamentos alheios sem a necessidade de acesso a uma entidade terceira centralizada. Cada integrante do sistema incorpora conhecimentos de outras pessoas que, juntos, produzem um novo conhecimento.

O acesso às entidades terceiras é de fato muito mais simples do que a abordagem de se calcular periodicamente reputações. Porém, com as informações sobre o comportamento dos usuários armazenadas de forma distribuída, a disponibilidade de um conteúdo altamente dinâmico é garantida. O dinamismo do conhecimento do sistema reflete a atualização do mesmo a cada interação entre os integrantes.

A reputação direta de um *peer* $Pj$ do ponto de vista do *peer* $Pi$ é calculada utilizando a Equação 1.

$$RP(Pi, Pj) = \begin{cases} \frac{\sum_{k=1}^{h} e_{ij}^{k}}{h} & h \neq 0 \\ 0 & h = 0 \end{cases} \tag{1}$$

Nesta equação, $e_{ij}^{k} \in [0, 2]$ representa o valor da k-ésima experiência direta mais recente de $Pi$ com $Pj$ e $h$ é o número de entradas consideradas. A equação corresponde a uma média simples que, apesar de pouco robusta, não apresenta complexidade elevada. Esse é um fator importante considerando o uso da mesma em operações frequentes e em dispositivos móveis de processamento limitado.

A reputação indireta de um *peer* $Pj$ do ponto de vista do *peer* $Pi$ é calculada utilizando a Equação 2.

 ©2012 SBC — Soc. Bras. de Computação

$$RI(Pi, Pj) = \begin{cases} \frac{\sum_{k=1}^{L} Rep(Pi,Pk) \cdot Rep(Pk,Pj) \cdot \alpha(Ti,Tk)}{L} & L \neq 0 \\ 0 & L = 0 \end{cases} \quad (2)$$

Nesta equação, $L$ é o número de recomendações, $Rep(Pi, Pk)$ a reputação, do ponto de vista de $Pi$, do usuário $Pk$ que enviou a recomendação e $Rep(Pk, Pj)$ a recomendação em si, ou seja, a reputação de $Pk$ sobre $Pj$. Observando a equação, percebe-se que as recomendações sobre o requisitante são ponderadas de acordo com o nível de confiança do usuário sobre quem as envia.

O tempo em que cada recomendação foi atualizada pela última vez é ponderado pelo termo $\alpha(Ti, Tk)$. A adição desse fator permite obter resultados mais próximos à realidade do sistema em questão. O termo é descrito na Equação 3.

$$\alpha(Ta, Tb) = \frac{1}{\left(\frac{(Ta-Tb)}{\mu}\right) + 1} \quad (3)$$

Nesta equação, $Ta$ é o tempo atual no dispositivo do requisitado, $Tb$ o tempo no qual a recomendação foi atualizada pela última vez, e $\mu$ uma constante que representa o fator de decaimento para as recomendações.

Com os valores das reputações parciais, é possível calcular o valor de reputação de $Pj$ do ponto de vista de $Pi$, pela Equação 4.

$$Rep(Pi, Pj) = \theta \cdot RD(Pi, Pj) + (1 - \theta) \cdot RI(Pi, Pj) \quad (4)$$

Nesta equação, $\theta \in [0, 1]$ é uma constante que indica a importância da reputação direta sobre a indireta.

Mapeando os intervalos de confiança, têm-se que:

- $Pj$ será mapeado para *Very Untrustworthy* $\leftrightarrow Rep(Pi, Pj) \in [0.0, 0.4)$
- $Pj$ será mapeado para *Untrustworthy* $\leftrightarrow Rep(Pi, Pj) \in [0.4, 0.8)$
- $Pj$ será mapeado para *No Opinion* $\leftrightarrow Rep(Pi, Pj) \in [0.8, 1.2)$
- $Pj$ será mapeado para *Trustworthy* $\leftrightarrow Rep(Pi, Pj) \in [1.2, 1.6)$
- $Pj$ será mapeado para *Very Trustworthy* $\leftrightarrow Rep(Pi, Pj) \in [1.6, 2.0]$

## 4. Desenvolvimento da Aplicação

A implementação das regras da *PrOHand* em um aplicativo para dispositivos móveis, denominado *PrOHand Manager*, foi realizada utilizando a tecnologia J2ME, com o desenvolvimento de uma MIDLet gráfica. Essa tecnologia provê uma portabilidade entre diversos dispositivos, através do conceito de máquina virtual. Disponibiliza também uma API (*Application Programming Interfaces*) de comunicação que provê o estabelecimento

 ©2012 SBC — Soc. Bras. de Computação

de comunicação em rede de forma independentemente do protocolo. O *kit* de desenvolvimento utilizado foi o *Java Wireless Toolkit 2.5.2*, da *Sun Microsystems* [Muchow 2006]. Este software disponibiliza a API J2ME, o compilador, a KVM, e outras ferramentas como emuladores de execução em dispositivos de recursos limitados. As versões da API utilizadas foram CLDC 1.1 e MIDP 1.0.

### 4.1. O Modelo de Conexão

A conexão entre os dispositivos ocorre por meio do tráfego de dados em uma rede sem fio, através do uso de *sockets*. É utilizado um protocolo próprio de envio e recebimento de informações de sequências de caracteres. Caracteres especiais são utilizados como forma de separação semântica dos dados.

Cada dispositivo é identificado na rede por meio de seu *hostname*, necessário para estar presente na rede. O ideal é utilizar uma identificação única, o número IMEI dos dispositivos móveis. O número IMEI é fornecido pelo fabricante do *hardware* do aparelho, sendo único pois é dividido em faixas entre fabricantes distintos. Porém, por questões de segurança, a plataforma J2ME não provê acesso à essa identificação do aparelho.

A tecnologia J2ME, na versão 1.0 da API MIDP, impõe outra restrição para a implementação do aplicativo de acordo com o planejado, não permitindo que os dispositivos atuem como servidores de *socket*, mas somente como clientes. Essa limitação impede a utilização do modelo de conexão *ad-hoc*, pois um dispositivo não é capaz de fazer requisições diretamente para outro. Para contornar esse problema, foi criado um aplicativo que intermedia as conexões entre os dispositivos. Cada dispositivo realiza requisições para esse aplicativo, chamado de *PrOHand Access Point*, e também consulta as requisições pendentes para ele.

Como ao final deste trabalho de graduação em 2011 não foi possível superar essa limitação por falta de tempo, o aluno que está continuando o trabalho em 2012 já está trabalhando na solução.

### 4.2. O Modelo de Execução

Todos as requisições que o dispositivo envia, e também o tratamento e resposta para as requisições que recebe, via consulta ao *PrOHand Access Point*, são executadas em *threads* separadas. Esse modelo foi adotado visando a permitir que o usuário manipule a interface gráfica do aplicativo enquanto este efetua as operações referentes à comunicação e às regras da *PrOHand*.

Outra razão para o uso desse modelo é a separação da interface gráfica destas operações essenciais, permitindo evoluir o aplicativo para um software do tipo *daemon*. Atualmente o usuário necessita cadastrar os recursos, cujo conteúdo deve ser em formato texto, e também efetuar as requisições por recursos de outros dispositivos. Um aplicativo *daemon* poderia receber comandos de outros aplicativos solicitando os recursos, e o conteúdo dos mesmos poderia estar em formato binário, por exemplo, arquivos. Outra vantagem seria permitir que o usuário utilize outros aplicativos no dispositivo enquanto o gerenciador de recursos permanece em execução.

### 4.3. A Implementação

A ontologia *PrOHand* não foi implementada em sua totalidade. As requisições por recursos e por recomendações foram implementadas, bem como seus respectivos cálculos e

 ©2012 SBC — Soc. Bras. de Computação

respostas ao dispositivo solicitante. Além disso, a estrutura de dados criada contém todas as informações necessárias para o cálculo do nível de confiança, e o modelo de conexões contempla as operações de comunicação. Porém, durante o cálculo da reputação do dispositivo que solicita um recurso, as recomendações não estão sendo solicitadas, assim, o cálculo do nível de confiança considera apenas a reputação direta. O mecanismo que solicita as recomendações e aguarda pelas respostas para então calcular a reputação de modo completo já está em fase de implementação.

A *PrOHand* provê as equações para o cálculo do nível de confiança com base no histórico de interações entre os mesmos. Porém, a ontologia não contempla uma política que defina a forma como a confiança aumenta e/ou diminui ao longo do tempo. Na aplicação, quando uma requisição é bem sucedida, ou seja, a reputação do dispositivo solicitante é maior ou igual ao nível de confiança necessário para acessar o recurso solicitado, o nível de confiança no dispositivo solicitante aumenta em $0.2$ para o dispositivo detentor do recurso. Foi adotado um modelo no qual em nenhum momento o nível de confiança em um dispositivo é reduzido, pois em requisições não atentidas, por exemplo, quando o nível de confiança não é suficiente e/ou o recurso não existe, a reputação permanece inalterada. Esse foi modelo foi adotado com o propósito de simplificar os testes da aplicação, não se caracterizando como uma limitação da ontologia ou da aplicação, mas sim uma decisão de projeto.

### 4.4. A Aplicação Exemplo

A aplicação exemplo desenvolvida tem os seguintes objetivos: 1) demonstrar o uso do aplicativo móvel; e 2) validar as regras da *PrOHand*, por exemplo, se o aplicativo somente libera o recurso solicitado para o dispositivo solicitante se o nível de confiança calculado do mesmo for igual ou superior ao nível de confiança requerido pelo recurso.

Para a execução do teste, foi utilizado um dos emuladores disponibilizados pelo *kit* de desenvolvimento, que emula um telefone celular sem o recurso de toque na tela. Para utilizar o *PrOHand Manager*, é necessário instalar o software em um dispositivo compatível com as versões da API J2ME utilizadas no desenvolvimento (CLDC 1.1 e MIDP 1.0), através dos arquivos PrOHandJ2ME.jad (*Java Application Descriptor*) e PrOHandJ2ME.jar (*Java Archive*).

A execução do aplicativo se inicia com o carregamento dos dados necessários, seguido da exibição da tela de configuração, conforme a Figura 2a. Nesta tela é necessário informar o endereço (*hostname* ou IP) da máquina na qual o *PrOHand Access Point* está sendo executado. Para que o aplicativo funcione, o dispositivo móvel deve possuir uma conexão de rede local e/ou Internet passível de acessar esta máquina. A seguir, o aplicativo exibe a tela principal (Figura 2b), de monitoramento das requisições realizadas e/ou solicitadas. Nesta tela é exibida a opção para encerrar o aplicativo, e também o menu principal. No topo da tela, abaixo do título do aplicativo, é exibida a identificação do dispositivo, logo após a frase *Requests Monitor*.

Para que o dispositivo possa disponibilizar recursos, é necessário previamente cadastrá-los. A lista dos recursos cadastrados é acessada pela opção de menu ***Resources***. Nesta tela (Figura 2c), são exibidos os recursos com seus respectivos níveis de confiança necessários para acesso, uma opção para voltar à tela principal, e um menu de operações.

Para inserir um novo recurso, existe a opção de menu ***New Resource***, que abre a

 ©2012 SBC — Soc. Bras. de Computação

**Figura 2. Telas de: a) configuração; b) Tela principal; c) listagem de recursos**

tela para preenchimento das informações do recurso (Figura 3a). O usuário deve indicar o nome do recurso, seu conteúdo, e o nível de confiança que o dispositivo deve ter no dispositivo solicitante para liberar seu acesso. Existem as opções para voltar para a tela anterior, e confirmar a inclusão do recurso. Ao confirmar, os dados são validados, e se houver algum valor inconsistente, uma mensagem de aviso é exibida. Se a inserção ocorrer de modo satisfatório, uma mensagem de informação precede o retorno para a tela anterior.

A opção de menu ***Request Resource*** da tela principal abre a tela (Figura 3b) para que o usuário possa solicitar um recurso a outro dispositivo, indicando o nome do recurso e o nome do dispositivo detentor do mesmo. São disponibilizadas as opções de voltar para a tela anterior e confirmar a requisição.

**Figura 3. Telas para: a) inclusão de recurso; b) requisição de recurso**

Ao confirmar a requisição, os dados são validados, e se houver algum valor inconsistente, uma mensagem de aviso é exibida. Se a requisição for enviada, é exibida uma mensagem indicando ao usuário que acompanhe o andamento da requisição no monitor na tela principal, tendo em vista que a requisição é assíncrona. O dispositivo envia a requisição, e em outra conexão consulta sua resposta.

Ao solicitar um novo recurso, a requisição é exibida no monitor do dispositivo, indicando seu ID (único na rede), o dispositivo solicitado e o nome do recurso solicitado.

 ©2012 SBC — Soc. Bras. de Computação

No monitor do dispositivo solicitado, é exibida a requisição, composta por seu ID, o dispositivo solicitante e o nome do recurso solicitado. O dispositivo solicitado calcula a reputação do dispositivo solicitante de acordo com a *PrOHand*, e pode responder: i) recurso não foi encontrado; ii) recurso está indisponível, pois o dispositivo solicitante não possui nível de confiança necessário; e iii) conteúdo do recurso solicitado.

Quando a requisição é satisfatória, as informações do dispositivo solicitante são atualizadas no dispositivo solicitado. Se o mesmo não estiver na base de dados, é incluído, caso contrário, seu nível de confiança é incrementado.

## 5. Resultados

Com este trabalho obteve-se o desenvolvimento de uma aplicação base para a construção de um software funcional com o objetivo de gerenciar o compartilhamento de recursos entre dispositivos móveis. O aplicativo implementado, em versão inicial, requer a manipulação do usuário para efetuar a troca de recursos cadastrados pelo próprio usuário, cujo conteúdo está em formato texto. Para que o usuário possa utilizar o dispositivo para outros fin enquanto a aplicação está em execução, esta deve ser evoluído para um modelo de execução do tipo *daemon*. Não havendo a intervenção do usuário para manipular os recursos, é possível também que os mesmos possuam conteúdo em formato binário, proveniente de outros aplicativos. Para facilitar essa evolução, a arquitetura da implementação criada provê uma separação entre a interface gráfica e os algoritmos essenciais, como o de comunicação, que implementa a *PrOHand*, e o do gerenciamento das *threads*.

A partir do aplicativo desenvolvido, foram efetuados testes visando a comprovar a eficácia da *PrOHand* quanto ao controle dos recursos. Foi elaborado um plano de testes, com a utilização de recursos de diferentes níveis de confiança exigidos. O aplicativo se comportou de maneira satisfatória segundo as regras da ontologia, de modo que os recursos tiveram seu repúdio garantido até que o nível de confiança do requerente atingisse o valor necessário. A progressão da confiança ocorreu conforme o planejado, por meio das interações bem sucedidas. Por exemplo, para acessar um recurso cujo nível de confiança exigido era 0.4, um dispositivo não o pôde fazer sem antes interagir satisfatoriamente com o dispositivo detentor do recurso, aumentando assim sua reputação.

Durante o desenvolvimento do aplicativo, algumas limitações foram impostas pela tecnologia J2ME utilizada. Uma dessas limitações impede a obtenção do número IMEI dos dispositivos, inicialmente planejado para ser utilizado como forma de identicação, porém, foi utilizado o *hostname* para esse fim, tendo em vista que os dispositivos devem estar conectados a uma rede para a execução do aplicativo. Outra limitação, esta mais grave, impede que os dispositivos atuem como servidores de requisições, acarretando na necessidade da criação de um aplicativo que atue como intermediador das conexões. Sendo assim, a aplicação ainda não utiliza o modelo *ad-hoc* de fato. Para eliminar essa limitação, é necessário migrar a aplicação para a versão 2.0 da API MIDP, que possibilita a criação de servidores *socket*. A aplicação também precisa evoluir para utilizar as recomendações no cálculo da reputação

Com relação à *PrOHand*, ainda é preciso modificá-la para que ela possa representar regras de segurança, e uma política de aumento e redução do nível de confiança.

 ©2012 SBC — Soc. Bras. de Computação

## 6. Considerações Finais e Trabalhos Futuros

Em uma rede composta por dispositivos móveis, é interessante estabelecer uma estratégia que defina o nível de informações que podem ser trocadas entre os mesmos, visando a bloquear o acesso a informações confidenciais e manter a disponibilidade dos dispositivos. Este trabalho faz uso da *PrOHand* como modelo de privacidade, que propõe o estabelecimento de níveis de confiança entre os dispositivos através do armazenamento e compartilhamento de informações de privacidade entre os usuários.

Com o trabalho já realizado, parte da ontologia *PrOHand* foi implementada considerando as restrições do ambiente móvel, que são limitação de banda, memória e processamento. Atualmente, um outro aluno de graduação está desenvolvendo o trabalho e pretende sanar as limitações discutidas neste artigo, bem como evoluir a ontologia para que ela possa representar outros termos relacionados ao domínio em questão.

## Referências

Abdul-Rahman, A. and Hailes, S. (2000). Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, pages 6007–, Washington, DC, USA. IEEE Computer Society.

Gambetta, D. (2000). *Can We Trust Trust. Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. University of Oxford Press, New York, NY.

Gonçalves, M. R. P. (2008). Uso de ontologias para apoiar o gerenciamento de privacidade e segurança durante handovers em ngn (next generation networks. Technical report, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, SP.

Gonçalves, M. R. P., Moreira, E. S., and Martimiano, L. A. F. (2009). Trust and privacy: informal ways to assess risk on opportunistic exchanges. *International Journal of Computer Science and Applications*, 6(2):66–85.

Misztal, B. A. (1996). *Trust in Modern Societies: The Search for the Bases of Social Order*. Polity Press, Cambridge MA, second edition.

Muchow, J. W. (2006). *Core J2ME Tecnologia e MIDP*. Sun Microsystems Press.

Su, J., Scott, J., Hui, P., Crowcroft, J., Lara, E. D., Diot, C., Goel, A., Lim, M. H., and Upton, E. (2007). Haggle: seamless networking for mobile applications. In *Proceedings of the 9th International Conference on Ubiquitous Computing*, pages 391–408, Berlin, Heidelberg. Springer-Verlag.

Westin, A. (1967). *Privacy and Freedom*. Atheneum, New York.

Yu, B., Singh, M., and Sycara, K. (2004). Developing trust in large scale peer-to-peer systems. In *Proceedings of 1st IEEE Symposium on Multi-Agent Security and Survivability*, pages 1–10.

 ©2012 SBC — Soc. Bras. de Computação

# Computação Distribuída com Preservação de Privacidade

**Luís G. S. Pena, Olívia M. A. Coelho, Jeroen van de Graaf**

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627 – 31270-901 – Belo Horizonte (MG) – Brasil

luispena@dcc.ufmg.br, oliviamac@ufmg.br, jvdg@dcc.ufmg.br

***Abstract.*** *Distributed Computing preserving privacy refers to cryptographic protocols used to allow two parties to work together and perform a common task. This type of protocol is very important to preserve security in cloud computing and data mining. This paper presents two proposed protocols in this area, one to calculate the inner product of two vectors and one for integer comparison, both with safety and preserving privacy. This proposal was studied and implemented during the execution of our undergraduate research in the Department of Computer Science of UFMG.*

***Resumo.*** *Computação distribuída com preservação de privacidade se refere aos protocolos criptográficos utilizados para que duas partes se unam e realizem uma tarefa em comum. Este tipo de protocolo é muito importante para preservar a segurança em data mining e cloud computing, entre outros. Este artigo apresenta duas propostas de protocolos nessa área, uma para o cálculo do produto escalar de dois vetores e outra para a comparação de inteiros, ambas de forma segura e com preservação da privacidade das partes. Estas propostas foram estudadas e implementadas durante a execução do nosso trabalho de iniciação científica no Departamento de Ciência da Computação da UFMG.*

## 1. Introdução

Algumas tendências vem surgindo devido à revolução digital das últimas décadas, uma delas é a grande quantidade de dados que precisam ser armazenados. Isso acontece porque a informação é muito relevante e ajuda, principalmente, na tomada de melhores decisões.

Para a exploração de grandes quantidades de dados, desenvolveu-se uma nova área na ciência da computação, a mineração de dados, ou *data mining*. Um problema decorrente, contudo, é que muitas vezes esse tipo de exploração entra em conflito com a privacidade de indivíduos; esses perdem o controle sobre quem tem acesso a certas informações pessoais e desconhecem a quantidade de informação a que terceiros teriam acesso.

Outra tendência, ainda, é a terceirização do armazenamento da informação; cada vez mais utilizamos a web para isso, exemplos são o *GoogleDocs* e o *DropBox*. Não temos, porém, garantia da confiabilidade dos servidores.

Vê-se nas inovações supracitadas, e em outros fatores, a necessidade de um desenvolvimento de pesquisas na área de computação distribuída segura. Serão apresentadas ao longo deste trabalho algumas soluções nesse campo, a saber: um protocolo para o cálculo do produto escalar entre dois vetores e um protocolo para a comparação de inteiros

 ©2012 SBC — Soc. Bras. de Computação

utilizando um circuito booleano. Tratam-se de soluções de problemas básicos que servem como subprotocolos para outros maiores.

Nosso modelo supõe-se a existência de um inicializador confiável. Este modelo foi proposto no [Rivest 1999] (veja também [Blundo et al. 2002]) e tem como grande vantagem o seu alto desempenho, já que elimina a necessidade de usar operações criptográficas pesadas entre as partes. Caso a existência de um inicializador confiável não seja realista, existem soluções baseadas em suposições computacionais como por exemplo [Pinkas et al. 2009].

## 2. Protocolos criptográficos e computação distribuída segura

Um protocolo é uma série de passos que foram definidos de forma a serem executados para realizar uma tarefa entre duas ou mais partes. Um protocolo criptográfico é aquele em que as partes não confiam uma na outra e por isso utilizam da criptografia para garantir alguma forma de segurança, como por exemplo: sigilo, autenticação e integridade da mensagem.

Uma importante área de pesquisa é o desenvolvimento de protocolos criptográficos para a computação distribuída segura, ou seja, protocolos que garantem a privacidade em sistemas onde partes se unem para realizar uma tarefa comum. Considerando duas partes, chamadas de Alice e Bob. Alice tem a informação $x$ e Bob tem a $y$, eles querem calcular a função $f(x, y)$ de forma a preservar sua privacidade. Alice não deve ter nenhuma informação sobre $y$, a entrada de Bob além daquelas que podem ser obtidas através da análise de sua própria entrada e saída, e vice-versa.

A Computação Distribuída Segura é muito importante para mineração de dados com preservação de privacidade, duas partes querem realizar essa operação sobre seus bancos de dados, sem revelá-los uma a outra e nem a uma terceira parte; compartilhamento de segredo verificável, uma mensagem é cifrada de forma distribuída e posteriormente recuperada utilizando as informações de um subconjunto de participantes; *cloud computing*, quando a informação é terceirizada e se deseja alguma privacidade.

Para desenvolver estes protocolos utilizamos de algumas primitivas criptográficas, como o modelo do Inicializador Confiável, a emulação do problema através de um circuito lógico, e *Oblivious Transfer*.

### 2.1. Inicializador confiável

O modelo do Inicializador Confiável, primeiro proposto em [Rivest 1999], supõe a existência de uma terceira parte em quem Alice e Bob devem confiar, mas que participará apenas da fase inicial do protocolo. Essa fase pode ser feita antes da execução do restante do protocolo, sem um limite de tempo entre os passos. Sua função é enviar um conjunto de dados para Alice e outro, correlacionado, para Bob; esses dados serão usados para o restante da execução do protocolo.

Pode-se ver que este modelo é muito interessante tendo em vista que a terceira parte não tem nenhuma informação sobre as entradas apesar de participar do protocolo. Ainda, o passo inicial pode ser executado sem um limite de tempo entre ele e os subsequentes.

 ©2012 SBC — Soc. Bras. de Computação

### 2.2. Circuito lógico

Outro artifício utilizado é a emulação do problema através de um circuito lógico, que é a utilização de propriedades de portas lógicas e álgebra booleana para realizar operações bit a bit e obter a solução para um determinado desafio. Para se utilizar esse recurso, é necessária a modelagem do problema a ser resolvido em um formato onde existam comparações cuja resposta é binária ou booleana.

O principal benefício de se utilizar a modelagem através de um circuito lógico é a eficiência contra ataques que possam interceptar dados e revelar informações sigilosas. Quando o problema é modelado, diversas operações são realizadas com os dados de entrada a fim de chegar à solução. Cada etapa do processo, porém, se analisada de forma independente, não apresenta nenhuma informação relevante sobre os dados que estão sendo processados, garantindo-se assim a privacidade das informações de entrada do circuito.

### 2.3. Oblivious Transfer

O protocolo *Oblivious Transfer* (OT) consiste em uma troca de informações entre duas partes, cliente e servidor, de maneira que ambas só saibam e revelem informações realmente necessárias, evitando a troca excessiva de dados e consequente perda de privacidade. Mostraremos que OT é uma condição suficiente para fazer uma porta AND. E já que OT pode ser implementado com inicializador confiável, obteremos uma implementação de comparação de inteiros neste modelo.

## 3. Protocolo do produto escalar

O cálculo do produto escalar entre dois vetores é um passo utilizado como sub-rotina de diversos algoritmos. Há várias propostas de protocolos as quais fazem este cálculo. Será apresentado neste artigo a proposta de Marques [Marques 2009][Dowsley et al. 2010], que utiliza o modelo do Inicializador Confiável, e também sua proposta para um protocolo de solução de sistemas lineares utilizando o produto escalar como subprotocolo, que foi baseado no [Pontual 2005].

### 3.1. Produto escalar

Dado um vetor $x$ de Alice e um vetor $y$ de Bob, procura-se calcular o produto escalar entre eles sem que nenhuma das partes obtenha informações a respeito do vetor da outra. Tudo isso considerando operações sobre um corpo finito. Veja a Tabela 1 para uma descrição formal do protocolo.

O Inicializador Confiável envia os dados indicados para cada uma das partes, e essas realizam o restante do protocolo quando quiserem. Ao final do protocolo, Alice e Bob possuem um valor aleatório $u$ e $v$ cuja soma é o produto escalar. É preciso que as partes cooperem partilhando seus resultados intermediários para que, somados, obtenha-se o produto escalar de seus vetores.

Esse protocolo é seguro de acordo com a definição de privacidade do modelo semi-honesto, ou seja, Alice e Bob seguem corretamente os passos do protocolo e apenas guardam as informações intermediárias para posterior análise na tentativa de descobrir a entrada da outra parte. A prova de segurança formal desse protocolo pode ser encontrado no [Marques 2009].

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 1. Protocolo do Produto Escalar**

| | Alice | Bob |
|---|---|---|
| IC | $\overrightarrow{x_0} \in_R \mathbb{Z}_m^n$ | $\overrightarrow{y_0} \in_R \mathbb{Z}_m^n$;<br>$s_0 := \langle \overrightarrow{x_0} \cdot \overrightarrow{y_0} \rangle$ |
| dados enviados pelo IC | $\mu_A := \overrightarrow{x_0}$ | $\mu_B := (\overrightarrow{y_0}, s_0)$ |
| entrada | $\overrightarrow{x} \in \mathbb{Z}_m^n$ | $\overrightarrow{y} \in \mathbb{Z}_m^n$ |
| protocolo | (2)<br>Se $\mu_1$ é inválido: aborta<br>$\overrightarrow{x_1} := \overrightarrow{x} + \overrightarrow{x_0} \in \mathbb{Z}_m^n$<br>$r \in_R \mathbb{Z}_m$<br>$r_1 := \langle \overrightarrow{x} \cdot \overrightarrow{y_1} \rangle - r$<br>Envia $\mu_2 := (\overrightarrow{x_1}, r_1)$ para Bob | (1)<br>$\overrightarrow{y_1} := \overrightarrow{y} - \overrightarrow{y_0} \in \mathbb{Z}_m^n$<br>Envia $\mu_1 := (\overrightarrow{y_1})$ para Alice<br><br>Se $\mu_2$ é inválido: aborta |
| saída | $u := r$ | $v := \langle \overrightarrow{x_1} \cdot \overrightarrow{y_0} \rangle + r_1 - s_0$ |

### 3.2. Sistema linear

Suponha um sistema linear formado pelos dados de Alice em conjunto com os de Bob. Alice possui uma matriz $X_{n \times n}$ e um vetor $x$ com $n$ elementos e Bob possui uma matriz $Y_{n \times n}$ e um vetor $y$ com $n$ elementos. Considere então a equação: $(X+Y).z = (x+y)$. O objetivo do protocolo é calcular o vetor $z$ de forma a não revelar para nenhuma das partes informações sobre os dados da outra. Veja Tabela 2.

O Inicializador Confiável envia os dados indicados para cada uma das partes, que realizam o restante do protocolo quando quiserem. Ao final do protocolo, Bob possui o vetor $z$, que é a solução do sistema linear indicado. Assim como no produto escalar, esse protocolo é seguro de acordo com a definição de privacidade do modelo semi-honesto. A prova de segurança é feita no [Marques 2009].

## 4. Protocolo de comparação de inteiros

Um problema clássico da literatura é o Yao's Millionaires' Problem [Yao 1982]: dois milionários desejam saber qual é o mais rico, porém não querem revelar a quantidade de dinheiro que possuem ao outro. É importante ressaltar que a ordem de grandeza da diferença entre as fortunas não pode ser revelada, pois estaria-se assim fornecendo informações sigilosas ao adversário da disputa.

A comparação de dois números inteiros pode ser obtida facilmente, contudo é importante garantir a privacidade dos dados com relação à ordem de grandeza e também não se deve utilizar o intermédio de nenhuma entidade. Sendo assim, ao final da comparação, cada um dos participantes tem como resposta apenas um valor binário, como convenção, 1 para o valor maior e 0 para o menor na comparação.

A construção de um circuito lógico para realizar a comparação dos valores inteiros torna-se uma abordagem prática, já que atende aos requisitos de privacidade de dados e

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 2. Protocolo do Sistema Linear**

| | | |
|---|---|---|
| **Inicializador Confiável** | $Q, R \in_R SL(\mathbb{F}_q)$<br>$U \in_R \mathbb{F}_q^{n\times n}$<br>$\overrightarrow{s} \in_R \mathbb{F}_q^n$ | |
| Dados enviados pelo IC | $\mu_A := (R, V = RQ + U, \overrightarrow{s})$ | $\mu_B := (Q, U)$ |
| | **Alice** | **Bob** |
| **Entrada** | $X \in SL(\mathbb{F}_q), \overrightarrow{x} \in \mathbb{F}_q^n$ | $Y \in SL(\mathbb{F}_q), \overrightarrow{y} \in \mathbb{F}_q^n$ |
| Passo 1:<br>Chama $\Pi_{\langle \overrightarrow{x} \cdot \overrightarrow{y} \rangle}$<br>Chama $\Pi_{\langle \overrightarrow{x} \cdot \overrightarrow{y} \rangle}$ | | $P \in_R SL(\mathbb{F}_q)$ |
| | $(R; [PX - R]) := \Pi_{\langle \overrightarrow{x} \cdot \overrightarrow{y} \rangle}(X; P)$<br>$(\overrightarrow{s}; [P\overrightarrow{x} - \overrightarrow{s}]) := \Pi_{\langle \overrightarrow{x} \cdot \overrightarrow{y} \rangle}(\overrightarrow{x}; P)$ | |
| | | $M := [PX - R]Q + PYQ - U$<br>$\overrightarrow{c} := [P\overrightarrow{x} - \overrightarrow{s}] + P\overrightarrow{y}$<br>Envia $\mu_1 := (M, \overrightarrow{c})$ para Alice |
| Passo 2: | Se $\mu_1$ é invalido: aborta<br>$N := M + V$<br>$\overrightarrow{d} := \overrightarrow{c} + \overrightarrow{s}$<br>Encontra $\overrightarrow{t}$ tal que $N\overrightarrow{t} = \overrightarrow{d}$<br>Envia $\mu_2 := (\overrightarrow{t})$ para Bob | |
| Passo 3: | | Computa $\overrightarrow{z} = Q^{-1}\overrightarrow{t}$ |
| **Saída** | $u := \perp$ | $v := \overrightarrow{z}$ |

garante uma segurança contra interpretação de pequenos trechos do circuito, se feito de forma independente.

Outro paradigma computacional que apresenta características satisfatórias à construção de um circuito booleano é a divisão e conquista, como apresentado no [Garay et al. 2007]. Tal paradigma é útil, porque, ao segmentar o problema em problemas menores, pode-se encontrar um padrão e, através de recursividade, diminuir o tamanho do circuito e, portanto, o número de passos no protocolo.

### 4.1. Conceito do circuito

A comparação de 1 bit funciona da seguinte maneira: se $B \geq A$, responder 1, caso contrário, 0. Como representado na Tabela 3.

**Tabela 3. Relação de Entrada e Saída Esperados do Circuito Comparador de 1 Bit**

| | $A = 0$ | $A = 1$ |
|---|---|---|
| $B = 0$ | 1 | 0 |
| $B = 1$ | 1 | 1 |

Então o resultado é 1 para os casos de A = 0 ou B = 1. Assim, é formulado o circuito como: $\overline{A} \vee B$

 ©2012 SBC — Soc. Bras. de Computação

Para obter a informação de que dois elementos são iguais, é necessário fazer a operação de Ou exclusivo (XOR) entre A e B. Esse valor será usado futuramente para saber qual tabela é significativa.

A representação do circuito inicial, comparando-se números de apenas 1 bit, é mostrada na Figura 1.

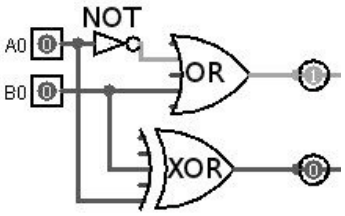



**Figura 1. Circuito Comparador de 1 Bit**

A representação acima obtém dois valores de saída, sendo o primeiro o resultado e o segundo um parâmetro de comparação utilizado quando o circuito se torna maior e mais complexo.

Para a comparação de 2 bits, utiliza-se a mesma ideia de construir tabelas de comparação, porém serão necessárias 2 tabelas parecidas.

Usando-se a notação $A_1A_0$ e $B_1B_0$ para representar os números tem-se duas tabelas, uma para comparação dos bits $A_0$ e $B_0$, e outra para comparação dos bits $A_1$ e $B_1$. A tabela de comparação dos bits menos significativos será idêntica à já mostrada no exemplo anterior e é representada por $\overline{A_0} \vee B_0$. Esse esquema é representado na Tabela 4.

**Tabela 4. Relação de Entrada e Saída Esperados do Bit Menos Significativo Circuito Comparador de 2 Bits**

| | $A_0 = 0$ | $A_0 = 1$ |
|---|---|---|
| $B_0 = 0$ | 1 | 0 |
| $B_0 = 1$ | 1 | 1 |

A tabela que representa a comparação dos bits mais significativos tem uma pequena modificação. Essa tabela é construída da mesma maneira que as anteriores porém a diagonal principal não representará que o valor da entrada de B é maior que o da entrada de A, mas sim que os valores são iguais e será utilizada a notação $x$ para esta diagonal. O valor $x$ da tabela é colocado, pois como essa tabela representa o bit mais significativo do valor de entrada, caso o valor de $A_1$ seja igual o de $B_1$, não se pode concluir que $B \geq A$, pois é necessário recorrer à outra tabela, que compara os bits menos significativos dos dados de entrada (apresentada nesta construção como a primeira tabela, que é a comparação de $A_0$ e $B_0$). Esse esquema é representado na Tabela 5.

Para verificar se o valor se encontra na diagonal principal, ou seja, se é igual a $x$, basta verificar se $A_1$ e $B_1$ são iguais (XOR). Assim tem-se em paralelo, 2 circuitos para a tabela $A_1A_0/B_1B_0$.

É necessário saber qual das duas tabelas construídas será a tabela determinante para o resultado da comparação, ou seja, deve-se descobrir qual bit (mais ou menos sig-

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 5. Relação de Entrada e Saída Esperados do Bit Mais Significativo Circuito Comparador de 2 Bits**

| | $A_1 = 0$ | $A_1 = 1$ |
|---|---|---|
| $B_1 = 0$ | X | 0 |
| $B_1 = 1$ | 1 | X |

nificativo) será o responsável pela comparação. Esse fato está relacionado ao caso de que, em situações com bits mais significativos diferentes, o resultado é determinado por esta comparação, em caso de bits iguais, o próximo bit mais significativo será o novo responsável pelo resultado da comparação. Tal recurso exemplifica como o problema foi subdividido em problemas menores, pois todas as comparações são feitas (bit a bit) e, após isto, só é necessário descobrir qual comparação será a relevante.

Utiliza-se o seguinte recurso: XOR da segunda tabela (a mais significativa) será negado e será uma multiplicação (AND) com o resultado da primeira tabela (caso o XOR seja 1, a multiplicação irá anular o valor da primeira tabela, caso seja 0, representará que é para ser usada a primeira tabela). Para ter o resultado final, tem-se uma soma (OU) do valor da primeira tabela com o da segunda (uma delas terá o resultado zerado pelo XOR, assim, a operação $X$ OU 0, será sempre igual a $X$). Esse esquema é representado na Figura 5.

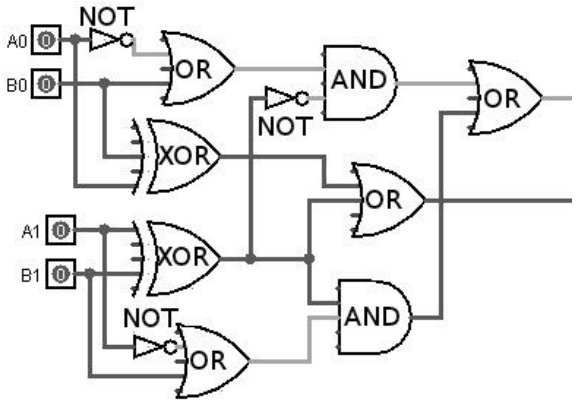



**Figura 2. Circuito Comparador de 2 Bits**

Ao analisar o problema, percebe-se que ao duplicar a estrutura já construída, obtém-se um circuito que compara 4 bits, sendo esta a medida adotada para comparar números maiores. A única modificação que deve ser construída é que o bit menos significativo não é modificado pela estrutura do XOR, já que caso todos os resultados sejam iguais até o bit menos significativo, é ele quem define qual será o resultado do problema.

O circuito descrito será utilizado para fazer a comparação de um número de bits predeterminado, podendo ser adaptado para números maiores, utilizando a duplicação do circuito.

Para garantir a privacidade dos dados foi utilizado o protocolo OT para a construção da porta lógica AND, como será descrito a seguir. Porém, para o circuito utilizar este recurso ao máximo, foi feita a substituição das portas lógicas OR e XOR, por respectivas

 ©2012 SBC — Soc. Bras. de Computação

portas AND, como segue a demonstração:

$A \vee B$ é substituído por $\overline{A \wedge B}$ e $A \oplus B$ é substituído por $\overline{(\overline{A} \wedge B) \wedge (A \wedge \overline{B})}$. Essas modificações podem ser provadas através de tabela verdade das configurações de cada porta.

### 4.2. Construição da porta AND utilizando Oblivious Transfer

O protocolo oblivious transfer é constituído por uma troca de mensagens em que nenhuma das partes pode saber informações sobre o outro participante. Foi utilizado o modelo em que as duas partes fornecem uma entrada e é retornado apenas uma saída. A construção descrita a seguir é baseado no [Ostrovski 2005] (veja também [Goldreich 2004]).

A partir da entrada de bits de Alice (a) e de Bob (b), cada uma delas gera uma entrada aleatória da seguinte forma: Alice gera uma entrada aleatória de bits $a^B$ e computa o resultado de $a^A = a \oplus a^B$. De maneira similar, Bob gera uma entrada aleatória de bits $b^A$ e computa o resultado de $b^B = b \oplus b^A$. O próximo passo é Alice enviar $a^B$ para Bob e Bob enviar $b^A$ para Alice.

Supondo que Alice e Bob possuem $x = x^A \oplus x^B$ e $y = y^A \oplus y^B$, respectivamente, e desejam obter o resultado de $x \wedge y$.

$$x \wedge y = (x^A \oplus x^B) \wedge (y^A \oplus y^B) = (x^A \wedge y^A) \oplus (x^B \wedge y^A) \oplus (x^A \wedge y^B) \oplus (x^B \wedge y^B)$$

Com essas informações, Alice consegue computar $x^A \wedge y^A$ e Bob consegue computar $x^B \wedge y^B$. Para computar os valores de $x^B \wedge y^A$ e $x^A \wedge y^B$, é necessário utilizar do protocolo OT. Será apresentada a prova de como resolver a expressão $x^A \wedge y^B$. O caso $x^B \wedge y^A$ é resolvido de forma similar.

1. Alice gera um bit aleatório $r^A$.
2. Alice fornece de entrada para o OT, os seguintes dados: $(x^A \wedge 0) \oplus r^A$, $(x^A \wedge 1) \oplus r^A$
3. Bob fornece a entrada $y^B$ para o protocolo OT.
4. OT retornará a Bob a saída $(x^A \wedge y^B) \oplus r^A$, que pode ser interpretado como $w^B$.

Após este processo, é concluído que $(x^A \wedge y^B) = r^A \oplus w^B$. De maneira similar, a expressão $(x^B \wedge y^B)$ será associada a um valor aleatório conhecido por Bob, $r^B$, e o valor obtido por Alice no OT, $w^A$.

1. Alice computa $(x \wedge y)^A = (x^A \wedge y^A) \oplus r^A \oplus w^A$
2. Bob computa $(x \wedge y)^B = (x^B \wedge y^B) \oplus r^B \oplus w^B$

Ao combinar o resultado das saídas de ambos os cálculos, é obtido o valor esperado da porta AND.

## 5. Implementação

Toda a implementação foi feita utilizando a linguagem de programação Python, esta linguagem foi escolhida por ser muito poderosa e de fácil utilização. Para realizar a comunicação foram utilizadas as funções da biblioteca *socket*. Outras bibliotecas utilizadas para auxiliar na implementação foram: *random, cPickle e numpy*.

Na implementação do protocolo do produto escalar e do sistema linear foi criado um programa que simulava o Inicializador Confiável, um programa que representava

 ©2012 SBC — Soc. Bras. de Computação

Alice e outro Bob. Como as operações foram feitas em cima de corpos finitos também foi necessário desenvolver algoritmos especiais para realizá-las, como por exemplo para fazer a inversão de matrizes, e também para utilizar algoritmos já existentes, como o Algoritmo de Euclides.

Na implementação do circuito comparador de inteiros, foi criado um programa que simula a comparação das entradas, um algoritmo que representava Alice e outro Bob. A implementação de portas lógicas foram reduzidas apenas à porta AND, já que através dessa porta e da porta lógica unária NOT, é possível representar todas as outras portas lógicas.

A estrutura da porta NOT simplesmente transforma o valor 0 em 1 e vice-versa, assim foi utilizado um incremento de uma unidade e retirado o módulo 2, do resultado. A porta AND exige a comunicação e troca de informações, onde o circuito utiliza-se de algoritmos criados para representação da oblivious transfer.

## 6. Conclusão

Foram apresentadas soluções para o cálculo do produto escalar entre dois vetores e para a comparação de inteiros, ambas seguras e com preservação de privacidade. As duas servem como base para aplicações maiores e demonstram a utilização de técnicas alternativas na implementação de protocolos criptográficos.

O protocolo do produto escalar utiliza o conceito do inicializador confiável, onde uma terceira parte realiza apenas o passo inicial do protocolo, e o da comparação de inteiros utiliza a emulação do problema através de um circuito booleano, que por sua vez pode ser implementado em software utilizando a simulação da porta AND com Oblivious Transfer.

Como nos dois protocolos testados as computações locais são relativamente simples, o maior fator limitante é o número de trocas de mensagens do protocolo. Nos testes executados numa rede local, o tempo de execução foi sempre menos de cinco segundos, assim mostrando que nossa abordagem é computacionalmente praticável.

## 7. Agradecimentos

Esta pesquisa foi financiada pela Fapemig, processo APQ-02719-1.

## Referências

Blundo, C., Masucci, B., Stinson, D. R., and Wei, R. (2002). Constructions and bounds for unconditionally secure non-interactive commitment schemes. *Des. Codes Cryptography*, 26(1-3):97–110.

Dowsley, R., van de Graaf, J., Marques, D., and Nascimento, A. C. A. (2010). A two-party protocol with trusted initializer for computing the inner product. In *WISA*, pages 337–350.

Garay, J. A., Schoenmakers, B., and Villegas, J. (2007). Practical and secure solutions for integer comparison. In *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 330–342. Springer.

 ©2012 SBC — Soc. Bras. de Computação

Goldreich, O. (2004). *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, New York, NY, EUA.

Marques, D. (2009). Protocolos para computação distribuida segura utilizando um inicializador confiável. Master's thesis, Universidade Federal de Minas Gerais, Brasil.

Ostrovski, R. (2005). Foundations of Cryptography – Oblivious Transfer – Course Notes. http://www.cs.ucla.edu/ rafail/TEACHING/WINTER-2005/L10/L10.pdf.

Pinkas, B., Schneider, T., Smart, N. P., and Williams, S. C. (2009). Secure two-party computation is practical. In *ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*.

Pontual, M. (2005). Protocolos para computaçãoo segura entre dois participantes para álgebra linear e estatística. Master's thesis, Universidade Federal de Pernambuco, PE, Brasil.

Rivest, R. (1999). Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer. http://people.csail.mit.edu/rivest/Rivest-commitment.pdf.

Yao, A. C.-C. (1982). Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, 3-5 November 1982, Chicago, Illinois, EUA*, pages 160–164. IEEE.

 ©2012 SBC — Soc. Bras. de Computação

# Senhas descartáveis em dispositivos móveis para ambientes de Telemedicina

**Thaís Bardini Idalino**[1] **and Dayana Pierina Brustolin Spagnuelo**[1]

[1] Universidade Federal de Santa Catarina
Departamento de Informática e de Estatística
Campus Universitário - Florianópolis - SC - Brazil
{thais.idalino,dayspagnuelo}@inf.ufsc.br

***Resumo.*** *Este trabalho é parte de um projeto que está sendo desenvolvido no Laboratório de Segurança em Computação [LabSEC ] da Universidade Federal de Santa Catarina [UFSC ] em parceria com o Laboratório de Informática Médica e Telemedicina [LabTelemed ] financiado pela Financiadora de Estudos e Projetos [FINEP ]. Tem como objetivo implantar o uso de senhas descartáveis na Rede Catarinense de Telemedicina, através de uma versão melhorada do Google Authenticator e do uso do ASI-HSM como servidor de autenticação. Para a implantação na Telemedicina, propõe-se o uso de um serviço de autenticação desenvolvido também no LabSEC para abstrair a camada do HSM.*

***Abstract.*** *This paper is part of a project that is being developed at the Laboratory of Computer Security (LabSEC) of the Federal University of Santa Catarina (UFSC) in association with the Laboratory of Medical Informatics and Telemedicine (LabTelemed) and sponsored by the Financier of Studies and Projects (FINEP). It aims to deploy the use of One-Time Passwords in the Santa Catarina Telemedicine Network, through an improved version of Google Authenticator and the use of the ASI-HSM as an authentication server. For the deployment in the Telemedicine environment, we propose the use of a authentication service developed at LabSEC to abstract the HSM's layer.*

## 1. Introdução

A utilização de senhas é algo que está presente no dia-a-dia da maioria das pessoas, desde senhas para efetuar transações bancárias, até senhas para acesso à sistemas mais simples ou redes sociais. Elas são a barreira mais utilizada em sistemas que precisam de alguma forma de autenticação e autorização de acesso, pelo fato de serem fáceis de aplicar tanto em nível de implementação quanto de usabilidade.

No ambiente de Telemedicina, a autenticação de um médico perante o sistema tem uma importância ainda maior. Ao provar sua identidade, o profissional tem em mãos ferramentas capazes de emitir laudos e analisar exames de pacientes, auxiliando principalmente os que residem em lugares distantes. Tais ferramentas em mãos erradas, podem apresentar risco à saúde dos pacientes pela possibilidade de emissão de

 ©2012 SBC — Soc. Bras. de Computação

laudos falsos. Também é possível expor dados que seriam de interesse apenas dos pacientes e seus respectivos médicos.

O fato de utilizar um *login* e senha para se autenticar no sistema não garante que quem os está inserindo é realmente a pessoa autorizada ao acesso. Diversos tipos de ameaças existem atualmente, dentre eles estão os ataques capazes de capturar as senhas que estão transitando pela rede, ou, de acordo com [da Silva 2007], até mesmo o elo mais fraco nesse mecanismo de autenticação: o próprio usuário. As pessoas tendem a escolher senhas fáceis e/ou utilizar a mesma para se autenticar em diferentes locais. Isso as torna vulneráveis, pois são fáceis de adivinhar e uma vez descobertas, podem comprometer uma quantidade imensurável de informações e dados.

Uma das maneiras de contornar esse problema é através da autenticação baseada em múltiplos fatores, que vem se tornando popular e que tem fácil implantação. Neste modelo, não apenas *login* e senha são utilizados, mas também a apresentação de um segundo fator de autenticação que pode ser desde um objeto de posse única do usuário, até características biométricas. Entre os mecanismos de autenticação de múltiplos fatores está o de senhas descartáveis (do inglês *One-Time Passwords*, OTP), apresentado como alternativa principalmente em sistemas bancários, mas também nas contas de usuário do Google e Facebook.

Este trabalho propõe a implantação de senhas descartáveis na autenticação do sistema da Rede Catarinense de Telemedicina (RCTM), utilizando como base a implementação já existente do Google, o Google Authenticator [Google Inc. 2012]. O restante do trabalho está organizado da seguinte maneira: Uma breve discussão da importância da segurança na autenticação em ambientes de telemedicina é discutida na seção 2. Na seção 3 é apresentada a proposta do trabalho, com foco no gerador de senhas descartáveis na subseção 3.1 e no servidor de autenticação na 3.2. Na seção 4 é apresentada a integração entre o sistema de telemedicina e a proposta sugerida. Por fim, na seção 5 são apresentadas as considerações finais e proposta de trabalhos futuros.

## 2. Segurança em ambientes de Telemedicina

A RCTM tem por finalidade facilitar o acesso a serviços de saúde por pacientes e médicos, como o acesso e emissão de exames e laudos, fazendo o uso da Internet. Dessa maneira, o paciente pode fazer um exame em sua cidade e enviá-lo pela rede a um médico especializado, este por sua vez pode analisá-lo e emitir o laudo de qualquer lugar que esteja [Cyclops 2010].

A RCTM pode ser caracterizada como um sistema de caráter social, cujas principais vantagens estão na acessibilidade e facilidade de assistência da população em geral. Segundo [Nakamura 2007], com a evolução constante e surgimento de novas tecnologias, as pessoas ganham em acessibilidade, produtividade e diversas facilidades, mas também os riscos devem ser avaliados e minimizados. Como apresentado em [Magalhães and Santos 2003], os problemas de segurança tornam-se ainda mais evidentes com o uso cada vez maior dessas tecnologias e, dentre estes problemas, tem-se

 ©2012 SBC — Soc. Bras. de Computação

como destaque o da autenticação.

Ainda de acordo com [Magalhães and Santos 2003], a segurança no processo de autenticação é uma questão fundamental, já que o acesso indevido à informações pode ser de grande risco. De acordo com [FFIEC 2005] o roubo de identidade é frequentemente encontrado como o resultado de uma autenticação utilizando um único fator (utilizando apenas um *login* e senha por exemplo), afirmando que esse modo de autenticação é inadequado. No ambiente de Telemedicina, o impacto de um ataque desta natureza é ainda maior, pela sensibilidade dos dados envolvidos. Se um indivíduo mal intencionado conseguir acesso às senhas de pacientes ou médicos, poderia desde visualizar informações particulares, como resultados de exames, até emitir laudos falsos, apresentando um risco à vida dos pacientes.

Tendo em vista estes ataques, acredita-se que a adoção de um processo de autenticação mais seguro neste ambiente pode ser uma boa solução para o problema. Através da garantia de identidade dos usuários que acessam o sistema, pode-se reduzir as chances de sucesso em ataques desta natureza, ou até extinguí-las.

### 2.1. Trabalhos Correlatos

De acordo com Haller et al [Haller et al. 1998], a ideia por trás do OTP foi inicialmente proposta por Leslie Lamport em seu artigo de 1981 [Lamport 1981], que também inspirou a criação do sistema de S/KEY [Haller 1995], no qual o OTP foi derivado.

Hoje em dia sua utilização está em grande evidência em ambientes bancários, como o Bradesco, por exemplo, que utiliza um tipo de OTP impresso em papel, contendo 70 senhas descartáveis que devem ser apresentadas juntamente com a senha do usuário para autenticação [Banco Bradesco S.A. 2012].

O *RSA SecurID Authenticator* [RSA Security Inc 2010] é outro exemplo. Ele utiliza um token que pode ser tanto por hardware (como pequenos dispositivos que podem ser carregados no bolso) quanto por software (disponíveis para celulares, para Windows e Mac OSX). Este token possui uma senha semente e gera o OTP através de cálculos matemáticos baseados nessa semente e num *timestamp* sincronizado com o servidor de autenticação. A cada 60 segundos uma nova senha é gerada, e para a autenticação, o usuário precisa entrar com seu *login*, sua senha e o OTP concatenado à ela.

Diversos trabalhos sobre a utilização de múltiplos fatores de autenticação em ambientes de Telemedicina foram encontrados, porém nenhum contribui de forma significativa para este trabalho.

## 3. Proposta

Com base na importância dos dados armazenados na RCTM, propõe-se a adoção de um modelo de autenticação que apresente maior segurança. Como citado em [FFIEC 2005], métodos de autenticação baseados em múltiplos fatores são mais difíceis de comprometer e mais confiáveis. Neste trabalho, será utilizado como segundo fator de autenticação as senhas descartáveis (do inglês, *One-Time Passwords*,

 ©2012 SBC — Soc. Bras. de Computação

OTP), onde a apresentação destas prova que o usuário além de possuir seu *login* e senha, também tem posse do dispositivo que as gera. O aplicativo de geração de senhas descartáveis utilizado nesse trabalho será baseado na implementação do Google Authenticator [Google Inc. 2012]. Nas subseções seguintes descreve-se as principais tecnologias envolvidas e o desenvolvimento da proposta.

### 3.1. Gerador de senhas

O gerador de senhas descartáveis Google Authenticator [Google Inc. 2012] está disponível para as plataformas Android, iOS e Blackberry e é utilizada para uma autenticação mais forte nas contas da empresa. Nesta implementação, para que o cliente e o servidor de autenticação possam gerar as mesmas senhas e de maneira independente, eles precisam passar por um processo de inicialização. Neste processo faz-se um acordo de uma senha inicial (conhecida como *master secret*) entre eles, e a partir dela são capazes de gerar as mesmas senhas descartáveis isoladamente.

O aplicativo suporta a implementação de OTP com parâmetros baseados em contador (*HMAC-Based One Time Password*, HOTP) [M'Raihi et al. 2005] e em tempo (*Time-based One Time Password*, TOTP) [M'Raihi et al. 2011]. Ela está em conformidade com as respectivas RFCs, qualquer alteração nas mesmas por avanços na criptoanálise por exemplo, pode também ser alterada no aplicativo. Ao se utilizar o parâmetro de tempo, o relógio do dispositivo deve ser sincronizado com o do servidor. Já com um parâmetro baseado em contador, o mesmo recebe um valor inicial, geralmente zero. Este contador é incrementado à cada geração de senha no dispositivo do usuário e à cada autenticação realizada com sucesso no servidor. Em ambos os casos há a possibilidade de cliente e servidor estarem dessincronizados, neste caso é previsto que o servidor adote uma política que possua uma janela de ressincronização, gerando uma sequência de OTPs anteriores e posteriores, se certificando assim em que posição o cliente se encontra para sincronizar-se à ele.

Para utilizar o aplicativo, o usuário instala o mesmo em seu celular e realiza o cadastro de uma conta, inserindo seus dados e a *master secret*. Esta pode ser fornecida manualmente, ou obtida através da leitura de um código de barras em 2D, conhecido como *QR Code*. O acordo desta senha é feito pelo browser, e o usuário a recebe direto na tela do computador, em texto claro caso tenha sido escolhida a primeira opção, e em forma de imagem para a segunda.

De acordo com [Cheng 2011], geradores de senhas descartáveis em software armazenam a *master secret* no próprio dispositivo, e se copiada por um indivíduo malicioso, pode comprometer a segurança do sistema. Com base na análise do código do aplicativo Google Authenticator, pode-se observar uma possível vulnerabilidade proveniente do processo de armazenamento da *master secret*, tendo em vista que além de armazená-la no *smartphone* do usuário, o mesmo é feito em claro no seu banco de dados. A *master secret* pode ser considerada a base de todo o processo de geração das senhas descartáveis e apenas o servidor e o usuário devem detê-la. Entretanto, se um indivíduo mal intencionado obter acesso ao *smartphone* do usuário original e extraí-la

 ©2012 SBC — Soc. Bras. de Computação

do seu banco de dados, ele pode ser capaz de gerar as mesmas senhas e se passar por esse usuário no processo de autenticação.

Tendo em vista este problema, uma solução sugerida é a de armazenar essa *master secret* cifrada com um PIN (*Personal Identification Number*), sendo este uma senha que apenas o usuário conheça. Dessa maneira, além de ter acesso ao celular do usuário e extrair a *master secret*, o indivíduo mal intencionado deveria também ter conhecimento do PIN, caso contrário de nada adiantaria tê-la. Para evitar que o PIN seja extraído do banco de dados, este não fica armazenado no *smartphone*. Sua validação é dada unicamente pelo sucesso ou fracasso na decifragem da *master secret*, quando o usuário o fornece na aplicação.

Outra alternativa proposta visa garantir que a senha descartável foi gerada única e exclusivamente no *smartphone* para o qual foi destinada. Para isto, propõe-se a modificação do algoritmo de geração de OTPs para levar em consideração não só a *master secret* e um dos parâmetros (de tempo ou contador), mas também incluir em sua geração a Identificação Internacional de Equipamento Móvel (do inglês, *International Mobile Equipment Identity*, IMEI), que é um valor único que identifica o dispositivo. Com isso, de nada adiantaria um indivíduo mal intencionado obter a *master secret* do usuário, já que o dispositivo em que ele geraria os OTPs não possuiria o mesmo IMEI. Esta alternativa também impede a clonagem do aplicativo, uma vez que a obtenção do IMEI é diretamente dependente dos métodos disponibilizados pela plataforma do dispositivo. No entanto, se um indivíduo mal intencionado conseguir obtê-lo por meio de outras ferramentas ou aplicativos, ainda seria necessário obter o PIN para decifrar a *master secret*.

### 3.2. Servidor de autenticação

Em modelos de autenticação baseados em OTP, proteger a *master secret* contra roubos não significa somente proteger o gerador, pois existe uma cópia da mesma no servidor de autenticação. Além disto, é um requisito comum a todos os modelos de autenticação que a máquina responsável pelo armazenamento das credenciais dos usuários seja confiável. Desta forma este trabalho também propõe a troca do servidor de autenticação por um Módulo de Segurança Criptográfico (do inglês, *Hardware Security Module*, HSM), para torná-lo mais resistente a ataques.

HSMs são dispositivos usados principalmente em aplicações que requerem um maior rigor no armazenamento e processamento de informações sensíveis, como por exemplo em aplicações militares e bancárias. Normalmente são usados para o armazenamento de chaves criptográficas, processamento de autenticação de usuários, execução de funções criptográficas, entre outros [de Souza 2008].

Para o desenvolvimento deste trabalho, foi utilizado o ASI-HSM. De acordo com [de Souza 2008], ele é um módulo de segurança criptográfica com tecnologia brasileira, de código aberto e de menor custo. Foi patrocinado pelo GT ICPEDU II e projetado pela empresa brasileira Kryptus [Kryptus ] em parceria com a Rede Nacional de Ensino e Pesquisa [RNP ]. O software responsável pela gerência do ASI-HSM, o

 ©2012 SBC — Soc. Bras. de Computação

OpenHSMd [OpenHSMd ], foi desenvolvido pelo LabSEC.

Por se tratar de um HSM, este já possui barreiras contra os principais ataques sendo, portanto, menos vulnerável se comparado a um servidor simples. O ASI-HSM utiliza o OpenHSMd, que é um *firmware* voltado à gerência do ciclo de vida de chaves criptográficas, o qual precisaria ser completamente modificado para atender às necessidades do modelo proposto. Porém uma modificação no OpenHSMd não faria sentido, uma vez que seu propósito inicial pouco tem a ver com autenticações baseadas em OTP. Além do mais, isso aumentaria a interface de interação com o HSM, aumentando assim a vulnerabilidade do mesmo, uma vez que se aumenta a variedade de ataques aplicáveis.

A melhor alternativa encontrada foi a criação de um novo *firmware* específico para o modelo. Este possui interface somente para duas ações: criação de usuários e autenticação. Para a criação de usuários, utiliza-se um identificador único do usuário, o IMEI do seu *smartphone* e o tipo de OTP escolhido por ele e retorna a *master secret* já computada. A autenticação de usuários se dá pelo fornecimento do identificador do usuário e do OTP do mesmo e retorna somente se este é um OTP válido ou não. Desta forma o novo firmware possui uma interface bastante restrita.

O fato de todas as comunicações serem feitas através de um túnel seguro (*Secure Sockets Layer*, SSL) e a interface ser bastante restrita aumenta a dificuldade de um atacante obter sucesso em uma de suas tentativas. Além disto, se ainda assim um atacante conseguir penetrar o HSM, as *master secrets* estão cifradas pela chave do HSM, fazendo com que seja necessário obter-se também a chave privada deste, do contrário somente se obteria texto ilegível.

## 4. Integração com o Framework de Autenticação

Para que seja possível a implantação de um sistema mais seguro de autenticação na RCTM, foi desenvolvido um serviço de autenticação que serve como uma interface entre o HSM e o sistema. Este serviço funciona como um *web service* que tem o propósito de ser genérico, independente de linguagem e abstrair a camada do HSM. O mesmo provê diversos métodos de autenticação que podem ser utilizados como um fator adicional, e dentre estes métodos, está o de senhas descartáveis proposto neste trabalho. O framework é flexível, de maneira a não impedir que um médico emita um lado caso esqueça seu *smartphone* em casa, por exemplo.

O sistema de Telemedicina é o responsável por prover a interface gráfica e ele se comunica diretamente com o serviço para solicitar a autenticação e operações gerenciais. O sistema repassa os dados provenientes do usuário ou dele próprio para o serviço, que os manipula e solicita operações no HSM quando necessário.

Para o cadastro de um usuário que deseja utilizar OTP como um segundo fator de autenticação, o mesmo precisa entrar com seus dados no sistema de Telemedicina, escolhendo o tipo de OTP (contador ou tempo) que deseja utilizar. A Telemedicina por sua vez solicitará ao serviço a vinculação desse usuário com este tipo de

 ©2012 SBC — Soc. Bras. de Computação

autenticação. O serviço irá solicitar ao HSM a criação de um usuário de OTP, passando como parâmetro um identificador do usuário, o IMEI fornecido pelo mesmo e o tipo de OTP. O HSM calcula a *master secret* e a retorna ao serviço. Este finaliza o processo de cadastro do usuário enviando a *master secret* juntamente com o tipo de OTP escolhido pelo usuário e o *username* que ele já utiliza para o sistema de Telemedicina, codificados em um QR Code. A Telemedicina será a responsável por disponibilizá-lo ao usuário pelo *browser*, que através do aplicativo em seu celular pode ler o QR Code e obter todos os dados necessário para a criação de seu usuário no aplicativo, sem precisar entrar com dados, como no modelo original. Por fim, antes de a *master secret* ser salva no banco de dados do celular, é solicitado um PIN ao usuário para cifrá-la. Este processo pode ser visto na Figura 1.

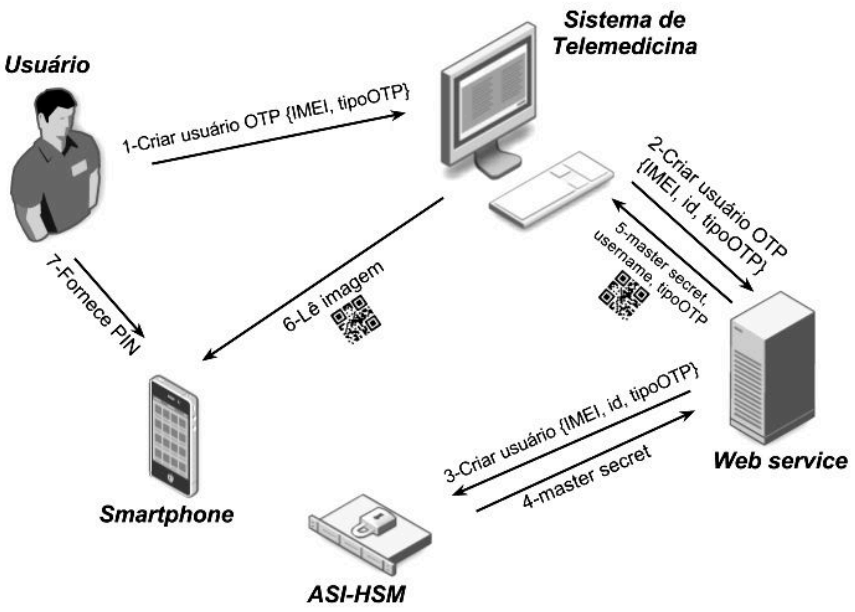



**Figura 1. Processo de criação de usuário OTP**

Já o processo de autenticação se dá pelo fornecimento do *login*, senha e OTP do usuário através da interface gráfica do serviço de Telemedicina. Para que o usuário obtenha seu OTP, primeiro precisa fornecer o PIN com o qual a *master secret* foi cifrada. A Telemedicina por sua vez repassa os dados do usuário ao serviço de autenticação, que faz a verificação de usuário e senha. Este solicita ao HSM a verificação do usuário e OTP apresentados. Caso todas retornem com sucesso, o usuário é autenticado no sistema. Este processo pode ser visto na Figura 2.

## 5. Considerações Finais

Neste trabalho foi apresentado um modelo de segundo fator de autenticação baseado em senhas descartáveis e sua respectiva implementação. Esse modelo tem como propósito aumentar a confiabilidade na autenticação do sistema de Telemedicina através do uso de um aplicativo derivado do Google Authenticator. Propõe também o uso do

 ©2012 SBC — Soc. Bras. de Computação

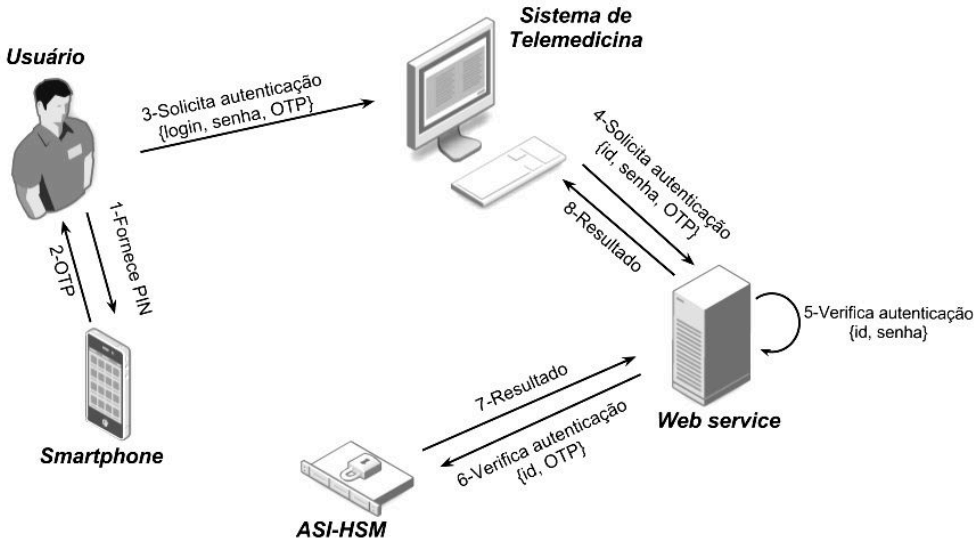



**Figura 2. Processo de autenticação de usuário OTP**

ASI-HSM com um novo firmware para fazer o papel do servidor de autenticação, e um *web service* que fará a comunicação entre o sistema de Telemedicina e o HSM, além de prover outros meios de autenticação.

Como foi mencionado, o uso de autenticação de apenas um fator compromete o sistema como um todo, principalmente pelos descuidos do usuário na hora de escolher e manipular suas senhas. A autenticação de múltiplos fatores utilizando senhas descartáveis se mostrou uma solução adequada, pelo fato de serem geradas por um aplicativo e de serem de natureza aleatória, não sendo passíveis de ataques de dicionário ou engenharia social. Tendo em vista que cada senha é utilizada apenas uma vez, a interceptação da mesma através da rede deixa de ser um problema. Um atacante que possui uma sequência de senhas com o intuito de construir uma função para geração das próximas, tem tantas chances de obter sucesso quanto um que não a possui, dado que o melhor ataque contra o algoritmo é a força bruta [M'Raihi et al. 2005].

Ainda de acordo com [M'Raihi et al. 2005], análise de segurança pode ser aproximada pela função:

$$Sec = \frac{s.v}{10^{Digit}}$$

Onde **Sec** é a probabilidade de sucesso de um atacante, **s** é o tamanho da janela de ressincronização prevista pelo modelo, **v** é o número de tentativas de autenticação antes do bloqueio do usuário e **Digit** é a quantidade de dígitos da senha OTP.

A implementação apresentada utiliza como gerador de senhas descartáveis o próprio *smartphone* do usuário. Pelo fato de este ser um dispositivo de uso pessoal, os usuários estão habituados com ele e não terão grandes problemas para entender o funcionamento do aplicativo. Além do mais, um ataque através do roubo do dispositivo seria facilmente percebido.

 ©2012 SBC — Soc. Bras. de Computação

A integração através de um *web service* de autenticação trouxe muitas vantagens para o sistema de Telemedicina, pois abstraiu completamente o uso do HSM, separou o processo de autenticação do de autorização e é independente de linguagem. Este último é muito importante pois torna possível a integração do *web service* com outros sistemas.

Como melhoria do trabalho proposto, sugere-se a utilização de outras características do dispositivo móvel do usuário na geração das senhas descartáveis. Uma alternativa seria a utilização dos dados contidos no seu *smartcard*, aumentando ainda mais a relação entre a senha gerada e o *smartphone*. Outra melhoria consiste em modificar o processo de *setup* da *master secret*, pois da maneira como é implementado hoje, a mesma é mostrada diretamente na tela do computador do usuário. Qualquer um que conseguir obtê-la no período em que ela está exposta pode utilizá-la para gerar as mesmas senhas do usuário original.

## Referências

[Banco Bradesco S.A. 2012] Banco Bradesco S.A. (2012). Cartão chave de segurança bradesco. http://www.bradescoseguranca.com.br.

[Cheng 2011] Cheng, F. (2011). Security attack safe mobile and cloud-based one-time password tokens using rubbing encryption algorithm. *Mob. Netw. Appl.*, 16(3):304–336.

[Cyclops 2010] Cyclops (2010). Sistema catarinense de telemedicina e telessaúde. https://www.telemedicina.ufsc.br/rctm.

[da Silva 2007] da Silva, D. R. P. (2007). *A memória humana no uso de senhas*. PhD thesis, Pontifícia Universidade Católica do Rio Grande do Sul.

[de Souza 2008] de Souza, T. C. S. (2008). Aspectos técnicos e teóricos da gestão do ciclo de vida de chaves criptográficas no openhsm. Master's thesis, Universidade Federal de Santa Catarina.

[FFIEC 2005] FFIEC (2005). Authentication in an internet banking environment. http://www.ffiec.gov/press/pr101205.htm.

[FINEP ] FINEP. Financiadora de Estudos e Projetos. http://www.finep.gov.br/.

[Google Inc. 2012] Google Inc. (2012). Google authenticator. http://code.google.com/p/google-authenticator.

[Haller 1995] Haller, N. (1995). The S/KEY One-Time Password System. RFC 1760 (Informational).

[Haller et al. 1998] Haller, N., Metz, C., Nesser, P., and Straw, M. (1998). A One-Time Password System. RFC 2289 (Standard).

[Kryptus ] Kryptus. Kryptus. http://kryptus.com.br/.

[LabSEC ] LabSEC. Laboratório de Segurança em Computação. http://www.labsec.ufsc.br/.

 ©2012 SBC — Soc. Bras. de Computação

[LabTelemed ] LabTelemed. Laboratório de Informática Médica e Telemedicina. http://www.telemedicina.ufsc.br/joomla/.

[Lamport 1981] Lamport, L. (1981). Password authentication with insecure communication. *Commun. ACM*, 24(11):770–772.

[Magalhães and Santos 2003] Magalhães, P. S. and Santos, H. D. (2003). Biometria e autenticação. In *Actas da 4a Conferência da Associação Portuguesa de Sistemas de Informação*, pages 2 – 5, Porto. Portugal.

[M'Raihi et al. 2005] M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., and Ranen, O. (2005). HOTP: An HMAC-Based One-Time Password Algorithm. RFC 4226 (Informational).

[M'Raihi et al. 2011] M'Raihi, D., Machani, S., Pei, M., and Rydell, J. (2011). TOTP: Time-Based One-Time Password Algorithm. RFC 6238 (Informational).

[Nakamura 2007] Nakamura, E. T. (2007). *Segurança de redes em ambientes corporativos*. Novatec Editora Ltda., São Paulo, SP - Brasil.

[OpenHSMd ] OpenHSMd. OpenHSMd. https://projetos.labsec.ufsc.br/openhsmd.

[RNP ] RNP. Rede Nacional de Ensino e Pesquisa. http://www.rnp.br/.

[RSA Security Inc 2010] RSA Security Inc (2010). *RSA SecurID Two-factor Authentication*. RSA Security Inc.

[UFSC ] UFSC. Universidade Federal de Santa Catarina. http://ufsc.br/.

 ©2012 SBC — Soc. Bras. de Computação

# Modelagem e Verificação Formal de Aspectos de Tempo Real do Protocolo Kerberos

**Bruno Marques Cremonezi**[1], **Guilherme Albuquerque Pinto**[1]

[1]Departamento de Ciência da Computação (DCC)
Instituto de Ciências Exatas (ICE) – Universidade Federal de Juiz de Fora (UFJF)
Rua José Lourenço Kelmer, s/n, São Pedro – 36.036-900 – Juiz de Fora – MG – Brasil

{brunomarques,guilherme.pinto}@ice.ufjf.br

***Abstract.*** *The Kerberos authentication protocol uses timestamps inside encrypted messages in order to prevent replay attacks, allowing application servers to distinguish between new messages and old ones. In this way, the service availability depends on the encryption and decryption times, and on the message congestion in the server. In this paper, we use the Timed Automata formalism and the UPPAAL verification tool to model these real-time aspects of the Kerberos protocol. The modelling allowed the suggestion of a modification in the protocol which makes it less dependent on those factors, increasing the service availability when many clients try to access the server in a short time interval.*

***Resumo.*** *O protocolo de autenticação Kerberos utiliza* timestamps *dentro de mensagens criptografadas para prevenir ataques do tipo* replay*, permitindo que um servidor de aplicação possa distinguir mensagens novas de antigas. Assim, a disponibilidade do serviço se torna dependente dos tempos de cifragem e decifragem, e do acúmulo de mensagens no servidor. Neste trabalho, utilizamos o formalismo de Autômatos Temporizados e a ferramenta UPPAAL para modelar estes aspectos de tempo real do Kerberos. A modelagem permitiu a identificação de uma alteração no protocolo que o torna menos dependente daqueles fatores, aumentando a disponibilidade do serviço em situações onde muitos clientes tentam utilizar o servidor em um periodo curto de tempo.*

## 1. Introdução

O protocolo Kerberos [Neuman et al. 2005] é amplamente utilizado na prática para prover autenticação e distribuição de chaves em redes inseguras. Ele é baseado no protocolo Needham-Schroeder de autenticação com chaves simétricas [Needham and Schroeder 1978] e utiliza *timestamps*, como sugerido em [Denning and Sacco 1981], para diminuir o número de mensagens no protocolo e prevenir ataques do tipo *replay*. O uso de *timestamps* permite que um servidor de aplicação, ao receber uma mensagem de um cliente, possa se certificar de que ela é nova, gerada recentemente no cliente, e não uma repetição mal-intencionada (*replay*). Dessa forma, o protocolo é dependente do tempo real de ocorrência dos eventos e, consequentemente, do tempo gasto pelo servidor de aplicação em decifragem e o acúmulo de mensagens. As vantagens e desvantagens do uso de *timestamps* são discutidas, por exemplo, em [Bellovin and Merritt 1991]. Estes aspectos de tempo real do Kerberos não estão diretamente relacionados à segurança do conteúdo das mensagens criptografadas

 ©2012 SBC — Soc. Bras. de Computação

e sim com a disponibilidade dos serviços. O administrador do sistema precisa definir constantes de tempo para expiração de mensagens no cliente e no servidor de aplicação. Intuitivamente, se estes tempos não estiverem bem calibrados, em relação ao número de clientes, a disponibilidade do serviço pode ficar prejudicada.

A área de verificação formal de protocolos de segurança é bastante tradicional e várias abordagens e ferramentas são utilizadas na prática. Veja, por exemplo, o *survey* [Blanchet 2012]. Em geral, a verificação busca encontrar erros lógicos no protocolo, não considerando o tempo real de ocorrência dos eventos. Temos conhecimento de dois trabalhos recentes que modelam tempo real em protocolos como o Kerberos [Corin et al. 2007, Li and Pang 2010], porém relacionando custos de tempo de cifragem e decifragem com os tempos estimados de quebra de uma chave simétrica por criptoanálise. Nestes trabalhos citados aparece a figura de um intruso e a modelagem considera uma única execução do protocolo entre um cliente e um servidor de aplicação. Neste presente artigo, utilizamos o formalismo de Autômatos Temporizados [Alur and Dill 1994, Bengtsson and Yi 2004], também utilizado em [Corin et al. 2007], para modelar o protocolo Kerberos considerando os aspectos de tempo real e disponibilidade envolvendo vários clientes legítimos e um servidor de aplicação, sem considerar a presença de um intruso e sem fazer considerações sobre a segurança da primitiva criptográfica utilizada. A verificação foi realizada com a ferramenta UPPAAL (uma colaboração das universidades de Uppsala, Suécia e Aalborg, Dinamarca) [Behrmann et al. 2004, Behrmann et al. 2011] e permitiu a indentificação de uma alteração na implementação do protocolo que o torna menos dependente dos custos de tempo de cifragem e decifragem e do acúmulo de mensagens no servidor, aumentando a disponibilidade do serviço em situações onde muitos clientes tentam utilizar o servidor em um periodo curto de tempo.

O restante deste artigo está organizado da seguinte forma. A Seção 2 explica resumidamente o protocolo Kerberos e o seu uso de *timestamps*. A Seção 3 apresenta conceitualmente o formalismo de Autômatos Temporizados e a ferramenta UPPAAL. Em seguida, a Seção 4 discute os modelos utilizados e a proposta de modificação na implementação do protocolo. Por fim, a Seção 5 conclui o artigo discutindo possíveis direções de trabalhos futuros.

## 2. O protocolo Kerberos e *Timestamps*

O protocolo Kerberos utiliza criptografia de chave simétrica para realizar autenticação mútua e distribuição de chaves entre clientes e servidores em uma rede insegura. Uma execução típica do protocolo consiste de 6 mensagens, como representado na Figura 1. Nas mensagens 1 e 2, um usuário logado no cliente $C$ utiliza a sua senha pessoal para se autenticar no servidor *Kerberos Authentication Server* (KAS), que envia ao cliente o *Ticket Granting Ticket* (TGT) cuja validade geralmente é de algumas horas. A função do TGT é permitir que o cliente possa requerer ao Kerberos conexão com vários serviços diferentes sem a necessidade de o usuário digitar novamente a sua senha pessoal (*single sign-on*). Nas mensagens 3 e 4, o cliente solicita ao Kerberos uma chave de sessão para conexão com o servidor de aplicação $S$, utilizando como credenciais o TGT e um autenticador $\{C, T_a\}K_{\text{cg}}$, cuja função será explicada em breve. O *Ticket Granting Service* (TGS) devolve ao cliente um *ticket* de serviço (TS) contendo a chave de sessão $K_{\text{cs}}$. Nas mensagens 5 e 6, finalmente o cliente envia ao servidor de aplicação $S$ o *ticket* de serviço TS e um autenticador $\{C, T_b\}K_{\text{cs}}$. De fato, as mensagens 3 e 4 são análogas às mensa-

 ©2012 SBC — Soc. Bras. de Computação

gens 5 e 6, podendo o TGS ser considerado como um servidor de aplicação especial do Kerberos, que não necessita estar rodando na mesma máquina do KAS. Vale notar que as chaves simétricas $K_{\text{ag}}$, $K_{\text{c}}$ e $K_{\text{gs}}$ são chaves de longo prazo (*long-term keys*) previamente cadastradas no banco de dados do Kerberos. Já as chaves simétricas $K_{\text{cg}}$ e $K_{\text{cs}}$ são chaves de sessão de curto prazo (*short-term keys*) geradas pelo Kerberos permitir a comunicação entre clientes e servidores.

1. AS_REQ $C \rightarrow A : G, N_1$
2. AS_REP $A \rightarrow C : \overbrace{\{C, G, K_{\text{cg}}, T1_{\text{inicio}}, T1_{\text{fim}}\}K_{\text{ag}}}^{\textit{Ticket Granting Ticket}\ (\text{TGT})}, \{C, G, K_{\text{cg}}, T1_{\text{inicio}}, T1_{\text{fim}}, N_1\}K_{\text{c}}$
3. TGS_REQ $C \rightarrow G : S, N_2, \text{TGT}, \{C, T_a\}K_{\text{cg}}$
4. TGS_REP $G \rightarrow C : \overbrace{\{C, S, K_{\text{cs}}, T2_{\text{inicio}}, T2_{\text{fim}}\}K_{\text{gs}}}^{\text{Ticket do Serviço (TS)}}, \{C, S, K_{\text{cs}}, T2_{\text{inicio}}, T2_{\text{fim}}, N_2\}K_{\text{cg}}$
5. AP_REQ $C \rightarrow S : \text{TS}, \overbrace{\{C, T_b\}K_{\text{cs}}}^{\text{Autenticador}}$
6. AP_REP $S \rightarrow C : \{T_b\}K_{\text{cs}}$

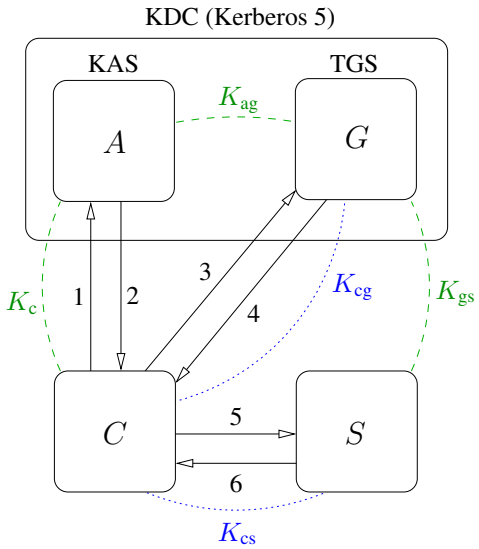



**Figura 1. Sequência de mensagens no protocolo Kerberos**

O servidor de aplicação decifra o TS utilizando a sua chave $K_{\text{gs}}$, obtendo a chave de sessão $K_{\text{cs}}$, com a qual ele pode decrifrar o autenticador para obter o *timestamp* $T_b$. Neste momento o servidor $S$ compara o seu tempo físico com $T_b$. Tipicamente a diferença deve ser de no máximo 2 minutos, para que o servidor considere as mensagens como novas. Além disso, para garantir mais proteção contra ataques do tipo *replay*, o servidor deve manter uma *cache* de autenticadores, de modo que mesmo dentro da tolerância de 2 minutos ele seja capaz de identificar o recebimento de autenticadores repetidos. Por final, a mensagem 6 realiza a autenticação mútua (de $S$ em $C$) e é opcional. Os detalhes da especificação e implementação do Kerberos podem ser encontrados em [Neuman et al. 2005].

**Ataques do tipo *replay*.** Caso não houvesse o autenticador contendo o *timestamp* na mensagem 5, um intruso poderia gravar o TS de uma execução legítima do protocolo, e num momento futuro ainda dentro da validade do TS, reenviar o TS para o servidor $S$, que aceitaria a mensagem. Neste exemplo simples de ataque, o intruso estaria autenticado em $S$ como $C$. Isso pode ser considerado uma falha de segurança, ainda que a chave de sessão $K_{\text{cs}}$ não esteja comprometida e, portanto, o intruso não possa se comunicar com $S$.

## 3. Autômatos Temporizados e UPPAAL

Conceitualmente, Autômatos Temporizados são uma generalização natural de Autômatos Finitos para um domínio de tempo contínuo. O autômato possui um número finito de variáveis reais, chamadas relógios, cujos valores crescem com derivada 1 em relação

 ©2012 SBC — Soc. Bras. de Computação

à passagem do tempo. Cada transição do autômato é anotada com uma restrição sobre os valores de cada relógio e uma trajetória pode tomar uma transição apenas se os relógios satisfizerem a sua restrição. Uma transição pode, ainda, zerar o valor de um conjunto de relógios após ser tomada, o que é a única operação permitida sobre os relógios. Nas restrições, é importante notar, o valor de um relógio pode ser comparado apenas contra constantes racionais e não contra o valor de outros relógios. Intuitivamente, o autômatos podem relembrar os tempos de ocorrências de um conjunto finito de eventos. Por exemplo, na Figura 4 da Seção 4 a transição entre os estados n1 e n2 zera o valor do relógio t, de modo que a transição entre n2 e n3 pode ser tomada apenas exatamente time_decrypt*2 unidades de tempo depois. Para uma descrição detalhada da sintaxe e semântica de autômatos temporizados veja, por exemplo, [Alur and Dill 1994, Bengtsson and Yi 2004].

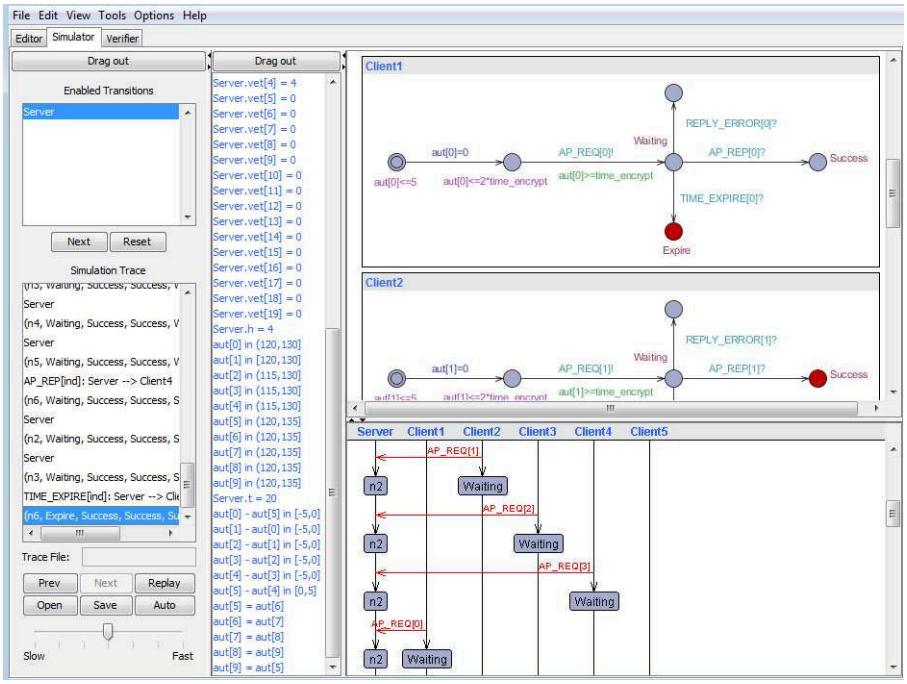



**Figura 2. UPPAAL: Tela do Simulador**

A ferramenta UPPAAL [Behrmann et al. 2004, Behrmann et al. 2011] permite a modelagem de um sistema através da definição de vários autômatos básicos, um para cada componente do sistema. Além de um editor para os autômatos, a ferramenta possui um simulador de trajetórias do sistema e um módulo de verificação automática de propriedades. O módulo de verificação utiliza algoritmos e estruturas de dados disponíveis na literatura para realizar *model-checking* do sistema, que formalmente é o produto cartesiano dos autômatos básicos, contra propriedades expressas em um subconjunto da lógica TCTL (*Timed Computation Tree Logic*). Para o presente trabalho, foram utilizadas apenas propriedades básicas de alcançabilidade, expressas pelo predicato `E<>`$\varphi$, que denota a existência de uma trajetória em que a fórmula $\varphi$ se torna válida em algum momento futuro. Por exemplo, na Figura 2, após a verificação da propriedade `E<>Client1.Expire`, o UPPAAL carrega no simulador de trajetórias um exemplo de trajetória que satisfaz a pro-

 ©2012 SBC — Soc. Bras. de Computação

priedade. Note que, nesta figura, o autômato `Client1` está no estado `Expire` e, por exemplo, o valor do relógio `aut[0]` está no intervalo $(120, 130]$.

Na lógica TCTL, conceitualmente, o quantificador `A` denota "para toda trajetória", enquanto `E` denota "existe uma trajetória". Esses quantificadores devem ser combinados com os quantificadores <> e `[]` que denotam, respectivamente, "em algum estado da trajetória (*eventually*)" e "em todos os estados da trajetória". A ferramenta UPPAAL, consegue apresentar uma trajetória no seu simulador quando uma propriedade do tipo `E<>`$\varphi$ é verdadeira (exemplo), ou quando uma propriedade do tipo `A[]`$\varphi$ é falsa (contra-exemplo).

## 4. Modelos e Resultados

Inicialmente, modelamos todas as seis mensagens de uma execução do Kerberos, para vários clientes e um servidor. O modelo possuia 5 autômatos: cliente, servidor, KAS, TGS e um canal modelando a rede. Por um lado, como os algoritmos implementados no UPPAAL têm consumo de memória exponencial em relação ao número de autômatos e relógios, tivemos dificuldades em realizar verificações completas mesmo de propriedades básicas como alcançabilidade. Por outro lado, porém, a experiência de modelar todo o protocolo permitiu observar em simulações que, intuitivamente: a modelagem das mensagens 1, 2, 3 e 4 não influenciavam decisivamente na questão da temporização no servidor de aplicação; os aspectos lógicos do Kerberos não dependem exatamente dos aspectos de tempo real, que estão relacionados apenas a ataques do tipo *replay*; e, principalmente, que a implementação padrão do Kerberos torna a disponibilidade dos serviços dependente dos tempos de cifragem e decifragem, e do acúmulo de mensagens no servidor (essa última questão será detalhada melhor na seção seguinte). Com isso, verificamos que poderíamos modelar apenas as mensagens 5 e 6, sem prejuízo para a verificação, gerando um modelo simplificado que permitiu o uso do UPPAAL com mais clientes.

### 4.1. Implementação padrão RFC4120

Consideramos uma situação onde muitos clientes legítimos requisitam acesso ao mesmo servidor de aplicação em um curto período de tempo. O servidor precisa, de alguma forma, acumular as mensagens em uma fila para processá-las sequencialmente. Partimos da hipótese de que o servidor possui apenas um processador, de modo que mesmo utilizando *threads*, o processamento é na prática sequencial. Todas as mensagens, dos clientes legítimos, chegam ao servidor dentro do prazo de validade dos respectivos autenticadores. É importante notar que, se não houver qualquer atraso no processamento, todas as mensagens serão consideradas válidas e o protocolo será completado. Para efeito de proteção contra ataques do tipo *replay*, o servidor de aplicação precisa ser capaz de identificar que uma mensagem chegou fora do prazo de validade, quer dizer, que a diferença entre o *timestamp* e o instante em que a mensagem chegou no servidor é maior do que o prazo de validade. No entanto, a implementação padrão do Kerberos [Neuman et al. 2005] assume que qualquer atraso devido à rede ou a custos de processamento está considerado dentro do prazo típico de 2 minutos de validade dos autenticadores. Esta situação está modelada nos autômatos descritos a seguir.

O autômato paramétrico da Figura 3 modela o envio da mensagem 5 (AP_REQ) e recebimento da mensagem 6 (AP_REP) por um cliente. A parametrização pelo índice id

 ©2012 SBC — Soc. Bras. de Computação

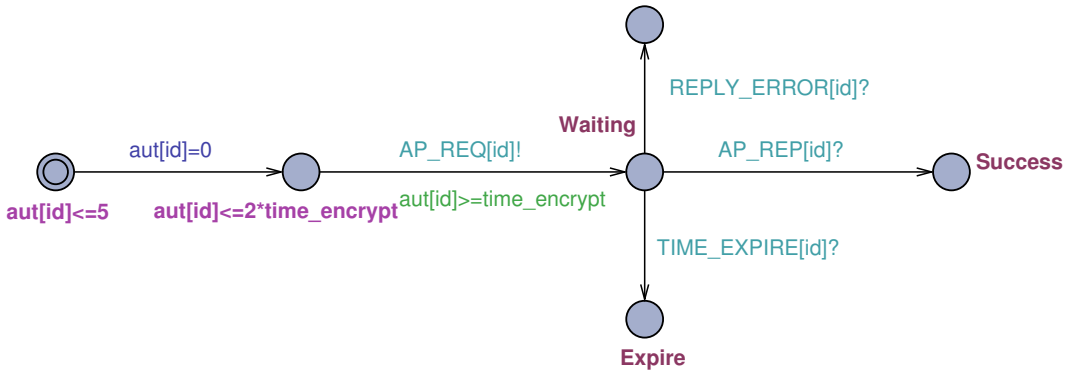



**Figura 3. Modelo Kerberos – Cliente Parametrizado**

gera um autômato, com um relógio aut[id], para cada cliente. O relógio modela a diferença entre o *timestamp* $T_b$ e o instante em que o servidor de aplicação, Figura 4, compara $T_b$ com o seu relógio físico. A sincronização entre o cliente e o servidor se dá através do evento AP_REQ[id]. Sempre que um cliente toma a transição rotulada com AP_REQ[id]!, o servidor toma uma transição rotulada com AP_REQ[id]?, o que marca o momento em que a mensagem chegou no servidor. Note que o autômato do servidor acumula mensagens em uma fila a qualquer momento. Ao processar uma mensagem AP_REQ[id], o servidor gasta duas vezes o tempo de decifragem: uma vez para decifrar o *ticket* TS e obter a chave $K_{cs}$; e outra vez, usando $K_{cs}$, para decifrar o autenticador e obter o *timestamp* $T_b$.

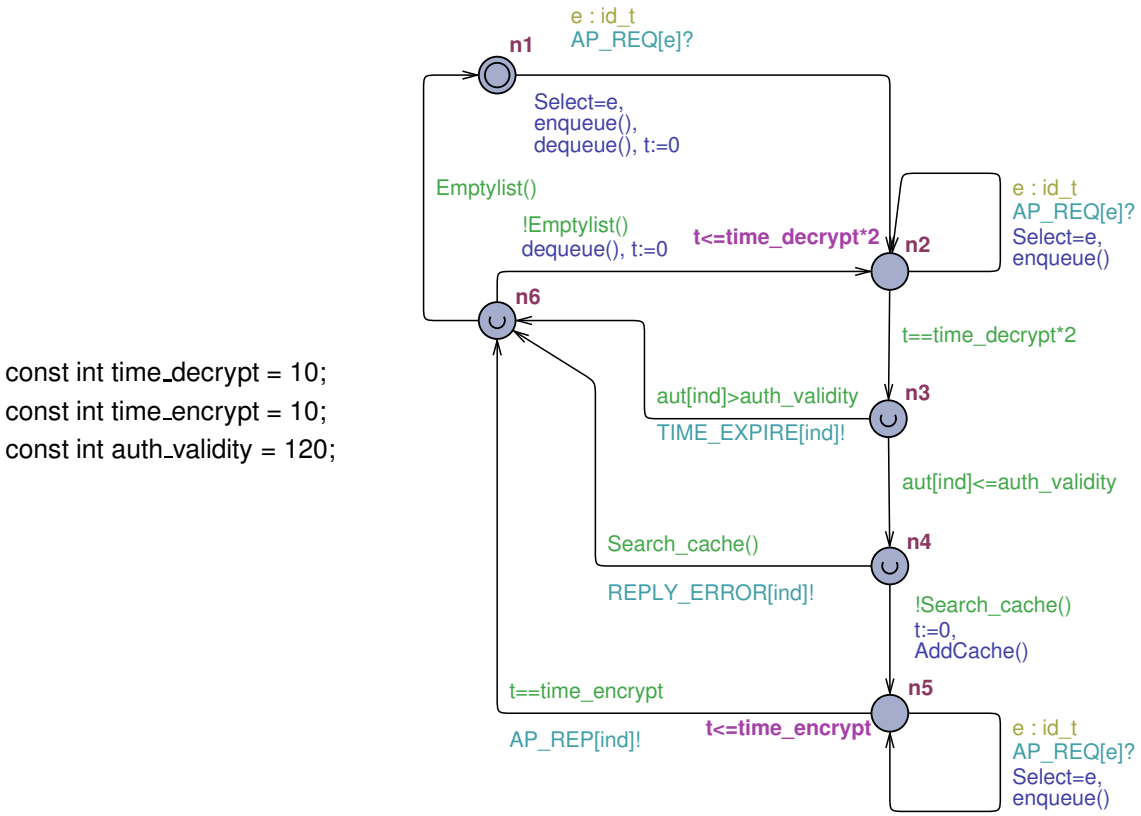



**Figura 4. Modelo Kerberos – Servidor de Aplicação**

 ©2012 SBC — Soc. Bras. de Computação

Como todos os clientes são iguais, a propriedade básica de alcançabilidade no UPPAAL pode ser verificada para um cliente fixo: `E<>Client1.Expire`. Nos nossos testes, o limite de relógios (e, portanto, de autômatos) para este modelo simplificado, foi de 8 relógios. A partir de 9 relógios, o consumo de memória ultrapassa 4Gb em questão de minutos e a execução é interrompida antes que a verificação seja concluída. Por essa razão, as constantes escolhidas para os tempos de cifragem e decifragem são apenas teóricas, bem maiores do que na experiência prática, pois com isso podemos realizar a verificação usando poucos clientes. Na Figura 2, o simulador do UPPAAL apresenta uma trajetória onde o Client1 atinge o estado Expire. Como esperado intuitivamente, mais ou menos clientes legítimos têm suas requisições rejeitadas, de acordo com a relação entre as constantes de cifragem, de validade dos autenticadores, e do acúmulo de mensagens.

### 4.2. Implementação proposta

Como visto na seção anterior, a implementação padrão do Kerberos inclui dentro do prazo de validade do autenticador qualquer atraso devido a custos de processamento. Dessa forma, os tempos de cifragem e decifragem e o acúmulo de mensagens podem fazer com que o servidor de aplicação rejeite muitas mensagens que seriam consideradas válidas caso não houvesse atraso no processamento. Essa rejeição pode ser corrigida com a se-

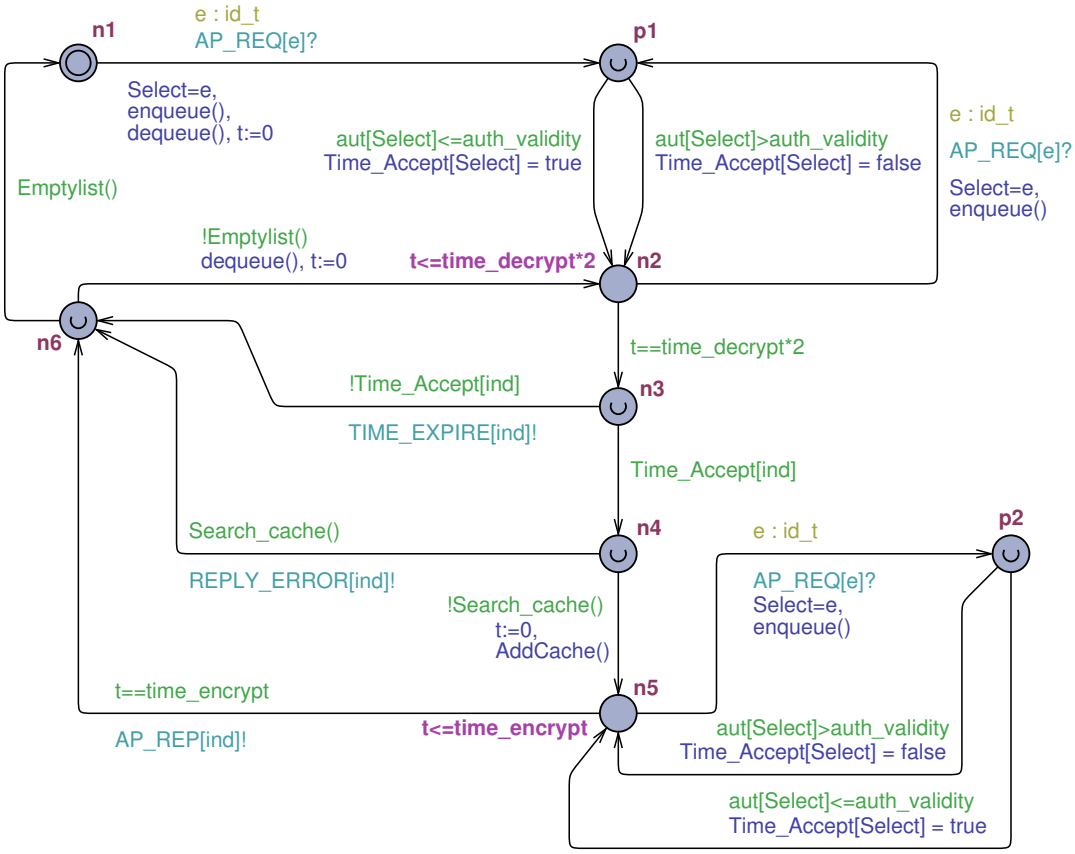



**Figura 5. Modelo Kerberos proposto – Servidor de Aplicação**

guinte proposta: toda mensagem, no instante em que chega no servidor de aplicação, é anotada com um *timestamp* $T_{b'}$. Ao sair da fila e ser processada pelo servidor, o *timestamp* $T_b$ é comparado com $T_{b'}$ e não com o tempo físico do servidor. A mensagem será

 ©2012 SBC — Soc. Bras. de Computação

considerada válida se $T_{b'} - T_b$ for menor do que o prazo de validade do autenticador (geralmente 2 minutos). O autômato da Figura 5 modela esta modificação e a verificação no UPPAAL mostra que, de fato, nenhuma mensagem válida é rejeitada.

É importante notar que essa proposta mantém a mesma proteção contra ataques do tipo *replay*. Uma mensagem recebida pelo servidor fora do prazo de validade do seu autenticador, será rejeitada normalmente. Por outro lado, a proposta aumenta a disponibilidade do serviço, pois o tempo de *timeout* dos clientes pode ser relaxado em uma situação de congestionamento do servidor, sem comprometer a segurança em relação a ataques do tipo *replay*, pois o servidor pode distinguir mensagens novas de antigas sem a dependência do acúmulo de mensagens e custos de decifragem.

## 5. Conclusão

Neste trabalho de Iniciação Científica, estudamos os aspectos de tempo real do protocolo de autenticação e distribuição de chaves Kerberos, através da modelagem e verificação do protocolo na ferramenta UPPAAL, que utiliza o formalismo de Autômatos Temporizados. Embora a ferramenta seja expressiva o suficiente, em teoria, para a modelagem; na prática o seu uso é limitado a sistemas relativamente pequenos, com aproximadamente 7 ou 8 autômatos e relógios, em razão do consumo exponencial de memória. Conseguimos realizar verificações completas sobre um modelo simplificado do protocolo, que permitiram a identificação de uma alteração no protocolo que o torna menos dependente dos custos de tempo devidos ao acúmulo de mensagens e ao custo das primitivas criptográficas.

Os resultados alcançados apontam para duas direções de trabalhos futuros. Por um lado, seria interessante implementar a alteração proposta na prática, para medir a disponibilidade de um servidor de aplicação usando o Kerberos, através de experimentos. Por outro lado, a abordagem de verificação formal com Autômatos Temporizados e a ferramenta UPPAAL poderia ser estendida para analisar outros protocolos de segurança que envolvem aspectos de tempo real.

## Referências

Alur, R. and Dill, D. (1994). A theory of timed automata. *Theoretical Computer Science*, 126:183–235.

Behrmann, G., David, A., and Larsen, K. G. (2004). A tutorial on uppaal. In Bernardo, M. and Corradini, F., editors, *SFM*, volume 3185 of *Lecture Notes in Computer Science*, pages 200–236. Springer. `www.uppaal.org`.

Behrmann, G., David, A., Larsen, K. G., Pettersson, P., and Yi, W. (2011). Developing UPPAAL over 15 years. *Software: Practice and Experience*, 41(2):133–142.

Bellovin, S. M. and Merritt, M. (1991). Limitations of the Kerberos authentication system. In *USENIX Conference Proceedings*, pages 253–267, Dallas, TX. USENIX.

Bengtsson, J. and Yi, W. (2004). Timed automata: Semantics, algorithms and tools. In *Lecture Notes on Concurrency and Petri Nets. W. Reisig and G. Rozenberg (eds.)*, number 3098 in LNCS, pages 87–124.

Blanchet, B. (2012). Security protocol verification: Symbolic and computational models. In Degano, P. and Guttman, J., editors, *First Conference on Principles of Security and*

 ©2012 SBC — Soc. Bras. de Computação

*Trust (POST'12)*, volume 7215 of *Lecture Notes on Computer Science*, pages 3–29. Springer Verlag.

Corin, R., Etalle, S., Hartel, P. H., and Mader, A. (2007). Timed analysis of security protocols. *Journal of Computer Security*, 15(6):619–645.

Denning, D. E. and Sacco, G. M. (1981). Timestamps in key distribution protocols. *Communications of the ACM*, 24(8).

Li, Y. and Pang, J. (2010). Extending the strand space method with timestamps: Part II application to kerberos V. *Journal of Information Security*, 1(2):56–67.

Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12).

Neuman, C., Yu, T., Hartman, S., and Raeburn, K. (2005). The kerberos network authentication service (v5). RFC 4120 (Proposed Standard). Updated by RFCs 4537, 5021, 5896, 6111, 6112, 6113.

 ©2012 SBC — Soc. Bras. de Computação

# Análise e implementação de um método para prover integridade a sistemas de banco de dados

**Gabriel Garcia Becker[2], Lucas Pandolfo Perin[2], Anderson Luiz Silvério[2], Marcelo Carlomagno Carlos[1], Ricardo Felipe Custódio[2]**

[1]Information Security Group
Royal Holloway University of London
TW20 0EX, Egham, United Kingdom

[2]Laboratório de Segurança em Computação
Universidade Federal de Santa Catarina
Florianópolis – SC – Brazil

marcelo.carlos.2009@rhul.ac.uk,
{gabrielbecker, lucasperin, anderson.luiz, custodio}@inf.ufsc.br

***Abstract.*** *Unauthorized changes of database content can result in significant losses for organizations and individuals and it is extremely important the assurance of privacy and integrity of databases. The Database Management Systems (DBMS) provides features to encrypt the database and protect it against unauthorized access. However, open-source DBMSs do not provide means to protect the database against unauthorized modifications and advanced DBMSs that provides such features are too expensive. In this paper, we analyse and implement a method to protect database systems against unauthorized modifications. Furthermore we tested the analysed method, showing its efficiency.*

***Resumo.*** *Modificações não autorizadas a sistemas de banco de dados podem causar prejuízos para pessoas e organizações, sendo de extrema importância a garantia de sigilo e integridade a tais sistemas. Geralmente as aplicações utilizam os recursos disponibilizados por Sistemas Gerenciadores de Banco de Dados (SGBDs) para assegurar o sigilo e a integridade dos dados armazenados no SGBD. Entretanto, os SGBDs que provêm os recursos necessários para garantir o sigilo e a integridade dos dados possuem um custo muito elevado, muitas vezes inviável para organizações de médio e pequeno porte. Este trabalho analisa e implementa um método de verificação de integridade de dados, baseado no uso de Hash-based Message Authentication Code (HMAC), independente de SGBD. Os testes realizados mostram a eficiência do método. Em média, a sobrecarga de processamento para o cálculo e verificação do HMAC para um registro do banco de dados não ultrapassa 100% do tempo de execução de determinada operação.*

## 1. Introdução

O uso de sistemas de bancos de dados tornou-se recorrente para os mais diversos tipos de aplicações. Com o uso extensivo da internet, as aplicações que utilizam sistemas de bancos de dados *online* são cada vez mais comuns. Estas aplicações normalmente armazenam dados sensíveis, como salários e outras informações pessoais [Kamel 2009]. Devido ao

 ©2012 SBC — Soc. Bras. de Computação

conteúdo potencialmente sigiloso, o acesso ou a modificação não autorizada a tais dados pode não ser desejado. Dessa forma, o sigilo e integridade de sistemas de banco de dados tem atraído pesquisadores das áreas de banco de dados e segurança.

Os sistemas de banco de dados, quando utilizados em ambientes compartilhados, possuem diversas ameaças de segurança provenientes de usuários não-autorizados, mau-uso e ameaças externas. Baseando-se nestas características, pode-se classificar as ameaças em quatro tipos principais:

1. leitura não autorizada de dados;
2. modificação não autorizada de dados;
3. remoção não autorizada de dados;
4. adição não autorizada de dados.

O problema da leitura não autorizada de dados (sigilo dos dados) já foi pesquisado extensivamente e normalmente é resolvido através da cifração do banco de dados [Ceselli et al. 2005, Samarati and di Vimercati 2010], em conjunto com métodos de indexação [Ceselli et al. 2005, Damiani et al. 2003], para agilizar as consultas. Entretanto, os problemas de modificações não autorizadas de dados (integridade dos dados) ainda necessitam de soluções eficientes [Samarati and di Vimercati 2010]. Os métodos existentes para tal problema geralmente requerem o desenvolvimento de um novo Sistema gerenciador de banco de dados (SGBD) ou alterações significantes nos SGBDs já existentes [Xie et al. 2007].

A seção 2 apresenta o método para tratar a integridade dos dados estudados neste trabalho e os testes de desempenho realizados são apresentados na seção 3. A seção 4 apresenta a implementação do método e, por fim, na seção 5 são apresentadas as considerações finais.

## 2. Integridade e sigilo de dados

Um dos problemas mais comuns quando se trata do sigilo em base de dados é a leitura não autorizada de informações sensíveis presentes nas tabelas. A proteção contra este tipo de acesso pode ser ser feita a nível de aplicação, o que é de fato realizada na maioria dos casos. Para tal, sugere-se a cifração de dados em colunas das tabelas. Deve-se selecionar colunas onde existem informações sigilosas nas tabelas e mante-las cifradas. A cifração destes dados pode ser feita diretamente pela aplicação ou pelo SGBD.

Para não afetar muito o desempenho das consultas, é comum o uso de métodos de indexação juntamente com a cifração dos dados [Ceselli et al. 2005, Damiani et al. 2003, Hacigümüs et al. 2004].

Para evitar a modificação não autorizada de registros contidos na base de dados, a aplicação deve restringir e definir as permissões de cada usuário. O SGDB também deve conter um mapeamento correto de usuários e seus respectivos privilégios a fim de impedir a execução de ações não autorizadas. Entretanto, existem perfis de usuários, como os administradores do servidor, administrador de banco de dados (DBA) e programadores, que podem acessar o SGDB sem o conhecimento da aplicação, podendo ocorrer situações onde, embora o usuário não seja autorizado a realizar determinada modificação, ele tenha permissões suficientes no sistema que permitem que ele o faça, mesmo que não intencionalmente. A cifração de colunas que contém dados sensíveis já reduz consideravelmente

 ©2012 SBC — Soc. Bras. de Computação

este problema, uma vez que um usuário mal-intencionado não conseguirá modificar o registro por não possuir a chave, e possivelmente sequer será capaz de identificar qual registro ele deve modificar para obter os efeitos que ele deseja. Embora as chances do atacante sejam reduzidas, ainda há a possibilidade de realizar modificação de registros a fim de corromper o sistema.

Infelizmente, os SGBDs mais comuns, como MySQL, PostreSQL, Firebird, etc, não possuem mecanismos para evitar tais problemas. Outros sistemas mais específicos possuem opções mais avançadas, porém também possuem custo bastante elevado. Para prover tal funcionalidade, foi estudada uma proposta que faz uso de Hash-based Message Authentication Code (HMAC) [Bellare et al. 1996, Turner and Chen 2011].

## 2.1. Adição do cálculo do HMAC nos registros da base de dados

Para se implementar este mecanismo, a aplicação será a única responsável pela manipulação dos registros da base de dados, e, para prover tal garantia, a aplicação terá uma chave simétrica conhecida somente por ela. Esta chave, por sua vez, será utilizada para gerar o HMAC dos registros.

Através da utilização do HMAC, é possível detectar qualquer modificação não autorizada realizada nos registros. Isto deve-se ao fato de que o atacante não tem conhecimento da chave necessária para gerar o HMAC, impossibilitando-o que consiga calcular um valor de HMAC válido. Este mecanismo também soluciona a questão de adição de registros pelo atacante, pois novamente, este necessitará da chave da aplicação para realizar o cálculo do HMAC de forma correta.

### Exemplo de implementação

Para exemplificar o funcionamento do método utilizando HMAC, considere uma tabela de pessoas, com as colunas id, nome, email e a coluna para armazenar o valor do HMAC. Os dados são apresentados na Tabela 1.

**Tabela 1. Tabela de exemplo de dados.**

| id | nome | email | hmac |
|---|---|---|---|
| 1 | Joao | joao@foo.com.br | abcd |
| 2 | Maria | maria@foo.com.br | qwer |
| 3 | Ana | ana@foo.com.br | kjhd |
| 4 | Roberto | roberto@foo.com.br | vpiu |

### Adicionando registros

Para adicionar um novo registro, a aplicação deve primeiramente calcular o valor do HMAC para o novo registro. Esse cálculo é feito através da concatenação dos valores de todas as colunas da tabela. Para adicionar uma nova pessoa com o nome “Jose” e email “jose@foo.com.br”, o cálculo do HMAC é apresentado na expressão (1).

$$HMAC(chave, Josejose@foo.com.br) = ohn4 \quad (1)$$

 ©2012 SBC — Soc. Bras. de Computação

Após o calculo do HMAC, a aplicação envia o comando de insert ao banco de dados, conforme ilustrado no trecho de código 1.

**Código Fonte 1. Código SQL para inserir um registro com HMAC**

```
INSERT INTO exemplo (nome, email, hmac)
VALUES ('Jose',
        'jose@foo.com.br',
        'ohn4');
```

### Modificando registros

A modificação de um registro é semelhante à adição. Por exemplo, para a alteração do registro número 3, da Tabela 1, primeiramente deve-se consultar o banco de dados para obter o registro, conforme ilustrado no trecho de código 2. O próximo passo é calcular o HMAC, apresentado na expressão (2). Por fim, atualiza-se o registro com o comando update ilustrado no trecho de código 3.

**Código Fonte 2. Código SQL para selecionar um registro**

```
SELECT nome, email FROM exemplo WHERE id='3';
```

$$HMAC(chave, Anaana.new@foo.com.br) = m3cx \tag{2}$$

**Código Fonte 3. Código SQL para atualizar um registro com HMAC**

```
UPDATE exemplo SET email='ana.new@foo.com.br', hmac='m3cx'
    WHERE id='3';
```

### Verificando a integridade de registros

Por fim, para verificação da integridade de um registro, deve ser feita uma consulta ao banco de dados pelo registro. Calcula-se o HMAC para o registro e compara o HMAC calculado com o HMAC obtido do banco de dados. A igualdade desses valores indica que o registro está íntegro.

## 2.2. Medidas de proteção contra remoção não autorizada

Através da utilização do HMAC, já é possível detectar qualquer modificação e adição não autorizada de registros. Entretanto, não é possível detectar a remoção não autorizada de registros.

Para o problema de remoção não autorizada de registros, sugere-se a criação de uma nova coluna para as tabelas do banco de dados, chamada de "Histórico cifrado". O histórico cifrado possui as seguintes características:

- permite relacionar dois ou mais registros de forma que possa se detectar a ausência de um deles, caso este seja removido;
- não permitir que uma terceira parte possa calcular o "histórico cifrado" sem conhecer as chaves de cifração;

 ©2012 SBC — Soc. Bras. de Computação

- utilização de operações de baixo custo computacional: criptografia simétrica e a operação lógica "ou exclusivo" (XOR);
- requer pouco espaço de armazenamento;
- não permite que sejam detectadas deleções dos $n$ últimos registros.

Para calcular o histórico cifrado de um registro $n$, obtém-se o HMAC desse registro e do registro anterior a ele. Em seguida é aplicada a função XOR nestes HMACs e o resultado é cifrado com uma chave simétrica. Esse cálculo é apresentado na expressão (3).

$$HistoricoCifrado(chave, HMAC_n, HMAC_{n-1}) = Cifrao(k, (HMAC_n \oplus HMAC_{n-1})) \quad (3)$$

### Removendo um registro

Para remover um registro $n$, deve-se excluir o registro e atualizar o histórico cifrado do registro $n + 1$. Para atualizar o histórico cifrado, utiliza-se o HMAC do registro $n + 1$ e $n - 1$.

### Verificando se um registro foi removido

Para verificar se um registro $t_n$, de uma tabela $T$ foi removido, a seguinte propriedade deve ser satisfeita:

$$\forall t_n, t_{n+1} \in T : t_n.HistoricoCifrado = HistoricoCifrado(chave, t_n, t_{n+1}) \quad (4)$$

### Remoção do ultimo registro

Esta proposta possui uma vulnerabilidade, o método apresentado não detecta quando os últimos registros são removidos indevidamente, uma vez que a verificação é feita com base no registo anterior. Uma possível solução para o problema consiste em adicionar ao final da tabela uma enupla com valores aleatórios conhecidos. Dessa forma, se o último registro for removido, poderá ser identificado, uma vez que os valores de controle não estarão mais presentes.

## 3. Análise da eficiência

Foi desenvolvido um protótipo utilizando a liguagem de programação PHP para verificar a eficácia do método proposto. Os testes foram executados no ambiente descrito na Tabela 2

Foram executados testes simulando um ambiente de uso real de um *software*, com o uso das operações *Select*, *Insert*, *Updade* e *Delete*. Executado também a verificação da integridade do banco de dados, através do HMAC e do histórico cifrado.

Após a execução dos testes, verificou-se que para as operações básicas de um banco de dados, o uso do HMAC não trouxe grandes alterações no desempenho. Entretanto o uso do Histórico Cifrado acarreta em uma grande sobrecarga na execução das

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 2. Descrição do ambiente de simulação**

| | |
|---|---|
| **Processador** | Intel Core 2 Duo 2.53Mhz |
| **Memória RAM** | 4GB |
| **Sistema Operacional** | Mac OS X 10.6.4 |
| **Linguagem** | PHP 5.3 |
| **SGDB** | MySQL 5.1 |
| **Algoritmo de Hash** | SHA-1 |
| **Tamanho de Chave HMAC** | 128 bits |
| **Algoritmo de cifração** | AES 128 bits |
| **Tamanho de chave simétrica** | 128 bits |

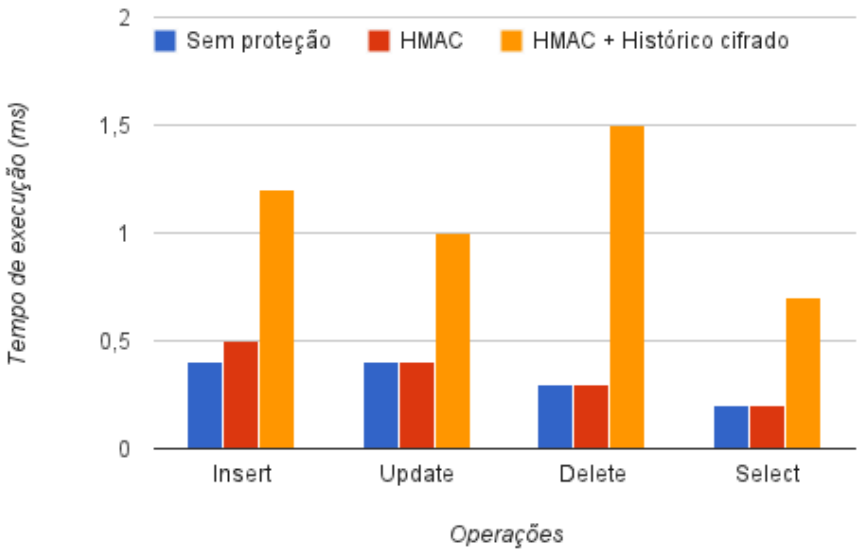



**Figura 1. Tempo de execução em segundos.**

operações. Nas operações de Delete e Select o impacto foi maior, pois é preciso atualizar e consultar outros registros. Os resultados completos podem ser vistos na Figura 1.

Verificou-se também o custo para realizar o cálculo de HMAC e de Histórico cifrado em tabelas já existentes. Para este cenário, foram realizados testes com diferentes volumes de dados, entre mil e dez milhões de registros. Neste cenário é visível a diferença no desempenho do cáclculo de HMAC e do Histórico cifrado, como mostrado na Figura 2. Entretanto, como em uma base de dados reais este cáclculo seria necessário apenas uma vez, considera-se satisfatório o tempo de execução de ambas as soluções.

Por fim, foi simulado o custo de verificar a integridade de diversos registros. Considera-se que este cenário em uma aplicação real seria executado em lote, em momentos em que o servidor de banco de dados está com uma sobrecarga menor. Assim como no cenário anterior, o Histórico cifrado mostrou um desempenho muito inferior ao do HMAC, conforme ilustrado na Figura 3.

Observa-se na Figura 3 que a operação de maior custo é do calculo do histórico

 ©2012 SBC — Soc. Bras. de Computação

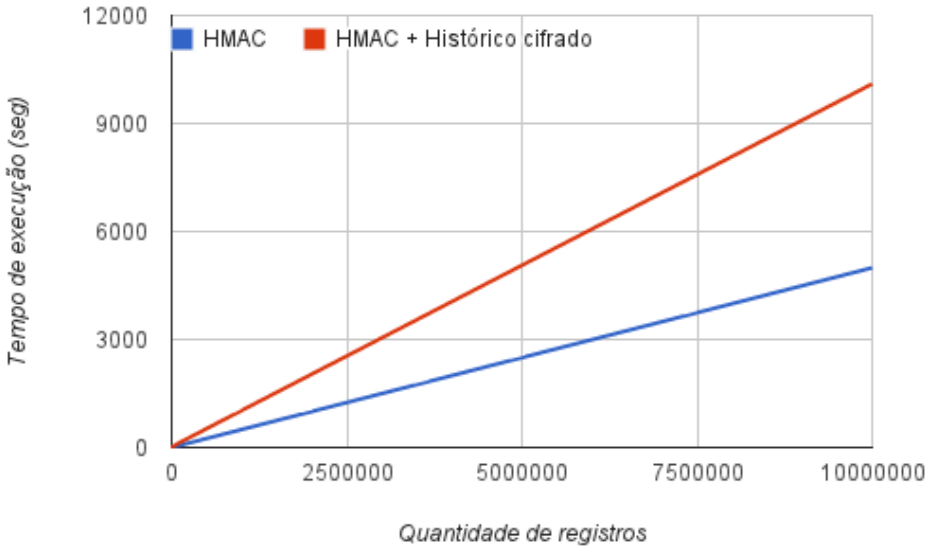



**Figura 2. Tempo de execução em segundos.**

cifrado. Assim, o desenvolvedor deve analisar se o mecanismo de detecção de remoção é relevante no escopo da aplicação.

Como os resultados obtidos nos testes foram satisfatórios, foi desenvolvido uma biblioteca mais completa em java para prover os serviços de cálculo de HMAC e do histórico cifrado. Para isso a biblioteca disponibiliza toda a estrutura necessária para essas funções, como a geração de chaves HMAC e de chaves simétricas.

## 4. Implementação

Para o cálculo do HMAC há uma classe responsável pela geração das chaves (HMacKeyGenerator), uma pelo cálculo do HMAC (HMacCalculator), além das classes que representam a chave para ser utilizada no cálculo do HMAC (HMACKey) e o HMAC em si (HMAC). A partir de uma chave HMAC é possível inicializar o HMACCalculator. Nele, é possível adicionar dados para o calculo do HMAC e finalizar a operação. O trecho de código 4 demonstra como é feito o cálculo do HMAC de uma mensagem qualquer.

**Código Fonte 4. Exemplo do cálculo do HMAC**

```
HMacKey key = HMacKey::generate("SHA1");
HMACCalculator hmacCalc = new HMACCalculator(key);
hmacCalc.update("mensagem");
HMAC hmac = hmacCalc.doFinal();
```

Para o histórico cifrado, tem-se a classe responsável pelo cálculo (EncryptedHistoryCalculator) e a classe que representa o histórico cifrado (EncryptedHistory). Para iniciar o cálculo do histórico, é necessário uma chave simétrica (SymmetricKey), que pode ser gerada de forma semelhante à chave utilizada pelo HMAC, pela classe SymmetricKeyGenerator. Para o cálculo do histórico, é necessário fornecer uma lista de HMACs

 ©2012 SBC — Soc. Bras. de Computação

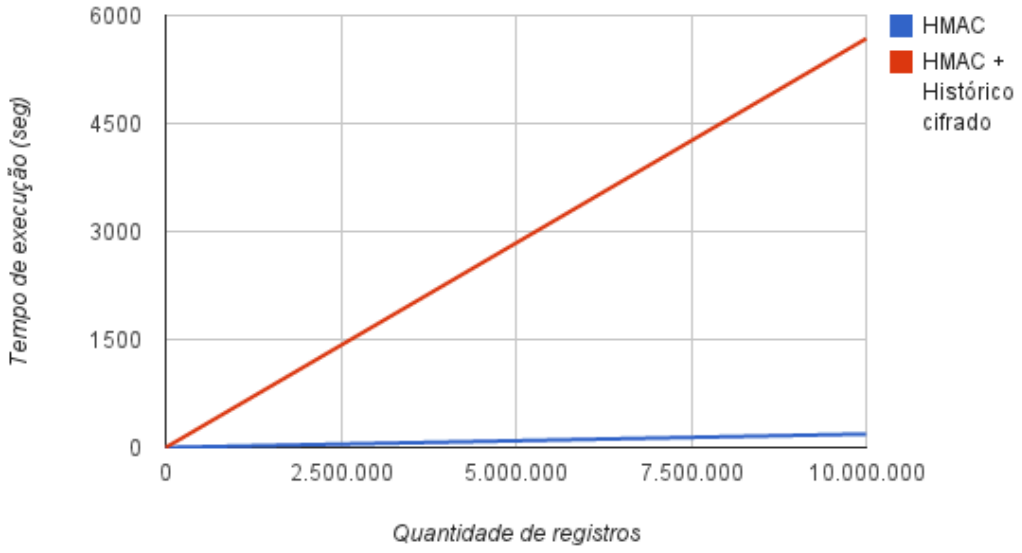



**Figura 3. Tempo de execução em segundos.**

e uma chave simétrica, coforme ilustrado no trecho de código 5.

**Código Fonte 5. Exemplo do cálculo do HMAC**

```
public EncryptedHistory generate(List<HMac> hMacs) throws
    SymmetricCipherException, EncryptionException {
        List<HMac> h = new ArrayList<HMac>();
        for(HMac hMac : hMacs) h.add(hMac.clone());
        byte[] xor = h.get(0).getBytes();
        for (int i = 1; i < h.size(); i ++) {
                byte[] hmac = h.get(i).getBytes();
                for (int j = 0; j < hmac.length; j++)
                        xor[j] = (byte) (xor[j] ^ hmac[j]);
        }
        byte[] result = cipher.doFinal(xor);
        return new EncryptedHistory(result);
}
```

A classe SecurityService possibilita efetuar diretamente o cálculo de HMAC, Histórico Cifrado e cifração, sendo preciso apenas usar os métodos correspondentes diretamente, realizando toda a operação em apenas uma função, por exemplo, para realizar o cálculo de um HMAC, devemos apenas chamar o método *getHmac(fields:List<objs>)* passando a lista de objetos que sera utilizada no cálculo. As chaves simétricas e assimétricas usadas são definidas pela clase SecurityServiceSpi. A classe SecurityServiceSpi deve ser implementada por um provider, com a finalidade de suportar os mais diversos dispositivos criptográficos através de uma mesma interface. Desta forma, pode-se adicionar diferentes dispositivos criptográficos simplesmente implementando uma interface, não sendo necessários realizar grandes alterações na aplicação. O diagrama de classes da Figura 4 apresenta esta abstração das classes SecurityService e SecurityServiceSpi.

 ©2012 SBC — Soc. Bras. de Computação

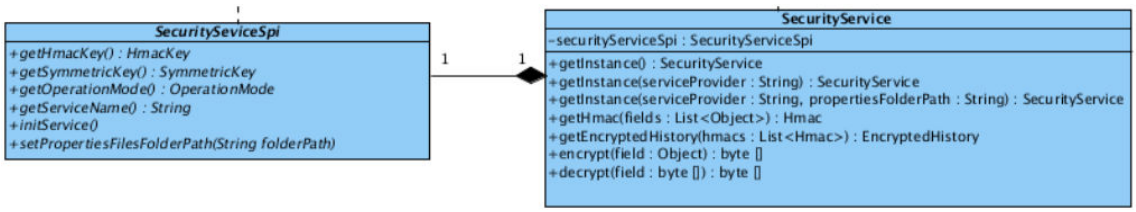



**Figura 4. Representação do *provider* da biblioteca**

## 5. Considerações finais

Este trabalho apresenta a análise e a implementação de um método para verificar a integridade de sistemas de banco de dados.

Os trabalhos que tratam da integridade de dados em Bancos de dados relacionais normalmente requerem o desenvolvimento de um novo SGBD ou alterações significativas nos SGBDs já existentes. Entretanto, o método apresentado é independente de SGBD, uma vez que todos os cálculos necessários são feitos pela aplicação.

Para verificar a eficácia do método foi implementado um protótipo em PHP. Foram considerados diferentes cenários para os testes do método, que apresentaram resultados satisfatórios. O primeiro cenário consistiu em testar a sobrecarga de operações SQL para a alteração de registros únicos. Nestes testes constatou-se que a adição do cálculo do HMAC é imperceptível na execução das operações INSERT, DELETE e UPDATE. No segundo cenário foi considerado o custo para a verificação da integridade de diferentes volumes de registros, enquanto que no terceiro cenário verificou-se o custo para a adição das colunas de HMAC e histórico cifrado em bases de dados já existentes. Ambos os testes executaram em um tempo satisfatório, considerando que são operações executadas em lote.

Constada a eficácia do método, implementou-se uma biblioteca em Java. Esta biblioteca fornece os serviços de cálculo de HMAC e histórico cifrado através de uma interface simples, abstraindo todas as operações necessárias para os cálculos das operações. Também foi criado uma interface que *providers* devem implementar para prover o suporte para chaves em dispositivos criptográficos.

## Referências

[Bellare et al. 1996] Bellare, M., Canetti, R., and Krawczyk, H. (1996). Keying hash functions for message authentication. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 1–15, London, UK, UK. Springer-Verlag.

[Ceselli et al. 2005] Ceselli, A., Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2005). Modeling and assessing inference exposure in encrypted databases. *ACM Transactions on Information and System Security (TISSEC)*, 8(1).

[Damiani et al. 2003] Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., and Samarati, P. (2003). Balancing confidentiality and efficiency in untrusted relational DBMSs. In *Proc. of the 10th ACM Conference on Computer and Communications Security*, Washington, DC, USA.

 ©2012 SBC — Soc. Bras. de Computação

[Hacigümüs et al. 2004] Hacigümüs, H., Iyer, B. R., and Mehrotra, S. (2004). Efficient execution of aggregation queries over encrypted relational databases. In *DASFAA*, pages 125–136.

[Kamel 2009] Kamel, I. (2009). A schema for protecting the integrity of databases. *Computers & Security*, 28(7):698–709.

[Samarati and di Vimercati 2010] Samarati, P. and di Vimercati, S. D. C. (2010). Data protection in outsourcing scenarios: issues and directions. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 1–14, New York, NY, USA. ACM.

[Turner and Chen 2011] Turner, S. and Chen, L. (2011). Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms. RFC 6151 (Informational).

[Xie et al. 2007] Xie, M., Wang, H., and Yin, J. (2007). Integrity auditing of outsourced data. *Very large data bases*, pages 782–793.

 ©2012 SBC — Soc. Bras. de Computação

# Um Protocolo Criptográfico para Controle de Medicamentos

**Bruno Imhof**[1], **Eduardo dos Santos**[1], **Ricardo Felipe Custódio**[1]

[1] Laboratório de Segurança em Computação (LabSEC)
Departamento de Informática e Estatística (INE)
Universidade Federal Santa Catarina (UFSC) – Florianópolis, SC – Brazil

{bimhof,esantos,custodio}@inf.ufsc.br

***Abstract.*** *The falsification of medicines is a worldwide problem, and to resolve the traceability of this problem, the Federal Government instituted the National Control of Medicines. The full deployment of this new logistic model, makes industry, commerce and government, need to adapt technology, generating demand for tools that are able to work concurrently aspects of information security. The contribution of this work is the formalization of a cryptographic protocol that aims to ensure integrity, authentication and confidentiality of information relating to tracking of medicines in Brazil.*

***Resumo.*** *A falsificação de medicamentos é um problema em todo o mundo, e para resolver a rastreabilidade deste problema, o Governo Federal instituiu o Sistema Nacional de Controle de Medicamentos. A implantação completa deste novo modelo logístico, faz com que indústria, comércio e governo, precisem se adequar tecnologicamente, gerando demanda de ferramentas que consigam trabalhar de forma concomitante os aspectos de segurança da informação. A contribuição deste trabalho é a formalização de um protocolo criptográfico que tem por objetivo garantir a integridade, autenticação e sigilo das informações referentes ao rastreamento dos medicamentos no Brasil.*

## 1. Introdução

O relatório anual da Organização das Nações Unidas (ONU) [Organização das Nações Unidas 2006], traz um alerta para o risco que os medicamentos falsificados podem trazer à saúde dos pacientes e as implicações que sua comercialização trazem à sociedade. Os danos à saúde se referem não apenas aos danos originários de consumo de medicamentos adulterados, mas também ao consumo sem orientação e/ou abusivo. Sem a correta orientação, certas substâncias podem tornar-se nocivas. Segundo dados da Organização Mundial da Saúde (OMS), este é um problema real e abrangente. Estima-se que, de 25% a 50% do total de remédios consumidos em países em desenvolvimento, sejam falsificados [Organização das Nações Unidas 2006].

Com o intuito de coibir o comércio de medicamentos falsificados, em 2009, o Governo Federal instituiu o Sistema Nacional de Controle de Medicamentos (SNCM). Esta nova lei estabelece que todo medicamento produzido e vendido no Brasil, deve ser rastreado. Este rastreamento deve ocorrer desde o momento da saída da fábrica e só deve ser encerrado com o consumo final pela população [Brasil 2009]. Para que este rastreamento seja possivel, os medicamentos passam a contar com um dispositivo especial, o Identificador Único do Medicamento (IUM)

 ©2012 SBC — Soc. Bras. de Computação

[ANVISA - Agência Nacional de Vigilância Sanitária 2009]. O IUM é um identificador baseado no padrão internacional do *DataMatrix* [ISO/IEC 2006], que consiste numa matriz bidimencional de blocos brancos e pretos, semelhante a um código de barras.

A implantação do novo sistema que faz o rastreamento de todos os medicamentos é feita em três etapas [Brasil 2009]. A primeira, faz o rastreamento do medicamento junto com seu lote; da fábrica, passando pelo distribuidor ou fornecedor até chegar a farmácia. Esta etapa já está em vigor desde 2011.

Por sua vez, a segunda etapa faz o rastreamento do comprador, medicamento e quantidade consumida. Fechando o ciclo, temos a terceira e última etapa, a qual consiste na correlação de médico, paciente e a correspondente prescrição médica utilizada. Esta terceira etapa é sem dúvida a mais complexa, pois há a necessidade de garantir a correta correspondência mesmo quando nem todos os produtos constados numa receita são de fato adquiridos.

Anterior ao rastreamento dos medicamentos, em 2007, com o intuito de monitorar a dispensação, das substâncias controladas no Brasil, a Agência Nacional de Vigilância Sanitária (ANVISA) normatizou o Sistema Nacional de Gerenciamento de Produtos Controlados (SNGPC) [ANVISA - Agência Nacional de Vigilância Sanitária 2007]. Entende-se por dispensação, o ato do farmacêutico fornecer um medicamento a um paciente a partir de uma prescrição médica. Com a instituição dessa normativa, ficou estabelecido que a dispensação das substâncias que são sujeitas a controle especial, precisam ser registradas em meio eletrônico.

Com a instituição do SNCM, o SNGPC foi integrado ao novo sistema, tornando necessário o entendimento do seu mecanismo de funcionamento. Com a integração destes sistemas, farmacêuticos precisam enviar semanalmente as informações referentes a entrada e saída de medicamentos controlados de seu estabelecimento. Atualmente, estas informações são armazenadas de maneira manuscrita num livro próprio, chamado de Livro de Registro Específico (semelhante aos livros de registros contábeis). As informações então são enviadas à agência reguladora através de um *web service* ou interface *web*. A Figura 1, extraída de [Agência Nacional de Vigilância Sanitária 2006], apresenta um exemplo do documento XML que é enviado ao SNGPC semanalmente.

Quando o farmacêutico utiliza a interface *web* para realizar a entrada das informações, dados como número do lote, dosagem, paciente destinatário, entre outros, estão sujeitos a erros de digitação. A segurança da origem do medicamento está intimamente ligada à precisão da transcrição dos dados de controle para o SNGPC. Os problemas com a ilegibilidade e a imprecisão das prescrições manuscritas, além de ocasionar danos à saúde do paciente [Noto et al. 2002], também prejudicam o monitoramento da dispensação pelo Estado [Mastroianni 2009].

O uso da certificação digital apresenta-se como um dos meios para a solução deste problema. A mesma já é utilizada por diversos profissionais da área da saúde. Em São Paulo, desde 2011, já se utilizam identidades digitais para emitir atestados eletrônicos para falta ao trabalho. Recentemente, com a instituição do CRM-Digital, os médicos passaram a contar com uma identidade exclusiva dentro do âmbito da ICP-Brasil [CFM - Conselho Federal de Medicina 2012], impulsionando ainda mais o uso de recursos eletrônicos.

 ©2012 SBC — Soc. Bras. de Computação

```xml
<saidaMedicamentoVendaAoConsumidor>
  <tipoReceituarioMedicamento>1</tipoReceituarioMedicamento>
  <numeroNotificacaoMedicamento>213213</numeroNotificacaoMedicamento>
  <dataPrescricaoMedicamento>2006-02-10</dataPrescricaoMedicamento>
  <prescritorMedicamento>
    <nomePrescritor>XFulano</nomePrescritor>
    <numeroRegistroProfissional>99999</numeroRegistroProfissional>
    <conselhoProfissional>CRM</conselhoProfissional>
    <UFConselho>DF</UFConselho>
  </prescritorMedicamento>
  <usoMedicamento>1</usoMedicamento>
  <compradorMedicamento>
    <nomeComprador>XCicrano</nomeComprador>
    <tipoDocumento>1</tipoDocumento>
    <numeroDocumento>123456 </numeroDocumento>
    <orgaoExpedidor>SSP</orgaoExpedidor>
    <UFEmissaoDocumento>DF</UFEmissaoDocumento>
  </compradorMedicamento>
  <medicamentoVenda>
    <registroMSMedicamento>1010000430215</registroMSMedicamento>
    <numeroLoteMedicamento>RGJ0400</numeroLoteMedicamento>
    <quantidadeMedicamento>3</quantidadeMedicamento>
  </medicamentoVenda>
  <dataVendaMedicamento>2006-02-11</dataVendaMedicamento>
</saidaMedicamentoVendaAoConsumidor>
```


**Figura 1. Relatório XML enviado ao SNGPC**

Este artigo propõe dois protocolos criptográficos que têm como intuito garantir a autenticidade, confidencialidade e integridade das informações referentes ao controle de medicamentos estabelecido pela nova legislação. Cada um dos protocolos visa contemplar as etapas não cobertas pelo nível de implantação atual do SNCM, ou seja, etapas 2 e 3.

O desenvolvimento deste trabalho esta focado em resolver as lacunas ainda pendentes ao processo de rastreamento dos medicamentos estabelecido, propondo um modelo computacional utilizando as identidades digitais da ICP-Brasil, Serviços WEB, e XAdES. O XAdES é uma extensão do XML-Dsig, que é um *framework* para documentos eletrônicos assinados digitalmente. Este tipo de extensão especifica perfis precisos de assinaturas que precisam de validade legal e jurídica [European Telecommunications Standards Institute 2009]. Um benefício importante do XAdES é que o documento assinado eletronicamente pode permanecer válido por longos períodos, mesmo para algoritmos criptográficos que forem posteriomente quebrados.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta a notação formal que será utilizada e descreve as informações que serão rastreadas para os fins deste protocolo. Na Seção 3 são propostos os dois protocolos, principal artefato produzido neste trabalho. A Seção 4 especifica a estrutura da aplicação utilizada na implementação dos protocolos. Por último, na Seção 5 são feitas as considerações finais, abordando projetos futuros a partir deste.

## 2. Método

Este estudo consiste em uma análise da legislação e da política nacional de controle de medicamentos [Brasil 2009], [Ministério da Saúde 1998]. A arquitetura proposta para este sistema é similar ao *design* projetado no trabalho de [Song et al. 2002]. Os mode-

 ©2012 SBC — Soc. Bras. de Computação

los computacionais para precrições e assinaturas eletrônicas são concebidos a partir de padrões já estabelecidos [Dutch 2011], [ISO/IEC 2011].

Através da análise das informações que precisam atender os requisitos de rastreamento do SNCM, dentro de um modelo formal, junto com a utilização de tecnologias e padrões já estabelecidos, será possível projetar os requisitos do sistema.

### 2.1. Notação Formal

De modo a melhor representar as trocas de mensagens da maneira mais precisa possível, adotamos uma notação matemática. A definição de seus componentes consta a seguir:

- **Entidades Participantes:** $P$, Paciente; $M$, Médico; $F$, Farmacêutico; e $S$ – Servidor.
- **Dados do Medicamento:** $Q$, Quantidade; $Med$, Medicamento - Inclui dosagem e forma farmacêutica; e $Tinfo$, Informações do tratamento medicamentoso (posologia, duração).
- **Dados da Prescrição:** $IdP$, Identificador da Prescrição;$IdD$, Identificaqdor da Dispensação; $V$, Validade da Prescrição; $Dmatrix$, Identificador único do medicamento - $\{$*Dmatrix*$\}$ representa um conjunto para dispensação com quantidade maior que 1; $Request$, Requisição de prescrição; e $D$, Data.
- **Controle de Sessão:** $Nonce$, Número aleatório usado para identificar a sessão.

A Figura 2 é um exemplo de como é feita a notação dos protocolos que são especificados posteriormente.

O número 1 indica o passo do protocolo em execução, neste caso, trata-se do primeiro passo. Em seguida, a primeira letra maiúscula representa a entidade que está enviando a mensagem. A seta direcional indica o sentido em que a mensagem está sendo enviada, neste caso, em direção à entidade receptora, a qual é representada pela segunda letra maiúscula.

O conteúdo propriamente dito da mensagem é representado entre chaves ({ e }) na parte mais a direita. Em alguns casos, operações criptográficas são feitas, por exemplo: $\{\ldots\}_{eKu_S}$ indica uma cifragem com a chave pública da entidade $S$, ao passo que, $\{\ldots\}_{sKr_M}$ indica uma assinatura digital feita com a chave privada da entidade $M$ [Ryan and Schneider 2001].

$$1. \quad M \quad \rightarrow \quad S \quad : \qquad \{\{mensagem\}_{sKr_M}\}_{eKu_S}$$

**Figura 2. Exemplo de notação formal.**

## 3. Proposta dos Protocolos

As descrições em alto-nível de nossos protocolos virão acompanhadas de suas respectivas notações formais. Começamos pelo protocolo Médico-Farmacêutico.

### 3.1. Protocolo Médico-Farmacêutico

Das três propostas de protocolo apresentadas neste trabalho, este pode ser considerado como o mais simples. O processo envolve a atuação de apenas duas partes e tem um número menor de passos.

As etapas necessárias para a execução deste protocolo são (Figura 3):

 ©2012 SBC — Soc. Bras. de Computação

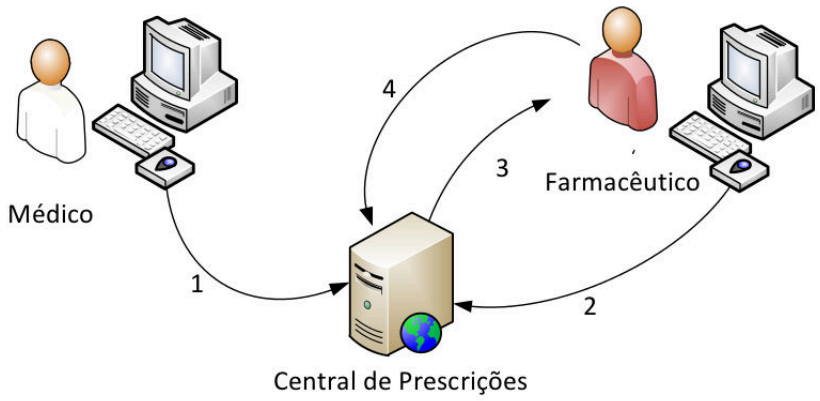



**Figura 3. Versão simplificada do protocolo.**

1. O médico prescreve um medicamento, assina e submete a nova entrada para a central.
2. O farmacêutico envia uma solicitação das prescrições pendentes do paciente, ele assina esta requisição para garantir sua identidade.
3. A central envia as precrições do paciente solicitadas, cifrando os dados com a chave pública do farmacêutico.
4. O farmacêutico, junto com o paciente presente no mesmo recinto, escolhem a prescrição desejada. O medicamento é fornecido ao paciente, e o farmacêutico envia uma dispensa desta medicação assinada por ele e envia a central.

A Figura 4 apresenta a notação formal da versão simplificada do protocolo.

| | | | | | |
|---|---|---|---|---|---|
| 1. | $M$ | $\rightarrow$ | $S$ | : | $\{\{IdP, P, Med, Q, V, M, D, Tinfo\}_{sKr_M}\}_{eKu_S}$ |
| 2. | $F$ | $\rightarrow$ | $S$ | : | $\{Request(P, Nonce)_{sKr_F}\}_{eKu_S}$ |
| 3. | $S$ | $\rightarrow$ | $F$ | : | $\{\{IdP, P, Med, Q, V, D, M, Tinfo\}_{sKr_M}, Nonce\}_{eKu_F}$ |
| 4. | $F$ | $\rightarrow$ | $S$ | : | $\{\{IdP, IdD, P, Med, Q, D, M, \{Dmatrix\}\}_{sKr_F}\}_{eKu_S}$ |

**Figura 4. Notação formal da proposta simplificada do protocolo.**

A ausência do paciente caracteriza este como um protocolo de transição. A verificação da identidade do paciente é responsabilidade do farmacêutico na hora em que a compra estiver acontecendo. Este procedimento é análogo ao processo atual, a diferença entre os dois, é que não seria armazenada a assinatura do paciente.

Soluções como o e-CPF e o Registro de Identidade Civil (RIC), que é uma proposta de identidade digital para concentrar todos os outros documentos de um cidadão [BRASIL 2012], ainda tem um longo caminho até serem largamente utilizados pela população em geral. Essa restrição temporária traz a necessidade de implantar o sistema em diferentes etapas, aumentando a importância de uma versão simplificada. A utilização das identidades digitais por médicos e farmacêuticos, é a primeira etapa para a informatização total do controle de medicamentos.

O estabelecimento dessas novas tencnologias de identidade, permite a incorporação do paciente ao processo, aumentando a segurança do controle. Sob esta

 ©2012 SBC — Soc. Bras. de Computação

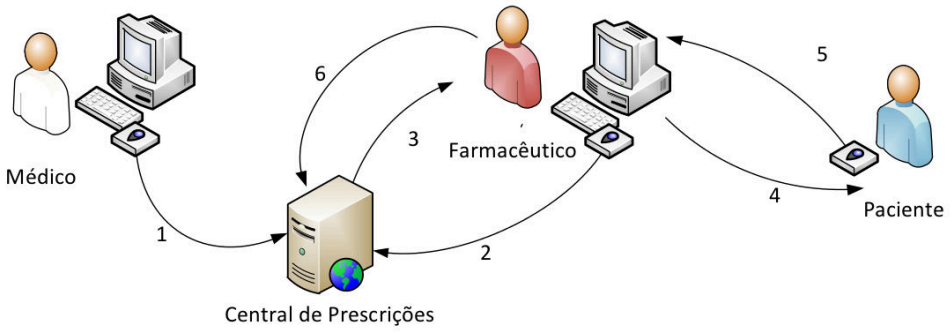



**Figura 5. Versão completa do protocolo.**

perperctiva, outro protocolo, um mais completo é proposto.

### 3.2. Protocolo Médico-Farmacêutico-Paciente

A participação do paciente no processo completa a cadeia de fabricação, comercialização e dispensação de medicamentos. Das três etapas de implantação do SNCM, este protocolo não contempla apenas a primeira, fabricante e fornecedor. Todas as outras informações que são descritas nas duas ultimas etapas estão contempladas nesta versão completa.

As etapas necessárias para a execução desta versão do protocolo são (Figura 5):

1. O médico prescreve um medicamento, assina e submete a entrada para a central.
2. O farmacêutico envia uma solicitação das prescrições pendentes do paciente, ele assina esta requisição para garantir sua identidade.
3. A central envia as precrições do paciente solicitadas, cifrando os dados com a chave pública do paciente.
4. O farmacêutico, encaminha as precrições para o paciente escolher qual delas é a desejada.
5. O paciente, no intuito de adquirir o medicamento, assina a baixa do mesmo, referindo a quantidade desejada, e envia ao farmacêutico.
6. O farmacêutico, também assina a baixa, que é o pedido de compra de um medicamento, gerada pelo paciente, e submete esta com as 2 assinaturas a central.

A Figura 6 apresenta a notação formal da versão completa do protocolo.

| | | | | | |
|---|---|---|---|---|---|
| 1. | $M$ | $\rightarrow$ | $S$ | : | $\{\{IdP, P, Med, Q, V, M, D, Tinfo\}_{sKr_M}\}_{eKu_S}$ |
| 2. | $F$ | $\rightarrow$ | $S$ | : | $\{Request(P.Nonce)_{sKr_F}\}_{eKu_S}$ |
| 3. | $S$ | $\rightarrow$ | $F$ | : | $\{\{\{IdP, P, Med, Q, V, D, M, Tinfo\}_{sKr_M}\}_{eKu_P}, Nonce\}_{eKu_F}$ |
| 4. | $F$ | $\rightarrow$ | $P$ | : | $\{\{IdP, P, Med, Q, V, D, M, Tinfo\}_{sKr_M}\}_{eKu_P}$ |
| 5. | $P$ | $\rightarrow$ | $F$ | : | $\{\{IdP, IdD, P, Med, Q, D, M\}_{sKr_P}\}_{eKu_F}$ |
| 6. | $F$ | $\rightarrow$ | $S$ | : | $\{\{\{IdP, IdD, P, Med, Q, D, M\}_{sKr_P}, \{Dmatrix\}\}_{sKr_F}\}_{eKu_S}$ |

**Figura 6. Notação formal da proposta do protocolo completa.**

A configuração de médicos,farmacêuticos e pacientes utilizam todos e-CPFs ou e-CFM, e-CPF e e-CFF respectivamente, atende as necessidades de segurança para execução desta tarefa.

 ©2012 SBC — Soc. Bras. de Computação

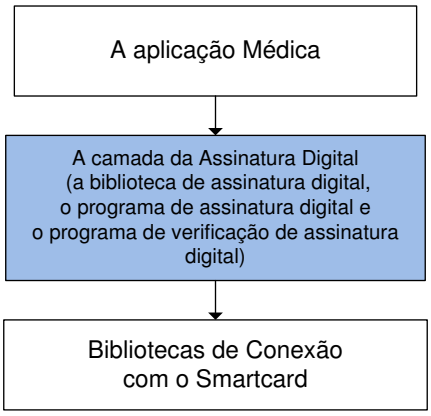



**Figura 7. Camadas de processamento da prescrição eletrônica.**

No intuito de tornar o sistema ainda mais seguro, pode ser exigidos mais do que um fator de autenticação. A seguir é proposta o mesmo protocolo, mas utilizando o suporte fornecido pelo RIC para verificar as impressões digitais.

## 4. Implementação

Com o intuito de provar a viabilidade da implementação, apenas a versão simplificada protocolo, descrita nas figuras 3 e 4, foi implementada neste estudo. A extensão deste para sua versão completa depende diretamente do estabelecimento de alguma tecnologia para identificação dos pacientes.

### 4.1. Signer

A norma [ISO/IEC 2011], trata sobre documentos médicos eletrônicos, especificando modelos de assinatura, e as aplicações que geram esses documentos, sendo referência para o desenvolvimento deste estudo.

O assinador de documentos que gera as prescrições eletrônicas, é uma camada intermediária entre a aplicação final e a plataforma criptográfica. A Figura 7 ([ISO/IEC 2011]), em tradução livre, representa o fluxo de processamento para realizar a assinatura digital. Através desta figura, é possível identificar que o programa Signer trabalha de maneira independente da aplicação médica, e da plataforma computacional.

Deste modo, prescrições e dispensações podem ser emitidas em uma estrutura que independe de aplicação(sistema médico) e a plataforma computacional.

A escolha da tecnologia Java Applet, é fundamental para que o assinador alcance a característica multiplataforma. O Java é uma linguagem de programação fortemente tipada, ela possui uma máquina virtual para interpretação dos comandos chamados Byte-CodeJava e possui implementações desta máquina virtual para diferentes plataformas computacionais como: Windows, Linux e Mac [Oracle 2012].

Além das funções disponibilizadas pelas interfaces de programação da linguagem, foram utilizadas outras duas bibliotecas para a construção deste assinador, a LibCryptoDev e Códigos-de-Referência.

A LibCryptoDev é um projeto interno do LabSEC que tem como finalidade abstrair as operações referentes a um chaveiro criptográfico. Um chaveiro crip-

 ©2012 SBC — Soc. Bras. de Computação

tográfico é uma ferramenta que gerencia uma ou mais chaves criptográficas, incluindo a chave privada, utilizada para a assinatura de documentos. Como existem diferentes implementações destes artefatos, a separação desta camada é necessária para permitir que a implementação do protocolo seja independente do sistema operacional da máquina.

A bibloteca Codigos-de-Referencia, é uma iniciatica do Instituto Nacional de Técnologia desenvolvida pelo LabSEC, para a disponobilização de uma referência de como devem ser feitas as assinaturas eletrônicas seguindo o Padrão Brasileiro de Assinatura Digital (PBAD) [Laboratório de Segurança em Computação 2012].

### 4.2. Central

A Central de Prescrições é um Serviço WEB, que recebe as prescrições, pedidos de consulta, e as dispensações geradas no processo.

A implementação deste serviço independe da plataforma computacional, uma vez que seja utilizado o Web Service Definition Language, como padrão das mensagens trocadas (WSDL) [W3C 2012]. A utilização do WSDL como formato intermediário para transporte da informação, garante a compatibilidade computacional desejada.

Para o desenvolvimento deste protótipo, foram escolhidas a tecnologia Java EE [Oracle 2012], utilizando um servidor de aplicação Tomcat [Apache Software Foundation 2012], um Banco de dados MySQL [MySQL 2012].

## 5. Considerações Finais

O protótipo desenvolvido utiliza os padrões brasileiros para assinatura digital em uma estrutura independente de aplicação, multiplataforma e livre.

As evidências geradas com este estudo, servem como roteiro para a gestão pública sobre possíveis caminhos para a implantação do SNCM.

A identificação unívoca de pessoas, é um dos requisitos na implementação de registros eletrônicos em saúde longitudinais, que são informações clínicas que acompanham um mesmo paciente durante sua trajetória por diferentes insituições de saúde. O desenvolvimento destes registros centrados no cidadão, permitem aumentar a qualidade do cuidado a saúde.

A análise do processo de controle de medicamentos resultou na notação formal de um protocolo e a prototipação deste sistema mostrou-se prática, alcançando os objetivos do trabalho.

A integridade, confidencialidade e o sigilo das informações referente ao controle e rastreamento dos medicamentos são resolvidas através da utilização dos recursos computacionais da criptografia.

O rastremanto das etapas 2 e 3 do SNCM, que ainda não está definido, é alcançado através da utilização das identidades digitais da ICP-Brasil.

### 5.1. Trabalhos Futuros

Os autores vislumbram algumas propostas de extensão deste trabalho. Dentre elas, destacam-se:

 ©2012 SBC — Soc. Bras. de Computação

1. Integração com SNCM - Emissão automatizada de relatórios ao SNGPC, disponibilização das informações do rastreamento.
2. Proposta Offline - Uma necessidade em todos os sistemas que dependem de conexão com a Internet, é um fluxo alternativo para quando a conexão esta indisponível. Para as prescrições, uma opção offline é desejada para trazer mais facilidades ao paciente. A materializaçao do documento através de códigos bidimensionais, armazenamento em cartões inteligentes pessoais ou temporários, juntamente mecanismos de sincronização com repositoórios assim que a conexão fosse restabelecida são propostas para um modelo offline.
3. Comprador não é o paciente - Estender a versão completa do protocolo permitindo a adição de um quarto ator. O comprador do medicamento precisa de uma autorização para poder adquirir os medicamentos, isso pode se dar através de um cadastro prévio, ou então utilizando outras técnologias como Certificados de Atributos.
4. Sistema de Prevenção de Erros e Interações Medicamentosas - As informações envolvendo os medicamentos consumidos, são pertinentes ao prontuário eletrônico do paciente. Estas poderiam ser fornecidas a um sistema terceiro para acompanhamentos longitudinais e análise para diagnósticos.
5. Canal de comunicação Farmacêutico-Médico - Em havendo necessidade, o farmacêutico deve entrar em contato com o profissional prescritor para esclarecer eventuais problemas que tenha detectado.
6. Lembretes para tomar remédio - Aplicativos que utilizam das informações sobre o tratamento medicamentoso para enviar mensagens aos pacientes lembrando o horário correto de tomar o medicamento.

## Referências

Agência Nacional de Vigilância Sanitária (2006). *Guia para geração do padrão de transmissão SNGPC*. Brasília.

ANVISA - Agência Nacional de Vigilância Sanitária (2007). Resolução rdc n 27, de 30 de março. Dispõe sobre o Sistema Nacional de Gerenciamento de Produtos Controlados - SNGPC, estabele implantação do módulo para drogarias e farmácias e da outras providências.

ANVISA - Agência Nacional de Vigilância Sanitária (2009). Resolução rdc n 59, de 24 de novembro. Dispõe sobre a implantação do Sistema Nacional de Controle de Medicamentos e definição dos mecanismos para rastreamento de medicamentos, por meio de tecnologia de captura, armazenamento e transmissão eletrônica de dados e dá outras providências.

Apache Software Foundation (2012). Welcome to the apache software foundation! `http://apache.org/`, September.

Brasil (2009). Lei n. 11903. Dispõe sobre o rastreamento da produção e do consumo de medicamentos por meio de tecnologia de captura, armazenamento e transmissão eletrônica de dados.

BRASIL (2012). Novo registro de identidade civil (RIC) -. August.

 ©2012 SBC — Soc. Bras. de Computação

CFM - Conselho Federal de Medicina (2012). Resolução n 1.983. Normatiza o CRM Digital para vigorar como cédula de identidade dos médicos inscritos nos Conselhos Regionais de Medicina.

Dutch (2011). Health informatics - messages - electronic exchange of messages on prescriptions and dispensations. Technical report, Standards Committee.

European Telecommunications Standards Institute (2009). *ETSI TS 101 903 XML Advanced Electronic Signatures (XAdES)*. France.

ISO/IEC (2006). 16022:data matrix bar code symbology specification. Technical report.

ISO/IEC (2011). Health informatics – public key infrastructure - -part 4: Digital signatures for healthcare documents. Technical report.

Laboratório de Segurança em Computação (2012). Laboratório de segurança em computação. `http://www.labsec.ufsc.br/`, September.

Mastroianni, P. (2009). Análise de prescrição de medicamentos. *Revista de Ciências Farmacêuticas Básica e Aplicada*, (30).

Ministério da Saúde (1998). Portaria n. 344. Dispõe sobre o Regulamento Técnico sobre substâncias e medicamentos sujeitos a controle especial.

MySQL (2012). MySQL :: The world's most popular open source database. `http://www.mysql.com/`, September.

Noto, A. R., Carlini, E. d. A., Mastroianni, P. C., Alves, V. C., Galduróz, J. C. F., Kuroiwa, W., Csizmar, J., Costa, A., Faria, M. d. A., Hidalgo, S. R., de Assis, D., and Nappo, S. A. (2002). Analysis of prescription and dispensation of psychotropic medications in two cities in the state of são paulo, brazil. *Rev Bras Psiquiatr*, 24(2):68–73.

Oracle (2012). Oracle technology network for java developers. `http://www.oracle.com/technetwork/java/index.html`, September.

Organização das Nações Unidas (2006). Perigo dos medicamentos falsos. Technical Report Relatório Anual 2006 Press Release No 2, Junta Internacional de Fiscalização de Entorpecentes - JIFE.

Ryan, P. and Schneider, S. (2001). *The Modelling and Analysis of Security Protocols*. Addison Wesley.

Song, W. J., Ahn, B. H., and Kim, W. H. (2002). Healthcare information systems using digital signature and synchronized smart cards via the internet. In *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pages 177–182.

W3C (2012). Web services description language (WSDL) version 2.0 part 1: Core language. `http://www.w3.org/TR/wsdl20/`, September.

 ©2012 SBC — Soc. Bras. de Computação

# Adaptações em um HSM para Homologação na ICP-Brasil

**Camila Maia Cardozo[1], Guilherme André Welter[1], Cláudio Dettoni Júnior[1]**
**André Bereza Júnior[1], Ricardo Felipe Custódio[1]**

[1]Departamento de Informatica e Estatística
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 88040-900 – Florianopolis – SC – Brasil

{camilamaia, guilherme.welter, dettonijr}@inf.ufsc.br

{andredex, custodio}@inf.ufsc.br

***Abstract.*** *The level of trust that the entities have on a PKI is related to the security of its private keys. The HSM is a device that manages and protects cryptographic keys, used to store private keys of a PKI. The Brazilian PKIs also use HSMs. To store private keys of the ICP-Brasil (a Brazilian PKI), the device needs to be evaluated by this PKI. This paper describes a set of changes that must be made in an HSM to evaluate it in the ICP-Brasil.*

***Resumo.*** *A confiabilidade de uma ICP depende da segurança de suas chaves privadas. O HSM é um dispositivo que gerencia e protege chaves criptográficas, próprio para guardar as chaves privadas de uma ICP. HSMs também são utilizados em ICPs brasileiras. Para que um HSM possa guardar chaves privadas da ICP-Brasil, o dispositivo deve ser homologado pela ICP-Brasil. Este artigo descreve um conjunto de adaptações necessárias para que um HSM possa ser homologado na ICP-Brasil.*

## 1. Introdução

Documentos eletrônicos são uma alternativa para reduzir a utilização de documentos em papel, reduzindo o espaço físico necessário para armazená-los. Documentos eletrônicos estão sujeitos a adulteração, portanto é necessário verificar sua autenticidade e integridade.

Para facilitar e proporcionar a verificação de autenticidade e integridade de documentos eletrônicos, infra-estruturas de chaves públicas (ICP) são utilizadas.

Uma ICP é uma cadeia hierárquica e de confiança, baseada em par de chaves criptográficas, uma chave pública e outra privada. Uma ICP tem como fim viabilizar a emissão de certificados digitais.

Autoridades certificadoras (AC) são componentes de uma ICP. Uma AC é responsável por emitir e assinar certificados digitais. O procedimento de assinar um certificado digital utiliza a chave privada da autoridade certificadora. Esta assinatura coloca os certificados na cadeia de confiança desta ICP.

Cada autoridade certificadora deve manter sua chave privada em segurança para que os certificados emitidos por ela sejam confiáveis. Para proteger a chave privada, utiliza-se um dispositivo especializado, chamado Módulo de Segurança Criptográfica ou HSM (*Hardware Security Module*).

 ©2012 SBC — Soc. Bras. de Computação

HSMs também são utilizados em ICPs brasileiras. A infra-estrutura de chaves públicas brasileira (ICP-Brasil) é uma ICP utilizada em larga escala no Brasil.

A Medida Provisória 2.200-2 [ITI 2001] de 24 de agosto de 2001 deu início à implantação do Sistema Nacional de Certificação Digital da ICP-Brasil. A ICP-Brasil possui normas para utilização de HSMs em suas ACs. Um HSM que guardará chaves privadas da ICP-Brasil deve ser um dispositivo homologado ICP-Brasil.

Para receber uma homologação ICP-Brasil um HSM deve atender a todos os requisitos da norma estabelecida para homologação de HSMs pela ICP-Brasil. Os trabalhos de [Bereza 2009] e [de Souza 2008] apresentam aspectos de desenvolvimento e adaptação de um HSM para atender aos requisitos da ICP-Brasil.

Segundo nosso conhecimento, o trabalho de Bereza e o trabalho de De Souza são os únicos que apresentam preocupação com o desenvolvimento de hardwares criptográficos que estão de acordo com a norma MCT 7.

A contribuição deste artigo é a descrição das alterações necessárias em um HSM genérico com algumas características que o impedem de receber a homologação, para que o dispositivo possa ser homologado na ICP-Brasil. Este trabalho discute os problemas existentes no HSM que impedem que ele seja homologado, apresenta alternativas de se corrigir estes problemas e como a solução deve ser implementada.

O artigo é estruturado em 5 seções. A seção 2 apresenta as características do HSM que será adaptado. A seção 3 aborda a homologação, suas etapas e níveis. Em seguida, na seção 4, as adaptações no HSM genérico são apresentadas, explicando o motivo de cada alteração. A seção 5 fecha o artigo apresentando as conclusões e sugestões de trabalhos futuros.

## 2. HSM

Um HSM é um dispositivo criptográfico capaz de gerenciar chaves criptográficas e executar algoritmos criptográficos em um ambiente seguro. Um HSM é o responsável pela gerência do ciclo de vida de chaves criptográficas, que inclui geração, uso e destruição destas chaves.

O HSM genérico que este artigo considera para a adaptação utiliza a biblioteca criptográfica OpenSSL FIPS versão 1.2 para executar suas funções.

As operações do OpenSSL FIPS que mais importam neste artigo são: a utilização de um subconjunto de algoritmos aprovados FIPS; execução de testes de resposta conhecida das funções criptográficas; execução de testes de consistência do par de chaves; geração de números aleatórios com algoritmo aprovado FIPS.

A biblioteca OpenSSL FIPS é utilizada pelo fato de possuir várias funções que atendem a requisitos das normas de homologação e padronização de HSMs. A homologação de HSMs existente no Brasil é a homologação na ICP-Brasil.

Para que um HSM possa gerenciar chaves criptográficas na ICP-Brasil é necessário que ele possua uma homologação na ICP-Brasil. A homologação é um processo que testa as proteções de um HSM para comprovar que ele pode gerenciar chaves com segurança.

 ©2012 SBC — Soc. Bras. de Computação

# 3. Homologação

A homologação de um equipamento é um processo solicitado pelo fabricante, o qual deseja comprovar que seu produto segue as normas e padrões de um determinado órgão. O órgão que estabelece as normas e padrões é chamado de órgão homologador (OH).

Além de estabelecer as normas, o OH é o responsável por averiguar se um equipamento está em conformidade com uma determinada norma. Inúmeros testes e análises são realizados no equipamento e na sua documentação. Estes procedimentos podem ser efetuados pelo próprio OH ou por laboratórios credenciados para realizar a análise.

A homologação de equipamentos é realizada em diversos setores, tais como o automotivo, semicondutores, comunicações e construção, porém o foco deste trabalho será na homologação de módulos de segurança criptográfica.

O objetivo de homologar um HSM, ou outros dispositivos tais como cartões inteligentes e tokens criptográficos, é padronizar estes equipamentos e garantir a operação segura do serviço criptográfico oferecido por eles. Estar de acordo com estes dois aspectos proporciona uma maior aceitação de tais equipamentos.

## 3.1. Normas de Homologação

Para ser homologado um HSM deve cumprir uma série de requisitos. Uma norma de homologação é o conjunto destes requisitos. A norma FIPS 140-2 é uma das mais seguidas no mundo e o MCT 7 é o padrão mais utilizado no Brasil.

### 3.1.1. FIPS 140-2

A norma FIPS 140-2 [Information Technology Laboratory 2001] é um documento que contém os requisitos para a homologação de um módulo de segurança criptográfica segundo os critérios do NIST (National Institute of Standards and Technology). A FIPS 140-2 é um dos padrões de mercado mais utilizados no mundo.

### 3.1.2. MCT 7

O Manual de Condutas Técnicas 7 (MCT 7) é o documento que contém os requisitos para a homologação de um HSM na ICP-Brasil. O órgão homologador da MCT 7 é o Instituto de Tecnologia da Informação (ITI), que é o responsável pela manutenção da ICP-Brasil.

O MCT 7 é um padrão baseado na FIPS 140-2, mas ressalta-se que muitos requisitos são diferentes. Assim, para uma boa compreensão da primeira, recomenda-se a leitura integral da segunda.

Assim como a FIPS 140-2 homologa dispositivos que podem ser utilizados no governo americano, o MCT 7 homologa dispositivos que podem ser utilizados no governo brasileiro.

Módulos de segurança criptográfica homologados na ICP-Brasil são compatíveis com os Padrões de Interoperabilidade de Governo Eletrônico (ePING).

 ©2012 SBC — Soc. Bras. de Computação

### 3.2. Etapas do Processo de Homologação

O processo de homologação é dividido em várias etapas. A figura 1 apresenta um fluxograma com as etapa do processo de homologação.

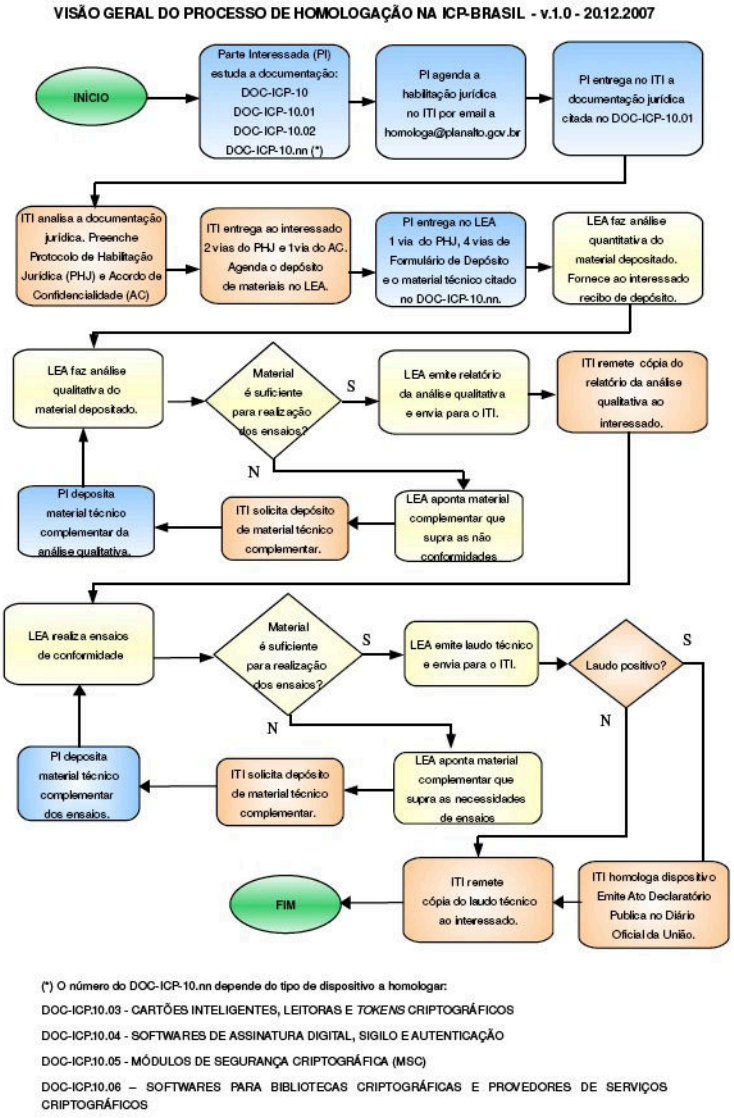



**Figure 1. Fluxograma do Processo de Homologação [ITI 2007]**

O processo inicia com o pedido de homologação de um HSM. O pedido parte do fabricante do HSM, que é a parte interessada (PI) na homologação.

O próximo passo é o depósito dos materiais necessários para a homologação. A PI deve enviar os materiais para o OH, que realizará a análise quantitativa dos materias. A análise quantitativa consistem em verificar se os materiais são suficientes para a realização dos ensaios de conformidade. Se os materiais não forem suficientes é possível depositar o material restante em um processo iterativa entre OH e PI, até que os materiais sejam suficientes.

Após a conclusão da análise quantitativa, com o depósito de todos os materiais, o

 ©2012 SBC — Soc. Bras. de Computação

OH inicia a análise qualitativa. A análise qualitativa consiste nos ensaios de conformidade para verificar se o HSM e documentação estão de acordo com a norma em questão.

Se durante a análise qualitativa o HSM não estiver em conformidade com algum requisito, é possível que o fabricante corrija a documentação e realize um novo depósito para que o HSM esteja em conformidade. Este processo de depósito de materiais e resultado da análise qualitativa é iterativo.

Ao fim do processo a PI recebe um laudo informando se o HSM está em conformidade com a norma ou não. O resultado negativo reflete que um ou mais requisitos da norma não foram cumpridos.

### 3.3. Níveis de homologação

Um HSM pode ser homologado em diferentes níveis de homologação sob o MCT 7. Estes níveis representam qual é o nível de proteções oferecidas pelo HSM homologado.

Existem dois tipos de níveis de homologação: nível de segurança de homologação (NSH) e nível de segurança física (NSF). O primeiro refere-se à segurança do software e o segundo à segurança do hardware.

A PI define em quais níveis deseja submeter seu equipamento antes de começar o processo de homologação.

Este artigo considera os níveis NSH 3 e NSF 2 do MCT 7 para homologar o HSM genérico, ambos níveis máximos. Quanto maior os NSH, mais material precisa ser depositado. Para homologação no NSH 3 é necessário que o código-fonte completo do HSM seja depositado.

Sobre os níveis de segurança física, para homologação no NSF 1 é necessário que o HSM suporte mecanismos de segurança física que evidenciam e resistam a violação. No NSF 2 é exigido o mesmo que no NSF 1, porém o HSM também deve suportar mecanismos de segurança física que detectam e respondem à violação.

## 4. Adaptações implementadas

Um HSM deve estar em conformidade com os requisitos de uma norma para ser homologado. Esta seção trata das adaptações que devem ser realizadas no HSM genérico, explicando em detalhes o porquê de cada alteração e como a solução deve ser implementada.

### 4.1. Uso de chaves ECDSA

A execução de operações que envolvem o algoritmo criptográfico ECDSA é uma demanda de mercado e de instituições da ICP-Brasil. Esta é a motivação para utilizar chaves ECDSA no HSM.

O ECDSA é um algoritmo criptográfico de assinatura digital baseado no DSA, porém utilizando criptografia de curvas elípticas.

Os algoritmos de criptografia por curvas elípticas tem algumas vantagens sobre os algoritmos tradicionais como o RSA. Por exemplo, é possível ter um mesmo nível de segurança com uma chave algumas vezes menor, tomando assim operações menores e mais rápidas, sendo portanto mais eficiente [Stebila 2002].

 ©2012 SBC — Soc. Bras. de Computação

#### 4.1.1. Remoção do módulo OpenSSL FIPS

O OpenSSL mantém uma versão homologada pela norma FIPS 140-2, chamada de OpenSSL FIPS (em oposição ao OpenSSL padrão). O OpenSSL FIPS versão 1.2 não suporta a execução do algoritmo ECDSA ou geração de chaves ECDSA.

O HSM tratado neste artigo utiliza o OpenSSL FIPS versão 1.2 para execução de suas funções criptográficas. O HSM precisa gerar chaves ECDSA, portanto o OpenSSL FIPS versão 1.2 não pode ser utilizado.

Existem duas alternativas para resolver este problema: a utilização do OpenSSL FIPS 2.0 (com suporte a ECDSA); a utilização do OpenSSL padrão. Do ponto de vista funcional, a utilização do OpenSSL FIPS 2.0 traz resultados imediatos, porém o HSM continuaria atrelado ao OpenSSL FIPS.

Manter um HSM atrelado ao OpenSSL FIPS traz problemas de longo prazo. O OpenSSL FIPS não possui tratamento adequado para a utilização indevida de algoritmos criptográficos, pois esta biblioteca finaliza a aplicação se detectar a utilização de um algoritmo não aprovado FIPS.

A biblioteca OpenSSL padrão dá maior flexibilidade para o HSM, sendo possível existir um modo de operação apenas com algoritmos criptográficos aprovados FIPS e outro modo com qualquer algoritmo. O processo de troca de versão da biblioteca força a adaptação de software do dispositivo.

Portanto, a solução escolhida para a adaptação é a utilização do OpenSSL padrão, pois assim haverá a flexibilidade para utilização de diferentes algoritmos no HSM e erros na aplicação poderão ser tratados adequadamente.

A remoção do OpenSSL FIPS resulta em vários problemas relacionados a outras funções do HSM. Estes problemas são descritos nas seções posteriores.

#### 4.1.2. Troca da biblioteca criptográfica

O OpenSSL FIPS é um módulo que é instalado em conjunto com alguma versão do OpenSSL. É possível possuir duas instalações do OpenSSL em um computador, uma versão é o OpenSSL padrão e a outra é o OpenSSL com o módulo FIPS.

A utilização do OpenSSL FIPS em uma aplicação, consiste em executar os métodos desta biblioteca. Além disso é necessário utilizar o ligador do OpenSSL FIPS na ligação dos objetos da sua aplicação. A utilização do ligador do OpenSSL FIPS direciona os métodos que seriam executados pelo OpenSSL padrão, para os métodos do OpenSSL FIPS.

No processo de compilação do OpenSSL, o OpenSSL FIPS deve ser removido. Todas as referências ao OpenSSL FIPS devem ser direcionadas ao OpenSSL padrão.

### 4.2. Testes de funções criptográficas

Como visto na seção 4.1, a solução para a adição do algoritmo ECDSA é a troca da versão do OpenSSL utilizada no HSM, passando do OpenSSL FIPS para o OpenSSL padrão.

 ©2012 SBC — Soc. Bras. de Computação

O OpenSSL padrão não contém testes do tipo resposta conhecida e auto-testes de consistência de par de chaves, previstos no MCT 7. Assim, estas funcionalidades devem ser acrescentadas ao HSM.

### 4.2.1. Testes de resposta conhecida

A seção 3.9 do MCT 7 requisita que o módulo criptográfico teste seus algoritmos criptográficos com testes do tipo resposta conhecida. O teste de resposta conhecida consiste em testar uma função conhecendo a sua saída para uma entrada conhecida.

É possível verificar o funcionamento correto de uma função HASH com um teste de resposta conhecida. Seja f uma função HASH. Seja $x$ uma entrada para qual sabemos que sua saída deve ser $y$ quando aplicada $f$. O algoritmo do teste de função HASH $f$ é o seguinte:

$$f(x) = e$$

$$se\ e = y \Rightarrow resultado\ positivo$$

$$se\ e \neq y \Rightarrow resultado\ negativo$$

Resultado negativo implica que há erro na função HASH $f$ testada. Resultado positivo implica que não há erros na função HASH $f$ testada.

Desta forma, os testes de resposta conhecida devem ser implementados para os algoritmos criptográficos utilizados no HSM, a citar os algoritmos:

- RSA
- ECDSA
- SHA-1
- AES
- 3DES
- HMAC
- RNG

### 4.2.2. Auto-testes de consistência de par de chaves

Os auto-testes condicionais de consistência de par de chaves também devem ser acrescentados ao HSM, assim como os testes de resposta conhecida.

Os testes de consistência de par de chaves são executados sempre que um par de chaves é gerado e tem a finalidade de verificar se a geração de chaves ocorreu sem erros, como no exemplo a seguir.

Seja $P$ um texto claro, $E$ o algoritmo de criptografia, $D$ o algoritmo de decifragem, $C$ o texto cifrado, $U$ a chave pública e $R$ a chave privada. O algoritmo do auto-teste de consistência de par de chaves é o seguinte:

$$Eu(P) = C$$

$$se\ P = C, par\ de\ chaves\ falhou\ no\ teste.$$

Se o par de chaves ainda não falhou no teste,

 ©2012 SBC — Soc. Bras. de Computação

$Dr(C) = X$

$Se\ X \neq P, par\ de\ chaves\ falhou\ no\ teste.$

Se o par de chaves não falhou no teste, o par de chaves foi gerado de maneira correta. Caso contrário, houve erros na geração do par de chaves.

Para estar em conformidade com a norma, os auto-testes de consistência de par de chaves devem ser implementados no HSM, sempre sendo executados quando um par de chaves RSA ou ECDSA é gerado. O mesmo procedimento é executado para a importação de chaves, para verificar a consistência do par.

### 4.3. Gerador de números aleatórios aprovado FIPS

A seção 3.7.1 do MCT 7 requer que o gerador de números aleatórios (RNG, da sigla em inglês) utilizado na geração de chaves seja um algoritmo determinístico aprovado FIPS.

Existem dois tipos de geradores de números aleatórios, o determinístico e o não-determinístico.

Geradores de números aleatórios não-determinísticos geram números a partir de fenômenos físicos. Só é possível capturar fenômenos físicos através de hardware, portanto geradores não-determinísticos só podem ser implementados em hardware.

Geradores de números aleatórios determinísticos utilizam uma semente para gerar os dados aleatórios. A semente de um RNG é um número inicial utilizado no algoritmo de geração. RNGs determinísticos são implementados em software.

Ainda na seção 3.7.1 do MCT 7, é informado que algoritmos RNG não aprovados FIPS podem apenas ser utilizados para gerar sementes para os algoritmos RNG aprovados FIPS.

Com a troca da versão do OpenSSL utilizado no HSM, visto na seção 4.1, é necessário também alterar o algoritmo RNG utilizado. O OpenSSL FIPS já possui um algoritmo RNG determinístico aprovado FIPS, porém o OpenSSL padrão não.

Desta forma, é preciso implementar um novo algoritmo RNG determinístico aprovado FIPS no HSM.

#### 4.3.1. Engine OpenSSL RNG

O OpenSSL dá suporte ao carregamento de bibliotecas dinâmicas para sobrescrever suas funções padrão. Estas bibliotecas seguem uma interface disponibilizada pelo OpenSSL. Uma biblioteca deste tipo é chamada Engine OpenSSL.

O algoritmo implementado no HSM pode ser implementado em uma engine OpenSSL. Esta engine deve ser carregada no OpenSSL do dispositivo, sobrescrevendo os métodos de geração de números aleatórios padrão do OpenSSL.

O algoritmo RNG implementado no HSM deve seguir as recomendações do NIST para a implementação de geradores de números aleatórios determinísticos.

A recomendação que está no item 5 do anexo C da norma FIPS 140-2 descreve como deve ser implementado um algoritmo RNG baseado em ANSI X9.31 com AES 128

 ©2012 SBC — Soc. Bras. de Computação

bits [Information Technology Laboratory 2012]. Esta recomendação pode ser implementada utilizando apenas o algoritmo criptográfico AES.

## 5. Conclusão

A homologação de HSMs dá credibilidade aos dispositivos e é uma forma de padronizar os HSMs e prover interoperabilidade entre os sistemas que os utilizam. Este artigo apresenta o impacto da modificação de uma biblioteca em um HSM que será submetido para homologação na ICP-Brasil.

O impacto pode ser grande a curto prazo, que é o momento da homologação, mas pode ser moderado ou pequeno se o tempo de vida do HSM for considerado.

Um HSM é um produto que geralmente é comercializado e está sujeito a atualizações de software, algoritmos e necessidades dos clientes. O esforço de manutenção pode ser reduzido com implementações simples e redução de utilização de bibliotecas instáveis.

Este artigo apresenta um HSM em que a remoção de uma biblioteca produz um impacto imediato grande, mas reduz o esforço de manutenção a longo prazo.

### 5.1. Trabalhos Futuros

Alguns aspectos do MCT 7 devem ser revisados para aumentar a confiabilidade dos HSMs homologados. A utilização de mecanismos de autenticação externos é um exemplo. Quando um dispositivo externo ao HSM é utilizado para a autenticação de usuários no HSM, não é necessário que este dispositivo seja homologado. Desta forma o dispositivo utilizado para a autenticação pode ser falho, sendo possível cloná-lo ou seu conteúdo sensível pode ser extraído. O MCT 7 deve tratar questões envolvendo dispositivos externos de forma mais criteriosa.

Outro aspecto é a auditoria em normas de homologação. A auditoria não é tratada, ou é tratada superficialmente nas normas. É necessário adicionar requisitos nas normas de homologação para demonstrar a importância da auditoria em dispositivos criptográficos.

## Referências Bibliográficas

Bereza, A. (2009). Aprimoramento de um hsm para homologacao na icp-brasil. Trabalho de Conclusão do Curso de Ciências da Computação da Universidade Federal de Santa Catarina.

de Souza, T. C. S. (2008). Aspectos técnicos e teóricos da gestão do ciclo de vida de chaves criptográficas no openhsm. Master's thesis, Universidade Federal de Santa Catarina.

Estrutura de Chaves Públicas Brasileira (2007a). *Manual de Condutas Técnicas 7 - volume I*, 1 edition.

Estrutura de Chaves Públicas Brasileira (2007b). *Manual de Condutas Técnicas 7 - volume II*, 1 edition.

Information Technology Laboratory (2001). *FIPS 140-2 - Security Requirements for Cryptographyc Modules*.

 ©2012 SBC — Soc. Bras. de Computação

Information Technology Laboratory (2012). *FIPS 140-2 - Security Requirements for Cryptographyc Modules - Annex C: Approved Random Number Generators for FIPS PUB 140-2*.

Infra-Estrutura de Chaves Públicas Brasileira (2007). Glossário ICP-Brasil. http://www.iti.gov.br/twiki/pub/Certificacao/Legislacao/Glossario_ICP-Brasil_-_Versao_1.2.pdf.

ITI (2001). Medida provisória 2.200-2. http://www.iti.gov.br/twiki/bin/view/Certificacao/MedidaProvisoria.

ITI (2007). Histórico homologação. http://www.iti.gov.br/twiki/bin/view/Homologacao/WebHome.

OpenSSL Team (2007). OpenSSL FIPS 140-2 User Guide. www.openssl.org/docs/fips/UserGuide-1.1.1.pdf.

Stebila, D. (2002). Performance analysis of elliptic curve cryptography for ssl. *Measurement*, pages 87–94.

 ©2012 SBC — Soc. Bras. de Computação

# Detecção de Phishing em Páginas Web Utilizando Técnicas de Aprendizagem de Máquina

**Francisco F. R. Cunha[1], Eulanda M. dos Santos[1], Eduardo J. P. Souto[1]**

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

fagner.cunha99@gmail.com, {emsantos,esouto}@icomp.ufam.edu.br

***Abstract.*** *This research aimed to develop mechanisms for detecting phishing web pages. Phishing is a form of electronic fraud in which the criminal posing as a legitimate entity, tries to convince his victims to provide confidential information such as usernames and passwords. We used three machine learning techniques to perform the classification process: SVM, KNN and Logistic Regression. During the research process were identified twelve characteristics that highlight the difference between phishing and legitimate pages. The SVM classifier achieved the highest accuracy rate, of 95.60%.*

***Resumo.*** *Este trabalho de pesquisa teve como objetivo desenvolver mecanismos para a detecção de phishing em páginas web. O phishing é uma forma de fraude eletrônica em que o criminoso, passando-se por uma entidade legítima, tenta convencer suas vítimas a fornecerem informações confidenciais como nomes de usuários e senhas. Foram utilizadas três técnicas de aprendizagem de máquina para realizar o processo de classificação: SVM, KNN e Regressão Logística. Durante o processo de pesquisa foram identificadas doze características que evidenciam a diferença entre páginas phishing e páginas legítimas. O classificador SVM obteve a maior taxa de acertos com 95,60%.*

## 1. Introdução

A Internet trouxe grandes facilidades para a vida das pessoas. Hoje, através do comércio eletrônico é possível comprar qualquer coisa sem sair de casa, desde bombons de chocolate e livros até carros e aviões. Outra facilidade é a possibilidade de fazer transações bancárias pelos serviços de *Internet banking*, eliminando, na maioria dos casos, a necessidade de ir ao banco. Contudo, para realizar tais operações é necessário que o usuário forneça dados sensíveis que o identifiquem tais como números de cartão de crédito, *login* e senha. Isso torna a Internet cada vez mais atrativa para a ação de criminosos que tentam roubar essas informações.

O *phishing* consiste numa forma de fraude eletrônica realizada por uma pessoa mal intencionada (ou *phisher*) que, passando-se por uma entidade verídica, tenta convencer suas vítimas a fornecerem informações confidenciais como *login*, senhas, números de cartão de crédito, conta bancária, entre outros dados pessoais. Para alcançar seu objetivo, o *phisher* envia mensagens eletrônicas, normalmente emails, alegando ser uma entidade legítima como instituições financeiras. O conteúdo dessas mensagens

 ©2012 SBC — Soc. Bras. de Computação

normalmente induz o usuário a clicar em um *link* que leva a uma página com um formulário que fará a coleta dos dados.

Esse tipo de fraude tem se tornado muito comum e cada vez mais usuários descuidados têm seus dados roubados. A empresa de consultoria Gartner Inc. (2009) estima que só em 2008 esses criminosos causaram um prejuízo de mais de 1,7 bilhão de dólares nos Estados Unidos. Já em um artigo publicado no jornal on-line China Daily [Changxin 2011], a empresa de segurança Beijing Rising Information Technology Co. afirma que páginas *phishing* roubaram cerca de 3 bilhões de dólares na China em 2010. No Brasil, nenhuma estatística sobre os prejuízos causados por esse tipo de ataque é divulgada.

Assim, o principal objetivo deste trabalho é desenvolver mecanismos que identifiquem atividades de *phishing* em páginas *web* por meio de técnicas de aprendizagem de máquina. Como objetivos específicos: montar uma base de *sites phishing*; identificar e extrair características que distinguem *sites phishing* de *sites* legítimos; realizar experimentos com diversas técnicas de aprendizagem de máquina; analisar os resultados das técnicas empregadas.

O restante deste trabalho está organizado em quatro seções. A seção 2 apresenta o conceito de phishing e algumas abordagens utilizadas para detecção desse tipo de fraude. Na seção 3 é apresentada a metodologia seguida neste trabalho. A seção 4 descreve o procedimento experimental adotado e apresenta os resultados obtidos. Por último, são apresentadas as conclusões e os trabalhos futuros.

## 2. Revisão Bibliográfica

### 2.1. Conceito de *Phishing*

Segundo o Anti-Phishing Working Group (2010), o *phishing* é uma prática fraudulenta em que os criminosos (conhecidos como *phishers*) podem utilizar tanto engenharia social quanto artifícios técnicos para roubar informações confidenciais das vítimas. Em ataques usando engenharia social, o *phisher* envia mensagens eletrônicas, normalmente emails, alegando ser uma entidade legítima como instituições financeiras. O conteúdo dessas mensagens normalmente induz o destinatário a acessar uma página *web* construída especificamente para enganar e convencer o usuário a fornecer dados como nomes de usuários e senhas. Já em ataques com artifícios técnicos, o criminoso instala no computador da vítima *softwares* maliciosos, como *trojans* e *key logger spywares*, que visam roubar os dados sigilosos.

De acordo com Sheng (2009), a prevenção e a detecção de *phishing* baseado em engenharia social podem ser feitas em dois níveis de abordagens: filtragem de emails e de páginas *web*. Para o nível de correio eletrônico utilizam-se os tradicionais filtros de *spam*, mas como o email *phishing* é feito para parecer legítimo, normalmente ele consegue passar por esses sistemas. Com isso, foram desenvolvidos alguns filtros específicos para detectar emails *phishing* [Fette 2007; Abu-Nimeh 2007]. O segundo nível se concentra na filtragem das páginas *web* quando acessadas em um navegador de Internet, sendo este o foco deste trabalho.

 ©2012 SBC — Soc. Bras. de Computação

## 2.2. Abordagens para a Detecção de Phishing em Páginas *Web*

Existem basicamente duas grandes categorias de abordagens para a detecção de *phishing* em páginas *web*: *blacklists* e heurísticas *anti-phishing* [Sheng 2009].

O método usando *blacklists* consiste em manter uma lista de URLs de *sites* já identificados como *phishing*. Diversas entidades fornecem listas desse tipo. Por exemplo, o serviço PhishTank (2011) disponibiliza uma lista de *sites phishing* verificados manualmente pela comunidade de usuários, o que a torna bastante confiável. Já o Google publica uma *blacklist* que é automaticamente atualizada como descreve Whittaker (2010), sendo esta usada pelos navegadores Mozilla Firefox, Google Chrome e Apple Safari. A empresa Netcraft (2010) também fornece uma lista a seus clientes, além de disponibilizar uma barra de ferramentas para navegadores.

O uso de heurísticas procura verificar se a página possui características de um ataque *phishing*. Uma solução bastante conhecida é a ferramenta SpoofGuard (2011) que analisa sintomas como URLs obscuras e sentenças suspeitas. Zhang (2007) propõe um método usando TF-IDF, popular algoritmo de recuperação de informação que mede o quão importante uma palavra é para um documento numa determinada coleção de documentos, em conjunto com outras seis características da página. Rosiello (2007) propõe uma técnica que analisa a similaridade do *layout* entre duas páginas utilizando *tags* html e propriedades DOM. A Microsoft, por exemplo, implementa em seu navegador Internet Explorer 7 uma solução que faz uso tanto de *blacklist* quanto análise do conteúdo da páginas para detectar sintomas de *phishing* [Ludl 2007].

Ambos os métodos têm vantagens e desvantagens. *Blacklists* têm um alto nível de precisão, no entanto não conseguem detectar novos ataques *phishing* imediatamente, devendo ser atualizada para isso. Ainda, essas listas normalmente requerem intervenção e verificação humana, o que pode acarretar um grande consumo de recursos. O emprego de heurísticas, por outro lado, consegue detectar *phishing* assim que são disseminados, mas podem produzir falsos positivos, ou seja, páginas legítimas incorretamente classificadas como *phishing*. Este trabalho se baseou na aplicação de técnicas de aprendizagem de máquina em heurísticas para a detecção de *phishing*.

## 2.3. Técnicas de Aprendizagem de Máquina na Detecção de Phishing

Os algoritmos de aprendizagem de máquina utilizados para classificação (como foi o foco deste trabalho) normalmente são chamados de algoritmos de aprendizagem supervisionados. Neste caso, o algoritmo ou classificador tenta aprender a predizer a classe correta a qual determinada amostra pertence de acordo com as características fornecidas [Abu-Nimeh 2007]. Assim, trata-se de uma "aprendizagem por exemplo", onde a partir de um treinamento com amostras rotuladas, o algoritmo consegue aprender a classificar corretamente uma nova amostra não rotulada.

É comum na literatura o uso de aprendizagem de máquina para o problema de detecção de *phishing*: classificadores são aplicados nas heurísticas, criando os modelos de classificação capazes de rotular novas páginas. Fette (2006) propõe um sistema para a detecção de *phishing* em emails utilizando o classificador SVM (*support vector machine*). Abu-Nimeh (2007) faz uma comparação entre diversos classificadores para o problema de detecção de emails *phishing*. Whittaker (2010) descreve um sistema que usa o classificador regressão logística para manter atualizada a *blacklist* de páginas

 ©2012 SBC — Soc. Bras. de Computação

*phishing* fornecida pelo Google. Muitas das características utilizadas por esses três autores são compartilhadas neste trabalho, porém, diferente de Whittaker que utiliza apenas regressão logística, foram também realizados experimentos com os classificadores SVM e KNN (*k-nearest neighbor*).

## 3. Metodologia Utilizada

A metodologia adotada neste trabalho está dividida em três etapas bem definidas: *download* da página, extração das características e classificação. A figura 1 ilustra a organização desta metodologia.

**Figura 1. Etapas da metodologia adotada**

### 3.1. *Download* da Página

A análise das páginas pode ser feita tanto em tempo real (*online*) ou a partir de arquivos salvos (*offline*). No entanto, como páginas *phishing* ficam *online* por um curto período de tempo, em média 72 horas, o *download* é necessário. Isto permite que análises posteriores sejam feitas na página, quando esta já não estiver disponível na Internet.

### 3.2. Extração das Características

A partir dos trabalhos relacionados e da análise das páginas na base de dados, foram selecionadas doze características que evidenciam a diferença entre páginas *phishing* e páginas legítimas. Essas características podem ser classificadas de acordo com a fonte de dados de onde podem ser extraídas. Este trabalho se concentrou em três fontes: características baseadas na URL, características extraídas de bases de dados *online*, características baseadas no conteúdo da página.

#### 3.2.1. Características Extraídas da URL

Construir URLs que pareçam legítimas é umas das formas que os *phishers* utilizam para tentar convencer suas vítimas de que a página é legítima. Desta forma, é possível identificar, em diversas situações, um *site phishing* apenas olhando para sua URL. Assim, foram extraídas seis características:

a) URL baseada em IP: Algumas páginas *phishing* são hospedadas em máquinas que não tem registro no DNS, a saída para o *phisher* é referenciar pelo IP; isso também previne que o *site* seja tirado do ar pelo desativamento do domínio. Além disso, entidades legítimas raramente usam IPs em suas URLs.

b) Quantidade de pontos na URL: Uma das formas de obscurecer a URL e tentar fazer com que os usuários acreditem que seja verdadeira é usar subdomínios como em http://www.bank.com.br.badsite.com/ ou utilizar o domínio alvo de *phishing* no caminho como em http://badsite.com/www.bank.com.br/, em ambos os casos, há uma grande quantidade de pontos.

 ©2012 SBC — Soc. Bras. de Computação

c) Tamanho da URL: Através da análise da base, foi verificado que as URLs de páginas *phishing* costumam ter uma quantidade muito maior de caracteres do que as páginas legítimas, numa tentativa de desviar a atenção do usuário.

d) Palavras-chave na URL: Existem determinadas *strings* que são comuns em URLs de *phishing*, usadas na tentativa de passar segurança ao usuário: *account, update, confirm, verify, secur, notif, log, click, inconveninen, ebay, paypal*.

e) Caracteres suspeitos: A ferramenta SpoofGuard identifica dois caracteres encontrados na URL como suspeitos: o "@" e o "-". Destes, foi utilizado apenas o arroba, sendo mais perigoso pois quando presente no domínio, tudo o que vem antes dele é considerado nome de usuário e repassado ao endereço que vem depois desde símbolo. Assim, ao acessar http://www.bank.com.br@badsite.com/, o usuário pode pensar que está acessando o *site* http://www.bank.com.br, mas na realidade está navegando em http://badsite.com com o nome de usuário "www.bank.com.br".

f) Domínio de topo (TLD): Um nome de domínio é composto por uma série de nomes separados por pontos. O último desses nomes é o chamado domínio de topo que pode ser organizados em dois grupos: TLD com duas letras que representam países (como ".br" que representa o Brasil) e TLD com mais de duas letras usados para propósitos genéricos (como ".gov" que é de uso restrito para entidades do governo). Domínios podem ser registrados em alguns desses TLD sem restrições em quanto outros devem seguir alguns pré-requisitos, assim determinados TLD podem oferecer maior facilidade de serem usados por *phishers*.

### 3.2.2. Características Extraídas a Partir de Informações em Bases de Dados *Online*

Informações sobre *sites* são armazenadas por diversas companhias em bases de dados com finalidades específicas como prover informações de rastreamento e indexação. Essas informações por si só não podem comprovar que uma página é *phishing*, mas podem reforçar evidências encontradas em outras características. Duas características desse tipo foram extraídas:

a) Geolocalização da página: A hospedagem de páginas *phishing* pode se concentrar em determinadas regiões do planeta. Essa característica foi implementada a partir da base de dados de fornecida pela empresa MaxMind (2011), onde blocos de IPs estão relacionados aos países.

b) Google PageRank™: o *pagerank* é um valor numérico que representa a importância de uma página num conjunto de páginas *web*. Quanto maior o *pagerank* de uma página, mais importante ela é. Mas como páginas *phishing* tem um curto período de vida, seu *pagerank* é muito baixo ou não existe.

### 3.2.3. Características Extraídas a Partir do Conteúdo da Página

Mesmo que a URL pareça legítima, pode-se determinar se uma página é *phishing* analisando o conteúdo da página. Para isso, foram extraídas quatro características a partir do HTML da página:

a) Presença de formulário: É através de um formulário com campos para entradas de texto que o *phisher* consegue obter os dados pessoais das vítimas. Caso a *tag* <INPUT> esteja presente no HTML da página e for do tipo "*textfield*" ou "*password*" a página é

 ©2012 SBC — Soc. Bras. de Computação

sinalizada. Páginas legítimas que tem essa característica são facilmente distinguíveis de *phishing* através das outras características.

b) Razão de *links* para outros domínios: Normalmente os *links* em uma página apontam para o mesmo domínio. Em páginas *phishing*, na tentativa de ficar o mais parecido possível com a página real, os *links* normalmente apontam para o domínio real. Nesta característica é calculada a razão $R_1$ conforme a equação:

$$R_1 = L_o/L$$

Onde $L_o$ é a quantidade de *links* para outros domínios e L é a quantidade total de *links* na página.

c) Razão de objetos carregados a partir de outros domínios: Uma página *web* é composta de diversos objetos incluindo imagens, css, *iframes*, *scripts*, etc. Em uma página comum, a grande maioria desses objetos é carregada a partir do próprio domínio. Em páginas *phishing* é comum esses objetos serem carregados a partir do *site* real. A razão $R_2$ entre a quantidade de objetos carregados a partir de outros domínios é calculada pela equação:

$$R_2 = O_o/O$$

Onde $O_o$ é a quantidade de objetos para outros domínios e O é a quantidade total de objetos na página.

d) Palavras-chave no título do *site*: Assim como nas URLs, algumas *strings* são comuns no título de páginas *phishing*. Foi utilizado o mesmo conjunto de *strings* aplicado à extração dessa característica na URL para a extração no título da página.

### 3.3. Classificação

Como citado na Seção 2.3, um classificador depois de treinado consegue predizer a qual classe uma amostra não rotulada pertence a partir da leitura do seu vetor de características. O uso desses algoritmos torna a detecção de *phishing* muito mais eficiente, onde ao invés de ser criar e atualizar manualmente as regras de filtragem de dados, eles o fazem automaticamente. Neste trabalho foram utilizados os classificadores SVM, KNN e regressão logística.

## 4. Resultados e Discussões

Esta seção apresenta alguns dos resultados obtidos a partir da metodologia adotada. Inicialmente, é feita uma descrição da base de dados utilizada. Em seguida, é feita a apresentação do procedimento experimental. Depois, são feitas algumas definições específicas das métricas de desempenho utilizadas. Finalmente são apresentados os resultados e uma comparação entre os classificadores utilizados.

### 4.1. Base de Dados

Como não existe uma base de dados de *phishing* normalizada, ou seja, rotulada e com as características extraídas, foi necessário preparar uma base de cunho próprio. Esta base contém 12.912 páginas, sendo que 6.456 amostras de *phishing* e 6.456 amostras de páginas legítimas. As amostras de *phishing* foram coletadas a partir do repositório PhishTank (2011) entre os dias 10 de maio e 15 de maio de 2011. Já as amostras de

 ©2012 SBC — Soc. Bras. de Computação

páginas legítimas foram sorteadas da lista composta por mais de 4 milhões de páginas fornecida pelo Open Directory Project (2011). Cada amostra da base é correspondente a uma única URL. No entanto, durante um ataque de *phishing*, podem ser utilizadas várias URLs para uma mesma página. Situações desse tipo estão representadas na base.

### 4.2. Procedimento Experimental

Os experimentos foram realizados no ambiente WEKA (2011) utilizando a estratégia de validação cruzada com dez partes. A base de páginas é dividida em dez partes iguais e mutuamente exclusivas entre si. Cada parte é testada e as outras nove partes são usadas como base de treino do classificador. Este processo é realizado para cada uma das dez partes e ao final, é calculada a média da classificação obtida nas dez interações.

Inicialmente, foram realizados experimentos utilizando todas as doze características. Foram utilizados os classificadores KNN, SVM e regressão logística. O classificador KNN é um dos mais simples de todos: uma amostra é classificada de acordo com o voto majoritário de seus k vizinhos mais próximos. Assim, durante os testes, foi ajustado experimentalmente o parâmetro K de forma a obter o melhor resultado (alcançado para k igual a 5). Já o classificador SVM procura traçar um hiperplano que separa as duas classes, maximizando a margem entre os pontos mais próximos. Assim como no KNN, os parâmetros do SVM foram ajustados experimentalmente a fim de obter melhores resultados, que, dentre os valores testados, foram de 100 para o parâmetro C e 0.05 para o gama, sendo este último específico do *kernel* RBF utilizado. O parâmetro C permite uma certa flexibilidade na separação das classes, controlando o equilíbrio entre permitir certas erros no treinamento e forçar margens mais rígidas. Já o classificador regressão logística foi treinado utilizando a configuração padrão disponível no ambiente WEKA (2011).

Em seguida, cada característica foi testada individualmente com o classificador SVM para avaliar o impacto na detecção de *phishing*. As características que apresentaram individualmente uma taxa de falso positivo menor que 10% foram selecionadas para serem novamente testadas em conjunto. O objetivo desses experimentos foi verificar se haveria melhoria na taxa de falsos positivos.

### 4.3. Métricas de Desempenho

Quatro métricas foram utilizadas para avaliar o desempenho da classificação:

a) Precisão: definida como a razão entre quantidade de páginas *phishing* corretamente classificadas e a quantidade total de páginas classificadas como *phishing*.

b) Taxa de verdadeiros positivos: corresponde à razão entre a quantidade de páginas *phishing* corretamente classificadas e quantidade total de páginas *phishing*.

c) Taxa de falsos positivos: calculada pela razão entre o número de páginas legítimas classificadas como *phishing* e a quantidade total de páginas legítimas.

d) Taxa de acertos: é a razão entre a quantidade de páginas corretamente classificadas e o número total de páginas.

 ©2012 SBC — Soc. Bras. de Computação

### 4.4. Resultados

A Tabela 1 apresenta os resultados obtidos com os classificadores utilizando todas as características. O classificador KNN obteve uma melhor precisão com 94,7%, ou seja, ele oferece uma confiança maior que os outros classificadores ao classificar uma amostra como *phishing*. Já o classificador SVM foi o que apresentou melhor taxa de acertos com 95,6%, obtendo uma precisão um pouco menor que o KNN.

**Tabela 1. Resultados usando todas as características**

| Classificador | Precisão | Taxa de verdadeiros positivos | Taxa de falsos positivos | Taxa de acertos |
|---|---|---|---|---|
| **SVM** | 94,20% | 97,10% | 5,90% | 95,60% |
| **KNN** | 94,70% | 93,20% | 5,20% | 93,97% |
| **Regressão Logística** | 93,70% | 97,00% | 6,50% | 95,22% |

Cada característica foi testada individualmente, os resultados são apresentados na Tabela 2. URL baseada em IP foi a característica que obteve a maior precisão com 99,3%, havendo somente um caso de falso positivo. Já o *pagerank* foi a característica que obteve a melhor taxa de acertos individualmente, chegando a 90,07%. A ocorrência de palavras-chave tanto na URL quanto no título da página obtiveram uma boa precisão, conseguindo, respectivamente, 97,2% e 93,1%. Um destaque negativo foi a característica "URL tem caractere suspeito" onde o classificador praticamente "não aprendeu" a qual classe cada amostra pertencia.

**Tabela 2. Desempenho individual das características**

| Característica | Precisão | Taxa de verdadeiros positivos | Taxa de falsos positivos | Taxa de acertos |
|---|---|---|---|---|
| URL baseada em Ip | 99,30% | 2,30% | 0,00% | 51,13% |
| Quantidade de pontos na URL | 86,50% | 26,40% | 4,10% | 61,15% |
| Quantidade de caracteres na URL | 79,50% | 46,90% | 12,10% | 67,41% |
| URL tem caractere suspeito | 50,00% | 70,00% | 70,00% | 49,99% |
| Domínio de top | 78,40% | 50,80% | 14,00% | 68,42% |
| Palavras-chave na URL | 97,20% | 28,30% | 0,80% | 63,75% |
| Pagerank | 93,30% | 86,40% | 6,20% | 90,07% |
| Geolocalização da hospedagem | 71,90% | 44,00% | 17,20% | 63,42% |
| Página tem formulário | 55,00% | 39,50% | 32,30% | 53,59% |
| Razão de links para outros domínios | 84,60% | 37,60% | 6,80% | 65,39% |
| Razão de objetos para outros domínios | 84,90% | 47,90% | 8,50% | 69,67% |
| Palavras-chave no título da página | 93,10% | 28,80% | 2,10% | 63,32% |

Manter uma taxa de falso positivo baixo é importante, pois classificar uma página de uma entidade legítima como *phishing* é prejudicial tanto para a entidade quanto para a credibilidade da classificação. Dentre as doze características, sete apresentaram taxa de falsos positivos inferiores a 10%: URL baseada em IP, quantidade de pontos na URL, palavras-chave na URL, *pagerank*, razão de *links* para outros domínios, razão de objetos para outros domínios e palavras-chave no título da página. Foram realizados experimentos com esse conjunto de características para verificar se haveria melhora na taxa de falso positivo. Os resultados são mostrados na tabela 3.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 3. Resultado utilizando as características que tiveram taxa de falsos positivos menores que 10%**

| Classificador | Precisão | Taxa de verdadeiros positivos | Taxa de falsos positivos | Taxa de acertos |
|---|---|---|---|---|
| SVM | 92,90% | 97,00% | 7,40% | 94,78% |
| KNN | 93,80% | 96,80% | 6,40% | 95,16% |
| Regressão Logística | 92,60% | 97,10% | 7,70% | 94,71% |

Todos os classificadores obtiveram uma taxa de falsos positivos pior que usando todas as características. Em geral, todas as métricas foram piores para os três classificadores, a exceção da taxa de acertos do KNN que obteve 95,16% frente a 93,97% obtido anteriormente.

Diversas outras combinações das características foram testadas, mas sempre com resultado inferior que utilizando todas as características. Isso pode ter ocorrido devido o fato de haver uma grande variabilidade das características presentes em páginas *phishing*. Isso faz com que sejam necessárias diferentes combinações de características para detectar cada ataque, assim a remoção de uma característica impacta diretamente na precisão da detecção.

## 5. Conclusões

Este trabalho apresentou uma metodologia para a detecção de *phishing* em páginas *web*. Foi identificado um conjunto de doze características que evidencia a diferença entre páginas *phishing* e páginas legítimas. Foram realizados testes que mostraram o impacto de cada característica na classificação bem como o conjunto de todas elas.

Os resultados indicam que a classificação depende de todas as características identificadas, sendo que a remoção de uma acarreta na redução da taxa de detecção.

Uma das principais contribuições deste trabalho foi a montagem de uma base de páginas *phishing*. Além disso, o conjunto de características identificado pode ser usado como ponto de partida para um trabalho que utilize um conjunto mais amplo.

Como trabalhos futuros, serão feitas análises estatísticas mais apuradas dos resultados como as famosas curvas ROC (*receiver operator characteristic curve*). Além disso, pretende-se ampliar o conjunto de características e combinar classificadores a fim de reduzir os falsos positivos. Pretende-se, também, implementar o sistema *online* para avaliar o desempenho na detecção de *phishing* em tempo real.

## Referências

Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, S. (2007) "A Comparison of Machine Learning Techniques for Phishing Detection", In: Proc. of the 2nd Anti-Phishing Working Group Ecrime Researchers Summit, Pittsburgh, US, p. 60-69.

Anti-Phishing Working Group. (2010) "Phishing Activity Trends Report Q1 2010", http://www.antiphishing.org/reports/apwg_report_Q1_2010.pdf, dezembro 2010.

Changxin, G. (2011) "Phishing websites rake in $3 billion", In: China Daily, http://www.chinadaily.com.cn/china/2011-01/15/content_11859319.htm, janeiro 2011.

 ©2012 SBC — Soc. Bras. de Computação

Fette, I., Sadeh, N. and Tomasic, A. (2007) "Learning to Detect Phishing Emails", In: Proc. of the 16th International Conference On World Wide Web, Banff, Canada, p. 649-656.

Gartner Inc. (2009) "Gartner Says Number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008", http://www.gartner.com/it/page.jsp?id=936913, dezembro 2010.

Ludl, C., McAllister, S., Kirda, E. and Kruegel C. (2007) "On the Effectiveness of Techniques to Detect Phishing Sites", In: Proc. of the 4th International Conference On Detection Of Intrusions And Malware, And Vulnerability Assessment, Lucerne, Switzerland, p. 20-39.

MaxMind, Inc. (2011) "Geolocation and Online Fraud Prevention from MaxMind", http://www.maxmind.com, março 2011.

Netcraft. (2011) "Netcraft Toolbar", http://toolbar.netcraft.com, janeiro 2011.

Open Directory Project. (2011) "ODP - Open Directory Project", http://www.dmoz.org, janeiro 2011.

PhishTank. (2011) "PhishTank", http://www.phishtank.com, janeiro 2011.

Rosiello, A. P. E., Kirda, E., Kruegel, C. and Ferrandi, F. (2007) "A Layout-Similarity-Based Approach for Detecting Phishing Pages", In: Proc. of the 3rd International Conference On Security And Privacy In Communication Networks, Nice, France, p. 454-463.

Sheng, S., Wardman, B., Warner, G., Cranor, L. F., Hong, J. and Zhang, C. (2009) "An Empirical Analysis of Phishing Blacklists", In: Proc. of the 6th Conference On Email And Anti-Spam, Mountain View, US.

Spoofguard. (2011) "SpoofGuard", http://crypto.stanford.edu/SpoofGuard/, janeiro 2011.

WEKA. (2011) "Data Mining with Open Source Machine Learning in Java, 2010. University of Waikato", http://www.cs.waikato.ac.nz/~ml/weka,março 2011.

Whittaker, C., Ryner, B. and Nazif, M. (2010) "Large-Scale Automatic Classification of Phishing Pages", In: Proc. of the 17th Network And Distributed System Security Symposium, San Diego, US.

Zhang, Y., Hong, J. and Cranor, L. (2007) "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites", In: Proc. of the 16th International Conference On World Wide Web, Banff, Canada, p. 639-648.

 ©2012 SBC — Soc. Bras. de Computação

# Desenvolvimento de Roteiros Laboratoriais de Segurança Computacional em Ambientes Virtualizados

**Rodrigo Fantinati Ferreira[1], Carla Merkle Westphall[2]**

[1]Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

[2]Laboratório de Redes e Gerência - Departamento de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

{rodrigofantinati,carlamw}@inf.ufsc.br

***Abstract.*** *The computer security is a key factor for the smooth running and the success of the applications. Quality training becomes increasingly necessary to train good professionals in the current scenario. Given the present difficulty for the realization of practical classes in computer security due to the limitations of the infrastructure at the academic environments, this work proposes using guides in virtual environment to carry out practical security activities, without compromising the physical machine's installations and student learning due to access limitations.*

***Resumo.*** *A segurança computacional é um fator fundamental para o bom funcionamento e sucesso das aplicações. Um treinamento de qualidade torna-se cada vez mais necessário para formar profissionais capacitados no cenário atual. Tendo em vista a presente dificuldade para a realização de aulas práticas de segurança computacional devido às limitações da infraestrutura disposta pelos ambientes acadêmicos, este trabalho propõe o uso de roteiros em ambiente virtual para a realização das atividades práticas de segurança, sem comprometer as instalações das máquinas físicas e o aprendizado do aluno devido a limitações de acesso.*

## 1. Introdução

No aprendizado de segurança computacional, experimentos laboratoriais são importantes e imprescindíveis para a formação do aluno.

Muitas vezes é difícil encontrar o foco de um problema específico rapidamente em um sistema computacional devido a sua complexidade. O envolvimento de muitas variáveis pode interferir nos estudos específicos, deturpando os resultados.

Este artigo resulta do desenvolvimento de um trabalho de conclusão de curso de graduação e tem como objetivo principal a criação de roteiros laboratoriais, que permitem ao aluno seguir uma sequência de atividades definidas, em ambientes virtualizados, para o ensino prático de segurança computacional [Ferreira 2012]. Desta forma, o maior número possível de variáveis é controlado dentro do ambiente, permitindo resultados mais claros e potencialmente livres da interferência de fatores externos. Nos roteiros são abordados dois dos principais problemas de injeção [Owasp-b 2011] – de SQL e XSS (*Cross Site Scripting*) - encontrados na literatura e nas organizações. Os roteiros também usam duas ferramentas importantes na análise da segurança: Nmap [Nmap 2012] e Nessus [Tenable Network Security 2012].

 ©2012 SBC — Soc. Bras. de Computação

O restante deste artigo está organizado da seguinte forma: a seção 2 descreve os trabalhos relacionados; a seção 3 descreve dois dos principais problemas de injeção encontrados em aplicações Web; a seção 4 descreve duas ferramentas importantes de análise de segurança; a seção 5 descreve o desenvolvimento dos roteiros e a seção 6 apresenta as conclusões deste trabalho.

## 2. Trabalhos Relacionados

Segundo [Gregg 2008] o uso de máquinas virtuais é uma boa escolha para laboratórios, pois permite testar com facilidade ferramentas de segurança, testar atualizações e estudos de exames de certificação. Máquinas virtuais apresentam a vantagem de isolamento, não colocando em risco a máquina hospedeira, sua rede, nem outros usuários da Internet [Nance, Hay, Dodge, Seazzu e Burd 2009].

Apesar da possibilidade de ataques à máquina hospedeira pela utilização de máquinas virtuais [Vacca 2009], ambientes virtualizados são considerados seguros e são recomendados por diversos autores como ferramenta para o ensino de segurança e outras atividades.

Os autores [Chen e Tao 2011] desenvolveram uma nova ferramenta de ensino de segurança para aplicações web, chamada SWEET (*Secure WEb dEvelopment Teaching*). O propósito deste projeto é reforçar a experiência de aprendizado disponibilizando vários roteiros existentes nos módulos que o aluno deve executar.

O trabalho de [Willems, Klingbeil, Radvilavicius, Cenys e Meinel 2011] apresenta o projeto Tele-Lab. O projeto Tele-Lab oferece um sistema para treinamento de segurança em um ambiente de laboratório virtual remoto, online e acessível por usuários registrados. A plataforma Tele-Lab oferece um ambiente individual e prevê o uso de roteiros, que são chamados unidades de aprendizado e são executados pelos alunos durante o aprendizado.

O projeto OWASP Mantra [Mantra 2012] é um framework de segurança multiplataforma construído na forma de browser, usado para o aprendizado de testes manuais de penetração de aplicações web. Permite a realização de vários testes como injeção de comandos SQL e *Cross Site Scripting* (XSS).

## 3. Principais Falhas de Injeção

Os problemas de injeção atingem grande parte das aplicações web e podem trazer diversas complicações ao servidor e aos usuários destas aplicações. Nesta seção serão abordados os dois principais problemas de injeção que mais afetam as aplicações web segundo os dados levantados pela OWASP [Owasp-b 2011]: Injeção de SQL (SQL Injection) e *Cross Site Scripting* (XSS).

Segundo a definição da [Owasp-c 2011] um ataque de *SQL Injection* consiste na inserção ou "injeção" de uma requisição SQL (*Structured Query Language*) através de uma entrada de dados de um cliente em uma aplicação.

Diferentemente de um ataque *Cross Site Scripting* (XSS) que tem como alvo os usuários, um ataque *SQL Injection* tem como alvo o banco de dados.

Um ataque de injeção SQL é possível quando uma aplicação utiliza dados que podem ser controlados por um atacante como parte de uma requisição SQL. O atacante

 ©2012 SBC — Soc. Bras. de Computação

pode enviar uma entrada de dados maliciosa de forma que a requisição enviada ao banco de dados será interpretada de uma forma diferente da forma prevista pelo programador da aplicação [Daswani, Kern e Kesavan 2007].

Se um atacante pode controlar a entrada que é enviada para uma requisição SQL e manipula essa entrada de forma que o dado seja interpretado como código ao invés de dado, então o atacante pode executar código no banco de dados [Clarke 2009]. As informações podem ser inseridas, modificadas ou até apagadas. A ocorrência deste fato pode resultar na incoerência das informações contidas no banco de dados e significar a existência de grandes perdas para instituições e organizações.

Segundo a definição da OWASP [Owasp-a 2011] ataques de *Cross Site Scripting* são um tipo de problema de injeção, no qual scripts maliciosos são injetados em sites web. O documento [Owasp-a 2011] divide os ataques XSS em 2 categorias:

- Armazenados (*Stored*): São os ataques nos quais o código injetado é permanentemente armazenado nos servidores alvo (um banco de dados, fórum de mensagens, dentre outros). Assim, a vítima recebe o código malicioso do servidor quando ela solicita a informação armazenada.
- Refletidos (*Reflected*): São os ataques nos quais o código injetado é refletido dos servidores web, como em uma mensagem de erro, resultado de pesquisa, ou qualquer outra resposta que inclua alguma parte ou toda a entrada enviada ao servidor como parte da requisição. Quando o usuário clica em um link malicioso ou submete um formulário malicioso, o código injetado vai até o servidor web vulnerável, que reflete o ataque de volta ao navegador do usuário. O navegador então executa o código.

A figura 1 mostra um possível cenário passível de ataque XSS, no qual a aplicação utiliza dados não confiáveis na construção da página HTML sem validação ou escape de caracteres:

```
(String) page += "<input name='creditcard' type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

**Figura 1. Cenário passível de ataque XSS [Owasp-d 2012]**

O atacante pode modificar o parâmetro ‘CC’ no browser para o conteúdo mostrado na figura 2:

```
'<script>document.location= 'http://www.attacker.com/cgi-
bin/cookie.cgi?foo='+document.cookie</script>'.
```

**Figura 2. Parâmetro 'CC' utilizado na figura 1 [Owasp-d 2012]**

A conseqüência desta ação mostrada nas figuras 1 e 2 é o roubo de sessão do usuário, o qual terá os valores do *cookie* associado a este documento enviados para o site do atacante [Owasp-d 2012].

## 4. Principais Ferramentas de Análise de Segurança: NMAP e NESSUS

Ferramentas de análise de segurança podem ser utilizadas por pessoas mal intencionadas à procura de falhas em sistemas, mas também podem ser utilizadas para a detecção e prevenção de problemas por profissionais da área de segurança.

 ©2012 SBC — Soc. Bras. de Computação

O Nmap é uma ferramenta de código aberto para exploração de rede e auditoria de segurança, o qual serve tanto para examinar redes amplas bem como hosts individuais. Utilizando pacotes IP em estado bruto (pacote IP sem modificações), o Nmap consegue descobrir de maneira eficiente, quais hosts estão disponíveis na rede e quais serviços estes oferecem [Nmap 2012].

O Nmap possui técnicas de descobrimento de hosts na rede de difícil detecção. Utilizando-se da flag SYN marcada em pacotes TCP, o Nmap tenta estabelecer uma conexão com o host alvo, mas ao receber uma resposta positiva (TCP / SYN / ACK), ele envia um pacote RST ao host, resetando a conexão, ao invés de confirmá-la com o envio de outro pacote TCP/ACK. Desta forma o Nmap não irá completar a conexão com o host alvo e dificilmente será detectado.

O Nessus é uma ferramenta bem conhecida, utilizada para realizar exames (*scans*) em sistemas computacionais a procura de vulnerabilidades. Além de vulnerabilidades, é possível também verificar especificações de conformidade, violações de políticas de conteúdo e outras anomalias em um computador local [Tenable Network Security 2012].

O Nessus realiza os testes de segurança nos alvos através do uso de *plugins* que são escritos em uma linguagem chamada NASL (*Nessus Attack Scripting Language*) e se ligam a cada porta especificada para fazerem as verificações à procura de falhas.

## 5. O Desenvolvimento de Roteiros

A metodologia utilizada para a construção dos roteiros baseou-se em técnicas para testar diretamente o problema envolvido, provocando uma reflexão sobre os motivos que nos levaram a realizar a tarefa de tal maneira. Desta forma, o aluno compreende o problema, passando por uma linha de raciocínio lógico para atingir os objetivos.

Inicialmente é descrito o ambiente virtual criado e depois são descritos, de forma resumida, as tarefas que os alunos executam em cada um dos roteiros elaborados. Foram elaborados quatro roteiros: uso do Nmap, uso do Nessus, teste de injeção de SQL usando o Mantra, teste de XSS usando o Mantra. Os roteiros são descritos como uma seqüência de passos que o aluno deve executar.

### 5.1 O Ambiente Virtual

Foram criadas três máquinas virtuais – uma máquina com Linux Ubuntu 11.10, uma máquina com Windows XP e outra máquina com o sistema CentOS5. Diferentes sistemas operacionais possibilitam testes com maior abrangência. A máquina com sistema CentOS5 é chamada *Capture The Flag* e serve de suporte para testes envolvendo servidores. A figura 3 mostra uma imagem das máquinas Windows XP e Ubuntu 11.10.

Para criar um ambiente controlado é importante começar a criação do início, pois ambientes prontos podem conter configurações indesejadas e ou até problemas pré-existentes não resolvidos que podem influenciar e atrapalhar os testes a serem realizados. Em ambos os sistemas Ubuntu e Windows XP, alguns serviços foram instalados e ativados propositalmente para a realização de testes de detecção de portas e vulnerabilidades.

 ©2012 SBC — Soc. Bras. de Computação

Para o processo de virtualização, utilizamos o software VMware Player. Na construção das máquinas virtuais utilizamos as configurações padrões sugeridas pelo VMware, exceto pela configuração de memória RAM, a qual foi configurada para 512MB em todas as máquinas virtuais, pois é um fator limitante para o número de máquinas que conseguimos executar simultaneamente.

**Figura 3. Máquinas virtuais criadas**

O ambiente virtual é formado por várias máquinas virtuais instaladas em um mesmo host, criando uma rede virtual. O VMware aloca automaticamente as máquinas virtuais inicializadas em uma mesma faixa de IPs. Nas máquinas Windows XP e Ubuntu foram instaladas as ferramentas Nmap e Nessus para a execução dos roteiros.

### 5.2 NMAP

Os objetivos deste roteiro são: (a) Descobrir os hosts ativos da rede; (b) Examinar os hosts descobertos a procura de portas abertas; e, (c) Realizar as ações anteriores de diferentes formas em diferentes contextos.

**Tarefa 1: A descoberta de hosts**

Situação 1: Suponha que você conheça o endereço IP do alvo que deseja examinar em sua rede. Este endereço IP é 192.168.6.2. Realize o exame simplesmente digitando o endereço alvo no campo "Alvo" da interface Zenmap (interface gráfica do Nmap).

Situação 2: Executando o número máximo de máquinas virtuais na rede que você puder, suponha que você desconheça o número de máquinas existentes na rede e os seus endereços IPs. Suponha também que a rede é de sua propriedade e que não há problema nenhum examinar a rede a procurar de hosts ativos.

1. Digite no campo "Comando": nmap –sP -v 192.168.6.1/24

Situação 3: Tome por base a situação 2, mas desta vez, suponha que a rede não é de sua propriedade e que você deve evitar ao máximo levantar suspeitas sobre o exame (*scan*).

2. Execute o exame utilizando pacotes com a flag SYN: nmap –PS -v 192.168.6.1/24
3. Execute o exame utilizando pacotes com a flag ACK: nmap –PA -v 192.168.6.1/24

Situação 4: Suponha que você queira descobrir os hosts da rede e você sabe que podem existir firewalls bloqueando o caminho. Caso os pacotes do Nmap encontrem um

 ©2012 SBC — Soc. Bras. de Computação

firewall, a resposta pode ser comprometida e não condizer com a realidade. Sabendo disso, você precisa maximizar as chances de obter bons resultados.

4. Certifique-se de que o Firewall de uma das máquinas Windows esteja desativado e faça o teste utilizando pacotes UDP: nmap –PU -v 192.168.6.1/24
5. Ative o Firewall do Windows de uma das máquinas virtuais e repita o procedimento anterior.

**Tarefa 2: Exame (*scan*) de portas**

Situação 1: Você não tem privilégios de administrador (ou root) mas quer fazer um exame nos hosts descobertos a procura de portas.

1. Certifique-se de que haja na rede pelo menos 1 máquina virtual Windows com o Firewall ativado. Em um terminal de comandos Ubuntu sem privilégios de root execute: nmap –sT -v 192.168.6.X (troque o X pelo número do IP a ser varrido).
2. Desative o Firewall do Windows da máquina virtual e repita o procedimento anterior.

Situação 2: Você tem privilégios de administrador (ou root) e quer fazer um exame nos hosts descobertos a procura de portas.

3. Certifique-se de que haja na rede pelo menos 1 máquina virtual Windows com o Firewall ativado.Execute o comando em um terminal de comandos do Ubuntu: sudo nmap –v –sS 192.168.6.X (troque o X pelo número do IP a ser varrido e utilize a senha labtest).
4. Desative o Firewall do Windows da máquina virtual e repita o procedimento anterior.
5. Faça um exame mais completo: sudo nmap –v –sS -sU 192.168.6.X.

## 5.3 NESSUS

Os objetivos deste roteiro são: (a) Entender e criar políticas de exame (*scan*) para os nossos testes; (b) Descobrir vulnerabilidades de segurança em diferentes hosts da rede interna e externa; e, (c) Analisar o resultado de tais vulnerabilidades e as medidas que podem ser tomadas para mitigá-las ou eliminá-las.

Depois do processo de instalação e ativação da ferramenta, não descritos aqui, iremos realizar 3 tarefas seqüenciais. Não é possível descrever as tarefas com detalhes suficientes e, por isso, são descritas na forma resumida.

**Tarefa 1: Definir as políticas**

1. Passos para a criação da política 1 (Rede virtual) e da política 2 (Rede externa).

**Tarefa 2: Realizar o exame**

1. Passos para a realização do exame na rede virtual e na rede externa.

**Tarefa 3: Analisar os relatórios**

1. Acessando e visualizando a tela de relatórios “Reports”.
2. Visualizando a lista de vulnerabilidades encontradas em um host específico da rede. A figura 4 mostra um resultado exemplo obtido no Nessus que exibe as

 ©2012 SBC — Soc. Bras. de Computação

vulnerabilidades encontradas em um dos hosts da rede virtual, mostrando a porta e protocolo, o grau de severidade e o nome da vulnerabilidade.

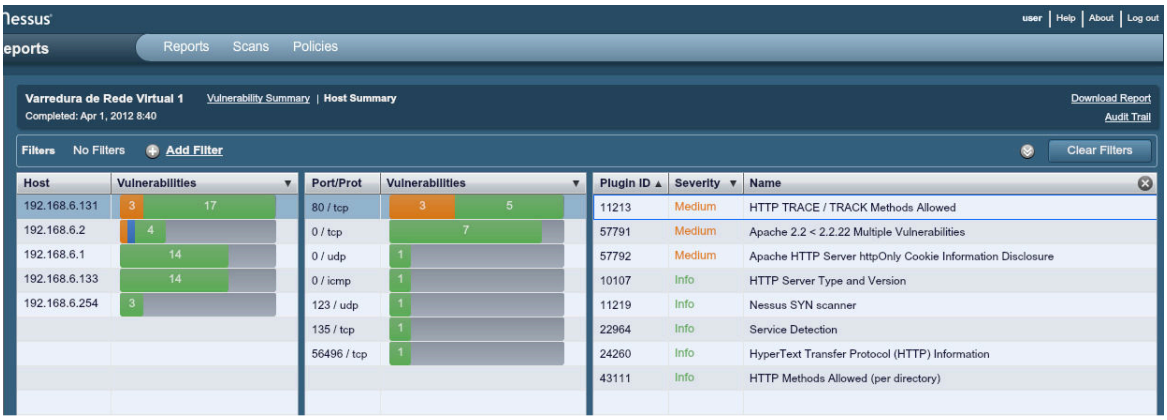



**Figura 4. Relatórios no Nessus**

## 5.4 Injeção de SQL Utilizando o MANTRA

Os objetivos deste roteiro são [Ferreira 2012]: (a) Utilizar conhecimentos da linguagem SQL para obter informações armazenadas em um servidor vulnerável; e, (b) Utilizar comandos SQL para obter acesso privilegiado ao servidor vulnerável.

### Tarefa 1: Obter acesso de administrador sem digitar a senha

1. Abra o navegador Mantra e em seguida digite o endereço encontrado para acessar o servidor HTTP do LAMPSecurity CTF6. (Ex: http://192.168.6.135). Clique no menu "Log In" para acessarmos a tela de autenticação.
2. Utilize a ferramenta *Live HTTP Headers* para capturar e editar os cabeçalhos de requisições HTTP feitas entre a sua máquina e o alvo e se esquivar do filtro Javascript. Depois, acesse a página de "Log in" e digite quaisquer dados ("teste", por exemplo) para usuário e senha. Observe a captura na figura 5.
3. Nesta janela poderemos editar as informações de login e senha e reenviá-las ao servidor sem que elas sejam filtradas pelo Javascript. Substitua a linha para que fique da seguinte forma: [ username='or 1=1 #&password=') or 1=1 # ] sem os colchetes e em seguida clique em "Replay".
4. Finalmente conseguimos efetuar o login com sucesso sem digitar dados como usuário e senha (foram necessários passos intermediários, como a descoberta do banco de dados do tipo MySQL, que foram omitidos).

### Tarefa 2: Obter dados do banco de dados do servidor, como o usuário e senha do administrador

1. Através do Mantra, acesse o endereço do servidor alvo. Depois de executar alguns outros passos intermediários (todos os passos podem ser consultados em [Ferreira 2012]), coloque a seguinte linha na *hackbar* do Mantra: `http://192.168.6.135/?id=01 UNION SELECT 1,user_username,user_password,4,5,6,7 from user`. Depois de executar esta linha, é obtido o nome de usuário e a senha do administrador das tabelas pesquisadas (Figura 6).

 ©2012 SBC — Soc. Bras. de Computação

2. Após o uso da ferramenta de tradução de MD5 do MANTRA, é feita a descoberta da senha obtida e feito então o teste de acesso no servidor alvo com nome de usuário e senha obtidos.

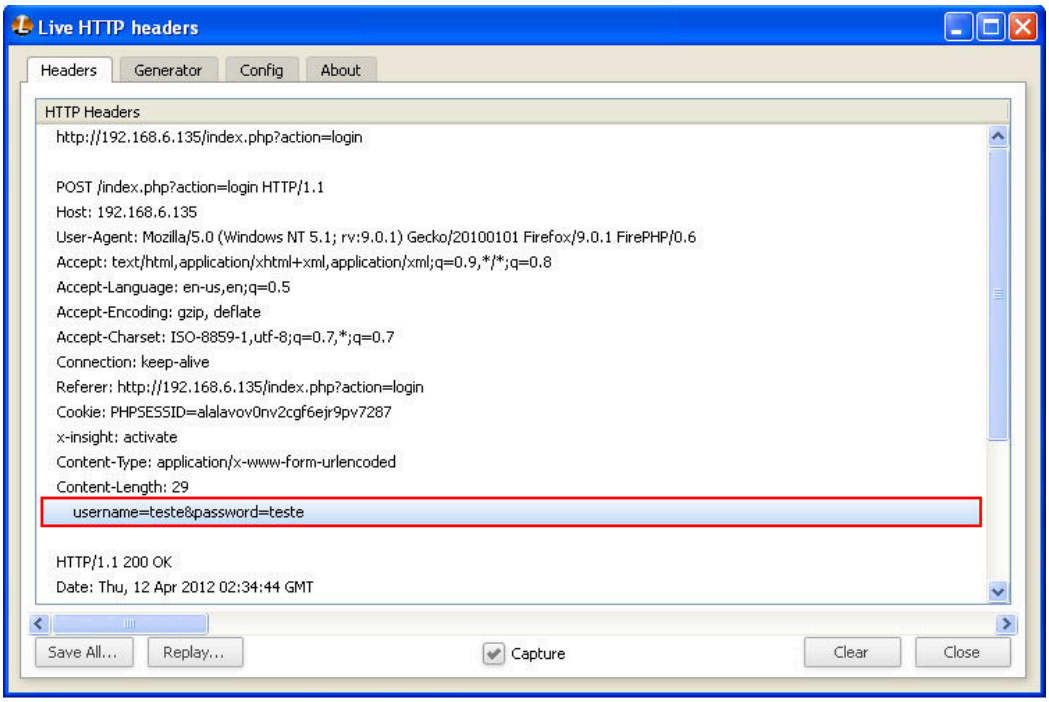



**Figura 5. Live HTTP headers**

**Figura 6. Visualização da página após injeção de SQL**

## 5.5 Cross Site Scripting (XSS) Utilizando o MANTRA

O objetivo deste roteiro é causar o redirecionamento automático de uma página por meio de um ataque XSS. Outras tarefas deste roteiro não foram descritas.

 ©2012 SBC — Soc. Bras. de Computação

**Tarefa 1: Teste de um ataque de XSS refletido.**

1. Crie no diretório "C:\Arquivos de programas\Apache Software Foundation\Apache2.2\htdocs" uma página HTML chamada "teste.php" que cria um link com o endereço passado como parâmetro no navegador.
2. Acesse a página criada do navegador Mantra passando como parâmetro "?linklocation=index.html". Clique no link da página e veja o resultado.
3. Altere o endereço acessado inserindo um código Javascript no parâmetro da página acessada para testar um ataque XSS refletido. Coloque no navegador Mantra a seguinte linha:

```
http://192.168.6.131/teste.php?linklocation=index.html"><script>
alert("Teste de XSS Refletido")</script><a
href="index.html”>LINK</a>
```

Note que um alerta surgiu na tela antes de o *link* ser mostrado, o que significa que conseguimos executar um JavaScript ao acessar a página, simplesmente acoplando o JavaScript ao endereço de acesso. A inserção de um fechamento de tag (">) logo após "index.html" indica o fechamento da tag (<a href= ") permitindo a inserção seqüencial de um código JavaScript através da tag (<script>). Ao final, repare que a abertura e fechamento de tags ajustam-se perfeitamente, produzindo um código consistente.

O sucesso dessa requisição representa que podemos, por exemplo, inserir um código JavaScript que referencia outro código JavaScript externo, o qual não precisa estar contido no *link*, de forma a não deixá-lo muito longo. Parte do *link* pode ainda ser codificada, de modo a impedir a leitura e compreensão pelo usuário.

## 6. Conclusões

Com o aumento do uso de aplicativos web, varejo online e o crescimento da utilização de tecnologia computacional para fins governamentais e sociais, a segurança torna-se indispensável. As falhas de injeção tornaram-se grandes vilãs deste cenário. Por isso, são importantes estudos, aperfeiçoamentos e criação de novas tecnologias.

O uso da virtualização para a realização dos testes e experimentos mostrou-se satisfatório para atingir os objetivos almejados. A virtualização permitiu com que as cópias das máquinas virtuais originais permanecessem inalteradas em um estado consistente, enquanto que cópias utilizadas para a realização dos roteiros laboratoriais pudessem sofrer modificações e até danos. Desta forma, o aluno pode experimentar na prática diversas atividades que seriam danosas caso fossem executadas em um computador de um laboratório da universidade, por exemplo.

O nosso ambiente é composto pelas máquinas virtuais criadas e configuradas (Windows XP, Linux Ubuntu, CentOS5), pelos softwares instalados nas máquinas (Nmap e Nessus) e pelos roteiros desenvolvidos.

Uma avaliação sobre a eficácia e eficiência do nosso ambiente ainda não foi feita. O passo inicial foi a aplicação do roteiro do Nmap na disciplina de redes de computadores no curso de graduação. Foi extremamente satisfatório o uso do roteiro em laboratório para possibilitar o entendimento de portas, serviços, informações obtidas com o nmap, *firewalls* e possíveis exames que uma rede pode sofrer. O uso dos outros roteiros está previsto para futuras disciplinas de segurança e redes de computadores.

 ©2012 SBC — Soc. Bras. de Computação

O desenvolvimento deste trabalho se aproxima bastante da proposta de [Chen e Tao 2011] que desenvolveram a ferramenta SWEET, pois ambos os trabalhos apresentam o conteúdo de forma modular. Já o projeto Tele-Lab [Willems, Klingbeil, Radvilavicius, Cenys e Meinel 2011], diferente deste, utiliza recursos para que o aprendizado aconteça de forma remota, atendendo usuários sem recursos locais.

Como trabalhos futuros, novos roteiros podem ser criados e incorporados a este ambiente virtual. Além disso, pode ser feita uma avaliação sobre esta forma de ensino.

## Referências

Chen, L. e Tao, L. (2011) "Teaching Web Security Using Portable Virtual Labs", In: 11th IEEE Int. Conference on Advanced Learning Technologies, EUA, p. 491-495.

Clarke, J. (2009) "SQL Injection Attacks and Defense", Ed. Syngress Publishing, EUA.

Daswani, N., Kern, C. e Kesavan, A. (2007) "Foundations of Security – What every Programmer Needs to Know", Editora Apress, EUA.

Ferreira, R. F. (2012) "Desenvolvimento de Roteiros Laboratoriais de Segurança Computacional em Ambientes Virtualizados", Trabalho de Conclusão de Curso, UFSC, http://projetos.inf.ufsc.br/arquivos_projetos/projeto_1279/TCC-%20DOCUMENTO%20FINAL.pdf, Agosto.

Gregg, M. (2008) "Build Your Own Security Lab: A Field Guide for Network Testing", Editora Wiley Publishing Inc, EUA.

Mantra. (2012) "OWASP Mantra Lexicon", http://www.getmantra.com/download/mantra-security-toolkit.html, Julho.

Nance, K., Hay, B., Dodge, R., Seazzu, A. e Burd, S. (2009) "Virtual Laboratory Environments: Methodologies for Educating Cybersecurity Researchers", Methodological Innovations Online, v. 4, n. 3, p. 3-14.

Nmap. (2012) "Guia de Referência do Nmap", http://nmap.org/man/pt_BR/, Janeiro.

Owasp-a. (2011) "Cross-site Scripting (XSS)", https://www.owasp.org/index.php/Cross-site_Scripting_(XSS), Junho.

Owasp-b. (2011) "OWASP Top 10 – 2010 The Ten Most Critical Web Application Security Risks", http://www.owasp.org/index.php/Top_10, Junho.

Owasp-c. (2011) "SQL Injection", https://www.owasp.org/index.php/SQL_Injection, Junho.

Owasp-d. (2012) "Top 10 2010-A2-Cross-Site Scripting (XSS)", https://www.owasp.org/index.php/Top_10_2010-A2, Maio.

Tenable Network Security. (2012) "Nessus 5.0 - Installation and Configuration Guide", http://static.tenable.com/documentation/nessus_5.0_installation_guide.pdf, Agosto.

Vacca, J. R. (2009) "Computer and Information Security Handbook", Editora Morgan Kaufmann, Canada.

Willems, C., Klingbeil, T., Radvilavicius, L., Cenys, A., e Meinel, C. (2011) "A Distributed Virtual Laboratory Architecture for Cybersecurity Training", In: 6th Int. Conference on Internet Technology and Secured Transactions, p. 408-415.

 ©2012 SBC — Soc. Bras. de Computação

# CURUPIRA: Uma Solução para Controle dos Pais em Navegadores Web

**Diego Santos Azulay, Hugo Assis Cunha, Eduardo Luzeiro Feitosa**

Instituto de Computação (IComp)
Universidade Federal do Amazonas (UFAM)
CEP 69077-000 Campus Universitário - Manaus - AM

{dsazulay@gmail.com, hugo.assis.cunha@gmail.com, efeitosa@icomp.ufam.edu.br}

***Abstract.*** *This research aims to develop a Web browsers Parental Control solution. The goal of Parental Control Web Solution is to filter content, moderate the use and control the activities in any environment that can offer some kind of risk to any user (especially children and adolescents). Focusing on Internet security, Parental Control solutions are very important for parents to control and monitor access to their children and also filter the information coming to them. Using filtering techniques, combined with a simple and intuitive interface, the proposed solution, called CURUPIRA, will monitor and control access to Internet, according to the definition of parents and guardians, preventing children and adolescents are exposed the inappropriate content.*

***Resumo****. Este trabalho de pesquisa visa desenvolver uma solução de Parental Control para navegadores Web. Soluções de Parental Control (traduzindo do inglês, controle dos pais) tem como objetivo filtrar conteúdo, moderar o uso e controlar as atividades em qualquer ambiente que possa oferecer algum tipo de risco a algum usuário (especialmente crianças e adolescentes). Focando na segurança na Internet, as soluções de Parental Control são de suma importância para os pais poderem controlar e monitorar o acesso de seus filhos e também filtrar as informações que chegam aos mesmos. Utilizando técnicas de filtragem, aliada a uma interface de configuração simples e intuitiva, a solução proposta, denominada CURUPIRA, irá monitorar e controlar o acesso a Internet, de acordo com a definição de pais e responsáveis, evitando que crianças e/ou adolescentes sejam expostos a conteúdos considerados impróprios.*

## 1. Introdução

É impossível, nos dias de hoje, negar que o advento da Internet tenha trazido uma série de benefícios aos seres humanos nas mais diferentes áreas de atividade e conhecimento. Uma das principais características da Internet é a sua capacidade de fornecer, de maneira transparente, acesso a informações e serviços diferenciados a uma quantidade ilimitada de usuários. Além disso, a Internet é uma das poucas plataformas operando sem controle centralizado, com base na boa fé dos seus membros. Embora esta descentralização seja considerada um fator de sucesso, a falta de controle também traz consigo uma série de problemas. Um deles é a exposição de crianças e adolescentes a uma gama de conteúdos

 ©2012 SBC — Soc. Bras. de Computação

inapropriados (pornografia, violência, automutilação e incitamento a atos ilícitos, por exemplo).

Com a disponibilização da Internet de banda larga a um número cada vez maior de pessoas, a exposição de conteúdo ofensivo vem preocupando pais e responsáveis em todo o mundo. Independente do status socioeconômico, basta um simples e inocente "clique" para que a criança ou adolescente seja exposto a material não apropriado. De acordo com um levantamento do Escritório de Justiça e Delinquência Juvenil do Departamento de Justiça dos Estados Unidos (Finkelhor et al., 2011), 19% dos adolescentes americanos utilizando a Internet já recebeu uma mensagem ou um contato com conotação sexual. No Brasil, cerca de 25% das crianças que acessam a Internet no Brasil já sentiram medo ou perigo no ambiente online. As informações são da pesquisa TIC Crianças 2010 (Cetic.br, 2010), publicada pelo CGI (Comitê Gestor da Internet), que avalia o uso de tecnologias da informação e comunicação por este público.

Especialistas (educadores, pedagogos, psicólogos, entre outros) são unânimes em dizer que a abordagem mais recomendada contra os perigos da Internet é o diálogo entre pais e filhos (Santos e Manteigas, 2010). Contudo, muitas vezes essa solução não pode ser alcançada e uma das saídas é a utilização de ferramentas que exerçam o papel de controle dos pais (do inglês, *parental control* ou *parental monitoring*), permitindo aos pais ou responsáveis monitorar ou limitar o que crianças e adolescentes veem ou restringir o tempo de execução de certas atividades (Safer Internet Program, 2012)(Cioffi et al. 2009).

Embora existam soluções (tipicamente na forma de plug-ins) para os mais utilizados navegadores Web (Firefox, Chrome e Internet Explorer), tais soluções ou são muito específicas, realizando apenas uma ou duas funções e obrigando os pais a procurarem e instalarem diferentes extensões que atendam as suas necessidades, ou apresentam difícil manuseio e com interfaces nada intuitivas, exigindo que os pais sejam profundos conhecedores da área.

A fim de solucionar este problema, este trabalho propõe uma solução de controle dos pais para navegadores Web, denominada **CURUPIRA**, capaz de restringir acesso a conteúdo inofensivo através de uma interface simples e intuitiva.

As demais seções deste artigo explicam o CURUPIRA. Primeiro são apresentadas definições e funcionalidade de soluções de *parental control*. Além disso, alguns exemplos são discutidos. Em seguida, uma visão geral da solução é apresentada, incluindo sua implementação. Depois, uma avaliação inicial é apresentada para validar a ferramenta desenvolvida. Por último, algumas considerações são feitas.

## 2. *Parental Control* na Internet

Segundo a Common Sense Media (Common Sense Media, 2010), ferramentas de *parental control* são instrumentos para proteger crianças e adolescentes das ameaças existentes na Internet. Podem ser software ou hardware e sua finalidade é filtrar ou bloquear qualquer acesso ou conteúdo que possa ser considerado ofensivo ou inapropriado para crianças ou adolescentes.

 ©2012 SBC — Soc. Bras. de Computação

De acordo com o Programa Internet Segura da Comissão Europeia (Safer Internet Program, 2012), soluções de *parental control* desempenham três funções principais de proteção:

- **Personalização de filtros para conteúdo Web**: permitem que crianças e adolescentes visualizem o conteúdo de acordo com um conjunto de critérios específicos definidos durante a configuração da ferramenta. Tipicamente, os pais fazem o bloqueio ou liberação de algum conteúdo através de tópicos, lista de URLs ou usando palavras-chave específicas.
- **Bloquear o uso**: permitem bloquear o uso de um protocolo, aplicação ou serviço, independente da inadequação do conteúdo.
- **Monitorar a utilização do protocolo/aplicação/serviço e do conteúdo Web acessado:** permitem acompanhar a utilização da Internet e reportar o que foi, quando foi e por quanto tempo foi acessado um sítio Web específico, uma aplicação ou protocolo.

### 2.1 Soluções existentes

Dentre os três mais utilizados navegadores Web (Firefox, Chrome e Internet Explorer), somente o Internet Explorer fornece suporte nativo a funções de *parental control*. Embora não possua tal suporte nativamente, o Firefox possui uma gama variada de extensões (plug-ins) para este fim. Entre os mais populares destacam-se: Glubble (Glubble, 2012), que permite customização de configurações e diferenciação entre os usuários (criação de contas); FoxFilter (InspiredEffect.com, 2012), que inclui proteção baseada em senhas; LeechBlock (Anderson, 2012), que permite restringir o acesso a sítios Web através de tempo; e ProCon Latte Content Filter (Corvineum, 2012), que utiliza expressões regulares para limitar e bloquear acessos. Em relação ao Chrome, apenas em Maio deste ano surgiu uma primeira solução de *parental control*, uma adaptação do FoxFilter.

Na área acadêmica, existem algumas soluções como: o PlayerGX (Holthe, 2010), um plug-in focado em filtrar gráficos, imagens e dados multimídia considerados inapropriados; e o XFighter, um sistema de filtragem de conteúdo Web baseado em aprendizagem de máquina. Já os autores em (Cioffi et al. 2009) desenvolveram um sistema de *parental control* baseado na detecção de anomalias alimentado por sugestões de pais e/ou responsáveis.

## 3. CURUPIRA

Sistemas de controle dos pais são softwares que permitem aos pais ou responsáveis monitorar ou limitar o que uma criança ou adolescente pode ver ou fazer na Internet e/ou limitar o tempo de navegação. Eles ajudam a prevenir que crianças e adolescente acessem conteúdos inadequados ou mesmo prejudiciais (por exemplo, pornografia, violência, racismo, incitação ao uso de drogas).

 ©2012 SBC — Soc. Bras. de Computação

Para esse fim, este trabalho desenvolveu uma solução, **CURUPIRA**, capaz de identificar e bloquear conteúdo Web considerado inapropriado para crianças e adolescentes. Mais especificamente, CURUPIRA utiliza duas estratégias principais de filtragem: listas negras (*blacklist*) de URL e listas de conteúdo. Na primeira, pais ou responsáveis classificam as URLs com conteúdo inapropriado, inserindo-as em uma *blacklist*. Na segunda, pais ou responsáveis inserem palavras-chaves que remetem ou representam conteúdo inapropriado em uma *blacklist*.

O fluxo de funcionamento do CURUPIRA é apresentado na Figura 1 e pode ser descrito como se segue. Quando uma URL é digitada pelo usuário, seu endereço é analisado pelo filtro de URL para verificação. Se o endereço da URL estiver na *blacklist* de páginas consideradas inapropriadas, seu acesso é automaticamente bloqueado e um alerta (e-mail e/ou SMS) é enviado para o pai ou responsável. Caso contrário, a URL segue para avaliação de conteúdo. Nesta etapa, cada página em questão tem seu conteúdo avaliado pelo filtro de conteúdo. O objetivo é verificar se alguma palavra ou texto classificado como inapto está presente no conteúdo da página. Para tanto, o filtro de conteúdo utiliza expressões regulares. É importante ressaltar que todas as classificações (URL e conteúdo) são configuradas por pais ou responsáveis.

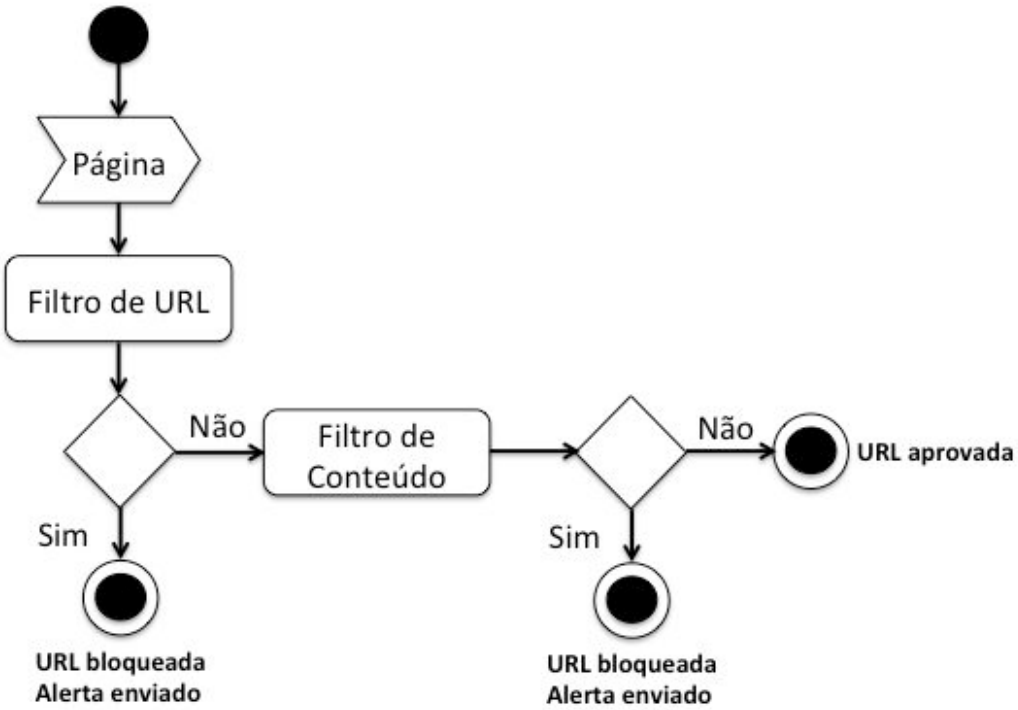



Figura 1. Fluxo de funcionamento do Curupira.

## 4. IMPLEMENTAÇÃO E INSTALAÇÃO

Antes de iniciar a descrição da implementação, é necessário esclarecer os motivos da escolha do navegador Web Chrome como objeto de estudo e implementação da solução CURUPIRA.

 ©2012 SBC — Soc. Bras. de Computação

Primeiro, após meses de exaustiva busca na Internet iniciada em Abril de 2011, percebeu-se que não existiam soluções de *parental control* para o Chrome. Somente em Maio deste ano que a InspiredEffect.com anunciou uma versão do FoxFilter para Chrome. Em segundo lugar, o Chrome possui uma API simples e prática, o que facilita o desenvolvimento da solução. Além disso, nenhum deles possui versões para a língua portuguesa.

### 4.1 Implementação

A solução CURUPIRA foi desenvolvida na forma de uma extensão (plug-in) para o navegador Google Chrome. A extensão funciona como uma página Web. Sua interface foi feita através de HTML e CSS e a programação em si é feita com JavaScript. Para interação com o navegador, foram utilizadas funções da API do Chrome.

A Figura 2 ilustra os componentes internos da solução. O API do plug-in fornece a interface para o navegador Chrome. Ela também implementa os módulos que executam as opções de filtragem (URL e Conteúdo) para o usuário (pais ou responsáveis), adicionando ou removendo filtros do processo de validação. O plug-in faz uso do módulo *chrome.tabs* para obter informações sobre cada atualização das abas do navegador, permitindo assim fazer a validação dos sítios Web através dos filtros. O plug-in também possui um arquivo JSON de manifesto que inclui informações como nome, versão e permissões das ações executadas pela extensão.

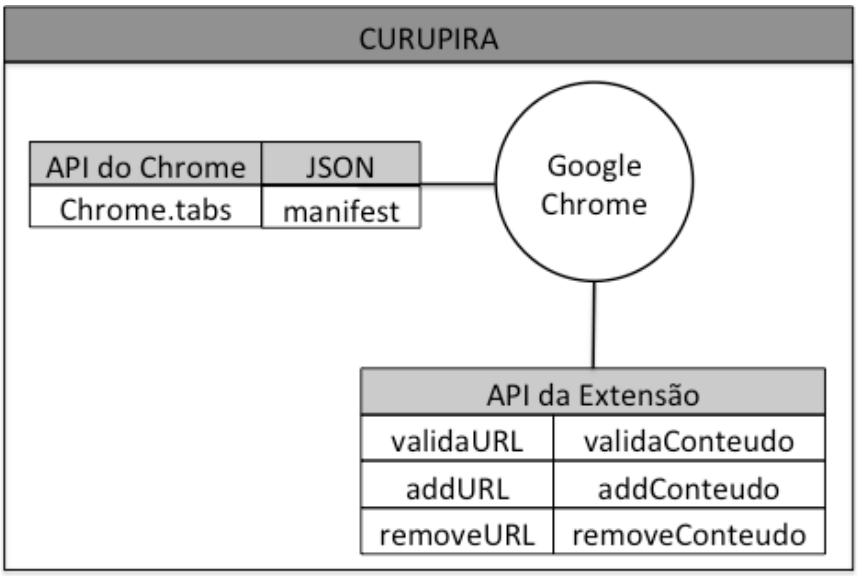



Figura 2. Visão interna do plug-in CURUPIRA.

### 4.2 Interface

A interface do CURUPIRA é responsável por efetivar as configurações desejadas e consideradas necessárias por pais ou responsáveis, a fim de evitar que crianças e

 ©2012 SBC — Soc. Bras. de Computação

adolescente tenham acesso a conteúdo Web inapropriado. Para isso, ela fornece diferentes tipos de entradas para configuração dos filtros. São elas:

- **Sugestão de filtros**: potenciais tópicos de conteúdo inapropriado são apresentados durante o processo de configuração dos filtros, tornando o uso da ferramenta mais simples e intuitiva (Figura 3). Na atual versão da ferramenta, tais tópicos são previamente estabelecidos.
- **Filtragem de URLs**: para bloquear ou filtrar o acesso à páginas da Internet, são utilizadas *blacklist*. A ideia é gravar URLs de sítios Web considerados ofensivos e/ou perigosos na *blacklist* e caso o usuário (criança e/ou adolescente) tente acessar algum dos sítios listado, ele é automaticamente bloqueado. A Figura 4 apresenta um exemplo da interface para essa função.
- **Filtragem por Palavras-Chave**: além da *blacklist* de URLs, também foram usadas expressões regulares para filtragem. As expressões regulares possuem o mesmo método de funcionamento da *blacklist*, porém não bloqueiam diretamente uma URL, mas sim palavras (expressões) encontrados na página propriamente dito, podendo ser aplicada em todo o conteúdo da pagina e também no título.

Figura 3. Sugestão de filtros na forma de tópicos.

 ©2012 SBC — Soc. Bras. de Computação

Figura 4. Lista de URLs definidas pelo usuário.

É importante ressaltar que durante o desenvolvimento da interface do CURUPIRA levou-se em consideração, como principal premissa, a facilidade de uso. A ideia sempre foi possibilitar que usuários sem qualquer experiência em segurança de computadores e/ou na configuração de aplicativos pudessem utilizar a solução de forma simples e prática, evitando um longo tempo de aprendizagem para uso e a necessidade de tutoriais extensos.

### 4.3 Segurança

No aspecto segurança, o CURUPIRA utiliza um mecanismo inicial básico (login e senha) para impedir que a ferramenta seja desativada ou tenha suas configurações alteradas por usuários indevidos.

## 5. AVALIAÇÃO E RESULTADOS INICIAIS

A avaliação da solução CURUPIRA foi dividida em duas etapas: testes de funcionalidade de interface e teste de funcionalidade dos recursos. As subseções seguintes detalham cada um desses testes e o ambiente de teste.

### 5.1 Ambiente de teste

Para avaliar a solução CURUPIRA, foi utilizada a ferramenta SIKULI (Projeto SIKULI, 2012), versão X 1.0 RC3, em conjunto com o RobotFramework (RobotFramework, 2012), versão 2.7.3. A ferramenta SIKULI foi utilizada na criação de scripts para a execução e automação dos testes e a ferramenta RobotFramework para a mesma automação, organização e criação dos casos de teste, bem como a emissão dos relatórios de resultado. É

 ©2012 SBC — Soc. Bras. de Computação

importante ressaltar que a integração entre essas ferramentas é feita utilizando as linguagens python (versão 2.7.3) e jython (versão 2.5.2).

Todos os testes de avaliação do CURUPIRA foram realizados no laboratório ETSS (Grupo de Pesquisa em Tecnologias Emergentes e Segurança de Sistemas) do Instituto de Computação (IComp) da Universidade Federal do Amazonas (UFAM). O ambiente de teste foi composto por um computador Intel Core i3 M370 2.4 Ghz, 4 Gb de RAM, 1 Gb de Vídeo, 500 Gb de HDD, executando Windows 7 64 bits.

### 5.2 Teste de funcionalidade da interface

Este tipo de teste foca exatamente no funcionamento da interface em si, no que diz respeito ao comportamento esperado de seus componentes, tais como botões, *checkboxes* e entre outros. Para os testes voltados à seção de tópicos, foram realizados oito testes individuais, um para cada tópico e dois testes coletivos. Estes testes, como dito anteriormente foram automatizados pela ferramenta SIKULI e consistem em selecionar a respectiva opção, fechar o *plugin* e em seguida abri-lo novamente, a fim de verificar se a opção continua marcada. Os testes coletivos consistem em selecionar todas os tópicos existentes, sendo que o primeiro segue o mesmo intuito dos testes individuais, ou seja, selecionar as opções e verificar se ao sair e entrar na solução CURUPIRA, as mesmas continuam marcadas. Porém, o segundo teste coletivo consiste em: ao entrar na solução com todas as opções marcadas previamente, a ferramenta as desmarca e remarca, depois sai e entra novamente no *plugin*, constatando assim o comportamento correto e adequado da interface.

Quanto às opções de filtro de URL e palavras-chave, os testes consistiriam em avaliar a entrada de caracteres especiais nos respectivos campos, contudo atualmente a solução não possui suporte a tais caracteres.

### 5.3 Teste de funcionalidade dos recursos

Enquanto os testes anteriores focam na verificação do adequado comportamento dos componentes de interface, os testes de recursos visam averiguar a correta filtragem dos sites considerados indevidos.

A solução CURUPIRA teve seu recurso de filtragem testado de duas maneiras diferentes. Na primeira, a ferramenta SIKULI configurou o plug-in, fazendo uso da opção de tópicos e palavras-chave. Dessa forma, para cada um dos oito tópicos, em conjunto com um grupo de palavras-chave (associado ao tópico da vez), a ferramenta de automação de testes acessou um grupo de 28 páginas Web, registrando devidamente quantos bloqueios corretos e incorretos acometeram os testes. Por exemplo, na verificação sobre PORNOGRAFIA, das 28 páginas inseridas, 5 foram bloqueadas corretamente (http://www.redtube.com, http://www.brazzers.com/, http://www.pornfail.com/, http://xvideos.com e http://xtube.com) e 5 não foram bloqueadas corretamente (*http://www.bestgore.com*, *http://www.ladonegro.net*, *http://suzi9mm.com/*, *http://www.adultfriendfinder.com* e *http://www.fuckbook.com/*). A principal razão para o não bloqueio reside no fato das páginas em questão não estarem listadas na *blacklist* e não possuírem em sua página principal palavras-chave relacionadas a pornografia. Por exemplo, o sítio *bestgore.com* possui fotos pornográficas, algo que a solução não está apta a detectar.

 ©2012 SBC — Soc. Bras. de Computação

No segundo tipo de teste de funcionalidade de recurso, foi testado filtro de URLs. Para isso, um conjunto de 23.400 URLs (adquiridas do firewall do Centro de Processamento de Dados da Universidade Federal do Amazonas), dividido em 3 categorias, foi inserida na solução CURUPIRA. A Tabela 1 apresenta os resultados da avaliação.

| **Categoria** | **URLs** | | |
|---|---|---|---|
| | **Disponíveis** | **Bloqueadas** | **Não Bloqueadas** |
| Pornografia | 22.333 | 21.710 | 623 |
| Violência | 1.040 | 840 | 200 |
| Proxy | 27 | 23 | 4 |

Assim como no teste anterior, a principal razão para o não bloqueio das URLs reside no fato das páginas em questão não estarem listadas na *blacklist*.

## 6. CONCLUSÃO

Este artigo apresentou uma solução de *parental control* projetada para dar maior segurança a crianças e adolescentes durante a navegação na Internet. CURUPIRA é a implementação de um prova de conceito que combina o uso de *blacklist* e expressões regulares para filtrar sítios Web com conteúdo considerado inapropriado para crianças e adolescente.

Atualmente, a ferramenta está sendo testada por um grupo de voluntários com e sem experiência na área de segurança, tendo o intuito de avaliar a satisfação com o uso da ferramenta e verificar se a experiência em segurança e/ou o uso de outras ferramentas de *parental control* influenciam na avaliação.

### 6.1. Dificuldades Encontradas

Durante o desenvolvimento da solução, um grande tempo foi gasto na curva de aprendizagem das tecnologias usadas: linguagem JavaScript e a API do Google Chrome. Outra dificuldade encontrada foi o tratamento das exceções no uso de expressões regulares, afim de evitar que termos ou expressões não fossem bloqueados indevidamente.

### 6.2. Trabalhos Futuros

Como trabalhos futuros, pretende-se:

- Melhorar a utilização dos mecanismos de *blacklist*, assim como sua empregabilidade para filtragem de conteúdos;
- Adicionar novos recursos como controle de horário.
- Propor e implementar um mecanismo semiautomático de configuração, onde os pais possam utilizar configurações predefinidas para permitir a acesso a determinados conteúdos. Para tanto, perfis de acesso a Internet baseado em faixa etária serão estabelecidos.
- Projetar e implementar um serviço Web responsável pelo gerenciamento das futuras atualizações da solução;

 ©2012 SBC — Soc. Bras. de Computação

- Validar a solução desenvolvida em dispositivos móveis.

## 7. AGRADECIMENTOS

Este trabalho teve o financiamento do CNPq através do PIBIC PIB-E/0132/2011 realizado na Universidade Federal do Amazonas.

## 8. REFERÊNCIAS

Anderson, J. (2006) "LeechBlock", http://www.proginosko.com/leechblock.html.

A.C.M. Fong, S.C. Hui, P.Y. Lee, (2009),"XFighter: an intelligent web content filtering system", Kybernetes, Vol. 38 Iss: 9 pp. 1541 - 1555

CETIC.br. (2010) "TIC Crianças 2010: Pequisa sobreo uso das Tecnologias de Informação e Comunicação no Brasil", http://www.cetic.br/tic/criancas/2010/index.htm, Outubro.

Cioffi, C.; Pagliarecci, F.; Spalazzi, L. (2009). An anomaly-based system for parental control, International Conference on High Performance Computing & Simulation, 2009. HPCS '09. Páginas 193-199, Junho 2009. doi: 10.1109/HPCSIM.2009.5194728

Corvineum. (2007) ProCon - Parental Filter Add-on for Firefox", http://procon.mozdev.org/.

Finkelhor, D.; Mitchell K.; Wolak, J. (2011) "Highlights of the Youth Internet Safety Survey", https://www.ncjrs.gov/pdffiles1/ojjdp/fs200104.pdf, March.

Glubble. (2012) "Glubble", http://www.glubble.com.

Holthe, O. (2010). Player GX: Safer Content Only Techniques. Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE , vol., no., pp.1-2, 9-12 Jan. 2010.

InspiredEffect.com. (2009) "FoxFilter", http://www.inspiredeffect.com/FoxFilter.

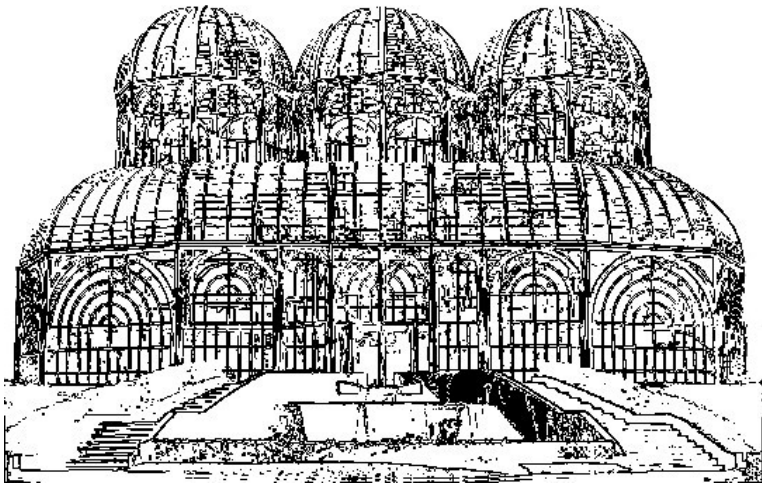
Projeto SIKULI. (2009) "SIKULI", http://sikuli.org.

RobotFramework. (2012) "RobotFramework", http://code.google.com/p/robotframework.

Safer Internet Program. (2012) "Study on benchmarking of parental control tools for the online protection of children - SIP-Bench III (SMART 2012/0044)", http://ec.europa.eu/information_society/activities/sip/docs/sip_bench2_results/report_jan11.pdf, March.

Santos, P.; Manteigas, J. (2010). Internet Segura para Crianças - Guia para Pais e Educadores. Editora Lidel – Zamboni. Páginas 248. 2010.

 ©2012 SBC — Soc. Bras. de Computação



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# WGID – Workshop de Gestão de Identidades Digitais

# Um Survey sobre Ferramentas para Single Sign-On

**Hendri Nogueira**[1], **Douglas Bayer Santos**[1], **Ricardo Felipe Custódio**[1]

[1]Laboratório de Segurança em Computação (LabSEC)
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

{jimi,douglasbs,custodio}@inf.ufsc.br

***Abstract.*** *There are many identity management tools that enable Single Sign-On (SSO) and the grouping of identity providers and services in the form of federations. This paper describes and compares some of these tools. The tools that were part of the study were: OAuth, CoSign, OpenID, OneLogin, OpenAM, JOSSO, WebAuth, CAS and BrowserID. The comparative study was the result of work done by a group of students of Computer Science, at Federal University of Santa Catarina (UFSC) in the first half of 2012. All tools were installed, configured and evaluated.*

***Resumo.*** *Existem muitas ferramentas de gestão de identidades que permitem o Single Sign-On (SSO) e o agrupamento de provedores de identidades e serviços na forma de federações. Este trabalho descreve e compara algumas dessas ferramentas. As ferramentas que fizeram parte do estudo foram: OAuth, CoSign, OpenID, OneLogin, OpenAM, JOSSO, WebAuth, CAS e BrowserID. O estudo comparativo foi resultado do trabalho feito por um grupo de alunos do curso de Ciência da Computação da Universidade Federal de Santa Catarina (UFSC) no primeiro semestre de 2012. Todas as ferramentas foram instaladas, configuradas e avaliadas.*

## 1. Introdução

O crescimento da Internet e a grande quantidade de serviços sendo oferecidos tem levado ao aumento significativo do trânsito e compartilhamento de informações. Essas informações, principalmente as pessoais (consideradas privadas), requerem um gerenciamento mais cuidadoso por parte dos usuários e serviços.

O uso de informações de forma distribuída exige uma atenção ainda maior por parte das entidades envolvidas. Os serviços que lidam com essas informações utilizam-se de diversos mecanismos e tecnologias para cuidar da privacidade e da identidade dos usuários. Tais serviços, muitas vezes, são distribuídos geograficamente e caracterizam-se por permitir o compartilhamento das informações e muitas vezes dos dados que identificam os usuários.

Um grande desafio para os usuários é o mecanismo normalmente utilizado para sua autenticação perante os sistemas de informações. Em geral, por ser mais simples e barato, os usuários são autenticados a partir de uma senha compartilhada, isto, um par *username* e senha.

No entanto, uma vez que são diversos os sistemas utilizados, os usuários acabam por repetir essas mesmas credenciais para sua autenticação, ou seja, repetem senhas (pela maior facilidade de lembrá-las).

 ©2012 SBC — Soc. Bras. de Computação

Outro desafio, é a criação do perfil do usuário. Tal perfil consiste de sua identidade. Muitas vezes isso é feito pela entrada manual de dados digitados pelo usuário e o uso de um protocolo de desafio resposta para a conta de e-mail cadastrada, com o único objetivo de verificar se o usuário sendo cadastrado detém o acesso a tal conta. Entretanto, não é raro observar-se que, seja por precaução ou por não querer divulgar dados privados, os usuários acabam criando perfis falsos (com informações que não refletem os seus reais atributos).

Para melhorar o gerenciamento das contas dos usuários (e suas informações pessoais), provedores de serviços e organizações se uniram para especificar mecanismos e métodos que permitam o compartilhamento de suas identidades através de diferentes serviços de forma segura e sem o risco de revelar sua senha ou qualquer informação confidencial de forma indevida.

E para reduzir os custos inerentes à manutenção das informações dos usuários e permitir uma interface a mais simples e natural possível entre o usuário e a máquina, seria ideal que uma vez autenticado com um determinado privilégio, a sessão do usuário e suas credenciais possam ser compartilhadas pelos mais diversos serviços. Esses mecanismos são conhecidos como *Single Sign-On* (SSO) [Shaer 1995].

SSO tem sido usado por décadas. Primeiramente foi implementado utilizando soluções proprietárias [Volchkov 2001], mas em anos recentes o SAML (*Security Assertion Markup Language*) tem se tornado o protocolo padrão para essa finalidade.

Ao longo de vários anos, tem sido proposto e implementado diversos sistemas de gerenciamento de identidades que possibilitam o SSO, e ainda, permitem o compartilhamento de credenciais pelos provedores de identidade. E algumas ainda, possibilitam o agrupamento desses provedores, na forma de uma federação. Este trabalho descreve, na forma de um survey, algumas dessas ferramentas e soluções procurando compará-las.

Este trabalho está assim organizado. *Single Sign-On* é tratado na Seção 2. Na sequência, serão descritas as ferramentas estudadas (Seção 3): OAuth, CoSign, OpenID, OneLogin, OpenAM, JOSSO, WebAuth, CAS e BrowserID. A Seção 4 realiza a análise comparativa entre as ferramentas estudadas. E a Seção 5 contém as considerações finais.

## 2. Single Sign-On

*Single Sign-On*, em sua forma mais simples, significa um mecanismo onde o usuário pode acessar vários serviços por meio da realização de sua autenticação apenas uma vez. Geralmente utiliza-se um gerenciador central de informações cadastrais de usuários. Em ambiente SSO típico, envolvem os serviços cujo recursos são protegidos, chamados de Provedores de Serviços (PS), e entidades nos quais gerenciam as informações de identidades do usuário, chamados de Provedores de Identidades (PI).

Quando um provedor de serviço requisita a autenticação do usuário, ele delega a tarefa para o provedor de identidade realizar a atual autenticação do usuário. Os métodos de autenticação podem diferenciar, desde o uso de um *username* e senha até uma solução biométrica.

Depois do provedor de identidade ter autenticado-o, ele fornece essas informações para o provedor de serviço poder proceder e entregar os serviços. O provedor de serviço

 ©2012 SBC — Soc. Bras. de Computação

pode realizar processos de autorização, solicitando ao provedor de identidade informações (atributos) extras sobre o usuário identificado.

Quando há a configuração de quais PI os PS estão habilitados a efetuar a comunicação entre si, cria-se o chamado Circulo de Confiança. Isso permite uma maior confiança nas informações informadas pelo PI e os PS que estão requisitando. Este circulo de confiança também é bastante usado no conceito de Federação [Bertino et al. 2010].

Uma vez autenticado em um serviço através do SSO, então o usuário também estará autenticado em todos os serviços do mesmo domínio. Para poder encerrar esta sessão de SSO, é necessário realizar o Single Log-Out (SLO). Nos ambientes web, isso é realizado pelo redirecionamento do usuário sequencialmente para todos os provedores de serviços necessários e passando a mensagem SAML de encerramento da sessão.

O gerenciamento centralizado das informações dos usuário requer um cuidado maior de segurança e privacidade. Riscos são evidentes mesmos se o provedor de identidade for totalmente confiável. Questões para a melhoria e alternativas relacionado à privacidade do usuário são propostas crescente na comunidade através do uso de identidades temporais, anonimato, uso de múltiplas identidades por usuário, etc.

## 3. Ferramentas e Protocolos

Nesta seção serão descritas nove ferramentas e protocolos que provêem o *Single Sign-On*: OAuth, CoSign, OpenID, OneLogin, OpenAM, JOSSO, WebAuth, CAS e BrowserID. Essas soluções são amplamente utilizadas nas comunidades acadêmicas e empresariais.

### 3.1. OAuth

OAuth é um protocolo que permite um usuário delegar a uma aplicação cliente para acessar recursos em um servidor, em nome do dono do recurso (usuário), sem que compartilhe os dados de autenticação [Youn 2011]. Ou seja, o OAuth concede acesso (delega autorização) para os recursos sem compartilhar sua identidade (ou informações privadas) para todos [Kiani 2011]. OAuth 1.0 foi publicado em 2010 via RFC 5849 [Hammer-Lahav 2010].

O protocolo OAuth introduz um terceiro fator para o modelo tradicional de autenticação cliente-servidor: o "dono do recurso". Em um modelo OAuth, o cliente (aquele que não é o dono do recurso, mas está atuando em seu interesse) requisita o acesso para os recursos controlados pelo "dono do recurso", mas armazenado pelo servidor. OAuth permite que o servidor verifique além da autorização do dono do recurso, a identidade da aplicação cliente realizando a requisição.

OAuth utiliza-se de *tokens*, *Access Token* e *Request Token*, para representar a autorização concedida ao cliente pelo usuário. O *token* contêm um subconjunto dos privilégios do usuário, e assim a aplicação cliente fica restrita ao conjunto de ações contidas no *token*.

O *Request Token* é a credencial temporária usada pelo cliente para se autenticar e obter o *Access Token* junto ao servidor. O *Access Token* é a credencial usada pelo cliente para acessar os recursos no servidor e possui um período de validade definido pelo servidor. Este *token* pode ser revogado a qualquer momento pelo usuário ou pelo servidor.

 ©2012 SBC — Soc. Bras. de Computação

Tanto *Access Token* quanto *Request Token* contém informações de autenticação do cliente e do usuário, como um *timestamp*, o método de assinatura (e.g. HMAC-SHA1), uma chave, *token* e a assinatura. As credenciais temporárias fornecidas ao cliente são de uso único e são revogadas imediatamente após seu uso.

A figura 1 demonstra o fluxo do protocolo OAuth 1.0. A obtenção do *Access Token*, ou seja, da obtenção do recurso protegido ocorre da seguinte forma:

1. O usuário acessa, através do seu navegador, um serviço da aplicação cliente. Este serviço suporta a comunicação;
2. O cliente faz uma requisição de compartilhamento com o usuário ao servidor, este responde a operação com as URL para redirecionamento do usuário e URL para autenticação da aplicação cliente e obtenção do *Access Token*;
3. O usuário é redirecionado ao servidor onde este pede sua autorização para conceder acesso ao cliente, para isto é necessário que o usuário se autentique junto ao servidor;
4. Com a permissão concedida, o servidor envia o *Request Token* para aplicação cliente;
5. Agora com o *Request Token*, o cliente se autentica no servidor e faz a requisição de um *Access Token*. Este *token* é atrelado e restrito com a permissão concedida pelo usuário e enviado ao cliente;
6. O cliente usa o *Access Token* para acessar os recursos em nome do usuário no servidor.

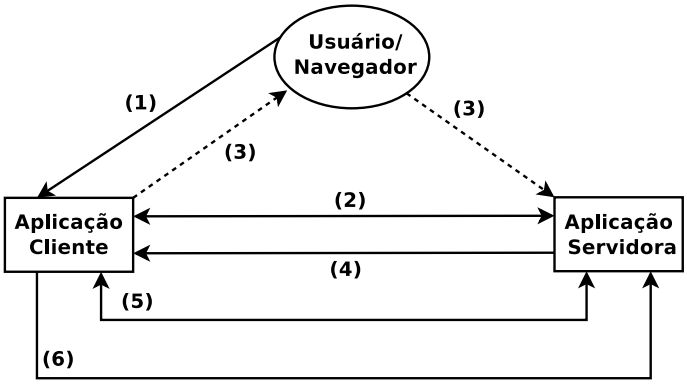



**Figura 1. Fluxo do Protocolo OAuth 1.0.**

Recentemente foi lançado um *draft* do OAuth 2.0 [Hardt 2012]. O protocolo OAuth 1.0 não é compatível com OAuth 2.0. A nova versão possui características que permite estendê-lo de acordo com as necessidades do desenvolvedor, dando estruturas de um *framework* e também a utilização em aplicações não-web. Diversos serviços utilizam partes do OAuth 2.0 e estão implementando suas próprias extensões de acordo com suas necessidades, como por exemplo o Facebook Connect [van Delft and Oostdijk 2010].

O protocolo OAuth 1.0 utiliza assinatura nas requisições e *tokens* isso garante a integridade dos dados, porém não garante confidencialidade dos dados, o que torna possível interceptar e ler a todas as informações trocadas. Como o protocolo não verifica a autenticidade do servidor é possível um ataque do homem-do-meio, interceptar e devolver

 ©2012 SBC — Soc. Bras. de Computação

mensagens errôneas ao cliente se passando pelo servidor. O desenvolvedor necessita usar mecanismos de segurança na camada de transporte.

O protocolo não especifica como as credenciais e os *token shared-secret* usados pelo protocolo devem ser armazenados, mas estes são informações sensíveis que devem ser tradados da mesma forma que *username* e senhas pelo servidor armazenando a informação cifrada. Ataques de *phishing* também são um problema devido aos redirecionamentos de sites web do protocolo. O usuário pode ser induzido a colocar sua senha em um site malicioso.

Por estes e outros motivos é recomendado o uso de TLS/SSL para todas as transações realizadas com o protocolo e qualquer outra medida de segurança que o implementador julgar necessário [Hammer-Lahav 2010]. Também recomenda-se que o servidor tenha uma política de boas práticas para os usuários e os incentivem a verificar a autenticidade dos sites web que pedem suas senhas.

### 3.2. CoSign

CoSign[1] é um projeto de código aberto, desenvolvido originalmente pela Universidade de Michigan, com estrutura de autenticação web *Single Sign-On* e é parte da *National Science Foundation Middleware Initiative*. Sua versão atual é de fevereiro de 2012.

CoSign é a abreviação de "cookie signer". Os *cookies* do CoSign não contêm qualquer informação sensível, apenas caracteres aleatórios. Então não estão no risco de serem "craqueados" ou "quebrados". Os *cookies* são válidos para a duração da sessão do navegador, não sendo armazenados no disco local.

CoSign pode ser configurado para utilizar *tickets* do tipo Kerberos[2] para usuários finais do servidor central CoSign. O comprometimento do servidor web cliente CoSign que armazena *tickets* Kerberos é mais severo do que um comprometimento do servidor que não o utiliza. Então neste caso requer um critério mínimo de segurança em sua configuração. Usuários podem se autenticar para CoSign via certificados X.509. Também possui suporte a integração com Shibboleth[3].

Atualmente, a infraestrutura é composta de apenas um único servidor CoSign central. O sistema CoSign pode ser dividido em três componentes principais: Daemon, Filtro e CGI. O Daemon central é responsável por manter os estados de todas as sessões CoSign, incluindo o rastreamento dos usuário que iniciaram ou terminaram a sessão.

O Filtro, localizado no servidor de aplicação e não na infraestrutura centralizada do CoSign, é responsável por determinar as áreas protegidas pelo CoSign. Ele assegura a autenticação do usuário, obtenção do nome, domínio de autenticação, endereço de IP e, opcionalmente, um *ticket* Kerberos, e transmite essas informações ao longo de todas as aplicações protegidas. O Filtro possui compatibilidade com o Apache.

O CGI CoSign é responsável por gerenciar a entrada e a saída de usuários do servidor central do CoSign (login e logout), por registrar cada serviço que o usuário entrar – esta ação amarra o *cookie* de login central do usuário com sua sessão em servidores individuais de aplicação, como cliente web mail, cliente de diretório web, entre outros. O

[1] `http://weblogin.org/`
[2] `http://web.mit.edu/kerberos/`
[3] `http://shibboleth.net/`

 ©2012 SBC — Soc. Bras. de Computação

CGI foi desenvolvido para utilizar o Kerberos V/GSSAPI para autenticação, e também o uso de banco MySQL, Apache BasicAuth e certificados X.509.

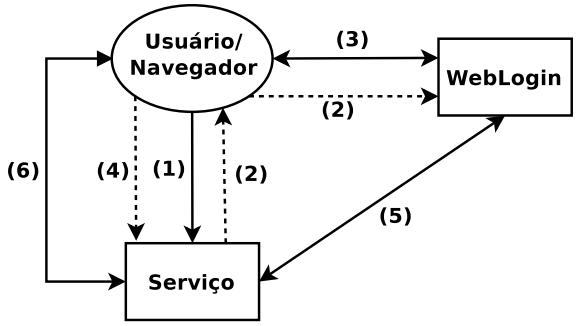



**Figura 2. Fluxo do CoSign.**

É possível através de uma única URL desconectar-se de todos os serviços protegidos pelo CoSign, ou seja, oferece suporte ao *Single Log-Out* e permite a criação de contas para visitantes. A figura 2 demonstra o fluxo do protocolo utilizado pelo CoSign e sua descrição segue abaixo:

1. O usuário acessa um serviço web[4] via navegador;
2. Caso o usuário não tenha feito sua autenticação em um período próximo, ou seja, caso não haja um *cookie* de login registrado, então o serviço redireciona o usuário para a página de autenticação weblogin[5];
3. O usuário insere suas credenciais e confirmando a autenticação, um *cookie* de login é criado, associado ao serviço;
4. O usuário é retornado para o serviço web;
5. O Filtro CoSign verifica se o *cookie* de serviço é válido;
6. Validando o *cookie* de serviço, o serviço libera os recursos para o usuário.

A segurança do sistema CoSign não é comprometida quando um servidor de aplicação sofre qualquer tipo de comprometimento. O sistema permite múltiplas formas de autenticação, podendo autenticar-se através de senha ou certificado X.509. Oferece também a autenticação por múltiplos fatores (autenticação forte). As comunicações são realizadas via SSL e o sistema controla o tempo de inatividade e o tempo limite nas sessões. Carimbo do tempo é utilizado nos *cookies*.

### 3.3. OpenID

OpenID[6] é um protocolo que fornece a autenticação e o uso de diferentes perfis do usuário. O processo é realizado sem que a aplicação necessite acessar as credenciais do usuário (como sua senha ou outras informações sensíveis).

OpenID é descentralizado, ou seja, nenhuma autoridade central deve aprovar ou registrar uma aplicação ou um provedor OpenID. O usuário poderá escolher qual provedor OpenID utilizar, e pode preservar sua identificador se mudar de provedor. Não é necessário o uso de *cookies*.

[4]e.g. `https://service.umich.edu`
[5]e.g. `https://weblogin.umich.edu`
[6]`http://openid.net/`

 ©2012 SBC — Soc. Bras. de Computação

A primeira versão, o OpenID 1.0, foi desenvolvido em 2005 e atualmente é utilizado a versão 2.0 [Foundation 2007]. OpenID 2.0 tem como objetivo fazer com que o protocolo de transferência de dados para a autenticação suporte casos de usos do método *pull* e *push*. Também possui suporte para troca de dados de perfis e mensagens de usuário para usuário [Recordon and Reed 2006].

Para suportar o serviço de descoberta e troca de dados bi-direcionais, a plataforma OpenID é baseado em endereço. Usuários são identificados (publicamente ou privadamente) por um endereço digital resolvível em uma das duas formas:

1. Uma URL OpenID (ex. http://usuario.exemplo.com.br ou http://exemplo.com.br/usuario);
2. Um XRI i-name como =exemplo.usuario (ou em forma completa, xri://=exemplo.usuario, ou em forma de resolução de proxy, http://xri.net/=exemplo.usuario).

Uma vez que o usuário possui seu endereço digital OpenID, o próximo passo é a descoberta dos serviços associados com a identidade do usuário. Isso é realizado pelo uso do protocolo de descoberta Yadis [Miller 2006] ou o protocolo de resolução XRI [Wachob and Reed 2008]. Ambos os protocolos compartilham o formato XRDS (*Extensible Resource Description Sequence*) especificado pela OASIS.

O processo de descoberta tem a intensão de ser bem simples: recuperação via HTTP ou HTTPS de um documento XRDS descrevendo os serviços disponíveis para uma URL em particular ou XRI.

Autenticação OpenID é um serviço que permite que os usuários provem que eles são os donos da URL ou *i-name*. O processo é realizado com a comunicação entre o serviço web e o provedor OpenID do usuário. O serviço descobre a URI para a extremidade do serviço de autenticação OpenID. O uso de URLs e XRIs como identificadores significa que o OpenID pode ser totalmente descentralizado.

A figura 3 mostra o fluxo protocolo de autenticação OpenID. As comunicações acontecem como descrito abaixo:

1. O usuário acessa o serviço (ex. uma página web compatível com autenticação OpenID) e inicia a autenticação apresentando sua identificação (sua URL OpenID) para o serviço requerente via usuário-agente (browser). A URL é normalizado (extrai o identificador, o provedor OpenID do usuário);
2. O serviço e o Provedor de Identidade (OpenID) estabiliza uma associação, que é uma chave secreta utilizando Diffie-Hellman Key Exchange [Rescorla 1999]. Essa chave serve para assinar mensagens subsequentes e o serviço verifica-as;
3. O serviço redireciona o usuário-agente do usuário final para o Provedor OpenID com uma requisição de autenticação OpenID;
4. Após a autenticação realizada com sucesso, o Provedor OpenID redireciona o usuário-agente de volta para o serviço com a asserção de autenticação aprovada ou falha. O serviço verifica as informações recebidas do Provedor OpenID (URL de retorno, a assinatura, etc). Autorizado o usuário, o serviço libera seus recursos.

O uso da Autenticação OpenID 2.0, usuários poderão optar pelo uso do endereço digital privado (aquele que não é identificado publicamente, mas somente em contexto

 ©2012 SBC — Soc. Bras. de Computação

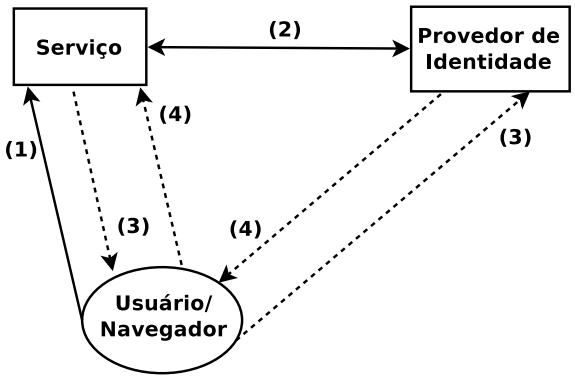



**Figura 3. Fluxo do protocolo OpenID 1.0.**

específicos do Serviço) apenas entrando com o URL ou *i-name* de seu provedor de identidade. Neste cenário, o serviço ainda utiliza a descoberta Yadis para determinar a extremidade do serviço OpenID e começa o mesmo fluxo do protocolo, mas o provedor de identidade reconhece seu próprio endereço digital e pode apresentar o usuário com a opção de selecionar um endereço digital para enviar para o serviço.

Em uma implementação ideal, o provedor de identidade deve dar a escolha ao usuário final de usar seu endereço digital público, um de múltiplas personalidades que o usuário poderia desejar em expor para o serviço, ou um gerado aleatório usado especificamente para este serviço.

O usuário pode escolher qual provedor de identidade usar. O *framework* OpenID suporta que os endereços digitais e os dados das identidades digitais sejam portáveis através dos provedores de identidade e i-brokers, e para os usuários operar seus próprio provedores de identidade se eles quiserem.

A maior diferença entre OpenID e Federação SSO é o conceito de circulo de confiança. Usualmente existe uma relação de confiança entre SP e IdP. No OpenID esta relação não existe, qualquer um pode se tornar um provedor de serviço e/ou provedor de identidade.

O OpenID possui diversas brechas [van Delft and Oostdijk 2010, Oh and Jin 2008] que podem tornar o protocolo inseguro, no qual alguns deles foram citados em sua especificação [Foundation 2007], mas nem todas possui uma solução específica e fácil. Alguns delas são: ataque de espionagem, ataque do homem do meio, *cross-site-scripting*, *phishing*, uso de HTTP e HTTPS, ataque de negação de serviço.

O maior problema de segurança do OpenID é a vulnerabilidade de *phishing*. Quando um usuário é atraído para uma *Relying Party* maliciosa e insere seu identificador OpenID, a *Relying Party* ao invés de redirecionar para o atual provedor OpenID, é realizado a busca da página no provedor OpenID e mostrado isso ao usuário. Quando o usuário insere seu *username* e senha, a *Relying Party* pode roubá-los.

O OpenID não possui suporte para *Single Log-Out* (SLO) e isso torna uma vulnerabilidade de segurança. Por não possuir a possibilidade de controlar de forma central o tempo de vida das sessões do usuário, é possível que o usuário saia ou expira a sessão

 ©2012 SBC — Soc. Bras. de Computação

no OP, mas ainda mantém aberta na RP.

Os parâmetros da resposta de autenticação são assinados, a fim de garantir o processo de autenticação e evitar adulteração. No entanto, estes parâmetros não são cifrados, o que significa que todos estes parâmetros, e em particular (provavelmente único) o identificador OpenID do usuário (openid.identity), pode ser visto por qualquer pessoa que tenha acesso a URLs completa do pedido de autenticação ou as mensagens das respostas da autenticação.

Portanto, devido a esta combinação de codificação de parâmetro URL OpenID e o campo referenciador HTTP, há uma vulnerabilidade de privacidade potencial em todos os provedores de OpenID e as partes confiantes onde a página web gerada a partir da URL de pedido/resposta da autenticação contém qualquer referência a um recurso de terceiros.

Outros problemas de segurança citado na especificação do OpenID é o envenenamento de DNS, *Cross-site scripting*, *cross-site request forgery* (CSRF) e também pode ocorrer problemas de privacidade quanto ao rastreamento dos serviços acessados.

### 3.4. OneLogin

Lançado em 2010, o OneLogin[7] objetiva uma solução de gerenciamento de identidade de fácil utilização. Possui suporte ao SSO, no qual o usuário entra no OneLogin [8] (via navegador), usando seu e-mail (como *username*) e senha, e escolhe quais serviços deseja utilizar o OneLogin. A autenticação nas aplicações é feita de forma transparente para o usuário.

O funcionamento do OneLogin começa através de um cadastro na aplicação web OneLogin. Após a realização da autenticação do usuário na aplicação OneLogin, o usuário tem acesso a diversas configurações. O usuário pode determinar quais aplicações web ele gostaria que a autenticação fosse realizada pelo OneLogin. Caso o usuário seja o administrador de algum serviço web, ele pode registrar sua aplicação (informando a URL) e gerenciar seus usuários.

A gerência dos usuários que podem acessar o serviço é feita por meio de seus cadastros (via endereço de mail), determinando os papéis que cada um pode executar e consequentemente suas permissões. Cada pessoa cadastrada pelo administrador se autentica no OneLogin usando seu e-mail e uma senha (determinado pelo administrador). O usuário pode mudar a senha, se desejar.

Segundo a organização desenvolvedora, várias aplicações se integram com o OneLogin, entre eles o WordPress, Drupal, Linkedin, Google Apps, Dropbox. Além disso, suporta diversas formas diferentes de autenticação além da tradicional *username* e senha, como certificado digital, *One-Time Password*, múltiplos fatores de autenticação, autenticação via RADIUS, SAML e ainda suporta REST API.

Um dos pontos fortes do OneLogin é permitir o gerenciamento dos usuários de forma centralizada. É possível gerar o registro das atividades dos usuários na aplicação, assim sabe-se quem está acessando certas aplicações. OneLogin é uma aplicação fechada, no qual é necessário pagar para poder usufruir de todos seus recursos. Há um plano gratuito, mas com recursos e tempo de uso limitados. Isso por ser um problema pois o seu uso

[7] `http://www.onelogin.com/`
[8] `https://app.onelogin.com/login`

 ©2012 SBC — Soc. Bras. de Computação

torna a infraestrutura amarrada a uma tecnologia apenas. Também não há conhecimento de sua implementação interna e não é expansível.

### 3.5. OpenAM

OpenAM[9] é uma ferramenta (de código aberto) para gerenciamento de autenticação centralizada, autorização e serviços federados. É um projeto originado da continuação do projeto OpenSSO da Sun, no qual foi encerrado.

Desenvolvido em Java, sua execução pode ocorrer sobre o Apache Tomcat, GlassFish, ou JBoss, e possui grandes possibilidades de customização. Sua instalação e configuração são realizados via interface gráfica web, mas também possui um modo de configuração através de comandos texto.

Essa ferramenta fornece um sistema único capaz de gerenciar toda autenticação e autorização de diversas aplicações web. Possui suporte ao uso de *smartcards* e certificados digitais, auditoria de autenticação e acesso aos serviços, *Fedlets* e OAuth. Possibilita criar um módulo próprio de autenticação (*plugins*) e sua política é baseado em XACML. Suporta autenticação federada com SAML, WS-Federation e Liberty ID-FF 1.x.

OpenAM fornece segurança na integridade, confidencialidade dos serviços web através das aplicações que suportam Liberty ID-WSF 1.x, WS-I Basic Security Profile, WS-Trust (STS) and WS-Policy.

Suportam também SOAP *Services Description Language* (WSDL) e *Representational State Transfer* (REST). É escalável horizontalmente e verticalmente para lidar com o aumento de processos e alguma mudança de segurança necessária. Sua arquitetura pode ser distribuída. Suporta diversos módulos de autenticação, com Active Directory, certificados digitais, HTTP básico, JDBC, LDAP, registro do próprio usuário, *One Time Password*, RADIUS, entre outros.

Apesar de robusto, o OpenAM pode possuir algumas dificuldades de configuração. Isso ocorre devido às inúmeras opções diferentes de configuração, tornando assim confuso e deixando brechas de segurança (e.g. configuração padrão ou incompleta, deixar ativado serviços sem necessidade). É necessário um administrador com experiência.

Caso se deseje autenticar serviços externos como Apache, Apache Tomcat, existem Agentes responsáveis por essa tarefa já implementados. Eles são instalados no local onde se deseja a autenticação. Quando o usuário faz uma requisição, verifica se esse usuário já esta autenticado, caso o mesmo esteja autenticado é entregado os recursos, caso contrário redireciona-o para a pagina de autenticação do OpenAM.

Agentes são responsáveis por autenticação externa. Quando um agente é instalado e configurado, as requisições feitas são redirecionadas e tratadas pelo agente que realiza a verificação de permissões para acessar determinado site.

O OpenAM oferece uma API em JAVA, ou seja, uma coleção de classes para permitir com que aplicações Java possam acessar diretamente o sistema OpenAM. Oferece também uma REST API, ou seja, oferece serviços web para permitir que qualquer tipo de aplicação que se comunique por HTTP e consiga interpretar os formatos JSON ou XML, acessem os serviços de autenticação e autorização do OpenAM.

[9] `http://www.forgerock.com/openam.html`

 ©2012 SBC — Soc. Bras. de Computação

Provavelmente um dos aspectos mais importantes do OpenAM é disponibilizar o controle de autenticação, logout, autorização, auditoria e gerenciamento de identidades através de uma API RESTful. Isso permite que qualquer aplicação com acesso a Internet poderá utilizar os serviços do OpenAM.

RESTful é um tipo de serviço web que, através do HTTP, disponibiliza recursos ou serviços para qualquer tipo de cliente que consiga comunicação através do HTTP. Uma aplicação móvel, por exemplo, poderá utilizar a API RESTful para controlar a autenticação e a autorização da aplicação. Isso ocorre através do uso de *tokens* de autorização, ou seja, a cada autenticação através dessa API, responderá com um *token*.

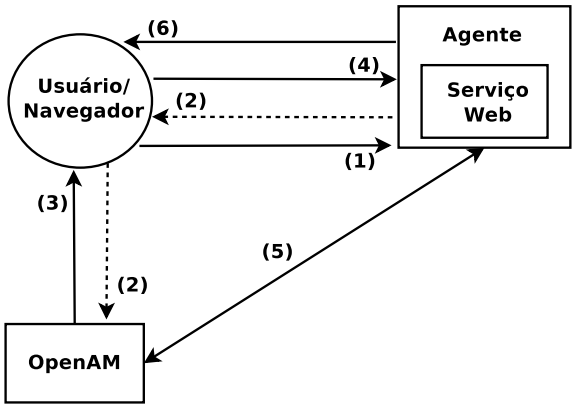



**Figura 4. Fluxo da autenticação pelo OpenAM.**

A figura 4 mostra a interação das entidades dentro do OpenAM. Seus fluxos são explicados abaixo:

1. O usuário acessa uma aplicação no qual possui um Agente instalado para proteger seus recursos;
2. O Agente redireciona o navegador do usuário para o OpenAM. O usuário insere suas credenciais para autenticação;
3. Caso autenticação realizado com sucesso, um *cookie* de sessão é criado para provar autenticação e enviado para o navegador do usuário;
4. O navegador requisita o acesso dos recursos novamente a aplicação enviando o *cookie* de sessão ao Agente;
5. O Agente verifica junto ao OpenAM a integridade, autenticidade do *cookie* do usuário. Caso o *cookie* estiver correto, é verificado as políticas de acesso referente a este usuário;
6. O Agente da aplicação aplica as questões de segurança e autorização e requisita a aplicação para a liberação dos recursos para o usuário.

### 3.6. Java Open Single Sign-On - JOSSO

*Java Open Single Sign-On* (JOSSO)[10] é um *framework* de código aberto baseado em J2EE que objetiva uma infraestrutura SSO para fornecer uma solução de plataforma centralizada de autenticação. Por ser um *framework* (escalável e extensível), permite que se implemente e combine múltiplos esquemas de autenticação.

[10] `http://www.josso.org`

 ©2012 SBC — Soc. Bras. de Computação

Desenvolvido pela Atricore[11], não é uma ferramenta totalmente gratuita. Sua versão gratuita é limitada quanto ao suporte da Atricore, mas possui uma boa documentação. Sua infraestrutura para autenticação e autorização suporta SAML para autenticação federada tanto para funções realizadas por um provedor de identidade quanto provedor de serviço. Sua integração pode ser realizada junto ao Tomcat, JBoss, Weblogic ou Windows IIS e servidor web Apache.

Atualmente o JOSSO possui duas versões. O JOSSO 1 permite uma maior flexibilidade em sua customização, e sua configuração é realizada em arquivos XML. Além disso, existe melhores chances de obter suporte gratuito a partir da grande comunidade de adeptos. O JOSSO 2 é uma abordagem "tudo-em-um", que permite a entrega ponto-a-ponto das configurações federadas SSO, com base em uma abordagem puramente orientada a modelo para diminuir a barreira de entrada e encurtamento do tempo para o valor. É mais adequado para um cenário cuja a configuração é mais padronizada e controlada.

Na infraestrutura JOSSO, o provedor de identidade é o *gateway* que fornece o SSO. É responsável por agir como uma autoridade de gestão web para o acesso SSO aos aplicativos habilitados e seus usuários. Outro elemento é o Agente SSO, que possui o objetivo de manipular os casos de uso SSO e os detalhes de execução do ambiente de integração para aplicativos SSO habilitados. Os agentes consomem serviços de identidade do *gateway*.

Atricore Console é uma ferramenta web do JOSSO 2 onde é possível configurar a infraestrutura SSO através do uso de interface gráfica. O administrador seleciona as partes que deseja configurar sua infraestrutura (provedor de identidade, provedor de serviço, aplicações compatíveis, bases de dados, etc.) e vai adicionando e criando o tipo de ligação entre cada um. Cada elemento adicionado fornece campos/informações para serem configurados.

De acordo com o site do projeto JOSSO, existe diversas empresas e organizações nacionais e internacionais que fazem o seu uso. JOSSO pode ser interessante em uso interno de uma empresa, e possui interface simples e fácil de configurar. Sua documentação é boa, e sua versão paga possui suporte direto da Atricore.

### 3.7. WebAuth

O WebAuth[12] é um sistema de *Single Sign-On* desenvolvido na universidade de Stanford, Califórnia. Como o nome diz, é desenvolvido para sistemas web e funciona na forma de um módulo para o servidor HTTP Apache, estendendo-o com a capacidade de realizar a autenticação pelo sistema WebAuth. Inicialmente sua intenção foi para ser aplicado dentro do sistema da universidade.

A estrutura do sistema é baseada em 3 componentes: a aplicação com suporte ao WebAuth, o WebKDC (*Web Key Distribution Center*) e internamente o WebLogin, além do usuário com seu navegador. Eles se comunicam para fazer o processo de autenticação, distribuição e manutenção de *tokens*.

O navegador do usuário necessita suportar *cookies* e não são necessários *plugins* Java ou JavaScripts. O navegador é redirecionado para o WebKDC como requisitado pela

[11] `http://www.atricore.com/`
[12] `http://webauth.stanford.edu/`

 ©2012 SBC — Soc. Bras. de Computação

aplicação web.

A aplicação web utiliza o WebAuth para realizar a autenticação do usuário. Caso o usuário ainda não tiver feito a autenticação (indicado pelo possessão e apresentação de um *cookie* pelo navegador), ele é redirecionado para o WebKDC. Caso já autenticado, é realizado o processo de controle de acesso do usuário. Todas as interações entre as entidades devem ser protegidas via o uso de TLS/SSL (*Transport Layer Security*/*Secure Sockets Layer*) para prevenir que os *cookies* sejam interceptados por terceiros e se passar por ele.

O WebKDC reconhece dois tipos de requisição, distintos pelas URLs. O primeiro tipo são requisições originados do navegador do usuário depois de ser redirecionado pela aplicação. Este componente do WebKDC é conhecido como o serviço WebLogin. Quando o WebKDC recebe uma dessas requisições, verifica se contêm um *cookie* com as credenciais. Se for *cookie* válido, ele é utilizado para obter novas credenciais e o usuário redirecionado para a aplicação (aplicando o SSO).

Caso o *cookie* não estiver presente ou tiver expirado, então o usuário se autentica. Depois de inserir suas informações de login para o WebKDC, o WebKDC verifica suas informações e gera dois *tokens*. Um é inserido em um *cookie* de escopo para o WebKDC fornecer o SSO para futuras requisições, e o outro é enviado para a aplicação no qual é verificado após o recebimento.

O segundo tipo de *cookie* consiste em uma mensagem XML que é enviado do WebKDC diretamente para a aplicação. Essa mensagem é utilizada para estabelecer as chaves para a aplicação ou para requisitar credenciais adicionais para o usuário. O WebKDC é também responsável por assegurar que o servidor realizando a requisição está autorizado em fazer a requisição. Por exemplo, uma aplicação não pode usar o *token* que foi criado para o uso de outra aplicação.

Os *tokens* utilizados nas comunicações do WebAuth provem um mecanismo padrão para trocar e armazenar informações que são criptograficamente seguros da adulteração e divulgação. *Token* também contêm informações de tempo de sua criação para detectar repetições, e tempo de expiração e permite a autenticação dos servidores. Eles são transferidos entre servidores utilizando URL de consultas de parâmetros, postagem de informações, *cookies* e documentos XML. Eles são cifrados com AES, pela chave privada ou chave compartilhada. Também usam SHA1 HMAC para detectar adulterações.

Os *tokens* são gerados pelo servidor Kerberos (padrão), externo ao sistema WebAuth. O WebKDC é responsável por encapsular e distribuir esses *tokens*. A ferramenta suporta outros modos de autenticação além do Kerberos. Esses modos são suportados pelo Apache.

O logout do usuário é realizado pelo fechamento do navegador do usuário, pois os *cookies* podem ter origem de mais de uma aplicação. Dessa forma, reforça-se o uso do fechamento do navegador, por ser a forma mais confiável de fazer um logout completo. Após a expiração da validade do *token* também ocorre o logout do usuário.

Ubuntu e Debian são as duas únicas distribuições que possuem o WebAuth em seus repositórios, facilitando muito a instalação da ferramenta. Possui suporte ao LDAP,

 ©2012 SBC — Soc. Bras. de Computação

é escalável (suporta conjuntos de servidores servindo o mesmo conteúdo e conjuntos de servidores weblogins e WebKDCs, todos interagindo sem quebrar a autenticação.). Todos os estados são armazenados cifrados no navegador do cliente e no servidor central.

Um dos pontos fracos do WebAuth é a implementação no Windows IIS, no qual não possui suporte direto como uma extensão. Para isso, pode-se utilizar o Shibboleth no IIS para autenticação e WebAuth como um provedor de identidade Shibboleth, mas dificulta a configuração.

### 3.8. CAS

CAS[13] (*Central Authentication Service*) é um sistema de autenticação de código aberto, originalmente criado pela universidade de Yale e atualmente é um projeto gerenciado pela Jasig. O CAS fornece um caminho seguro para uma aplicação autenticar um usuário.

O CAS permite a integração com um provedor de autenticação, gerenciamento de políticas de segurança, autorização, considerações de disponibilidade e liberação de atributos. Suporta diversos mecanismos de autenticação, tal com Active Directory, JDBC, LDAP, RADIUS, Certificados X.509, entre outros.

Desenvolvido em Java, possui facilidades de compatibilidade com diversos sistemas operacionais. Existe uma extensa documentação no qual explica como instalar, configurar o CAS, aplicá-lo nos serviços, etc. Por meio da comunidade, existe diversas aplicações no qual suportam o CAS.

No protocolo do CAS, as credenciais do usuário são passadas do navegadores para o servidor de autenticação via canal seguro (TLS/SSL). Através do *Ticket Granting Cookie* (TGC) a re-autenticação é feita de forma transparente. O TGC é emitido pelo servidor CAS para cada navegador autenticado.

Através do uso de *One-Time Service Tickets* (ST) o serviço conhece a identidade do usuário. Esses *tickets* são emitidos pelo servidor de autenticação, transmitindo para a aplicação web pelo navegador, e validado pelo servidor de autenticação. Desta forma, as aplicações ficam sabendo de nenhuma senha de autenticação.

No CAS 2.0 são utilizados *tickets* dedicados (*Proxy Granting Ticket* - GT e *Proxy Ticket* - PT) permitindo que as aplicações de terceira parte tenham certeza da identidade dos usuários, sem a grande demanda de comunicação entre navegador e aplicação dos mecanismos comum de autenticação. A figura 5 representa o fluxo da comunicação realiza no protocolo CAS. O funcionamento acontece da seguinte maneira:

1. O usuário acessa pela primeira vez um serviço web com suporte à autenticação CAS (ainda não há nenhum *Service Ticket*). Ele requisita sua autenticação e a aplicação web redireciona o usuário para o servidor CAS;
2. O usuário insere suas credenciais de autenticação (*username* e senha por exemplo). As credenciais são verificadas em um banco de dados externo e se estiverem corretas, o servidor CAS emite um *Ticket Granting Cookie* para o navegador do usuário. Este *ticket* é privado, não contém informações do usuário, apenas a validade e identificador da sessão;
3. O servidor CAS entrega para a aplicação web um *Service Ticket*. Este *ticket* é usável somente pelo serviço que o requereu;

[13] `http://www.jasig.org/cas`

 ©2012 SBC — Soc. Bras. de Computação

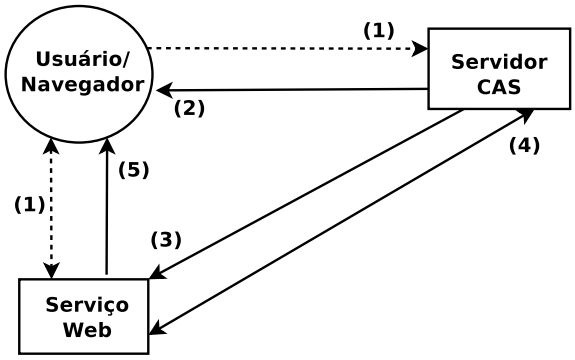



**Figura 5. Fluxo do protocolo CAS.**

4. Ao receber o *Service Ticket*, a aplicação web verifica-o enviando para o servidor CAS (o *Service Ticket* é um parâmetro CGI);
5. O *Service Ticket* é então validado e o recursos requerido pode ser entregue ao navegador do usuário.

Todos os redirecionamentos são transparentes ao usuário. O *Service Ticket* é o passaporte do navegador através do cliente CAS. É um *ticket One-Time* (não pode ser apresentado mais de uma vez para o servidor CAS), com validade somente para o cliente CAS que foi entregue, e somente para um curto período de tempo.

O CAS permite ser configurado com outros modelos e mecanismos de autenticação e autorização. Este suporte faz com que o processo se torne mais seguro e amplo, atendendo assim, uma maior variedade de modelos de autenticação. [Aubry et al. 2004] e [Naito et al. 2007] demonstram a utilização do CAS junto com outros mecanismos, como LDAP, PAM, ACLs, certificados digitais X.509. [Aubry et al. 2004] demonstra ainda, a facilidade de alguns serviços web em se tornar compatível com o protocolo CAS.

### 3.9. BrowserID

BrowserID [Callahan and Bamberg 2012] é um protocolo de autenticação desenvolvido pela Mozilla em 2011. A Mozilla desenvolveu o Mozilla Persona[14], no qual é um sistema descentralizado para serviços web e baseado no protocolo BrowserID. Através deste protocolo, torna-se possível que usuários comprovem suas identidades por meio de seus endereços de e-mail.

O Persona possui algumas características semelhantes com o OpenID e Facebook Connect. As principais diferenças são no uso de endereços de e-mail para identificadores, possui foco na privacidade, e tem a intenção de ser integrado nos navegadores. A privacidade vem da questão de que o provedor de identidade não sabe qual site o usuário é identificado.

BrowserID utiliza criptografia assimétrica e assinatura digital para permitir que os navegadores criem asserções assinadas sobre a identidade do usuário, e para os provedores de identidades atestarem (através da assinatura de um par de chaves) a identidade dos usuários de forma segura.

[14] `https://login.persona.org/`

 ©2012 SBC — Soc. Bras. de Computação

O protocolo do BrowserID envolve três entidades: o usuário, os serviços web (site web) e os provedores de identidades. Os provedores de identidades são serviços que fornecem identidades do usuário na forma de um endereço de e-mail (provedor/servidor de e-mail). Os serviços web podem desenvolver suporte ao Persona e atestar pela identidade de seus usuários.

Para que o usuário prove a posse do e-mail selecionado, um certificado assinado pelo provedor de identidade é usado para certificar a conexão entre o navegador do usuário e a identidade dada dentro do domínio do provedor de identidade.

O certificado do usuário é assinado pela chave privada do provedor de identidade e contêm o endereço de e-mail do usuário, a chave pública do usuário para aquele endereço e naquele navegador, a data no qual foi emitido o certificado, a data de expiração e o nome do domínio do provedor de identidade. A figura 6 representa o fluxo da comunicação realiza quando o protocolo BrowserID é utilizado e seu fluxo é descrito abaixo:

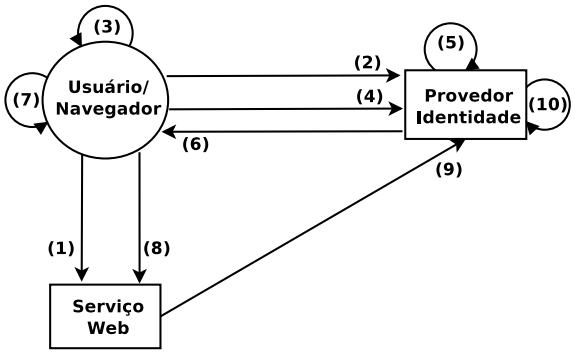



**Figura 6. Fluxo do protocolo do BrowserID.**

1. O usuário ao acessar um serviço web com suporte ao protocolo BrowserID. Ele solicita ao serviço a realização de sua autenticação;
2. O usuário é redirecionado para seu provedor de identidade escolhido. Nele, é inserido seu e-mail e senha correspondente;
3. É gerado no navegador do usuário um par de chaves e armazenado a chave privada;
4. O navegador envia sua chave pública para o provedor de identidade em questão;
5. O provedor de identidade gera e assina com sua chave privada, um certificado contendo a chave pública do usuário, seu endereço de e-mail e determina sua validade;
6. O provedor de identidade envia o certificado para o usuário;
7. Para provar a posse da chave privada, o navegador do usuário cria e assina um novo documento chamado de "asserção de identidade". Este documento contém o domínio do serviço que o usuário deseja se autenticar e o tempo de expiração da asserção (normalmente menos de cinco minutos);
8. O navegador apresenta o certificado do usuário e a asserção de identidade para o serviço verificar;
9. O serviço web busca a chave pública do provedor de identidade no documento de asserção de identidade e verifica que combina com a assinatura dentro do certificado do usuário. Se combinar, então o serviço pode ter certeza que o certificado foi realmente emitido pelo domínio em questão;

 ©2012 SBC — Soc. Bras. de Computação

10. O serviço web verifica o domínio e o tempo de expiração da asserção (previne o reuso malicioso da asserções). Depois o serviço valida a assinatura da asserção com a chave pública do certificado do usuário. Se a chave e a assinatura combinar, o serviço está seguro de que o usuário corrente realmente possui a chave associada com o certificado.

O navegador do usuário gera um par de chave diferente para cada endereço de e-mail do usuário e esses pares de chaves não são compartilhados entre navegadores. Quando um usuário seleciona uma identidade para usar no login, o navegador verifica se existe um certificado do usuário válido. Caso afirmativo, o navegador continua o processo para geração da asserção. Caso negativo, ele tenta obter um novo a partir do domínio associado com a identidade escolhida.

O navegador do usuário armazena algumas informações, como um endereço de e-mail, um par de chaves, e um certificado da associação com provedor de identidade. O par de chaves do usuário é um padrão DSA ou RSA, e é gerado e armazenado localmente no navegador do usuário. O certificado é assinado pelo provedor de identidade e associado com a identidade escolhida pelo usuário. Estes certificados tem validade de no máximo vinte e quatro horas, e pode ser reemitido até durar a sessão do usuário. As chaves não são compartilhadas nem entre computadores nem entre navegadores.

Para que um serviço web utilize o protocolo de autenticação BrowserID, é necessário que obtenha uma biblioteca do BrowserID e configure-o ao próprio serviço. Esta biblioteca funciona na últimas versões dos navegadores Opera, Firefox, Chrome, Safari e Internet Explorer.

A segurança do protocolo do BrowserID depende da segurança da chave privada do usuário, chave privada do domínio, e a conexão SSL para o provedor de identidade. A quebra de qualquer um desses pontos pode comprometer a segurança da identidade do usuário [Halpin 2012, Hackett and Hawkey 2012].

O protocolo BrowserID é desenvolvido em JavaScript e por isso é importante fazer a verificação da asserção no servidor, e não no navegador do usuário (no qual tem grande probabilidade de ser forjável). Pelo fato do protocolo envolver o navegador do usuário para realizar funções criptográficas, e não há uma maneira de saber se o navegador do usuário é totalmente confiável.

Atualmente todas as verificações envolvendo as chaves criptográficas são realizadas por um serviço centralizado[15], no qual seria mais vulnerável caso comprometido. A utilização de *username* e senhas enfraquecem o protocolo, no qual poderia utilizar outros fatores de autenticação.

É necessário que toda a comunicação envolvente e seus serviços utilizem TLS/SSL para a troca segura de mensagens. Apesar do certificados criados no navegador do usuário possuir tempo de vida curto, o mesmo pode ser reemitido de forma automática até terminar a sessão do usuário. Com isso, pode-se obter vários certificados sem quantidades limitadas. Também não é possível revogar esse certificado antes do tempo de validade. [Halpin 2012] comenta ainda sobre ataques e soluções sobre *phishing*, identificadores recicláveis, privacidade e rastreamento, robustez e usabilidade.

[15] `https://login.persona.org/`

 ©2012 SBC — Soc. Bras. de Computação

## 4. Comparação das ferramentas

A tabela 1 compara as ferramentas estudadas em relação a algumas características. Essas características são:

**Tabela 1. Tabela resumo de comparação das ferramentas estudadas.**

| Ferramentas | Custos | Mecanismos Autenticação | Integração | Instalação | Configuração | Segurança | Suporte |
|---|---|---|---|---|---|---|---|
| **OAuth** | Gratuito | Escolha do provedor OAuth | Linguagens Java, DotNET, C# JavaScript, PHP, etc | Fácil | Depende das aplicações compatíveis | Previsto e com solução | Muito Bom |
| **Cosign** | Gratuito | Multi-fator; Kerberos; LDAP; MySQL | Apache; Windows | Regular | Regular | Resolvido | Regular |
| **OpenID** | Gratuito | Escolha do provedor OpenID | Apache; C#; C++; Coldfusion; Java; PHP; etc. | Fácil | Depende das aplicações compatíveis | Fraco | Muito Bom |
| **OneLogin** | Pago | OTP; Multi-fator; RADIUS; SAML; X.509; REST API | ASP/.NET, Java, PHP, Python e Ruby; | Fácil | Fácil | - | - |
| **OpenAM** | Gratuito e Pago | Vários | Java | Fácil | Difícil | Depende das configurações | Muito Bom |
| **JOSSO** | Gratuito e Pago | SAML; X.509; LDAP; JDBC | Apache; Tomcat; PHP; Microsoft ASP; Java; etc. | Fácil | Moderado | Depende das configurações | Muito Bom |
| **WebAuth** | Gratuito | Login/Senha; Kerberos; LDAP | Apache; Shibboleth; | Fácil | Fácil | Previsto e com solução | Muito Bom |
| **CAS** | Gratuito | JDBC; LDAP; RADIUS; X.509; SAML | Apache; Java; PERL; PHP; ASP, Shiboleth; etc. | Fácil | Moderado | Previsto e com solução | Muito Bom |
| **BrowserID** | Gratuito | E-mail/senha; | PHP; JavaScript; | Fácil | Fácil | Previsto e com solução, mas fraca | Muito Bom |

**Característica 1** (Custos)**.** Informa se é de código aberto e não precisa pagar para usar, ou se é necessário pagar pelo software/serviço (de forma parcial ou completa).

**Característica 2** (Mecanismos de Autenticação)**.** Informa quais mecanismos de autenticação a ferramenta suporta, ou de forma direta (estabelecido pela ferramenta) ou de forma indireta (utilizando o mecanismo de outra aplicação em conjunta). Alguns exemplos são: *username*/senha (está implicitamente que todos aceitam, por isso não está na tabela), certificado digital X.509, SAML, LDAP, MySQL, JDBS, RADIUS, Kerberos, *One-Time Password* (OTP), Multi-fator, Active Directory, etc.

**Característica 3** (Integração)**.** Se a ferramenta permite a integração com outras tecnologias e aplicações, tais como: linguagens de programação (para implementar algum fator do provedor de identidade ou provedor de serviço), Shibboleth, Apache, Tomcat, Windows IIS, etc.

**Característica 4** (Instalação)**.** Dificuldade de instalação da ferramenta. A partir do momento em que começa a instalação sem nenhum pré-requisito e termina com a ferramenta ativada (funcionando) em sua forma mais básica.

 ©2012 SBC — Soc. Bras. de Computação

**Característica 5** (Configuração)**.** Dificuldade de se configurar a ferramenta. A partir do momento em que ela esteja funcionando da forma mais básica, até o momento que esteja funcionando de acordo com as necessidades e comunicações necessárias para uma infraestrutura de autenticação funcionar.

**Característica 6** (Segurança)**.** Está relacionado se a ferramenta possui algum problema de segurança em seu desenvolvimento.

**Característica 7** (Suporte)**.** Visa avaliar se existe (via página web dos autores, fabricantes, organizações, *wikis*, comunidades usuárias, fóruns, etc.) documentações informando sobre a ferramenta/protocolo, procedência de instalação e configuração, formas de integração com outras tecnologias, etc.

Normalmente as aplicações e/ou protocolos prevêem algumas situações onde podem ocorrer brechas ou falha de segurança na infraestrutura. Para essas situações, existem informações para resolver os problemas de segurança (previsto e com solução). Algumas ferramentas foram informadas que tiveram problemas de segurança um sua versão, mas que já foi resolvido em uma versão mais atual.

Quando um ferramenta necessita de vários detalhes para sua configuração, e necessitando de uma pessoa experiente para o mesmo, quem determina a segurança é a coerência das configurações. Existe ainda os protocolos que informam que possa haver problemas de segurança mas não determinam nenhuma solução para o mesmo, deixando para o desenvolvedor decidir. As ferramentas pagas não foi avaliado este critério por não poder testar com uma versão completa das mesmas.

A nível de instalação e configuração avaliados foram determinados sabendo que o administrador possui conhecimentos intermediários em Linux. A quantidade da documentação disponível para suporte também influencia na dificuldade de instalação e configuração.

Algumas ferramentas não puderam ser testadas em simulações em máquinas virtuais, pois suas configurações são complexas e requer um conhecimento maior de seus desenvolvimentos e também um maior tempo de estudo. A ferramenta OpenAM foi instalada, mas sua configuração não foi realizada com sucesso. O JOSSO, no qual possui duas versões JOSSO 1 e JOSSO 2, foram realizados testes bem sucedidos com o JOSSO 1, mas o JOSSO 2 faltou um melhor entendimento para a estrutura funcionar como deveria.

OpenAM possui vários mecanismos de autenticação (de acordo com sua especificação) e a listagem de todos é inviável para o espaço da tabela. As ferramentas nas quais não possuem descrição da sua segurança, se justifica por não serem gratuitas em sua totalidade. Sendo assim não foi possível encontrar informações sobre este item.

A Federação CAFe[16], gerenciado pela RNP, é uma infraestrutura de autenticação e autorização que utiliza o Shibboleth para os processos. O Shibboleth utiliza asserções SAML. De acordo com a tabela acima, as ferramentas que possui a integração com o Shibboleth já implementada, são boas candidatas para servir de alternativa para utilizar na Federação CAFe. As ferramentas que possuem suporte ao SAML, poderiam ser integrado

[16] `http://www.cafe.rnp.br`

 ©2012 SBC — Soc. Bras. de Computação

a uma infraestrutura Shibboleth, como provedor de serviço, mas é necessário verificar se não haverá conflito no processo de autenticação nativo com o do Shibboleth.

## 5. Considerações Finais

Este trabalho apresentou diversas ferramentas e sistemas de gerenciamento de identidades. Foram descritas as ferramentas OAuth, CoSign, OpenID, OneLogin, OpenAM, JOSSO, WebAuth, CAS e BrowserID. Essas ferramentas foram comparadas em termos de suas características em comum. Estas características foram definidas para auxiliar na escolha de qual melhor implementá-las.

Para ambientes cooperativos no qual não possuem, ou não querem, um setor e grupo de trabalho especializado em infraestrutura de tecnologia da informação, recomenda-se o uso das ferramentas que possuem um melhor suporte, tal como o OneLogin, OpenAM ou JOSSO. Seus suportes podem ser garantidos por meio da compra dos pacotes oferecidos.

As ferramentas JOSSO e CAS são melhores que a ferramenta OpenAM em termos de melhor facilidade de configuração, mas as três são boas quanto ao suporte de diferentes protocolos integração com outras tecnologias. Já a ferramenta OAuth tem a possibilidade de ser integrada nos serviços que requerem recursos dos usuários que estão armazenados e gerenciados por outros serviços.

As ferramentas OpenAM e JOSSO (versão 2) foram parcialmente estudas, no sentido que tivemos diversos problemas de carácter técnico para sua configuração e utilização. Já a ferramenta OneLogin teve seu estudo limitado pelo fato da necessidade da compra de um pacote para utilização de todas as suas funções.

Apesar de termos estudado e comparado nove ferramentas, existem vários outras ferramentas que devem ser estudas. Entre elas, a mais recente é a OpenID Connect [17] que é uma ferramenta baseada no OAuth 2.0 com diversas melhorias e extensões para realização da autenticação e autorização segura de usuários.

## Referências

[Aubry et al. 2004] Aubry, P., Mathieu, V., and Marchal, J. (2004). Esup-portail: open-source single sign-on with cas (central authentication service). In *EUNIS2004*.

[Bertino et al. 2010] Bertino, E., Martino, L., Paci, F., Squicciarini, A., Bertino, E., Martino, L. D., Paci, F., and Squicciarini, A. C. (2010). Digital identity management and trust negotiation. In *Security for Web Services and Service-Oriented Architectures*, pages 79–114. Springer Berlin Heidelberg. 10.1007/978-3-540-87742-4_5.

[Callahan and Bamberg 2012] Callahan, D. and Bamberg, W. (2012). Browserid protocol overview. https://developer.mozilla.org/en-US/docs/Persona/Protocol_Overview.

[Foundation 2007] Foundation, O. (2007). Openid Authentication 2.0 - Final. http://openid.net/specs/openid-authentication-2_0.html.

[Hackett and Hawkey 2012] Hackett, M. and Hawkey, K. (2012). Security, privacy and usability requirements for federated identity. *Web 2.0 Security & Privacy*.

[17] `http://openid.net/connect/`

 ©2012 SBC — Soc. Bras. de Computação

[Halpin 2012] Halpin, H. (2012). Web authentication: The next step in the evolving identity eco-system? *Web 2.0 Security & Privacy*.

[Hammer-Lahav 2010] Hammer-Lahav, E. (2010). The OAuth 1.0 Protocol. RFC 5849 (Informational).

[Hardt 2012] Hardt, D. (2012). The oauth 2.0 authorization framework draft-ietf-oauth-v2-31. Internet-Draft.

[Kiani 2011] Kiani, K. (2011). Four attacks on oauth - how to secure your oauth implementation. Four Attacks on OAuth - How to Secure Your OAuth Implementation.

[Miller 2006] Miller, J. (2006). Yadis specification 1.0. http://svn.infogrid.org/infogrid/docs/yadis/yadis-v1.0.pdf.

[Naito et al. 2007] Naito, H., Kajita, S., Hirano, Y., and Mase, K. (2007). Multiple-tiered security hierarchy for web applications using central authentication and authorization service. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, page 27.

[Oh and Jin 2008] Oh, H.-K. and Jin, S.-H. (2008). The security limitations of sso in openid. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, volume 3, pages 1608 –1611.

[Recordon and Reed 2006] Recordon, D. and Reed, D. (2006). Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, DIM '06, pages 11–16, New York, NY, USA. ACM.

[Rescorla 1999] Rescorla, E. (1999). Diffie-Hellman Key Agreement Method. RFC 2631 (Proposed Standard).

[Shaer 1995] Shaer, C. (1995). Single sign-on. *Network Security*, 1995(8):11 – 15.

[van Delft and Oostdijk 2010] van Delft, B. and Oostdijk, M. (2010). A security analysis of openid. In de Leeuw, E., Fischer-Hübner, S., and Fritsch, L., editors, *Policies and Research in Identity Management*, volume 343 of *IFIP Advances in Information and Communication Technology*, pages 73–84. Springer Boston.

[Volchkov 2001] Volchkov, A. (2001). Revisiting single sign-on: a pragmatic approach in a new context. *IT Professional*, 3(1):39 –45.

[Wachob and Reed 2008] Wachob, G. and Reed, D. (2008). Extensible resource identifier (xri) resolution version 2.0. http://docs.oasis-open.org/xri/xri-resolution/2.0/specs/cs01/xri-resolution-V2.0-cs-01.pdf.

[Youn 2011] Youn, P. (2011). Creating a safer oauth user-experience.

 ©2012 SBC — Soc. Bras. de Computação

# A Framework for Secure Single Sign-On

**Bernardo Machado David[1], Anderson C. A. Nascimento[1], Rafael Tonicelli[1], Rafael de Sousa Júnior[1]**

[1] Department of Electrical Engineering
University of Brasilia, Brazil
bernardo.david@aluno.unb.br, andclay@ene.unb.br, tonicelli@redes.unb.br, desousa@unb.br

***Abstract.*** *Single sign-on solutions allow users to sign on only once and have their identities automatically verified by each application or service they want to access afterwards. There are few practical and secure single sign-on models, even though it is of great importance to current distributed application environments. We build on proxy signature schemes to introduce the first public key cryptographic approach to single sign-on frameworks, which represents an important milestone towards the construction of provably secure single sign-on schemes. Our contribution is two-fold, providing a framework that handles both session state across multiple services and granular access control. The intrinsic centralized access control functionality adds no additional cost to the single sign on protocol while providing an easy way to manage access policies and user rights revocation. Moreover, our approach significantly improves communication complexity by eliminating any communication between services and identity providers during user identity and access permission verification. Relying on simple primitives, our methods can be easily and efficiently implemented using standard cryptography APIs and libraries. We base our constructions on standard cryptographic techniques and a threat model that captures the characteristics of current attacks and the requirements of modern applications. This is the first approach to base single sign-on security on proxy signatures.*

## 1. Introduction

The growing complexity of current corporate and internet based application ecosystems poses unprecedented access control challenges. Both the new cloud computing paradigm and Service Oriented Architecture (SOA) advocate that applications and systems be run as distributed interdependent services, which may be accessed remotely by users as they need them. Thus, many applications and systems that once ran locally have been transferred to remote distributed systems, each of them with different characteristics and access control methods. The proliferation of different remote applications and services makes it impractical for large corporations to manage each separate authentication system or for users to hold individual access credentials to each of them. This raises the need for centralized user identity management solutions that increase security while decreasing management complexity and costs.

### 1.1. Single Sign-on

Most of current application architectures require the user to memorize and utilize a different set of credentials (*e.g* username/password or tokens) for each application he/she wants to access. However, this approach is inefficient and insecure with the exponential

 ©2012 SBC — Soc. Bras. de Computação

growth in the number of applications and services a user has to access both inside corporative environments and at the Internet. Mainly, it is difficult for a corporation to manage potentially multiple authentication solutions and databases individually used by each application. Furthermore, most users tend to rely on the *same* set of credentials for accessing all of their systems, posing a serious security threat since an attacker who discovers these credentials can easily access all of the user's applications.

In a single sign-on platform, the user performs a single initial (or primary) sign-on to an *identity provider* trusted by the applications he wants to access. Later on, each time he wants to access an application, it automatically verifies that he is properly authenticated by the identity provider without requiring any direct user interaction. Single sign-on solutions eliminate the need for users to repeatedly prove their identities to different applications and hold different credentials for each applications. Furthermore, a well designed and implemented single sign-on solution significantly reduces authentication infrastructure and identity management complexity, consequently decreasing costs while increasing security.

### 1.2. The Problem: Secure Single Sign-On

Current corporate environments require employees to access multiple systems on a daily basis in order to perform their activities. Similarly, the number of personal web applications regularly accessed by millions of users grows each day (*e.g.* social networks, instant messaging, music streaming, webmail and digital content providers). Usually, users have to individually sign on to each system before being able to acces their data or perform any actions.

Each individual authentication process is commonly handled by a classical username and password credential mechanism, requiring the user to remember multiple credentials. In most cases, users tend to assign the same username/password credentials to all their user accounts in different systems. Thus, it is easy for an attacker to compromise all the other systems after obtaining credentials for only one of them. Alternatively, One Time Password (OTP) [8] methods are used, introducing an extra authentication factor which renders such attacks ineffective. However, such methods require the user to carry specific tokens or devices and contribute to increase sign on complexity.

The goal of a single sign on platform is to eliminate individual sign on procedures by centralizing user authentication and identity management at a central identity provider. Ideally, in a single sign-on solution, the user should seamlessly authenticated to his multiple user accounts (across different systems) once he proves his identity to the identity provider. Nevertheless, in many current solutions, the user is required to repeat sign on for each service using the same set of credentials, which are validated at the identity provider by each service.

For example, Google Accounts allows a user to sign on to different services provided by Google using the same username/password pair. Another infamous example is RSA SecurID [14], which a two factor authentication solution based on a OTP token and classical username/password credentials, allowing a user to sign on to several SecurID enabled services using the same token. However, a recent attack to EMC facilities exposed the overall fragility of this heuristic system. Even though their security was unaffected by current attacks, both solutions still require the user to repeatedly perform the sign on

 ©2012 SBC — Soc. Bras. de Computação

procedure.

In most of current transparent single sign-on architectures [6, 13], the user receives some kind of "authentication ticket" after he successfully signs on to the identity provider. When the user desires to sign on, he sends this ticket to the intended service provider or application, which then verifies it's validity by direct communication with the identity provider. This approach has several drawbacks, such as complex management and the requirement of secure online communication between applications and identity providers, which increases network traffic and processing loads.

### 1.3. Our Contributions

We propose a novel proxy signature based approach for single sign-on architectures that is both efficient and cryptographically sound. The proposed framework enjoys the following characteristics:

- Seamless and transparent session state maintenance and user identity verification after the initial sing-on.
- Granular access control and permissions enforcement without any additional cost.
- Easily manageable centralized access policies and user rights revocation.
- No communication required between service providers and identity providers.
- Efficient implementation based on Delegate-by-Certificate proxy signatures and standard cryptographic APIs.

The security of this framework can be reduced to the security of the underlying proxy signature scheme, assuming that the user's secret is kept secure. However, for the sake of brevity we present a detailed security analysis, leaving the formal security definitions and the complete proof to the full version of this work. To the best of our knowledge there is no other single sign-on framework based on proxy signatures and that also provides granular centralized access control. Such nice results are achieved through a clever association between recent cryptographic results on proxy signatures and the problem of delegated identity verification.

This is the first single sign on platform based on proxy signatures schemes. Although this association may seem clear, it had not been pointed out in current literature yet. Our contributions pave the way for the design of provably secure and efficient practical single sign-on architectures, providing a flexible framework for building centralized authentication solutions.

### 1.4. Roadmap

In section 2, we analyse several existing single sign-on solutions, observing their strengths and weaknesses in different scenarios. In section 3, we introduce the threat model under which our scheme's security will be analysed. In section 4, we give a brief introduction to proxy signatures and describe the framework for single sign-on. Finally, in section 5, we summarize our results and conclude with directions for future research.

## 2. Related Works

Several approaches have been proposed in current literature to address the problem of single sign-on in different scenarios. In this section we analyse classical and promising methods for achieving secure single sign-on.

 ©2012 SBC — Soc. Bras. de Computação

### 2.1. Network services single sign-on: Kerberos

Kerberos [15] was one of the first single sign-on solutions proposed in the literature and implemented as a network service. It is formally described as a network authentication system, initially designed for providing single sing-on to network services.

A Kerberos ”realm” infrastructure is composed by an *Authentication Server*, a *Ticket Granting Server* and a set of service providers. The Authentication Server is responsible for verifying the user's identity while the Ticket Granting Server generates tickets for authenticated users. The service providers are simply networked servers that authenticated users are allowed to access. The two servers act together as an identity provider, handing the user an authentication ticket that he can use to sign-on to the relying service providers. In fact, the sign-on process in Kerberos is extremely complex, requiring several interactions between the user and the servers (which can be combined into an identity provider).

Although it provides a nice practical single sign-on solution, Kerberos infrastructure management is extremely complex, being prone to several mistakes that may severely compromise security. Both the identity provider (composed by the Kerberos servers) and all the service providers must be tightly time synchronized. This rules out the utilization of Kerberos as a single sign-on framework for distributed applications that may reside in the internet or the cloud. Furthermore, Kerberos relies solely on unproven symmetric encryption mechanisms to authenticate users and maintain session state. It may also be possible to impersonate users and steal authentication tickets through simple network based attacks.

### 2.2. Web applications single sign-on: OpenID

Among the many commercial single sign-on solutions, one of the most successful is OpenID [12], which provides a framework for deploying flexible centralized user authentication for web based applications. In OpenID, the user can choose from a variety of identity providers, which may be any website or web based application where he already has an user account (*e.g.* Google). In order to sign-on to a given web based application that supports OpenID, the user first signs on to the identity provider of his choice and OpenID exchanges the necessary authentication data between the identity provider and the application.

However, it may be possible to compromise a given identity provider or the session state maintenance mechanism using simple social engineering techniques and client side or network based attacks. In order to transfer authentication information from the identity provider to relying applications, Open ID relies on a complex mechanism involving authentication information stored as cookies in the user's machine and background HTTP requests. This mechanism can be attacked through network based techniques (such as DNS poisoning) and methods based on client side website vulnerabilities (such as cross site scripting).

### 2.3. Secure login for network and web applications: Snap2Pass

The most promising approach for provably secure single sign-on in current literature is Snap2Pass, which was recently introduced in [7]. Snap2Pass allows users to sign-on to different web based or networked services using their mobile phones as credentials. In

 ©2012 SBC — Soc. Bras. de Computação

this framework, the users first shares a secret key with a service provider, storing this key in a mobile phone running a sign-on application.

Each time the user wishes to log in to the service provider, he issues an authentication request and receives a random challenge encoded as a QR-Code [9]. The user then launches the log in application and acquires the QR-Code with the mobile phone's camera. The application generates an HMAC [2] of the random challenge under the user's shared secret key and sends it back to the service provider through the internet (using 3G networks or Wi-Fi). The service provider accepts the user's sign-on if the HMAC is valid. The authors also propose a public-key based approach where a digital signature scheme is used instead of the HMAC.

Although this approach seems secure, the protocols proposed in [7] require the mobile phone to have direct online communication links to the identity provider, which increases costs and may severely affect system performance. Also, it may still be attacked by an adversary that controls the user computer or internet connection. A simple man-in-the-middle attack would consist in luring the user into log in to an arbitrary service by modifying the authentication challenge, which is not signed or verified through the HMAC. An attacker could set up a fake website or email message compelling the user to login to some trusted service (*e.g.* his internet banking website) but instead handling the user authentication challenges from other services.

### 2.4. Session state maintenance: Secure Cookie Protocol

A "Secure Cookie Protocol" for maintaining session state through authenticated cookies is presented in [10]. This protocol can be used for both maintaining session state and controlling access to different areas inside the same web service. The authors propose to check the authenticity and integrity of authentication data cookies using the HMAC. In their solution, the cookies act as a secure token that the remote service providers and applications use to verify user's identity. This solution is also the first to propose the introduction of access control information in the authentication tickets (in this case, cookies).

However, no detailed security analysis is given and it may be possible to subvert this protocol to obtain unauthorized access to applications. It cannot be used as a mechanism for maintaining session state in a global single sign-on framework, since it is based on cookies and thus only suited for web applications. Moreover, managing the symmetric keys required for the HMAC algorithm is complex.

## 3. Threat Model

It is necessary to define a threat model in order to analyse the security of a given sign-on solution. Such a model must comprise the several threats and attacks to which the solution may be submitted. First of all, we consider an infrastructure model where the user accesses services and applications from his personal computer, communicating to the identity provider and respective service providers through an Internet connection.

A threat model that captures the security requirements and threats of current internet services and application is presented in [7]. We base our security analysis on a similar threat model that considers attackers with powers to completely control all the communications links between he user's machine, the identity provider and the application/service

 ©2012 SBC — Soc. Bras. de Computação

providers. The protocol introduced in this paper is designed to be secure against the following threats:

**Phishing:** An attacker may try to lure a user into disclosing his access credentials or accidentally performing unwanted sign-on operations [16]. The attacker may set up spoofed websites and email messages or use other social engineering techniques to compel the user into performing actions that he would not normally carry out.

**Network attacks:** The attacker has complete control over the user's internet link and overall network, disrupting communication or altering data as he wants. Such attacks can be carried out by adversaries who are naturally in a privileged "gateway" position in the network (which can be achieved, for example, by infecting firewalls). Furthermore, ARP spoofing techniques may be used to divert traffic from the user's machine through the adversary's machine and back to the original destination, effectively giving control the user's link.

The adversary is also given complete control over the communication links between the identity provider and the individual service providers.

This threat model captures most of the attacks directed against current single sign-on platforms, only excluding malware and trojan horses. However, it does not make sense to design a session state and access control protocol secure against malware or trojan horses, which can assume complete control of the user's machine. Notice that, even if this protocol was secure against attackers that completely control the user's machine, the attacker could simply wait for the user to legitimately sign-on and then perform arbitrary operations on the application. Thus, it is irrelevant to consider attacks that allow the adversary to obtain unrestricted control over the user's machine.

Considering this model, for the sake of security analysis, we can assume the user's secret key (and any other sensitive information) to be securely stored in his machine. Note that it is also possible to keep user's secret keys in an external device that is able to exchange information with his personal computer, such as a mobile phone or a token. In this case it is possible to rely on the assumption that the user's mobile phone is not controled by malware, which is also relied upon in [7].

## 4. The Framework

In this section we introduce our framework for maintaining session state across the various application servers that trust the identity provider. While most single sign-on solutions keep session state by means of application servers that directly communicate to the identity provider, we introduce a novel approach where the user himself proves to each application server he wants to access that he is already authenticated by the identity provider. Moreover, our approach allows for centralized application access control by the identity provider, *i.e.* the service providers are able to verify whether the user possesses the right access permissions.

The main ingredient in our protocol is a Proxy Signature scheme, a primitive first introduced in [11]. Proxy signatures allow the owner of a private key to delegate to a

 ©2012 SBC — Soc. Bras. de Computação

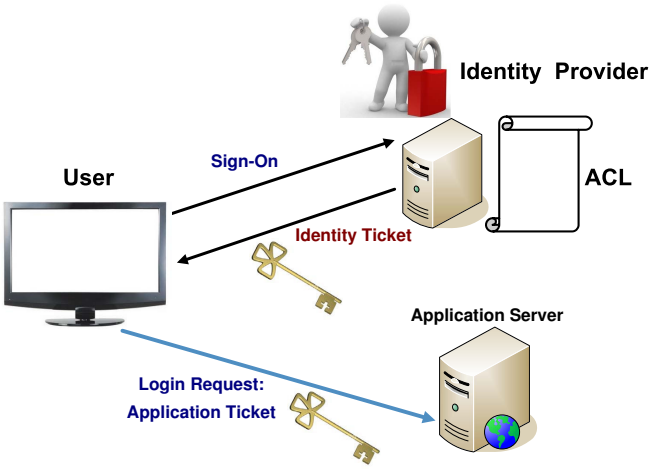



**Figure 1. Interactions between User, Identity Provider and Application Server**

third party the right to sign *specific* messages on its behalf, allowing anyone to verify whether the signature is valid. Even though they enjoy these interesting properties, proxy signatures can be realized from standard digital signatures, allowing easy and efficient implementation from current cryptographic APIs and libraries.

In this framework, each user, application server (service provider) and the identity provider are considered to have key pairs whose public keys are known to each other. The identity provider also holds an access control list (ACL) containing a description of the applications that each user is allowed to access. Considering the threat model presented in the previous session, the user's secret key is stored in his personal computer, which also runs an application responsible for computing the necessary cryptographic operations.

Basically, after a successful sign-on, the identity provider sends the user an *identity ticket*, delegating to the user the right to sign messages containing only his application access permissions (*e.g.* the names of the applications he is allowed to access). Therefore, when the user wants to access some application, he simply generates an *application ticket*, which is a message containing the appropriate access credentials on behalf of the identity provider, and sends it to the application server, who verifies whether the signature is valid. The framework is depicted in Fig. 1. Notice that no communication is takes place between the Identity Provider and the Application Server.

Different sign-on methods can be used for the initial sign-on to the identity provider, but it would be nice to couple this framework with a sign-on method that relies on the same credentials (*i.e.* key pair). For this purpose we point out that cryptographic identification protocols [3]. Such protocols enable one party to prove the possession of a secret key to another party without revealing it (*i.e.* a zero-knowledge proof of key possession). Hence, by proving the he possesses a the secret key corresponding to a pre-exchanged public key, a user can sign-on to the identity provider. Such a scenario is illustrated in Fig. 2.

We assume that the required cryptographic operations are carried out by an *Identity Broker* application running on the user's machine, since it would be unnatural to require the user to perform the signature generation and verification steps. The identity broker has access to the user's private key, which is securely stored in the user's machine

 ©2012 SBC — Soc. Bras. de Computação

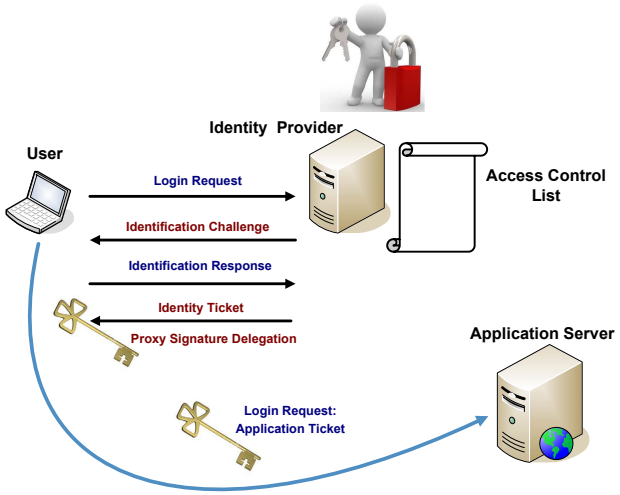



**Figure 2. Interactions between User, Identity Provider and Application Server considering and identification based sign-on**

(considering the threat model of section 3, refer to that section for further discussion). Such an identity broker could be easily implemented as a web browser plug-in or as a local proxy that captures access attempts to applications and automatically performs the necessary operations.

### 4.1. Proxy Signatures

In a proxy signature scheme, a party called the designator or original signer delegates to a party called the proxy signer the right to sign messages inside an specific message space. The basic (informal) security properties of proxy signatures stated in [1] are the following:

**Strong Unforgeability:** The original signer and third parties who are not designated as proxy signers cannot create a valid proxy signature.

**Verifiability:** From proxy signature a verifier can be convinced of the original signer's agreement on the signed message either by a self-authenticating form or by an interactive form. (The proxy signer can only sign messages inside a message space specified by the designator)

**Strong Identifiability:** Anyone can determine the identity of the corresponding proxy signer from a proxy signature.

**Strong Undeniability:** Once a proxy signer creates a valid proxy signature for an original signer, the proxy signer cannot repudiate his signature creation against anyone.

Although proxy signatures have been extensively studied, the first formal security definitions that capture the above properties and provably secure constructions were only introduced in [4]. Apart from the basic algorithm in digital signature schemes

 ©2012 SBC — Soc. Bras. de Computação

$(KeyGen(1^n),\ Sign(sk, m)$ and $Verify(pk, m, sig))$, proxy signature schemes comprise four more components:

**Delegation Algorithms:** $\mathtt{D}(pk_i, sk_i, j, pk_j, \omega)$ is run by the designator, who inputs his key pair $(pk_i, sk_i)$ along with a proxy signer ID $i$ and public key $pk_i$. It also inputs a descriptor of the delegated message space. $\mathtt{P}(pk_j, sk_j, pk_i)$ is run by the proxy signer in order to obtain the proxy signing key $skp$ and takes as input the proxy signer's key pair and the designator's public key.

**Proxy signing algorithm:** $\mathtt{PS}(skp, m)$ takes as input a proxy signing key and a message $m \in \omega$, outputting a proxy signature $psig$.

**Proxy verification algorithm:** $\mathtt{PV}(pk, psig, m)$ takes as input a public key $pk$, a proxy signature $psig$ and a message m, outputting 1 if the signature is valid for $m$ and $pk$. Otherwise, it outputs 0.

**Proxy identification algorithm:** $\mathtt{ID}(psig)$ takes as input a proxy signature and outputs the identity of the proxy signer.

The security definitions for proxy signatures are formalized in [4], and it is shown that proxy signature schemes can be obtained from any digital signature schemes. Moreover, it is shown that aggregate signature schemes [5] can be used in such constructions in order to obtain shorter signatures. One should notice that it is possible to add a timespan parameter to the proxy delegation algorithm, allowing the designator to specify a timespan during which the proxy signer may use the key. This new functionality can be added by a simple modification of the Delegate-by-certificate and Aggregate Signature Based proxy signature schemes in [4], requiring only minor (trivial) modifications in the security proofs. The altered delegation algorithm is denoted by $\mathtt{D}(pk_i, sk_i, j, pk_j, \omega, t)$, where $t$ is the timespan information.

### 4.2. The session state and access control framework

In this public key based framework, we assume that the identity provider, the user and the application servers share their public keys. However, we do not require any Certificate Authority functionalities or a full blown Public-Key Infrastructure since application and user public keys are centrally stored and controlled by the identity provider, which is already trusted by the applications and users. Also, in actual networked systems, public keys can be exchanged during user account registration. Our framework has four main components:

**ACL:** An access control list that is stored at the identity provider and contains unique IDs of the applications a user is allowed to access. It is denoted by $ACL_{uid} := \{AppID_0, \cdots, AppID_n\}$ for each user identified by $uid$.

**Identity ticket:** A proxy signature key generated by the identity provider. It del-

 ©2012 SBC — Soc. Bras. de Computação

egates signing rights of a message space representing the $ACL_{uid}$ during a session timespan $t$ to a user $uid$. It is represented as $IDt_{uid} := \mathtt{D}(pk_i, sk_i, j, pk_{uid}, ACL_{uid}, t)$, where $(pk_i, sk_i)$ is the identity provider's key pair.

**Application ticket:** A message containing a given application's unique ID and a proxy signature of this message generated by a user who has an identity ticket. It is denoted by $APPt := \langle \mathtt{PS}(IDt_{uid}, AppID_i), AppID_i \rangle$, where $AppID_i \in ACL_{uid}$.

**Identity broker:** An application that runs on the user's computer and manages identity tickets and application ticket requests.

After the user successfully signs on to the identity provider, it receives an identity ticket. The application sign-on procedure is as follows:

1. The identity broker in the user's computer stores the identity ticket and waits for application sign-on requests.
2. Upon receiving an application sign-on request from the user, the identity broker generates an application ticket and sends it to the application server.
3. The application server verifies that the application ticket is valid by running $\mathtt{PV}(pk_i, APPt)$. If the ticket is valid, it confirms that the user is already authenticated by the identity provider and that he has the proper access rights. Otherwise, it aborts.
4. The application server establishes a session with the user, relying on standard cryptographic methods for ensuring secure communications (*e.g.* establishing a secure channel or signing messages).

Once again notice that an application sign-on requests can be easily captured by watching local network traffic, enabling the identity broker to be implemented as a web browser plug-in or a local proxy.

### 4.3. Security Analysis

The security of this protocol can be proved based on standard assumptions under the threat model proposed. The basic technique is to show that any adversary who is able to sign-on to an application server without a valid identity ticket is also able to break the underlying proxy signature scheme, considering that legitimate user secret keys are safely stored in user's machines. However, for the sake of brevity and simplicity we present a concise but detailed security analysis. The full security proof will be presented in the full version of this paper.

First, remember that in the threat model for this protocol, the adversary is assumed not to control the user's machine. Notice that only the user can generate proxy signatures using his identity ticket, since it is necessary to use his secret key (which is safely stored in his computer) for proxy signing. Hence, even if an attacker sniffs the network and captures the identity ticket, it remains useless. The scheme is also secure against replay attacks, since current signature schemes [5] output fairly random signatures that also work as nonces in our scheme, allowing applications to detect whether they have received the same application ticket more than once.

Now we analyze this approach according to the proxy signature's security properties. By the Strong Unforgeability property, an attacker cannot forge an application ticket

 ©2012 SBC — Soc. Bras. de Computação

on behalf of the identity provider in order to obtain access to applications. By the Verifiability property, the application server can verify that the user has the necessary access rights (*i.e.* he can really sign the message containing the application's ID on behalf of the identity provider). The Strong Identifiability property allows the application to obtain the user's identity (possibly using it for further fine grained access control). The Strong Undeniability property adds a nice benefit for digital forensic analysis, since it is possible to determine that a user has really accessed an application.

## 5. Conclusion

We presented a new approach for practical efficient and secure single sign-on frameworks based on proxy signature schemes. The proposed framework provides seamless and transparent single sign-on without undermining overall network security and without requiring *any* online communications between service providers and the identity provider. Additionally, it allows for fine grained access control without any increase in the protocol's computational or communication complexity. Moreover, it offers simple access policy and user revocation management while providing nice forensic and audit data by building on common proxy signature strong unforgeability and undeniability properties. Our framework is also the first to apply public key cryptography techniques to the problem of practical single sign-on. We remark that our approach is the first to associate proxy signature schemes to this practical problem, even though the association seems straightforward. These results represent an important step towards the formalization of single sign-on and user authentication protocols, and the construction of provably secure schemes for these practical applications.

As a future work, the integration of our approach with other cryptographic protocols remains to be studied. It is important to provide a formal analysis and security proof for this framework. Another promising research direction is obtaining efficient and secure proxy signature key revocation protocols in order to implement better single sign-off mechanisms. It is still an open problem to obtain a session state and access control protocol that remains secure if the adversary is given control of the user's computer. A promising direction for solving this issue is to adapt our protocol to rely on an external tamper-proof token or mobile device to run the identity broker.

## References

[1] B. Lee, H.K., Kim, K.: Strong proxy signature and its applications. In: Proc. of the 2001 Symposium on Cryptography and Information Security (SCIS'01). vol. 2, pp. 603–608 (2001)

[2] Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology. pp. 1–15. CRYPTO '96, Springer-Verlag, London, UK (1996), http://portal.acm.org/citation.cfm?id=646761.706031

[3] Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification protocols secure against reset attacks. In: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology. pp. 495–511. EUROCRYPT '01, Springer-Verlag, London, UK (2001), http://portal.acm.org/citation.cfm?id=647086.715697

[4] Boldyreva, A., Palacio, A., Warinschi, B.: Secure proxy signature schemes for delegation of signing rights. Journal of Cryptology pp. 1–59 (2010), http://dx.doi.org/10.1007/s00145-010-9082-x, 10.1007/s00145-010-9082-x

 ©2012 SBC — Soc. Bras. de Computação

[5] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques. pp. 416–432. EUROCRYPT'03, Springer-Verlag, Berlin, Heidelberg (2003), http://portal.acm.org/citation.cfm?id=1766171.1766207

[6] De Clercq, J.: Single sign-on architectures. In: Davida, G., Frankel, Y., Rees, O. (eds.) Infrastructure Security, Lecture Notes in Computer Science, vol. 2437, pp. 40–58. Springer Berlin / Heidelberg (2002)

[7] Dodson, B., Sengupta, D., Boneh, D., S., L.M.: Secure, consumer-friendly web authentication and payments with a phone. In: Proceedings of the Second International ICST Conference on Mobile Computing, Applications, and Services (MobiCASE) (2010)

[8] Haller, N., Metz, C., Nesser, P., Straw, M.: A One-Time Password System. No. 2289 in Request for Comments, Internet Engineering Task Force, IETF (Feb 1998), http://www.ietf.org/rfc/rfc2289.txt

[9] ISO 18004:2005: Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification Automatic. ISO, Geneva, Switzerland

[10] Liu, A., Kovacs, J., Huang, C.T., Gouda, M.: A secure cookie protocol. In: Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on. pp. 333 – 338 (oct 2005)

[11] Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: Proceedings of the 3rd ACM conference on Computer and communications security. pp. 48–57. CCS '96, ACM, New York, NY, USA (1996), http://doi.acm.org/10.1145/238168.238185

[12] OpenID: www.openid.net

[13] Pashalidis, A., Mitchell, C.: A taxonomy of single sign-on systems. In: Safavi-Naini, R., Seberry, J. (eds.) Information Security and Privacy, Lecture Notes in Computer Science, vol. 2727, pp. 219–219. Springer Berlin / Heidelberg (2003)

[14] SecurID, R.: http://www.rsa.com/node.aspx?id=1156

[15] Steiner, J.G., Neuman, C., Schiller, J.I.: Kerberos: An authentication service for open network systems. In: in Usenix Conference Proceedings. pp. 191–202 (1988)

[16] Yu, W.D., Nargundkar, S., Tiruthani, N.: A phishing vulnerability analysis of web based systems. In: IEEE Symposium on Computers and Communications (ISCC 2008). pp. 326–331. IEEE Computer Society, Marrakech (Jul 2008), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4625681

 ©2012 SBC — Soc. Bras. de Computação

# Geração de Certificados Digitais a partir da Autenticação Federada Shibboleth*

**Michelle S. Wangham[1], Emerson Ribeiro de Mello[2], Davi da Silva Böger[3], Joni da Silva Fraga[3], Marlon Candido Guérios[4]**

[1] Grupo de Sistemas Embarcados e Distribuídos–GSED/CTTMAR
Universidade do Vale do Itajaí (UNIVALI) – São José – SC – Brasil

[2]Instituto Federal de Santa Catarina – São José – SC –Brasil

[3]Departamento de Automação e Sistemas (DAS)
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

[4]Inohaus Consultoria e Desenvolvimento de Sistemas- Florianópolis – SC– Brasil

wangham@univali.br, mello@ifsc.edu.br
{dsboger,fraga}@das.ufsc.br, marlonguerios@inohaus.com.br

***Resumo.*** *O framework Shibboleth é a infraestrutura de autenticação e autorização mais empregada para constituição de federações acadêmicas, possibilitando que usuários, através de um navegador web, acessem serviços disponibilizados pela federação usufruindo do conceito de autenticação única. O Shibboleth faz uso do padrão SAML, porém, nem todas as aplicações usadas pela comunidade acadêmica operam com credenciais SAML ou não são web. Diversos projetos surgiram para interligar o Shibboleth com a tecnologia de autenticação comumente usada em grids, as credencias X.509, permitindo assim a conversão de credenciais SAML em certificados digitais. O presente trabalho descreve e compara duas abordagens para geração de certificados X.509, a partir da autenticação federada Shibboleth. Uma abordagem está vinculada ao provedor de serviços e a outra ao provedor de identidades. Por fim, as abordagens propostas são comparadas com os projetos relacionados.*

***Abstract.*** *Shibboleth framework is an authentication and authorization infrastructure widely adopted by academic federations. It offers a way for users to access multiple services with a federated single sign-on framework. Shibboleth makes use of SAML standard, however some academic distributed applications do not work with SAML credentials or they are not web applications. There are several projects to mapping Shibboleth SAML credentials in X.509 credentials, authentication technology commonly used in grid. This paper describes and compares two approaches to generate X.509 certificates from the Shibboleth federated authentication. The first approach is linked to service providers and the other one to the identity provider. Finally, these approaches are compared with related projects.*

*Desenvolvido dentro do escopo do projeto "Serviços para Transposição de Credenciais de Autenticação Federadas", GT-STCFed, financiado pela RNP.

 ©2012 SBC — Soc. Bras. de Computação

## 1. Introdução

Nas atuais redes nacionais de ensino e pesquisa (*National Research and Education Network* - **NREN**)[1], a crescente necessidade de compartilhar recursos e serviços para usuários de diferentes instituições que possuam algum tipo de relação de confiança motivaram a constituição de **federações acadêmicas** [TERENA 2012]. Uma federação é uma forma de associação de parceiros de uma rede colaborativa que usa um conjunto comum de atributos, práticas e políticas para trocar informações e compartilhar serviços de forma segura, possibilitando a cooperação e transações entre os usuários da federação [Carmody et al. 2005]. Estas federações agrupam pessoas do meio acadêmico que vão desde alunos, técnicos administrativos e professores. O conceito de federação acadêmica visa minimizar as demandas dos provedores e dos usuários de serviços disponibilizados por instituições de ensino e pesquisa no que diz respeito à manutenção de informações usadas para autenticação e autorização de acesso a esses serviços [Moreira et al. 2011].

A noção de federação é construída a partir do gerenciamento de identidades obtido com o uso de uma Infraestrutura de Autenticação e Autorização (AAI). No contexto das NRENs, o *framework Shibboleth*[2] é a infraestrutura mais empregada para constituição de federações acadêmicas. As federações *Incommon*, da rede norte-americana *Internet2*, e a *CAFe*, da Rede Nacional de Ensino e Pesquisa (RNP) do Brasil, são exemplos de federações construídas tendo como base este *framework*.

O projeto *Shibboleth* começou em 2000, como uma iniciativa da *Internet2*, com o intuito de permitir que usuários de instituições acadêmicas pudessem interagir com serviços providos por outras instituições, bastando que estas façam parte de uma mesma federação acadêmica. Em uma federação acadêmica, uma vez autenticado em sua instituição de origem, um usuário pode acessar, através de um navegador *web*, qualquer serviço da federação sem novas autenticações, caracterizando o que é chamado de autenticação única *web* (*Web Single Sign-On* - Web SSO). O *framework Shibboleth* está baseado em padrões abertos, principal no SAML (*Security Assertion Markup Language*) da OASIS, e fornece um pacote de software, de código aberto, para a construção de aplicações *web* federadas [Internet2 2012].

Dentro de um domínio *Shibboleth* existem dois papéis: provedor de identidades (*Identity Provider* – IdP) e provedor de serviços (*Service Provider* – SP). O primeiro é responsável por autenticar seus usuários, antes que estes possam usufruir dos serviços oferecidos pelo segundo [Shibboleth 2005]. Os IdPs são responsáveis por manter as informações sobre as pessoas vinculadas a uma instituição, incluindo dados pessoais (nome, data de nascimento, CPF, nomes dos pais, sexo, data de nascimento etc.) e vínculos internos (data de admissão, cargo ocupado, número de matrícula, número VoIP etc.). Um IdP estabelece seu método de autenticação interno e deve garantir que cada pessoa da instituição tenha um identificador único. Os SPs oferecem serviços de acesso restrito, podendo requisitar informações adicionais sobre os usuários para garantir o acesso a um determinado recurso, por exemplo, uma página *web* para realizar renovação de empréstimos de livros. Na implementação do serviço, são definidos os privilégios de

[1] São provedores de serviço de Internet especializados para comunidade de ensino e pesquisa dentro de um país que oferecem serviços de conectividade avançados e canais dedicados para o desenvolvimento de projetos de pesquisa e de capacitação [TERENA 2012]

[2] `http://shibboleth.net`

 ©2012 SBC — Soc. Bras. de Computação

acesso e as informações adicionais que serão solicitadas. Não cabe ao SP manter essas informações dos usuários, mas apenas solicitá-las aos IdPs [Moreira et al. 2011].

Apesar das relações de confiança garantirem que as asserções de segurança emitidas pelos provedores de identidades serão consideradas válidas pelos provedores de serviços, essas não indicam a robustez do processo de autenticação pelo qual o usuário passou junto ao provedor de identidades. A Federação *InCommon* possui um programa para avaliação do nível de garantia (LoA - *Level of Assurance*) das identidades utilizadas na federação. O *framework* para avaliação do nível de garantia (*Identity Assurance Assessment Framework*)[Federation 2011], define dois perfis - o prata e o bronze - para que as instituições da federação possam aderir a fim de proporcionar maior nível de garantia aos provedores de serviço. Os perfis definem os requisitos específicos que operadores IdPs)devem cumprir para poderem incluir qualificadores de garantia de identidade *InCommon* nas asserções de identidades que estes oferecem aos provedores de serviço.

Com o intuito de encorajar que mais usuários, em especial acadêmicos vinculados às redes nacionais de pesquisa e educação (NREN), utilizem os serviços de um *Grid* ou de uma *cyberinfrastructure*[3], diversos projetos surgiram tendo como proposta interligar o *framework Shibboleth* com o modelo de segurança comumente utilizado em *grid services* e em *cyberinfrastructure*. Entre estes, destacam-se os projetos da *Joint Information Systems Committee* (JISC) - *ShibGrid*, *SHEBANGS* e *SARoNGS*, e os projetos da *National Science Foundation* (NSF) - *GridShib*, *CILogon* e *go.teragrid*. O objetivo destes projetos é conceber uma ponte entre as credenciais SAML geradas no processo de autenticação em um IdP da Federação *Shibboleth* e os certificados X.509 necessários para autenticação de pesquisadores que desejem usufruir de um *grid* ou de uma *cyberinfrastructure*.

Este artigo tem por objetivo descrever e comparar duas abordagens para geração de certificados digitais X.509, a partir da autenticação federada *Shibboleth*. A primeira abordagem permite que usuários da federação, através de um navegador *web*, acessem um serviço confiável (*SP Shibboleth*) para obter um certificado digital utilizando suas credenciais institucionais (identidade federada). Já a segunda abordagem, possibilita que usuários, através de uma aplicação não *web*, se autentiquem em provedores de identidades estendidos, chamados *IdP+*, para obterem suas credenciais de autenticação traduzidas para certificados X.509. Para as duas abordagens, os certificados são emitidos por Autoridades Certificadoras on-line e possuem curta duração de vida (*short lived credentials*), conforme os requisitos do perfil definido em [TAGPMA 2009].

O artigo está organizado da seguinte forma. Na Seção 2, são descritas as soluções utilizadas para implantação de autoridades certificadoras on-line, responsáveis pela emissão dos certificados digitais. Na Seção 3, as duas abordagens propostas para geração de certificados são descritas e comparadas, bem como a apresentação dos resultados obtidos com a implementação destas abordagens. Na Seção 4, os projetos relacionados são descritos e comparados com as abordagens desenvolvidas. Por fim, na Seção 5, são apresentadas as conclusões.

[3]Termo usado pela *National Science Foundation (NFS)* para representar ambientes de pesquisa (recursos) que proveem suporte, armazenamento, gerenciamento e visualização de dados avançados, tais como: supercomputadores, sistemas de alta capacidade de armazenamento em massa, ferramentas de visualização interativa escalável, repositórios de dados em larga escala, sistemas de gerenciamento de dados científicos digitalizados, redes de várias granularidades, etc.

 ©2012 SBC — Soc. Bras. de Computação

## 2. Sistemas de Gerenciamento de Certificados Digitais

Um Sistema de Gerenciamento de Certificados é um software responsável pela gestão do ciclo de vida de certificados digitais, incluindo a emissão e revogação de certificados. Nas duas abordagens para geração de certificados desenvolvidas neste trabalho, pedidos de emissão de certificados (*Certificate Signing Request* - CSR) devem ser encaminhados, de forma segura, para uma autoridade certificadora on-line para que esta possa emitir os certificados dos usuários da federação Shibboleth. Duas autoridades certificadoras podem ser utilizadas, uma concebida a partir do Sistema de Gerenciamento de Certificados Digitais (*SGCI*)[4] do projeto ICPEdu [RNP 2012] e a outra a partir do *MyProxy*[5].

O sistema de gerenciamento do *MyProxy* foi o escolhido devido ao seu amplo uso na comunidade de grid e em *cyberinfrastructure*. Tanto os projetos do JISC quanto os da NSF, citados anteriormente, adotam o *MyProxy*. Possibilitar que os certificados possam também ser emitidos por uma autoridade certificadora da cadeia de certificação do serviço ICPEdu se mostrou muito interessante para a RNP, pois algumas de suas instituições usuárias fazem uso deste serviço, inclusive a própria RNP.

### 2.1. *MyProxy*

***MyProxy***[6] é um software de código aberto para o gerenciamento de credenciais de segurança X.509 PKI (*Public Key Infrastructure*) que provê um repositório de credenciais on-line e uma autoridade certificadora on-line, possibilitando aos usuários obterem credenciais de forma segura, quando e onde for necessário. As funcionalidades de repositório de credenciais e de autoridade certificadora podem ser combinadas em um serviço ou podem ser usadas separadamente. Usuários executam o comando `myproxy-logon` para se autenticar e obter credenciais, incluindo certificados de autoridades certificadoras (ACs) confiáveis e listas de certificados revogados (*Certificate Revocation list* - CRLs) [MyProxy 2012].

O uso do repositório de credenciais *MyProxy* permite que usuários obtenham *proxy credentials* (RFC 3820[7]), sem que estes precisem se preocupar com a gestão da chave privada e com o arquivo do certificado. Os usuários podem usar o *MyProxy* para delegar credenciais aos serviços que atuam em seu nome, por exemplo, um portal de grid, armazenando as credenciais no repositório *MyProxy* e enviando a senha de acesso do *MyProxy* para o serviço. Os usuários podem ainda usar o *MyProxy* para renovar suas credenciais, por exemplo, quando *jobs* de longa duração são utilizados. O gerenciamento profissional de servidores *MyProxy* podem oferecer um ambiente de armazenamento de chaves privadas mais seguro que os sistemas usados pelos usuários finais. O *MyProxy* pode ser configurado para criptografar todas as chaves privadas armazenadas no repositório com senhas escolhidas pelo usuário, porém, de acordo com políticas de qualidade de senhas aplicadas pelos servidores. Usando o protocolo de delegação de *proxy credential*, o *MyProxy* permite aos usuários obter credenciais sem transferir as chaves privadas destes através da rede [MyProxy 2012].

[4] `https://projetos.labsec.ufsc.br/sgci`
[5] `http://grid.ncsa.illinois.edu/myproxy/`
[6] Em Julho de 2012, foi lançada a versão *MyProxy* 5.9
[7] *X.509 Proxy Certificate Profile [Tuecke et al. 2004].*

 ©2012 SBC — Soc. Bras. de Computação

Para os usuários que ainda não tem credenciais X.509, a Autoridade Certificadora *MyProxy* (*MyProxy CA*) fornece um método conveniente e seguro para obtê-las. A *MyProxyCA* emite certificados de vida curta[8], assinados com a chave da autoridade certificadora configurada, somente diante de pedidos recebidos de usuários autenticados (*Certificate Signing Request* - CSR). Usuários executam o comando `myproxy-logon` para se autenticar e obter as credenciais da *MyProxy CA*, sem necessitar armazenar os certificados e chaves de vida longa no repositório de credenciais ou em qualquer lugar. A *MyProxy CA* atende aos requisitos do perfil *Short Lived Credential Services X.509 Public Key Certification Authorities* [TAGPMA 2009] da autoridade de gerenciamento de políticas de Grid das Américas (*The Americas Grid Policy Management Authority*), membro da *International Grid Trust Federation* (IGTF) [MyProxy 2012].

Um conjunto de mecanismos de autenticação são suportados no *MyProxy*, dentre estes destacam-se: senha (*passphrase*), Kerberos, SAML, OpenID, *Virtual Organization Membership Service* (VOMS), *Simple Authentication and Security Layer* (SASL), *Pluggable Authentication Modules* (PAM), *Remote Authentication Dial In User Service* (RADIUS), *Moonshot* e *One Time Passwords* (OTP). Mecanismos e políticas de autorização implantados no servidor *MyProxy* podem controlar o acesso às credenciais e como estas podem ser usadas [MyProxy 2012].

A funcionalidade da *MyProxy CA* requer a configuração do PAM e/ou do SASL para suportar autenticação baseada em senha ou via Kerberos para obter certificados. Via PAM, o *MyProxy* pode ser configurado para usar um mecanismo de autenticação externo - como contas locais, um servidor LDAP remoto, ou senhas OTP - em vez de usar credenciais criptografadas por senhas para verificar a identidade. Via SASL, o *MyProxy* pode suportar uma variedade maior de protocolos de autenticação.

### 2.2. SGCI

O Sistema de Gerenciamento de Certificados Digitais (**SGCI**) [LabSEC 2012] da Infraestrutura de Chaves Públicas para Ensino e Pesquisa (ICPedu) [RNP 2012] é um software desenvolvido para o âmbito acadêmico, em uso em diversas universidades e centros de pesquisa brasileiros, que permite a implantação e o gerenciamento de Autoridades Certificadoras (AC) para emissão de certificados digitais, provendo as funcionalidades necessárias para o gerenciamento de Infraestruturas de Chaves Públicas (ICPs).

Com o SGCI é possível criar Infraestruturas de Chaves Públicas (ICP) completas, com AC raiz, ACs intermediárias e Autoridades de Registro (AR). Uma AR é responsável por verificar informações de identidade dos usuários antes de enviar um pedido de emissão de certificado para a AC. Se há uma relação de confiança entre a AC e a AR, então a AC pode emitir o certificado para o usuário conforme o pedido da AR [Silvério et al. 2011].

Na versão 2.0.0[9], o SGCI suporta ACs on-line para envio automático de pedidos de emissão de certificados. Uma AC on-line pode ser de dois tipos: de resposta manual e de resposta automática. ACs on-line de resposta manual exigem que um operador verifique a emissão de certificados manualmente. ACs on-line de resposta automática emitem certificados para quaisquer requisições vindas de ARs de confiança. A comunicação entre uma

[8]Por padrão, a *MyProxy CA* emite certificados com validade de doze horas, porém, este parâmetro pode ser configurado no servidor.

[9]Versão Beta.

 ©2012 SBC — Soc. Bras. de Computação

AR e uma AC é padronizada pelo Protocolo de Gerenciamento de Certificados (*Certificate Management Protocol* - CMP) especificado na RFC 4210 [Adams et al. 2005]. Esse protocolo estabelece formatos e mecanismos de transporte para mensagens de emissão e revogação de certificados, bem como emissão de LRCs [Silvério et al. 2011].

As mensagens de requisição e resposta de emissão de certificado entre ARs e ACs são especificadas na RFC4210 por um esquema ASN.1. O SGCI 2.0.0 oferece suporte a um subconjunto do CMP, utilizando como codificação para as mensagens as regras de codificação em XML da ASN.1 (XML Encoding Rules - XER) e o transporte das mensagens CMP é feito sobre o protocolo HTTP (ou HTTPS) [Silvério et al. 2011].

## 3. Abordagens para Geração de Certificados Digitais

No *framework Shibboleth* [Shibboleth 2005], a troca informações de autenticação de usuários entre provedores de identidades e provedores de serviços é feita através de asserções SAML. Além disto, o *Shibboleth* foi concebido exclusivamente para aplicações *web*, de forma que usuários acessam os recursos, dispostos em servidores *web* (provedores de serviço), através de um navegador *web*.

O fato de algumas aplicações usadas pela comunidade acadêmica, como por exemplo *grids*, não serem *web* ou por necessitarem de credenciais de segurança em um formato diferente do SAML, por exemplo certificados X.509, não garantem aos usuários todas as facilidades trazidas com o modelo de gerenciamento de identidades federadas, cuja principal implementação é o *Shibboleth*. Neste trabalho, são descritas duas abordagens desenvolvidas no contexto do projeto GT-STCFed da RNP que permitem que estas aplicações possam usufruir da autenticação federada do *Shibboleth*.

### 3.1. Serviço Gerador de Certificados Digitais (SGC)

Em alguns *grids* e *cyberinfrastructure*, o usuário interage através de clientes baseados em linha de comando, como por exemplo, GSI-OpenSSH, GridFTP, GRAM (*Globus resource allocation manager*) e Condor-G. No caso de acesso por console remoto, o usuário deverá fazer uso de um cliente SSH, sendo que o *grid* exigirá uma credencial de autenticação para garantir o acesso deste usuário ao console. Por exemplo, um nome de usuário e senha ou um certificado digital X.509 de curta duração emitido por uma Autoridade Certificadora específica. O uso de credencias X.509 é a forma mais recomendada e utilizada para autenticação de pesquisadores que desejam usufruir de grids e de *cyberinfrastructure* [Basney et al. 2010, Barnett et al. 2011].

A geração de certificados digitais de curta duração em ambientes dinâmicos e de larga escala, conforme o perfil definido em [TAGPMA 2009] só consegue ser operacionalizada através Autoridades Certificadoras On-line (AC), como exposto na Seção 2. A AC pode possuir uma base de usuários, para quem emitirá certificados, ou a AC pode fazer parte de uma federação *Shibboleth* e confiar nas informações fornecidas pelos provedores de identidades desta federação. A segunda opção é ideal para ambientes dinâmicos e de larga escala e traz mais benefícios para os usuários, pois estes não precisam realizar e manter cadastros individuais em cada AC (gerenciar múltiplas identidades).

Segundo [Barnett et al. 2011], de forma a contribuir com a interoperabilidade e usabilidade, os grids e *cyberinfrastructure* de organizações nacionais, estaduais e acadêmicas deveriam adotar um único e aberto sistema de gestão de identidades para

 ©2012 SBC — Soc. Bras. de Computação

prover a autenticação de seus usuários, sendo, neste documento, especificamente recomendado o uso do *framework Shibboleth*. Com a adoção de identidades federadas, os projetos de *cyberinfrastructure* não precisam implantar seus próprios sistemas de gestão de identidades e podem se beneficiar da usabilidade da autenticação via *Shibboleth*.

Na abordagem proposta, o Serviço Gerador de Certificados (SGC) é uma aplicação *web*, hospedada em um provedor de serviços *Shibboleth*, que tem por objetivo permitir que usuários possam usufruir dos benefícios da federação para a emissão de certificados digitais por uma AC on-line. O SGC possui uma relação de confiança com a AC e esta só aceita requisições para emissão de certificados que forem originadas por este serviço. Este requisito, além de aumentar a segurança, torna o modelo adequado para ambientes dinâmicos e de larga escala, pois a AC só precisa estabelecer uma única relação de confiança e todos os usuários da federação usufruem da mesma. Sendo assim, o usuário interage com o serviço e é este último que envia a requisição para emissão de certificado. Por fim, o serviço retorna ao usuário o certificado que fora emitido.

O Serviço Gerador de Certificados pode utilizar como autoridades certificadoras on-line, uma AC online de resposta automática concebida com o software SGSI 2.0 e um AC on-line do tipo *MyProxy*. Quando a AC SGSI é utilizada, o SGC assume o papel de uma AR (autoridade de registro da ICPEdu) e todas as trocas entre o Serviço e à AC passam a ser realizadas pelo protocolo CMP, conforme descrito na Seção 2. Vale destacar que uma relação de confiança precisa ser estabelecida previamente entre o Gerador de Certificados e a AC SGSI. Quando o software MyProxy CA é utilizado, o SGC executa o aplicativo myproxy-logon para solicitar o certificado à CA. O Myproxy CA verifica se o usuário para quem está sendo solicitado a emissão de certificados está presente em sua base de dados (LDAP) e retorna o certificado gerado.

Como explicitado na Seção 2, a requisição para geração de certificados deve ser assinada com a chave privada correspondente à chave pública que está contida na própria requisição. É desta forma que a AC garante que estará emitindo um certificado para o detentor da chave privada. Como esta requisição é encaminhada pelo Serviço Gerador de Certificados à AC, e não pelo usuário, é necessário que: (1) o usuário forneça o *Certificate Signing Request* (CSR) ao SGC; ou (2) o próprio SGC gere o CSR em nome do usuário.

Para a opção (2), o SGC também deve ter acesso ao par de chaves do usuário, podendo este ter sido encaminhado pelo usuário ou o próprio SGC deve gerar este par de chaves. O fato do Serviço Gerador de Certificado ser um intermediário entre o usuário final e a AC, causa fragilidades no modelo de confiança, pois não é possível garantir a propriedade de segurança de não-repúdio por parte do usuário, haja visto que o usuário pode alegar que o Serviço Gerador de Certificados possui completa autonomia para agir em seu nome, pois teve acesso a sua chave privada, sem que este tenha controle ou mesmo sem que este possa rastrear tal ato. Diante desta análise, a opção adotada no serviço proposto foi a de o usuário gerar o par de chaves assimétricas, montar o CSR localmente e encaminhá-lo ao Serviço Gerador de Certificado (opção 1).

Uma das motivações para a criação do Serviço Gerador de Certificados, foi usufruir da usabilidade oferecida aos usuários da federação no processo de autenticação. Seguindo esta linha, o processo para geração do par de chaves do usuário localmente também deve ser facilitado para o usuário. Definiu-se como requisito que o usuário deve

 ©2012 SBC — Soc. Bras. de Computação

usar o navegador *web* para gerar do par de chaves, gerar e encaminhar o CSR e receber o certificado gerado pela AC. A solução também deve ser compatível com os principais navegadores *web* e sistemas operacionais para computadores pessoais.

Diante destes requisitos, duas soluções tecnológicas foram avaliadas: (1) rotinas em *JavaScript* que são executadas localmente no navegador do usuário combinadas ao uso de HTML 5 e (2) uma aplicação em *Java Web Start* (JWS), que permite instalar e iniciar aplicativos Java direto de um navegador *web*[10]. Rotinas em *Javascripts* são altamente dependentes do navegador e do sistema operacional o que demanda o desenvolvimento de diversas rotinas ou versões, bem como mantê-las. Por estes motivos, a aplicação JWS, que oferece portabilidade e uma manutenção mais simples, mostrou-se como a mais adequada para o serviço proposto.

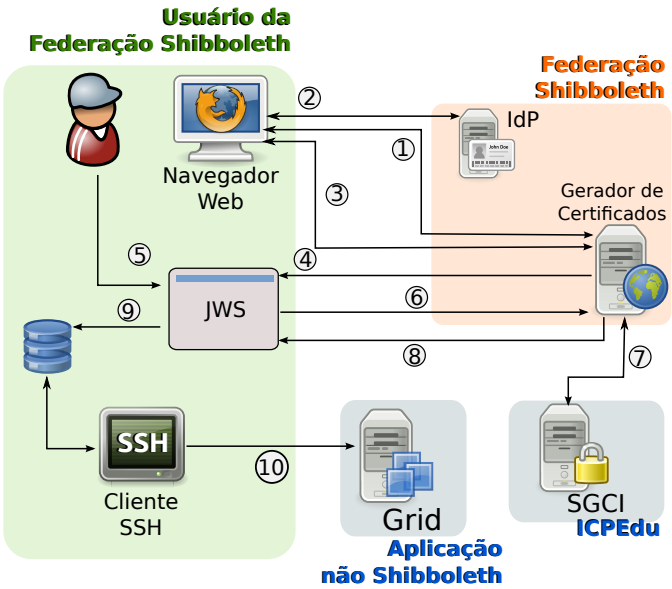



**Figura 1. Acesso a uma aplicação não Shibboleth que requer certificados X.509**

A Figura 1 ilustra os passos para que um usuário de uma federação *Shibboleth* acesse um *grid*, uma aplicação não *Shibboleth*, que exige um certificado X.509 emitido por uma Autoridade Certificadora (AC) SGCI da ICPedu [RNP 2012]. No passo 1, o usuário tenta acessar o Serviço Gerador de Certificados, que por ser um SP Shibboleth, exige que o usuário se autentique em um IdP da Federação. Após indicar o seu IdP, o navegador do usuário é redirecionado para o IdP para que o usuário proceda a autenticação (passo 2). Após autenticação bem sucedida, no passo 3, o navegador do usuário é redirecionado para o SGC que apresenta para o usuário o DN (*Distinguished Name*) que estará em seu certificado X.509 e informa a URL para *download* da aplicação JWS (passo 4). No passo 5, o usuário fornece a senha que será usada para proteger a chave privada. Em seguida, a aplicação JWS gera o par de chaves, monta o pedido de certificado (CSR) e o encaminha ao SGC (passo 6). No passo 7, o CSR é encaminhado à AC SGSI para que esta emita o certificado. No passo 8, o certificado gerado é encaminhado para o usuário,

[10]O software Java Web Start é iniciado automaticamente quando é feito o primeiro download de um aplicativo Java que utiliza essa tecnologia. O Java Web Start armazena todo o aplicativo localmente, na memória cache do computador. Assim, todas as inicializações subsequentes são quase instantâneas, pois todos os recursos necessários já estão disponíveis localmente.

 ©2012 SBC — Soc. Bras. de Computação

**DN do Sujeito do Certificado**

CN = *cn* (atributo do esquema *br.EduPerson obtido da asserção SAML*) + eppn
OU = *OrganizationDisplayName* (elemento do *EntityDescriptor* do IdP do usuário)
O = *Service Organization* (organização responsável pelo Serviço para Geração de Certificados)
C = *country* (país da Federação)

**Exemplo**:
CN = Michelle Wangham 2335565@univali.br
OU = idp.univali.br
O = RNP GT-STCFed
C = BR

**Figura 2. Mapeamento do *Distinguished Name* (DN) do sujeito do certificado**

que deve indicar o caminho em disco no qual o arquivo será salvo (passo 9). É importante observar que o usuário receberá o certificado no formato PKCS#12, sendo que a chave privada estará protegida pela senha digitada no passo 5. Por fim, o usuário, através de uma aplicação cliente[11], acessa a grid, fazendo uso do certificado recebido (passo 10).

No Serviço Gerador de Certificados, após a autenticação do usuário em seu provedor de identidades (IdP), os atributos do usuário são enviado para o serviço através de asserções SAML emitidas pelo IdP. Tais atributos, pertencentes ao conjunto de atributos comuns da Federação CAFe[12] da RNP [RNP 2009], devem ser mapeados para compor os campos do *Distinguished Name* (DN) do sujeito do certificado, conforme indicado na Figura 2. De forma a tratar os possíveis homônimos, a abordagem recomenda o uso do atributo *eduPersonPrincipalName* (*eppn*) ou do atributo *mail* do esquema brEduPerson [RNP 2009]. É importante observar que o mapeamento dos atributos da asserção SAML para os campos do DN do sujeito podem ser facilmente configurados no SGC. Já o *Distinguished Name* (DN) do emissor do certificado deve ser configurado nas Autoridades Certificadoras On-line suportadas no serviço proposto, conforme ilustrado na Figura 3.

**DN do Emissor do Certificado**

CN = *Nome da AC On-line (do tipo SGCI CA ou MyProxy CA)*
O = *Nome da Federação Shibboleth*
C = *country* (país da Federação)

**Exemplo**:
CN = onlinecaSTCFed SGCI
O = RNP Chimarrão
C = BR

**Figura 3. Mapeamento do *Distinguished Name* (DN) do emissor do certificado**

### 3.2. IdP+: Provedor de Identidades com Suporte a Geração de Certificados Digitais

A abordagem apresentada na Seção 3.1, apesar de permitir aos usuários da federação interagirem com aplicações não *Shibboleth*, esta exige que a interação do usuário com o Serviço Gerador de Certificados seja feita através de um navegador *web*. Este tipo de funcionamento impossibilita que aplicações *desktop*, que necessitem autenticar seus usuários, usufruam das facilidades da federação.

[11] Aplicação GSI-Putty (`www.ngs.ac.uk/research-and-development/gsi-putty`)
[12] `http://www.rnp.br/servicos/cafe.html`

 ©2012 SBC — Soc. Bras. de Computação

Na segunda abordagem proposta, o provedor de identidades com suporte a geração de certificados, denominado IdP+, consiste de um IdP com as mesmas habilidades do Serviço Gerador de Certificados, porém com funcionalidades adicionais que permitem que aplicações não *web* usufruam dos benefícios da autenticação federada *Shibboleth*. Ou seja, o IdP+ pode ser acessado através de um navegador *web* ou não.

Visando garantir a interoperabilidade e assim atender uma gama maior de aplicações, o IdP+ segue a arquitetura orientada a serviços e está baseado nas principais especificações relacionadas aos Serviços *Web*. O IdP+ consiste de um IdP padrão do *framework Shibboleth* acrescido da implementação do *Secure Token Service* (STS), definido pela especificação WS-Trust [OASIS 2009], do *Credential Translation Service* (CTS), responsável por realizar a tradução de credenciais de diferentes tecnologias e de uma aplicação *web*, utilizada quando um usuário, via um navegador *web*, solicita a geração de um certificado digital (ver cenário de uso ilustrado na Figura 4).

O Security Token Service (STS) consiste de um Serviço *Web* que tem como funções emitir e validar credenciais de segurança, de acordo com as especificações WS-Trust , WS-Security e WS-Policy. É importante ressaltar que o serviço STS atua como *gateway* confiável entre provedores de identidade de uma federação Shibboleth e aplicações não *web*. O Credential Translation Service (CTS), por sua vez, trata de aspectos de tradução de credenciais entre diferentes tecnologias de segurança e é sempre invocado pelo STS quando a aplicação requer uma credencial de segurança diferente da usada pela federação (p. ex. certificados X.509). A aplicação *web* é a mesma do Serviço Gerador de Certificados apresentado na Seção 3.1, porém é disponibilizada no mesmo contêiner *web* do IdP+ e não em um SP separado.

A geração de certificados digitais X.509, emitidos por uma AC on-line, a partir das asserções SAML, segue o mesmo conceito apresentado na Seção 3.1. A AC, que pode ser do tipo *MyProxy CA* ou uma AC on-line de resposta automática SGSI, deve possuir relações de confiança com os IdP+ da federação e somente estes poderão encaminhar requisições para emissão de certificados.

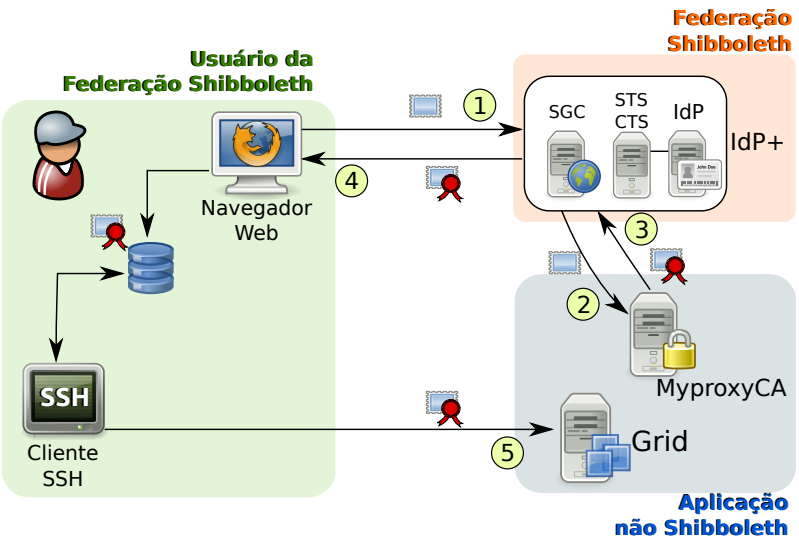



**Figura 4. Geração de Credenciais X.509 com o IdP+**

 ©2012 SBC — Soc. Bras. de Computação

A Figura 4 apresenta os passos simplificados para um usuário, através de seu navegador *web*, obter um certificado digital necessário para acessar um *grid*, a partir de uma aplicação *web* que se encontra em seu IdP+. O cenário é semelhante ao apresentado na Figura 1. No passo 1, o usuário, tenta acessar a aplicação *web* para geração de certificados, como não está autenticado, este é redirecionado para a página de autenticação do IdP+. Após autenticação bem sucedida, a asserção SAML é disponibilizada ao CTS para que este conduza o processo de tradução de credencial. Este processo segue as mesmas regras de mapeamento ilustradas nas Figuras 2 e 3. A aplicação *web* recebe do CTS o DN do certificado digital, o apresenta para o usuário e indica a URL para baixar a aplicação JWS. Em seguida, a aplicação *web* insere o DN do certificado do usuário em questão em uma base de dados LDAP que o *MyProxy CA* usa para gerar os certificados. De forma semelhante a abordagem da Seção 3.1, a aplicação JWS gera o par de chaves na máquina do usuário, monta o pedido de certificado (CSR) e reencaminha o pedido ao IdP+ (aplicação *web*). No passo 2, o IdP+ encaminha o pedido à Autoridade Certificadora, neste exemplo o *MyProxy CA* (Ver Seção 2.1). O *MyProxy CA* verifica se o usuário para quem está sendo solicitado a emissão de certificados está presente em sua base de dados e retorna o certificado gerado (passo 3). No passo 4, a aplicação *web* do IdP+ disponibiliza o certificado para que o usuário possa fazer o *download* em seu computador. Por fim, o usuário acessa o *grid* usufruindo do certificado recebido (passo 5).

A base LDAP instalada junto ao *MyProxy CA* foi configurada para aceitar somente conexões sob o SSL/TLS. Diferentes IdPs+ terão acesso de escrita a essa base, apresentando nome de usuário e senha, e assim foram criadas listas de controle de acesso (*Access Control List* – ACL) a fim de evitar que um IdP+ possa ver e/ou modificar as entradas feitas por outros IdPs+. Foi desenvolvido um serviço de configuração[13] para permitir que cada administrador de IdP+, que tenha interesse em prover geração de certificados digitais, possa criar seu nome de usuário e senha nesta base LDAP.

A Figura 5 ilustra um cenário de uso de uma aplicação não *web* (aplicação ICE) solicitando a geração de um certificado digital. A aplicação *desktop* ICE, desenvolvida pelo projeto MonIPÊ[14], é usada para visualização de dados de monitoramento de redes do projeto perfSONAR[15], o qual está baseado em uma arquitetura orientada a serviços. A aplicação ICE original realiza a autenticação de seus usuários de forma isolada, cabendo aos administradores desta ferramenta gerenciar tal base. A aplicação ICE precisou ser estendida para que pudesse interagir com o IdP+ e assim permitir a autenticação federada de usuários.

Conforme ilustrado na Figura 5, o perfSONAR possui um Serviço de Autenticação (*Authentication Service*), o qual consiste em um Serviço *Web* que consome asserções SAMLv1 ou certificados X.509 para garantir a autenticação e autorização dos usuários para os demais serviços de monitoramento. A autenticação se dá pela confiança no emissor da asserção ou na AC on-line do certificado X.509. No passo 1 da Figura 5, o usuário faz sua autenticação junto ao IdP+, através da própria aplicação ICE, isto é, não é feito uso de navegador *web*. O IdP+ solicita a uma AC on-line a emissão de um certificado

[13]Este serviço de configuração é um SP Shibboleth e somente usuários com atributo de administrador do IdP+ poderão acessá-lo.

[14]Um serviço da RNP – `http://www.monipe.rnp.br`

[15]`http://www.perfsonar.net`

 ©2012 SBC — Soc. Bras. de Computação

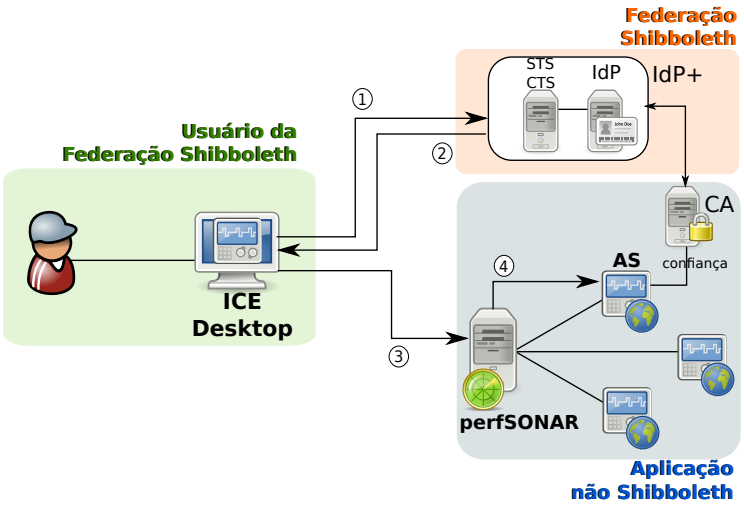



**Figura 5. Geração de Certificados Digitais através da interface STS do IdP+**

digital para o usuário e o entrega a aplicação ICE (passo 2). No passo 3, a aplicação ICE invoca o serviço perfSONAR, já com o certificado digital recebido. Por fim, o AS verifica se o certificado fora emitido por uma AC online com quem possui relação de confiança e verifica se o usuário possui os atributos necessários para garantir o acesso.

De forma semelhante ao cenário de uso da Figura 4, o CTS é usado para que as asserções SAML, emitidas pelo IdP+, possam ser mapeadas para certificados digitais e é este componente que interage com a AC on-line para emissão do certificado. É importante destacar que, a aplicação ICE é uma aplicação *desktop* que não possui mecanismos para interagir diretamente com o IdP padrão do *Shibboleth*. Logo, a aplicação ICE, para usufruir da autenticação federada, interage com o IdP+ via interface de Serviço *Web* do STS.

### 3.3. Análise das Abordagens Desenvolvidas

As abordagens apresentadas na Seções 3.1 e 3.2 são complementares e adequadas para diferentes necessidades. A disponibilização do Serviço Gerador de Certificados (SGC) em um provedor de serviço é a abordagem mais próxima do modelo de governança da Comunidade Acadêmica Federada (CAFe) da RNP. Nesta abordagem, o usuário fornece seu nome de usuário e senha através da interface *web* do seu provedor de identidade e o par de chaves é gerado localmente na máquina do usuário. Na abordagem com o IdP+, apesar do usuário só fornecer os dados ao seu provedor de identidade, isto pode também ser feito através da página *web* do provedor de identidade, como através de uma invocação do Serviço *Web*.

O Serviço Gerador de Certificados pode ser disponibilizado por cada uma das instituições pertencentes a federação CAFe ou pode ser um Serviço da gestora da federação (RNP), que pode ser responsável por oferecer este serviço a todos os usuários da federação, constituindo assim um único Serviço para toda a federação. A vantagem de centralizar um único Serviço para a federação está no gerenciamento das relações de confiança com as Autoridades Certificadoras, trazendo ao modelo a habilidade de operar em ambientes de larga escala. Por outro lado, a centralização tira a liberdade das instituições em estabelecer relações de confiança bilaterais com ACs específicas e não

 ©2012 SBC — Soc. Bras. de Computação

atrativas para toda a federação. De qualquer forma, a abordagem apresentada na Seção 3.1 permite que existam um ou mais Serviço Gerador de Certificados na federação, atendendo assim ambas as necessidades.

De acordo com as regras de governança da federação CAFe, uma instituição para ingressar na federação precisa disponibilizar um IdP, ou seja, não é requisito que a instituição também disponibilize um provedor de serviços (SP). Sendo assim, toda instituição participante da CAFe obrigatoriamente possui um IdP, mas não necessariamente um SP. Supondo que a instituição tenha interesse em um serviço gerador de certificados, mas a federação não ofereça tal serviço e a instituição não tenha interesse em disponibilizar um SP, a escolha ideal seria a implantação do IdP+.

O fato do IdP+ ser um complemento do IdP, apresenta um número menor de requisitos para instalação, culminando também uma menor complexidade para implantação e gerenciamento. Porém, para as instituições que já estão na CAFe e já possuem IdPs em produção, estas precisarão implantar as modificações necessárias para ser um IdP+. Esta abordagem teria como vantagem a permissão de estabelecimento de relações de confiança bilaterais com ACs, porém perdendo a habilidade de operar em ambientes de larga escala. O usuário continua fornecendo suas informações sensíveis somente ao IdP+, o que não fere o modelo de governança da federação CAFe. Por fim, o IdP+ por possuir um Serviço *Web* abre a possibilidade para que aplicações *desktop* possam usufruir das facilidades da federação.

## 4. Trabalhos Relacionados

Diversos projetos surgiram para interligar o Shibboleth com a tecnologia de autenticação comumente usada em grids, as credencias X.509. Dentre estes, destacam-se os projetos do *Joint Information Systems Committee* (JISC) da Grã-Bretanha, que visam permitir que usuários autenticados por IdPs Shibboleth confiáveis obtenham credenciais temporárias para acessar recursos do *National Grid Service* (NGS) da Grã-Bretanha (ShibGrid/Shebangs/Sarongs). O GridShib é outro importante projeto que integra o Shibboleth ao modelo computacional de grid e foi desenvolvido a partir de uma motivação diferente dos projetos do JISC. A principal motivação deste projeto é estender o modelo de autorização do Globus Toolkit para que atributos Shibboleth possam ser utilizados para autorização em grids construídos com o Globus. Dentre os projetos mais recentes, destacam-se o CILogon e o go.teragrid, ambos financiados pela *National Science Foundation* (NSF). A seguir, tem-se uma descrição a análise destas iniciativas.

### 4.1. Projeto *GridShib*

O projeto *GridShib* foi criado em 2004 com o apoio da National Science Foudation (NSF) para introduzir um modelo de autorização baseado em atributos no Globus toolkit. O *GridShib* permite que o Globus Toolkit (GT) e o *Shibboleth* interoperem. Neste caso, um par de plug-ins de software, um para o Globus Toolkit e outro para o Shibboleth, permitem que um *GT Grid Service Provider* (SP) solicite atributos de um usuário a um provedor de identidade (IdP) *Shibboleth*[Scavo and Welch 2007].

O GridShib possui dois modos de operação: *push* e *pull*. No modo *pull*, o cliente ao acessar um *grid SP*, também fornece seu certificado digital X.509. O *grid SP* autentica o pedido e extrai o DN (*distinguished name*) das credenciais X.509 recebidas. Se autenticação ocorrer com sucesso, o *grid SP* solicita mais atributos ao IdP do

 ©2012 SBC — Soc. Bras. de Computação

próprio domínio administrativo do cliente. O IdP, ou mais especificamente, a autoridade de atributos do IdP, autentica a solicitação de atributos, mapeia o DN para um nome local utilizando o *plugin GridShib for Shibboleth*, recupera os atributos para o cliente (convenientemente filtrados pela política de liberação de atributos do Shibboleth), formula uma asserção de atributos e a envia ao *grid SP*. O *grid SP* analisa a asserção de atributos, armazena os atributos, toma uma decisão de controle de acesso, processa a solicitação do cliente (assumindo que o acesso está concedido) e devolve uma resposta para o cliente [Scavo and Welch 2007].

No modo *push*, o próprio cliente obtém os atributos SAML e os apresenta ao *grid SP* no momento da requisição. Isto é feito através de um *proxy certificate* que embute uma asserção de atributos SAML. Neste caso, foi desenvolvido no projeto o software *GridShib SAML Tools* que emite asserções SAML e as embute em proxy certificates [Scavo and Welch 2007]. Deve-se destacar que a forma como o cliente obtém os atributos SAML não são especificadas no *GridShib*, ou seja, não há uma aplicação cliente para que usuário se autentique no IdP e receba a asserção SAML de forma segura. Esta limitação é decorrente na inviabilidade de uma aplicação não Web de se autenticar em um IdP Shibboleth.

### 4.2. Projetos da *Joint Information Systems Committee* (JISC)

Com um propósito diferente do *GridShib*, os projetos conduzidos pelo JISC, permitem o uso mecanismo de autenticação federada do Shibboleth para acessar um *grid service* ou um *grid portal*. Nestes projetos, as credenciais de acesso ao grid (X.509 proxy certificates) são geradas dinamicamente. Vale ressaltar que estes projetos, ao contrário do *GridShib*, não modificam o mecanismo de autenticação do Shibboleth e nem o grid toolkit. Um serviço intermediário, chamado CTS (Credential Translation Service), foi introduzido para traduzir credenciais Shibboleth em credenciais GSI temporárias (X.509 *proxy certificates*). Estes certificados são chamados de *low assurance certificates* e assinados por autoridades certificadoras on-line, as quais geram certificados de autenticação baseados em um método eletrônico que não necessita de uma identificação fotográfica para representar um humano e outras burocracias. A duração máxima destes certificados é de um milhão de segundos [Spence et al. 2006].

O *ShibGrid*, desenvolvido entre fevereiro de 2006 a janeiro de 2007, foi o primeiro projeto do JISC e teve como objetivo apresentar somente um protótipo. O *SHEBANGS* (*Shibboleth Enabled Bridge to Access the National Grid Service*) surgiu como uma solução mais consolidada para o problema. No desenvolvimento deste projeto, o seguinte cenário foi considerado: "Um usuário final, pertencente a uma organização que possui um IdP Shibboleth, deseja acessar alguns recursos ou serviços provisionados no *National Grid Service* (NGS) da Grã-Bretanha. Além disso, assume-se que o usuário final (a) não possui um certificado digital do tipo normalmente exigido para acessar o NGS, (b) não tem qualquer treinamento em computação em grid, (c) não tem instalado qualquer software cliente para acessar o grid, e (d) pertence a uma organização virtual (VO- *virtual organization*) que é reconhecida pelo NGS e cujos membros herdam direitos para usar recursos do NGS".

No projeto SHEBANGS, foi desenvolvido um serviço tradutor de credenciais (CTS- *Credential Translation Service*) que traduz as credenciais *Shibboleth* em credenciais X.509 de curta duração e compatíveis com as credenciais VOMS. O *Virtual*

 ©2012 SBC — Soc. Bras. de Computação

*Organization Membership Service* (VOMS) é um sistema para gerenciar dados de autorização em colaborações multi-institucionais. O serviço fornece uma base de dados de papéis de usuário, suas competências e um conjunto de ferramentas para acessar e manipular o banco de dados para gerar as credenciais de Grid para usuários [Jones and S.Pickles 2007]. No SHEBANGS, o CTS atua como um SP Shibboleth, como uma AC online e como um cliente de um repositório de credenciais *MyProxy*.

No CTS, o mapeamento entre os atributos de identidade em SAML e nomes X.500, por exemplo o *distinguished name* (DN), não é uma tarefa trivial [Jones and S.Pickles 2007]. A especificação SAML fornece uma série de esquemas para a identificação de indivíduos, mas a computação em *grid* não lida bem com indivíduos com múltiplas identidades. Em *grid*, um usuário final que possui diferentes nomes será tratado como diferentes indivíduos. Em [Jones and S.Pickles 2007] é apresentado um algoritmo que descreve o mapeamento para criação das credenciais.

O nível de garantia (*Level of Assurance* – LoA) permite a um *grid service* tomar decisões de autenticação baseado nas garantias oferecidas pelo IdP. Essas garantias são fornecidas nas asserções SAML emitidas pelo IdP. *Grid services* com alto grau de segurança podem exigir somente credenciais assinadas por AC com alto grau de confiabilidade. Por outro lado, alguns *grid services* que não executam aplicações críticas, podem garantir o acesso a um usuário que tenha credencial de autenticação com um nível razoalvemente baixo. Por este motivo, o uso de múltiplas AC é a prática que melhor se adapta a comunidade de grid [Jones and S.Pickles 2007].

O projeto *SHEBANGS* foi finalizado em junho de 2007 e o JISC deu início a um outro projeto chamado *SARoNGS* (*Shibboleth Access for Resources on the National Grid Service*), o qual passou a se ocupar da integração *Shibboleth* e infraestruturas baseadas no X.509. O objetivo deste projeto é combinar as vantagens do *SHEBANGS* e do *ShibGrid*, porém, ao contrário dos projetos anteriores considerados de demonstração, visa oferecer um serviço de produção para ser utilizado no NGS. No projeto *SARoNGS*, o usuário usufrui da arquitetura desenvolvida para acessar o portal NGS através de um navegador web. Assim como no projeto anterior, o *SARoNGS* faz uso de um serviço intermediário confiável, o CTS, implantado como um SP Shibboleth, para realizar a tradução das asserções SAML para o certificados X.509. Porém, para emissão das credencias temporárias X.509, este faz uso de um servidor MyProxy CA modificado, como desenvolvido no projeto *ShibGrid*. Além disso, o CTS também interage com um servidor VOMS para obter uma credencial de atributos VOMS [16]. Por fim, o SARoNGS delega as credencias do usuário para o repositório de credencias MyProxy para prover facilidades no acesso ao Portal NGS.

### 4.3. Projeto de Logon Federado para o TeraGrid/XSEDE)

O site `https://go.teragrid.org/` permite aos usuários acessarem os recursos do XSEDE[17] (anteriormente TeraGrid) usando o mecanismo de autenticação federada fornecido por sua universidade. Somente os usuários de recursos ativos do projeto XSEDE podem usar este site. O sistema de autenticação federada do TeraGrid [Basney et al. 2010] é

[16]Um certificado de atributos X.509 emitido pela servidor VOMS para o sujeito do certificado digital.

[17]The Extreme Science and Engineering Discovery Environment (XSEDE) é financiado pela NSF e substitui e amplia o projeto TeraGrid

 ©2012 SBC — Soc. Bras. de Computação

uma aplicação *web* para o servidor Apache HTTPD e que faz uso das funcionalidades desenvolvidas no projeto *GridShib* para converter credenciais Shibboleth em certificados. A aplicação *web* liga um identidade federada de campus a uma identidade TeraGrid/XSEDE via uma base de dados de ligação de contas.

O funcionamento consiste dos seguintes passos. Primeiro, o pesquisador visita o site e seleciona, em uma lista, o provedor de identidade de seu campus e realiza o processo de autenticação neste. A aplicação redireciona o navegador para a página de login do TeraGrid para que o pesquisador faça o login no sistema e assim ligue as contas de seu IdP com a sua conta no TeraGrid (esta operação deve ser feita apenas uma vez). Após o login, a aplicação indica ao usuário que este pode fazer o *download* do certificado e, através de um aplicativo Java Web Start, acessar ao recurso via uma aplicativo desktop (GSI-SSH terminal applet), ou que este pode acessar o recurso do TeraGrid via User Portal. Quando o aplicativo Java Web Start é utilizado é este o responsável por gerar o par de chaves, montar o pedido de certificado, enviá-lo para o servidor My Proxy Ca, receber o certificado da AC on line e fazer o *download* do certificado na máquina do usuário [Basney et al. 2010]. Uma vez que o mapeamento entre a instituição de origem e identidades TeraGrid estiver completo, o pesquisador pode usar o seu login e senha institucional para acessar os seus serviços digitais do TeraGrid.

### 4.4. Projeto CILogon

O projeto CILogon, iniciado em setembro de 2009, está sendo desenvolvido pelo *National Center for Supercomputing Applications* (NCSA) da Universidade de Illinois e oferece um serviço online[18], de código aberto, para emissão de certificados digitais que proporciona à comunidade de pesquisa da *National Science Foundation* (NSF) o acesso seguro a *cyberinfrastructure*(CI).

O serviço CILogon oferece uma ponte entre as credenciais geradas no processo de autenticação em um provedor de identidades (IdP) da Federação *Incommon* e os certificados necessários para autenticação de pesquisadores que desejem usufruir da *cyberinfrastructure* da NSF [CILogon 2012]. Grande parte da abordagem do serviço do CILogon foi demonstrada previamente no sistema de login federado do TeraGrid [Basney et al. 2010]. Tal como acontece com o login federado do TeraGrid, o serviço CILongon usa o MyProxy CA para gerar esses certificados.

Reconhecendo que muitos usuário interessados em usar a *cyberinfrastructure* da NSF ainda estão fora da federação *InCommon*, o serviço CILogon permite ainda que certificados sejam gerados a partir da autenticação em um provedor OpenID, tal como o Google, PayPal e Verisign.

Para dar suporte aos dois níveis de garantia (LoA) definidos na Federação InCommon, o CILogon opera com duas autoridades certificadoras: uma para os membros da *InCommon* do nível bronze e outra para os membros que atendem o perfil prata. Operadores dos CI estão aptos a confiar em uma ou em ambas ACs com base em seu nível de garantia (LoA) e no tamanho de sua base de usuários. Visando atender os usuários que utilizam provedores OpenID, o serviço possui ainda a *CA CILogon OpenID* que fornece certificados com base na autenticação OpenID. As ACs se diferem apenas em seus processos de autenticação dos sujeitos, na validação e nomeação da identidade. Usando contas

[18]Em setembro de 2010, o serviço tornou-se operacional (`https://cilogon.org`)

 ©2012 SBC — Soc. Bras. de Computação

do Google, PayPal ou VeriSign, os pesquisadores podem se autenticar no CILogon via OpenID para obter um certificado CILogon OpenID. Embora este tipo de certificado tem um nível inferior de segurança, este também considerado como válido. O *Open Identity Exchange* (OIX) certificou esses provedores OpenID com LoA nível 1. Em muitos casos, este LoA é suficiente para o acesso a uma CI.

O serviço CILogon pode entregar uma credencial X.509 para um sistema local do usuário (via linha de comando) ou para a um portal *web* do projeto do CI. Neste último caso, o usuário pode ser redirecionado ao serviço CILogon, que irá autenticá-lo através da Federação *Incommon*. Por fim, será gerada uma credencial X.509 e encaminhada de forma segura para o portal do projeto. Essa credencial serve tanto para estabelecer a identidade do usuário para o portal quanto para ser usada pelo portal para acesso a outros serviços em nome do usuário.

O serviço CILogon faz uso do protocolo OAuth[19] para coordenar a autenticação de usuários e para delegar credenciais de usuários para o portal. A interação entre o serviço CILogon e portal só é possível se o portal integrar o código do cliente CILogon e se houver uma negociação prévia entre os administradores do serviço CILogon e portal para o estabelecimento dos mecanismos criptográficos para garantir a troca segura das mensagens [Barnett et al. 2011].

Um grid ou uma *cyberinfrastructure* oferece serviços que são acessados por aplicativos que rodam computador do usuário e a autenticação ocorre através de uma credencial X.509, armazenada no lado do usuário. Neste cenário, a entrega das credenciais (*download* do certificado) para o computador do usuário é feita exclusivamente pelo Serviço CILogon, conforme os passos resumidos a seguir:

1. Através de um navegador *web*, um usuário acessa o serviço (`https://cilogon.org/`) e indica o seu provedor de identidade;
2. Assumindo que este usuário ainda não tenha se autenticado, este é redirecionado para o seu IdP;
3. Após autenticar-se, o usuário é redirecionado para a aplicação *web*. Esta aplicação extrai da asserção SAML informações necessárias para montar o DN do certificado do usuário e o apresenta ao usuário solicitando que o mesmo informe a senha que protegerá a chave privada;
4. A aplicação monta o pedido de certificado (CSR) e através do `myproxy-logon` envia para o servidor *MyProxy CA*. Se este servidor, de acordo com suas políticas, após aceitar o pedido de certificado, gera um par de chaves e assina o certificado X.509 que contém a chave pública e o DN do usuário. É feito então o encapsulamento das chaves e certificado em um objeto PKCS12 protegido com a senha que fora escolhida pelo usuário. Por fim, um *link web* é disponibilizado para que o usuário possa baixar o certificado gerado;
5. Com o certificado salvo em sua máquina, o usuário poderá utilizar uma aplicação de linha de comando (p.ex. *GSI-OpenSSH*) para acessar a CI.

Assim como no sistema de autenticação federada do TeraGrid, o CILogon em suas primeiras versões também fazia uso do *Java Web Start* para gerar o par de chaves e o pedido de certificado diretamente na máquina do usuário. Contudo, na versão atual do CILogon esta funcionalidade não é mais oferecida.

[19]http://oauth.net/

 ©2012 SBC — Soc. Bras. de Computação

### 4.5. Comparação com os Projetos Relacionados

A Tabela 1 resume e compara algumas características dos projetos mais recentes que proveem a geração de certificados digitais a partir da autenticação federada *Shibboleth*. Pode-se observar que todos os projetos utilizam como autoridade certificadora on-line o MyProxy. As abordagens propostas neste trabalho (SGR e IdP+) oferecem ainda o suporte a uma AC online SGCI de resposta automática alinhada ao Serviço ICPedu da RNP. Como o Shibboleth é uma tecnologia baseada na *web*, desenvolvido para usuários com navegadores *web*, todas as soluções analisadas permitem que os usuários acessem o serviço para geração de certificados a partir de um navegador e que este serviço seja uma aplicação *web*. Em todos os projetos relacionados e na abordagem proposta do SGR, a aplicação *web* encontra-se hospedada em um provedor de serviços (SP) *Shibboleth*.

| **Características** | **Solução** | | | | |
|---|---|---|---|---|---|
| | SARoNGS | go.teragrid | CILogon | SGC | IdP+ |
| Suporte a AC on-line | MyProxy CA | | | MyProxyCA e SGCI | MyProxyCA e SGCI |
| Acesso via navegador *web* | Sim | Sim | Sim | Sim | Sim |
| Acesso via aplicação não *web* | Não | Não | Não | Não | Sim |
| *Shibboleth* | SP | SP | SP | SP | IdP |
| Geração de par de chaves e CSR | No CTS | Na máquina do usuário | No serviço | Na máquina do usuário | Na máquina do usuário |
| Uso de LoA | Sim | Não | Sim | Não | Não |
| Integrado com Portal de Grid | Sim *proxy certificates* | Não | Sim *proxy certificates* | Não | Não |
| Certificados X.509 | Credenciais temporárias | Credenciais perfil SLCS | Credenciais perfil SLCS | Credenciais perfil SLCS | Credenciais perfil SLCS |

**Tabela 1. Comparação dos projetos relacionados com SGC e IdP+**

A proposta IdP+, por possuir um Serviço *Web*, abre a possibilidade para que aplicações *desktop* possam usufruir das facilidades da autenticação federada para geração de certificados digitais. É importante ressaltar que a solução proposta não se limita na tradução apenas para certificados X.509, podendo ser estendida para suportar a tradução para outros tipos de credenciais. Este é a única solução que oferece esta funcionalidade.

Em relação a geração do par de chaves do usuário e da preparação do pedido de certificado (CSR), a solução do go.teragrid e as abordagens propostas neste trabalho realizam estas tarefas na máquina do usuário através de aplicativos JWS, o que que garante a propriedade de não repúdio.

Como a Federação CAFe da RNP ainda não definiu seu *framework* de avaliação de nível de garantida de seus provedores de identidades, a possibilidade de oferecer múltiplas AC on-lines de acordo com o LoA do provedor que emitiu a asserção SAML ainda não é provida nas abordagens propostas neste trabalho.

Na maioria das soluções analisadas, as ACs on-line emitem credencias de vida curta conforme o perfil SLCS [TAGPMA 2009]. Em relação a integração da solução para

 ©2012 SBC — Soc. Bras. de Computação

geração de certificados com portais de acesso a grids e *cyberinfrastructure*[20], apesar de estarem implementados de formas diferentes, os projetos SARoNGS e CILogon oferecem esta funcionalidade a seus usuários.

## 5. Conclusão

Neste trabalho foram apresentadas duas abordagens para permitir que credenciais SAML, empregadas em federações acadêmicas e que fazem uso do framework *Shibboleth*, possam ser convertidas em outros tipos de credenciais, por exemplo em certificados X.509, e assim usadas para acessar aplicações acadêmicas específicas, como os *grids*. Dentre as aplicações distribuídas que também podem se beneficiar das abordagens desenvolvidas, destacam-se as ferramentas visualização e de monitoramento do desempenho de redes, por exemplo, as utilizadas na arquitetura *perfSONAR*, as ferramentas de gerência e configuração de serviços de circuitos de de redes avançadas e as ferramentas utilizadas nas redes experimentais para Internet para pesquisa em Internet do Futuro.

Cada abordagem buscou atender requisitos de usabilidade ou mesmo de aderência às regras das federações e aos protocolos de comunicação de aplicações que não podem fazer uso direto do *framework Shibboleth.* O Serviço Gerador de Certificados mostrou ser uma opção adequada para ambientes dinâmicos e de larga escala, desde que a gerência da federação opte por oferecer este serviço para todos os membros de sua federação. O IdP+ tem como vantagens a oferta de uma interface baseada em Serviço *Web* para integração com aplicações não *web* e por ser uma solução integrada ao IdP, não sendo assim necessário a oferta de um provedor de serviços pela instituição, haja visto que algumas instituições optam por somente ingressar na federação com o seu provedor de identidades.

## Referências

Adams, C., Farrell, S., Kause, T., and Mononen, T. (2005). *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP).* IETF RFC 4210.

Barnett, W., Welch, V., Walsh, A., and Stewart, C. A. (2011). A roadmap for using nsf cyberinfrastructure with incommon. `http://www.incommon.org/nsfroadmap.html`.

Basney, J., Fleury, T., and Welch, V. (2010). Federated login to teragrid. In *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, IDTRUST '10, pages 1–11, New York, NY, USA. ACM.

Carmody, S., Erdos, M., Hazelton, K., Hoehn, W., Morgan, B., Scavo, T., and Wasley, D. (2005). Incommon technical requirements and information. vol. 2005.

CILogon (2012). Cilogon service. CILogon. `http://www.cilogon.org/service`.

Federation, I. (2011). Identity assurance assessment framework. `http://www.incommonfederation.org/docs/assurance/IAAF_V1.1.pdf`.

Internet2 (2012). Shibboleth. `http://shibboleth.net`.

[20]Possibilitar que o certificado gerado possa ser delegado para um repositório de credenciais (*proxy certificates*) e possa ser usado no acesso via Portal.

 ©2012 SBC — Soc. Bras. de Computação

Jones, M. and S.Pickles (2007). Shebangs: Final report. JISC development programmes. http://www.jisc.ac.uk/media/documents/programmes/middleware/shebangsfinalreport.pdf.

LabSEC (2012). Sistema de gerenciamento de certificados digitais da infraestrutura de chaves púlicas para ensino e pesquisa (sgci). https://projetos.labsec.ufsc.br/sgci.

Moreira, E., Foscarini, E., da Silva Junior, G. C., Aliandrina, L. A. O., Neto, L. P. V., and Rossetto, S. (2011). *Federação CAFe: Implantação do Provedor de Identidade*. Escola Superior de Redes. RNP.

MyProxy (2012). Credential management service. MyProxy. http://grid.ncsa.illinois.edu/myproxy/.

OASIS (2009). Ws-trust 1.4. OASIS Standard. http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.pdf.

RNP (2009). Esquema breduperson - versão 1.0. http://www.rnp.br/_arquivo/servicos/Esquema_brEduPerson.pdf.

RNP (2012). Infraestrutura de chaves públicas para ensino e pesquisa (icpedu). http://www.rnp.br/servicos/icpedu.html.

Scavo, T. and Welch, V. (2007). A grid authorization model for science gateways. In *International Workshop on Grid Computing Environments*.

Shibboleth (2005). *Shibboleth Architecture*. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.

Silvério, A., Kohler, J., and Custódio, R. (2011). Análise e implementação de um protocolo de gerenciamento de certificados. In *WTICG - SBSeg 2011*.

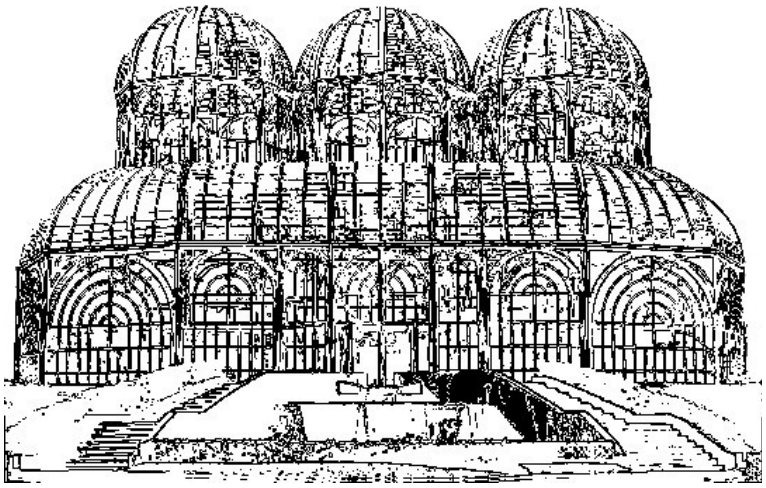
Spence, D., Geddes, N., Jensen, J., M., A. R., Viljoen, Martin, A., Dovey, M., Norman, M., Tang, K., Trefethen, A., Wallom, D., Allan, R., and Meredith, D. (2006). Shibgrid: Shibboleth access for the uk national grid service. In *Proceedings of the Second IEEE international Conference on E-Science and Grid Computing*.

TAGPMA (2009). Profile for short lived x.509 credential services (slcs) x.509 public key certification authorities with secured infrastructure. SLCS-2.1b. http://www.tagpma.org/files/SLCS-2.1b.pdf.

TERENA (2012). Research and education networking faq - general. http://www.terena.org/activities/development-support/r+e-faq/general.html#about.

Tuecke, S., Welch, V., Engert, D., Pearlman, L., and Thompson, M. (2004). *X.509 Proxy Certificate Profile*. IETF RFC 3820.

 ©2012 SBC — Soc. Bras. de Computação



# SBSeg 2012 Curitiba PR

XII Simpósio Brasileiro em Segurança da Informação
e de Sistemas Computacionais

# WFC – Workshop de Forense Computacional

# Metodologia para Análise Forense de Imagens Digitais utilizando Sensor Pattern Noise

**Alisson S. Nascimento, Leonardo Torres, Valter Ramos, José Alencar-Neto**

[1]Departamento de Pesquisa & Desenvolvimento
Laboratório de Investigação e Perícia Digital LTDA – RASTRU
57057-780 – Maceió – Brasil

{alisson.sa.nascimento, ljmtorres, valter, alencar}@rastru.com.br

***Abstract.*** *This paper proposes a methodology based on the principle of divisibility of the problem in forensic analysis. The proposal is focused on outline a workflow for the experiments using the Sensor Pattern Noise in the forensic field. The methodology has three stages: (i) Acquisition and Storage of Images, (ii) Extraction of Attributes and (iii) Decision-making. In the first stage we propose a model entity-relationship. Following, we surveyed the main feature extraction techniques. The third stage relates to the decision and classification methods widely used. Finally, we present the conclusions and next steps.*

***Resumo.*** *Este artigo propõe uma metodologia baseada no princípio da divisibilidade do problema em análise forense. A proposta tem como foco delinear um fluxo de trabalho para os experimentos utilizando o Sensor Pattern Noise no uso forense. A metodologia possui três etapas: (i) Aquisição e Armazenamento das Imagens, (ii) Extração de Atributos e (iii) Processo de Decisão. Na primeira etapa propomos um modelo entidade-relacional para este fim. Em seguida, foram levantadas as principais técnicas de extração de atributos. A terceira etapa refere-se aos métodos de decisão e classificação largamente utilizados. Por fim, apresentamos as considerações finais e próximos passos.*

## 1. Introdução

Imagens e vídeos tornaram-se os principais veículos de informação na era digital. O potencial expressivo da mídia visual aliado a falicidade de aquisição, distribuição e de armazenamento contribuem fortemente para sua crescente exploração de tais mídias como meio para transmição de informações. Como consequência, as imagens e vídeos, hoje, representam uma fonte comum de evidência. Atualmente, a verificação da autenticidade dessas mídias digitais é de fundamental importância, pois cada vez mais nos deparamos com fatos e acontecimentos que são colocados à prova e estas podem ser utilizadas como evidências num tribunal.

O acesso aos formatos digitais traz uma grande desvantagem à segurança dos dados. Especialistas em processamento de imagens podem facilmente acessar e modificar o seu conteúdo e, portanto, seu significado, podendo não deixar vestígios visuais. Além disso, com o baixo custo de ferramentas de edição com um *front-end* amigável, a adulteração e falsificação de conteúdo visual não é mais restrita a especialistas. Outro ponto negativo é a modificação de imagens para fins maliciosos.

 ©2012 SBC — Soc. Bras. de Computação

Redi et al. trazem um bom exemplo em seu trabalho [Redi et al. 2011]: em julho de 2010, Jeffrey Su Wong En, um político da Malásia, alegou ter sido nomeado cavaleiro pela rainha Elizabeth II; como reconhecimento pela sua contribuição para a organização internacional de ajuda *Médecins Sans Frontières*. Uma foto dele sendo condecorado pela Rainha da Inglaterra acompanhou seu comunicado, difundido nos meios de comunicação locais (veja Figura 1(a)). Quando questionado sobre o prêmio, o Alto Comissariado Britânico em Kuala Lumpur deixou claro que o nome do Sr. Wong En não foi incluído nas listas oficiais, mas que a imagem era incompatível com o protocolo usual adotado para cerimônias de cavaleiros. A imagem foi finalmente revelada ser uma junção entre uma foto cerimônia original (veja Figura 1(b)) e o busto do Sr. Wong En.

(a) Condecoração falsa (b) Condecoração verdadeira

**Figura 1. (a) Jeffrey Su Wong En recebendo o título das mãos da Rainha Elizabeth II, publicado em jornais da Malásia; e (b) imagem original de Ross Brawn recebendo a Ordem do Império Britânico da Rainha.**

A análise forense de vídeos e imagens digitais é uma área recente, cujo interesse da comunidade científica vem aumentando no decorrer dos anos. Uma das técnicas bastante utilizada na análise forense são as *blind techniques*, que usam atributos extraídos das próprias imagens em análise. Assim, não necessitam inserir características em qualquer etapa da formação da imagem para posterior verificação ou validação, características que levam vantagens sobre outras técnicas, como exemplo, *watermarks*.

Uma outra objeção em se utilizar técnicas baseadas em características físicas é a facilidade de adulteração das informações que as imagens levam no cabeçalho (e.g. EXIF), por exemplo, diminuindo sua credibilidade.

Muitos autores vêm se dedicando bastante ao uso do *Sensor Pattern Noise* (SPN) como principal característica para identificar um dispositivo, como em [Lukáš et al. 2006, Celiktutan et al. 2008, Li 2010, Kang et al. 2012]. O SPN vem se mostrando cada vez mais eficaz, visto que cada dispositivo deixa, na imagem, um tipo de impressão digital (*fingerprint*), uma marca única e inerente a cada sensor, mesmo em marcas e modelos iguais.

Lukáš et al. descrevem que o ruído produzido no processo de aquisição pode ser dividido em duas categorias: (i) *Fixed Pattern Noise* (FPN) e (ii) *Photoresponse Nonuniformity* (PRNU). O FPN é formado por pontos escuros, pontos do sensor (CCD ou CMOS) não expostos à luz. Em uma imagem natural o que predomina é a PRNU, que é formada pelo ruído devido a não uniformidade dos pixels (*Pixel Nonuniformity* – PNU) e pelos componentes de baixa frequência. Todos os sensores não recebem a mesma in-

 ©2012 SBC — Soc. Bras. de Computação

tensidade de luz da cena, tornando-se mais uma característica do processo de fabricação do dispositivo. Porém, diferentemente do FPN, a temperatura não é considerada para contribuição na PNU [Lukáš et al. 2006].

Este trabalho apresenta uma metodologia para a condução dos experimentos para a análise forense de imagens digitais com a utilização do ruído padrão do sensor. A metodologia proposta possui como base um dos princípios da forense clássica e é indiferente a aplicação: a **divisibilidade do problema**, onde divide-se o problema em partes que sejam de mais fácil resolução, essas partes retêm características do problema original, bem como adquirem características geradas pela divisão em si [Böhme et al. 2009]. O número de partes deve ser moderado, para que o problema original não seja muito dissolvido.

A Figura 2 apresenta o diagrama em blocos da metodologia proposta, que consiste em três etapas: (i) Aquisição e Armazenamento das Imagens, (ii) Extração de Atributos e (iii) Processo de Decisão.

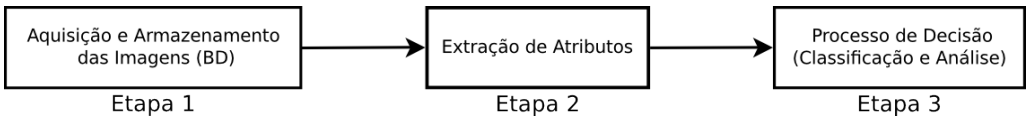



**Figura 2. Diagrama em blocos da metodologia proposta.**

A metodologia proposta está descrita nas próximas seções: A Seção 2 apresenta as principais condições para captura das imagens e uma modelo entidade-relacional para armazenamento destas imagens. A Seção 3 levanta as técnicas usadas para extração de atributos. A Seção 4 elenca as técnicas de classificação e processo decisão. E por fim, na Seção 5 nós discutimos as contribuições e trabalhos futuros.

## 2. Aquisição e Armazenamento das Imagens

A primeira etapa da metodologia é a aquisição e armazenamento das imagens capturadas para formação do modelo do dispositivo e das imagens investigadas. Nesta etapa definimos os dispositivos a serem usados, quantidade de imagens capturadas, ambiente (urbano, em laboratório), dentre outros. Podemos definir também, dependendo da necessidade, se serão usadas imagens de uma base de dados disponível.

A Figura 3 mostra um modelo de aquisição de imagem digital: a luz refletida na cena e nos objetos contidos nela passa através de lentes e de filtros óticos, para que seja capturada pelos sensores de cores. A maioria das câmeras digitais usam sensores *Charge-coupled Device* (CCD) ou *Complementary Metal-Oxide-Semiconductor* (CMOS) e capturam as imagens usando um CFA. Os CFAs são matrizes, em sua maioria, com três filtros de cores em padrão RGB (*red*, *green*, *blue*), acoplados a cada sensor. Cada pixel de uma imagem digital tem três valores, um de cada cor ou banda, que compõem a cor do pixel, porém cada ponto do CFA captura um único valor de uma banda específica, os valores das outras bandas são inexistente, até então. O processo de interpolação usa os valores da vizinhança do pixel nas outras bandas para estimar os valores inexistentes, afim, de torna possível a exibição da imagem e, finalmente, pode ser aplicado um pós-processamento onde, por exemplo, pode ser feito uma compressão JPEG.

 ©2012 SBC — Soc. Bras. de Computação

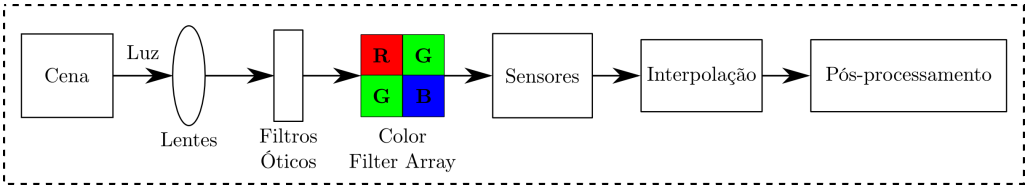



**Figura 3. Modelo de aquisição de imagem digital.**

## 2.1. Banco de Dados

Em uma imagem digital há muitas características inerente ao processo de aquisição e formação, marcas como a PNU já descrita. E ainda existem outros fatores que podem influenciar na saída final da imagem, como: foco, zoom da câmera, luz natural, campo aberto, local fechado, dentre outros. Estas, por sua vez, contribuem à PRNU [Lukáš et al. 2006].

Por todas estas variáveis, quanto mais supervisionado o ambiente em que essas imagens são adquiridas, mais controlado é o experimento e melhor será o entendimento sobre os resultados.

Os autores descrevem algumas formas de adquirir imagens que irão compor o banco de dados para os testes. Inicialmente, obtém-se algumas câmeras de vários modelos e fabricantes, se possível algumas câmeras de mesmo modelo e fabricante. Destas câmeras são tiradas dois grupos de fotografias: um grupo para treinamento e outro para testes. O grupo de fotografias para treinamento tem o objetivo de ajustar o algoritmo, estimar uma *fingerprint* e armazenar estes dados para comparações posteriores. Para compor este grupo são adquiridas no mínimo $50$ imagens, geralmente, com câmera fixada em um mesmo objeto ou de céu azul com o mínimo de nuvens, afim de coletar atributos.

O segundo grupo de imagens são para os testes, propriamente dito, onde os atributos extraídos destas imagens serão comparados com as *fingerprints* anteriormente salvas. O classificador deve ser capaz de identificar de qual câmera a imagem de teste foi adquirida ou, pelo menos, se não pertence a nenhuma das câmeras. Este grupo pode conter imagens heterogêneas, ou seja, imagens naturais obtidas ao acaso. Há trabalhos que defendem que os dois grupos de imagens sejam de ambientes variados.

A coleta das imagens de várias câmeras não é uma tarefa fácil e nem muito econômica, para pesquisadores acadêmicos ou independentes. O *Dresden Image Database for Benchmarking Digital Image Forensics* [Gloe and Böhme 2010] contém mais $14$ mil imagens em JPEG e TIFF de vários tipos de cenas, capturadas de $73$ modelos de câmeras diferentes. Esse banco de dados é uma ferramenta de alto valor para a comunidade forense, pois apresenta uma visão clara do estado da arte em identificação de câmeras.

Inspirados pelo trabalho de [Gloe and Böhme 2010], nós propomos um modelo entidade-relacional para o armazenamento de imagens e informações do método de aquisição. Um Banco de Dados (BD) é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico e um Sistema Gerenciador de Banco de Dados (SGBD) provê ferramentas para acesso aos BDs.

A Figura 4 mostra um modelo simplificado de banco de dados proposto. As

 ©2012 SBC — Soc. Bras. de Computação

entidades são representadas por retângulos, os relacionamentos são representados por losangos e os atributos são as elipses. O relacionamento entre um `Fabricante` e um `Modelo` caracteriza um `Dispositivo`. Podem existir vários `Modelos` para cada `Fabricante`, assim como vários `Dispositivos` com um mesmo `Modelo` e `Fabricante`. O `Ambiente` tem tipo (urbano, céu azul, natural) e uma *flag* indicando se é `indoor` ou `outdoor`. As condições em que uma imagem foi capturada (câmera fixa, distância focal, flash) são registradas no relacionamento `Condições`, caracterizado pelo relacionamento entre o `Ambiente` e um `Dispositivo`. Cada `Imagem` recebe um identificador (`Id`) criptografado.

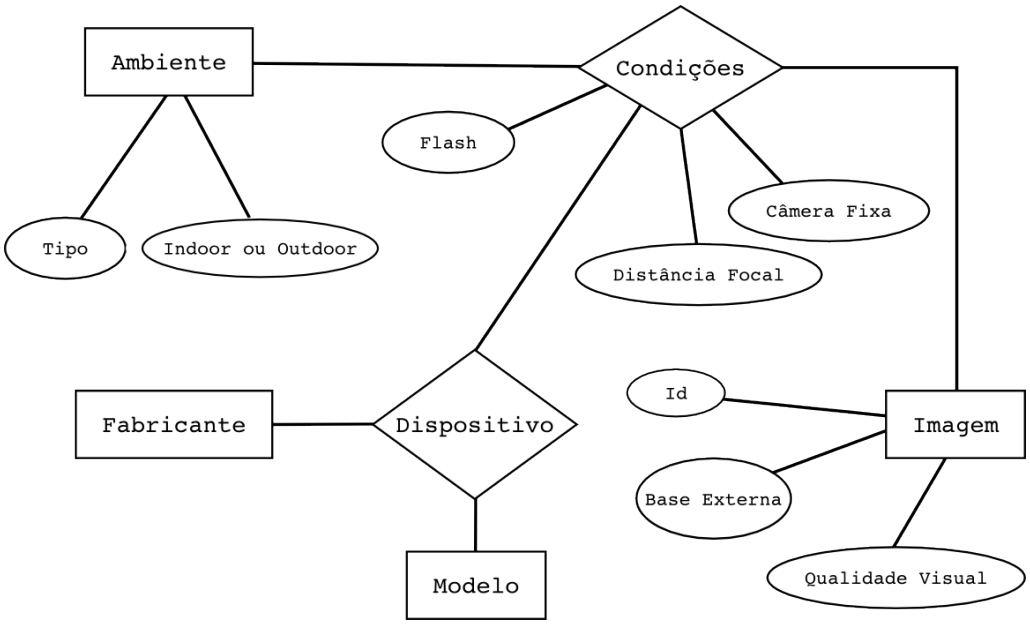



**Figura 4. Modelo Entidade-Relacional simplificado.**

Como se trata de um modelo simplificado será necessário sua adequação para cada proposta de pesquisa, ficando a critério a adição de atributos ou entidades.

## 3. Extração de Atributos

### 3.1. Ruído Residual como Atributo

A supressão do ruído é um passo importante no processo de investigação e identificação de dispositivos através do SPN. Trabalhos como de [Lukáš et al. 2006, Chen et al. 2008, Li 2010, Goljan et al. 2011] e [Kang et al. 2012], obtém-se o ruído residual, denotado $w_k$, por técnicas de filtragem e dado por

$$w_k = I_k - F(I_k), \tag{1}$$

em que $I_k$ representa uma imagem natural e $F$ um função de filtragem. Assim, a média do ruído residual de $n$-imagens, denotada $\eta$, é dada por

$$\eta = \frac{1}{n}\sum_{k=1}^{n} w_k, \tag{2}$$

 ©2012 SBC — Soc. Bras. de Computação

em que $n$ representa o número total de imagens usadas para estimar a impressão digital do sensor.

Em estudos complementares [Chen et al. 2008, Goljan et al. 2011], propõem refinar o método anterior pela estimação da PRNU para o ruído padrão $w_k$. Observa-se que o ruído padrão pode ser expresso como a contribuição conjunta de dois termos, dado por

$$w_k = \boldsymbol{IK} + \Xi \tag{3}$$

em que $\boldsymbol{IK}$ é o sinal da PRNU e $\Xi$ são os vestígios do processo de aquisição da imagem. Assumindo $d$ imagens $I_k$, $k = 1, \ldots, d$, obtidas a partir da mesma câmera $C$, o fator referente a PRNU desta câmera, denotada $\widehat{\boldsymbol{K}}^C$, pode ser estimado por máxima verossimilhança pela equação

$$\widehat{\boldsymbol{K}}^C = \frac{\sum_{k=1}^{d} w_k I_w}{\sum_{k=1}^{d} (I_k)^2} \tag{4}$$

em que $w_k$ é obtido pela Equação 1.

### 3.2. Atributos Estatísticos

Outros trabalhos implementam filtros de extração de ruído, porém, não usam a *fingerprint* do sensor e sim extraem atributos estatísticos para possíveis combinações ou ter um maior número de dados para o processo de tomada de decisão.

Gou et al. extraem atributos de três processos diferentes. No primeiro processo usam quatro tipos de diminuição de ruído na imagem em blocos de $3 \times 3$ e $5 \times 5$. No segundo processo os atributos são resultado de uma análise wavelet em cada banda da imagem (vermelho, verde e azul), para cada banda obtém-se atributos baseados nas frequências, nomeados: alta-alta, alta-baixa, baixa-alta e baixa-baixa, descartando-se a baixa-baixa, utilizando-se apenas atributos de alta frequência como entrada na tomada de decisão. O ultimo grupo de atributos vêm do cálculo do erro estimado na análise da vizinhança [Gou et al. 2007a].

Celiktutan et al. extraem dois tipos de estatísticas do ruído da cada imagem [Celiktutan et al. 2008]: O primeiro é o *image-quality measures* (IQM), com um filtro Gaussiano (sigma = $0.5$) em áreas $3 \times 3$; o segundo grupo de atributos, é também estatísticas de alta frequência, aqui chamada de *high-order wavelet statistics* (HOWS).

## 4. Processo de Decisão

O processo de decisão é responsável pela autenticação do dispositivo investigado. Esta proposta utiliza a autenticação do tipo verificação "Um para Um" e o tipo de identificação "Um para Muitos". O primeiro tipo verifica se uma imagem investigada pertence a um certo dispositivo conhecido, ou não. Enquanto, no segundo tipo identifica para um dado conjunto de câmeras se uma imagem investigada pertence a alguma das câmeras, ou não.

A grande maioria dos autores utilizam o SVM (*Support Vector Machine*) como processo de decisão supervisionado. Os resultados alcançados são satisfatórios, como em [Gou et al. 2007a, Gou et al. 2007b, Celiktutan et al. 2008, Kang et al. 2012].

A Curva ROC (*Receiver Operating Characteristic*) é um método estatístico, comumente utilizado para classificar indivíduos em grupos. Este método é fortemente

 ©2012 SBC — Soc. Bras. de Computação

apoiado no grupo e amostras de treinamento. A quantificação da medida do erro ou da chance do algoritmo validar corretamente é o objetivo deste método, como visto em [Gou et al. 2007a, Kang et al. 2012].

As taxas de falsa rejeição (*False Rejection Rate* – FRR) e as taxas de falsa aceitação (*False Acceptance Rate* – FAR) são um dos métodos estatísticos utilizados para traçar a Curva ROC. Porém, trabalhos como de [Lukáš et al. 2006, Chen et al. 2008] tratam diretamente com estes dados para classificar os grupos com limiares adaptativos.

## 5. Considerações

Foi apresentada uma metodologia de trabalho para condução de experimentos de análise forense em imagens digitais usando como abordagem o *Sensor Pattern Noise*. A adoção de etapas distintas, claras e de simples entendimento contribuiu para manter nossa linha de pesquisa centrada. Em cada etapa foram realizadas as devidas revisões bibliográficas citando-se importantes trabalhos relacionados. A proposição do modelo entidade-relacional proporcionou maior rigor quanto ao processo de aquisição e registro das imagens utilizadas nos experimentos. Como trabalho futuro, ampliaremos a proposta de modelagem entidade-relacional para que cubra não só a etapa de armazenamento de imagens, mas também a etapa de extração de atributos e a de processo de decisão para que todo o experimento esteja registrado.

## Referências

Böhme, R., Freiling, F. C., Gloe, T., and Kirchner, M. (2009). Multimedia forensics is not computer forensics. In Geradts, Z., Franke, K., and Veenman, C., editors, *Proceedings Computational Forensics*, volume 5718 of *Lecture Notes in Computer Science*, pages 90–103, Berlin. Springer.

Celiktutan, O., Sankur, B., and Avcibas, I. (2008). Blind identification of source cell-phone model. *IEEE Trans. on Information Forensics and Security*, 3(3):553–566.

Chen, M., Fridrich, J., Goljan, M., and Lukáš, J. (2008). Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90.

Gloe, T. and Böhme, R. (2010). The dresden image database. Online. acessado em 17 de agosto de 2012.

Goljan, M., Fridrich, J., and Chen, M. (2011). Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security*, 6(1):227–236.

Gou, H., Swaminathan, A., and Wu, M. (2007a). Noise features for image tampering detection and steganalysis. In *IEEE International Conference on Image Processing*, volume 6, pages VI–97–100.

Gou, H., Swaminathan, A., and Wu, M. (2007b). Robust scanner identification based on noise features. In *Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA.

Kang, X., Li, Y., Qu, Z., and Huang, J. (2012). Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 7(2):393–402.

 ©2012 SBC — Soc. Bras. de Computação

Li, C.-T. (2010). Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287.

Lukáš, J., Fridrich, J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214.

Redi, J. A., Taktak, W., and Dugelay, J.-L. (2011). Digital image forensics: a booklet for beginners. *Multimedia Tools and Applications*, 51(1):133–162.

 ©2012 SBC — Soc. Bras. de Computação

# Além do óbvio: a análise forense de imagens e a investigação do conteúdo implícito e explícito de fotografias digitais

**Tiago Carvalho**[1], **Ewerton Silva**[1], **Filipe Oliveira Costa**[1], **Anselmo Ferreira**[1], **Anderson Rocha**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – SP – Brazil

{tjose, ewerton, filipe, ra023169, anderson.rocha}@ic.unicamp.br

***Abstract.*** *Nowadays the use of tools for manipulating images and videos is increasingly common. Such tools facilitate the task of creating manipulations and deceiving the perception of observers on the semantics of these documents. Although there are image manipulations considered innocent (*e.g.*, correction of brightness), there are those considered malicious, such as the copy-paste and composition operations. In this paper, we discuss the main challenges present in the forensic authentication of digital documents such as images and videos as well as our most recent contributions in this context.*

***Resumo.*** *Atualmente, torna-se cada vez mais comum a utilização de ferramentas para manipulação de imagens e vídeos. Tais ferramentas facilitam a criação de alterações em documentos, enganando a percepção de observadores quanto à semântica desses documentos. Apesar de existirem alterações consideradas inocentes (como uma correção de brilho), existem aquelas consideradas maliciosas como, por exemplo, operações de cópia e colagem e composição. Neste trabalho, discutimos os principais desafios tratados no cenário forense de autenticação de documentos digitais como imagens e vídeos, bem como as nossas mais recentes contribuições nesse contexto.*

## 1. Introdução

Atualmente, uma das formas mais comuns de comunicação presente na vida das pessoas é a comunicação visual, a qual retrata fatos e imortaliza momentos utilizando vídeos e imagens como meios. Graças aos avanços da tecnologia, produzir um vídeo ou uma foto torna-se mais fácil a cada dia, dado que o número de dispositivos para esse tipo de tarefa disponíveis no mercado vem aumentando significativamente e se tornando cada vez mais sofisticados.

Além do aumento no número de dispositivos para captura, o número de ferramentas para manipulação de imagens e vídeos também vem crescendo. Ferramentas de software, tais como o Adobe Photoshop, GIMP e outros, popularizam, cada vez mais, técnicas de manipulação de imagens antes empregadas apenas por pessoas com grande e conhecimento especializado. Tais ferramentas têm se tornado mais robustas e de fácil manipulação, possibilitando que usuários com um conhecimento mínimo sejam capazes de fazer grandes adulterações em imagens. Estas adulterações podem ir muito além de um simples ajuste de brilho ou a correção de pequenos defeitos na pele. Manipulações

 ©2012 SBC — Soc. Bras. de Computação

de imagens visando enganar o observador por meio da introdução de pessoas ou artefatos, de modo a criar momentos que nunca ocorreram, são hoje uma prática comum em praticamente todos os meios de comunicação.

A falsificação de imagens, de modo a representar um momento histórico que nunca existiu é quase tão antiga quanto a arte da fotografia em si. Pouco depois que o francês Nicéphore Niepce criou a primeira fotografia em 1814, já apareciam as primeiras fotografias adulteradas [Rocha et al. 2011]. Stalin, Mao, Hitler, Mussolini, Castro e Brezhnev possuíam fotografias manipuladas (fosse para criar uma aparência mais heroica, para apagar inimigos ou remover garrafas de cerveja) [Rocha et al. 2011]. Tanto as alterações dos dias de Stalin, como a maior parte das adulterações anteriores à era digital, necessitavam de alta capacidade técnica e muitas horas (talvez dias) de trabalho em salas escuras de fotografia [Popescu 2004]. Hoje, mesmo um leigo com um computador pode facilmente produzir falsificações de difícil detecção [Farid 2008].

O poder da influência de uma foto fica evidente no trabalho publicado por Sacchi et al. [Sacchi et al. 2007], em que os autores exibem os resultados de um estudo que mostra como imagens de fotografias de eventos públicos ocorridos no passado afetam a memória das pessoas em relação a estes eventos.

Todo esse conjunto de fatores torna necessário um esforço cada vez maior da comunidade forense para criar métodos capazes de detectar quaisquer tipos de adulterações em imagens, gerando, desta forma, uma "disputa armamentista" entre os métodos forenses e os métodos de adulteração. Isso faz com que nossa principal motivação neste trabalho seja contribuir com a comunidade forense na exposição dos métodos por nós desenvolvidos para a identificação de adulterações em imagens.

Nesse trabalho, apresentamos as mais recentes contribuições desenvolvidas no laboratório *Reasoning for Complex Data* (RECOD) da Universidade Estadual de Campinas (UNICAMP), algumas das quais já foram publicadas em eventos e periódicos de renome internacional. Tais contribuições abrangem os problemas de atribuição de fonte para imagens, *linking* de dispositivos, identificação de cópia e colagem bem como identificação de composições.

## 2. Estado da arte em manipulações de imagens e vídeos

A Computação Forense é uma área da computação que vem crescendo e ganhando muita importância. Segundo Delp et al. [Delp et al. 2009], ela pode ser definida como *"a coleção de técnicas científicas para a preservação, coleção, validação, identificação, análise, interpretação, documentação e preservação de evidências digitais derivadas de fontes digitais com a proposta de facilitar ou promover a reconstrução de eventos, na maioria das vezes de natureza criminal"*.

Uma subárea que vem ganhando muito espaço dentro de Computação Forense é a Análise Forense de Documentos Digitais (AFD). Rocha e Goldenstein [Rocha and Goldenstein 2010] definem AFD como *o campo de pesquisas relacionado à análise de documentos digitais para verificação de sua autenticidade e integridade*. Ela tem produzido diversos tipos de métodos focados em objetivos específicos tais como a identificação de dispositivos de origem, identificação de criações sintéticas de imagens, identificação de adulterações em documentos, entre outras.

 ©2012 SBC — Soc. Bras. de Computação

Rocha et al. [Rocha et al. 2011] indicam duas classes de adulterações podem ocorrer em uma imagem: as que melhoram a qualidade (ou Melhoria de Imagens) e aquelas que procuram enganar os observadores (Manipulação Intencional de Imagens) através da mudança da semântica da imagem. Apresentamos algumas técnicas desta última categoria e também técnicas de edição de imagens que facilitam esse processo a seguir:

- **Composição**. Consiste na união de partes de duas ou mais imagens numa única imagem. Esse tipo de falsificação pode afetar o comportamento das bordas, da iluminação, da compressão e também a do ruído do dispositivo de aquisição da imagem.
- **Ajuste fino de bordas (*feather edges*)**. Consiste no ajuste das fronteiras (bordas) dos objetos após uma operação de, por exemplo, composição, visando adequá-los às regiões da imagem onde estes se encontram e remover artefatos indesejáveis provenientes das imagens originais. Esse tipo de falsificação pode afetar o comportamento das bordas da imagem.
- **Casamento de padrões de iluminação (*light matching*)**. Auxilia no ajuste de iluminação de uma composição de forma a homogeneizar o aspecto da iluminação da imagem para torná-la verossímil. Esse tipo de falsificação pode afetar o comportamento da iluminação da imagem.
- **Realce de nitidez (*sharpening*)**. Consiste no realce de certos detalhes da imagem de forma a torná-los mais, ou menos, visíveis. Esse tipo de falsificação pode afetar o comportamento das bordas e cores da imagem.
- **Geração em computador**. Consiste na criação de objetos tridimensionais a partir de imagens ou vídeos. Aos objetos podem ser adicionadas texturas, cores, iluminação, etc. O objetivo desse processo é enriquecer os objetos com detalhes e aproximá-los de suas representações reais.
- **Cópia-colagem ou Clonagem (*cloning*)**. Consiste na alteração de partes de uma imagem usando segmentos ou propriedades da própria imagem. Esse tipo de falsificação pode afetar o comportamento de iluminação da imagem. Esse tipo de falsificação pode afetar o comportamento de iluminação da imagem.
- **Retoque e Conciliação (*retouching* e *healing*)**. Permitem o ajuste de regiões da imagem em termos de sombras, texturas, brilho, contraste, iluminação etc. Usando estas técnicas, é possível rejuvenescer uma pessoa, bem como eliminar vestígios da execução de outras operações, como cópia-colagem. Esse tipo de falsificação pode afetar o comportamento da iluminação da imagem.
- ***Lazy-Snapping***. Proposta por Li et al. [Li et al. 2004], esta técnica permite separar um objeto de interesse (*foreground*) da informação restante (*background*) em uma imagem. O método utiliza duas etapas: primeiramente o usuário define o que é fundo e o que é objeto através de curvas e, através de algoritmos de corte de grafos, a borda do objeto é computada. Na segunda etapa há a edição das bordas por parte do usuário caso o mesmo deseje corrigir erros da etapa anterior. Essa operação pode ser utilizada para selecionar de maneira precisa o objeto de uma imagem e utilizá-lo para realizar uma operação de composição.
- **Propagação Estrutural (*Structure Propagation*)**. Proposta por Sun et al. [Sun et al. 2005], essa técnica consiste em completar um espaço vazio da imagem de tal forma que o preenchimento resulte numa imagem consistente. Este método requer a especificação de informações estruturais pelos usuários através de segmentos de

 ©2012 SBC — Soc. Bras. de Computação

linha ou curvas, utilizadas para propagação de informações das regiões conhecidas para a região vazia [Rocha and Goldenstein 2010]. Após essa etapa, as regiões desconhecidas restantes são completadas com informações de textura provenientes da vizinhança, processo este que facilita operações de cópia-colagem.

- **Cópia-Colagem Baseada em Conteúdo (*Content-Aware Fill*)**: Nova ferramenta integrante do *Photoshop CS5* para cópias e colagens sofisticadas que levam o conteúdo da região analisada em questão no processo de clonagem. Nesse processo, um objeto pode ser removido da cena e a informação perdida pela sua remoção é recuperada a partir da seleção, por parte do usuário, do objeto e também de parte do fundo. A informação estrutural do fundo é então sintetizada em remendos (*patches*), que preenchem o vazio criado pela remoção da imagem.

Com essas numerosas ferramentas de manipulação de imagens, faz-se necessária a criação de metodologias de detecção eficazes da utilização das mesmas. Discutiremos esse assunto na próxima seção.

## 3. Metodologias

Uma vez que adulterações de vídeos e imagens vêm se tornando cada vez mais sofisticadas e eficazes, a necessidade de desenvolver métodos capazes de detectar tais adulterações torna-se eminente. Nesta seção, apresentamos os principais métodos desenvolvidos em nosso laboratório, os quais incrementam o estado da arte com diversas contribuições.

### 3.1. Atribuição de Fonte de Imagens

Uma forma de verificar a integridade e a autenticidade de imagens é identificar a fonte geradora da imagem em questão. Isso geralmente é feito pela detecção de "marcas" deixadas na imagem pelo dispositivo gerador no momento da captura e geração da imagem. Estas marcas são provenientes de características próprias do dispositivo gerador, como defeitos de fabricação, modo de interação entre os componentes da câmera e a luz, algoritmos de geração de imagem implementados nos componentes do dispositivo, entre outros fatores.

Pesquisas na área de atribuição de fonte em imagens digitais procuram identificar a marca ou fabricante do dispositivo utilizado na geração de uma imagem, bem como o dispositivo exato. Em particular, a tarefa de se encontrar o dispositivo específico utilizado na captura de uma imagem é a mais estudada. Existem propostas na literatura voltadas para a identificação da origem de uma imagem por meio do padrão de ruído deixado na imagem pelo dispositivo [Lukas et al. 2006,Li 2010,Goljan et al. 2008], artefatos gerados por imperfeições dos sensores de captura de um dispositivo [Kurosawa et al. 1999,Geradts et al. 2001], e presença de partículas de poeira no sensor [Dirik et al. 2008].

Identificar o dispositivo que gerou uma determinada imagem é uma forma de se garantir, por exemplo, que um documento foi gerado por uma câmera e não é resultado de qualquer manipulação digital; a informação de que uma foto foi obtida por uma câmera digital apreendida sob possessão de um suspeito poderia classificá-lo não mais como um consumidor mas sim como produtor de, por exemplo, fotos de pornografia infantil.

Atualmente, as técnicas mais efetivas para a identificação do dispositivo de captura específico analisam os efeitos do ruído inserido no processo de captura de imagens. Nossa

 ©2012 SBC — Soc. Bras. de Computação

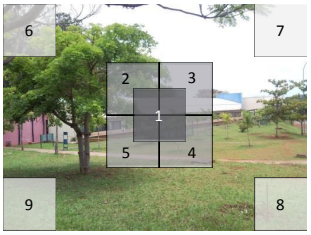



**Figura 1. Regiões de interesse (ROIs) de dimensão $512 \times 512$ *pixels*.**

abordagem para atribuição de fonte é baseada na proposta de Lukáš et al. [Lukas et al. 2006], onde os autores propõem uma maneira de se realizar a estimativa do padrão de ruído dos sensores para identificar o dispositivo gerador de uma imagem. Em suma, o ruído residual médio de várias imagens da mesma câmera é obtido, gerando um padrão de referência da câmera sob investigação. Em seguida, é efetuada a correlação entre o ruído de uma imagem a ser avaliada e o padrão de referência calculado. Se o valor dessa correlação ultrapassa um limiar determinado em uma etapa de treinamento, os autores consideram que a imagem foi obtida pela câmera referente ao padrão usado na correlação.

Embora essa abordagem seja eficaz para a identificação da câmera que gerou uma imagem sob investigação, a pesquisa foi realizada considerando um cenário fechado (*closed-set*), no qual os autores assumem que uma imagem sob investigação foi gerada por uma entre $n$ câmeras disponíveis durante a etapa de treinamento. Na prática, uma imagem a ser avaliada pode ter sido gerada por uma câmera totalmente desconhecida que não faz parte de nosso grupo de câmeras suspeitas, o que torna importante a identificação deste fato. Portanto, é importante modelar o problema de atribuição de fontes considerando um cenário aberto (*open-set*), no qual temos acesso somente a um conjunto limitado de câmeras suspeitas e temos que treinar o modelo de classificação considerando somente este conjunto enquanto buscamos classificar corretamente imagens geradas por câmeras às quais não necessariamente temos acesso. Assim sendo, propomos uma abordagem para resolver o problema de atribuição de fonte em cenário aberto, que consiste em três etapas:

**A. Definição de regiões de interesse (ROIs).** De acordo com Li e Satta [Li and Sata 2011], diferentes regiões da imagem podem ter diferentes informações sobre o padrão de ruído da câmera. Assim, visamos considerar várias regiões de uma imagem. Para cada imagem, foram extraídas nove regiões de interesse (*Regions of Interest* – ROI) de tamanho $512 \times 512$ *pixels*, de acordo com a Figura 1. A escolha dessas regiões de interesse nos permite trabalhar com imagens de diferentes resoluções.

**B. Definição de características.** Para cada região apresentada na Figura 1, nós calculamos o padrão de ruído conforme apresentado em Lukas et al. [Lukas et al. 2006], utilizando o filtro para extração de ruído no domínio da Transformada Discreta de Wavelet (DWT) proposto em [Mihcak et al. 1999], considerando os canais de cores R (vermelho), G (verde), B (azul). Foram realizados experimentos com outros espaços de cores e notamos que, para o problema de atribuição de fonte, a utilização do canal Y (luminância, do espaço de cor YCbCr [Wang and Weng 2000]) em conjunto com os canais R, G e B se mostrou eficaz.

 ©2012 SBC — Soc. Bras. de Computação

Para cada ROI, extraímos o ruído residual de cada canal de cor utilizando um filtro baseado na DWT. Em seguida, calculamos a média entre os ruídos de mesmo canal de várias imagens, gerando o padrão de ruído para cada canal de cor que representa a câmera sob investigação. Com isso, temos 36 padrões de ruído para representar uma câmera. Para cada imagem, calculamos seu ruído residual e criamos um vetor de características considerando a correlação entre cada ROI de uma imagem e o padrão de referência correspondente para cada câmera. Com essas correlações, temos 36 características para cada imagem, considerando uma câmera, rotulando imagens geradas pela câmera sob investigação como a classe positiva e as câmeras restantes disponíveis como classe negativa. Observe que algumas dessas imagens serão consideradas como sendo pertencentes à classe negativa desconhecida, ou seja, são imagens geradas por câmeras às quais não temos acesso na etapa de treinamento.

**C. Atribuição de fonte em um cenário aberto.** A principal contribuição deste trabalho é a utilização de aprendizado de máquina visando fazer a identificação da origem de imagens em um cenário aberto. Primeiramente, encontramos um classificador para treinar um conjunto de amostras considerando a classe de interesse e outras classes as quais temos acesso.O classificador escolhido se baseia no clássico algoritmo de Máquina de Vetores de Suporte (*Support Vector Machine* – SVM) [Bishop 2006] que transporta as amostras para um espeço de alta dimensão de forma que seja possível encontrara um hiperplano que faça a separação entre os dados da classe de interesse e das demais classes conhecidas.

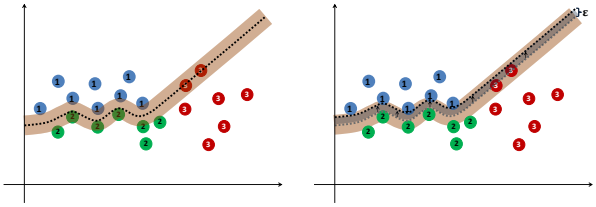
Após o cálculo do hiperplano na etapa de treinamento, propomos um meio de mas classes desconhecidas na classificação por meio da movimentação do hiperplano de decisão por um valor $\varepsilon$ se aproximando da classe positiva ou se afastando da(s) classe(s) negativa(s). A lógica é que, movendo o hiperplano, podemos ser mais restritos para o que nós sabemos como amostras positivas e, portanto, classificar qualquer outra amostra "muito diferente" como negativa, ou podemos ser pouco rigorosos sobre o que sabemos em relação às amostras positivas e aceitar pontos mais distantes do hiperplano como possíveis amostras positivas. Essa movimentação de plano tem como objetivo minimizar o erro de classificação na etapa de treinamento.

A Figura 2 descreve um exemplo para o caso não-linear. A Figura 2(a) apresenta o hiperplano de separação calculado, considerando amostras das classes azul e verde como as classes de interesse (1) e a classe negativa conhecida (2), respectivamente, e as amostras em vermelho representam as classes desconhecidas (3). A região laranja representa a distância entre as margens dos vetores de suporte das classes positiva e negativa. A Figura 2(b) mostra a operação de DBC sobre o hiperplano calculado, representado pela região azul.

Para os experimentos, nós construímos um conjunto de dados com 8500 imagens proveniêntes de 35 câmeras diferentes. As imagens foram obtidas considerando a resolução nativa das câmeras, em diversas configurações de iluminação, *zoom* e foco. O conjunto de dados é público e se encontra disponível para *download* em `http://www.recod.ic.unicamp.br/~filipe/image-source-attribution.zip`.

Nossa abordagem se mostrou bastante eficaz na identificação da câmera que capturou uma fotografia, obtendo um acerto médio de, aproximadamente, 98%, considerando

 ©2012 SBC — Soc. Bras. de Computação



(a) (b)

**Figura 2. Nossa implementação do cenário aberto para atribuição de fonte de imagens utilizando *Decision Boundary Carving* (DBC).**

um cenário onde temos acesso a 15 das 35 câmeras no treinamento, mas uma imagem a ser avaliada pode ter sido gerada por qualquer uma das 35 câmeras. Publicamos esse método recentemente em [Costa et al. 2012].

### 3.2. Detecção de Cópia-Colagem

As possibilidades proporcionadas pelas ferramentas de software atuais, aliadas à criatividade e uma pitada de intenção (boa ou má) de seus usuários, são suficientes para que imagens digitais possam passar de genuínas a fraudulentas sem muito trabalho. Dentre outras, a Cópia-colagem (ou Clonagem) é uma forma de adulteração de imagens que pode gerar resultados supreendentes por meio de um esforço reduzido.

O objetivo da clonagem é ocultar ou multiplicar elementos presentes em uma cena, tais como folhagem, objetos e pessoas. Este efeito é alcançado (i) copiando-se um segmento da imagem, (ii) aplicando-se uma transformação geométrica neste, (iii) posicionando-se o segmento em outra região da mesma imagem e (iv) aplicando-se uma operação global nesta. A Figura 3 retrata o potencial da cópia-colagem.

(a) (b)

**Figura 3. Exemplo de cópia-colagem. (a) é imagem original e (b) é a imagem manipulada.**

Identificar uma cópia-colagem consiste em investigar uma imagem suspeita em busca de segmentos idênticos ou similares. Neste último caso, transformações como *Rotação*, *Redimensionamento* (ou *Escala*), *Espelhamento* (horizontal ou vertical) e *Suavização de Bordas* podem ser empregadas para modificar a região clonada e eliminar vestígios visuais da adulteração. Adicionalmente, operações globais como *Ruídos* e *Compressão JPEG* da imagem podem ser aplicadas com o mesmo propósito.

Neste trabalho, nós apresentamos uma nova heurística para detecção de cópia-colagem. Nossa estratégia tem como base os algoritmos *PatchMatch* e *PatchMatch Generalizado*, que são utilizados na busca por correspondências de *patches* em uma ou mais

 ©2012 SBC — Soc. Bras. de Computação

imagens. A seguir, fornecemos uma breve descrição desses algoritmos e da nossa proposta para identificação de clonagens.

### 3.2.1. PatchMatch Generalizado

O *PatchMatch* [Barnes et al. 2009] é um algoritmo randomizado que se propõe a encontrar correspondências aproximadas de *patches* (blocos de *pixels* de tamanho definido, e.g., $7 \times 7$) em uma ou mais imagens por meio de um mecanismo baseado em Propagação e Busca Aleatória de correspondências na imagem. Uma correspondência (ou pareamento) para um *patch* pode ser entendida como um *patch* similar a este dentro da imagem.

Dando continuidade ao método original, em [Barnes et al. 2010] Barnes et al. propõem o *PatchMatch Generalizado*. O método consiste em averiguar a imagem de entrada de maneira similar à efetuada no *PatchMatch*, com a diferença de que, nesta abordagem, são encontradas não apenas uma, mas $K$ correspondências para cada *patch* existente na imagem. Além disso, novas etapas (Enriquecimento Direto e Inverso) contribuem positivamente nos resultados. Essas etapas visam propagar boas correspondências de um *patch* para o espaço de correspondências desse mesmo *patch* (em oposição à etapa de propagação, que efetua tal difusão através das dimensões espaciais da imagem). Logo, o método passa a considerar um conjunto mais rico de pareamentos satisfatórios que podem aprimorar o NNF.

O *PatchMatch Generalizado* pode acomodar diversas métricas de similaridade entre *patches* e é facilmente adaptável para o emprego de descritores de imagens, tais como SIFT [Lowe 1999] e SURF [Bay et al. 2006].

### 3.2.2. Detecção de Cópia-colagem usando o PatchMatch Generalizado

Para esse problema, apresentamos uma heurística para identificação de clonagens que diverge daquelas existentes na literatura. Primeiramente, dada uma imagem suspeita, nós aplicamos o *PatchMatch Generalizado* com o propósito de encontrar, para cada um de seus *patches*, um conjunto de correspondências. Estas são tomadas como potenciais candidatas à duplicação dentro da própria imagem. Examinamos cada conjunto de *patches* candidatos a partir da verificação da vizinhança em que estes se encontram.

Dado que o *PatchMatch Generalizado* comporta medidas de similaridade diversas, nós adotamos uma métrica baseada em comparação de histogramas de intensidade/cor. A razão para isto é que a SSD não é robusta a pequenas variações nas regiões duplicadas. Para imagens em tons de cinza, computamos os histogramas de intensidade de ambos os *patches* sendo inspecionados e efetuamos o cálculo da Soma das Diferenças Absolutas (*Sum of Absolute Differences* – SAD) entre os bins de mesma posição. O valor desta soma é a distância de similaridade entre os dois *patches*. Caso a imagem possua informação de cor, calculamos os histogramas de cada canal (R, G e B) e cada *patch* passa a ser descrito por 3 histogramas. Em seguida, realizamos os cálculos de SAD para os histogramas de mesmo canal e aplicamos os valores encontrados na fórmula para conversão de cor em intensidade conforme $I = 0.299R + 0.587G + 0.114B$, na qual $I$ representa a distância de similaridade. Descrevemos o método a seguir e o esquematizamos na Figura 4.

 ©2012 SBC — Soc. Bras. de Computação

1. Empregamos o *PatchMatch Generalizado* na busca pelas $K$ correspondências para cada *patch* da imagem. A inicialização do NNF é aleatória;
2. Percorremos o NNF final obtido em *scan order*;
3. Visitamos a lista de correspondências de cada *patch* $(x, y)$;
4. Caso uma correspondência $(x_i, y_i)$, sendo $i \leq K$, de $(x, y)$ esteja a uma distância física deste inferior a um limiar $T$, não prosseguimos com a análise e passamos para a correspondência seguinte;
5. Comparamos a região ao redor do *patch* $(x, y)$ com a região ao redor de cada uma de suas correspondências $(x_i, y_i)$. Esta região de comparação abrange o *patch* mais um acréscimo de dois *pixels* em todas as direções. Caso um *patch* não possa crescer em quaisquer direções (borda da imagem), a mesma restrição se aplicará ao outro *patch* em comparação;
6. Se as duas regiões (ao redor de $(x, y)$ e $(x_i, y_i)$) forem similares, isto é, se a distância de similaridade entre elas for menor do que um limiar $D$, marcamos ambas as regiões como duplicadas;
7. Se uma região já foi marcada, ela não é examinada novamente.

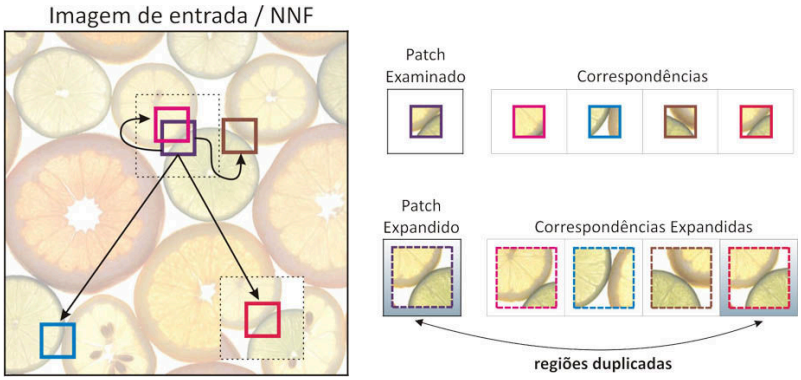



**Figura 4. Esquematização da nossa proposta para detecção de clonagens baseada no *PatchMatch Generalizado*. O método examina todos os conjuntos de correspondências do NNF em busca daquelas que se encontram em uma vizinhança similar à vizinhança de um *patch*$(x, y)$. Na Figura, a vizinhança do *patch* de borda roxa sendo avaliado é similar à vizinhança do *patch* de borda vermelha (duplicação).**

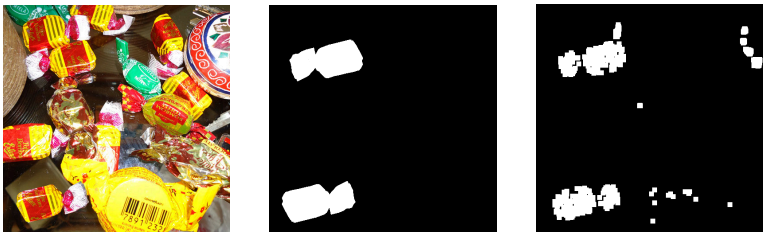
Com esta abordagem, nós conseguimos contemplar, além de clonagens sem transformações adicionais, operações de espelhamento horizontal e vertical e rotações de 90, 180 e 270 graus porventura aplicadas nos segmentos duplicados. Este resultado é alcançado em decorrência da utilização de histogramas de intensidade/cor, que são invariantes a tais operações. Acreditamos que o método possui elevado potencial para detecção de clonagens nos cenários de operações de escala, suavização, compressão JPEG e de rotações em graus diversificados. Uma possível solução para isto seria a utilização de descritores SIFT ou SURF e a execução de modificações na ideia central do algoritmo.

A Figura 5 mostra um exemplo de clonagem e o resultado de detecção encontrado pelo algoritmo proposto.

### 3.3. Detecção de Composições (*Splicing*) em Imagens

Uma das formas mais comuns de adulteração de imagens são as realizadas através de uma operação conhecida como composição. Consiste na construção de uma nova ima-

 ©2012 SBC — Soc. Bras. de Computação



(a) (b) (c)

**Figura 5. Resultado de uma detecção positiva utilizando o algoritmo proposto. (a) é a cópia-colagem, (b) é o mapa de referência que mostra as regiões duplicadas (em branco) e (c) é o resultado da detecção usando o nosso método.**

gem utilizando partes do conteúdo de outras. Para identificar este tipo de adulteração, os métodos desenvolvidos pela comunidade forense baseiam-se em diversos tipos de características: inconsistências em descritores [Popescu 2004], inconsistências no processo de aquisição [Lin et al. 2005], inconsistências no processo de compressão [Luo et al. 2010] e inconsistências no processo de iluminação [Kee and Farid 2010].

Em especial, as abordagens baseadas em inconsistências de iluminação vêm ganhando espaço no cenário forense. Este tipo de abordagem destaca-se uma vez que um ajuste perfeito de iluminação em uma composição digital é extremamente difícil de se obter, uma vez que ao se realizar a composição utilizando partes de duas ou mais imagens, o fato de cada uma delas ser obtida em uma condição de iluminação diferente dificulta a criação deste tipo de falsificação. Outra vantagem desta classe de métodos é que eles podem ser utilizados para analisar imagens analógicas [Rocha et al. 2011].

A seguir, apresentamos uma de nossas contribuições para detecção de composições utilizando inconsistências de iluminação. O trabalho foi publicado em 2011 [Saboia et al. 2011] e utiliza os reflexões da luz nos olhos de pessoas para detectar falsificações em imagens contendo pessoas.

### 3.3.1. Eye Specular Highlights Telltales For Digital Forensics: A Machine Learning Approach

Nosso trabalho para detecção de composições através de inconsistências de iluminação presentes na imagem é baseado em uma extensão do artigo proposto por Johnson e Farid [Johnson and Farid 2007]. Nele, os autores se baseiam no fato de que a posição dos raios de luz refletidos nos olhos em imagens contendo pessoas é determinada pela posição relativa da fonte de luz, da superfície de reflexão do olho e do visualizador (neste caso, a câmera).

A Figura 6 exibe os três estágios que compõem o método original. O primeiro estágio estima a direção da fonte de luz para cada olho presente na fotografia de pessoas em análise. O segundo estágio (caracterização) procura estimar a posição da fonte de luz da imagem baseado nos raios de luz refletidos nos olhos e na correspondente fonte de luz estimada para cada olho. A posição calculada da fonte de luz é utilizada para calcular o erro angular para cada raio de luz. Finalmente, o terceiro estágio (decisão) calcula o erro angular médio e utiliza o teste de hipótese clássico com 1% de significância para decidir

 ©2012 SBC — Soc. Bras. de Computação

se uma imagem sob investigação é uma composição.

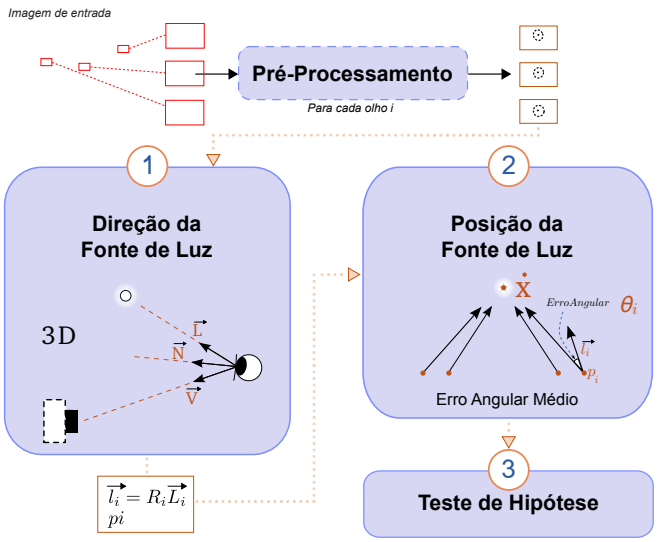



**Figura 6. Diagrama mostrando os três estágios do método original proposto por Johnson e Farid [Johnson and Farid 2007]. Imagem retirada de [Saboia et al. 2011].**

Procuramos por características mais robustas além das já propostas na literatura. No trabalho original de Johnson e Farid [Johnson and Farid 2007], os autores levaram em consideração apenas as informações referentes à iluminação da cena. No entanto, descobrimos que uma outra informação, antes ignorada, também era de grande importância para a identificação de manipulações. Tal informação diz respeito à posição do visualizador (neste caso, o visualizador é a câmera fotográfica) da cena. Essa informação, juntamente com a posição da fonte de luz, foi utilizada para descrever a imagem.

Outra característica importante é que, dado que os métodos que estimam a posição do observador e da fonte de luz são não determinísticos, eles podem produzir uma pequena variação em seu resultado. Nós exploramos tal variação para produzir diferentes caracterizações para cada imagem.

Uma vez que cada imagem possuía diferentes caracterizações (em nossos experimentos, utilizamos 5 descritores por imagem), foi possível substituir o teste de hipótese clássico da etapa de classificação por uma combinação de classificadores, em que cada um dos vetores de características era classificado por um classificador independente (em nossos experimentos, utilizamos o SVM [Bishop 2006]). Isso produziu classificações diferentes para uma mesma imagem. Por fim, combinamos os resultados dessas classificações para obter a classificação final.

Como principais contribuições decorrentes deste trabalho, podemos citar:

- um resultado com uma taxa de erro na classificação 20% menor que o trabalho original;
- a proposta de novas características não levadas em conta pelos autores originais.

A Figura 7 mostra um exemplo de composição e o resultado de detecção encontrado pelo algoritmo proposto.

 ©2012 SBC — Soc. Bras. de Computação

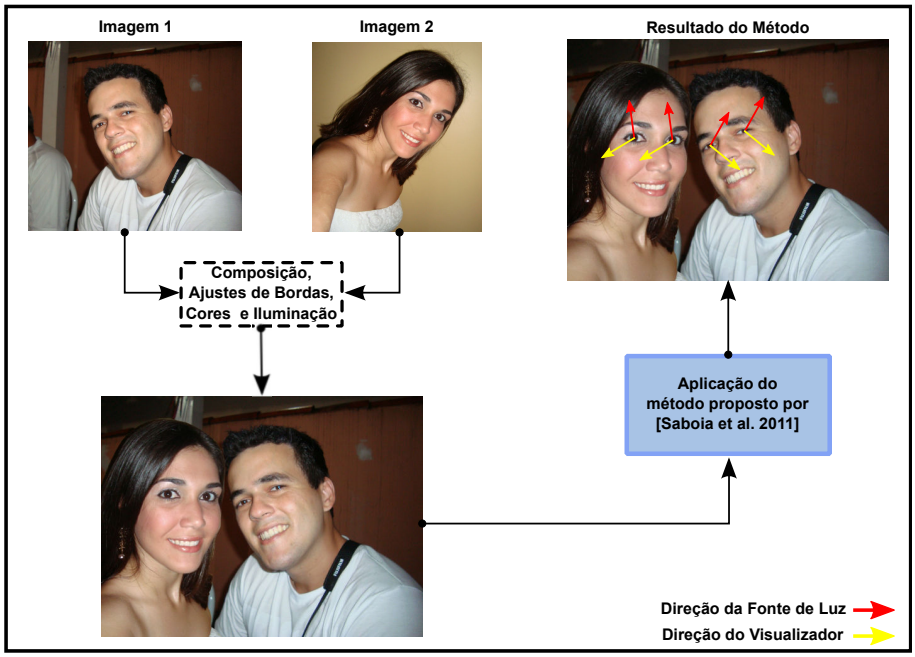



**Figura 7. Resultado de uma detecção positiva utilizando o método proposto por [Saboia et al. 2011]. As direções estimadas para o visualizador e para a fonte de luz de cada uma das pessoas apontam em direções opostas, caracterizando uma imagem de composição.**

## 4. Conclusões e Trabalhos Futuros

No cenário forense atualmente, existem diversos problemas em aberto esperando por soluções eficientes e eficazes. Neste trabalho, discutimos nuances de tais problemas, bem como algumas contribuições desenvolvidas em nosso laboratório na Universidade Estadual de Campinas para solucioná-los.

A atribuição de fonte de dispositivos tem como objetivo identificar qual foi o dispositivo gerador de uma imagem. Nesse contexto, nós exploramos soluções para o problema de atribuição de fonte de imagens geradas por câmeras digitais, uma tarefa de fundamental importância em um cenário criminal. Consideramos um cenário mais realístico, denominado cenário aberto, onde uma imagem sob investigação pode ter sido gerada por qualquer dispositivo, e não somente pelos dispositivos disponíveis no momento do treinamento. A abordagem proposta apresentou bons resultados, e com ela é possível analisar imagens de diferentes resoluções. Além disso, podemos identificar a fonte de imagens considerando métodos de caracterização complementares, tirando vantagem de todos os potenciais métodos de classificação de padrões por aprendizado de máquina.

O desafio da detecção de cópia-colagem em imagens digitais ainda possui diversas lacunas a serem preenchidas. Contemplar operações de rotação e escala tem sido uma das principais frentes de estudo dos pesquisadores acerca do tema. Acreditamos que a nossa metodologia, baseada no mecanismo aleatório e de propagação de correspondências do *PatchMatch Generalizado*, contribui com a literatura ao apresentar uma heurística inovadora para identificação de clonagens. Uma extensão direta do método proposto englobaria a combinação de descritores invariantes àquelas operações, tais como SIFT e SURF, para caracterização dos *patches* inspecionados.

 ©2012 SBC — Soc. Bras. de Computação

Finalmente, a detecção de composições em imagens utilizando inconsistências de iluminação é uma das formas mais promissoras existentes atualmente. No entanto, o atual estado da arte, incluindo nossa contribuição, ainda trata o problema baseando-o em diversas premissas, as quais muitas vezes restringem o cenário de aplicação do método a cenários específicos, como imagens contendo pessoas por exemplo.

Mais especificamente, nosso método apresentado para detecção de composições é aplicável, apenas, em imagens contendo duas ou mais pessoas, e que tais pessoas estejam com os olhos visíveis. Assim, torna-se necessário o desenvolvimento de métodos capazes de utilizar inconsistências de iluminação para detectar composições em qualquer tipo de cenário, ou pelo menos com um número menor de restrições. Para isso, como trabalhos futuros abordaremos uma nova forma de representação menos restritiva para a iluminação da cena tentando desta forma aumentar o domínio de aplicação de nossos futuros métodos.

## Agradecimentos

Agradecemos à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Microsoft Research, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento dessa pesquisa.

## Referências

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM ToG*, pages 24:1–24:11.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2010). The generalized patchmatch correspondence algorithm. In *ECCV*, pages 29–43.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *ECCV*, pages 404–417.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning, 1st edition*. Springer.

Costa, F. O., Eckmann, M., Scheirer, W. J., and Rocha, A. (2012). Open set source camera attribution. In *SIBGRAPI*, pages 71–78.

Delp, E., Memon, N., and Wu, M. (2009). Digital Forensics [From the Guest Editors]. *IEEE SPM*, 26:14–15.

Dirik, A. E., Sencar, H. T., and Memon, N. (2008). Digital single lens reflex camera identification from traces of sensor dust. *IEEE T.IFS*, 3(3):539–552.

Farid, H. (2008). Digital Image Forensics. *Scientific American*, 6:66–71.

Geradts, Z. J., Bijhold, J., Kieft, M., Kurosawa, K., Kuroki, K., and Saitoh, N. (2001). Methods for identification of images acquired with digital cameras. *Enabling Technologies for Law Enforcement and Security*, 4232:505–512.

Goljan, M., Fridrich, J., and LukÃ¡š, J. (2008). Camera identification from printed images. In *SPIE Conference on Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, volume 6819.

 ©2012 SBC — Soc. Bras. de Computação

Johnson, M. and Farid, H. (2007). Exposing Digital Forgeries Through Specular Highlights on the Eye. In *Information Hiding*, pages 311–325.

Kee, E. and Farid, H. (2010). Exposing Digital Forgeries from 3-D Lighting Environments. In *IEEE WIFS*, pages 1–6.

Kurosawa, K., Kuroki, K., and Saitoh, N. (1999). CCD fingerprint method – identification of a video camera from videotaped images. In *IEEE ICIP*, pages 537–540.

Li, C.-T. (2010). Source camera identification using enhanced sensor pattern noise. *IEEE T.IFS*, 5(2):280–287.

Li, C.-T. and Sata, R. (2011). On the location-dependent quality of the sensor pattern noise and its implication in multimedia forensics. In *ICDP*, pages 1–6.

Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. (2004). Lazy snapping. *ACM ToG*, pages 303–308.

Lin, Z., Wang, R., Tang, X., and Shum, H. (2005). Detecting Doctored Images Using Camera Response Normality and Consistency. In *IEEE CVPR*, pages 1087–1092.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*, page 1150.

Lukas, J., Fridrich, J., and Goljan, M. (2006). Digital Camera Identification from Sensor Pattern Noise. *IEEE T.IFS*, 2:205–214.

Luo, W., Huang, J., and Qiu, G. (2010). JPEG Error Analysis and Its Applications to Digital Image Forensics. *IEEE T.IFS*, 5:480–491.

Mihcak, M. K., Kozintsev, I., Ramchandran, K., and Moulin, P. (1999). Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303.

Popescu, A. (2004). *Statistical Tools for Digital Image Forensics*. PhD thesis, Department of Computer Science, Dartmouth College.

Rocha, A. and Goldenstein, S. (2010). CSI: Análise Forense de Documentos Digitais. *Atualizações em Informática (JAI)*, pages 263–317.

Rocha, A., Scheirer, W., Boult, T., and Goldenstein, S. (2011). Vision of the Unseen: Current Trends and Challenges in Digital Image and Video Forensics. *ACM Computer Survey*, 43(4):1–42.

Saboia, P., Carvalho, T., and Rocha, A. (2011). Eye Specular Highlights Telltales for Digital Forensics: a Machine Learing Approach. In *IEEE ICIP*, pages 1937–1940.

Sacchi, D., Agnoli, F., and Loftus, E. (2007). Changing History: Doctored Photographs Affect Memory for Past Public Events. *Applied Cognitive Psychology*, 21(8):1005–1022.

Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. (2005). Image completion with structure propagation. *ACM ToG*, pages 861–868.

Wang, X. and Weng, Z. (2000). Scene abrupt change detection. In *Canadian Conference on Electrical and Computing Engineering*, pages 880–883.

 ©2012 SBC — Soc. Bras. de Computação

# Identificação de Autoria *Offline* em Documentos Manuscritos

**Aline Maria Malachini Miotto Amaral[1], Cinthia O. A. Freitas[2], Flavio Bortolozzi[1]**

[1]Departamento de Informática, Centro Universitário de Maringá (CESUMAR)
Maringá – PR – Brazil

[2]Escola Politécnica, Pontifícia Universidade Católica do Paraná (PUCPR)
Curitiba, PR, Brazil

{amiotto75,flavio.bortolozzi.53}@gmail.com.br, cinthia@ppgia.pucpr.br

***Abstract.*** *This paper aims to present the state of the art related to writer identification in offline handwritten documents. For this purpose, we analyze the following elements of each approach studied: granularity and features extracted for the identification process; the basis used for the experiments validation; classification algorithm used and the identification rate obtained. It can be observed that approaches which use textural characteristics (at the document level) have very high rates, however requires a high computational power. While the approaches which perform direct image manipulation, pixel by pixel, require less computational resources, but have lower rates.*

***Resumo.*** O objetivo deste trabalho é apresentar o estado da arte relacionado à identificação de autoria *offline* em documentos manuscritos. Para tanto, foram analisados os seguintes elementos de cada abordagem estudada: granularidade e características extraídas para o processo de identificação, base utilizada para validação dos experimentos de identificação, algoritmo de classificação utilizado e taxa de reconhecimento. Pode-se observar que abordagens que utilizam características texturais (em nível de documento) apresentam taxas muito elevadas, no entanto exigem um alto poder computacional. Enquanto que as abordagens que realizam manipulação direta, ou seja, pixel a pixel da imagem, exigem menos recursos computacionais, mas possuem taxas de acerto com valores reduzidos.

## 1. Introdução

O estudo dos aspectos fisiológicos envolvidos na escrita humana é alvo de pesquisas (Morris, 2000; Cavalcanti; Lira, 1996; Mendes, 2003), as quais têm por o objetivo identificar as características que permitem definir a individualidade de uma pessoa por meio de sua escrita. Os resultados destes estudos são de interesse de várias outras áreas de pesquisa, como por exemplo, a psicologia (análise do perfil de uma pessoa com base em sua escrita) e o direito (análise da autenticidade de documentos manuscritos com base em sua escrita).

De acordo com Mendes (2003), a "documentoscopia é a parte da criminalística que estuda os documentos para verificar se são autênticos e, em caso contrário, determinar a sua autoria". Dentro deste contexto, o mesmo autor indica que "a

 ©2012 SBC — Soc. Bras. de Computação

grafoscopia refere-se à parte da documentoscopia que estuda as escritas com a finalidade de verificar se são autênticas e, em caso contrário, determinar a sua autoria". Por se tratar de uma área na qual predomina a subjetividade no uso da técnica pericial ou grafometria, a mesma tem se tornado campo de interesse da computação, visto esta área constituir meio para propor e experimentar métodos e técnicas que possam auxiliar na automatização e padronização de procedimentos tradicionalmente aplicados pelos peritos (Hanusiack, 2010).

Atualmente, o processo de análise de manuscritos é realizado pela perícia por meio de métodos manuais (aparelhamento ótico) e/ou químicos. De acordo com Sheikholesmani et al. (1996), a extração manual de características de manuscritos é tediosa e sujeita a erros. Além disso, diferentes grafotécnicos podem extrair as mesmas características do mesmo do manuscrito de formas diferentes. É também possível que o conteúdo do manuscrito influencie a análise grafoscópica. Umas das alternativas para auxiliar os grafotécnicos a superarem estes problemas é a automatização de todo ou parte do: processo de extração; análise de características e reconhecimento da escrita humana.

Diferentes abordagens para identificação de autoria em documentos manuscritos vêm sendo apresentadas na literatura, cada uma delas focando em aspectos específicos da escrita, bem como em aspetos referentes à imagem dos manuscritos.

Dentro do contexto acima descrito, o objetivo deste trabalho é apresentar o estado da arte relacionado à identificação de autoria *offline* em documentos manuscritos. Para tanto, foram analisados os seguintes elementos de cada abordagem estudada: granularidade e características extraídas para o processo de identificação, base utilizada para validação dos experimentos de identificação, algoritmo de classificação utilizado e taxa de reconhecimento.

Este artigo está organizado da seguinte forma: na Seção 2 uma introdução à identificação de autoria em documentos manuscritos é apresentada. Na Seção 3 são apresentados os trabalhos sobre identificação de autoria *offline* estudados. Na Seção 4 é apresentada uma classificação das abordagens estudadas. Finalmente, na Seção 5 são apresentadas as considerações finais desta pesquisa.

## 2. Identificação de Autoria em Documentos Manuscritos

De acordo com Bulacu et al. (2007), dois importantes fatores naturais estão em conflito direto na tentativa de identificar uma pessoa com base em amostras de manuscritos: variações entre diferentes escritores (variabilidade interpessoal) em contraposição a variabilidade na escrita de um único escritor (variabilidade intrapessoal). Nesse contexto, as abordagens automáticas para identificação de autoria consistem na extração de representações computacionais (primitivas) com o objetivo de maximizar a separação entre diferentes escritores, enquanto apresentam um padrão de escrita nas amostras do mesmo escritor.

Segundo He et al. (2008), abordagens para identificação de autoria podem ser classificadas de diferentes formas, contudo a mais simples e direta é a divisão em *online* e *offline* que se refere ao processo pelo qual o manuscrito é capturado para sua posterior análise.

 ©2012 SBC — Soc. Bras. de Computação

Abordagens para identificação de autoria *offline* utilizam manuscritos digitalizados após a escrita ter sido realizada, sendo obtidos arquivos que correspondem a representações de imagens bidimensionais. Segundo Said et al. (2000), apesar dos contínuos esforços, a pesquisa na área de sistemas para identificação de autoria *offline* ainda é uma questão desafiadora.

Nas abordagens *online* um equipamento é conectado ao computador, o qual converte o movimento de escrita em uma sequencia de sinais e os envia ao computador conectado (He et al., 2008). Este trabalho focou no estudo e classificação apenas de abordagens de identificação autoria *offline,* uma vez que as abordagens *online* não auxiliam análises grafoscópicas que visam encontrar o escritor de um manuscrito já existente (He et al., 2008).

Outras categorizações que levam em consideração o tipo de características extraídas dos manuscritos são encontradas na literatura. Uma classificação bastante interessante é apresentada no trabalho de Sreeraj e Idicula (2011) na qual as abordagens *offline* e *online* são classificadas de acordo com o nível de granularidade da característica extraída (por exemplo: documento, parágrafo, linhas, palavras e caracteres). .

Neste trabalho adotou-se dois níveis de granularidade para categorizar as características para identificação de autoria *offline*. Considera-se como características globais, aquelas que envolvam a extração de informações em nível de documento, parágrafo e linhas, enquanto que aquelas que envolvam a extração de informações em nível de palavra e caracter serão consideradas como locais. Para uma melhor compreensão a Figura 1 apresenta este esquema de classificação.

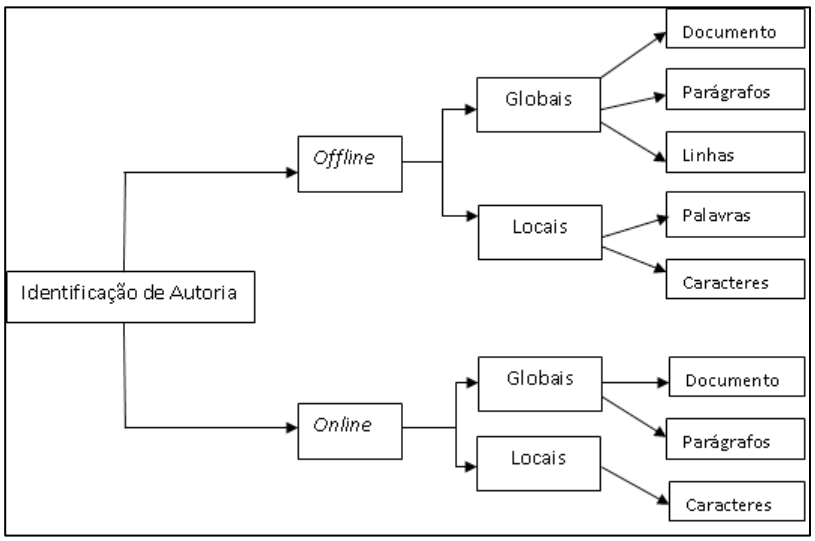



**Figura 1. Classificação para abordagens de identificação de autoria [Fonte: Adaptado de (SREERAJ; IDICULA, 2011)]**

## 3. Abordagens para Identificação de Autoria *Offline*

Nas próximas seções são apresentadas as abordagens estudadas, categorizadas de acordo com a classificação demonstrada na Figura 1.

 ©2012 SBC — Soc. Bras. de Computação

### 3.1. Abordagem Global

***a) Características Extraídas do Documento***

O trabalho de Said et al. (1998) propõe uma abordagem independente de texto para identificação de escrita que deriva de características relativas a textura usando filtros de Gabor e Matrizes de co-ocorrência de Escalas de Cinza. Resultados com taxas de 96% de acerto foram obtidos utilizando 150 documentos teste de 10 diferentes escritores. Trabalhos similares podem ser observados em (Tan, 1998; Zhu et al., 2001).

Said et al. (2000) apresentam uma abordagem global baseada na análise de textura, na qual cada manuscrito de um escritor é considerado como uma diferente textura. Isto permitiu aos autores aplicar algoritmos padrões de reconhecimento de texto para a tarefa de identificação (por exemplo, a técnica de filtragem de multicanais de Gabor). Resultados de 96% de taxas de acerto foram obtidos em documentos teste de 40 diferentes escritores.

Ainda utilizando características relativas à textura, o trabalho de Helli e Morghaddam (2010) apresenta um método independente de texto para identificação de autoria na escrita Persa. Este método é baseado em características que são extraídas do manuscrito utilizando filtros de Gabor e de XGabor. As características extraídas de cada escritor são inseridas em um grafo de características relacionadas. Este grafo é construído usando relações entre as características extraídas por meio da aplicação do método *fuzzy*. Os resultados experimentais apresentaram desempenho próximo a 100% de acerto para uma base de dados de 100 escritores.

He et al. (2008) apresentam um método para identificação de autoria em manuscritos Chineses independente de texto baseado no modelo de árvores escondidas de Markov (HMT - *Hidden Markov Tree*). De acordo com os autores do trabalho, os resultados obtidos em seus experimentos apresentaram taxas de reconhecimento melhores quando comparados com modelos que utilizam filtros de Gabor multidimensionais e também reduziram significativamente o tempo de computação necessário para a realização dos experimentos.

Hanusiak et al. (2010) apresentam um método para análise e identificação da autoria de manuscritos tendo como base os atributos genéticos e genéricos da escrita, componentes fundamentais para a identificação da autoria de manuscritos. Segundo os autores deste trabalho, o processo definido não exige correções na inclinação das linhas de texto e dispensa os processos de correção dos alinhamentos entre palavras de uma mesma linha. Das imagens de textura geradas são calculadas as matrizes de co-ocorrência, as quais são utilizadas em conjunto com os descritores de textura de Haralick (1973). Os resultados experimentais com taxas de acerto em torno de 95% em uma base de dados composta por mais de 300 escritores demonstram que o método de segmentação proposto é capaz de manter as características genéricas e genéticas dos escritores.

O trabalho de Bulacu et al. (2007) apresenta uma combinação de características globais baseadas: na extração de características relacionadas a textura de um manuscrito; e em características alográficas. Segundo estes autores por meio da junção de tais características foi possível obter taxas de acerto de 82% tanto para o processo de identificação quanto para o processo de verificação de autoria.

Baranoski et al. (2007) propõem um método de extração global de características que utiliza a primitiva grafométrica denominada de inclinação axial. Esta primitiva

 ©2012 SBC — Soc. Bras. de Computação

representa o ângulo de inclinação da escrita em relação ao eixo vertical sendo o eixo horizontal representado por uma linha de base imaginária. Com base em uma análise de todo o documento é utilizada uma distribuição de tendências angulares de inclinação, o que faz com que a avaliação seja similar a utilizada pelo perito forense. Foram realizados experimentos com 200 escritores distintos para a fase de treinamento e 115 escritores distintos para a fase de teste, obtendo resultados próximos a 95% de acerto para verificação de autoria.

Karunakara e Mallikarjunaswamy (2011) propuseram uma abordagem para identificação de escrita, dependente de texto, na língua inglesa usando o modo de decomposição empírica (EMD - *Empirical Mode Decomposition*). Para estes autores, cada manuscrito de um escritor é considerado como uma diferente textura. Estas texturas são consideradas como uma única característica para a identificação de escrita. Os resultados obtidos atingiram taxas de acerto em torno de 94% com o classificador KNN e utilizando características registradas de manuscritos de 50 diferentes escritores.

Amaral et al. (2012) propõem um método para identificação de escrita, dependente de texto, na língua Portuguesa levando em consideração os hábitos de uso/posicionamento do texto na folha de papel. Segundo os autores deste trabalho, alguns escritores podem fazer um bom uso da folha de papel, escrevendo até seus limites físicos, enquanto que outros podem deixar espaços em branco, usualmente regulares em todas as linhas. Diferentes escritores iniciam e terminam suas escritas em diferentes posições. Assim, localizações tais como identação de sentenças, espaçamentos das margens, uso de espaços, pontos iniciais e finais são exemplos de "posicionamento do texto na folha". Neste trabalho foram extraídas características como: número de linhas e espaço percentual das margens superior, inferior e esquerda. Como algoritmo de classificação foi utilizado o SVM (*Support Vector Machine*), e a melhor taxa de acerto obtida foi de 80% em um base composta por 20 diferentes escritores.

### *b) Características Extraídas dos Parágrafos*

Siddiqi e Vincent (2008) propõem um método que leva em consideração as formas freqüentes do traçado tal como definido no trabalho de Bensefia et al. (2005). A abordagem de Bensefia utiliza a forma como as letras são desenhadas e segmentadas como se o objetivo desta tarefa fosse ler o texto. No entanto, no trabalho de Siddiqi e Vincent (2008) o reconhecimento do escritor é independente do que está escrito e se refere à forma com que fisicamente as linhas ou *loops* são produzidos.

No trabalho de Bensefia et al. (2005) o documento manuscrito é dividido em um grande número de sub-imagens, e imagens morfologicamente similares são agrupadas em *clusters*. Com base em tais *clusters*, os padrões que ocorrem com frequência para um escritor são então extraídos. Nesse contexto, o escritor do documento questionado é definido encontrando-se a classe de estilo de escrita a qual seu documento pertence. Foram utilizados nos experimentos duas bases de dados a IAM (*Informatik und Angewandte Mathematik*) (650 escritores) e a RIMES (*Reconnaissance et Indexation de données Manuscrites et de fac similÉS / Recognition and Indexing of handwritten documents and faxes*) (375 escritores) e taxas de acerto em torno de 91% foram atingidas.

Schomaker et al. (2007) descrevem em seu trabalho uma abordagem para identificação de autoria usando contornos de fragmentos de componente conectados a partir de manuscritos redigidos em estilo livre de escrita. O escritor é caracterizado

 ©2012 SBC — Soc. Bras. de Computação

como um gerador estocástico de padrões, produzindo uma família de fragmentos de caracteres (chamados de *fraglets*). Usando um *codebook* desses *fraglets* a partir de uma base de treinamento independente, a distribuição de probabilidades dos *fraglets* é computada para a base de teste. Resultados revelaram uma alta sensibilidade do histograma dos *fraglets* para a identificação de escritores individuais. Experimentos em grande escala utilizando redes de Kohonen foram realizados, e os melhores resultados obtidos chegaram a atingir 97% de acerto.

Kumar et al. (2011) apresentam um método para extrair informações manuscritas e de zonas de textos impressos de imagens de documentos com conteúdo mistos. Neste trabalho foram usadas características que codificam formas locais de texto. Foram construídos dois *codebooks* das características de forma extraídas de um conjunto de manuscritos e dos textos de documentos impressos respectivamente. Para o processo de classificação foi utilizado o classificador SVM. Taxas de reconhecimento de 98% foram obtidas, tendo sido utilizadas 732 imagens para a fase de treinamento e 625 imagens para a fase de teste.

***c) Características Extraídas das Linhas***

Luna et al. (2011) propõem uma abordagem para resolver o problema de identificação de autoria de manuscritos com uma modificação no algoritmo da família de classificadores supervisionados ALVOT (*Votacion Algorithm*). A metodologia proposta neste trabalho apresenta e exemplifica a possibilidade de identificação de autoria por meio de uma abordagem intermediária entre características de textura e estruturais. Características tanto em nível de linha quanto em nível de palavra são extraídas da imagem do manuscrito. Em nível de linha o espaço percentual da margem esquerda e margem direita, a separação entre linhas, a direção geral da escrita e o espaço entre palavras são considerados.

No trabalho de Luna et al. (2011) um esquema não diferencial de pesos tradicional atribui um valor de peso para cada uma das características extraídas. Este mecanismo fornece flexibilidade suficiente para precisamente discriminar padrões pertencendo a classes onde o mesmo subconjunto de características é mais relevante, mas com diferentes proporções em cada caso. Foram realizados experimentos com 30 escritores atingindo resultados próximos a 92%.

O trabalho de Bensefia et al. (2005) propõe-se a demonstrar que a tarefa de identificação e verificação de autoria pode ser realizada usando características locais como grafemas extraídos a partir da segmentação de manuscritos. Um modelo de recuperação baseado em texto é usado para o estágio de identificação de autoria. Isto permite o uso de um espaço de características particulares baseado nas freqüências das características. Documentos questionados são projetados neste espaço de características. Neste trabalho taxas de 95% de acerto foram obtidas na base PSI e 86% na base IAM. De acordo com Bensefia et al. (2005) o problema de identificação de autoria pode ser definido como um processo de encontrar conteúdos gráficos (conjuntos de grafemas do documento a ser identificado) em um grande conjunto de documentos (base de dados de treinamento). Os documentos recuperados devem ser classificados de acordo com uma similaridade com o documento questionado.

No trabalho de Schlapbach e Bunke (2004) um sistema independente de texto para identificação de autoria usando Modelos Escondidos de Markov (HMM - *Hidden*

 ©2012 SBC — Soc. Bras. de Computação

*Markov Models*) foi proposto. Para cada escritor conhecido foi construído um reconhecedor individual e treinado o mesmo com linhas de texto escritas por este escritor (cada linha de texto apresentada ao sistema é normalizada com relação à inclinação, altura e obliquidade e transformada em um vetor de primitivas). Uma linha de texto de origem desconhecida é apresentada para cada um destes reconhecedores identificados. Como resultado o reconhecedor conhecido que tiver maior *score* é definido como sendo o escritor da linha de texto cujo escritor é desconhecido. O sistema foi testado como 2200 linhas texto de 50 escritores e teve um taxa de reconhecimento de 94.4%.

Hertel e Bunke (2003) apresentam um sistema para identificação de escrita com características derivadas das linhas de um manuscrito (continuidade do traçado, regiões fechadas, contornos superiores e inferiores). Estas características são subsequentemente usadas no classificador KNN que compara o vetor de características extraídas de um texto de entrada com um conjunto de vetores protótipos que pertencem a escritores com identidade conhecida. Este método foi testado com uma base de dados com páginas de textos manuscritos produzidas por 50 escritores. Taxas de reconhecimento próximas a 90% foram atingidas usando uma única linha do texto como entrada. A taxa de reconhecimento aumenta para quase 100% se a página toda do texto é fornecida como entrada para o sistema.

No trabalho de Chen et al. (2010) foi desenvolvido um método para detecção e remoção de linhas de referência (pré-impressas em papéis). Este método foi testado para identificação de escritor na língua Árabe por meio de experimentos usando SVM, e tendo como características para identificação de escrita um conjunto de primitivas que levam em consideração informações relativas ao contorno de segmentos adjacentes. Os resultados iniciais, com 60 diferentes escritores e taxas de acerto em torno de 54,9% (com a remoção das linhas de referência pré-impressas) mostraram que em situações realísticas, nas quais linhas de referência são esperadas, removê-las melhora significantemente as taxas de identificação.

No trabalho de Amaral et al. (2012), citado anteriormente, além das características em nível de documento extraídas, o posicionamento do texto em relação à margem direita de cada linha também foi extraído.

## 3.2. Abordagem Local

### *a) Características Extraídas das Palavras*

No trabalho de Luna et al. (2011), descrito na seção anterior, um conjunto de características locais também é extraído das palavras, são elas: proporção da zona média das palavras comparada com as zonas ascendentes e descendentes e inclinação das palavras.

Zois e Anastassopoulos (2000) apresentam uma abordagem para o processo de identificação de autoria na qual o vetor de características de um escritor é derivado utilizando operadores morfológicos para obter o perfil horizontal das palavras (funções de projeções). As projeções são derivadas e processadas em segmentos de forma a aumentar a eficiência discriminatória do vetor de características. Para validar o método proposto no trabalho foram realizados experimentos tanto com classificador bayesiano quanto com redes neurais. As taxas de acerto chegaram próximas a 95%, em experimentos com 50 escritores usando palavras escritas em Inglês e em Grego.

 ©2012 SBC — Soc. Bras. de Computação

Al-Ma'adeed et al. (2008) apresentaram um método de identificação de autoria dependente de texto para a língua Árabe usando apenas 16 palavras. Neste trabalho as características extraídas incluem primitivas dependentes da direção da borda das palavras tais como altura, área, comprimento e direções (três medidas) com diferentes tamanhos. O classificador utilizado neste trabalho foi WED (*Weigthed Euclidean Distance)*. Os dados de teste foram 32000 imagens de textos Árabes de 1000 pessoas. A etapa de treinamento utilizou 75% desta base enquanto para a etapa de validação foram utilizados os 25% restantes. Os resultados obtidos chegaram próximos a 90% (para Top10) quando três (3) palavras são usadas.

Jain e Doermann (2011) propuseram um método para identificação *offline* de escritores usando a característica K segmentos adjacentes (KAS - *K Adjacent Segments*). Esta abordagem apresentou taxas de acerto para Top1 de 93% na base de dados IAM (base de dados na língua Inglesa). Resultados obtidos de testes subseqüentes demonstraram que as taxas de identificação melhoram quando o número de amostras de treinamento aumenta, e adicionalmente que o uso das características utilizadas pode ser estendido para a língua Árabe. K segmentos adjacentes foi introduzido por Ferrari et al. (2008) como uma característica que representa o relacionamento entre conjuntos de bordas vizinhas em uma imagem por meio de detecção de objetos.

Além das características já destacas, no trabalho de Amaral et al. (2012) a primeira palavra de cada linha foi "destacada" e sua altura e proporção de pixels pretos foram calculados.

***b) Características Extraídas dos Caracteres***

Blankers et al. (2007) propõem um método para identificação de autoria por meio da extração e análise de características alográficas como loops e entrada (inicio) do traçado dos caracteres. Características sub-alográficas são computados a partir de partes das letras. Tais características são particularmente importantes para a identificação forense. Características relativas aos loops são encontradas em loops ascendentes que aparecem em letras como "l", "k" e "b", e descendentes que aparecem em letras como "g" e "j". Características de entrada do traçado representam a primeira parte do caractere e são encontradas em quase todas as letras do alfabeto, especialmente na escrita manual cursiva. Foi usado neste trabalho o classificador KNN e obtida taxa de acerto próxima a 98% em uma base com 41 escritores.

Pervouchine e Leedham (2007) apresentaram um estudo sobre o uso de características extraídas de três caracteres, são eles: "d", "y" e "f" e o grafema "th" como primitivas básicas no processo de identificação de autoria. De acordo os autores deste trabalho, a escolha por um conjunto previamente definido de caracteres e grafemas se justifica pelo fato de que é impossível considerar todos os caracteres e grafemas que ocorrem nos documentos manuscritos. Como classificador foram utilizadas redes neurais e um algoritmo genético foi utilizado para encontrar o conjunto de características ótimo (uma vez que um grande conjunto de primitivas foi extraído). Com base nos resultados obtidos neste trabalho é possível observar que características locais apresentam alto pode discriminatório, o que justifica seu uso na análise de documentos manuscritos. Também é relevante destacar que nas conclusões do trabalho os autores reforçam a importância dos grafemas no processo de identificação de autoria, o que reforça a ideia de que o formato de um caractere é afetado pelos caracteres adjacentes a

 ©2012 SBC — Soc. Bras. de Computação

ele. As melhores taxas de classificação atingiram 58% em uma amostra com 165 escritores.

Bui et al. (2011) propõem um framework para identificação de autoria baseada em duas abordagens locais: caracteres e grafemas; e apresentam sua eficiência para a melhoria de resultados em processos de reconhecimento de manuscritos. Experimentos foram conduzidos usando o algoritmo K-means e taxas de acerto próximas a 100% foram atingidas em uma base com 32 documentos escritos por 16 diferentes escritores.

## 4. Classificação das Abordagens para Identificação de Autoria

Com base no estudo apresentado na seção anterior, a Tabela 1 apresenta um breve resumo do estado da arte levando-se em consideração os seguintes critérios: categoria (nível de granularidade da característica), características extraídas, número de escritores participantes dos experimentos, algoritmo de classificação utilizado e melhores taxas de acerto.

**Tabela 1. Resumo estado da arte das abordagens de identificação de autoria *offline***

| Categoria | Trabalho | Característica Extraída | Tamanho da Amostra | Classificador | Taxa de Acerto |
|---|---|---|---|---|---|
| Documento | Said et al. (1998) | Textura - Filtros de Gabor e Matrizes de Co-ocorrência | 10 escritores | WED (Weigthed Euclidean Distance) | 96% |
| | Said et al. (2000) | Textura - Filtros de Gabor | 40 escritores | WED (Weigthed Euclidean Distance) | 96% |
| | Bulacu et al. (2007) | Textura | 250 escritores | KNN | 82% |
| | He et al. (2008) | Textura | 500 escritores | Árvores Escondidas de Markov (HMT) | 32% |
| | Helli e Morghaddam (2010) | Textura - Filtros de Gabor e XGabor | 100 escritores | Método Fuzzy | 100% |
| | Hanusiack et al. (2010) | Atributos genéticos e genéricos | 300 escritores | SVM | 95% |
| | Karunakara e Mallikarjunaswamy (2011) | Textura | 50 escritores | KNN | 94% |
| | Amaral et al. (2012) | Número de linhas do documento, posição das margens superior, inferior e esquerda. | 20 escritores | SVM | 80% |
| Parágrafo | Schomaker et al. (2007) | Geração de codebooks | 250 escritores | Redes de Kohonen | 97% |
| | Siddiqi e Vicent (2008) | Geração de codebooks | 650 escritores - IAM 375 escritores – RIMES | WED (*Weigthed Euclidean Distance*) | 84% - IAM 74% - RIMES |
| | Kumar et al. | Geração de codebooks | 732 imagens | SVM | 98% |

 ©2012 SBC — Soc. Bras. de Computação

| | | | | | |
|---|---|---|---|---|---|
| | (2011) | | (fase treinamento) 625 imagens (fase de teste). | | |
| Linha | Hertel e Bunke (2003) | Continuidade do traçado, regiões fechadas, contornos superiores e inferiores. | 50 escritores | KNN | 90% |
| | Schlapbach e Bunke (2004) | Inclinação, altura e obliqüidade das linhas de texto. | 50 escritores | HMM | 94,4% |
| | Bensefia et al. (2005) | Grafemas | 88 escritores - PSI 150 escritores - IAM | Vector Space Model (VSM) | 96% - PSI 86% - IAM |
| | Chen et al. (2010) | Informações relativas ao contorno de segmentos adjacentes e remoção de linhas de referência pré-impressas. | 60 escritores | SVM | 54,9% |
| | Luna et al. (2011) | Espaço percentual da margem esquerda e margem direita, separação entre linhas, direção geral da escrita, e espaço entre palavras. | 30 escritores | Algoritmo ALVOT | 88% |
| | Amaral et al. (2012) | Posição da margem direita. | 20 escritores | SVM | 80% |
| Palavras | Luna et al. (2011) | Proporção da zona média das palavras comparada com as zonas ascendentes e descendentes e inclinação das palavras. | 30 escritores | Algoritmo ALVOT | 88% |
| | Zois e Anastassopoulos (2000) | Uso de operadores morfológicos para obter o perfil horizontal das palavras. | 50 escritores | Bayesiano e Redes Neurais | 95% |
| | Al-Ma'adeed et al. (2008) | Primitivas dependentes da direção da borda das palavras tais como altura, área, comprimento e direções (três medidas) com diferentes tamanhos. | 32000 imagens de textos Árabes de 1000 pessoas | WED (*Weigthed Euclidean Distance*) | 90% (para Top-10) |
| | Jain e Doermann (2011) | K-segmentos adjacentes (KAS) | 650 escritores | KNN | 93% |
| | Amaral et al. (2012) | Altura e proporção de pixels pretas da primeira palavra. | 20 escritores | SVM | 80% |
| Caracteres | Blankers et al. (2007) | Características alográficas | 41 escritores | KNN | 98% |
| | Pervouchine e Leedham (2007) | Uso de características extraídas de três caracteres, são eles: "d", "y" e "f" e o grafema "th". | 165 escritores | Algoritmo DistAl – baseado no Perceptron | 58% |
| | Bui et al. (2011) | Protótipos de caracteres e grafemas. | 16 escritores | Kmeans | 100% |

 ©2012 SBC — Soc. Bras. de Computação

## 5. Considerações Finais

Este trabalho apresentou uma revisão sobre abordagens para identificação de autoria *offline*. Estas abordagens foram analisadas, levando-se em consideração os seguintes elementos: granularidade e características extraídas para o processo de identificação, base utilizada para validação dos experimentos de identificação, algoritmo de classificação utilizado e taxa de reconhecimento obtida.

Pode-se observar que muitos trabalhos que apresentam boas taxas de acerto concentram-se em características texturais do documento manuscrito, bem como na geração de *codebooks*. Embora muito relevantes, estes trabalhos não apresentam processos de extração de características alinhados com as abordagens adotadas pelos peritos forenses. Outra questão que deve ser destacada é que tais abordagens dependem de um alto poder computacional, uma vez que as mesmas envolvem a aplicações de filtros sobre as imagens dos manuscritos.

Trabalhos que realizam análises em nível de linha, palavras e caracteres, normalmente exigem uma manipulação direta, ou seja, pixel a pixel, da imagem do documento. Isto traz um alto nível de complexidade para o processo de extração, bem como distorções nos resultados e com isto as taxas de acerto podem ter seus valores reduzidos. Em contrapartida, o processo de extração destas características exige menos recursos computacionais.

## 6. Agradecimentos

Os autores agradecem a todos os alunos da PUCPR e do CESUMAR que contribuíram fornecendo exemplares de cartas forenses. E, ainda, agradecem à Fundação Araucária (PR) a qual vem apoiando o desenvolvimento deste projeto de pesquisa.

## 7. Referências

Al-ma´adeed, S. and Mohammed, E. and Alkassis, D. and Al-muslih, F. “Writer identification using edge-based directional probability distribution features for Arabic words”. In: IEEE/ACS International Conference on Computer Systems and Applications, p. 582-590, 2008.

Amaral, A. M. M. M and Freitas, C. O. A. and Bortolozzi, F. “The graphometry applied to writer identification”. To be publishes in: Proceedings of the 2012 International Conference on Image Processing, Computer Vision, and Pattern Recognition. july, 2012, Las Vegas, USA.

Baranoski, F. and Oliveira, L. S. and Justino, E. J. R. “Writer identification based on forensic science approach”. In: Conferencia Latinoamericana de Informática, v.1. p. 25-32, 2007.

Bensefia, A. and Paquet, T. and Heutte, L. “A writer identification and verification system”. Pattern Recognition Letters, v.26, n.13, p.2080-2092, 2005.

Blankers, V. and Niels, R. and Vuurpijl, L. “Writer identification by means of explainable features: shapes of loops and lead-in strokes”. In: Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence, p. 17-24, 2007.

 ©2012 SBC — Soc. Bras. de Computação

Bui, Q. A. and Visani, M. and Prum, S. and Ogier, J. M. "Writer identification using TF-IDF for cursive handwritten word recognition". In: Proceedings of International Conference on Document Analysis and Recognition, p. 844-848, 2011

Bulacu, M. and Schomaker, L. and Brink, A. "Text-independent writer identification and verification on offline Arabic handwriting". In: Proceedings of the 9th Conference on Document Analysis and Recognition (ICDAR), 2007.

Cavalcanti, A. and Lira, E. "Grafoscopia essencial". Porto Alegre: Sagra Luzzatto. 1996. 151p.

Chen, J. and Lopresti, D. and Kavallieratou, E. "The impact of ruling lines on writer identification". In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition, 2010.

Ferrari, V. and Fevrier, L. and Jurie, F. and Schmid, C. "Groups of adjacent contour segments for object detection". IEEE Trans. PAMI, n. 30, p.36 - 51, 2008.

Hanusiak, R. K. and Justino, E. J. R. and Oliveira, L. S. and Sabourin, R. "Identificação de autoria de manuscritos baseados em atributos genéticos e genéricos". RITA, v. 17, n.2, 2010.

Haralick, R. M. and Shanmugam, K. and Dinstein, I. "Textural features for image classification". IEEE Transactions on Systems, and Cybernetics, n.6, november, 1973.

He, Z. and You, X. and Tang, Y. "Writer identification of Chinese handrwriting documents using hidden markov tree model". Pattern Recognition, v.41, p.1295-1307, 2008.

Helli, B. and Moghaddam, E. "A text-independent Persian writer identification based on feature relation graph (FRG)". Pattern Recognition, v.43, p.2199-2209, 2010.

Hertel, C. and Bunke, H. "A set of novel features for writer identification". In: AVBPA'03 Proceedings of the 4th international conference on Audio- and video-based biometric person authentication, p. 679-687, 2003.

Jain, R. and Doermann, D. "Offline writer Identification using K-adjacent segments". In: Proceedings of the International Conference on Document Analysis and Recognition, p.769-773, 2011.

Karunakara, K. and Mallikarjunaswamy, B. P. "Writer Identification based on offline handwritten document images in Kannada language using empirical mode decomposition method". International Journal on Computer Applications, v.30, n.6, p.31-36, 2011.

Kumar, J. and Prasad, R. and Cao, H. and Almageed, W. A. and Doermann; Natarajan, D. "Shape codebook based handwritten and machine printed text zone extraction". In: Proceedings of the 18th Conference on Document Recognition and Retrieval, 2011.

Luna, E. C. H. and Riveron, E. M. F. and Calderon, S. G. "A supervisoned algorihm with a new differentiated-weighting scheme for identifying the author of a handwritten text". Pattern Recognition Letters, v.32, p. 1139-1144, 2011.

Mendes, L.B. "Documentoscopia". Campinas: Millennium. 2003. 344 p.

 ©2012 SBC — Soc. Bras. de Computação

Morris, R. N. "Forensic handwriting identification: fundamental concepts and principles". San Diego, California: Academic Press. 2000. 238p.

Pervouchine, V. and Leedham, G. "Extraction and analysis of forensic document examiner features used for writer identification". Pattern Recognition, v.40, p.1004-1013, 2007.

Said, H.E.S. and Peake, G.S and Tan, T and Baker, K. "Writer identification from non-uniformly skewed handwriting images". In: Proceeding of British Machine Vision Conference, p. 478-487, 1998.

Said, H. E. S. and Tan, T. and Baker, K. "Personal identification based on handwritings". Pattern Recognition, v.33, n.1, p.149-160, 2000.

Schlapbach, A. and Bunke, H. "Off-line handwriting identification using HMM based recognizers". In: Proceedings of the Pattern Recognition, 17th International Conference on ICPR'04 , v.2, 2004.

Schomaker, L. and Franke, K and Bulacu, M. "Using codebooks of fragmented connected-component contours in forensic and historic writer identification". Pattern Recognition Letters, v 28, p.719–727, 2007.

Sheikholeslami, G. and SriharI, S. N. and Govindaraju, V. "Computer aided graphology". In: Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition. Essex, England. p.457-460. September, 1996.

Siddiqi, I. and Vincent N. "Combining global and local features for writer identification". In: Proceedings of the Eleventh International Conference on Frontiers in Handwriting Recognition, p.48-53, 2008.

Sreejaj, M. and Idicula, S. M. "A survey on writer identification schemas". International Journal of Computer Applications, v. 26, n. 2, july, 2011. p.23-33.

Tan, T. "Rotation invariant texture features and their use in automatic script identification". IEEE Transaction on Pattern Analysis and Machine Intelligence, v.20, n.7, p. 751-756, 1998.

Zhu, Y. and Tan, T. and Wang, Y. "Font recognition based on global texture analysis". Transaction on Pattern Analysis and Machine Intelligence, v. 23, n.10, p.1192-1200, 2001.

Zois, E. and Anastassopoulos, V. "Morphological waveform coding for writer identification". Pattern Recognition, v. 33, n.3, p. 385-398, 2000.

 ©2012 SBC — Soc. Bras. de Computação

# Aplicação de Observação de Pessoas em Computação Forense

**William Robson Schwartz**[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627 – Prédio do ICEx – Pampulha
31270-010 – Belo Horizonte – MG – Brasil

***Abstract.*** *One of the main goals of automatic monitoring of open spaces is to extract information regarding activities been performed by humans such that suspicious activities can be identified. In order to be analyzed, a set of problems, such as detection and identification of humans in the scene, tracking people among multiple cameras, recognition of individual actions, need to be solved. Such problems are included in the application domain known as looking at people, which is responsible for the analysis of images and videos containing humans. This work describes the concepts related to the looking at people application domain, its challenges and attempts to connect problems in that domain to the area of computer forensics. Finally, scalable approaches to face recognition and person re-identification problems are presented and discussed.*

***Resumo.*** *Um dos principais objetivos do monitoramento automático de ambientes é a extração de informações a respeito de atividades desempenhadas pelos humanos de modo a detectar interações entre agentes e identificar padrões de comportamentos que sejam suspeitos. Para que as atividades sejam analisadas, um conjunto de problemas, tais como detecção e identificação dos agentes na cena, rastreamento ao longo do tempo, possivelmente entre câmeras distintas, reconhecimento de ações individuais, precisa ser resolvido. Tais problemas compõem o domínio de aplicações denominado observação de pessoas, responsável pela análise de imagens e vídeos contendo humanos. Este trabalho descreve os conceitos do domínio de observação de pessoas, discute seus desafios e visa estabelecer algumas relações entre problemas deste domínio com a área de computação forense. Finalmente, abordagens escaláveis para o problema de reconhecimento de faces e reidentificação de pessoas serão apresentadas.*

## 1. Introdução

A crescente disponibilidade de dados visuais adquiridos a partir de câmeras de vigilância espalhadas em diversos pontos de grandes cidades proporciona uma maior segurança e tranquilidade para as pessoas que circulam nesses ambientes monitorados. Ao longo dos últimos anos, autoridades públicas têm utilizado cada vez mais câmeras no monitoramento de regiões críticas em grandes cidades brasileiras. Por exemplo, a cidade de Belo Horizonte que contava, até 2011, com 235 câmeras de vigilância planeja a instalação de outras 232 câmeras [de Minas 2012], Florianópolis que possui 40 câmeras de vigilância receberá outras 150 câmeras [Manhã 2012] e a cidade de Curitiba, que possui 116 câmeras, planeja a instalação de 334 novas câmeras até 2014 [de Curitiba 2012]. Com essas configurações, o número de imagens adquiridas diariamente chega facilmente na ordem de $10^9$, o que impossibilita seu processamento manual.

 ©2012 SBC — Soc. Bras. de Computação

Se por um lado as câmeras de vigilância proporcionam uma grande quantidade de dados visuais, por outro, a análise desses dados torna-se um grande desafio quando feita manualmente. Desta maneira, o entendimento e a interpretação automática de atividades desempenhadas por humanos em vídeos apresenta grande interesse de modo a auxiliar a tarefa de agentes de segurança. O monitoramento automático de ambientes vigiados permitirá que novas tecnologias sejam desenvolvidas, como por exemplo, sistemas de prevenção de acidentes envolvendo idosos em ambientes movimentados e sistemas capazes de identificar atividades suspeitas sendo executadas por indivíduos com objetivo de prevenir crimes. Esse monitoramento proporcionará, portanto, benefícios e um maior bem estar para a sociedade.

Para que atividades desempenhadas por humanos sejam analisadas de forma automática, tarefas como detecção, reconhecimento, rastreamento e reidentificação de pessoas e o reconhecimento de ações individuais, devem ser tratados de forma acurada e eficiente. Compreendendo o subconjunto de problemas da área de visão computacional que tratam da análise de imagens e vídeos contendo humanos, tais tarefas compõem o domínio de aplicações denominado *Observação de Pessoas* [Gavrila 1999]. Devido ao fato de que a interação entre humanos provê informações relevantes para o entendimento de atividades executadas, a observação de pessoas desperta significativo interesse na comunidade.

A área de observação de pessoas se propõe a estudar uma série de problemas tendo como foco central a presença de humanos e a análise de suas interações. A Figura 1 apresenta um diagrama contendo problemas normalmente relacionados com a observação de pessoas, bem como suas relações de dependência. Neste diagrama, os módulos localizados mais à esquerda definem problemas de mais *baixo nível*, no sentido de que possuem pouca dependência da resolução de outros problemas, como remoção de fundo ou detecção de faces e pedestres. Módulos localizados mais à direita são definidos como problemas de mais *alto nível*, pois necessitam de soluções providas por diversos módulos anteriores como, por exemplo, a resolução do reconhecimento de ações.

Além dos módulos centrais, o diagrama apresenta três módulos externos: extração de características, descrição da cena e reconhecimento de atividades. O primeiro provê descritores de características responsáveis pela extração da informação visual contida na cena, sendo utilizados para a resolução de diversos problemas indicados na parte central do diagrama da Figura 1. O segundo armazena informações obtidas com a resolução dos problemas com intuito de criar uma representação da cena em questão, de modo que, atividades de interesse como aquelas envolvendo a interação entre agentes, análise de comportamento suspeito ou identificação de colaboração entre agentes, possam ser reconhecidas e analisadas por sistemas de inferência, isto efetuado pelo terceiro módulo, o reconhecimento de atividades.

Devido ao alto grau de interdependência dos problemas que compõem o domínio de observação de pessoas, estes estão suscetíveis à propagação e amplificação de erros (e.g., erros durante a detecção podem prejudicar o rastreamento de um indivíduo, o que pode tornar impraticável o reconhecimento de suas ações). Desta maneira, é imprescindível que esses problemas sejam resolvidos de modo acurado. Em geral, descritores de características são utilizados para extrair e representar a informação visual contida em imagens. Diversos métodos de extração de características têm sido propostos, cada um focando em aspectos distintos da imagem [Tuytelaars and Mikolajczyk 2008,

 ©2012 SBC — Soc. Bras. de Computação

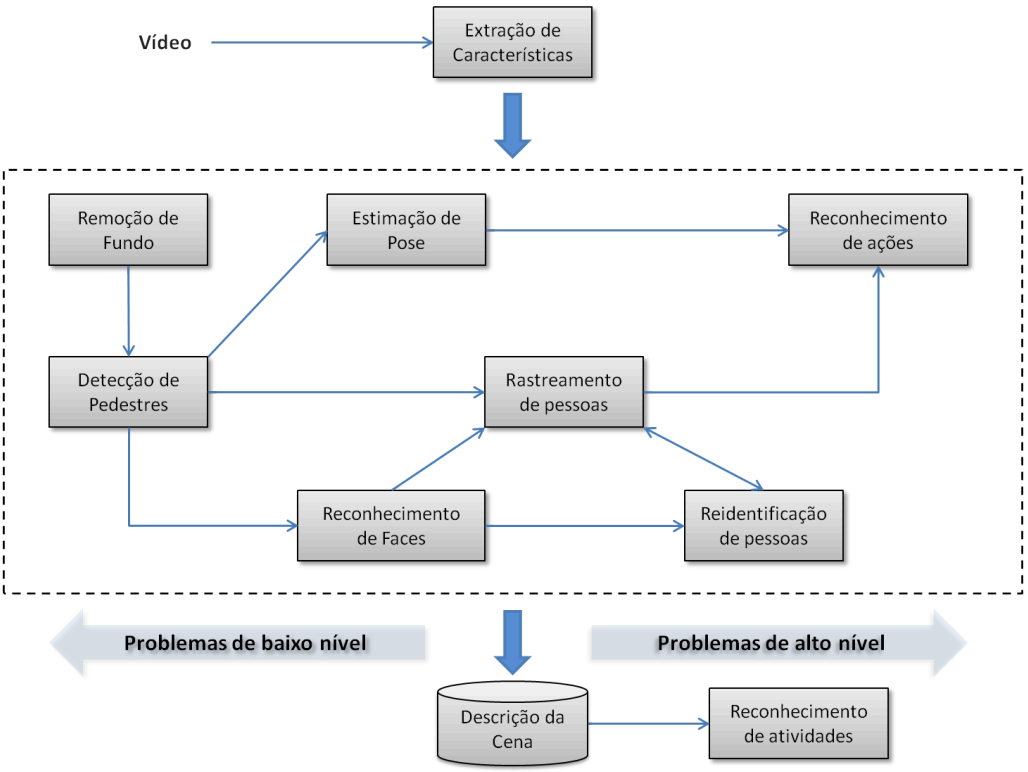



**Figura 1. Diagrama mostrando os problemas abordados no domínio de observação de pessoas. Os problemas dentro do retângulo tracejado obtêm a informação visual da cena por meio dos descritores de características, providos pela extração de características, e seus resultados são utilizados na criação de uma representação da cena, a qual será considerada, posteriormente, para efetuar o reconhecimento de atividades.**

Gauglitz et al. 2011]. Quando um único método é considerado, informações importantes podem estar sendo ignoradas. Desta maneira, a combinação de descritores é desejável para se obter uma representação mais robusta e com alto poder discriminativo.

Além da necessidade de soluções acuradas, outro fator que influencia a resolução de problemas que compõem o domínio de observação de pessoas é a quantidade de dados disponíveis. A utilização de uma baixa quantidade de dados (e.g., vídeo de baixa resolução adquirido a partir de uma única câmera), permite que métodos pouco eficientes para detecção e reconhecimento obtenham resultados em tempo real. No entanto, informações relevantes sobre a cena podem ser perdidas (e.g., o indivíduo suspeito passa estar ocluído por outras pessoas ou a câmera possui resolução insuficiente para efetuar o reconhecimento facial), o que prejudica diretamente o reconhecimento das atividades, objetivo final de aplicações de monitoramento e vigilância. Desta maneira, torna-se necessária a aquisição de uma grande quantidade de dados visuais (e.g., vídeos de alta resolução capturados a partir de múltiplas câmeras), o que requer a utilização de métodos eficientes para seu processamento.

Vigilância é uma das aplicações direta das tarefas dentro do domínio de observação de pessoas pois tem como objetivo criar uma representação da cena para que atividades suspeitas possam ser identificadas e alguma atitude por parte de agentes de segurança possa ser tomada, ou seja, a vigilância utiliza este ferramental com intuito de prevenir a

 ©2012 SBC — Soc. Bras. de Computação

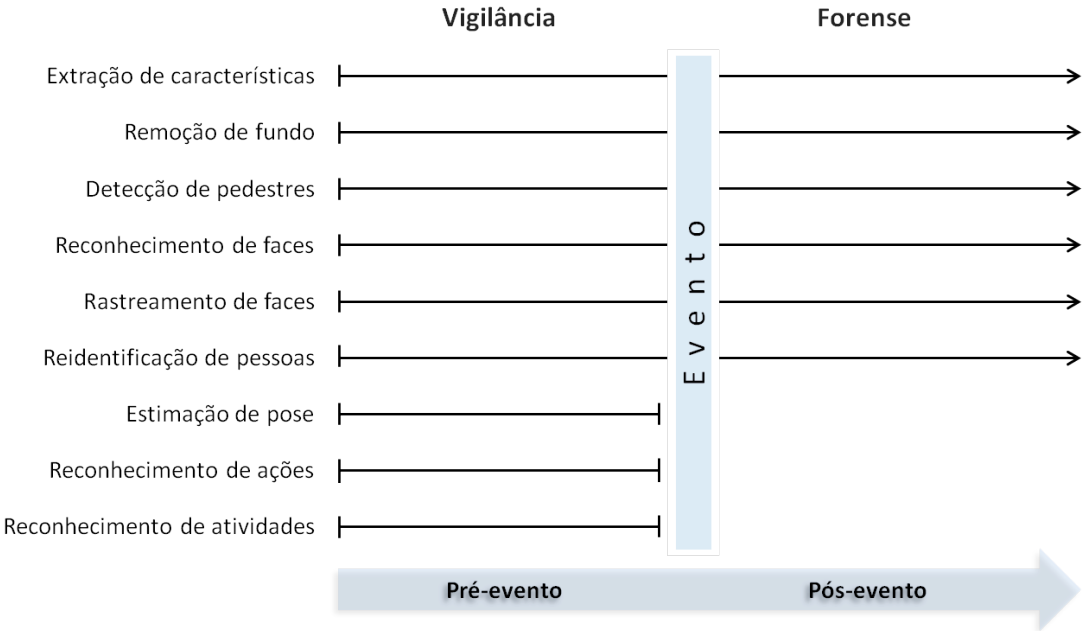



**Figura 2. Diagrama relacionando problemas do domínio de observação de pessoas considerados para vigilância que também podem ser aplicados na computação forense. A vigilância atua em um tempo anterior ao acontecimento de um determinado evento visando a prevenção de tal, enquanto que a computação forense trata de ações já ocorridas com objetivo de encontrar evidências que possam ser utilizadas para incriminar o agente envolvido no evento.**

execução de ações que podem levar a um crime. Embora seja focada no tempo após a execução de um evento ou ação, a área de computação forense também pode se beneficiar das ferramentas de observação de pessoas, principalmente para reduzir a busca manual em longas sequências de vídeos coletadas na cena de um crime, por exemplo.

O diagrama da Figura 2 apresenta as tarefas dentro do domínio de observação de pessoas utilizadas para vigilância e aquelas que podem ser utilizadas para computação forense com intuito de reduzir o espaço de busca e na identificação e localização de evidências da participação de suspeitos de um crime. Este diagrama é separado em dois períodos, antes do evento, onde a vigilância atua com papel de prevenção, e o tempo após a execução do evento, onde a computação forense atual com intuito de levantar evidências a partir dos dados visuais coletados ao longo do tempo.

Um exemplo do benefício da aplicação das ferramentas de observação de pessoas em computação forense seria a utilização de sequências de vídeo coletadas por uma rede de câmeras de vigilâncias localizadas em uma área próxima da cena de um crime. A enorme quantidade de dados, possivelmente coletados durante meses, torna impossível sua análise manual, no entanto, técnicas de reidentificação de pessoas e reconhecimento de faces podem ser utilizadas para buscar evidências contra um suspeito. Por exemplo, nos dias anteriores ao crime, o suspeito vigiou algum local específico com intuito de reconhecer a região ou analisou áreas de escape para serem utilizadas após sua execução.

O restante deste trabalho apresenta uma revisão bibliográfica dos problemas dentro do domínio de observação de pessoas (Seção 2) e descreve abordagens escaláveis para o reconhecimento de faces (Seção 3.1) e para o problema de reidentificação de pessoas

 ©2012 SBC — Soc. Bras. de Computação

(Seção 3.2), as quais podem ser aplicadas tanto em vigilância quanto em forense. Finalmente, resultados experimentais são apresentados na seção 4 e as conclusões obtidas com este trabalho são descritas na Seção 5.

## 2. Estado da Arte

Esta seção descreve os principais aspectos do problemas tratados no domínio de observação de pessoas, com a descrição de suas principais abordagens e dificuldades.

A *extração de características* [Gauglitz et al. 2011] apresenta-se como uma etapa fundamental na resolução dos problemas no domínio de observação de pessoas, devido, principalmente, à importância de se obter uma boa representação das informações visuais contidas na imagem ou vídeo para que não haja propagação de erros ao longo da cadeia de problemas a serem resolvidos. Permitindo, portanto, a análise acurada das atividades sendo executadas em uma cena.

Um dos poucos problemas que, em geral, não utiliza descritores de características como entrada, a *remoção de fundo* é responsável pela detecção de objetos em movimento em um vídeo, normalmente capturado a partir de câmeras estáticas. Tipicamente, este é o primeiro estágio de processamento em sistemas de vigilância [Moeslund et al. 2006]. Enquanto uma vantagem de sua aplicação é a obtenção de silhuetas bem definidas em torno dos objetos, a remoção de fundo apresenta o problema de não ser capaz de localizar objetos individualmente ou não detectar objetos com pouco movimento, além de não ser inerentemente robusto em relação à movimentação da câmera.

Robusto aos problemas enfrentados por abordagens de remoção de fundo, métodos de detecção de objetos têm sido propostos. Tais métodos dependem de um conjunto de treinamento (composto de exemplares dos objetos sendo considerados), a ser utilizado para estimar parâmetros de um classificador, o qual será aplicado para calcular a probabilidade de se encontrar objetos em regiões de uma imagem ou vídeo [Gavrila 2000]. Dentre as classes de objetos, a *detecção de pedestres* desperta amplo interesse [Dollar et al. 2012], principalmente na área de vigilância [Enzweiler and Gavrila 2009] e proteção de pedestres no trânsito [Gandhi and Trivedi 2007].

Uma vez detectados, os pedestres devem ser identificados unicamente de modo que o rastreamento desses indivíduos seja efetuado com sucesso ao longo do tempo e entre câmeras distintas. O *reconhecimento facial* é uma abordagem amplamente aplicada para identificação individual [Chellappa et al. 2010, Zhang and Gao 2009]. A representação de faces a partir de descritores de características tem se mostrado essencial. *Scale-invariant feature transform* (SIFT) [Lowe 2004] e *histogram of oriented gradients* (HOG) [Dalal and Triggs 2005a], que podem ser vistos como um código quantizado da informação de gradientes presentes na face, são efetivos quando aplicados ao reconhecimento de faces [Luo et al. 2007, Albiol et al. 2008, Schwartz et al. 2012]. Entretanto, ainda existe a necessidade de melhorias, principalmente quando o reconhecimento facial precisa ser efetuado em grandes conjuntos de dados.

Há alguns desafios importantes a serem superados para que o reconhecimento de faces atinja melhores resultados: (1) apresentar robustez quando as condições de aquisição dos dados não são devidamente controladas; (2) muitos algoritmos de re-

 ©2012 SBC — Soc. Bras. de Computação

conhecimento de faces necessitam de múltiplas amostras para estimar as características de cada indivíduo considerado; (3) a escalabilidade dos métodos deve ser mantida mesmo para galerias faciais contendo muitos indivíduos. Alguns trabalhos têm proposto abordagens para execução de reconhecimento em condições razoavelmente não controladas [Guillaumin et al. 2009, Wright et al. 2009], considerando a utilização de poucas amostras [Liu et al. 2007, Cao et al. 2010] e alguns aspectos relativos à escalabilidade [Zeng et al. 2009].

Além da identificação única de cada pedestre, é necessário que este seja rastreado ao longo do tempo. O rastreamento de pessoas apresenta grande importância principalmente quando múltiplos agentes (pedestres) são encontrados na cena pois permite a aplicação de algoritmos para analisar a interação entre agentes, a redução do espaço de busca para métodos de detecção baseados em janela deslizante e o reconhecimento de ações individuais [Yilmaz et al. 2006], dentre outros. O problema de rastreamento tem sido o foco de diversas pesquisas nos últimos anos. A melhoria em desempenho e a redução do preço de câmeras, assim como o aumento de recursos computacionais, têm permitido o desenvolvimento de algoritmos de rastreamento mais sofisticados. Especificamente, o advento de sistemas inteligentes de vigilância tem focado no rastreamento simultâneo de múltiplos agentes [Ko 2008].

O rastreamento de pedestres é aplicado quando os pedestres estão sendo capturados por uma câmera ou câmeras que apresentem intersecção do campo de visão. No entanto, em cenários onde um ambiente é observado por múltiplas câmeras, não necessariamente com intersecção do campo de visão, a *reidentificação de pessoas* visa à identificação de indivíduos que foram previamente identificados. Reidentificação de pessoas tem sido considerada para diversos domínios de aplicação, tais como vigilância e monitoramento [Tu et al. 2007] e eventos esportivos [Shitrit et al. 2011].

Uma vez que a identificação e o rastreamento individual tenham sido aplicados, as ações sendo efetuadas pelo pedestre podem sem reconhecidas. O reconhecimento de ações tornou-se uma área de pesquisa bem ativa nos últimos anos [Poppe 2010]. Abordagens usando *bag-of-words* baseadas no método clássico *k-means* têm sido extensivamente utilizadas para reconhecimento de ações. Métodos supervisionados que proporcionam uma maior discriminação também têm sido explorados, tais como *codebooks* [Yang et al. 2008], florestas aleatórias [Moosmann et al. 2006] e o aprendizado de matrizes de distância [Bilenko et al. 2004]. Apesar dos esforços, métodos baseados em *bag-of-words* geralmente ignoram a distribuição espaço-temporal dos descritores, o que tende a reduzir significativamente a discriminação obtida pelos descritores utilizados para representar ações distintas.

Uma etapa que também pode ser utilizada para se obter informações a serem utilizadas para o reconhecimento de ações é a *estimação de pose*, a qual é responsável por estimar e rastrear a pose do corpo humano em um espaço tridimensional utilizando informações normalmente adquiridas a partir de uma única câmera. Desta maneira, ações que envolvem, por exemplo, sinalização de um indivíduo para outro podem ser reconhecidas com o auxílio de informações a respeito da pose [Poppe 2007].

Finalmente, uma vez que informações sobre a localização, identificação, trajetórias e as ações sendo executadas pelos agentes tenham sido obtidas, uma repre-

 ©2012 SBC — Soc. Bras. de Computação

sentação da cena pode ser criada e utilizada para reconhecer atividades sendo executadas [Aggarwal and Ryoo 2011] (e.g., agente seguindo uma segunda pessoa, agente abrindo ou fechando o porta-malas de um veículo, agente interagindo com um grupo de pessoas). Desta maneira, este reconhecimento depende da correta solução para os demais problemas descritos anteriormente e ilustrados no diagrama da Figura 1. Além disso, o reconhecimento de atividades é um problema difícil por si só, pois depende de um sistema de inferências para detectar quais atividades estão sendo executadas em um dado momento. Em geral, tais sistemas (e.g., cadeias escondidas de Markov, máquinas de estados finitos, redes neurais e regras lógicas), necessitam de uma grande quantidade de dados para estimação de parâmetros [Ko 2008].

## 3. Abordagens Escaláveis para o Reconhecimento de Faces e Reidentificação de Pessoas

Esta seção descreve duas abordagens escaláveis que visam à redução do custo computacional na resolução da identificação de faces e reidentificação de pessoas. Tais problemas são encontrados tanto na área de vigilância como em computação forense. No primeiro caso, o objetivo é reconhecer indivíduos presentes em sequências de vídeos, gerando informações para um módulo de inteligência responsável por analisar comportamentos individuais ou coletivos de modo a reconhecer padrões de comportamento suspeito. Por outro lado, na área de computação forense o reconhecimento de faces e a reidentificação de pessoas podem auxiliar na localização e identificação de indivíduos específicos em longas sequências de vídeos, para subsequente recuperação de evidências.

A primeira abordagem, mostrada na Seção 3.1, descreve um método escalável baseado em uma árvore de busca para testar amostras apresentadas a um sistema de reconhecimento de faces [Schwartz et al. 2010, Schwartz et al. 2012]. A segunda abordagem, descrita na Seção 3.2, apresenta uma solução baseada no esquema de classificação um-contra-alguns [Carlos et al. 2012, Schwartz 2012] para adição de novos indivíduos na galeria quando o problema de reidentificação de pessoas é considerado.

### 3.1. Reconhecimento de Faces

Reconhecimento de faces tem se tornado uma área de pesquisa bastante ativa nos últimos anos, devido à vasta gama de aplicações relacionadas à vigilância, interação humano-computador e biometria. As duas tarefas principais relacionadas ao reconhecimento de faces são *verificação* e *identificação*. Enquanto na primeira o objetivo está em aceitar ou rejeitar um par de amostras como pertencentes a uma mesma pessoa, a tarefa de identificação objetiva o casamento de uma face desconhecida a uma galeria contendo pessoas previamente identificadas.

Neste trabalho, trata-se do problema de identificação de faces com o foco na redução do custo computacional para o casamento de amostras de teste, conforme descrito originalmente no trabalho proposto por Schwartz et al. [Schwartz et al. 2012], que é composto por três etapas principais: extração de características, modelagem dos indivíduos contidos na galeria e execução de regressões para efetuar o casamento de amostras de teste desconhecidas com indivíduos presentes na galeria.

Na extração de características, após recortada a região da imagem contendo apenas a face e reescalada para um dado tamanho, cada amostra é decomposta em um

 ©2012 SBC — Soc. Bras. de Computação

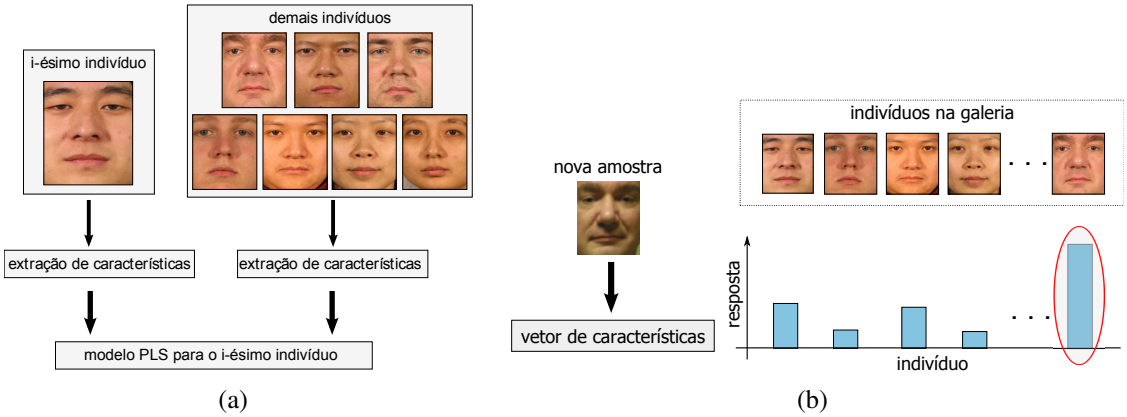



**Figura 3. Abordagem um-contra-todos aplicada à identificação de faces. (a) construção dos modelos PLS para o $i$-ésimo indivíduo presente na galeria; (b) casamento de uma amostra de face com os indivíduos contidos na galeria (figura adaptada de [Schwartz et al. 2012]).**

conjunto de blocos e um conjunto de descritores é extraído para cada bloco. Características exploradas incluem informação relacionada com forma (extraída com uso do descritor HOG [Dalal and Triggs 2005b]), textura (capturada utilizando-se o descritor LBP [Ahonen et al. 2004]) e cor (extraída a partir da média das intensidades dos pixels para cada banda de cor). Finalmente, um vetor de características $\boldsymbol{v}$ é composto pela concatenação dos descritores extraídos a partir de cada bloco. Este vetor é utilizado para descrever a face durante o reconhecimento.

O procedimento para estimar modelos para cada um dos $N$ indivíduos contidos na galeria $g = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N\}$, em que $\boldsymbol{s}_i$ representa exemplares do $i$-ésimo indivíduo, codificados por meio dos vetores obtidos durante a extração de características, dá-se da seguinte maneira. Como a identificação de faces é um problema contendo múltiplas classes, um esquema de classificação um-contra-todos é aplicado para estimar modelos de regressão utilizando o método estatístico para modelar relações entre variáveis denominado *Partial Least Squares* (PLS) [Wold 1985]. Dessa maneira, quando o $i$-ésimo indivíduo é considerado, as demais amostras, $g \setminus \boldsymbol{s}_i$, são utilizadas como contra-exemplos, conforme ilustrado na Figura 3(a). PLS estima a capacidade de discriminação para cada descritor e retorna os coeficientes de regressão $\boldsymbol{\beta}_i$. Este processo é executado para todos os $N$ indivíduos contidos na galeria.

Após a estimação de todos os modelos PLS e seus respectivos coeficientes de regressão, esses coeficientes são utilizados para efetuar o casamento de amostras desconhecidas com indivíduos contidos na galeria. Quando uma nova amostra é apresentada, seu vetor de características é projetado em cada um dos $N$ modelos. O melhor casamento da amostra ocorre com o modelo que apresentar a maior resposta, conforme ilustrado na Figura 3(b).

A abordagem um-contra-todos apresenta alto custo computacional para se encontrar o melhor casamento dada uma amostra de teste, pois o vetor de características desta amostra precisa ser projetado em todos os modelos PLS para que a maior resposta seja obtida. Essa característica deixa este método de identificação de faces pouco escalável para galerias contendo muitos indivíduos distintos.

 ©2012 SBC — Soc. Bras. de Computação

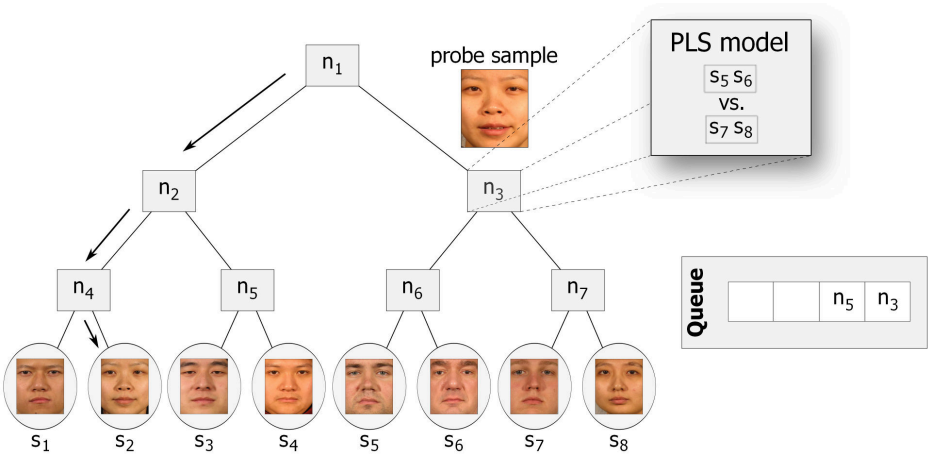



**Figura 4. Abordagem baseada em uma estrutura de árvore. Cada nó interno contém um modelo PLS para guiar a busca, como mostrado em detalhes para o nó $n_3$. Tal modelo é estimado de modo que sua resposta direcione a busca ou para o nó $n_6$ ou $n_7$. No exemplo, o primeiro caminho a ser buscado na árvore é indicado por flechas. Caminhos alternativos são obtidos com a adição de nós ainda não visitados em uma fila de prioridades (no exemplo os nós $n_3$ e $n_5$ foram adicionados). Após percorrer alguns caminhos, o melhor candidato da galeria é aquele que apresenta a maior resposta da classificação (figura adaptada de [Schwartz et al. 2012]).**

Com o objetivo de reduzir o número de projeções necessárias, utiliza-se uma árvore binária, na qual cada nó $n_j$ contém um subconjunto de indivíduos da galeria $t_j \subset g$. Um procedimento de divisão aleatório é utilizado para decidir quais indivíduos pertencem às subárvores esquerda e direita do nó $n_j$, de modo a deixar a árvore final balanceada. Cada nó interno é associado a um modelo PLS a ser utilizado posteriormente para guiar a busca para uma amostra de teste. A criação dos modelos PLS para cada nó $n_j$ dá-se de modo que os indivíduos pertencentes ao filho esquerdo de $n_j$ são considerados como uma classe e os indivíduos pertencentes ao filho direito de $n_j$ são considerados como uma segunda classe. O processo de divisão da árvore é aplicado recursivamente até que o nó contenha apenas um indivíduo (nó folha).

Quando um vetor de características descrevendo uma amostra é analisado para encontrar o indivíduo da galeria a qual ela pertence, uma busca iniciando na raiz da árvore é efetuada. A cada nó interno, o vetor de características é projetado no modelo PLS e, conforme a resposta, a busca continua para o filho esquerdo ou direito, até encontrar uma folha. Esse procedimento de busca é efetuado um dado número de vezes com o auxílio de uma fila de prioridades que mantém a localização do início do próximo caminho na árvore, de acordo com as respostas obtidas em cada nó, conforme ilustrado na Figura 4.

### 3.2. Reidentificação de Pessoas

O método descrito em [Schwartz and Davis 2009] propôs o uso do esquema de classificação um-contra-todos para o problema de reidentificação de pessoas. No entanto, quando um novo indivíduo é adicionado à galeria[1], todos os modelos precisam ser reconstruídos devido à estimativa da capacidade de discriminação de cada descritor. Portanto, se por um

[1]No problema de reidentificação, galeria se refere ao conjunto de aparências dos indivíduos considerados.

 ©2012 SBC — Soc. Bras. de Computação

lado a estimação de discriminabilidade proporciona melhorias consideráveis nos resultados, conforme mostrado em [Schwartz and Davis 2009], por outro, o custo computacional requerido para inserção de novos indivíduos na galeria é proibitivo.

Para eliminar a necessidade de se reconstruir todos os modelos, o trabalho descrito em [Schwartz 2012] propôs a utilização do esquema de classificação denominado um-contra-alguns. Neste esquema, modelos PLS são criados considerando-se apenas um subconjunto de indivíduos. Este esquema segue as mesmas etapas de [Schwartz and Davis 2009], exceto que enquanto na abordagem um-contra-alguns considera-se somente um subconjunto dos indivíduos como classe negativa, na abordagem um-contra-todos utiliza-se todos os indivíduos restantes.

Como a abordagem um-contra-alguns considera um subconjunto dos indivíduos como classe negativa ao invés de todos os indivíduos restantes, não é necessário conhecer todos os indivíduos no momento da adição de um novo indivíduo. Entretanto, por outro lado, mais que um modelo é criado, considerando-se um dado indivíduo como amostra positiva. Assim, haverá mais de uma resposta da regressão para cada indivíduo e, como na técnica um-contra-todos, o indivíduo associado com a maior resposta é considerado como a melhor escolha para a amostra testada.

Assim sendo, para a criação da galeria inicial (contendo todos os $N$ indivíduos conhecidos até o momento), calcula-se $N \times k$ modelos PLS (para $k \ll N$), em que $k - 1$ é o número de amostras negativas em cada modelo. Escolhe-se aleatoriamente quais serão os $k - 1$ indivíduos utilizados em cada modelo, procurando-se manter os indivíduos uniformemente distribuídos. Quando um novo indivíduo é adicionado à galeria, $k$ novos modelos PLS são computados, contendo amostras deste indivíduo como classe positiva e um subconjunto aleatoriamente escolhido de $k$ indivíduos na classe negativa. Devido à característica de apenas um subconjunto de indivíduos ser considerado, não há a necessidade de se reconstruir os modelos como no esquema de classificação um-contra-todos.

## 4. Resultados Experimentais

Esta seção mostra os resultados experimentais obtidos com os métodos de identificação de faces e reidentificação de pessoas descritos na seção anterior.

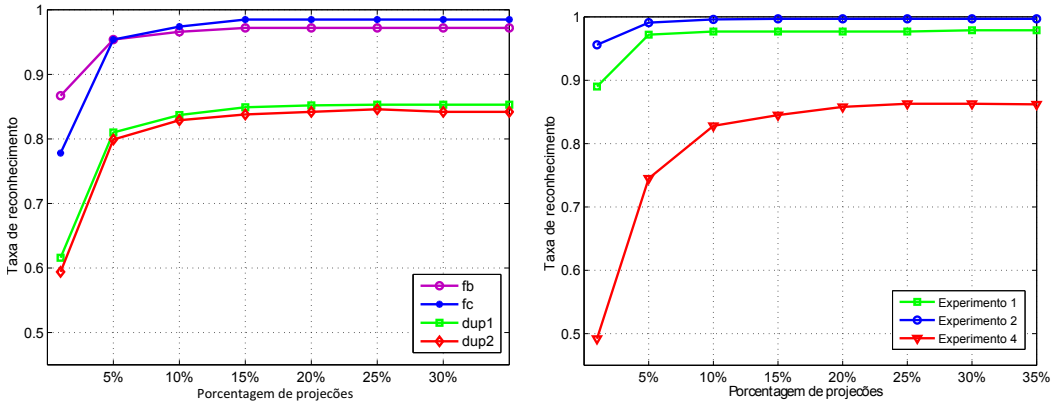
### 4.1. Reconhecimento de Faces

Conforme descrito em [Schwartz et al. 2012], as Figuras 5(a) e 5(b) mostram os resultados obtidos com a utilização da abordagem baseada em árvores para os conjuntos de dados FERET [Phillips et al. 2000] (cujo protocolo de avaliação é composto pelos experimentos fb, fc, dup1 e dup2) e FRGC [Phillips et al. 2005] (cujo protocolo de avaliação é composto pelos experimentos 1, 2 e 4), respectivamente. Percebe-se que quando o número de projeções requiridas pelo método um-contra-todos é reduzido para $20\%$ ou $25\%$, há uma redução muito baixa nas taxas de reconhecimento se comparado as taxas obtidas pelo método um-contra-todos original (Tabelas 1 e 2). De acordo com os gráficos, há uma redução de $6.66$ vezes no número de projeções para o conjunto de dados FERET e até $10$ vezes para o conjunto FRGC, dependendo do experimento considerado.

### 4.2. Reidentificação de Pessoas

Para avaliar a utilização da abordagem um-contra-alguns no problema de reidentificação, foi escolhido o conjunto de dados para modelagem de aparências

 ©2012 SBC — Soc. Bras. de Computação




(a) FERET (b) FRGC

**Figura 5. Taxas de reconhecimento como função da percentagem do número de projeções executadas pelo método baseado em árvore quando comparado com o método um-contra-todos original.**

**Tabela 1. Taxas de reconhecimento obtidas para os conjuntos de dados FERET utilizando a abordagem um-contra-todos sem a utilização da estrutura de árvore para otimização da busca.**

| Experimento | rank-1 (%) |
|---|---|
| fb | 97.20 |
| fc | 98.50 |
| dup I | 85.30 |
| dup II | 85.50 |

**Tabela 2. Taxas de reconhecimento obtidas para os conjuntos de dados FRGC utilizando a abordagem um-contra-todos sem a utilização da estrutura de árvore para otimização da busca.**

| Experimento | rank-1 (%) |
|---|---|
| Exp. 1 | 97.90 |
| Exp. 2 | 99.80 |
| Exp. 4 | 86.20 |

ETHZ [Schwartz and Davis 2009, Ess et al. 2008], composto de três sequências de vídeo.

O experimento aqui descrito tem como objetivo comparar a abordagem um-contra-todos com a abordagem um-contra-alguns em termos de acurácia. Considerando $k = 7$, a Tabela 3 mostra as taxas de reconhecimento para cada sequência do conjunto de dados ETHZ. Percebe-se que os resultados obtidos com ambas abordagens são muito similares. Embora não apresentados aqui, resultados experimentais mostram que a adição de novos indivíduos considerando-se a abordagem um-contra-alguns apresenta custo computacional linear enquanto a adição de indivíduos com o esquema um-contra-todos apresenta custo quadrático.

**Tabela 3. Taxas de reconhecimento obtidas para o conjunto de dados ETHZ.**

| Sequência | um-contra-todos (%) | um-contra-alguns (%) |
|---|---|---|
| #1 | 71.90 | 70.74 |
| #2 | 73.50 | 72.85 |
| #3 | 90.50 | 89.47 |

 ©2012 SBC — Soc. Bras. de Computação

## 5. Conclusões

Este trabalho descreveu os principais conceitos relacionados ao domínio de aplicação denominado observação de pessoas e mostrou sua aplicação em vigilância e computação forense. Adicionalmente, duas abordagens visando tornar possível o processamento de grandes quantidades de dados para o reconhecimento de faces e reidentificação de pessoas foram descritas. Embora apresentem melhorias em termos de redução do custo computacional sem decaimento da acurácia, ainda há um longo caminho para que se possa processar enormes quantidades de dados visuais capturados a partir de redes de câmeras.

## Referências

Aggarwal, J. and Ryoo, M. (2011). Human Activity Analysis: A Review. *ACM Comput. Surv.*, 43(3):1–43.

Ahonen, T., Hadid, A., and Pietikainen, M. (2004). Face Recognition with Local Binary Patterns. In *European Conference on Computer Vision*, 35, pages 469–481.

Albiol, A., Monzo, D., Martin, A., Sastre, J., and Albiol, A. (2008). Face Recognition Using HOG-EBGM. *Pattern Recognition Letters*, 29:1537–1543.

Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *International Conference on Machine Learning*.

Cao, Z., Yin, Q., Tang, X., and Sun, J. (2010). Face Recognition with Learning-Based Descriptor. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 2707 –2714.

Carlos, G. P., Pedrini, H., and Schwartz, W. R. (2012). Uma Abordagem Escalável para Manutenção de Galeria de Faces. In A. Paiva, J. P. P., editor, *Workshop of Undergraduate Works (WUW) in SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, pages 137–142.

Chellappa, R., Sinha, P., and Phillips, P. (2010). Face Recognition by Computers and Humans. *Computer*, 43(2):46–55.

Dalal, N. and Triggs, B. (2005a). Histograms of Oriented Gradients for Human Detection. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 886–893.

Dalal, N. and Triggs, B. (2005b). Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893.

de Curitiba, P. M. (2012 (acesso 25/04/2012)). *Curitiba*. http://www.curitiba.pr.gov.br/noticias/cameras-de-seguranca-reduzem-ocorrencias-na-agua-verde/25882.

de Minas, E. (2012 (acesso 25/04/2012)). *Belo Horizonte*. http://www.em.com.br/app/noticia/gerais/2011/04/09/interna_gerais,220669/belo-horizonte-vai-ter-quase-500-cameras-de-monitoramento-para-a-copa.shtml.

Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):743–761.

 ©2012 SBC — Soc. Bras. de Computação

Enzweiler, M. and Gavrila, D. M. (2009). Monocular Pedestrian Detection: Survey and Experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195.

Ess, A., Leibe, B., Schindler, K., , and van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*.

Gandhi, T. and Trivedi, M. M. (2007). Pedestrian Protection Systems: Issues, Survey, and Challenges. *Trans. Intelligent Transportations Systems*, 8(3):413–430.

Gauglitz, S., Höllerer, T., and Turk, M. (2011). Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. *International Journal of Computer Vision*, 94(3):335–360.

Gavrila, D. (2000). Pedestrian Detection from a Moving Vehicle. In *European Conference on Computer Vision*, pages 37–49, London, UK, UK. Springer-Verlag.

Gavrila, D. M. (1999). The Visual Analysis of Human Movement: A Survey. *Elsevier Computer Vision and Image Understanding*, 73(1):82–98.

Guillaumin, M., Verbeek, J., and Schmid, C. (2009). Is that you? Metric Learning Approaches for Face Identification. In *IEEE Intl. Conference on Computer Vision*, pages 498–505.

Ko, T. (2008). A Survey on Behavior Analysis in Video Surveillance for Homeland Security Applications. In *IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–8.

Liu, J., Chen, S., Zhou, Z., and Tan, X. (2007). Single Image Subspace for Face Recognition. In *International Conference on Analysis and Modeling of Faces and Gestures*, pages 205–219.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

Luo, J., Ma, Y., Takikawa, E., Lao, S., Kawade, M., and Lu, B.-L. (2007). Person-Specific SIFT Features for Face Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Manhã, F. (2012 (acesso 25/04/2012)). *Florianopolis*. http://floripamanha.org/2012/01/capital-vai-receber-mais-150-cameras-de-vigilancia/.

Moeslund, T. B., Hilton, A., and Krüger, V. (2006). A Survey of Advances in Vision-Based Human Motion Capture and Analysis. *Elsevier Computer Vision and Image Understanding*, 104(2):90–126.

Moosmann, F., Triggs, B., and Jurie, F. (2006). Fast discriminative visual codebooks using randomized clustering forests. In *Neural Information Processing Systems*.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., C., J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). Overview of the Face Recognition Grand Challenge. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 947–954.

 ©2012 SBC — Soc. Bras. de Computação

Phillips, P. J., Moon, H., Rizvi, S. A., and Rauss, P. J. (2000). The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:1090–1104.

Poppe, R. (2007). Vision-Based Human Motion Analysis: An Overview. *Elsevier Computer Vision and Image Understanding*, 108:4–18.

Poppe, R. (2010). A survey on vision-based human action recognition. *Elsevier Image and Vision Computing*, 28(6):976–990.

Schwartz, W. R. (2012). Scalable People Re-Identification Based on a One-Against-Some Classification Scheme. In *IEEE International Conference on Image Processing*.

Schwartz, W. R. and Davis, L. S. (2009). Learning Discriminative Appearance-Based Models Using Partial Least Squares. In *Brazilian Symposium on Computer Graphics and Image Processing*.

Schwartz, W. R., Guo, H., Choi, J., and Davis, L. S. (2012). Face Identification Using Large Feature Sets. *IEEE Transactions on Image Processing*, 21(4):2245–2255.

Schwartz, W. R., Guo, H., and Davis, L. S. (2010). A Robust and Scalable Approach to Face Identification. In *European Conference on Computer Vision*, volume 6316 of *Lecture Notes in Computer Science*, pages 476–489.

Shitrit, H. B., Berclaz, J., Fleuret, F., and Fua, P. (2011). Tracking Multiple People Under Global Appearance Constraints. In *IEEE Intl. Conference on Computer Vision*.

Tu, P. H., Doretto, G., Krahnstoever, N. O., Perera, a. A., Wheeler, F. W., Liu, X., Rittscher, J., Sebastian, T. B., Yu, T., and Harding, K. G. (2007). An Intelligent Video Framework for Homeland Protection. In *Defence and Security Symposium*.

Tuytelaars, T. and Mikolajczyk, K. (2008). Local Invariant Feature Detectors: A Survey. In *Foundation and Trends in Computer Graphics and Vision*, pages 177–280.

Wold, H. (1985). Partial Least Squares. In *Encyclopedia of Statistical Sciences*, volume 6, pages 581–591. Wiley, New York, NY, USA.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust Face Recognition via Sparse Representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 210–227.

Yang, L., Jin, R., Sukthankar, R., and Jurie, F. (2008). Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object Tracking: A Survey. *ACM Computing Surveys*, 38(4).

Zeng, Z., Fang, T., Shah, S., and Kakadiaris, I. (2009). Local Feature Hashing for Face Recognition. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*.

Zhang, X. and Gao, Y. (2009). Face Recognition Across Pose: A Review. *Pattern Recognition*, 42(11):2876–2896.

 ©2012 SBC — Soc. Bras. de Computação

# Content-Based Filtering for Video Sharing Social Networks

**Eduardo Valle[1], Sandra Avila[2], Fillipe de Souza[2], Marcelo Coelho[2,3], Arnaldo de A. Araújo[2]**

[1] RECOD Lab — DCA / FEEC / UNICAMP, Campinas, SP, Brazil

[2] NPDI Lab — DCC / UFMG, Belo Horizonte, MG, Brazil

[3] Preparatory School of Air Cadets — EPCAR, Barbacena, MG, Brazil

dovalle@dca.fee.unicamp.br,
{sandra, fdms, mcoelho, arnaldo}@dcc.ufmg.br

***Abstract.*** *In this paper we compare the use of several features in the task of content filtering for video social networks, a very challenging task, not only because the unwanted content is related to very high-level semantic concepts (e.g., pornography, violence, etc.) but also because videos from social networks are extremely assorted, limiting the use of* a priori *information. We propose a simple method, able to combine diverse evidence, coming from different features and various video elements (entire video, shots, frames, keyframes, etc.). We evaluate our method in two social network applications, related to the detection of unwanted content — pornographic videos and violent videos. Using challenging test databases, we show that this simple scheme is able to obtain good results, provided that adequate features are chosen. Moreover, we establish the use of spatiotemporal local descriptors as critical to the success of the method in both applications.*

***Resumo.*** *Neste trabalho, comparamos o uso de diferentes características na tarefa de filtragem de conteúdo para redes sociais de vídeo, uma tarefa muito desafiadora, não só porque o conteúdo indesejado está relacionado a conceitos semânticos de muito alto-nível (por exemplo, pornografia, violência, etc ), mas também porque os vídeos das redes sociais são extremamente variados, impedindo o uso de informação a priori. Propomos um método simples, capaz de combinar evidências diversa, provenientes de diferentes características e elementos de vídeo (vídeo inteiro, tomadas, quadros, quadros-chave, etc.) Avaliamos o nosso método em duas aplicações para redes sociais, relacionadas à detecção de conteúdos não desejados — vídeos pornográficos e vídeos violentos. Usando bases de dados de teste desafiadoras, mostramos que este esquema simples é capaz de obter bons resultados, desde que características adequadas sejam escolhidas. Além disso, mostramos que o uso descritores espaço-temporais locais é crítico para o sucesso do nas duas aplicações.*

## 1. Introduction

Content-based classification and retrieval of visual documents by high-level semantic concepts has been an elusive goal pursued by the scientific community for the last 20 years. The persistent absence of a general solution attests the task difficulty, which is in great part brought by the much discussed "semantic gap" between the low-level representation of the data (pixels, frames, etc.) and the high-level concepts one wants to take into account.

We have, however, witnessed many important breakthroughs. In what concerns the description of visual documents, we have watched not only the inception and evolution of local features [22], but mainly the introduction of representations based on codebooks [5], which have allowed conciliating the discriminative power of the former with the generalization abilities required by high-level semantic tasks. Specifically for video, the introduction of "motion-aware" local features, which take into account the

 ©2012 SBC — Soc. Bras. de Computação

dual nature of that media, at the same time spatial *and* temporal, has been an important achievement [6][13][14][15][16]. Meanwhile, the development of machine learning algorithms, like SVM [21], has created an effective framework for complex classification tasks.

In this paper, we are concerned with the detection of unwanted content on video sharing social networks — online communities built upon the production, sharing and watching of short video clips, which have been nourished by the popularization of broadband web access and the availability of cheap video acquisition devices. The crowds of users who employ the services of websites like *Dailymotion*, *MetaCafe* and *YouTube*, not only post and watch videos, but also share ratings, comments, "favorite" lists and other personal appreciation data.

The emergence of those networks has created a demand for specialized tools, including mechanisms to control abuses and terms-of-use violations. Indeed, the success of social networks has been inevitably accompanied by the emergence of users with non-collaborative behavior, which prevents them from operating evenly. Those behaviors include instigating the anger of other users (*trolling*, in the web jargon), diffusing materials of genre inappropriate for the target community (e.g., diffusing advertisement or pornography in inadequate channels), or manipulating illegitimately popularity ratings.

Non-collaborative behavior pollutes the communication channels with unrelated information, and prevents the virtual communities from reaching their original goals of discussion, learning and entertainment. It alienates legitimate users and depreciates the social network value as a whole [1].

In addition to the intricacies inherent to semantic classification, the challenges of content filtering are aggravated by the sheer amount of data social networks host and distribute. An automatic algorithm may be a strong ally to allow the detection of problematic content, but it is crucial that the number of false positives is kept low; otherwise the human agents will be overwhelmed.

In this paper, we address the posting of material considered inappropriate for the audience of the community, like pornographic or violent videos. Some hosts prohibit the posting of that material altogether, while others allow it, provided that it is especially flagged as "adult content". Nevertheless, the content still ends up appearing where it is not welcome, because of either user ignorance of the rules, or full-fledged malice, when it is used to elicit revolted or shocked reactions from other users. Social networks face particular challenges, since content hosts and providers may face economic and, in some jurisdictions, even legal drawbacks (see §1 in [2]) if they not provide adequate means to protect their users from that kind of material.

We propose a simple scheme, inspired on voting algorithms, a popular technique which has been used in many tasks ranging from parameter estimation [34] to object detection [7] and video classification [35]. Using our scheme, we are able to combine several evidences (coming from classifiers using different features computed over different video elements) in order to obtain the decision on whether or not the video belonging to the unwanted class (e.g., pornography, violence, etc.). The idea is to ask several classifiers which label they attribute to the video. The "opinions" are counted, and the final decision is given by majority vote.

 ©2012 SBC — Soc. Bras. de Computação

This paper presents two main contributions. First and foremost, the rigorous evaluation of several combinations of descriptors on two applicative scenarios, which has consistently indicated that a representation based on spatiotemporal bags of features is more discriminative than all alternative evidences.

In addition, we show that, with the right choice of features, a simple scheme, based on majority voting, is able to cope well with the difficult task of content filtering for social networks. We have evaluated our technique in very challenging datasets, conceived to represent the diversity of social networks, obtaining very promising results, with a good compromise between selectivity and specificity.

It is noteworthy that architecture proposed is very flexible, and can being easily adapted to any high-level concept the user might be interested in detecting.

## 2. Prior art

### 2.1. Video feature extraction

Semantic classification of visual documents has only become feasible after the emergence of effective feature extraction algorithms. Many of those may be applied to video, some being just still-image descriptors of individual frames, others being specially conceived to take into account the spatiotemporal nature of the moving image.

Though global image descriptors may be employed to characterize video frames, in the recent years a great deal of interest has been directed to local descriptors. Those are associated to different features of the image (regions, edges or small patches around points of interest) and have been shown to provide great robustness and discriminating power [17][18][22][23][24][25].

The most popular local descriptor, SIFT [7], is both a point of interest detector, based on differences of Gaussians and a local descriptor, based on the orientations of grayscale gradients. Using SIFT, visual content is represented by a set of scale and rotation invariant descriptors, which provides a characterization of local shapes. The generated descriptors allow for adequate levels of affine, viewpoint and illumination invariance.

Since color information is considered important for many tasks (e.g., nude detection), color extensions of SIFT have been proposed [9][19]. For example, a SIFT descriptor adapted to carry hue information (aptly named HueSIFT) had been proposed [9]. It provides color distinctiveness in addition to shape distinctiveness.

Intuition tells us that temporal information should be of prominent importance for recognition tasks in videos, for being likely to indicate interesting patterns of motion. Considering that, a few local features detectors and descriptors have been proposed, take into account the temporal nature of video [6][13][14][15][16]. For example, STIP [6] is designed as a differential operator, simultaneously considering extrema over spatial and temporal scales that correspond to particular patterns of events in specific locations. It extends the Harris corner detector [8] to the temporal domain, finding interest points as moving corner changes direction across a sequence — if the corner movement is constant, no interest point is detected. That allows detecting noteworthy “events” in the video sequence.

 ©2012 SBC — Soc. Bras. de Computação

## 2.2. Codebooks of visual features

The discriminating power of local descriptors is extremely advantageous when matching objects in scenes, or retrieving specific target documents. However, when considering high-level semantic categories, it quickly becomes an obstacle, since the ability to generalize becomes then essential. A solution to that problem is to quantize the description spaces by using codebooks of local descriptors, in a technique sometimes named "visual dictionary". The visual dictionary is nothing more than a representation that splits the descriptor space into multiple regions, usually by employing non-supervised learning techniques, like clustering. Each region becomes then a "visual word".

The idea is that different regions of the description space will become associated to different semantic concepts, for example, parts of the human body, corners of furniture, vegetation, clear sky, clouds, features of buildings, etc. Yet, it is important to emphasize that this association is latent; there is no need to explicitly attribute meanings to the words. The technique has been employed successfully on several works for retrieval and classification of visual documents [1][5][11].

Once the codebook is obtained, description is greatly simplified, since it is no longer based on the exact value of descriptors, but only on their associated "words". The condensed description may be, for example, a histogram or simply a set of the words the video contains. That has two advantages: the rougher description is better adapted to complex semantics; and the computational burthen is alleviated, since algorithms now operate on a single summarized description, instead of a myriad of individual local descriptors.

Building the dictionary requires the quantization of the description space, which can be obtained by a clustering algorithm. However, state-of-the-art clustering methods are seldom (if ever) conceived for the needs of visual dictionary construction: high-dimensional spaces, large datasets and a large number of clusters. The commonest choice found in the literature is a combination of aggressive sub-sampling of the dataset, dimensionality reduction using PCA (*Principal Component Analysis*), and clustering using a simple or hierarchical *k-means* algorithm with Euclidean distance. That typical choice however, may be considerable faulty on several grounds [20], and the design of good methods for visual dictionary creation is an active theme of investigation.

In addition to moderating the discriminating power of descriptors, the dictionaries allow adapting to visual documents techniques formerly available only to textual data. Among those borrowings, one of the most successful has been the technique of *bags of words* (which considers textual documents simply as sets of words, ignoring any inherent structure). The equivalent in the CBIR universe has been called *bags of visual words, bags of features* or *bags of visual features,* sometimes abbreviated as BoVF. It greatly simplifies document description, which becomes a histogram of the visual words it contains. The introduction of that technique had a huge impact on content-based retrieval and classification of visual documents [12].

The straightforward extension of BoVF to video uses individual frame images (or selected keyframes). That allows representing semantic concepts that are independent from motion. However, previous works in human annotation of video

 ©2012 SBC — Soc. Bras. de Computação

databases [36], indicate that even for humans, many important concepts can only be adequately apprehended by taking into account the temporal aspects of video. Therefore, an interesting possibility is making codebooks of space-time local descriptors [6], which take into account the dynamic aspects of video. In that work we evaluate the performance of both static and "motion-aware" bags of features.

### 2.3. Video content filtering

The importance of pornography detection in visual documents is attested by the large literature on the subject. The vast majority of those works is based on the detection of human skin, and suffers from a high rate of false positives in situations of non-pornographic body exposure (like in sports). Some works use secondary criteria (like the shape of the detected skin areas, rejection of facial close-ups, etc.) to lower that rate. A comprehensive survey on skin-detection based methods may be found in [2].

Few methods have explored other possibilities. Bags of visual features (explained in the previous section) have been employed for many complex visual classification tasks, including pornography detection in images and videos [1][4][10]. Those works, however, have explored only bags of static features. Kim et al. [3] compare the effectiveness of several MPEG-7 features, but again, concentrate only on static features, ignoring those related to motion. Very few works have explored spatiotemporal features or other motion information for detection of pornography [37][38][39]. Jansohn [39] uses bags of static visual features and analysis of motion, including motion histograms, as separate evidences, but does not consider bags of spatiotemporal features

Violence detection has been addressed in works targeting applications as diverse as surveillance systems and movie rating. As expected, the application scope greatly affects how the problem is attacked: for example, in video-surveillance, movies are often black and white, noisy and silent [28][29]; in feature action movies, the soundtrack is often very indicative of the scene action, and so on [31][32] (to the point that some works are based solely on soundtrack evidences [30]).

In [28], a hierarchical approach for detection of violence in surveillance videos is proposed. Several actions involving two people are detected: fist fighting, kicking, hitting with objects, among others. The information of motion trajectories of the image structures in the scenes is obtained with the computation of acceleration measure vectors and their jerks. However, that method poses some limitations, failing for situations involving more than two people and when fighters fall down to the ground. Siebel and Maybank [29] developed a surveillance system for aiding human operators in monitoring undesirable events in a metro station. In [33], regions whose color indicated the presence of skin and blood are analyzed to detect aggressive actions in movies. Later, motion intensities of those regions of interest are computed, higher values indicating violence. We could not find any of approach dealing within the diversity of videos found in social networks, neither employing bags of visual features representations.

## 3. The proposed scheme

The proposed scheme is very simple and works by extracting elements from the video (shots, frames, keyframes, etc.), extracting features from those elements (global

 ©2012 SBC — Soc. Bras. de Computação

features, bags of visual features based on local features, statistics, etc.) and training different classifiers for each type of feature used. In the classification phase, the classifier opinion is asked for each individual video element, and the final decision is reached by majority voting. The whole scheme is illustrated on Figure 1 and explained, in detail, below.

Pre-processing (video element and feature extraction) step:

1. The elements of each video are extracted (shots, frames, keyframes, etc.);
2. The features are extracted from the appropriate elements of the video. Those may be visual features, statistics, etc.

Training step:

1. A SVM classifier is created for each type of feature [21]. In our work, we have used a linear kernel (which in preliminary tests, has offered the best results);
2. Each classifier is trained with the corresponding features. Care is taken to balance the classes (positive and negative) so each is given roughly the same number of training samples at this step;

Classification step:

1. Each SVM classifier is asked about each single feature concerning all elements of the video related to that feature (i.e., if a feature is computed over keyframes, there will be a feature available for every keyframes, and the corresponding classifier will be asked once for each one of those features);
2. Every time it is enquired, a SVM classifier casts a vote: positive (the video is "unwanted") or negative (the video is "ok");
3. Those votes are counted for all classifiers on all features concerning the video. The majority label is given to the video.

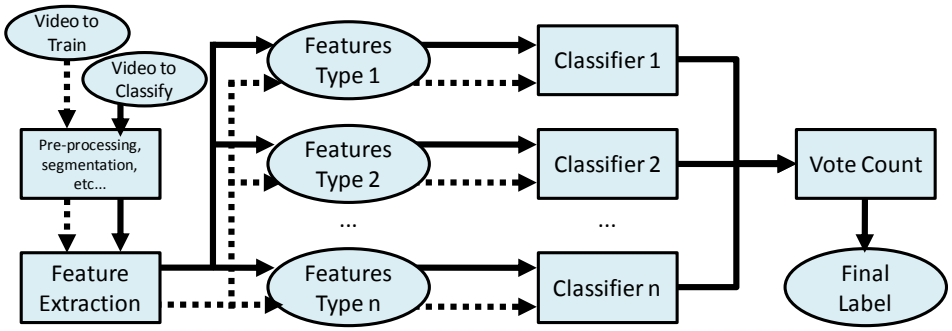



**Figure 1: The proposed scheme for video classification. The data flow for training is represented by the dashed lines, while the data flow for classification is on continuous lines. Each classifier works on a different type of feature (e.g., color histogram, "bag" of local features, etc.) potentially computed over different video elements (frames, shots, etc.). The final label is obtained by majority voting over the opinion of all classifiers. That makes the scheme very robust.**

## 4. First test application: pornography detection

Pornography is less straightforward to define than it may seem at first, since it is a high-level semantic category, not easily translatable in terms of simple visual characteristics. Though it certainly relates to nudity, pornography is a different concept: many activities

 ©2012 SBC — Soc. Bras. de Computação

that involve a high degree of body exposure have nothing to do with it. That is why systems based on skin detection [2] often accuse false positives in contexts like beach shots or sports.

A commonly used definition is that pornography is the portrayal of *explicit* sexual matter with the *purpose* of eliciting arousal. That raises several challenges. First and foremost what threshold of explicitness must be crossed for the work to be considered pornographic? Some authors deal with that issue by further dividing the classes [1][3] but that not only falls short of providing a clear-cut definition, but also complicates the classification task. The matter of purpose is still more problematic, because it is not an objective property of the document.

### 4.1. Test database

We have opted to keep the evaluation conceptually simple, by assigning only two classes (porn and non-porn). On the other hand, we took great care to make them representative of the diversity found on social networks.

For the pornographic class, we have browsed social networks which only host that kind of material (solving, in a way, the matter of purpose) and sampled 400 videos as broadly as we could (Table I) — the database contains several genres of pornography and depicts actors of many ethnicities (Table II).

For the non-pornographic class we have browsed general-public social networks and selected two samples: 200 videos chosen at random (which we called "easy") and 200 videos selected from textual search queries like "beach", "wrestling", "swimming", which we knew would be particularly challenging for the detector ("difficult").

**Table I: A summary of the test database for pornography detection.**

| Class | Videos | Hours | Shots per Video |
|---|---|---|---|
| Porn | 400 | 57 | 15.6 |
| Non-Porn ("Easy") | 200 | 11.5 | 33.8 |
| Non-Porn ("Difficult") | 200 | 8.5 | 17.5 |
| **All videos** | **800** | **77** | **20.6** |

**Table II: Ethnic diversity on the pornographic videos.**

| Ethnicity | % of Videos |
|---|---|
| Asians | 16 % |
| Blacks | 14 % |
| Whites | 46 % |
| Multi-ethnic | 24 % |

### 4.2. Experimental setup

For that application, the scheme has been parameterized as follows:

1. The two classes considered were porn (positive) and non-porn (negative). It is important to notice that the "easy" and "difficult" non-porn videos are considered in the same classification class. The differentiation is important only for the detailed analysis.

 ©2012 SBC — Soc. Bras. de Computação

2. The video elements considered are video shots and the middle-frame of each shot. Video shots were obtained by an industry-standard segmentation software[1].
3. The following features computed for the frames:
   - Color Histogram: a normalized 64-bin RGB color histogram;
   - SIFT-BoVF: 5000-bin normalized BoVF using the SIFT descriptor [7];
   - HueSIFT-BoVF: a 5000-bin normalized BoVF using the HueSIFT descriptor [9];
4. The following feature was computed for the shots:
   - STIP-BoVF: a 5000-bin normalized BoVF using the STIP descriptor [6].

Obtaining a baseline to compare with our method was a major challenge since, in general, the numbers reported on the literature are not comparable from one work to another. Often, the databases are given only very cursory description, making next to impossible to make a fair assessment of the actual experimental conditions. Therefore, we have opted to compare ourselves to PornSeer Pro, an industry standard video pornography detection system, which is readily available for evaluation purposes[2]. PornSeer Pro is based on the detection of specific features (like breast, genitals or the act of intercourse) on individual frames. It examines each individual frame of the video.

The experimental design was a classical 5-fold cross-validation, generating approximately 640 videos for training and 160 for testing on each fold.

### 4.3. Results

Figure 2 shows the performance, in the ROC space, of our detector using different combinations of features. It also shows the performance of the baseline method chosen, PornSeer. The graph reveals that several configurations of our detector are not significantly worse than PornSeer, and suggests that a few are significantly better. That latter statement, however, requires a more stringent statistical test, because we are comparing several configurations at once [26][27].

Therefore, we have performed an ANOVA test, using the 5 runs for all configurations shown in the graph. The model, using the configuration as a factor, was deemed as significant, with a *p*-value of less than 0.01 for both axes. That authorized us to perform pair-wise t-tests between the configurations, Table V shows the *p*-values obtained. We have highlighted values below 0.05 as significant.

The confusion matrix is another way to express the results shown in the ROC graphs. We showed the matrices for PornSeer (Table III) and for, arguably, the best configuration of our detector, using just the STIP descriptor (Table IV).

**Table III. The average confusion matrix for PornSeer.**

| | | Video was labeled as | |
|---|---|---|---|
| | | Porn | Non-porn |
| Video was | Porn | 65.1 % | 34.9 % |
| | Non-porn | 12.5 % | 87.5 % |

[1] http://www.stoik.com/products/svc/
[2] http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro.htm

 ©2012 SBC — Soc. Bras. de Computação

**Table IV. The average confusion matrix for our scheme using STIP.**

| | | Video was labeled as | |
|---|---|---|---|
| | | Porn | Non-porn |
| Video was | Porn | 91.3 % | 8.7 % |
| | Non-porn | 7.5 % | 92.5 % |

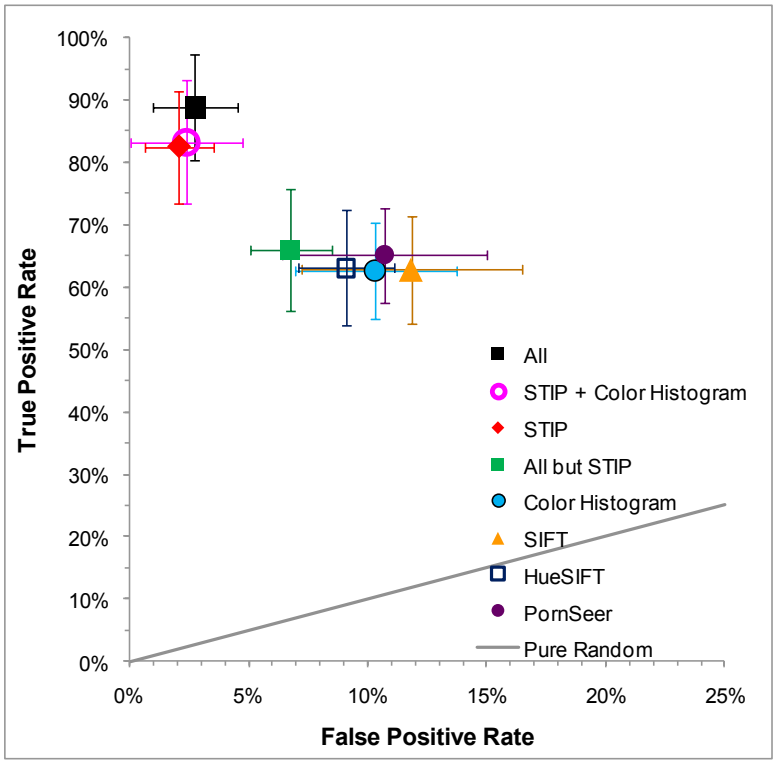



**Figure 2: A few selected points of the above curves. The error bars are confidence intervals on the respective dimensions (for $\alpha$ = 0.05). The graph shows that several configurations of our scheme are significantly better than the baseline (PornSeer).**

**Table V. P-values of pairwise t-test of detector configurations (and PornSeer), with significant differences in boldface. The pairwise tests were done after the ANOVA of all configurations was deemed significant (bottom row).**

| | True Positive Rates | | | | | | | | False Positive Rates | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **1 – Color Hist.** | — | .817 | .459 | **.015** | **.049** | .353 | **.041** | .228 | — | .169 | .723 | **.020** | **.006** | .763 | **.021** | .289 |
| **2 – SIFT** | .817 | — | .333 | **.008** | **.030** | .249 | **.024** | .327 | .169 | — | .300 | **.001** | **<.001** | .097 | **.001** | **.019** |
| **3 – Hue SIFT** | .459 | .333 | — | .075 | .202 | .848 | .174 | .057 | .723 | .300 | — | **.009** | **.002** | .513 | **.009** | .161 |
| **4 – STIP** | **.015** | **.008** | .075 | — | .590 | .108 | .650 | **.001** | **.020** | **.001** | **.009** | — | .589 | **.040** | .993 | .178 |
| **5 – STIP + Hist.** | **.049** | **.030** | .202 | .590 | — | .275 | .932 | **.003** | **.006** | **<.001** | **.002** | .589 | — | **.012** | .583 | .064 |
| **6 – All but STIP** | .353 | .249 | .848 | .108 | .275 | — | .240 | **.038** | .763 | .097 | .513 | **.040** | **.012** | — | **.041** | .444 |
| **7 – All** | **.041** | **.024** | .174 | .650 | .932 | .240 | — | **.002** | **.021** | **.001** | **.009** | .993 | .583 | **.041** | — | .181 |
| **8 – PornSeer** | .228 | .327 | .057 | **.001** | **.003** | **.038** | **.002** | — | .289 | **.019** | .161 | .178 | .064 | .444 | .181 | — |
| **Model *p*-value** | | | | **.0073** | | | | | | | | **.0009** | | | | |


©2012 SBC — Soc. Bras. de Computação

### 4.4. Discussion

In its optimal configuration, our scheme is able to correctly identify 9 out of 10 of the pornographic clips, with few false positives. That is very important, since, as we have discussed, the cost of false alarms is high on the social network context, for it tends to overwhelm the human operators. The false positive rate attained may appear high at first, but it must be taken in the context of a very challenging dataset. Considering that half of the non-pornographic test videos were difficult cases, the rates are, actually, low.

It is instructive to study the cases where our method fails. The stubborn false positives correspond to very challenging non-pornographic videos: breastfeeding sequences, sequences of children being bathed, and beach scenes. The method succeeds for many videos with those subjects, but those particular ones have the additional difficulty of having very few shots (typically 1 or 2), giving no allowance for classification errors. PornSeer gave a wrong classification for all those clips.

The analysis of the most difficult false negatives revealed that the method has difficult when the videos are of very poor quality (typical of amateur porn, often uploaded from webcams) or when the clip is only borderline pornographic, with few explicit elements. PornSeer also had difficulty with those clips, misclassifying many of them.

The study of Table V reveals interesting information. It shows that several configurations of our method have significantly better true positive rate than PornSeer, without increasing significantly the false positive rate.

More interestingly, it shows that STIP, the spatiotemporal descriptor, is critical in obtaining those good results. STIP used alone beats, in at least one of the axis, all configurations that do not use STIP; and it ties with all configurations that use STIP in combination with other descriptors. That suggests that not only spatiotemporal information is better for pornography detection in video, but also that combining it with other information (including color!) does not ameliorate the results.

Accumulation of evidences, however, seems to be useful when none of them is much compelling. Though none of the configurations using a single descriptors (other than STIP), is able to beat PornSeer, when the three “weak” descriptors (SIFT, HueSIFT and Color Histogram) are used together, they achieve significantly better detection rates.

## 5. Second test application: violence detection

Violence database suffers from the same problem of pornography detection: although it is a topic of great interest, with an abundant literature, the community lacks a shared violence dataset. In addition, existing works in do not describe the dataset used in enough details to allow a fair comparison. The matter is aggravated by the fact most works have more constrained applicative scope than ours: in the context of video-surveillance or feature movies, the data has more regularity to exploit than in the wild context of social networks, where there is less useful *a priori* information.

 ©2012 SBC — Soc. Bras. de Computação

### 5.1. Test database

We have assembled a database containing 216 videos, 108 violent and 108 non-violent. The violent video clips come from very diverse contexts, including violent sports, street fights, civil unrest, etc. They also come from diverse sources: broadcasting, cell phones, video-surveillance cameras, etc.

### 5.2. Experimental setup

For this application, the scheme has been parameterized as follows:

1. The two classes considered were violent (positive) and non-violent (negative).
2. The video elements considered are video shots and the middle-frame of each shot. Video shots were obtained by an industry-standard segmentation software1.
3. The following features computed for the frames:
   - SIFT-BoVF: 100-bin BoVF using the SIFT descriptor [7];
4. The following feature was computed for the shots:
   - STIP-BoVF: a 100-bin BoVF using the STIP descriptor [6].

The evaluation of the classification process was designed and conducted using the traditional 5-fold cross validation scheme, having approximately 160 videos for training and 40 for testing on each fold.

### 5.3. Results

The classification performances are presented in Table VI, for SIFT BoVF, and in Table VII for STIP BoVF. Though the results with SIFT are not bad (with over 80% accuracy for both classes) the results using STIP are impressive, scoring perfect results.

**Table VI: Violent video classification using SIFT-BoVF.**

| | | Video was labeled as | |
|---|---|---|---|
| | | Violent | Non-violent |
| Video was | Violent | 80.9% | 19.1% |
| | Non-violent | 5.0% | 95.0% |

**Table VII: Violent video classification using STIP-BoVF.**

| | | Video was labeled as | |
|---|---|---|---|
| | | Violent | Non-violent |
| Video was | Violent | 100% | 0% |
| | Non-violent | 0% | 100% |

### 5.4. Discussion

The analysis of the results indicates that local spatiotemporal features are decisive to distinguish between the violent and non-violent descriptors. A closer investigation at the SIFT features indicated that misclassification was, at least to some extent, due to cluttered backgrounds, low-quality frames and scenes of crowded people where the random poses make it extreme challenging to differentiate between violent and non-

 ©2012 SBC — Soc. Bras. de Computação

violent situations. In those situations, the spatiotemporal descriptor (STIP) was able to provide additional motion information, allowing the classifier to reach the right conclusion. In fact, the spatiotemporal events typical of violent videos are so distinctive, that, though the classifier sometimes misses a shot or two, after majority voting is applied, no video is misclassified.

## 6. Conclusions

In both tasks, the spatiotemporal bags of features performed significantly better than all competing representations. That indicates that motion information is relevant for identifying the complex semantic categories present in the pornography and violence tasks. That result is not trivial: the state-of-art approaches on pornography detection, for example, are still heavily based upon (static) color and texture skin detection; violence detection approaches often use motion information without considering the advantages of semantic generalization that the codebook representation is able to provide.

When employing that recommended representation, the proposed scheme shows encouraging results in both tasks, even though the datasets employed were very challenging. A large fraction of unwanted videos is detected, without incurring in excessive false negatives.

In this article, we have evaluated different *visual* features as competing representations for the task of content filtering. Therefore, we have considered visual content as the main source of information. However, it is important to note that the proposed scheme disregards the media modality from where the features are extracted: textual, soundtrack and social interaction information could all be fed to the classifiers. In fact, we believe that, for extremely complex semantic tasks, a multimodal approach is needed to warrant the best performance possible.

Though our experiments indicate that incorporating information from other descriptors does not significantly improves the performance of the scheme using only STIP, we would like to explore non-trivial ways of incorporating that information. We were, for example, surprised by the fact the addition of color information did not improve the results (STIP is “color blind”, and, intuitively, color should be an important indicator of pornographic content). We would like to test if incorporating color directly inside the descriptor might improve the results.

The extraction of STIP features is currently extremely expensive, at around 1 *frame* per second! That severely limits the usefulness of the highly discriminant spatiotemporal descriptors for industrial applications. It is thus important to find ways to compute or to approximate those descriptors at reduced cost, especially for social networks/web-scale applications. The other steps of feature extraction are not expensive and, more importantly, scale well. The SVM classifier is relatively expensive to train, especially in terms of memory (the training set must be in RAM), but the trained model is cheaper to apply (the candidate element is compared only to the support vectors).

One interesting observation for both the pornography and violence applications is that label confidence is asymmetric on the training phase. When a training video is labeled as negative, we may be confident that none of its elements is positive. But a training video is labeled as positive may contain several negative shots, frames and keyframes where no violence or pornography is present (opening and closing credits,

 ©2012 SBC — Soc. Bras. de Computação

"cut scenes", etc.). Therefore, at least for the positive class, the training is only weakly supervised. Future versions of the scheme could take that information into account.

## Acknowledgements

The authors thank CAPES, CNPq, FAPEMIG and FAPESP for the financial support that made this work possible. The local descriptors employed (SIFT, HueSIFT and STIP) were extracted with executable code provided by the authors of those methods.

## References

[1] T. Deselaers, L. Pimenidis and H. Ney. "Bag-of-Visual-Words Models for Adult Image Classification and Filtering", In: *Proceedings of the Int. Conference on Pattern Recognition (ICPR'08)*, pp. 1-4, 2008.

[2] W. Kelly, A. Donnellan, D. Molloy. "Screening for Objectionable Images: A Review of Skin Detection Techniques", In: *Int. Machine Vision and Image Processing Conference* (*IMVIP'08)*, pp. 151-158, 2008.

[3] W. Kim, S.J. Yoo, J-s. Kim, T.Y. Nam, and K. Yoon. "Detecting Adult Images Using Seven MPEG-7 Visual Descriptors", In: *Web and Communication Technologies and Internet-Related Social Issues – HSI 2005*, PP. 336-339, 2005.

[4] A. P. B. Lopes, S. E. F. de Avila, A. N. A. Peixoto, R. S. Oliveira, and A. de A. Araújo. "A Bag-of-Features Approach Based on Hue-SIFT Descriptor for Nude Detection", In: *European Signal Processing Conference (EUSIPCO'09)*, pp. 1552-1556, 2009.

[5] J. Sivic and A. Zisserman. "Video Google: A Text Retrieval Approach to Object Matching in Videos", In: *Proceedings of the IEEE Int. Conference on Computer Vision (ICCV'03)*, pp. 1470-1477, 2003.

[6] I. Laptev. "On Space-Time Interest Points", In: *Int. Journal of Computer Vision (IJCV'05)*, vol 64, number 2/3, p.107-123, 2005.

[7] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints", In: *Int. Journal of Computer Vision (IJCV'04)*, vol. 60, no. 2, pp. 91-110, 2004.

[8] C. Harris and M. Stephens. "A Combined Corner and Edge Detector", In: *Alvey Vision Conference*, pp. 147-152, 1998.

[9] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek. "Evaluating Color Descriptors for Object and Scene Recognition", In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (in press)*, 2010.

[10] A. P. B. Lopes, S. E. F. Avila, A. N. A. Peixoto, R. S. Oliveira, M. M. Coelho and A. A. Araújo. "Nude Detection in Video Using Bag-of-Visual-Features", In: *Proceedings of the 22th Brazilian Symposium on Computer Graphics and Image (SIBGRAPI)*, pp. 224-231, 2009.

[11] Y. G. Jiang, C. W. Ngo and J. Yang. "Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval", In: *Proceedings of the 6th ACM Int. Conference on Image and Video Retrieval (CIVR '07)*, pp. 494-501, 2007.

[12] J. Yang, Y.-G. Jiang, A. G. Hauptmann and C.-W. Ngo. "Evaluating Bag-of-Visual-Words Representations in Scene Classification", In: *Proceedings of the Int. Workshop on Multimedia Information Retrieval (MIR'07)*, pp. 197-206, 2007.

[13] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features", In: *Proceedings of the 14th Int. Conference on Computer Communications and Networks (ICCCN'05)*, pp. 65-72, 2005.

[14] H. Ning, Y. Hu and T. Huang. "Searching Human Behaviors Using Spatial-Temporal Words", In: *Proceedings of the IEEE Int. Conference on Image Processing (ICIP'07)*, pp. 337-340, 2007.

[15] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words", In: *Int. Journal of Computer Vision (IJCV'08)*, vol. 79, no. 3, pp. 299-318, 2008.

[16] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-Temporal Shape and Flow Correlation for Action Recognition", In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1-8, 2007.

 ©2012 SBC — Soc. Bras. de Computação

[17] K. Mikolajczyk and C. Schmid. "An Affine Invariant Interest Point Detector", In: *Proceedings of the 7th European Conference on Computer Vision-Part I (ECCV'02)*, vol. 2350 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Copenhagen, Denmark, pp. 128-142, 2002.

[18] T. Tuytelaars and L. Van Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions", In: *British Machine Vision Conference (BMVC'00)*, pp. 412-425, 2000.

[19] A. E. Abdel-Hakim and A. A. Farag. "CSIFT: A SIFT Descriptor with Color Invariant Characteristics", In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pp. 1978-1983, 2006.

[20] F. Jurie and B. Triggs. "Creating Efficient Codebooks for Visual Recognition", In: *Proceedings of the IEEE Int. Conference on Computer Vision (ICCV'05)*, vol. 1, pp. 604-610, 2005.

[21] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines", 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] T. Tuytelaars and K. Mikolajczyk. "Local Invariant Feature Detectors: A Survey", In: *Foundations and Trends in Computer Graphics and Vision*. vol. 3, no. 3, pp.177-280, 2008.

[23] K. Mikolajczyk and C. Schmid. "Indexing Based on Scale Invariant Interest Points", In: *Proceedings of the IEEE Int. Conference on Computer Vision (ICCV'01)*, pp. 525-531, 2001.

[24] C. Schuldt, I. Laptev and B. Caputo. "Recognizing Human Actions: A Local SVM Approach", In: *Proceedings of the Int. Conference on Pattern Recognition (ICPR'04)*, pp. III: 32-36, 2004.

[25] K. Mikolajczyk and C. Schmid. "A Performance Evaluation of Local Descriptors", In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n. 10, pp. 1615-1630, 2005.

[26] J. Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". In: *The Journal of Machine Learning Research*, vol. 7, 1-30, 2006.

[27] S. Salzberg. "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach". In: *Data Mining and Knowledge Discovery*, vol. 1, no. 3, 317-327, 1997.

[28] A. Datta, M. Shah, and N. Da Vitoria Lobo, "Person-on-Person Violence Detection in Video Data", In: *Proceedings of the Int. Conference on Pattern Recognition (ICPR'02),* vol. 1, p. 433-438, 2002.

[29] N. T. Siebel and S. J. Maybank, "The Advisor Visual Surveillance System". In: *Proceedings of the ECCV 2004 Workshop Applications of Computer Vision (ACV'04)*, pp. 103-111, 2004.

[30] T. Giannakopoulos, D. I. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence Content Classification Using Audio Aeatures", In: *Hellenic Artificial Intelligence Conference SETN-06, LNAI 3955*, pp. 502-507, 2006.

[31] W. Zajdel, J. D. Krijnders, T. Andringa, and D. M. Gavrila, "CASSANDRA: Audio-Video Sensor Fusion for Aggression Detection", In: *IEEE Int. Conference on Advanced Video and Signal Based Surveillance (AVSS'07)*, pp. 200-205, 2007.

[32] J. Lin and W. Wang, "Weakly-Supervised Violence Detection in Movies with Audio and Video based Co-Training", In: *Proceedings of the 10th Pacific Rim Conference on Multimedia (PCM'09)*, pp. 930-935, 2009.

[33] C. Clarin, J. Dionisio, M. Echavez, and P. Naval, "Dove: Detection of Movie Violence Using Motion Intensity Analysis on Skin and Blood", In: *Proceedings of the 6th Philippine Computing Science Congress (PCSC'06)*, pp. 150-156, 2006.

[34] D. Ballard. "Generalizing the Hough Transform to Detect Arbitrary Shapes". In: *Pattern Recognition*, vol. 13, n. 2, 111-122. Elsevier Inc., 1981.

[35] W.-N. Lie, and C-K. Su. "News Video Classification Based on Multi-Modal Information Fusion". In: *Proceedings of the IEEE Int. Conference on Image Processing (ICIP'06)*, 1213-1216, 2005.

[36] L. Kennedy. "Revision of LSCOM Event/Activity Annotations", DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, Columbia University ADVENT Technical Report #221-2006-7, 2006.

[37] T. Endeshaw, J. Garcia, and A. Jakobsson. "Classification of Indecent Video by Low Complexity Repetitive Motion Detection". In: *IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1-7, 2008.

[38] X. Tong, L. Duan, C. Xu, Q. Tian, L. Hanqing, J. Wang, and J. Jin. Periodicity Detection of Local Motion. In: *IEEE Int. Conference on Multimedia and Expo (ICME'05)*, pp. 650-653, 2005.

[39] C. Jansohn, A. Ulges and T. M. Breuel. "Detecting Pornographic Video Content by Combining Image Features with Motion Information". In: *ACM Int. Conference on Multimedia*, pp. 601-604, 2009.

 ©2012 SBC — Soc. Bras. de Computação

# Identificação da Origem de Imagens com *Zoom* utilizando *Sensor Pattern Noise*

**Leonardo Torres[1,2], Alisson S. Nascimento[1], Valter Ramos[1], José Alencar-Neto[1], Alejandro C. Frery[2]**

[1]Departamento de Pesquisa & Desenvolvimento
Laboratório de Investigação e Perícia Digital LTDA – RASTRU
57057-780 – Maceió – Brasil

[2]Instituto de Computação
Universidade Federal de Alagoas, UFAL
57072-970, Maceió, AL – Brasil

{ljmtorres, alisson.sa.nascimento, valter, alencar}@rastru.com.br,
acfrery@gmail.com

***Abstract.*** *This paper presents an approach for identifying the source device of images with zoom through the Sensor Pattern Noise. The pattern of each camera is obtained by extracting the residual noise of $n$-images and then calculated the mean and median of these $n$-images generating a fingerprint. An adaptive threshold is obtained from the correlation of $m$-training images, drawn from the same camera investigated, with the fingerprints. Finally, we validated the proposal by applying the same correlation between the investigated image and the fingerprint of each camera, concluding that such an image belonging to a given camera case the coefiente of variation is between the thresholds set for the device. The results of this approach were satisfactory and having obtained an accuracy of $100\%$ in most cases discussed in the paper.*

***Resumo.*** *Este artigo apresenta uma abordagem para identificar o dispositivo de origem de imagens com zoom através do Sensor Pattern Noise. O padrão de cada câmera é obtido com a extração do ruído residual de $n$-imagens, sobre os quais são aplicados operadoras estatísticos, média e mediana, para a obtenção da respectiva fingerprint. Um limiar adaptativo é obtido a partir correlação de $m$-imagens de treinamento, oriundas da mesma câmera investigada, com as fingerprints. Por fim, validamos a proposta aplicando a mesma correlação entre a imagem investigada e a fingerprint de cada câmera, concluindo que tal imagem pertencente a uma dada câmera caso o coefiente de variação esteja entre os limiares definidos para o dispositivo. Os resultados desta abordagem foram satisfatórios e tendo sido obtida uma acurácia de $100\%$ na maioria dos casos discutidos neste artigo.*

## 1. Introdução

No decorrer dos últimos anos, a análise forense de vídeos e imagens digitais vem ganhando o interesse da comunidade forense. Neste contexto, destacam-se as técnicas que utilizam atributos extraídos da própria imagem ou vídeo em análise, conhecidas como *blind techniques*. Por meio dessa abordagem não há necessidade de intervenções em

 ©2012 SBC — Soc. Bras. de Computação

quaisquer das etapa da formação da imagem para posterior verificação ou validação, proporcionando vantagens sobre outras técnicas, como exemplo, as *watermarks*.

Uma imagem digital contém bem mais que somente a cena a qual se captou, cada imagem leva consigo características inerentes a sua formação, como por exemplo: (i) o coeficientes gerados por uma tabela de quantização utilizada como parâmetro no processo de compressão JPEG [Farid 2009, Bianchi and Piva 2012]; (ii) a correlação entre os pixels referente a interpolação das cores utilizando *Color Filter Array* (CFA) [Farid 2009]; (iii) o ruído padrão do fotossensor deixado na imagem, denotado *Sensor Pattern Noise* (SPN) [Lukáš et al. 2006, Kang et al. 2012]. Estas e outras informações obtidas de uma imagem digital podem provar sua autenticidade e integridade, identificar o dispositivo de origem ou a correlação entre imagens como provenientes de um mesmo dispositivo, auxiliando a análise forense em crimes digitais.

Este trabalho avança nos estudos de [Torres et al. 2012a, Torres et al. 2012b], que relatam uma dificuldade relacionada ao erro obtido nos experimentos proporcionados por imagens com *zoom* acentuado. Assim, assumimos que o efeito do *zoom* altera o ruído padrão do sensor. O que nos motiva avaliar este efeito em vários aspectos.

Nós propomos uma metodologia para identificação da origem de imagens com *zoom* (digital e/ou óptico) baseado no SPN. Apresentamos as principais condições para captura das imagens no dispositivo investigado e um modelo entidade-relacional para armazenamento destes parâmetros. As imagens investigadas passam por uma processo de extração do ruído residual, também chamado de atributos. Estes atributos servem como entrada para o processo de formação do padrão do modelo do dispositivo. Neste processo são identificados dados estatísticos comuns e inerentes às imagens e sua formação. Tais características extraídas servem de entrada para o processo de decisão orientado à utilização de um limiar adaptativo. O objetivo principal é detectar o dispositivo de origem das imagens com *zoom*.

O artigo está organizado da seguinte forma: Na Seção 2 apresentamos os trabalhos relacionados, onde são expostas as principais técnicas utilizadas. Na Seção 3 descrevemos a metodologia adotada, enfatizando as entradas e saídas de cada etapa. Na Seção 4 relatamos os experimentos e análises dos resultados e por fim, na Seção 5 nós discutimos as contribuições e trabalhos futuros.

## 2. Trabalhos Relacionados

A análise forense de imagens digitais, mais especificamente em identificação de dispositivos, é comumente utilizada para responder às questões abaixo dando suporte à comunidade forense no combate aos crimes eletrônicos:

- Dada uma imagem pode-se provar se ela foi ou não produzida por uma certa câmera ou dispositivo digital (*Camera Identification*)?
- Duas ou mais imagens podem ser correlacionadas como originadas de um mesmo dispositivo desconhecido (*Device Linking*)?

### 2.1. Aquisição de Imagem Digital

A Figura 1 mostra um modelo de aquisição de imagem digital: a luz refletida da cena e dos objetos nela contidos passa através de lentes e de filtros óticos, para que seja capturada

 ©2012 SBC — Soc. Bras. de Computação

pelos sensores de cores. A maioria das câmeras digitais usam sensores *Charge-coupled Device* (CCD) ou *Complementary Metal-Oxide-Semiconductor* (CMOS) e capturam as imagens usando um CFA. Os CFAs são matrizes, em sua maioria, com três filtros de cores em padrão RGB (*red*, *green*, *blue*), acoplados a cada sensor. Cada pixel de uma imagem digital tem três valores, um de cada cor ou banda, que compõem a cor do pixel, porém cada ponto do CFA captura um único valor de uma banda específica, os valores das outras bandas são inexistente, até então. O processo de interpolação usa os valores da vizinhança do pixel nas outras bandas para estimar os valores inexistentes, tornando possível a exibição da imagem. Por fim, é aplicado um pós-processamento onde, por exemplo, pode ser feito uma compressão JPEG.

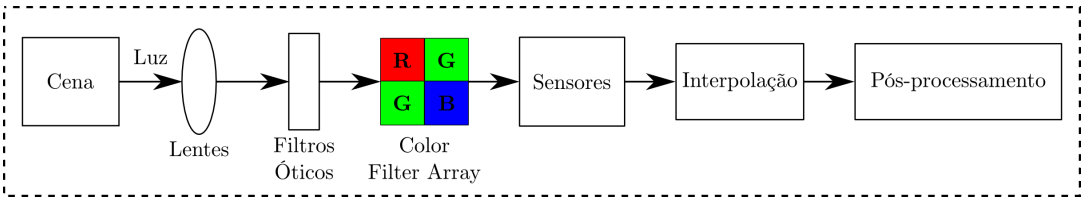



**Figura 1. Modelo de aquisição de imagem digital.**

## 2.2. Ruído Padrão do Sensor

Muitos autores vêm se dedicando no uso do SPN como principal característica para identificar um dispositivo, como em [Lukáš et al. 2006, Celiktutan et al. 2008, Li 2010, Kang et al. 2012]. Diferentemente de outras técnicas, a análise por SPN não necessita de conhecimento *a priori* das características físicas do dispositivo, evitando-se a necessidade de abordagem intrusiva sobre os dispositivos na investigação de *Camera Identification*, ou mesmo a necessidade da posse do dispositivo original nas perícias de *Device Linking*.

Uma outra objeção em se utilizar técnicas baseadas em características físicas é a facilidade de adulteração das informações que as imagens levam no cabeçalho (e.g. EXIF), por exemplo, diminuindo sua credibilidade.

A motivação deste trabalho é a utilização de técnicas não intrusivas e uma estrutura de difícil adulteração. Essas premissas levam à utilização do ruído padrão do fotosensor, o qual se torna eficaz visto que cada dispositivo deixa na imagem um tipo de impressão digital (*fingerprint*), uma marca única e inerente a cada sensor, mesmo em marcas e modelos iguais. Com esta *fingerprint* é possível identificar o dispositivo da qual uma imagem foi capturada [Redi et al. 2011].

Lukáš et al. [Lukáš et al. 2006] descrevem que o ruído produzido no processo de aquisição pode ser dividido em duas categorias: (i) *Fixed Pattern Noise* (FPN) e (ii) *Photoresponse Nonuniformity* (PRNU). O FPN é formado por pontos escuros, pontos do sensor (CCD ou CMOS) não expostos à luz. Em uma imagem natural o que predomina é a PRNU, que é formada pelo ruído devido a não uniformidade dos pixels (*Pixel Nonuniformity* – PNU) e pelos componentes de baixa frequência. Todos os sensores não recebem a mesma intensidade de luz da cena, tornando-se mais uma característica do processo de fabricação do dispositivo. Porém, diferentemente do FPN, a temperatura não é considerada para contribuição na PNU.

 ©2012 SBC — Soc. Bras. de Computação

### 2.3. Ruído Residual

Os componentes de baixa frequência de uma imagem digital advêm dos elementos da cena capturada. A luz refratada por partículas de poeira, superfícies ópticas e configurações de *zoom* também geram componentes de baixa frequência.

A supressão do ruído é um passo importante no processo de investigação e identificação de dispositivos através do SPN. Dos trabalhos como de [Lukáš et al. 2006, Chen et al. 2008, Li 2010, Goljan et al. 2011, Kang et al. 2012], obtém-se o ruído residual, denotado $w_k$, utilizando técnicas de filtragem e dado por

$$w_k = I_k - F(I_k), \tag{1}$$

em que $I_k$ representa uma imagem natural e $F$ um função de filtragem. Com isso, o SPN pode ser estimado pela média do ruído residual de $n$-imagens, denotado $\eta$ e dado por

$$\eta = \frac{1}{n}\sum_{k=1}^{n} w_k, \tag{2}$$

em que $n$ representa o número total de imagens usadas para estimar a impressão digital do sensor. A Figura 2 apresenta um exemplo do processo de pré-processamento em imagem digital para obter o ruído residual utilizando a Equação (1).

(a) Original (b) Intensidade (c) Filtrada (d) Ruído residual $w_k$

**Figura 2. Pré-processamento em imagem para obter o ruído residual.**

### 2.4. Identificação e correlação usando SPN

Para identificar um dispositivo através do SPN, utiliza-se características únicas que podem ser provenientes de imperfeições dos componentes, de efeitos do ambiente e condições de operação [Lukáš et al. 2006]. Estas características dão unicidade ao ruído produzido no processo de captação da imagem, fazendo com que seja viável a identificação do dispositivo somente através de imagens originadas deste dispositivo.

Obtido o ruído residual, como subsídio para o processo de classificação, alguns autores extraem características do ruído [Gou et al. 2007a, Celiktutan et al. 2008], outros estimam a PNRU [Chen et al. 2008, Li and Li 2012] e outros ainda usam o próprio ruído residual como entrada [Li 2010, Kang et al. 2012]. Alinhado a primeira abordagem, este trabalho utiliza características estatísticas extraídas do ruído residual como entrada para o processo de classificação.

 ©2012 SBC — Soc. Bras. de Computação

### 2.5. Processo de Classificação

O processo de decisão é responsável pela autenticação do dispositivo investigado. Esta proposta utiliza a autenticação do tipo verificação "Um para Um" e o tipo de identificação "Um para Muitos". O primeiro tipo verifica se uma imagem investigada pertence a um certo dispositivo conhecido, ou não. Enquanto, o segundo tipo identifica para um dado conjunto de câmeras se uma imagem investigada pertence a alguma das câmeras ou a nenhuma delas.

Muitos autores utilizam o *Support Vector Machine* (SVM) como processo de decisão supervisionado. Os resultados alcançados são satisfatórios, como em [Gou et al. 2007a, Gou et al. 2007b, Celiktutan et al. 2008, Kang et al. 2012].

A Curva ROC (*Receiver Operating Characteristic*) é um método estatístico, comumente utilizado para classificar indivíduos em grupos. Este método é fortemente apoiado no grupo e amostras de treinamento. A quantificação da medida do erro ou da chance do algoritmo validar corretamente é o objetivo deste método, como visto em [Gou et al. 2007a, Kang et al. 2012].

As taxas de falsa rejeição (*False Rejection Rate* – FRR) e as taxas de falsa aceitação (*False Acceptance Rate* – FAR) são um dos métodos estatísticos utilizados para traçar a Curva ROC. Porém, trabalhos como de [Lukáš et al. 2006, Chen et al. 2008] tratam diretamente com estes dados para classificar os grupos com limiares adaptativos. Mancomunado com esta última abordagem, este trabalho propõe um limiar adaptativo que é mensurado pelas taxas de FAR e FRR.

## 3. Metodologia Adotada

Nesta seção serão apresentados os materiais e métodos para validação da proposta.

### 3.1. Banco de Dados de Imagens

A primeira etapa da metodologia é a aquisição e armazenamento das imagens capturadas para formação do modelo do dispositivo e das imagens investigadas. Nesta etapa definimos alguns aspectos, como os dispositivos a serem investigados, quantidade de imagens capturadas, ambiente (urbano, em laboratório), dentre outros. Podemos definir também, dependendo da necessidade, se serão usadas imagens já existentes de uma base de dados disponível.

A coleta e armazenamento de imagens de várias câmeras não é uma tarefa fácil e nem muito econômica. O *Dresden Image Database for Benchmarking Digital Image Forensics* [Gloe and Böhme 2010] contém mais de 14 mil imagens em JPEG e TIFF de cenas variadas, capturadas de 73 modelos de câmeras diferentes. Este banco de dados é uma ferramenta de alto valor para a comunidade forense, pois apresenta uma visão clara do estado da arte em identificação de câmeras.

Inspirados pelo trabalho de Gloe e Böhme [Gloe and Böhme 2010], nós utilizamos um modelo entidade-relacional para o armazenamento de imagens e informações do processo de aquisição. Um Banco de Dados (BD) é uma coleção de dados interrelacionados, representando informações sobre um domínio específico e um Sistema Gerenciador de Banco de Dados (SGBD) provê ferramentas para acesso a estes dados.

 ©2012 SBC — Soc. Bras. de Computação

A Figura 3 mostra um modelo simplificado do BD proposto. As entidades são representadas por retângulos, os relacionamentos são representados por losangos e os atributos são as elipses. O relacionamento entre um `Fabricante` e um `Modelo` caracteriza um `Dispositivo`. Podem existir vários `Modelos` para cada `Fabricante`, assim como vários `Dispositivos` com um mesmo `Modelo` e `Fabricante`. O `Ambiente` tem tipo (urbano, céu azul, natural) e uma *flag* indicando se é `indoor` ou `outdoor`. As condições em que uma imagem foi capturada (câmera fixa, distância focal, flash) são registradas no relacionamento `Condições`, caracterizado pelo relacionamento entre o `Ambiente` e um `Dispositivo`. Cada `Imagem` recebe um identificador (`Id`) criptografado. Se a imagem for de um banco de dados externo ou online, o atributo `Base Externa` deve ser preenchido com uma descrição desta base.

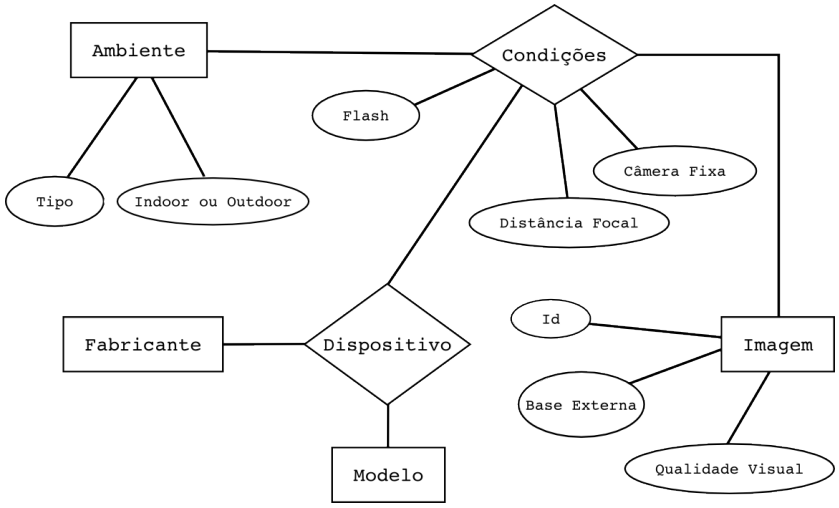



**Figura 3. Modelo Entidade-Relacional simplificado do BD usado para registrar dados e imagens de entrada dos experimentos.**

### 3.2. Extração de Atributos

Uma vez populado o BD, colhemos $n$-imagens de cada câmera registrada para estimar a *fingerprint*. Foram utilizados dois operadores: (i) a média, exposto na Equação (2) e (ii) a mediana. Cada operador é aplicado em cada ponto correspondente das $n$-imagens resultando em uma imagem de intensidade, denotada *fingerprint*.

### 3.3. Processo de Decisão

Após estimarmos a *fingerprint*, definimos um conjunto de $m$-imagens para treinamento (obtidas do mesmo dispositivo que originou a *fingerprint*). Calculamos as correlações entre a *fingerprint* e as $m$-imagens de treinamento usando a correlação de Pearson em blocos $8 \times 8$. O coeficiente de correlação de Pearson é dado por

$$\rho_v(X,Y) = \frac{\sum_{r=1}^{u}(x_r - \bar{x})(y_r - \bar{y})}{\sqrt{\sum_{r=1}^{u}(x_r - \bar{x})^2\ (y_r - \bar{y})^2}} \tag{3}$$

em que

$$\bar{x} = \frac{1}{u}\sum_{r=1}^{u} x_r, \quad \bar{y} = \frac{1}{u}\sum_{r=1}^{u} y_r$$

 ©2012 SBC — Soc. Bras. de Computação

são as médias dos blocos referente as imagens do ruído padrão da câmera $X$ e a imagem de treinamento $Y$, respectivamente, no $r$-ésimo pixel e $v = 1, \ldots, b$, onde $b$ é a quantidade de blocos $8 \times 8$ das imagens. Em posse das matrizes de correlações $Z_j$, $j = 1, \ldots, m$ uma para cada imagem de treinamento, calculamos o coeficiente de variação $\varsigma_{Z_j}$ destas matrizes, que é uma medida de dispersão relativa definida por

$$\varsigma_{Z_j} = \frac{s_{Z_j}}{\bar{Z}_j} \tag{4}$$

em que $s_{Z_j}$ é o desvio padrão e $\bar{Z}_j$ a média das matrizes de correlação $Z_j$. Assim, podemos definir um limiar $\mathcal{L}_{C_i}$ seguro para identificar o dispositivo de origem, dado por

$$\mathcal{L}_{C_i} = \bar{\varsigma_Z} \pm (2.698\ s_{\varsigma_Z}) \tag{5}$$

em que $s_{\varsigma_Z}$ é o desvio padrão e $\bar{\varsigma_Z}$ a média dos coeficientes de variação das $Z_j$, $j = 1, \ldots, m$ de $m$-imagens de treinamento. Os limiares, superior e inferior, são definidos com base no conceito do *Interquartile Range*, que é usado para construir *boxplots* sendo simples representações gráficas de uma distribuição de probabilidade. A Figura 4 apresenta um exemplo de *boxplot* com uma variação de *interquartil* de uma função densidade de probabilidade (f.d.p.) de uma Normal $N(0, \sigma^2)$.

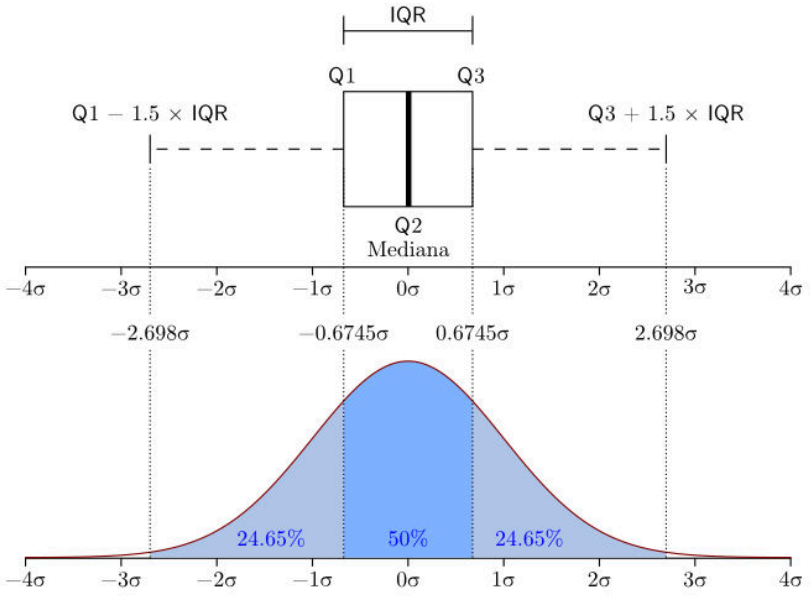



**Figura 4. *Boxplot* (com uma variação *interquartil*) de uma função densidade de probabilidade $N(0, \sigma^2)$.**

A Figura 5 apresenta um diagrama em blocos da metodologia abordada para identificação de dispositivos e o processo de validação.

## 4. Experimentos e Análises

Nesta seção apresentamos os resultados obtidos com a metodologia proposta para identificação de origem de imagens com *zoom* utilizando o ruído residual, que foi extraído com o processo de filtragem baseado em *Wavelet* 2-D. Os experimento tem por

 ©2012 SBC — Soc. Bras. de Computação

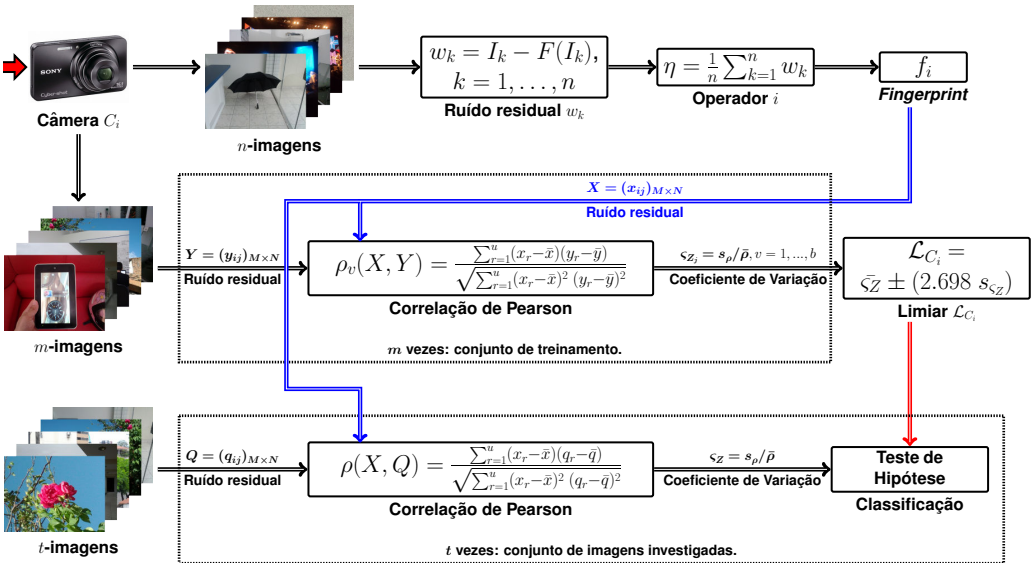



**Figura 5. Diagrama em blocos da metodologia adotada e o processo de validação.**

finalidade validar que o *zoom* altera as características do ruído padrão do sensor. Portanto, metodologias para identificar dispositivos com imagens de *zoom* é fundamental na análise forense de imagens digitais.

Para validar a proposta, registramos no BD 20 imagens com resolução de 2Mb e *zoom* máximo provenientes das 3 câmeras listadas na Tabela 1, num total de 60 imagens. As imagens tiveram uma variação entre *indoor* e *outdoor*, luz natural e iluminação controlada, sem *flash*, com *zoom* máximo em dias claros e nublados. Além disso, nenhuma imagem passou por conversão de formatos ou recompressão.

Dentre as 20 imagens de cada câmera, 10 imagens foram utilizadas para extrair o ruído padrão do sensor, 5 imagens para determinar o limiar $\mathcal{L}_{C_i}$ e as 5 imagens remanescentes foram utilizadas nos testes.

**Tabela 1. Fabricantes e modelos das câmeras utilizadas nos experimentos.**

| ID | Modelo da Câmera | Resolução Nativa | Resolução com 2Mb |
|---|---|---|---|
| $C_1$ | Samsung Galaxy Ace | $2560 \times 1920$ | |
| $C_2$ | Samsung Galaxy X | $2592 \times 1944$ | $1600 \times 1200$ |
| $C_3$ | Motorola EX115 | $2046 \times 1536$ | |

Realizamos um recorte central quadrado em todas as imagens disponíveis no BD. A ordem do recorte é igual a menor dimensão dentre as resoluções das câmeras (Tabela 1) com 2Mb, neste caso foi uma janela de ordem de 1200.

Em seguida, obtermos as *fingerprints* de cada câmera, denotada $f_1$ a obtida pela média e $f_2$ a obtida pela mediana. No experimento, cada *fingerprint* é correlacionada com as amostras destinadas para treinamento utilizando a Equação (3).

Ao final deste processo, utilizando a Equação (4) obtivemos duas tabelas. Na Tabela 2 apresentamos os coeficientes de variação entre a *fingerprint* $f_1$ e todas as amostras

 ©2012 SBC — Soc. Bras. de Computação

para treinamento; Na Tabela 3 são expostos os resultados obtidos utilizando a *fingerprint* $f_2$.

**Tabela 2. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_1$ e as imagens para treinamento.**

| Amostras | $\varsigma_{Z_j}$ da $C_1$ | $\varsigma_{Z_j}$ da $C_2$ | $\varsigma_{Z_j}$ da $C_3$ |
|---|---|---|---|
| #1 | 9.1082636 | 6.5536874 | 6.2476024 |
| #2 | 9.0166956 | 66.286325 | 9.2804927 |
| #3 | 7.8795224 | 13.314908 | 8.9350211 |
| #4 | 8.5867195 | 10.039887 | 6.9589715 |
| #5 | 8.1051964 | 12.633406 | 9.1485774 |

**Tabela 3. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_2$ e as imagens para treinamento.**

| Amostras | $\varsigma_{Z_j}$ da $C_1$ | $\varsigma_{Z_j}$ da $C_2$ | $\varsigma_{Z_j}$ da $C_3$ |
|---|---|---|---|
| #1 | 12.785679 | 12.558218 | 7.7295595 |
| #2 | 12.720633 | 52.931877 | 10.840593 |
| #3 | 12.340460 | 28.411868 | 11.189713 |
| #4 | 12.486977 | 14.835449 | 8.7692438 |
| #5 | 11.887271 | 16.970825 | 11.333466 |

Com o suporte nessas tabelas (Tabelas 2 e 3) calculamos os limiares para cada dispositivo utilizando a Equação (5). A Tabela 4 apresenta os limiares para as câmeras envolvidas neste experimento.

**Tabela 4. Limiares das câmeras envolvidas no experimento.**

| Câmeras | $\mathcal{L}_{C_i}$ da $f_1$ | $\mathcal{L}_{C_i}$ da $f_2$ |
|---|---|---|
| $C_1$ | $8.5401 \pm 1.4639$ | $12.444 \pm 0.9688$ |
| $C_2$ | $21.765 \pm 67.528$ | $25.141 \pm 45.031$ |
| $C_3$ | $8.1142 \pm 3.7970$ | $9.9725 \pm 4.3849$ |

Submetemos as 15 imagens de teste para validar a metodologia proposta. Então, correlacionamos cada uma delas com as *fingerprint* $f_1$ e $f_2$ das três câmeras utilizando a Equação (3). Em seguida, extraímos de cada resultado o coeficiente de variação $\varsigma_{Z_j}$. E por fim, para cada modelo de câmera, verificamos se cada coeficiente está dentro do limiar $\mathcal{L}_{C_i}$ (veja Tabela 4) correspondente de cada *fingerprint*.

Observamos que os testes validaram a proposta, pois obtemos como melhor resultado uma acurácia de $100\%$ ($\mathsf{FAR} = 0\%$ e $\mathsf{FRR} = 0\%$) para as câmeras $C_1$ e $C_3$. Nos testes envolvendo os padrões da câmera $C_2$ houveram algumas falsas aceitações, constatando-se para $f_1$ uma taxa de $\mathsf{FAR} = 20\%$, para o padrão da $f_2$ obteve-se $\mathsf{FAR} = 6.66\%$ e para ambos as taxas de falsa rejeição de $\mathsf{FRR} = 0\%$.

As Tabelas 5 e 6 apresentam uma sumarização dos dados provenientes dos testes com as *fingerprint* $f_1$ e $f_2$, respectivamente.

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 5. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_1$ e todas as imagens de teste.**

| Amostras | $f_1$ da $C_1$ | $f_1$ da $C_2$ | $f_1$ da $C_3$ |
|---|---|---|---|
| #1 da $C_1$ | **7.4739506** | 144.262259 | 200.317435 |
| #2 da $C_1$ | **7.6064666** | 276.449169 | 78.0545261 |
| #3 da $C_1$ | **7.9423650** | 243.403174 | 166.857883 |
| #4 da $C_1$ | **9.3948885** | 705.250362 | 283.320992 |
| #5 da $C_1$ | **12.915825** | 242.924493 | 167.770087 |
| #1 da $C_2$ | 80.4586162 | **47.7930814** | 107.406795 |
| #2 da $C_2$ | 98.1257967 | **6.55192869** | 57.2494114 |
| #3 da $C_2$ | 233.828978 | **32.0366588** | 1194.79124 |
| #4 da $C_2$ | 76.3919582 | **45.8203295** | 199.965727 |
| #5 da $C_2$ | 241.362758 | **7.47888534** | 143.754256 |
| #1 da $C_3$ | 150.275762 | 73.8402076 | **8.5283056** |
| #2 da $C_3$ | 99.3387020 | 229.770700 | **5.8790180** |
| #3 da $C_3$ | 640.188106 | 180.336746 | **6.3840531** |
| #4 da $C_3$ | 157.504542 | 225.851194 | **9.1928537** |
| #5 da $C_3$ | 1506.64921 | 146.818612 | **9.2021569** |

**Tabela 6. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_2$ e todas as imagens de teste.**

| Amostras | $f_2$ da $C_1$ | $f_2$ da $C_2$ | $f_2$ da $C_3$ |
|---|---|---|---|
| #1 da $C_1$ | **10.968657** | 611.652882 | 245.387964 |
| #2 da $C_1$ | **11.946643** | 194.629341 | 186.222023 |
| #3 da $C_1$ | **12.159662** | 463.148492 | 236.205318 |
| #4 da $C_1$ | **12.432451** | 128.644021 | 136.948880 |
| #5 da $C_1$ | **19.008034** | 222.856714 | 749.159791 |
| #1 da $C_2$ | 120.295278 | **49.2541563** | 708.06373 |
| #2 da $C_2$ | 543.796790 | **10.5647014** | 69.964838 |
| #3 da $C_2$ | 730.109501 | **43.1296481** | 267.16615 |
| #4 da $C_2$ | 134.161674 | **60.6481323** | 429.99705 |
| #5 da $C_2$ | 151.664311 | **11.7233525** | 1732.6377 |
| #1 da $C_3$ | 636.826067 | 134.206611 | **11.089052** |
| #2 da $C_3$ | 203.628253 | 160.961831 | **7.6647386** |
| #3 da $C_3$ | 263.777522 | 170.555654 | **7.9002187** |
| #4 da $C_3$ | 398.020619 | 3826.83251 | **10.635363** |
| #5 da $C_3$ | 1544.69379 | 88.6886813 | **12.378317** |

Fazendo uma análise com os trabalhos de Lukáš et al. [Lukáš et al. 2006] e Chen et al. [Chen et al. 2008], enquanto eles utilizam 300 e 30 imagens de cada câmera para obter o padrão do sensor, respectivamente, nós extraímos o padrão com 10 imagens; um número alto de imagens, neste caso, dificulta a replicação dos experimentos.

 ©2012 SBC — Soc. Bras. de Computação

Os trabalhos de [Torres et al. 2012a, Torres et al. 2012b] apresentam uma metodologia para identificar câmeras com imagens naturais e apontaram um problema relacionado as imagens obtidas com *zoom*. Nenhum dos trabalhos relacionados apontam uma investigação neste problema, o que nos motivou a apresentar esta metodologia.

Lukáš et al. [Lukáš et al. 2006] avaliam sua proposta fixando a taxa de falsa aceitação em $\mathsf{FAR} = 10^{-3}$ e para cada câmera com um limiar determinado obteve diferentes taxas de falsa rejeição e em um único caso um $\mathsf{FRR} = 0\%$. Chen et al. [Chen et al. 2008] utilizam o critério de correlação Neyman-Pearson e definem as taxas de erro por uma função de probabilidade com limiar determinado. Os resultados obtidos foram uma taxa de falsa aceitação em $\mathsf{FAR} = 10^{-5}$ para todos os testes e uma taxa de falsa rejeição variada por dispositivo investigado.

### 4.1. Experimentos com Imagens Naturais

Um segundo experimento foi realizado incluindo imagens naturais. Os objetivos deste experimento foram a verificação e validação de que as imagens com *zoom* possuem um *Sensor Pattern Noise* diferente do obtido com imagens naturais, i.e., imagens com resolução nativa do dispositivo. Neste caso, esperou-se que o teste rejeitasse todas as imagens naturais.

As Tabelas 7 e 8 apresentam uma sumarização dos dados provenientes dos testes com as *fingerprint* $f_1$ e $f_2$ (veja limiares na Tabela 4), respectivamente, e imagens naturais. Obtivemos apenas na fingerprint $f_1$ da câmera $C_2$ duas imagens naturais consideradas como imagens com *zoom*. Concluímos que realmente imagens com zoom possuem um padrão no ruído diferente das obtidas nas imagens naturais.

**Tabela 7. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_1$ e todas as imagens naturais de teste.**

| Amostras | $f_1$ da $C_1$ | $f_1$ da $C_2$ | $f_1$ da $C_3$ |
|---|---|---|---|
| #1 da $C_1$ | 357.63771 | 120.227020 | 755.13051 |
| #2 da $C_1$ | 118.59515 | 164.860738 | 640.49380 |
| #3 da $C_1$ | 116.28642 | 107.880599 | 522.40926 |
| #4 da $C_1$ | 220.03940 | 525.858653 | 92.061579 |
| #5 da $C_1$ | 287.88334 | 319.786794 | 78.285011 |
| #1 da $C_2$ | 269.47683 | 93.9499747 | 155.30173 |
| #2 da $C_2$ | 162.71504 | 102.368041 | 337.87713 |
| #3 da $C_2$ | 86.418182 | 95.5505330 | 309.06278 |
| #4 da $C_2$ | 201.81974 | 108.657699 | 720.39964 |
| #5 da $C_2$ | 174.58195 | 74.1696844 | 253.71220 |
| #1 da $C_3$ | 112.01078 | 180.758850 | 252.13608 |
| #2 da $C_3$ | 429.20247 | 161.793357 | 499.49541 |
| #3 da $C_3$ | 122.62244 | 333.947338 | 14950.007 |
| #4 da $C_3$ | 88.609697 | 170.424399 | 2912.8926 |
| #5 da $C_3$ | 75.023935 | 73.9257781 | 62.203697 |

 ©2012 SBC — Soc. Bras. de Computação

**Tabela 8. Coeficientes de variações $\varsigma_{Z_j}$ obtidos com a correlação entre o padrão $f_2$ e todas as imagens naturais de teste.**

| Amostras | $f_2$ da $C_1$ | $f_2$ da $C_2$ | $f_2$ da $C_3$ |
|---|---|---|---|
| #1 da $C_1$ | 1424.3586 | 667.054467 | 839.545042 |
| #2 da $C_1$ | 474.82659 | 86.8204075 | 178.847620 |
| #3 da $C_1$ | 166.93836 | 105.109248 | 323.747669 |
| #4 da $C_1$ | 206.81798 | 804.335190 | 144.090885 |
| #5 da $C_1$ | 228.45334 | 472.329364 | 495.148437 |
| #1 da $C_2$ | 4122.5380 | 97.4692083 | 304.207157 |
| #2 da $C_2$ | 1759.3627 | 207.321233 | 357.645415 |
| #3 da $C_2$ | 104.54454 | 185.029780 | 1731.20546 |
| #4 da $C_2$ | 141.05420 | 118.246127 | 477.151569 |
| #5 da $C_2$ | 149.48467 | 89.8849443 | 541.875687 |
| #1 da $C_3$ | 110.78581 | 135.570680 | 5817.11110 |
| #2 da $C_3$ | 1329.4938 | 941.156659 | 498.222330 |
| #3 da $C_3$ | 123.48787 | 664.965057 | 107.147830 |
| #4 da $C_3$ | 268.19444 | 92.4900005 | 1475.52870 |
| #5 da $C_3$ | 78.072928 | 94.5832611 | 82.3935090 |

## 5. Conclusões

Este trabalho propôs uma metodologia para análise forense em imagens digitais que responda à questão: Dada uma imagem, pode-se provar se ela foi ou não produzida por uma certa câmera ou dispositivo digital?

A metodologia proposta foi validada utilizando o *Sensor Pattern Noise* e duas *fingerprints* foram estimadas para cada dispositivo. Como método de organização e armazenamento apresentamos um modelo entidade-relacional simplificado, que abrange as informações das imagens capturadas e seus respectivos dispositivos.

Um grande diferencial foi abordar imagens com *zoom* e obter um limiar adaptativo baseado no coeficiente de variação $\varsigma_{Z_j}$, onde para cada *fingerprint* existe um intervalo que identifica o dispositivo. Nos melhores casos obtivemos $100\%$ de acurácia, nas câmeras $C_1$ e $C_3$ nas Tabelas 5 e 6 e, como pior resultado uma taxa de falsa aceitação $\mathsf{FAR} = 6.66\%$ com falsa rejeição de $\mathsf{FRR} = 0\%$ na câmera $C_2$ para ambos os padrões estimados usando $10$-imagens. Frente a literatura obtemos resultados satisfatórios utilizando poucas imagens para extrair o ruído padrão do sensor das imagens.

Os testes com imagens naturais serviram para constatar que o *Sensor Pattern Noise* possui características diferentes das imagens com *zoom*.

Como trabalhos futuros espera-se desenvolver um algoritmo que faça a triagem das imagens e classificando-as em imagens naturais ou imagens com *zoom* e, caso possua *zoom*, identifique qual a resolução para correlacionar com o padrão mais próximo da imagem investigada. Também espera-se utilizar um número menor de imagens para estimar o padrão do sensor. Tendo em vista, que a aquisição destas imagens na área forense, em certas condições, pode ser onerosa ou não ser possível. Outros métodos para classificação

 ©2012 SBC — Soc. Bras. de Computação

devem ser abordados para futuras comparações e junções de métodos.

Outros passos estão relacionados a ampliação do Banco de Dados para que os dados dos experimentos, da etapa de extração de atributos e do processo de decisão possam ser registrados e replicados facilmente por outros peritos.

## Referências

Bianchi, T. and Piva, A. (2012). Detection of nonaligned double jpeg compression based on integer periodicity maps. *IEEE Transactions on Information Forensics and Security*, 7(2):842–848.

Celiktutan, O., Sankur, B., and Avcibas, I. (2008). Blind identification of source cell-phone model. *IEEE Trans. on Information Forensics and Security*, 3(3):553–566.

Chen, M., Fridrich, J., Goljan, M., and Lukáš, J. (2008). Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90.

Farid, H. (2009). Image forgery detection. *IEEE Signal Processing Magazine*, 26(2):16–25.

Gloe, T. and Böhme, R. (2010). The dresden image database. Online. acessado em 17 de agosto de 2012.

Goljan, M., Fridrich, J., and Chen, M. (2011). Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security*, 6(1):227–236.

Gou, H., Swaminathan, A., and Wu, M. (2007a). Noise features for image tampering detection and steganalysis. In *IEEE International Conference on Image Processing*, volume 6, pages VI–97–100.

Gou, H., Swaminathan, A., and Wu, M. (2007b). Robust scanner identification based on noise features. In *Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA.

Kang, X., Li, Y., Qu, Z., and Huang, J. (2012). Enhancing source camera identification performance with a camera reference phase sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 7(2):393–402.

Li, C.-T. (2010). Source camera identification using enhanced sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 5(2):280–287.

Li, C.-T. and Li, Y. (2012). Color-decoupled photo response non-uniformity for digital image forensics. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(2):260–271.

Lukáš, J., Fridrich, J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214.

Redi, J. A., Taktak, W., and Dugelay, J.-L. (2011). Digital image forensics: a booklet for beginners. *Multimedia Tools and Applications*, 51(1):133–162.

 ©2012 SBC — Soc. Bras. de Computação

Torres, L., Nascimento, A. S., Ramos, V., and Alencar-Neto, J. (2012a). Identificação de câmeras baseada no ruído padrão do sensor usando métodos estatísticos. In Guliato, D. and Vieira, T., editors, *Workshop of Works in Progress (WIP) in SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, pages 21–22, Ouro Preto.

Torres, L., Ramos, V., Nascimento, A. S., Alencar-Neto, J., and Frery, A. C. (2012b). Identificação de câmeras de celulares usando sensor pattern noise. In *Conferência Internacional de Ciências Forenses em Multimídia e Segurança Eletrônica (ICMedia)*, Brasília.

 ©2012 SBC — Soc. Bras. de Computação

# Índice de autores

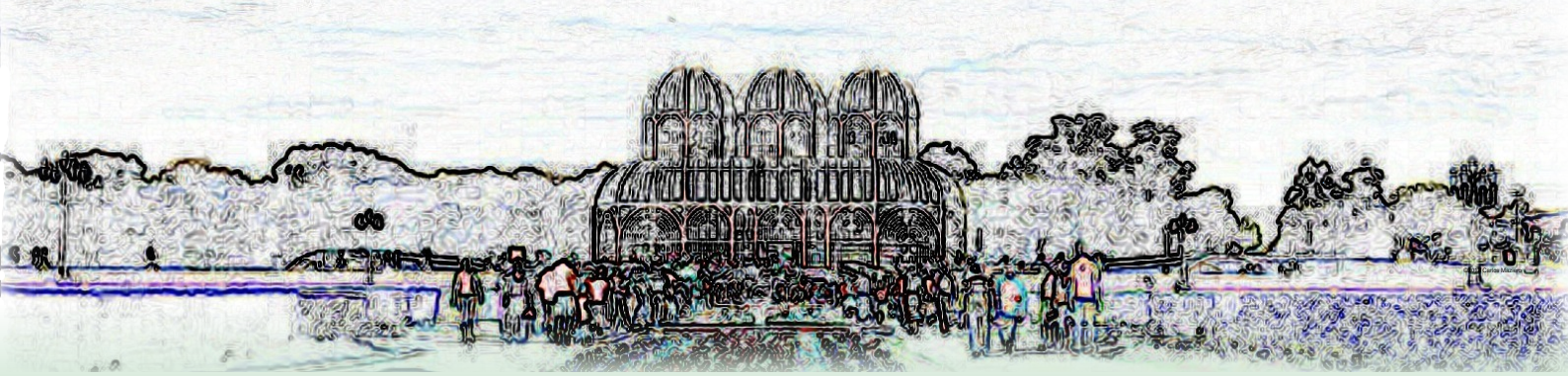

Abelém, Antônio ..... 324
Aguiar, Elisangela ..... 324
Alchieri, Eduardo ..... 182
Alencar-Neto, José ..... 576, 639
Almeida, Tiago ..... 2
Amaral, Aline ..... 598
Aranha, Diego ..... 346
Araújo, Arnaldo ..... 625
Araújo, Nelcileno ..... 310
Avila, Sandra ..... 625
Azulay, Diego ..... 511
Bachtold, Juliana ..... 263
Barcellos, Marinho ..... 154, 249
Bays, Leonardo ..... 235, 249
Becker, Gabriel ..... 461
Bereza Jr, André ..... 481
Bernardelli, Fernando ..... 331
Besrour, Hedi ..... 210
Bessani, Alysson ..... 182
Böger, Davi ..... 555
Bono, Tiago ..... 422
Bortolozzi, Flávio ..... 598
Buriol, Luciana ..... 154, 235, 249
Calverley, Marcus ..... 196
Câmara, Sérgio ..... 221
Cardoso, Sérgio ..... 303
Cardozo, Camila ..... 481
Carlos, Marcelo ..... 461, 625
Carvalho, Tiago ..... 584
Cerqueira, Eduardo ..... 324
Coelho, Olivia ..... 432
Costa, Filipe ..... 584
Cozzolino, Marcelo ..... 30
Cremonezi, Bruno ..... 452
Cunha, Francisco ..... 491
Cunha, Hugo ..... 511
Custódio, Ricardo ... 140, 461, 471, 481, 522
David, Bernardo ..... 126, 543
Dettoni Jr, Cláudio ..... 481
Deus, Flávio ..... 30
Farias, Fernando ..... 324
Feitosa, Eduardo ..... 362, 511
Felipe Jr, José ..... 413
Ferreira, Anselmo ..... 584
Ferreira, Ed' Wilson ..... 310
Ferreira, Rodrigo ..... 501
Follin, Mikkel ..... 196
Fraga, Joni ..... 182, 555
Freitas, Cinthia ..... 288, 598
de Freitas, Daniel ..... 86, 99
Frery, Alejandro ..... 639
Gaspary, Luciano ..... 154, 235, 249, 324
Gavinho Filho, Joffre ..... 58
Gielow, Fernando ..... 331
Gonçalves, Jesseildo ..... 338
Goya, Denise ..... 72, 354
van de Graaf, Jeroen ..... 126, 432
Guérios, Marlon ..... 555
Hernandez, Julio ..... 346
Hüttel, Hans ..... 196
Idalino, Thaís ..... 442
Imhof, Bruno ..... 471
Justino, Edson ..... 277
Karlsen, Henrik ..... 196
Macedo, Ricardo ..... 317
Machado, Raphael ..... 58, 221
Magri, Bernardo ..... 396
Mannes, Elisa ..... 388
Marcon, Daniel ..... 154, 235
Marcon Jr, Arlindo ..... 263
Martimiano, Luciana ..... 422
Martins, Gilbert ..... 30

Martins, Rafael ..... 44
Meira Jr, Wagner ..... 413
de Mello, Emerson ..... 555
Menegazzo, Cinara ..... 331
Mesquita, Daniel ..... 210
Mohsen, Machhout ..... 210
Monteiro, Fábio ..... 72
Nagahama, Fábio ..... 324
Nakahara Jr, Jorge ..... 86, 99
Nakamura, Dionathan ..... 112, 380
Nascimento, Alisson ..... 576, 639
Nascimento, Anderson ..... 126, 543
do Nascimento, Valtemir ..... 310
Neves, Miguel ..... 154, 235
Nogueira, Hendri ..... 522
Nogueira, Michele ..... 168, 317, 388
Okida, Cleber ..... 404
Oliveira, Luiz ..... 277
Oliveira, Rodrigo ..... 154, 235, 249
de Oliveira, Ruy ..... 310
Oliveira Jr, Walter ..... 277
Paraiso, Emerson ..... 288
Pari, Nadine ..... 331
Pena, Luís ..... 432
Pereira, Fernando ..... 44
Perin, Lucas ..... 461
Pinto, Guilherme ..... 452
Pinto, Adriana ..... 126
Pirmez, Luci ..... 58, 221
Pujolle, Guy ..... 317
Quadros, Gabriel ..... 44
Queiroz, Saulo ..... 168
Rached, Tourki ..... 210
Ramos, Valter ..... 576, 639
Rocha, Anderson ..... 584
Ruggiero, Wilson ..... 372
Rust, Luiz ..... 58, 221
Sadok, Djamel ..... 362
Salles, Ronaldo ..... 16, 303
Santin, Altair ..... 263, 288
Santin, Priscila ..... 288
dos Santos, Aldri ..... 331, 388
Santos, Douglas ..... 522
dos Santos, Eduardo ..... 471
dos Santos, Eulanda ..... 491
Schwartz, William ..... 611
Shinoda, Ailton ..... 310
Silva, Ewerton ..... 584
Silva, Francisco ..... 338
Silva, Nícolas ..... 16
Silva, Renato ..... 2
Silvério, Anderson ..... 461
Soto, Julio ..... 168
Souto, Eduardo ..... 30, 491
de Sousa Jr, Rafael ..... 543
de Souza, Fillipe ..... 625
Spagnuelo, Dayana ..... 442
Stihler, Maicon ..... 263
Teles, Ariel ..... 338
Terada, Routo … 72, 112, 354, 380, 396, 404
Thomsen, Bent ..... 196
Tonicelli, Rafael ..... 543
Torres, Jenny ..... 317
Torres, Leonardo ..... 576, 639
Ueda, Eduardo ..... 372
Valle, Eduardo ..... 625
Viegas, Eduardo ..... 263
Vigil, Martín ..... 140
Wangham, Michelle ..... 555
Welter, Guilherme ..... 481
Westphall, Carla ..... 501
Wognsen, Erik ..... 196
Yamakami, Akebo ..... 2

promoção

organização

apoio



Intel e seu logotipo são marcas registradas da
Intel Corporation nos EUA e/ou em outros países.

ISSN: 2176-0063