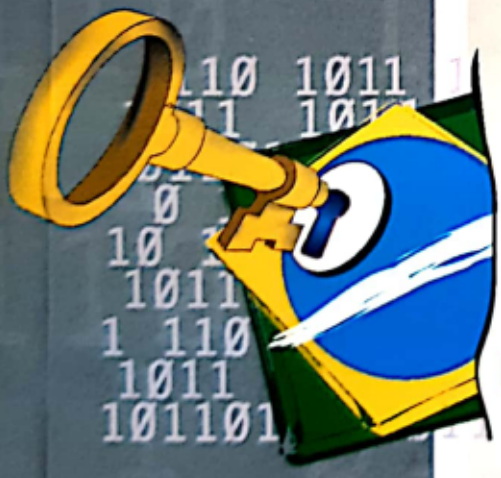


SBSEG'06

6º Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais

LIVRO TEXTO DOS MINICURSOS

28 de Agosto a 1 de Setembro de 2006
Mendes Convention Center
Santos - São Paulo - Brasil

Organizador
Lau Cheuk Lung

Minicursos do VI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2006)

Porto Alegre
Sociedade Brasileira de Computação – SBC
2022

Dados Internacionais de Catalogação na Publicação (CIP)

S612 Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (6. : 28 ago.-1 set. 2006 : Santos)
Minicursos do VI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2006) [recurso eletrônico] / organização: Lau Cheuk Lung. Dados eletrônicos. – Porto Alegre : Sociedade Brasileira de Computação, 2022.
256 f. ; PDF

Inclui bibliografia
ISBN 978-85-7669-501-1 (e-book)

1. Ciência da Computação. 2. Segurança da informação. 3. Sistemas computacionais. I. Lung, Lau Cheuk. II. Universidade Católica de Santos. III. Sociedade Brasileira de Computação. IV. Título.

CDU 004(063)

Ficha catalográfica elaborada por Jéssica Paola Macedo Müller – CRB-10/2662

Biblioteca Digital da SBC – SBC OpenLib

Índice para catálogo sistemático:

1. Ciência e tecnologia informáticas : Computação : Processamento de dados – Publicação de conferências, congressos, simpósios etc... 004(063)

Prefácio

Pela primeira vez no SBSeg, os minicursos são disponibilizados também na forma de livro texto, e é com muita satisfação que apresento este livro, contendo os textos de cada minicurso selecionado. Os textos selecionados abrangem tópicos diversos da área de segurança em sistemas computacionais: sistemas biométricos, técnicas de defesa contra spam, controle de acesso, rede ad-hoc e serviços web.

Este ano a chamada de minicurso contou com um número recorde de 16 propostas, o que demonstra um crescente interesse e amadurecimento deste evento. Gostaria de agradecer a todos os autores de propostas por esse recorde. Das submissões foram selecionados, para apresentação, 5 minicursos por ordem de classificação, definida pelo comitê de programa.

A qualidade das propostas foi bastante alta, o que dificultou na seleção das melhores. Todas as propostas tiveram três revisões feitas pelos membros do comitê de programa. Gostaria de agradecer aos membros deste comitê, constituído por 10 pesquisadores, pela competência e dedicação na avaliação destas propostas.

Finalizo esta mensagem agradecendo, primeiramente, aos professores Jorge Nakahara Junior (UNISANTOS) e Fabrício Alves Barbosa da Silva (UNISANTOS) pelo convite para coordenar todo o processo de seleção dos minicursos e pelo apoio na organização deste evento. Por fim, pelo apoio técnico dado pela equipe que mantém o sistema eletrônico de submissão e avaliação de artigos JEMS, coordenada pelo professor Lisandro Z. Granville (UFRGS).

Curitiba, Agosto de 2006
Prof. Lau Cheuk Lung, PUCPR
Coordenador dos minicursos SBSeg 2006

Promoção

Sociedade Brasileira de Computação - SBC

Presidente
Cláudia Maria Bauzer Medeiros (UNICAMP)

Vice-Presidente
José Carlos Maldonado (ICMC - USP)

Diretoria Administrativa e Finanças
Carla Maria Dal Sasso Freitas (UFRGS)

Diretoria de Eventos e Comissões Especiais
Karin Breitmann (PUC-Rio)

Diretoria de Educação
Edson Norberto Cáceres (UFMS)

Diretoria de Publicações
Marta Lima de Queirós Mattoso (UFRJ)

Diretoria de Planejamento e Programas Especiais
Virgílio Augusto Fernandes Almeida (UFMG)

Diretoria de Secretarias Regionais
Aline dos Santos Andrade (UFBA)

Diretoria de Divulgação e Marketing
Altigran Soares da Silva (UFAM)

Diretoria de Regulamentação da Profissão
Roberto da Silva Bigonha (UFMG)

Diretoria de Eventos Especiais
Carlos Eduardo Ferreira (USP)

Mandato 2005 - 2009
Ana Carolina Salgado (UFPE)
Ricardo de Oliveira Anido (UNICAMP)
Jaime Simão Sichman (USP)
Daniel Schwabe (PUC/RJ)
Marcelo Walter

Suplentes - Mandato 2005-2009
Robert Carlisle Burnett (PUCPR)
Ricardo Reis (UFRGS)
José Valdeni de Lima (UFRGS)
Raul Sidnei Wazlawick (UFSC)

Realização

Coordenação Geral
Jorge Nakahara Junior, UNISANTOS
Fabricio Alves Barbosa da Silva, UNISANTOS

Coordenação do Comitê de Programa
Paulo Licio de Geus, Unicamp
Paulo S. L. M. Barreto, USP

Coordenação de Minicursos
Lau Cheuk Lung, PUCPR

Coordenação de Workshops
Ricardo Dahab, Unicamp

Coordenação de Publicação

Comitê Consultivo do SBSeg
Carlos Alberto Maziero, PUCPR
Joni da Silva Fraga, UFSC
Lau Cheuk Lung, PUCPR
Luciano Paschoal Gaspary, UFRGS
Paulo Licio de Geus, Unicamp
Ricardo Dahab, Unicamp

Organização Local

Daniela Ushizima (UNISANTOS)
Denis Vidal de Jesus (UNISANTOS)
Hermes Senger (UNISANTOS)
Marta Rosatelli (UNISANTOS)
Maria Regina Ferreira (UNISANTOS)
Sérgio Guedes de Souza (NCE-UFRJ)

Comitê de Programa do Minicurso SBSeg 2006

Altair Olivo Santin, PUC-PR
Frank Augusto Siqueira, UFSC
Hao Chi Wong, Xerox Parc
Jorge Nakahara Junior, UNISANTOS
Lau Cheuk Lung, PUCPR (coordenador)
Luciano Paschoal Gaspary, UFRGS
Michelle Silva Wangham, UFSC
Miguel Pupo CorreiaUniversidade de Lisboa - Portugal
Ricardo Dahab, Unicamp
Thais Vasconcelos Batista, UFRN

Sumário

Capítulo 1

Segurança em Serviços *Web*

Emerson Ribeiro de Mello, Michelle S. Wangham,
Joni da Silva Fraga, Edson Camargo
Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
email:{emerson,wangham, fraga, camargo}@das.ufsc.br

Abstract

The use of open standards and integrative nature are features that made Web Services an interesting area to academic research and to industry. This chapter introduces the concepts behind the Service Oriented Architecture, Web Services, in particular. This text shows, through a use case, the benefits of this architecture and its security challenges. Afterwards, we present some research projects and technologies that deal with these security challenges.

Resumo

Devido a sua característica integradora e por fazer uso de padrões abertos, os Serviços Web se tornaram uma área de grande interesse para pesquisa e para a indústria. Neste capítulo, pretende-se introduzir ao leitor os conceitos da arquitetura orientada a serviços, e em particular, a sua mais atual caracterização, os Serviços Web. Será mostrado, através de um cenário de uso, os benefícios em utilizar tal tecnologia e também serão apresentados os desafios de segurança associados a esta. Por fim, são apresentados alguns trabalhos de pesquisa e tecnologias voltadas para tratar tais desafios de segurança.

1.1. Introdução

Há tempos que a Internet se consolidou como um importante veículo de comunicação e não demorou muito para se tornar uns dos principais meios para a realização de negócios. A Internet também é conhecida por agregar os mais diversos sistemas computacionais que variam desde a arquitetura de máquina, sistema operacional até os aplicativos finais aos usuários. O sucesso deste ambiente tão heterogêneo foi possível devido ao uso de protocolos padronizados, que garantem a interoperabilidade entre as aplicações, não importando em qual sistema operacional ou arquitetura de máquina esta esteja rodando.

A Internet surgiu diante de empresas que já faziam uso de seus sistemas computacionais e que, geralmente, não foram desenvolvidos para serem interoperáveis, por exemplo, com os sistemas computacionais de seus clientes, fornecedores, etc. Diante da necessidade da interação entre as aplicações distribuídas de diferentes organizações, uma nova caracterização de sistemas distribuídos surgiu possibilitando assim a troca de informações e a integração com os sistemas legados existentes - os **Serviços *Web***.

Os Serviços *Web* seguem uma Arquitetura Orientada a Serviços (AOS) e as principais características que os tornam uma tecnologia integradora e promissora são: (1) possuem um modelo fracamente acoplado e transparente que garante a interoperabilidade entre os serviços, sem que estes necessitem ter o conhecimento prévio de quais tecnologias estão presentes em cada lado da comunicação; (2) são auto-contidos e auto-descritivos; (3) usam padrões abertos como o HTTP e o XML, permitindo assim que aplicações sejam integradas através de linguagens e protocolos amplamente aceitos, e (4) tornam mais fácil a composição ou a combinação de diferentes provedores, visando formar serviços mais complexos e sofisticados.

Para a realização de negócios através da Internet, por onde circulam informações importantes para as corporações e, muitas vezes, sigilosas, garantir a segurança das informações é uma necessidade crítica. Com os Serviços *Web*, as aplicações tornam-se mais visíveis, expondo assim seus fluxos de negócio, processos e arquiteturas internas. Mecanismos de segurança estão sendo propostos para Serviços *Web*, porém, tais mecanismos ainda não contemplam todas as necessidades exigidas para segurança em Serviços *Web* e alguns são propostas iniciais que ainda não se consolidaram como um padrão de fato. Este cenário torna esta área um excelente ambiente para pesquisa.

O objetivo deste capítulo é analisar as questões de segurança provenientes da adoção de Serviços *Web*, discutir os principais padrões que visam minimizar as ameaças que esta tecnologia está suscetível, bem como apresentar algumas questões de pesquisa em aberto ou que estão sendo atualmente exploradas pelas instituições de pesquisa, que tratam da segurança em Serviços *Web*.

O presente capítulo está dividido em sete seções. Nesta primeira seção foi descrito o contexto geral em que o trabalho está inserido, destacando os objetivos do documento e a motivação para a escolha do tema. A seção 1.2 introduz os conceitos e as características da Arquitetura Orientada a Serviços e, em seguida, a seção 1.3 apresenta a arquitetura dos Serviços *Web*, com os padrões que formam a sua base. Os principais conceitos relacionados com segurança de informação, as questões de segurança em Serviços *Web* e as principais especificações que objetivam tratar parte destas questões são analisados na

seção 1.4. Na seção 1.5, as questões de segurança não tratadas nas especificações são discutidas e alguns trabalhos que visam tratar destas questões são analisados. As ferramentas que, atualmente, podem ser para o desenvolvimentos de Serviços *Web* seguros são apresentadas na seção 1.6. Por fim, a seção 1.7 apresenta a conclusão do capítulo, destacando os principais aspectos da segurança em Serviços Web que serviram e serve de motivação para pesquisas nesta área.

1.2. Arquitetura Orientada a Serviços

Segundo [Papazoglou 2003], a Arquitetura Orientada a Serviços (AOS) (*Service Oriented Architecture* – SOA) é uma caracterização de sistemas distribuídos, em que as funcionalidades do sistema são expostas via descrição de uma interface, permitindo a publicação, localização e a invocação por meio de um formato padronizado.

A AOS é constituída de relações entre três tipos de participantes: o *diretório para registro de serviços*, repositório que é utilizado para publicar e localizar as interfaces dos serviços; o *provedor de serviços*, entidade responsável por publicar as interfaces dos serviços providos por esta no registro de serviços e também responsável por atender as requisições originadas pelos clientes; e o *cliente*, aplicação ou um outro serviço que efetua requisições a um serviço. Cada participante da arquitetura pode ainda assumir um ou mais papéis, podendo ser por exemplo, um provedor e um cliente de serviços.

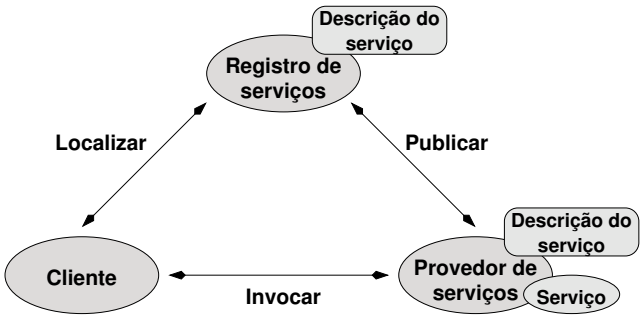


Figura 1.1. Interação entre entidades da AOS

Os participantes se relacionam através de três operações: *publicar*, *localizar* e *invocar*, como pode ser visto na Figura 1.1. Inicialmente, o provedor de serviço publica a interface do seu serviço junto ao diretório para registro de serviços. Desta forma, em algum momento posterior, o cliente pode efetuar uma busca por um determinado serviço (operação *localizar*), especificando as características desejadas, no diretório de registros. Se o serviço existir, a interface e a localização do respectivo serviço são retornados para o cliente. Por fim, o cliente efetua uma invocação ao provedor do serviço (operação *invocar*).

Os serviços estão baseados nas trocas de mensagens entre os provedores e os clientes, sendo assim, as mensagens seguem um formato padrão, garantindo aos serviços a neutralidade da tecnologia e permitindo que provedores e clientes utilizem diferentes implementações nas camadas inferiores. Os serviços também são definidos como *fracamente acoplados*, isso indica que possuem uma localização transparente e que não necessitam conhecer as estruturas internas presentes no lado do provedor e do cliente.

As interfaces dos serviços são auto-descritivas e baseadas em padrões abertos. Ou seja, a interface de um serviço define um conjunto de métodos públicos, juntamente com seus parâmetros, valores de retorno e meios para tratar possíveis exceções, porém não provê uma implementação. Com isto, pode-se assumir que a interface é um contrato entre o provedor do serviço e o cliente, sendo que o provedor deverá implementar todos os métodos ali descritos e o cliente só poderá invocar tais métodos.

Por estarem relacionados diretamente às funções de negócios, os serviços representam uma forma de modularidade diferente daquelas existentes nas linguagens de programação como os módulos, componentes e objetos. Componentes representam entidades e regras de negócio, já um serviço representa uma função de negócio completa, sendo composto por uma coleção de componentes. Serviços podem ser reutilizados e empregados em novas transações, na camada de negócios, dentro de uma organização ou através de organizações.

1.3. Arquitetura dos Serviços *Web*

Os Serviços *Web* (*Web Services* – WS) são classificados como um tipo específico de serviço, o qual é identificado através de um identificador uniforme de recursos (*Uniform Resource Identifier* – URI). Estes são independentes de linguagens de programação, de sistemas operacionais e das arquiteturas de máquinas. Através do uso de padrões abertos, como o XML e o HTTP, os Serviços *Web* conseguem garantir a interoperabilidade entre clientes e provedores de serviços, sem que os mesmos necessitem possuir o conhecimento prévio de quais tecnologias estão presentes em cada lado. Tal facilidade é ideal para que as relações de negócios entre empresas (*Business to Business* – B2B) sejam estabelecidas de maneira simples e dinâmica.

A definição para os Serviços *Web* dada em [W3C 2004a] é:

> *"Trata-se de uma aplicação identificada através de uma URI, que possui interfaces bem definidas e descritas em XML. As interações com outras aplicações se faz através de trocas de mensagens XML utilizando protocolos padrões da Internet."*

Um ponto importante a ressaltar é que os Serviços *Web* não são um outro tipo de objetos distribuídos, como aqueles presentes no CORBA[1], DCOM[2] e RMI[3] [OMG 2002, Brown e Kindel 1996, Sun 2002]. Em [Vogels 2003] é apresentada uma discussão sobre as semelhanças e diferenças entre Serviços *Web* e sistemas de objetos distribuídos. Para Vogels, os Serviços Web são um tipo de tecnologia de sistemas distribuídos que vem sendo utilizada em áreas em que as aplicações de objetos distribuídos falharam no passado.

As tecnologias de objetos distribuídos e de Serviços *Web* até possuem algumas características em comum, tais como: uma linguagem para descrição de interfaces (*Interface Definition Language* – IDL), que garante interações de rede bem definidas; e, mecanismos semelhantes para registro e localização de objetos ou serviços. Entretanto,

[1] *Common Object Request Broker Architecture*
[2] *Distributed Component Object Model*
[3] *Remote Method Invocation*

nos sistemas de objetos distribuídos, existe o conceito de *referência de objetos*, que não existe para os Serviços *Web*. A noção de *referência de objetos* é essencial dentro de um sistema de objetos distribuídos, visto que objetos, geralmente, possuem referências para outros objetos, possibilitando assim a computação com manutenção distribuída de estado . Todavia, a principal diferença entre os Serviços *Web* e os objetos distribuídos é o ciclo de vida dos mesmos. Um ciclo de vida de um objeto é composto pelas seguintes fases:

- diante de um pedido, uma fábrica cria uma instância de um objeto;
- o cliente que requisitou o pedido, executa operações no objeto instanciado;
- por fim, em algum momento posterior, o cliente remove a instância do objeto que não será mais utilizado.

Os Serviços *Web* não possuem um ciclo de vida com características como: objetos, referências e fábricas. Serviços *Web* não conseguem oferecer qualquer *facilidade para manter estado* na computação distribuída, característica básica de um sistema de objetos distribuídos. A arquitetura dos Serviços *Web* também não define relações entre as invocações realizadas em um mesmo serviço ou ainda em serviços relacionados, porém já estão sendo lançadas propostas para permitir tal interação, como a WS-Coordination [Cabrera et al. 2004].

Ambientes, como uma rede local, são caracterizados pela homogeneidade de plataforma e por possuírem um tempo de latência conhecido. Segundo [Vogels 2003], tal tipo de ambiente é ideal para a tecnologia de objetos distribuídos, visto que é uma tecnologia madura e, dentro de tal ambiente, bem robusta. Em ambientes como a Internet, em que a interoperabilidade e o suporte para plataformas e redes heterogêneas são essenciais, os Serviços *Web* demonstram ser os mais adequados.

A adoção dos Serviços *Web* não implica uso de qualquer aplicativo adicional no cliente ou no servidor. Para o cliente, basta uma linguagem de programação que dê suporte a XML e ao HTTP, por exemplo. Tal característica define os Serviços *Web* como *auto-contidos*. Serviços *Web* também são definidos como *auto-descritivos*, já que tanto o cliente como o servidor só precisam se preocupar com o formato e com o conteúdo das mensagens a serem trocadas, abstraindo assim os detalhes de implementação (fraco acoplamento). A arquitetura dos Serviços *Web* é composta basicamente por quatro elementos [Vogels 2003]:

- **serviço**: um aplicativo apto para processar documentos XML recebidos através de uma combinação de protocolos de transporte e de aplicação. Detalhes de como esse componente é construído, como técnicas de orientação a objetos, etc., não são especificados. O único requisito necessário para este tipo de componente, é que o mesmo esteja apto a tratar documentos XML;
- **endereço**: combinação entre protocolo e endereço de rede, utilizada para que um cliente possa acessar um serviço;
- **documento XML**: um documento que contém informações específicas à aplicação;

- **envelope**: encapsulamento que garante que documentos XML sejam processados de forma correta, separando as informações relacionadas a comunicação dos dados em si. Por exemplo, informações relacionadas a forma como a mensagem será cifrada ou assinada podem ser especificadas em um envelope sem que o documento XML original seja modificado.

Para tornar possível as três operações fundamentais de uma AOS - publicar, localizar e invocar - a arquitetura de Serviços *Web* adota as seguintes tecnologias baseadas em XML: a *Web Services Description Language* (WSDL)[W3C 2001], linguagem padrão usada para descrever as funcionalidades dos Serviços *Web*; o *Universal Description, Discovery and Integration* (UDDI) [OASIS 2004b], serviço padrão para publicação e localização de Serviços *Web*; e o SOAP [W3C 2003], protocolo usado para a invocação do serviço.

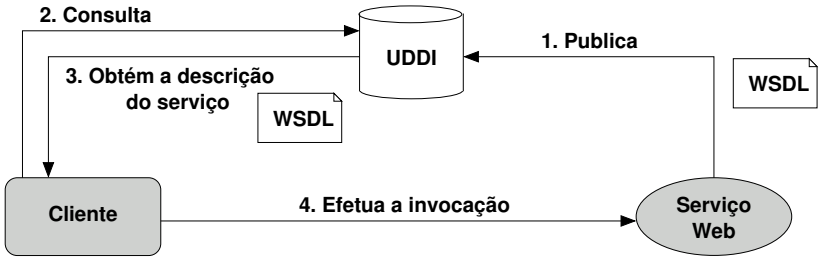


Figura 1.2. Colaboração típica na Arquitetura dos Serviços *Web*

A Figura 1.2 ilustra uma colaboração típica da arquitetura dos Serviços *Web*. O processo para tornar um Serviço *Web* publicamente disponível requer, inicialmente, que o provedor de serviços descreva a interface do serviço que deseja prover, utilizando a WSDL, e em seguida publique a interface em um serviço de busca público, como por exemplo o UDDI (passo 1 da Figura 1.2). A partir de então, o cliente pode localizar o serviço desejado e obter a sua WSDL (passos 2 e 3). A comunicação entre o provedor e o consumidor de um serviço é realizada através de trocas de mensagens XML, encapsuladas dentro de envelopes SOAP (passo 4).

A seguir, uma breve descrição das tecnologias empregadas na arquitetura dos Serviços *Web* é apresentada.

1.3.1. WSDL

A *Web Services Description Language* (WSDL) [W3C 2001] é uma gramática em XML, extensível, para especificar interfaces de Serviços *Web*. Um documento WSDL é independente de linguagem e de plataforma e tem por objetivo: (1) descrever quais são os serviços oferecidos; (2) mostrar como os clientes e provedores irão processar as requisições; e, (3) indicar em qual formato o serviço deve enviar as informações para um cliente.

Segundo [Weerawarana et al. 2005], um documento WSDL é composto por uma parte abstrata, que descreve o que o Serviço *Web* faz em termos de mensagem que este consome e produz, e por outra parte concreta que é referente a implementação e que define como e onde o serviços é oferecido. Os principais elementos XML presentes em um documento WSDL são:

- *types*: define os tipos de dados, utilizando o *XML Schema Definition* (XSD) ou ainda algum outro mecanismo para definição de tipos;
- *message*: define, de forma abstrata, as mensagens que serão trocadas;
- *operation*: define, de forma abstrata, a operação para uma mensagem;
- *portType*: descreve um conjunto abstrato de operações mapeadas para um ou mais serviços, os quais são descritos como pontos finais de rede ou portas;
- *binding*: especifica como mapear os elementos abstratos, *message* e *operation*, nos protocolos de rede que serão utilizados para transportar as mensagens até o destino (suas representações concretas);
- *port*: uma combinação entre o elemento *binding* e o endereço de rede, provendo assim um endereço único para acessar um serviço;
- *service*: declara o endereço das portas para os *bindings*. Ou seja, indica onde encontrar um serviço usando sua porta.

Os quatro primeiros elementos pertencem à parte abstrata da WSDL e os quatro últimos, à parte concreta. Cada um destes elementos pode ser descrito em diferentes documentos XML, e a combinação de todos estes formam uma descrição completa de um Serviço *Web*. A independência dos elementos principais traz uma grande flexibilidade para a disponibilização dos serviços. Uma mesma descrição de tipos de dados pode ser utilizada por diversos Serviços *Web* ou ainda múltiplos meios de transporte podem estar disponíveis para um serviço. A WSDL é uma linguagem flexível que também permite a inclusão de elementos não definidos pela especificação, possibilitando assim representar os atuais e os futuros formatos de mensagens.

1.3.2. UDDI

A especificação do *Universal Description, Discovery and Integration* (UDDI) [OASIS 2004b] define uma forma padronizada para publicação e descoberta de serviços dentro da Arquitetura Orientada a Serviço (AOS), que é parte fundamental na pilha de protocolos dos Serviços *Web*. A implementação de um servidor UDDI é composta por diversos Serviços *Web*, que provêem uma interface para que os clientes possam ter acesso as informações ali armazenadas. Os dados e meta-dados dos Serviços *Web* são armazenados em diretórios UDDI (*UDDI registry*), associando a cada estrutura de dados um identificador único, denominado *UDDI key*, criado de acordo com regras de classificação especificadas por cada organização. Tal classificação permite aos consumidores realizarem consultas mais refinadas, permitindo, por exemplo, buscar por provedores que forneçam determinado serviço dentro de uma localização geográfica específica.

Os diretórios UDDI não armazenam somente informações relativas a implementação de um Serviço *Web*, como a WSDL do serviço, estes também podem armazenar informações relacionadas diretamente a entidade que provê o serviço. O modelo de dados UDDI prevê os seguintes tipos de dados: *businessService*, descrições sobre a funções de negócio de um serviço; *businessEntity*, informações sobre a organização detentora do

serviço; *bindingTemplate*,informações técnicas do serviço, como por exemplo, endereço para invocação do mesmo; e, *tModel*, outros atributos, tais como taxonomia geográfica ou industrial. Nas especificações mais recentes do UDDI [OASIS 2002, OASIS 2004b], foram introduzidos dois novos tipos de dados voltados para a afiliação de registros, sendo estes: *publisherAssertion* e *subscription* [OASIS 2004a].

1.3.3. SOAP

Os Serviços *Web* adotaram diversas propostas para realizar trocas de mensagens entre clientes e provedores de serviços, mas foi o protocolo SOAP[4][W3C 2003] que surgiu como padrão de fato. Definido pelo consórcio W3C, o SOAP é um protocolo de comunicação baseado em XML para a troca de mensagens, independente de linguagem, que trabalha com diversos sistemas operacionais e sobre protocolos de aplicação já consolidados, como o HTTP, o SMTP, o FTP, o RMI/IIOP, etc.

O uso do SOAP sobre o protocolo HTTP se tornou comum nas atuais implementações de Serviços *Web* devido às facilidades providas pelo HTTP. Entre estas destacam-se: a infra-estrutura já existente dos servidores HTTP para disponibilizar os serviços e a facilidade em atravessar os limites de segurança impostos pelos *firewalls*, tendo em vista que o acesso à porta 80, utilizada por servidores HTTP, é geralmente liberada nestes mecanismos.

Uma mensagem SOAP é um documento XML que define o elemento ***envelope*** como sendo o elemento raiz do documento. O ***envelope*** SOAP contém as declarações dos espaços de nomes XML a serem utilizados, bem como as informações de codificação para a representação dos dados no documento e é composto pelos elementos ***header*** e ***body***. O ***header*** é um elemento opcional que contém informações, divididas em blocos, sobre como a mensagem deverá ser processada. Essas informações podem ser definições de roteamento, asserções de autenticação e autorização, entre outras. Já o elemento ***body*** é obrigatório e contém a mensagem em si. Qualquer tipo de informação que puder ser expressa em XML poderá fazer parte do corpo da mensagem. Dentro do elemento ***body*** pode estar contido o elemento ***fault***, que é usado para transportar informações sobre erros que possam vir a ocorrer no processamento das mensagens.

1.3.4. Uma Aplicação Exemplo

Um caso interessante para o uso dos Serviços *Web* é o de **portal de informações**. Um portal tem por objetivo agregar informações provenientes de diferentes origens em uma única e simples interface, se tornando um meio de fácil acesso para os usuários do sistema. Em [Wege 2002] são apresentadas algumas definições para portais, em que é possível destacar duas destas: os portais públicos e os portais corporativos. Os portais públicos são, geralmente, definidos por sítios que visam reunir informações de diferentes origens e aplicações, oferecendo uma interface padronizada e personalizável para seus usuários. Os portais corporativos também reúnem informações em uma interface padronizada, porém as informações ali reunidas só dizem respeito às necessidades da empresa em questão.

[4]Em sua criação, SOAP era um acrônimo para *Simple Object Access Protocol*, porém na versão 1.2 tal definição foi descartada pelo W3C, por achar que a mesma era equivocada. Assim, hoje SOAP é simplesmente o nome do protocolo e não mais um acrônimo.

Os portais de informações passaram por diversas evoluções desde o seu surgimento na década de 90, quando se resumiam em diretórios e máquinas de busca para catalogar sítios *Web*, até a mais atual versão que faz uso da tecnologia *Really Simple Syndication* (RSS) [RSS 2005]. O uso do RSS trouxe facilidades para provedores de informações e principalmente para os portais, pois apresenta uma forma padronizada e simples para disponibilizar resumos de informações. Tal tecnologia se constitui em um serviço de publicação e assinatura de notícias, porém não permite aos portais agregadores de notícias uma maior interatividade com seus usuários.

Os Serviços *Web* podem ser utilizados para a construção de um mesmo de tipo portal que hoje faz uso do RSS. Os provedores de serviços *web* disponibilizam uma interface de serviço padronizada para o fornecimento de informações para os portais e tais interfaces são publicadas em serviços como o UDDI. Desta forma, os clientes dos portais recorrem ao serviço UDDI para localizar as informações desejadas, assinando assim os respectivos serviços.Com os Serviços *Web* é possível obter um nível maior de interação entre todas as entidades participantes, seja um cliente interagindo com os provedores de serviços, ou seja estes últimos interagindo entre si.

O exemplo a ser apresentado neste capítulo consiste de um portal de informações voltado para o entretenimento. O objetivo do portal é reunir em uma única interface diversos provedores de serviços que tenham como área de atuação o entretenimento pessoal, como por exemplo, cinemas, parques de diversão, vídeo locadoras, teatros, etc. O portal também reunirá provedores de informações que não estão diretamente ligados ao entretenimento, mas que servem de base de apoio para a tomada de decisões dos usuários deste portal, como por exemplo, sítio de previsão do tempo, de resenhas de filmes, de companhias aéreas, de hotéis, de operadores de telefonia celular, etc.

Inicialmente, assim como a maioria dos serviços deste gênero, o portal exige um cadastro por parte de seus usuários, para que estes forneçam suas informações pessoais como nome, endereço, idade, sexo, etc. Após esta etapa, um usuário pode selecionar os provedores de serviços de sua preferência e assim configurar sua página pessoal no portal. Se for o caso, o usuário pode ainda indicar suas preferências para cada serviço selecionado. Por exemplo, no serviço de cinema, um usuário poderá informar os seus gêneros de filmes preferidos, o melhor dia da semana e horário, para ir ao cinema, etc. Em uma consulta, as informações retornadas podem ser ainda mais adequadas ao perfil do usuário se o serviço do cinema também souber a cidade e o bairro onde o usuário vive, sua faixa etária, podendo assim indicar ao usuário quais os cinemas mais próximos que estão exibindo os seus filmes prediletos.

Este cenário é totalmente possível de ser implementado quando se considera o ambiente dos Serviços *Web*. O serviço do cinema pode requisitar tais informações do usuário ao serviço de cadastro de usuários do portal, tendo em vista que ambos os serviços já possuem um acordo firmado para o compartilhamento de informação, acordo este que o usuário também possui ciência. A Figura 1.3 ilustra os relacionamentos entre os usuários, o portal e os provedores de serviços.

A listagem de filmes fornecidas pelo serviço do cinema já pode, por exemplo, vir acompanhada com a resenha de cada filme, informação esta obtida através da interação do serviço do cinema com o serviço de um sítio especializado em filmes. Enfim, com os

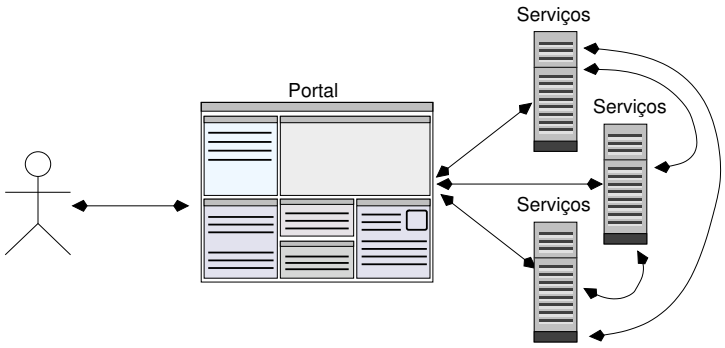


Figura 1.3. Portal de informações

Serviços *Web*, surge um novo tipo de interação, agora são aplicações se relacionando com aplicações sem a necessidade da intervenção dos usuários.

Nesta seção, preocupou-se em mostrar as facilidades que os Serviços *Web* podem trazer às aplicações de portais. Porém, sabe-se das inúmeras dificuldades e implicações associadas ao uso de Serviços *Web* neste domínio de aplicação. Este capítulo se resumirá em analisar os principais problemas relacionados à segurança computacional por trás desta aplicação de portal de informações.

1.4. Segurança em Serviços *Web*

Esta seção se inicia com uma breve introdução à segurança computacional, para que em seguida as questões e implicações de segurança relativas ao uso de Serviços Web possam ser adequadamente discutidas. Por fim, as principais especificações e trabalhos de segurança direcionados ao ambiente dos Serviços *Web* são descritos e analisados nesta seção.

1.4.1. Conceitos fundamentais de segurança

A segurança é vista como uma qualidade de serviço que garante o fornecimento do serviço, mesmo diante de ações de indivíduos não autorizados no sistema, sem que ocorram violações de segurança. Segundo [Russell e Gangeni 1991, Amoroso 1994], a segurança está fundamentada em quatro propriedades que devem ser garantidas:

- **confidencialidade**: a informação só deve ser revelada para usuários autorizados a acessá-la;
- **integridade**: a informação não poderá ser modificada, intencionalmente ou acidentalmente, por usuários que não possuam direito para tal;
- **disponibilidade**: o uso do sistema não poderá ser negado, de forma maliciosa, a usuários autorizados;
- **autenticidade**: o acesso ao sistema só deverá ser feito por usuários autênticos.

Algumas literaturas, como em [Landwehr 2001], também citam o ***não-repúdio*** como uma propriedade de segurança. O não-repúdio assegura que um usuário não poderá

negar a sua participação na ocorrência de um evento ou transação.

As violações de segurança ocorrem devido à exploração das vulnerabilidades existentes nos sistemas. As **vulnerabilidades** em sistemas computacionais sempre estiveram presentes. Um erro de programação, um erro de configuração ou mesmo um erro de operação, podem permitir que usuários não autorizados entrem no sistema ou mesmo que usuários autênticos executem ações não autorizadas, podendo assim comprometer o funcionamento correto do sistema [Bishop e Bailey 1996].

Uma **ameaça** consiste em uma possível ação que, se concretizada, poderá produzir efeitos indesejados ao sistema, comprometendo a confidencialidade, a integridade, a disponibilidade e/ou a autenticidade. Já o **ataque** é a concretização de uma **ameaça**, através da exploração de alguma **vulnerabilidade** do sistema, executado por algum intruso de forma maliciosa ou não. As quatro categorias de ataques, normalmente, identificados em sistemas distribuídos são:

- **interceptação**: uma parte não autorizada obtém acesso à informação (revelação não autorizada de informação);
- **interrupção**: o fluxo normal da mensagem é interrompida, impossibilitando que a informação chegue ao destino (negação de serviço);
- **modificação**: uma parte não autorizada modifica a informação recebida da origem e a transmite para o verdadeiro destino (modificação não autorizada da informação);
- **personificação**: entidade não autorizada transmite uma mensagem maliciosa pela rede, se passando por uma parte autêntica.

Em sistemas computacionais, as ameaças são constantes e uma maneira de evitar os ataques é identificar e corrigir as vulnerabilidades existentes nos sistemas, algo que não é tão simples quanto parece. Sistemas mais complexos tendem a possuir mais brechas de segurança, porém são nesses sistemas que a segurança é mais enfatizada, sabendo que o comprometimento desses sistemas gerariam enormes prejuízos financeiros.

A política de segurança de um sistema é um conjunto de diretrizes, normas e procedimentos, os quais estabelecem os limites de operação dos principais[5]. A política de segurança é concebida sob medida para um sistema específico, visto que cada sistema pode possuir diferentes necessidades. As diretrizes ditadas em uma política de segurança indicam o que cada componente do sistema (usuários, máquinas, etc) pode ou não pode fazer. As normas indicam o que cada componente está habilitado a fazer e como deverá ser feito.

As políticas de segurança de sistemas diferem em três ramos: a **segurança física**, objetiva proteger o meio físico em que opera o sistema (ex: imposição de restrições de acesso a determinadas áreas da empresas, medidas contra desastres, etc.); a **segurança**

[5]É universal que o termo **principal** identifique usuários, processos ou máquinas atuando em nome dos usuários de um sistema, que são considerados aptos pela política estabelecida (política de segurança lógica) em suas ações no sistema. Em contrapartida, usuários, processos e máquinas não autorizados pelas políticas são identificados como **intrusos**.

gerencial, que se ocupa com o ponto de vista organizacional, definindo processos para criação e manutenção das próprias políticas de segurança; e, a **segurança lógica**, que define quais usuários terão direitos de acesso ao sistema e quais os direitos que cada usuário possuirá.

1.4.2. Principais Questões de Segurança em Serviços *Web*

Os Serviços *Web* permitem que as aplicações se comuniquem sem a necessidade de qualquer tipo de interação com o usuário final. Voltando ao exemplo da Seção 1.3.4, o portal de entretenimento fornece aos seus usuários a opção para comprar ingressos para shows e ainda permite que o usuário utilize a mesma interface para comprar bilhetes aéreos e até mesmo para fazer a reserva em um hotel na cidade onde irá ocorrer o show. Neste caso, o usuário está interagindo diretamente com o portal e este, por sua vez, estaria interagindo com os demais sistemas, mediando assim a comunicação dos usuários com os sistemas da companhia aérea e do hotel. O problema aqui está em como garantir que as informações do usuário cheguem até o sistema da empresa aérea ou do hotel de forma segura, visto que as informações sensíveis do usuário, que só interessam a estes sistemas, estariam sendo roteadas e disponíveis pelo sistema do portal em si.

O roteamento entre múltiplos Serviços *Web* é comumente utilizado para obter escalabilidade e também para agir como uma ponte entre diferentes protocolos. Tecnologias como o TLS/SSL [Dierks e Allen 1999, Freier et al. 1996] permitem garantir a confidencialidade entre duas partes, porém não proporcionam segurança fim-a-fim, uma vez que a mensagem, para atingir o destinatário final, passa por diversos nós intermediários a nível de aplicação. Se a cifragem for empregada somente na camada de transporte, nós intermediários terão reveladas as informações que passam por eles, de forma proposital ou através das lacunas existentes entre uma sessão segura e outra.

As lacunas de segurança não ocorrem no transporte dos dados, mas sim quando os mesmos estão disponíveis nos nós intermediários. Assim, as informações confidenciais presentes nas mensagens SOAP, que deveriam permanecer confidenciais durante todo o percurso através dos nós SOAP intermediários, poderiam ficar expostas. Para tratar tal desafio, princípios de segurança devem ser aplicados em um contexto de segurança, que inclui muito mais que uma simples troca de mensagens SOAP. A Figura 1.4 ilustra diferentes contextos de segurança, em que o 1º contexto representa uma configuração ponto a ponto e o 2º uma configuração fim-a-fim.

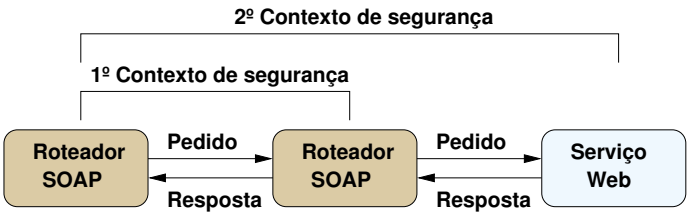


Figura 1.4. Contextos de segurança [IBM e Microsoft 2002]

Um outro desafio é como garantir os limites de segurança, antes determinados pelos *firewalls*. Os filtros de pacotes tradicionais se ocupam basicamente com a segurança na camada de rede, analisando se o pacote vem de uma origem confiável, porém não se

preocupa com o conteúdo dos pacotes. Assim, toda e qualquer requisição a um Serviço *Web* irá transpor o *firewall*. Os Serviços *Web* também estão suscetíveis a tipos de ataques já conhecidos como negação de serviço, mensagens antigas, estouro de pilha, entre outros, conforme apresentado nos trabalhos [Westbridge 2003, Demchenko et al. 2005]. Para garantir a segurança neste novo tipo de ambiente, novos mecanismos de segurança devem ser implantados também nas camadas superiores da pilha TCP/IP e devem operar em conjunto com os mecanismos presentes nas camadas inferiores (veja Figura 1.5).

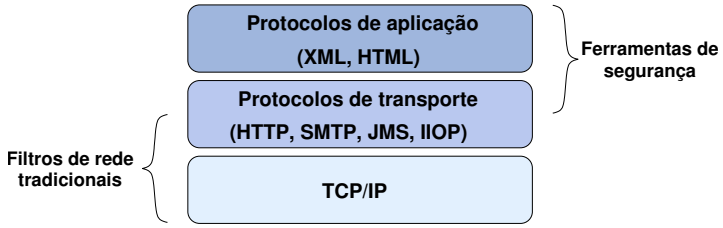


Figura 1.5. Segurança nas diferentes camadas

As interfaces dos Serviços *Web* são complexas e heterogêneas e é comum um Serviço *Web*, através de suas operações, acessar outros Serviços *Web*. A preocupação da negação de serviço que estava diretamente ligada ao sítio que hospeda o serviço, deve agora ser ampliada. Por exemplo, para uma instância de um Serviço *Web* é perfeitamente cabível atender 1000 requisições por segundo, porém essa instância pode fazer uso de serviço de terceiros, que para estes, o envio de mais de 10 requisições por segundo pode ser compreendido como um ataque de negação de serviço e assim interromper a comunicação.

Ao analisar o ambiente apresentado na aplicação exemplo da Seção 1.3.4, é possível notar que para a realização de um fluxo de negócio, a composição de diferentes Serviços *Web* se faz necessário. Quando se considera não mais um serviço único, mas sim uma orquestração ou coreografia de Serviços *Web*, constata-se que a segurança dos processos de negócios não foi ainda profundamente investigada e que ainda não existem soluções concretas e completas [Charfi e Mezini 2005].

Na aplicação exemplo, é dito que o usuário só precisa se cadastrar no portal, fornecendo suas informações pessoais, para que estas sejam propagadas por todos os serviços participantes. Como os limites administrativos precisam ser transpostos, as aplicações estarão sob diversos modelos administrativos e de segurança. Cada domínio transposto por um processo de negócio pode prover seu próprio conjunto de credenciais de segurança, tomando como base suas tecnologias subjacentes de segurança e suas políticas de segurança e de negócios. Por exemplo, em um determinado domínio, os usuários são autenticados através de um identificador único e senha; já em outro domínio parceiro, uma Infra-estrutura de Chave Pública (ICP) é usada para este fim. Para cada sistema, o cliente deverá possuir uma identidade e associada a esta diversos atributos de autorização.

Para o cliente o gerenciamento de tais informações pode se tornar muito custoso, visto que vai ter que gerenciar diferentes bases, fornecendo geralmente sempre as mesmas informações e ainda tendo que se preocupar em guardar os diferentes nomes de usuários e senhas. Já para os provedores de serviço, além da preocupação de ter que gerenciar inú-

meras identidades e credenciais, será necessário se preocupar também com a política de segurança que rege o sistema. Neste caso, a evolução das políticas é um fator que deve ser considerado, sabendo que uma política estática pode cobrir as necessidades de segurança de um determinado momento, porém pode já não conseguir suprir as necessidades que ainda estarão por vir. Assim, o provedor de serviço deverá considerar a necessidade da evolução de políticas, sabendo que tal ato não deverá desonrar as transações que estão em andamento, respeitando as políticas estabelecidas naquele momento.

Apesar de já existirem esforços para a definição de uma linguagem comum para expressar políticas, como a *WS-Policy* [WS-Policy 2004], ainda não existe nada padronizado para garantir a coesão destas políticas presentes em diferentes domínios. Em ambientes compostos por um número pequeno e conhecido de entidades, o gerenciamento das políticas de segurança não chega a ser um problema. Uma vez definida as políticas é possível que as mesmas continuem válidas por um longo período de tempo, visto que as entidades são conhecidas, bem como as tecnologias de segurança presentes no ambiente. Já os ambientes de larga escala, que são conhecidos pela sua dinâmica, apresentam o ingresso e egresso constante de entidades e ainda o uso de diferentes tecnologias de segurança. A gerência das políticas de segurança e a garantia de que as mesmas serão aplicadas são os grandes desafios em ambientes de larga escala.

Com o fluxo de negócios ultrapassando diversos domínios administrativos, a privacidade dos usuários também é um assunto que merece atenção. Em um cenário ideal, os usuários poderiam exercer o direito de determinar como suas informações serão manipuladas, indicando quais informações podem ser compartilhadas com terceiros, como esse compartilhamento deve ser feito e também indicando o período de tempo que essas informações podem ficar disponíveis nos sistemas. O projeto *Shibboleth* [Shibboleth 2005] apresenta uma preocupação com a privacidade das informações dos usuários, definindo como requisitos da arquitetura meios para gerenciar quais informações um sítio origem irá transferir para um sítio destino, com o consentimento do usuário.

Por fim, um dos pilares mais importantes para a construção de aplicações distribuídas e de processos de negócios é a confiança entre as entidades participantes. O termo confiança pode assumir diversos sentidos em uma aplicação distribuída. Em segurança, o mais usual é como garantir que as informações foram enviadas por uma origem confiável. No caso, a preocupação geralmente restringe-se a garantir as propriedades de autenticidade e integridade das mensagens. Para tratar tal problema diversos modelos foram propostos, como por exemplo, o X.509 [Housley et al. 2002], o PGP [Zimmerman 1994] e o SPKI/SDSI [Ellison et al. 1999, Rivest e Lampson 1996]. Porém, a confiança não se restringe simplesmente em garantir as propriedades de autenticidade e integridade. Em uma aplicação distribuída, as informações trocadas entre clientes e provedores de serviços possuem um certo valor e a manipulação indevida das mesmas pode acarretar em prejuízos para ambos os lados. Por exemplo, um cliente não gostaria de fornecer o número do cartão de crédito para qualquer provedor de serviço. A confiança entre clientes e provedores de serviço é algo que pode ser estabelecido com base, por exemplo, em uma base de reputações, o que poderia indicar que um determinado provedor de serviços sempre honrou suas comunicações.

Em alguns trabalhos, assume-se que o estabelecimento de confiança é um processo

manual que exige o cumprimento de diversos requisitos burocráticos antes da criação da relação de confiança. Por exemplo, para entrar na hierarquia das autoridades certificadoras do X.509 é necessário cumprir um conjunto de requisitos, sendo alguns destes relacionados a segurança física do local onde estará armazenada a chave privada da Autoridade Certificadora (AC). Outros trabalhos tratam a confiança de uma maneira mais dinâmica e volátil. Por exemplo, para um determinado fluxo de negócios é necessário que diversos provedores de serviço se agrupem e, uma vez que o fluxo tenha sido cumprido, tal relação é desfeita.

1.4.3. Especificações de Segurança para Serviços *Web*

Com o objetivo de tornar seguro o uso dos Serviços *Web* e assim garantir a sua ampla adoção, muitas propostas de segurança estão sendo submetidas a órgãos como: *World Wide Web Consortium* (W3C)[6] , *Organization for the Advancement of Structured Information Standards* (OASIS)[7] e *Web Services Interoperability Organization* (WS-I)[8]. As propostas visam cobrir diversas áreas de segurança e, em conjunto com as especificações de segurança para o padrão XML, estas permitem garantir alguns dos requisitos de segurança apontados na seção anterior.

XML Signature

O uso de assinaturas de digitais é uma forma para garantir as propriedades de integridade e autenticidade de informações digitais. A especificação *XML Signature* (XMLDSign) [Bartel et al. 2002], proposta conjunta entre W3C e IETF, define regras para gerar e validar assinaturas digitais expressas em XML. A XMLDSign possui pontos em comum com o *Public Key Cryptography Standard #7* (PKCS#7), porém apresenta formas para tratar os novos desafios em se trabalhar com documentos XML.

O desafio em criar assinaturas expressas em XML está justamente na forma de codificação dos documentos XML. Por exemplo, para interpretadores XML, o elemento `<Nome >` e o elemento `<Nome>` são tratados da mesma forma. Porém, quando aplicado um algoritmo para assinatura digital, duas assinaturas distintas seriam geradas.

A *XML Canonical* [Boyer 2001] define meios para representar documentos XML na forma canônica. Documentos XML, que sejam sintaticamente diferentes, porém logicamente equivalentes, serão representados por uma mesma forma canônica. Assim, o uso da forma canônica possibilita que os documentos XML possam ser assinados sem que haja preocupação com a sintaxe dos mesmos.

O uso da XMLDSign não está unicamente voltado para assinar documentos XML. É possível assinar qualquer tipo de documento eletrônico (arquivos binários ou textos), sendo que a assinatura será representada através de um documento XML. Também é possível assinar somente algumas partes de um documento XML, permitindo assim que outras partes de um documento XML sofram modificações, sem que isso invalide a parte assinada. A XMLDSign não define novos algoritmos criptográficos, mas faz uso dos

[6]http://www.w3.org
[7]http://www.oasis-open.org
[8]http://www.ws-i.org

algoritmos existentes, como o RSA [RSA 2002] e SHA-1 [Eastlake e Jones 2001]. As assinaturas podem ser representadas em três diferentes formas (veja Figura 1.6):

- ***enveloped***: a assinatura fica contida dentro do próprio documento XML a qual esta referencia. É ideal para ser utilizada com Serviços *Web*, inserida em mensagens SOAP;
- ***enveloping***: os dados assinados, em XML ou não ficam contidos dentro da própria estrutura do XMLDSig;
- ***detached signature***: a assinatura fica separada dos dados assinados. Isto é ideal para assinar documentos que não estão disponíveis localmente ou que sofrem constantes modificações.

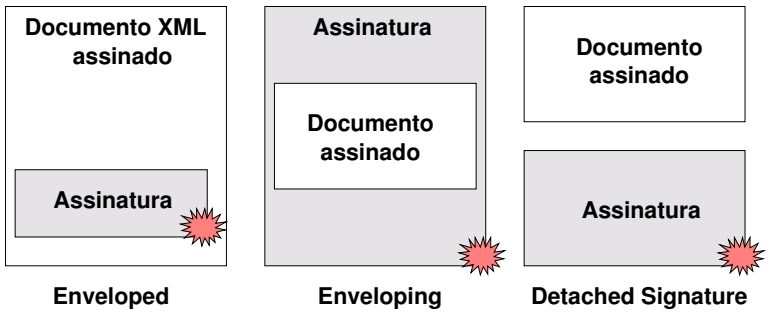


Figura 1.6. Formas de assinaturas XMLDSign

XML Encryption

A *XML Encryption* (XMLEnc) [Imamura et al. 2002] visa prover segurança fim-a-fim para aplicações que necessitem realizar troca de dados de forma segura. Diferentemente dos protocolos TLS/SSL [Dierks e Allen 1999, Freier et al. 1996], que só garantem a confidencialidade dos dados durante a sessão estabelecida entre duas partes, a XMLEnc garante confidencialidade persistente, garantindo assim a confidencialidade dos dados mesmo depois do término da sessão.

A XMLEnc provê soluções para algumas necessidades não cobertas pelo TLS/SSL, como a possibilidade de cifrar somente partes de um dado e o estabelecimento de sessões seguras entre mais de duas partes. Os dados cifrados são representados de uma forma estruturada e permitem que em um mesmo documento estejam presentes informações cifradas e não cifradas. Tal estrutura ainda possibilita o uso de diferentes chaves para cifrar partes de um documento, permitindo assim que um mesmo documento seja trocado entre diversas partes, sem que ocorra a revelação de informação para partes não autorizadas e garantam o acesso à informação, por partes autorizadas.

De forma análoga ao XMLDsig, o XMLEnc representa, de forma estruturada, dados cifrados e permite cifrar documentos XML ou não. A estrutura do XMLEnc, além de expressar os dados cifrados, também expressa detalhes sobre o tipo do documento

cifrado (jpeg, xml, etc.): a chave simétrica que será utilizada na sessão; informações sobre o tipo da chave simétrica; e o método de cifragem utilizado (ex: RSA para cifrar a chave secreta e AES [Daemen e Rijmen 2002] para cifrar os dados).

XACML

A autorização é uma propriedade básica de segurança que determina se um principal pode ou não executar alguma ação sobre algum recurso. Geralmente, cada sistema utiliza uma linguagem própria para definição das políticas, tornando assim um fator limitante para a concepção de sistemas distribuídos e abertos. Visando garantir a interoperabilidade entre os diversos sistemas, o órgão OASIS lançou a *eXtensible Access Control Markup Language* (XACML) [OASIS 2005a], um sistema de políticas de propósito geral, baseado em XML.

A XACML descreve uma linguagem para políticas de controle de acesso e também um formato para mensagens de *pedido* e *resposta*. A linguagem para política de controle de acesso é utilizada para definir quem possui direitos de acesso sobre o quê. O formato de *pedido* e *resposta* descreve como as consultas sobre o sistema de políticas deverão ser realizadas (pedido) e como deverão ser as respostas.

O formato de *pedido* e *resposta* define as trocas ente o *Policy Decision Point* (PDP) [Yavatkar et al. 2000], ponto este que efetua o processamento da política, e o *Policy Enforcement Point* (PEP) [Yavatkar et al. 2000], ponto este que concretiza as decisões de política. A XACML foi desenvolvida para garantir a interoperabilidade entre diversas aplicações. Assim, uma camada de abstração entre o ambiente da aplicação e a linguagem núcleo do XACML é feita através de um Contexto XACML. Um Contexto XACML é definido através de um esquema XML, que descreve uma representação canônica das entradas e saídas do PDP [OASIS 2005a].

Um pedido é composto: (1) por atributos associados ao sujeito que está originando a requisição; (2) pela identificação do recurso desejado; (3) pelas ações que serão executadas no recurso; e também (4) pelos atributos do ambiente. Já na resposta são contidas decisões como: *permit* – para acesso garantido; *deny* – para acesso negado; *not applicable* – para a inexistência de política ou de regras associadas ao recurso; ou ainda *indeterminate* – para a ocorrência de erros durante o processamento [Lorch et al. 2003].

A Figura 1.7 ilustra o fluxo de dados entre um cliente tentando acessar um recurso, utilizando-se do XACML. No passo 1 o sujeito (cliente) lança um pedido ao PEP, que monta um pedido XACML e encaminha ao Tratador de contexto (passo 2). O tratador de contexto encaminha o pedido para o PDP para que o mesmo decida sobre a tentativa de acesso (passo 3). O PDP pode requisitar ao Tratador de contexto atributos relacionados ao recurso e ao sujeito (passos 4, 5 e 6). De posse dos atributos, o PDP requisita as políticas associadas com as entidades envolvidas (passo 7) e assim gera uma resposta sobre a decisão tomada (passo 8). O Tratador de contexto gera uma resposta XACML e envia ao PEP (passo 9). E por fim, o PEP garante ou não o acesso ao recurso (passo 10).

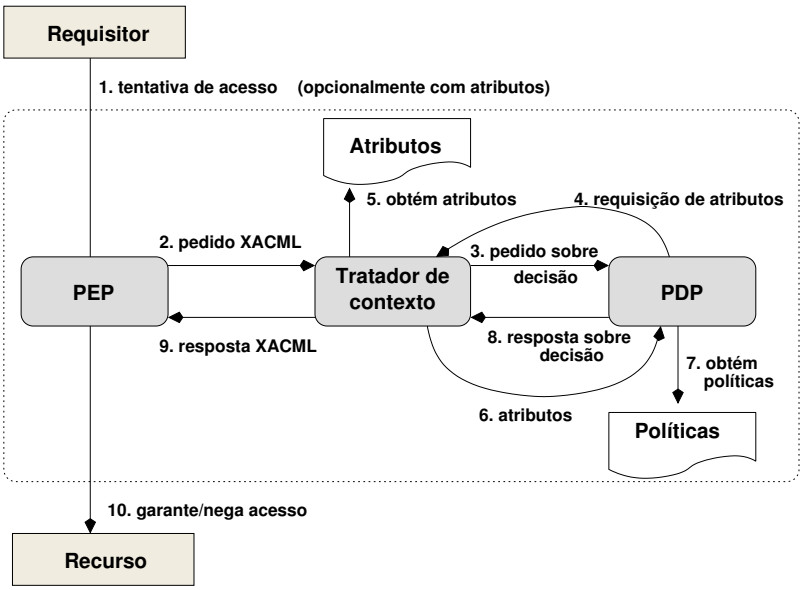


Figura 1.7. Fluxo de dados com o XACML [OASIS 2005a]

SAML

A *Security Assertion Markup Language* (SAML) [OASIS 2005c] consiste de um conjunto de especificações e esquemas XML, que juntos definem uma forma padrão para criar, trocar e interpretar asserções de segurança entre entidades de uma aplicação distribuída. No caso, são definidos meios para expressar, em XML, informações sobre autenticação, autorização e atributos de um sujeitos, porém as especificações da SAML não definem uma nova tecnologia ou forma para autenticação, mas sim uma tecnologia que visa garantir a interoperabilidade entre os diferentes sistemas de autenticação[9].

Uma *asserção de segurança* é um conjunto de afirmações, concedidas por um emissor SAML, sobre determinadas informações de um principal. Na especificações são definidas três tipos de asserções: de **autenticação**, fornecida pelo emissor SAML após o ato de autenticação com sucesso do usuário, que contém informações relacionadas ao emissor, o principal autenticado, o período de validade, etc.; de **atributo**, que contém detalhes específicos sobre o principal em questão, por exemplo, um papel que o principal desempenha dentro do sistema; e de **autorização**, que indica os direitos que um principal possui sobre um determinado recurso, sendo que esta asserção pode levar como base as asserções de autenticação e de atributos. Apesar do modelo de uso do SAML prever o uso de autoridades responsáveis pela emissão dessas asserções, as especificações não fazem qualquer menção sobre as mesmas. Todavia, as especificações definem os protocolos para que se possa interagir com essas autoridades.

Em sua primeira versão, o principal objetivo do SAML era permitir a transferência de autenticação e autorização entre aplicações *Web* (confiança portável). Já a versão

[9]A especificação da SAML prevê o uso de diferentes mecanismos para a autenticação: usuário e senha, Kerberos [Kohl e Neuman 1993], *Secure Remote Password* [Wu 1998], certificados TLS/SSL, chave pública (X.509 [Housley et al. 2002], SPKI [Ellison et al. 1999], XKMS [Hallam-Baker e Mysore 2005]), XMLDSign e ainda, possibilita o uso de mecanismos não definidos na especificação.

1.1 foi lançada com o intuito de melhorar a interoperabilidade e garantir uma melhor integração com o XMLDSign. Por fim, com base nas iniciativas do projetos *Liberty Alliance* (ver seção 1.5.1) e Internet2 Shibboleth [Carmody 2001], a versão 2.0 da SAML, recentemente lançada, tem como foco principal o uso de identidades federadas e ainda apresentando as seguintes características [OASIS 2005b]:

- **pseudônimos**: pseudônimos, ou identificadores opacos, permitem que principais interajam com o sistema sem a necessidade de revelar qualquer informação que o identifique, como e-mail, nome, etc. O uso de pseudônimos impede que provedores entrem em comum acordo para cruzar informações de um determinado principal e assim ferir sua privacidade;

- **gerenciamento de identificadores**: define como dois provedores poderão estabelecer e, em conseqüência, gerenciar os pseudônimos dos principais, com quem operam;

- **metadados**: estes definem como expressar dados de configuração e dados de confiança, para tornar mais simples o uso do padrão SAML, visto que as entidades participantes devem aceitar os mesmos papéis, identificadores, perfis, URL e certificados;

- **cifragem**: possibilita que atributos, identificadores ou toda a asserção seja cifrada. Tal característica permite garantir a confidencialidade fim-a-fim;

- **perfis de atributo**: estes simplificam a configuração e a implantação de sistemas que trocam dados de atributos. Definem como os atributos poderão ser transportados nas asserções SAML. Definem um perfil básico, que utiliza os tipos primitivos do XML para expressar os atributos e também define perfis como X.500/LDAP, UUID[10] e XACML;

- **manutenção da sessão**: o SAML 2.0 provê um protocolo que permite que todas as sessões, providas por uma autoridade de sessão, possam ser facilmente encerradas simultaneamente;

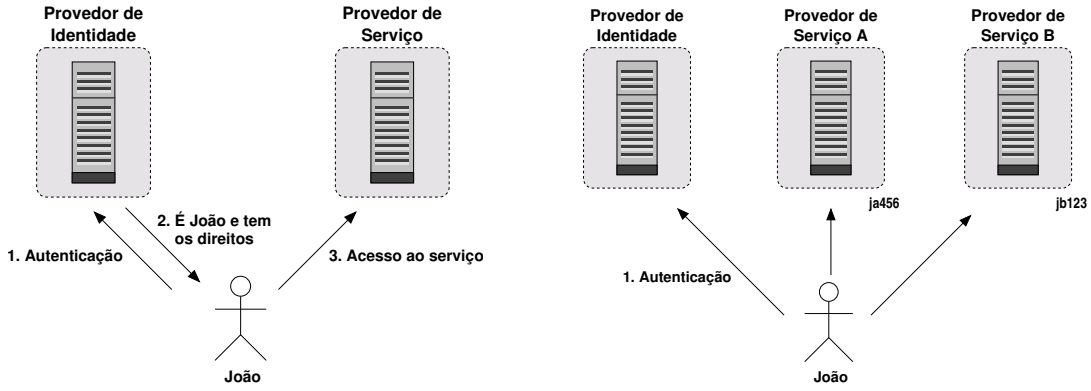
- **suporte a dispositivos móveis**: trata com as restrições de processamento dos dispositivos e com a largura de banda;

- **mecanismos de privacidade**:permitem expressar as configurações e políticas de privacidade dos provedores e principais, com relação ao uso da informação;

- **descoberta do provedor de identidade**: permite uma forma para localizar provedores de identidades, em ambientes em que existam mais de um provedor de identidade;

[10]Identificador único universal, definido pela *Open Software Foundation* (OSF) como parte do *Distributed Computing Environment* (DCE), uma vez criado por alguém, tem-se a garantia que o mesmo não será reutilizado por mais ninguém [OpenGroup 1997].

Nas versões 1.0 e 1.1 da SAML, o principal objetivo era transpor domínios através do uso da autenticação única (*Single Sign-On* – SSO), possibilitando que usuários autenticados em um domínio de segurança pudessem usufruir dessa autenticação em serviços presentes em outros domínios, sendo isto transparente para o usuário. Para isto, o conceito de identidade federada é utilizado. Neste caso, as entidades *Provedor de Identidades* e *Provedor de Serviços* entram em um acordo sobre os atributos dos usuários, como por exemplo, o nome do usuário e atributos de sessão, cabendo ao Provedor de Identidades garantir a autenticidade dos mesmos ao Provedor de Serviços. A Figura 1.8(a) ilustra um caso de identidade federada.



(a) Atributos federados (b) Ligação entre contas

Figura 1.8. Identidade federada

Com a SAML 2.0, surgiu uma nova forma de uso de identidade federada, que permite a "*ligação entre contas*". Neste caso, as diferentes identidades de um usuário, presentes em diferentes Provedores de Serviço, podem ser associadas de forma que possibilite o SSO, porém sem ferir a privacidade do usuário. A SAML 2.0 propõe o uso de pseudônimos, o que evita que Provedores de Serviços entrem em acordo, visando rastrear as informações de um determinado usuário.

No caso apresentado na Figura 1.8(b), o Provedor de Identidade estabeleceu diferentes pseudônimos com os Provedores de Serviços A e B, para referenciar um mesmo usuário, no caso João. Assim, João ao apresentar a asserção ao Provedor A, é reconhecido como o usuário local *ja456*; e ao apresentar a asserção ao Provedor B, é reconhecido como o usuário local *jb123*. Dessa forma, os provedores A e B não terão meios para rastrear o usuário João.

XKMS

Desenvolvida inicialmente pela VeriSign, em conjunto com a Microsoft e WebMethods, o padrão *XML Key Management Specification* (XKMS) [Hallam-Baker e Mysore 2005] é uma especificação aberta que define interfaces, baseadas em Serviços *Web*, visando retirar dos desenvolvedores de aplicações a complexidade em se trabalhar com Infra-estrutura de Chave Pública (ICP), podendo esta ser X.509, SPKI ou mesmo PGP [Zimmerman 1994]. A especificação é dividida em duas sub-especificações, *XML Key Information Service*

Specification (XKISS) e *XML Key Registration Service Specification* (XKRSS), que juntas definem meios para gerar pares de chaves, armazenar e localizar informações sobre chaves públicas, bem como para validar assinaturas.

A especificação XKISS define os serviços que visam retirar das aplicações a complexidade em se trabalhar com assinaturas expressas em XMLDSign [Bartel et al. 2002]. Informações como nome da chave, ou um certificado X.509, ou a própria chave, são descritas dentro do elemento XML `<ds:KeyInfo>` de uma assinatura em XMLDSign. Porém, a informação fornecida juntamente com a assinatura pode ser insuficiente para que o receptor possa validar a mesma, ou ainda, a informação pode estar em um formato no qual o receptor não é capaz de compreender. Por exemplo, dentro do elemento `<ds:KeyInfo>` só é fornecido um nome para a chave utilizada, mas não a própria chave em si. O XKISS define dois serviços: um para permitir a localização de informações relacionadas às chaves (*XKISS Locate*) e outro para verificar se estas informações relacionadas às chaves são válidas (*XKISS Validate*).

O objetivo do serviço *XKISS Locate* é de somente localizar informações relacionadas ao elemento `<ds:KeyInfo>`. Tais informações podem ser obtidas em uma base local de dados ou através do encaminhamento de um pedido a outros servidores. Por exemplo, dentro de um elemento `<ds:KeyInfo>` poderia estar contido somente o e-mail do criador da assinatura. Com essa informação, o *XKISS Locate* poderia localizar qual chave está associada com o e-mail e assim permitir que a assinatura seja validada. Porém as informações retornadas pelo *XKISS Locate* não são validadas, sendo tal tarefa atribuída ao serviço *XKISS Validate*. O *XKISS Validate* possibilita realizar as mesmas funções do *XKISS Locate*, porém o cliente pode obter uma asserção garantindo a validação das informações por este retornadas.

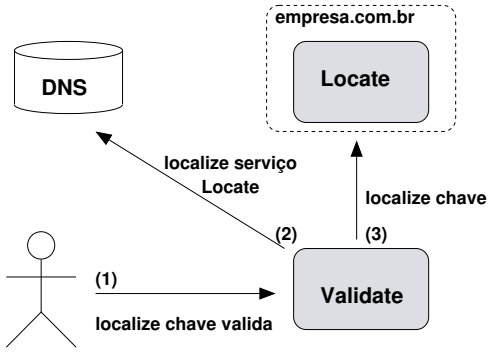


Figura 1.9. Uso combinado dos serviços *Locate* e *Validate* [Hallam-Baker e Mysore 2005]

A Figura 1.9 ilustra um exemplo, em que os serviços *XKISS Locate* e *Validate* são combinados com o intuito de localizar e validar uma assinatura. Neste exemplo, o usuário João aciona o seu serviço *XKISS Validate* para encontrar, de forma confiável, a chave pública do usuário `maria@empresa.com.br` (passo 1). No caso, o *XKISS Validate* utiliza o serviço de nomes (DNS) para localizar o serviço *XKISS Locate* responsável pelo domínio `empresa.com.br`[11] (passo 2). Por fim, o serviço *XKISS Validate* aciona o

[11] A especificação 2.0 do XKMS [Hallam-Baker e Mysore 2005] define meios para incluir informações do XKISS nos registros do DNS.

serviço *XKISS Locate* do domínio `empresa.com.br` para obter a chave pública do usuário *Maria* (passo 3).

Como visto, o objetivo dos serviços propostos na especificação XKISS é de localizar e validar as informações associadas com as chaves públicas, sendo que o registro e o gerenciamento destas informações estão dentro do contexto das facilidades providas pela especificação XKRSS. Tal especificação define serviços para: (1) o registro de informações; (2) a reemissão das informações associadas a chaves, permitindo gerar novas credenciais na ICP subjacente, por exemplo, no caso de um certificado expirar; (3) a revogação das informações associadas; e, (4) para a recuperação de uma chave privada, associada anteriormente. Neste último caso, só é possível recuperar a chave privada, somente se o par de chaves em questão foi gerado anteriormente pelo próprio XKRSS.

Cada protocolo definido pela XKMS provê suporte a diversas opções, incluindo opções de processamento das mensagens trocadas. Cabe ao cliente especificar quais opções este está apto a tratar e assim o serviço XKMS poderá decidir se aceita ou não o pedido, sendo que tal decisão dependerá de sua própria política de negócio. As opções para o processamento das mensagens podem ser: síncrona – o serviço responde ao pedido assim que o processamento for finalizado; assíncrona – o serviço pode não conseguir responder ao pedido imediatamente, mas notifica o cliente que o pedido ainda não foi satisfeito, posteriormente, o cliente poderá novamente invocar o serviço com o objetivo de obter a resposta final; pedidos de duas fases – diferente do assíncrono, nesta opção não há atraso entre o pedido inicial e o envio da resposta final. Tal forma de processamento é, basicamente, utilizada como um tipo de proteção contra ataques de negação de serviço.

A Figura 1.10 ilustra as opções de processamento previstas para o XKMS. No caso do processamento síncrono, o serviço, ao receber o pedido *P*, executa a operação requisitada e já envia uma resposta, composta pelo resultado e por um código (*Final*), o qual indica que a transação foi encerrada com sucesso.

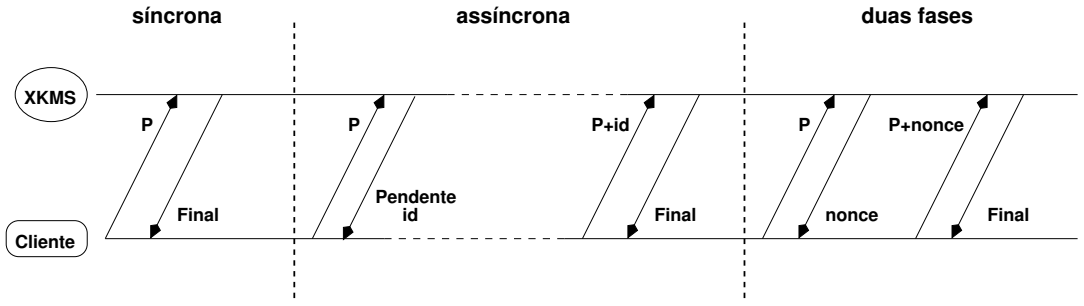


Figura 1.10. Tipos de processamento os pedidos XKMS

No processamento assíncrono, o cliente envia um pedido inicial *P* ao serviço para registrar uma chave, por exemplo. Segundo a política do serviço em questão, um pedido de registro de chaves requer a interação com o administrador do serviço, assim o pedido fica pendente até que o administrador aprove ou não o registro. Neste caso, o serviço responde o pedido inicial, feito pelo cliente, indicando que o processamento está pendente e fornece também um identificador para esta pendência (*id*). O cliente com este identificador em mãos pode, em um próximo pedido, verificar se o pedido inicial *P* já foi

processado e em caso afirmativo o serviço responde ao pedido *P* juntamente com o código *Final* para indicar o término da transação.

A política de segurança do serviço XKMS pode determinar que o serviço só aceitará requisições de clientes autenticados, estes já presentes na lista de controle de acesso do serviço, por exemplo. Dessa forma, todos os pedidos que chegam ao serviço devem ser assinados e cabe ao serviço verificar tais assinaturas. Sabe-se que as verificações de assinaturas digitais possuem um certo custo de processamento e intrusos poderiam inundar o serviço XKMS com requisições falsas, com o objetivo de provocar uma negação de serviço. O processamento de duas fases visa coibir tal tipo de ataque. Para isso, o serviço ao receber um pedido inicial *P* de um cliente (ver Figura 1.10) responde para este enviando um *nonce*[12] e indicando que o cliente deverá executar um novo pedido, juntamente com este *nonce*. O objetivo do *nonce* é garantir uma autenticação fraca, ou seja, o serviço só irá realmente verificar a assinatura de pedidos que estejam acompanhados de *nonces*, emitidos por ele. Assim, evita-se processar assinaturas de pedidos falsos. Pedidos, que forem novamente encaminhados com o *nonce*, serão processados e serão respondidos juntamente com o código *Final* para indicar o término da transação.

WS-Security

Proposta apresentada inicialmente pela IBM e Microsoft, a *WS-Security* [OASIS 2004c] é hoje uma especificação padronizada pela OASIS que tem como objetivo a proposição de extensões ao SOAP para permitir a construção de Serviços *Web* seguros. A especificação visa garantir a segurança fim-a-fim no nível de mensagem e não somente no nível de transporte, tendo três principais pontos:

- *credenciais de segurança*: incluir nas mensagens SOAP credenciais de segurança com informações de autenticação;
- *integridade da mensagem*: incluir nas mensagens SOAP informações relacionadas a assinaturas digitais de toda ou de parte da mensagem;
- *confidencialidade da mensagem*: mensagens SOAP podem ser cifradas, totalmente ou somente partes dela.

A WS-Security define um esquema XML, o qual possibilita incluir de forma padronizada as informações relacionadas a assinatura e a cifragem dos dados da mensagem SOAP em questão, fazendo uso das especificações XMLDSign [Bartel et al. 2002] e o XMLEnc [Imamura et al. 2002]. Tais informações são inseridas dentro de elementos XML `<wsse:Security>` e cada mensagem SOAP poderá conter um ou mais destes elementos. Isso se justifica devido ao fato que o caminho percorrido por uma mensagem SOAP, da origem até o destino final, pode ser composto por diversos nós SOAP intermediários. Neste caso, a WS-Security consegue garantir que somente determinadas partes de uma mensagem SOAP possam ser lidas, modificadas por determinados nós intermediários.

[12]Número pseudo-aleatório utilizado uma única vez (do inglês: *number used once*).

Dessa forma, a WS-Security permite a inclusão de múltiplas assinaturas e cifragens nas mensagens SOAP. Cada elemento `<wsse:Security>` deverá identificar, através do atributo `SOAP1.2:role`, o nó a qual aquela informação está direcionada. Não é permitido que haja dois elementos `<wsse:Security>` que tenham como alvo um mesmo nó SOAP, porém informações como a assinatura do emissor inicial podem ser interessantes para todos os nós SOAP intermediários ou final. Desta forma, é possível definir um único elemento `<wsse:Security>` sem que necessite indicar o nó relacionado com este elemento. Isso permite que todos os nós SOAP possam tratar tal elemento.

Cada nó intermediário só pode processar o elemento `<wsse:Security>` direcionado a ele, podendo assim removê-lo ou mesmo adicionar novos elementos `<wsse:Security>`, antes de encaminhar para o próximo nó, presente no caminho da mensagem SOAP. É possível também que cada nó intermediário adicione novos subelementos a um elemento `<wsse:Security>` já existente.

```
1  <soapenv:Envelope
2    xmlns:soapenv="..." xmlns:wsse="...">
3    <soapenv:Header>
4
5      <wsse:Security>
6        <wsse:UsernameToken wsu:Id="...">
7          <wsse:Username>joao</wsse:Username>
8        </wsse:UsernameToken>
9      </wsse:Security>
10
11   </soapenv:Header>
12   <soapenv:Body>
13     ...
14   </soapenv:Body>
15 </soapenv:Envelope>
```

Figura 1.11. Mensagem SOAP ilustrando o *WS-Security*

A Figura 1.11 apresenta um exemplo de uma mensagem SOAP com o cabeçalho da *WS-Security*. O exemplo consiste em enviar uma simples credencial, no caso "joão" (linha 7), sem qualquer tipo de proteção. Na linha 2 da figura, são informadas as URI[13] para os espaços de nomes XML do SOAP e da WS-Security. Cada elemento `<wsse:Security>` (linhas 5 a 9) pode expressar informações sobre a cifragem, a assinatura e sobre as credenciais de segurança. As linhas 6 a 8 expressam detalhes sobre uma credencial de segurança, porém os elementos `<wsse:Security>` podem conter mais de uma credencial de segurança, se desejado for.

Em um cenário de uso para a mensagem apresentada na Figura 1.11, um nó SOAP, após receber uma invocação de um cliente, autenticado através de um mecanismo presente nas camadas subjacentes (como TLS/SSL), encaminha tal mensagem para outro nó SOAP, sendo que ambos os nós estariam presentes dentro de um mesmo ambiente considerado confiável e seguro. Assim, o objetivo da mensagem é indicar ao nó SOAP final que, em um nó SOAP mais externo, a autenticação do cliente já foi realizada e esta informação está

[13] A URI foi suprimida para facilitar a visualização do código.

sendo repassada através do elemento `<wsse:Username>` (linha 7). Supõe-se também que a segurança da comunicação entre os nós SOAP é garantida através, por exemplo, do TLS/SSL. A Figura 1.12 ilustra tal cenário.

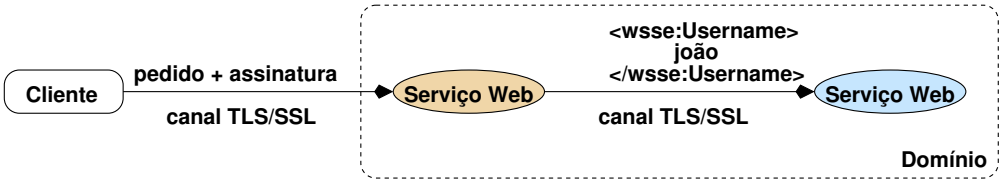


Figura 1.12. Encaminhando a identificação do cliente [Weerawarana et al. 2005]

No exemplo apresentado na Figura 1.12, a confidencialidade e a integridade das mensagens são garantidas através do uso do TLS/SSL, ou seja, no nível de transporte. Porém, pode-se ainda usar os padrões XMLEnc e do XMLDSign para garantir tais propriedades, evitando assim a necessidade do TLS/SSL. Outra forma ainda é a combinação do TLS/SSL com o XMLEnc e o XMLDSign.

Atualmente, a especificação *WS-Security* provê suporte a dois tipos de credenciais de segurança: credenciais `UsernameToken` e credenciais `BinarySecurityToken`. Um uso para a credencial `UsernameToken` foi descrito anteriormente e apresentado na Figura 1.12. Já a credencial `BinarySecurityToken` apresenta uma forma padrão para anexar a um pedido SOAP qualquer credencial de segurança codificada em forma binária; por exemplo, certificados X.509, *tickets* Kerberos, etc.

Políticas para os Serviços *Web*

Como visto anteriormente, a especificação WSDL surgiu da necessidade de um padrão para especificar as funcionalidades presentes em um Serviço *Web*. A WSDL permite aos provedores de serviços especificar quais são os serviços oferecidos, quais informações são necessárias para invocar um serviço e como deverá ser o formato para troca de informações com os clientes.

A WSDL só se preocupa em descrever as propriedades funcionais de um serviço, porém, é necessário uma forma padronizada e interoperável, para descrever as habilidades e requisitos não funcionais de um serviço. Tais habilidades não funcionais podem estar relacionadas com a segurança provida ou exigida pelo serviço. Os clientes, com base nessas informações, podem determinar qual serviço escolher, visando, por exemplo, serviços que apresentem uma política de privacidade bem definida, ou ainda, que garantam confidencialidade nas transações.

Como representar e anexar tais informações aos serviços ou recursos dentro do ambiente dos Serviços *Web*, serviu de motivação para o surgimento de documentos como o *WS-Policy* [WS-Policy 2004] e o *WS-PolicyAttachment* [WS-PolicyAttachment 2004]. A especificação *WS-Policy* provê um modelo de propósito geral para descrever políticas. Provê uma gramática flexível e extensível que permite descrever uma ampla variedade de requisitos e habilidades para o ambiente dos Serviços *Web*. A especificação *WS-PolicyAttachment* descreve como associar políticas com os determinados recursos e

também define como associar políticas a elementos XML que compõem um documento WSDL e a elementos UDDI.

Juntamente com o WSDL, o *WS-Policy* provê uma descrição declarativa dos requisitos que o serviço possui, os quais deverão ser cumpridos pelos requisitante. Porém, o uso de políticas não se limita somente aos serviços. Dentro do ambiente dos Serviços *Web*, existe uma ampla variedade de recursos, como documentos XML, sessões de mensagens confiáveis nas quais se podem associar políticas.

A especificação *WS-Policy* apresenta uma estrutura para descrição de políticas dividida em três principais componentes: *asserção de política* – expressa a habilidade do recurso, específica a um domínio, por exemplo, permitir a troca confiável de mensagens; *alternativas de políticas* – descrevem as combinações aceitáveis de obrigações e requisitos (conjunto de *asserções de política*), para a interação entre o serviço e um requisitor ou ainda para o acesso a um recurso; *política* – expressa um conjunto de alternativas de políticas válidas.

Segundo a *WS-Policy*, as *políticas* são representadas através de documentos XML, cujo elemento raiz do documento é o elemento `Policy`. Dentro deste elemento são representadas *coleções de asserções*, que quando combinadas representam um conjunto válido de *alternativas de políticas* . As *asserções* são combinadas através de dois tipos de *operadores de políticas*: `ExactlyOne` – indica que somente uma das *asserções* contidas na política poderá fazer parte de uma *alternativa de política*; `All` – permite a combinação de todas as *asserções* apresentadas como uma *alternativa de política*.

```
1 <wsp:Policy ...>
2   <wsp:ExactlyOne>
3     [ <wsp:All> [ <Assertion> ... </Assertion> ]* </wsp:All> ]*
4   </wsp:ExactlyOne>
5 </wsp:Policy>
```

Figura 1.13. Forma normal para expressar políticas [WS-Policy 2004]

Os operadores `ExactlyOne` e `All` podem ser combinados de diversas formas. Visando facilitar a interoperabilidade das políticas expressas pela *WS-Policy*, a especificação definiu uma *forma normal* para expressar as políticas. A Figura 1.13 apresenta a estrutura que uma política deve seguir para se adequar à *forma normal*. Dessa forma, cada alternativa de política válida fica contida dentro de um elemento `All` (linha 3) e todas as alternativas de políticas deverão estar contidas dentro de um único elemento `ExactlyOne` (linhas 2 a 4)[14]. Isso indica que só é possível escolher uma única alternativa e esta alternativa consiste na expressão completa de todas as asserções ali descritas [Weerawarana et al. 2005]. A especificação da *WS-Policy* também define um algoritmo para a tradução de qualquer expressão de política para a *forma normal*.

A Figura 1.14 ilustra uma política expressa de acordo com a *forma normal* da WS-Policy. No exemplo, a política indica que credenciais Kerberos ou X.509 podem ser utilizadas para prover a autenticação em um determinado recurso. As linhas 3 a 7 e 8 a

[14]O símbolo "*", de acordo com a notação do XML, indica a presença de 0 ou *mais* elementos.

12 apresentam duas *alternativas de política* e somente uma das duas alternativas poderá ser selecionada. Se a primeira for selecionada, indica que somente credenciais Kerberos serão aceitas e no caso de ser selecionada a segunda, então somente credenciais X.509 serão aceitas.

```
<wsp:Policy>
  <wsp:ExactlyOne>
    <wsp:All>
      <wsse:SecurityToken>
        <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
      </wsse:SecurityToken>
    </wsp:All>
    <wsp:All>
      <wsse:SecurityToken>
        <wsse:TokenType>wsse:X509v3</wsse:TokenType>
      </wsse:SecurityToken>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Figura 1.14. Uma política expressa de acordo com a WS-Policy [WS-Policy 2004]

Geralmente, o provedor de um Serviço *Web* expõe sua política com o objetivo de indicar sob quais condições irá prover seu serviço, ou seja, informa suas habilidades e seus requisitos. Um possível cliente, após analisar a política, pode decidir se está apto ou se deseja acessar o serviço ou não. A especificação da *WS-Policy* define somente uma gramática para expressar políticas, porém não especifica como associar tais políticas aos Serviços *Web* ou mesmo como divulgá-las, permitindo que outras especificações determinem como associar políticas de acordo com uma tecnologia específica. Essa separação da definição das políticas com a associação aos recursos permite que as políticas possam ser reutilizadas.

A *WS-PolicyAttachment* [WS-PolicyAttachment 2004] define diversos mecanismos para associar as políticas aos recursos. Dentro do ambiente dos Serviços *Web*, os recursos poderão ser uma troca de mensagens, um serviço, uma coleção de serviços, etc. A *WS-PolicyAttachment* define mecanismos que permitem que as políticas sejam anexadas diretamente dentro de documentos XML ou ainda permitem associar as políticas com os recursos de forma que não necessitem que as políticas e os recursos estejam presentes dentro de um mesmo documento XML. Diversos tipos, presentes nos documentos WSDL, podem constituir um recurso, como os elementos *messages*, *portType*, *binding*, *service*, entre outros. As políticas podem ser anexadas aos documentos WSDL, através de elementos `PolicyReference`.

Também vale citar a proposta *WS-SecurityPolicy* [WS-SecurityPolicy 2005] que tem por objetivo descrever como deverá ser a segurança no nível de mensagens, utilizando para isso as especificações *WS-Security* [OASIS 2004c], *WS-Trust* [WS-Trust 2005] e *WS-SecureConversation* [WS-SecureConversation 2005].

WS-Trust

Desenvolvida por um conjunto de empresas, lideradas pela Microsoft e IBM, a especificação *WS-Trust* [WS-Trust 2005] define serviços e protocolos visando a troca de atributos de segurança (p.ex.: asserções SAML), para possibilitar a comunicação entre diferentes domínios administrativos e de segurança. A especificação WS-Trust apesar de ser bastante citada em trabalhos acadêmicos e aparecer em algumas ferramentas de desenvolvimento para Serviços *Web*, ainda não recebeu aval de entidades padronizadoras. Porém, recentemente o órgão OASIS criou um comitê técnico, *OASIS WS-SX TC*[15], que objetiva definir padrões para a troca confiável de mensagens SOAP e o trabalho envolvido consistirá em refinamentos das propostas WS-Trust, *WS-SecurityPolicy* [WS-SecurityPolicy 2005] e *WS-SecureConversation* [WS-SecureConversation 2005], trazendo assim importância para tais propostas, que brevemente poderão se tornar padrões de fato.

O serviço de atributos de segurança (*Security Token Service – STS*) é definido pela *WS-Trust* como a autoridade responsável por emitir, renovar e validar os atributos de segurança, sendo este a base do modelo de confiança. O STS consiste de um Serviço *Web* que implementa uma interface WSDL, especificada pela *WS-Trust*, e que processa mensagens SOAP seguras, ou seja, que está de acordo com a especificação *WS-Security* [OASIS 2004c]. A interface do STS define duas operações, a `RequestSecToken` para realizar o pedido e a `RequestSecTokenResp` para a obtenção dos atributos de segurança.

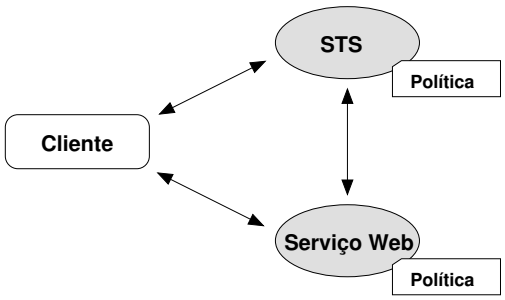


Figura 1.15. O uso do STS na mediação de confiança

A Figura 1.15 ilustra um caso típico de confiança mediada através do STS. O cliente deseja invocar o Serviço *Web*, porém de acordo com as políticas deste serviço (ex: expressas de acordo com a *WS-Policy*), é necessário que o cliente apresente credenciais de segurança emitidas pelo STS. Assim, o cliente deve obter as credenciais junto ao STS para que depois possa novamente invocar o serviço. É possível que o STS também exija algum tipo de autenticação do cliente.

Uma vez que o STS tenha analisado as credenciais fornecidas pelo cliente e verificado que as mesmas cumprem os requisitos necessários, o STS responde com a mensagem `RequestSecTokenResp`, que contém as credenciais necessárias para que o cliente consiga acessar o Serviço *Web*. A autenticidade da resposta pode ser garantida através de assinaturas digitais e a resposta ainda pode conter informações adicionais, como tempo

[15] `http://www.oasis-open.org/committees/ws-sx`

de vida da credencial e mecanismos para proteção contra ataque de mensagens antigas. Por fim, o cliente efetua um novo pedido ao Serviço *Web*, juntamente com as credenciais de segurança obtidas junto ao STS. O Serviço *Web* verifica as credenciais apresentadas e assim garante o acesso ao recurso.

A WS-Trust não se preocupa com as tecnologias de segurança subjacentes, deixando a escolha para o desenvolvedor da aplicação. O uso combinado das tecnologias de segurança presentes nas camadas de rede ou de transporte pode ser utilizado para garantir a autenticação do pedido em si. Por exemplo, o uso de assinatura de digital para provar que o cliente realmente é o detentor dos atributos apresentados ao serviço.

A especificação da WS-Trust não se preocupa com o estabelecimento das relações de confiança, mas usufrui das relações já estabelecidas, possibilitando assim que partes que não possuem relações estabelecidas possam se comunicar. As relações de confiança podem ser obtidas: (1) através de raízes fixas, em que é definido um conjunto fixo de entidades em quem se confia; (2) através de confiança hierárquica, em que a confiança é dada através de uma árvore e os nós inferiores confiam nos nós superiores; ou ainda, (3) através do uso das redes de confiança, em que cada entidade determina em quem confiar.

WS-Federation

A *WS-Federation* é uma proposta lançada pela Microsoft e IBM que apresenta meios para permitir que diferentes domínios de segurança possam criar federações de identidades e usufruir da mediação de confiança destas para permitir o compartilhamento de identidades, atributos e autenticação entre os participantes. Para isso, a *WS-Federation* faz uso dos modelos definidos nas especificações *WS-Security* [OASIS 2004c], *WS-Policy* [WS-Policy 2004] e *WS-Trust* [WS-Trust 2005], sendo que o STS, definido pela *WS-Trust*, assume também o papel de um *provedor de identidades*.

Ainda são definidas duas sub-especificações para a *WS-Federation*. A especificação *WS-Federation Passive Requestor* [WS-Federation 2003c] detalha como implementar as funcionalidades da federação em ambientes com clientes passivos. Os navegadores *Web* são considerados clientes passivos, pois não estão aptos a tratar as respostas SOAP enviadas por um Serviço *Web*. Desta forma, o processamento das mensagens deverá ter como base as funcionalidades do protocolo HTTP 1.1 (*GET*, *POST*, *redirects* e *cookies*). Já a especificação *WS-Federation Active Requestor* [WS-Federation 2003b] define como implementar as funcionalidades da federação em ambientes com clientes ativos. Clientes ativos são aqueles que estão aptos a emitir mensagens e reagir com as respostas de um Serviço *Web*, fazendo uso dos mecanismos definidos nas especificações *WS-Security*, *WS-Trust* e *WS-Federation*. Em geral, clientes ativos podem obter uma política, enviada através de uma mensagem de erro de um Serviço *Web*, processar essa política, obter as credenciais de segurança necessárias e refazer um novo pedido ao Serviço *Web* inicial.

1.5. Revisão da Literatura de Segurança em Serviços *Web*

Nesta seção serão apresentados alguns trabalhos que juntos cobrem diversas necessidades de segurança para a concepção de aplicações distribuídas, como o exemplo do portal de informações apresentado na seção 1.3.4. Os trabalhos estão focados em três dife-

rentes áreas, sendo alguns destinados aos problemas de gerenciamento de identidades, o que envolve questões de privacidade e anonimato; outros focados aos problemas de gerenciamento de políticas; e por fim alguns trabalhos voltados para o gerenciamento de confiança.

1.5.1. Gerenciamento de Identidades

Uma identidade digital consiste na representação de uma entidade em um domínio específico e geralmente está relacionada a domínios do mundo real. Uma entidade pode possuir múltiplas identidades, em que cada identidade é constituída por um conjunto de características, podendo estas serem únicas ou não a um domínio.

A identidade pode ser temporária ou permanente e pode assumir diferentes conotações, dependendo do contexto no qual esta se encontra. Para uma pessoa a identidade pode estar associada ao nome, endereço, documento de identidade. Já no contexto de uma empresa, a identidade pode estar associada com funções, privilégios, direitos e responsabilidades [Parr e Villars 2001].

O *gerenciamento de identidades* consiste de um sistema integrado de políticas, processos de negócios e tecnologias que permite às organizações controlar o acesso aos recursos providos aos seus usuários de forma segura, provendo confidencialidade às informações dos usuários. Diversos modelos foram propostos para o gerenciamento de identidades e em [Jøsang e Pope 2005, Jøsang et al. 2005] é apresentada uma breve descrição de alguns modelos.

O ***modelo tradicional*** de gerenciamento trata a identificação de forma isolada, sendo que o provedor de serviço também atua como o provedor de identidades e de credenciais (senhas associadas com os identificadores). Neste modelo, os usuários possuem identificadores únicos e específicos para cada serviço com o qual interagem, resultando assim em diferentes credenciais associadas com cada identificador.

O modelo de ***gerenciamento de identidades federadas*** surgiu para suprir as necessidades apresentadas pelo modelo de gerenciamento tradicional. Neste tipo de ambiente, é definido o conceito de **domínios**, nos quais estão presentes os provedores de serviço, de identidades e de credenciais, por exemplo, relacionados a uma determinada empresa. Assim, cada empresa constitui um domínio. O projeto *Liberty Alliance* (ver seção 1.5.1) e o projeto *Shibolleth* [Carmody 2001] são implementações abertas de modelos de gerenciamento de identidade federada.

No ambiente de identidades federadas, são estabelecidos acordos entre os domínios, os quais permitem que identidades locais a um domínio sejam reconhecidas nos demais domínios participantes do acordo. Neste caso, é estabelecido o mapeamento dos identificadores de um usuário em diferentes domínios. Por exemplo, o identificador `joao.pedro@empresa`, oriundo do domínio `empresa`, dentro do domínio `universidade`, será mapeado para o identificador `jp@universidade`. A federação de domínios de identificação dá a impressão aos usuários de possuírem um identificador único para todos os domínios que compõem a federação. Os usuários poderão continuar a manter identificadores locais a cada serviço ou mesmo domínio, porém o simples fato de possuírem tal identificador permite que estes usuários possam acessar serviços presentes

em qualquer domínio da federação.

No ***modelo centralizado*** de gerenciamento, considera-se a existência de um único provedor de identidades e de credenciais em uma federação, o qual é utilizado por todos os provedores de serviços da mesma. Neste modelo um usuário pode acessar todos os serviços presentes na federação utilizando um mesmo identificador. Em tese, o modelo se assemelha ao modelo de identidade federada, porém com a diferença de não necessitar do mapeamento de credenciais. A *WS-Federation* [WS-Federation 2003a] é um exemplo deste tipo de modelo.

[Damiani et al. 2003] apresenta um estudo sobre os problemas inerentes ao gerenciamento de múltiplas identidades, descrevendo os requisitos necessários que um sistema de gerenciamento de identidades deve atender. Dentre os requisitos apresentados, alguns estão diretamente preocupados com as necessidades de segurança dos clientes [W3C 2002, Rannenberg 2000, Asokan et al. 1997], como a privacidade, o anonimato, a responsabilidade, etc.

A seguir, serão apresentados alguns trabalhos que buscam atender estes requisitos dentro da arquitetura dos Serviços *Web*.

Privacidade

A especificação [W3C 2004a] apresenta algumas considerações sobre a privacidade na arquitetura dos Serviços *Web*, indicando que tal assunto ainda não está completamente solucionado e necessita de um estudo mais aprofundando. Em [W3C 2004b] são apresentados alguns requisitos para a arquitetura dos Serviços *Web*, necessários para garantir a proteção da privacidade dos clientes de um Serviço *Web*, sendo estes:

- a arquitetura deve permitir expressar políticas de privacidade sobre os Serviços *Web*;
- a política de privacidade de um Serviço *Web* deve ser expressa de acordo com a *Platform for Privacy Preferences* (P3P) [W3C 2002];
- a arquitetura deve prover um meio para que os clientes possam verificar as políticas de privacidade dos Serviços *Web*;
- a arquitetura deve permitir a propagação e a delegação da política de privacidade;
- os Serviços *Web* devem permitir interações onde uma ou mais partes são anônimas.

A *Platform for Privacy Preferences* (P3P) [W3C 2002] é um projeto do W3C que permite que os sítios *web* expressem suas políticas de privacidade de forma padronizada utilizando XML, dando aos usuários o conhecimento sobre como seus dados pessoais serão tratados.

A P3P provê ferramentas que permitem que o administrador de um sítio *web*, através de uma lista de requisitos, preencha como será a política de privacidade do sítio. Uma vez indicados todos os requisitos, a ferramenta retorna um código XML que pode ser facilmente associado a um sítio *web*. Navegadores *web*, compatíveis com a P3P, podem

obter a política dos sítios, comparar com as preferências de privacidade do usuário e decidir sobre a continuidade da transação com o sítio.

A P3P não define um padrão mínimo para garantir a privacidade e também não provê meios para monitorar se os sítios *web* estão honrando suas políticas. A P3P só define uma forma padrão para que sítios e clientes *web* possam informar e verificar as políticas de privacidade aplicadas naquele sítio *web*.

Segundo [Hung et al. 2004], o uso do P3P não pode ser diretamente aplicado no contexto dos Serviços *Web*, visto que o P3P foi projetado para que usuários de sítios *web* possam ter controle sobre suas informações pessoais. Outro problema é que os vocabulários da P3P estão direcionados principalmente para descrever as práticas de privacidade dos sítios *web*, sobre quais dados irão coletar dos usuários e o que irão fazer com essas informações. Dessa forma, é possível concluir que os requisitos apresentados em [W3C 2004b] ainda não atendem a real necessidade existente no ambiente dos Serviços *Web*, o que exige a criação de novas soluções para a área.

Anonimato

O anonimato é uma propriedade que está diretamente relacionada com a privacidade, porém com significado distinto. O acesso anônimo de um usuário a um sistema indica que o usuário não será identificado, garantindo assim a privacidade de sua identidade real.

Segundo [Cattaneo et al. 2004], em cenários onde os recursos podem ser acessados de forma anônima, porém por usuários autorizados pelas políticas de acesso do sistema, as soluções triviais poderiam seguir duas linhas:

- Restringir o acesso aos recursos com base no endereço de rede de um determinado domínio. Dessa forma, todos os usuários que estiverem dentro da rede, dita confiável, terão completo acesso ao recurso e de forma anônima. Porém, tal solução possui como pontos fracos a possibilidade de um usuário não autorizado conseguir acesso aos recursos pelo simples fato de estar na rede confiável, e negar o acesso a um usuário autorizado, pelo fato deste estar em uma rede não confiável.
- Fazer uso de credenciais de grupo. Por exemplo, o provedor do serviço poderia emitir certificados de grupo para todos os usuários autorizados, indicando que o usuário pertence ao grupo de usuários "confiáveis". Porém, um usuário, dito confiável, poderia ceder tal certificado a terceiros, comprometendo assim a segurança do sistema, sendo que em alguns casos fica impossível determinar qual dos usuários "confiáveis" delegou o certificado de grupo.

Em [Cattaneo et al. 2004] é apresentada uma extensão ao SOAP [W3C 2003] para permitir o acesso anônimo aos Serviços *Web*. A solução está baseada no fato de que os usuários só precisam provar, para um provedor de serviços, que pertencem a um determinado grupo, autorizado pelas políticas do sistema, evitando assim revelar sua identidade pessoal. A proposta dos autores consiste de uma variação do modelo para identificação

anônima de grupo introduzida em [Santis et al. 1998]. Este modelo apresenta um protocolo com *prova de conhecimento zero* (*zero-knowledge proof*) [Goldwasser et al. 1989] que permite um usuário se identificar, de forma anônima, para um sistema remoto.

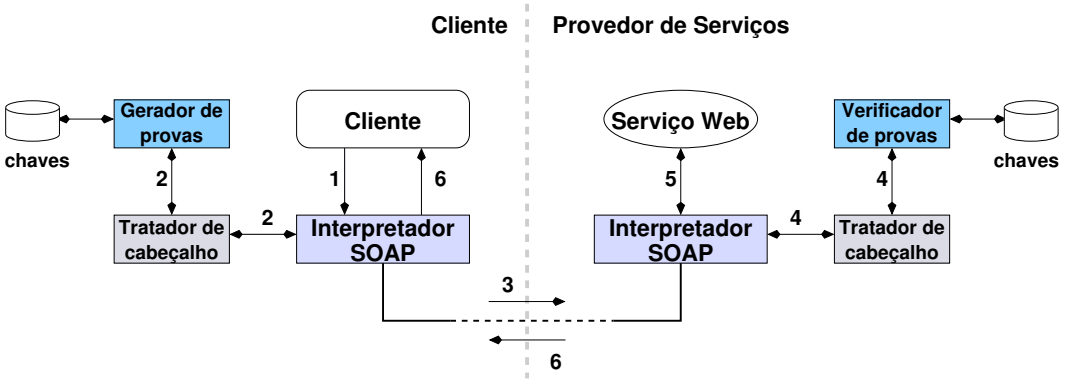


Figura 1.16. Pedido com identificação de grupo anônima [Cattaneo et al. 2004]

Para adicionar o anonimato de forma transparente para os Serviços *Web*, a proposta em [Cattaneo et al. 2004] define os componentes *gerador de provas*, presente no lado do cliente, e o *serviço para verificação de provas*, presente no lado do Serviço *Web* (veja a figura 1.16). As mensagens SOAP, originadas pela aplicação cliente são interceptadas, implicitamente ou explicitamente, pelo *gerador de provas* o qual irá gerar uma credencial de identificação anônima e incluir no cabeçalho SOAP, juntamente com um marcação temporal (passo 2). Da mesma forma, no lado do Serviço *Web* o pedido é interceptado e encaminhado ao *serviço para verificação de provas*, o qual verifica se a credencial é válida (passo 4).

Projeto Liberty Alliance

O projeto *Liberty Alliance* consiste em um conjunto de especificações produzidas por um consórcio de empresas atuantes nas mais diferentes áreas, como em telecomunicações, transportes, universidades, bancos, empresas de *software*, etc. Tem como principal objetivo criar especificações abertas para tratar o gerenciamento de identidades, usufruindo do conceito de *federação de identidades*. Os principais objetivos do projeto são [Liberty 2003a]:

- permitir aos usuários garantir a privacidade e a segurança de suas informações pessoais;
- prover um padrão aberto para permitir uma única autenticação (SSO), o que inclui a autenticação descentralizada e a autorização em múltiplos provedores de serviços;
- prover especificações, compatíveis com uma grande variedade de dispositivos;
- utilizar, em suas especificações, sistemas, padrões e protocolos existentes e amplamente aceitos;

- prover meios para que as empresas respeitem os requisitos de segurança e a privacidade dos clientes.

Esses objetivos podem ser alcançados quando provedores de serviços e clientes agrupam-se baseados em acordos comerciais e nas tecnologias propostas pela *Liberty*, formando assim os *círculos de confiança*. Tais círculos consistem na federação de provedores de serviços e serviços de identidade, juntamente com os clientes. A Figura 1.17 ilustra a arquitetura da *Liberty Alliance* dividida em quatro módulos.

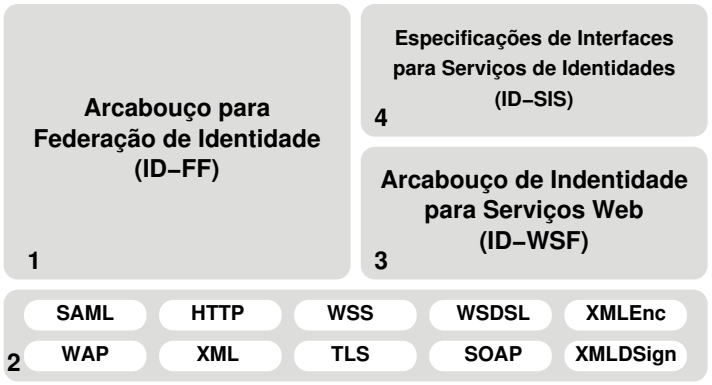


Figura 1.17. Arquitetura da *Liberty Alliance* [Liberty 2003a]

O módulo 1, *Liberty Identity Federation Framekwork* (IDFF), da figura 1.17 visa permitir a federação de identidades e o gerenciamento das mesmas através de características como ligação entre diferentes contas ou identidades, autenticação única (SSO) e o gerenciamento de sessões de forma simplificada. O módulo 2 ilustra a preocupação da *Liberty* em adotar e estender, de forma apropriada, os padrões da indústria ao invés de inventar novas especificações que atuariam de maneira semelhante às já existentes. Algumas das extensões da *Liberty* propostas para o SAML, por exemplo, foram aceitas pelo órgão OASIS e estão presentes na versão 2.0 da especificação do SAML [OASIS 2005c].

No módulo 3, *Liberty Identity Web Services Framework* (IDWSF), é definida uma estrutura para criação, descoberta e acesso aos serviços de identidade. A possibilidade de que empresas ofereçam serviços personalizados para os clientes, de acordo com os atributos e preferências que estes clientes escolheram compartilhar, é uma das principais características deste módulo.

No módulo 4, *Liberty Identity Services Interface Specifications* (IDSIS) , é definida uma coleção de especificações para a construção de serviços interoperáveis sobre a ID-WSF. Tais especificações foram definidas para permitir que organizações possam facilmente criar ou estender serviços sobre a estrutura da ID-WSF. Uma das especificações propostas foi a *ID-Personal Profile*, que define um serviço para obter informações pessoais de um usuário como nome, endereço, telefone, etc. Tal especificação permite que todas organizações que estejam de acordo com a *Liberty* possuam um conjunto de campos e valores conhecidos, tendo assim um dicionário e uma linguagem padrão para que possam interagir entre si. Já estão sendo definidas especificações para obter informações de um usuário relacionadas ao seu emprego, sobre sua geo-localização, etc.

Algumas especificações do projeto *Liberty Alliance* estão direcionadas para garantir a segurança e a privacidade dos clientes. Em [Liberty 2003b], um guia de "boas práticas" para serem seguidas pelos provedores de serviços que estejam de acordo *Liberty* é apresentado, frisando que cada empresa ainda deverá estar de acordo com a jurisdição a qual está submetida.

No guia é descrito que os serviços deverão informar, de forma clara, aos usuários quem está coletando suas informações pessoais, quais informações estão sendo coletadas e de que forma estão sendo coletadas. Os serviços deverão acatar as escolhas do usuário, com relação a privacidade de suas informações pessoais. O usuário deve ter o direito de escolher quais atributos um provedor de serviço terá acesso, bem como os meios para permitir gerenciar e indicar o tempo de vida das informações fornecidas. Deve-se também prover mecanismos para resolução de conflitos para o caso de um usuário acreditar que suas informações não estejam sendo manuseadas de forma incorreta. O guia também apresenta a necessidade de mecanismos que garantam o acesso às informações pessoais de outros usuários, principalmente quando amparado por uma ação judicial.

Os *identificadores opacos* ou *pseudônimos* foram propostos nas especificações da *Liberty* com o intuito de garantir a privacidade dos usuários dos serviços. Para cada provedor de serviço, o provedor de identidade poderá atribuir diferentes pseudônimos relacionados a um mesmo usuário. Dessa forma, o mesmo usuário seria representado por diferentes pseudônimos para cada serviço que fosse acessar, garantindo assim a proteção contra o rastreamento de suas transações. Identificadores opacos permitem aos provedores de serviços identificar quem são seus clientes, relacionando em suas contas locais, porém não possibilita que os provedores de serviços obtenham informações pessoais dos clientes de forma que possa comprometer a privacidade do mesmo.

1.5.2. Gerenciamento de Políticas de segurança

As políticas de segurança descrevem as necessidades e as obrigações de segurança para um dado domínio de segurança. No ambiente dos Serviços *Web*, é comum que um fluxo de negócios seja composto por diversos serviços presentes em diferentes domínios administrativos e de segurança. Dessa forma, um fluxo de negócios pode ser regido por diferentes políticas de segurança e o gerenciamento das mesmas para que a transação ocorra com sucesso e forma segura é um desafio.

Por exemplo, a comunicação de todos os nós de um determinado domínio deverá ser cifrada e assinada, garantindo as propriedades básicas de segurança. Visando remover a complexidade dos nós em ter que trabalhar com uma Infra-estrutura de Chave Pública (ICP), um único nó do domínio poderia ficar responsável pelos processos de cifragem, decifragem, assinatura e verificação de assinaturas. Já em um outro caso, tal solução não seria ideal, visto que é desejado garantir uma segurança fim-a-fim, ou seja, uma vez que a mensagem tenha sido cifrada e/ou assinada pela origem, somente o nó destino poderá ler e/ou modificar a mesma (diferentes contextos de segurança, apresentados na figura 1.4).

Neste caso, a flexibilidade para localização das operações de segurança também deve ser considerada como um requisito da especificação da política de segurança. Em ambiente multi-salto, a política deve ser flexível o suficiente para permitir regras que definam onde deverá ocorrer a cifragem, a decifragem, a assinatura ou a verificação da

assinatura.

Em [Chang et al. 2003], é apresentado um modelo para o gerenciamento de políticas de segurança para ambientes de larga escala compostos por Serviços *Web*. Segundo [Chang et al. 2003], para que uma política de segurança garanta um acordo fim-a-fim, deve-se considerar a interoperabilidade entre as versões das políticas de segurança; garantir a privacidade das partes envolvidas e visar o estabelecimento dinâmico das políticas de segurança, visto que as políticas definidas estaticamente podem se tornar inseguras depois de um certo tempo.

Porém, se por um lado há necessidade de uma evolução dinâmica das políticas, por outro lado tal fato pode trazer um problema. Em ambientes de larga escala, uma transação pode possuir uma longa duração e mudanças nas políticas de segurança não deveriam interferir nessa transação. Dessa forma, é importante também considerar a interoperabilidade entre as versões que uma política pode assumir.

A proposta de [Chang et al. 2003], baseada no uso de um *contrato interoperável* (*Interoperability Contract Document* – ICD), permite a colaboração entre as partes para estabelecer políticas de segurança dinâmicas e individuais para cada operação do serviço e provê ainda medidas para o controle de versão e interoperabilidade destas políticas. A Figura 1.18 ilustra os passos envolvidos em uma transação de acordo com o modelo.

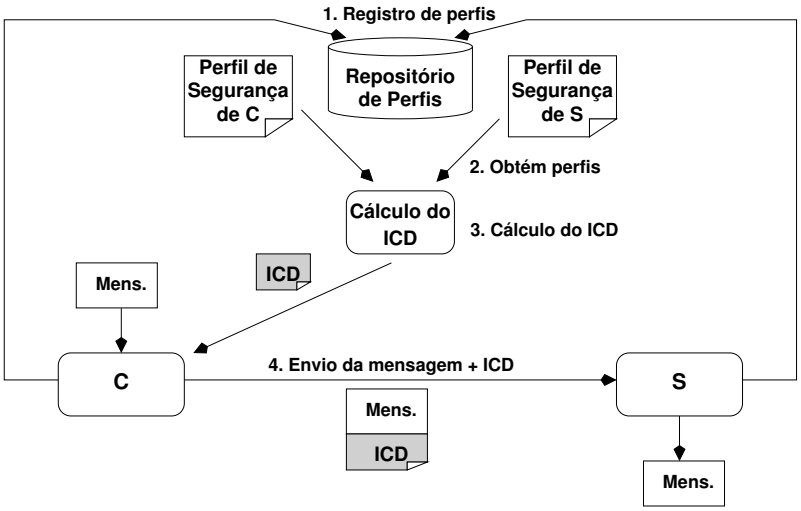


Figura 1.18. O uso do ICD [Chang et al. 2003]

Cada parte, no caso *C* e *S*, registram seus perfis de segurança no repositório de perfis (passo 1). Cada perfil pode estar relacionado a um grupo de serviços ou a serviços individuais e contém as políticas de segurança, suas preferências, etc. Tal repositório está acessível somente para as partes registradas, no exemplo, para *C* e *S*, respectivamente.

Uma vez que *C* queira enviar uma mensagem para *S*, este aciona o módulo para o cálculo do ICD, o qual obtém os perfis de segurança de *C* e de *S* no repositório de perfis (passos 2 e 3). O cálculo do ICD consiste de uma intersecção das preferências de segurança de ambas as partes. Se o resultado for um conjunto vazio, então o processo é abortado através de uma exceção. Por outro lado, se existir mais de um elemento dentro do conjunto, a seleção será baseada em uma ordem de prioridade, a saber: as preferências

do receptor; as preferências do emissor; o nível de segurança; e o desempenho.

Para todas as mensagens que forem enviadas por *C*, será anexado o ICD, indicando assim a política de segurança aplicada especificamente àquela mensagem (passo 4). A mensagem SOAP é personalizada de acordo com as informações de segurança baseadas no ICD. Por exemplo, se para um dado nó a credencial de autenticação for requerida, esta será inserida na parte de autenticação do cabeçalho SOAP. E, se a integridade for requerida, a mensagem será assinada e o algoritmo utilizado será identificado pelo ICD, o mesmo para a cifragem. Por fim, na recepção da mensagem, através do ICD, *S* consegue verificar se está de acordo com o combinado.

O ICD anexado à mensagem possibilita uma evolução da aplicação sem que isso acarrete problemas de segurança. Dessa forma, quando a versão da política de segurança de qualquer parte avançar, um novo ICD é recalculado e será anexado nas próximas mensagens que sairão. Com isso, uma parte consegue aplicar diferentes versões da política para interoperar com diferentes parceiros sem ambigüidades.

Conforme apresentado na Figura 1.18, os perfis de segurança de *C* e de *S* estão armazenados em um mesmo repositório, porém é previsto no modelo a presença de diversos repositórios, por exemplo, um por domínio administrativo e assim, no cálculo do ICD os perfis de segurança podem ser obtidos de diferentes repositórios. No trabalho [Chang et al. 2003] não é apresentada uma forma para localizar tais repositórios, bem como para saber em qual repositório estará armazenado o perfil de segurança de cada parte.

1.5.3. Gerenciamento de Confiança

A federação de identidades tem como ponto fundamental as relações de confiança entre provedores de serviços e clientes, e entre os próprios provedores de serviço. O fornecimento de informações pessoais de um cliente a um provedor de serviço só ocorre depois que o cliente tenha certeza de que suas informações serão manipuladas de maneira correta. Por outro lado, o provedor de serviço só irá conceder ao cliente o acesso ao recurso, se o mesmo confiar nas informações fornecidas pelo cliente. O mesmo ocorre nas relações de confiança entre provedores de serviço. Tais relações permitem, por exemplo, que um usuário autenticado em um determinado provedor possa usufruir dos recursos providos por outro provedor, já que ambos possuem um acordo indicando o compartilhamento de recursos para os usuários de ambos provedores.

A seguir, serão apresentados alguns trabalhos que tratam diretamente com o problema da confiança, com um enfoque voltado na negociação da confiança dentro do ambiente dos Serviços *Web*.

TrustBuilder

Em [Winslett et al. 2002], é apresentado o sistema *TrustBuilder* que tem por objetivo permitir o estabelecimento dinâmico da confiança entre partes estranhas dentro do contexto da Internet. O *TrustBuilder* permite que as partes, envolvidas na negociação, revelem gradualmente suas credenciais e políticas de controle de acesso para estabelecer a confiança

necessária para a realização da comunicação efetiva entre as partes. Este trabalho está focado na definição de estratégias e protocolos para o estabelecimento da confiança.

O estabelecimento da confiança leva em consideração que cada parte possui políticas de controle de acesso sobre os recursos que deseja proteger. Tais recursos podem ser os serviços providos por uma determinada parte ou ainda as credenciais de segurança, como o número do documento de identidade, número do passaporte, número do cartão de crédito, etc. As políticas também definem quais credenciais específicas, devem ser apresentadas para que se obtenha acesso ao recurso desejado.

As estratégias controlam quais e quando as credencias serão reveladas e também quando a negociação será finalizada. Neste caso, as estratégias estarão trabalhando juntamente com as políticas de controle de acesso. Já o protocolo indica a ordem das mensagens a serem trocadas, bem como quais informações deverão estar contidas nas mensagens.

A revelação gradual das credenciais trata de uma medida preventiva contra partes com quem ainda não exista uma relação de confiança. Por exemplo, João deseja comprar um produto na empresa XYZ. Para que se possa confirmar a compra, é necessário que João forneça o número do seu cartão de crédito, porém João não fará isso sem que antes a empresa XYZ forneça informações de que a mesma trata-se de uma empresa idônea, informação esta que pode ser emitida por um órgão no qual João já confia. Neste caso, o problema está em como revelar as credencias de cada parte, sem que isso venha trazer prejuízos caso alguma parte não honre suas obrigações.

Uma solução para o problema consiste na utilização de uma Terceira Parte Confiável (TPC), em que ambas as partes envolvidas na comunicação, revelam suas credenciais e políticas para a TPC e delegam a esta a tarefa de determinar a confiança entre cada parte. Porém, tal solução torna-se um gargalo em ambientes de larga escala e ainda um ponto único de vulnerabilidade. O uso do conceito de *prova de conhecimento zero* conseguiria provar que as credenciais respeitam as políticas sem que seja preciso revelar tanto as credenciais quanto as políticas (o trabalho de [Cattaneo et al. 2004], apresentado anteriormente faz uso deste conceito). Porém, segundo apresentado em [Winslett et al. 2002], tal solução é difícil de ser implementada de forma eficiente.

O *TrustBuilder* baseia-se na negociação direta entre as partes, através da revelação parcial das credenciais e políticas. Sabendo que cada parte pode possuir diversas políticas e credenciais de segurança, as quais podem ou não ser utilizadas em determinadas negociações, a medida, considerada por [Winslett et al. 2002] como a melhor alternativa, consiste em somente revelar as políticas necessárias para a comunicação em questão.

A Figura 1.19 ilustra um exemplo apresentado em [Winslett et al. 2002]. No passo 1, João deseja comprar um produto da empresa XYZ, indicando que quer um desconto no valor final do produto. A empresa XYZ definiu em suas políticas que somente os clientes que comprovarem que são "revendedores" poderão obter desconto, dessa forma, no passo 2, é enviada a política $P2$ para João, requerendo uma "credencial de revendedor" e também o número do cartão de crédito para que se possa efetivar a venda.

Por sua vez, João também possui um política local a qual indica que só fornecerá seu número de cartão de crédito a instituições que estejam devidamente regulamentadas

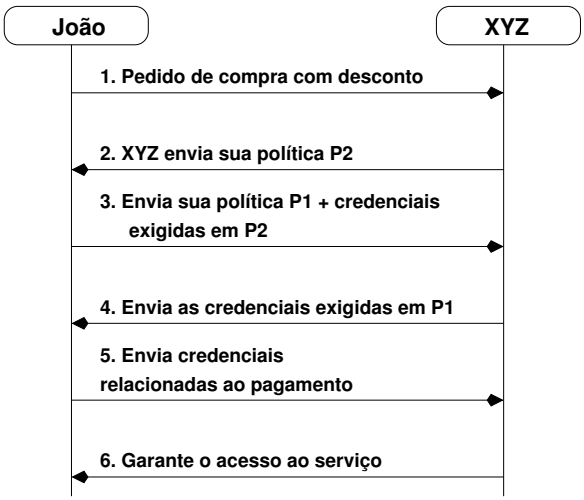


Figura 1.19. *TrustBuilder*: negociação de confiança

na *Federação da Indústria e do Comércio*. Assim, no passo 3 João fornece a credencial de que é um revendedor juntamente com a sua política P1, indicando que XYZ prove que faz parte da *Federação da Indústria e do Comércio*. No passo 4, XYZ fornece a credencial exigida por P1 e assim, no próximo passo, João fornece o número do cartão de crédito. Por fim, após a empresa XYZ confrontar as credenciais fornecidas com a sua política de controle de acesso, esta garante a João a venda do produto.

Trust-Serv

O *Trust-Serv* [Skogsrud et al. 2003] é uma infra-estrutura voltada para o estabelecimento de confiança dentro do ambiente dos Serviços *Web*. O trabalho apresenta um modelo de políticas para o estabelecimento de confiança baseado em máquinas de estado [Hopcroft e Ullman 1979]. Neste trabalho, um modelo para o gerenciamento do ciclo de vida das políticas o que permite a evolução, bem como a migração, das políticas sem que isso interrompa as negociações que já estejam em andamento é apresentado. A estratégia proposta também permite compensar o requisitor, caso os direitos de acesso concedidos a este sejam revogados em uma migração para uma nova política. Segundo [Skogsrud et al. 2003], o *Trust-Serv* é um trabalho complementar ao *TrustBuilder* [Winslett et al. 2002] e pode ser utilizado para prover o gerenciamento do ciclo de vida das políticas.

No *Trust-Serv*, os *estados* de uma máquina de estados representam o nível de confiança atingido pelo requisitor e para cada novo estado que o requisitor atingir o acesso a novos recursos será garantido. Os recursos são definidos como operações dos Serviços *Web* ou credenciais do próprio provedor de serviços. O *Trust-Serv* ainda adota o conceito de *papéis* [Ferraiolo et al. 2001] e, ao invés de associar os recursos diretamente aos *estados*, associam-se papéis. No modelo, os papéis são acumulativos e assim, os papéis ativados em um estado anterior não serão desativados ao se atingir um novo estado. Já as *transações* indicam as condições que um requisitor deve cumprir para que possa sair de um estado e ir para outro. Em [Skogsrud et al. 2003], são propostas extensões às tran-

sações de uma máquina de estado tradicional para capturar as abstrações de segurança, necessárias para o estabelecimento da confiança.

A arquitetura do *Trust-Serv* é dividida em camadas o que permite separar os mecanismos para o estabelecimento da confiança e o controle de acesso (nível de controle) da lógica de aplicação (nível de serviço). Para cada Serviço *Web*, é associado um *controlador*, o qual intercepta de forma transparente todas as mensagens direcionadas a este serviço. Os controladores podem aceitar ou recusar uma invocação ou ainda iniciar uma iteração com o *controlador* da outra parte para estabelecer um nível de confiança, antes de aceitar a invocação. A Figura 1.20 ilustra a disposição dos *controladores* na arquitetura do *Trust-Serv*.

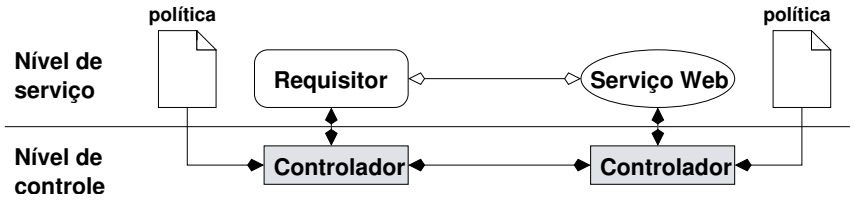


Figura 1.20. *TrustServ*: Níveis de serviço e de controle [Skogsrud et al. 2004]

O trabalho também propõe soluções para o gerenciamento do ciclo de vida das políticas, no caso, o foco do trabalho está direcionado ao contexto das políticas para o estabelecimento da confiança entre os Serviços *Web*. A substituição direta de uma política que já está sendo empregada por uma nova política pode não ser o ideal, visto que isso poderia acarretar no reinício de todas as negociações que já estão em andamento. O *Trust-Serv* provê diferentes estratégias para lidar com estes problemas:

- **coexistência**: permite que a negociação em andamento seja finalizada de acordo com a política antiga, porém requer que todas as novas negociações obedeçam a nova política;
- **abortamento**: aborta todas as negociações que estejam em andamento;
- **migração**: migra todas as negociações existentes para a nova política, desde que a política antiga esteja contida dentro da nova política. Dessa forma, todos os estados já visitados e todas as transações já disparadas na política antiga deverão estar presentes na nova política. As negociações em andamento que estejam de acordo com essa medida serão migradas para a nova política. Já as negociações que não estejam de acordo serão retornadas até atingirem um estado que esteja de acordo com a nova política.

Em alguns casos, as estratégias *Coexistência* e *Abortar* podem não ser ideais. A coexistência de duas políticas operando ao mesmo tempo pode não ser desejado, por exemplo, do ponto de vista legal. Clientes estariam recebendo tratamento diferenciados diante, por exemplo, de um mesmo pagamento. Já abortar todas as transações existentes poderiam trazer prejuízos para o provedor do serviço, visto que diversos clientes poderiam desistir de tentar recomeçar toda a negociação novamente. Por outro lado, a estratégia da

migração consegue unir as vantagens das duas estratégias anteriores. A regressão até um estado que seja comum às duas políticas evita que toda a transação tenha que ser refeita e também garante que os próximos estados estarão de acordo com a nova política, evitando assim a coexistência de diferentes políticas.

1.6. Ferramentas para o Desenvolvimento de Serviços *Web* Seguros

Um Serviço *Web* consiste de um componente de software que permite a interação entre aplicações através de uma rede [W3C 2004a]. Em resumo, qualquer aplicação que consiga enviar, receber e processar mensagens SOAP trocadas através de algum protocolo de transporte pode ser considerada um Serviço *Web*.

Existem hoje diversas ferramentas para o desenvolvimento de aplicações baseadas na arquitetura dos Serviços *Web*. Algumas destas são ambientes completos para o desenvolvimento e disponibilização dos serviços, como a *Java Web Services Development Pack* (JWSDP)[16], já outras são só implementações do SOAP, como é o caso do Apache Axis[17] e necessitam de outras ferramentas para permitir a disponibilização dos serviços ou mesmo para agregar características de segurança. As licenças de distribuição destas ferramentas também são outro ponto de divergência. Algumas estão sob licença de código fechado (proprietário) e exigem o pagamento para o uso de ferramentas, outras possuem uma licença de distribuição gratuita, porém não disponibilizam os códigos fontes, e, por fim, existem ainda as ferramentas sob licença de *software* livre. Esta seção aborda algumas ferramentas sob licença de *software* livre, devido à natureza destas ser mais adequada ao meio acadêmico e por que estas permitem a divulgação do conhecimento através da disponibilização do código e da possibilidade de modificar e redistribuir o mesmo.

O Apache Axis é uma ferramenta de código aberto que disponibiliza um servidor SOAP juntamente com um conjunto de APIs que facilita o desenvolvimento de aplicações clientes e de Serviços *Web*. Apesar do Axis apresentar um servidor HTTP próprio para a disponibilização dos serviços, os desenvolvedores geralmente fazem uso do servidor de aplicação Apache TomCat[18], que também possui uma licença de código aberto, já que este último é mais robusto e completo.

Atualmente, o Axis possui duas versões que estão sendo desenvolvidas em paralelo. A versão 1 possui implementações em Java e em C++, e além de ser um interpretador SOAP, apresenta ferramentas e APIs para tratar diretamente com documentos WSDL (ver seção 1.3). Já a versão 2 do Axis[19], recentemente lançada, só possui a implementação em Java e segundo sua documentação, trata-se de uma completa reestruturação da versão 1, possuindo uma melhor modularidade, mais focada no XML e mais eficiente que a versão 1. Porém, a principal diferença é que a versão 2 não se resume apenas na implementação das especificações SOAP 1.1 e SOAP 1.2. Esta versão apresenta também uma melhor integração com as propostas para os Serviços *Web* (como a *WS-Security* [OASIS 2004c], *WS-Coordination* [Cabrera et al. 2004], entre outras), permitindo a integração destas através de módulos de *software*.

[16] `http://java.sun.com/webservices/jwsdp`
[17] `http://ws.apache.org/axis`
[18] `http://tomcat.apache.org`
[19] `http://ws.apache.org/axis2`

Porém, o Axis não apresenta soluções para prover segurança nas aplicações desenvolvidas com ele. Para isso, a própria fundação Apache lançou diversas outras ferramentas que implementam as principais especificações de segurança descritas na seção 1.4. A Apache XMLSecurity[20] é uma implementação de código aberto para as especificações *XMLDSign* e *XMLEncryption*. Trata-se de um trabalho bem maduro e amplamente aceito, sendo até mesmo utilizado por outras plataformas de desenvolvimento, como a JWSDP da empresa Sun.

O projeto Apache WSS4J (*WS-Security for Java*) é uma implementação de código aberto da especificação *WS-Security* [OASIS 2004c], que consiste de uma biblioteca Java que pode ser usada para assinar e verificar mensagens SOAP que contenham informações expressas de acordo com a *WS-Security*. Diferentemente da biblioteca Apache XMLSecurity, a WSS4J está diretamente ligada ao Apache Axis, a XMLSecurity e ainda a biblioteca OpenSAML[21], uma implementação de código aberto para a SAML[OASIS 2005c]. Estão surgindo alguns esforços para agregar à WSS4J os conceitos definidos na especificação *WS-Trust* [WS-Trust 2005], mas a implementação ainda não está tão madura quanto as outras bibliotecas apresentadas aqui.

1.7. Conclusão

Pode-se dizer que o grande sucesso das aplicações para Internet se deu devido ao alto nível de abstração. Isto permitiu garantir a interoperabilidade entre as mais diversas aplicações, sistemas operacionais e equipamentos. Os Serviços *Web* exploram esse nível de abstração associado a uma lógica de negócios.

Neste capítulo buscou-se introduzir os conceitos que regem a arquitetura orientada a serviços, tendo como foco os Serviços *Web*. Foram apresentados alguns dos desafios de segurança relacionados à arquitetura dos Serviços *Web*, bem como alguns trabalhos que visam tratar tais desafios.

Como visto, além das propriedades básicas de segurança, a concepção de aplicações baseadas nos Serviços *Web* deve considerar pontos como a transposição de domínios administrativos e de segurança, o que acarreta em preocupações com a privacidade, o anonimato, a evolução das políticas de segurança, e principalmente, a interoperabilidade. Para muitos destes desafios, já foram lançadas diversas propostas com o aval de órgãos padronizadores, fornecendo assim um ponto de partida comum para que desenvolvedores possam criar suas aplicações e que as mesmas serão interoperáveis.

Porém, muitas especificações para o ambiente dos Serviços *Web* são projetadas para serem estendidas e ainda apresentam diversas formas para expressar a mesma função, o que permite diferentes interpretações e como conseqüência, tais características tornam-se uma barreira contra a interoperabilidade [WS-I 2005]. Por exemplo, a especificação *WS-Security* introduz muitas opções e escolhas. Se diferentes empresas selecionam diferentes opções, estas podem não mais interoperar, apesar de estarem seguindo a mesma especificação. Já existem preocupações a respeito e o órgão WS-I já está apresentando algumas recomendações para o uso de padrões e tecnologias de segurança para os Serviços

[20] `http://xml.apache.org/security/`
[21] `http://www.opensaml.org`

Web, de forma a garantir a real interoperabilidade entre as implementações.

Enquanto em algumas áreas de segurança já existem especificações consolidadas, mesmo diante de algumas ambigüidades, em outras áreas, como o gerenciamento de políticas e de confiança, embora este capítulo tenha apresentado alguns trabalhos, não existem ainda padrões de fato, sendo esta um bom ambiente para pesquisa.

No sítio `http://www.das.ufsc.br/seguranca/webservices` estão disponíveis informações sobre as recentes pesquisas[22] feitas pelo Grupo de Computação Segura e Confiável (GCSeg) da UFSC, no contexto do projeto "Infra-estrutura de Segurança para Aplicações Distribuídas Orientadas a Serviço", que tem apoio do CNPq. Além disso, também estão disponíveis documentos técnicos que detalham como implantar a infra-estrutura necessária para o desenvolvimento e provimento de aplicações baseadas nos Serviços *Web*, e alguns exemplos de código, estes disponíveis sob licenças de *software* livre.

Referências

[Amoroso 1994] Amoroso, E. G. (1994). *Fundamentals of Computer Security Technology*. Prentice Hall.

[Asokan et al. 1997] Asokan, N., Schunter, M., e Waidner, M. (1997). Optimistic protocols for fair exchange. In *CCS '97: 4th ACM conference on Computer and communications security*, pages 7–17, New York, NY, USA. ACM Press.

[Bartel et al. 2002] Bartel, M., Boyer, J., e Fox, B. (2002). *XML-Signature Syntax and Processing*. W3C. `http://www.w3.org/TR/xmldsig-core`.

[Bishop e Bailey 1996] Bishop, M. e Bailey, D. (1996). A critical analysis of vulnerability taxonomies. Technical Report CSE-96-11, Department of Computer Science at University of California, Davis.

[Boyer 2001] Boyer, J. (2001). *Canonical XML*. W3C. `http://www.w3.org/TR/xml-c14n`.

[Brown e Kindel 1996] Brown, N. e Kindel, C. (1996). *Distributed Component Object Model Protocol – DCOM/1.0*. Microsoft.

[Cabrera et al. 2004] Cabrera, F., Copeland, G., Freund, T., Klein, J., Langworthy, D., Orchard, D., Shewchuk, J., e Storey, T. (2004). *Web Services Coordination*. Web Services Interoperability Organization. `http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Coordination.pdf`.

[Carmody 2001] Carmody, S. (2001). *Shibboleth Overview and Requirements*. Shibboleth Working Group.

[Cattaneo et al. 2004] Cattaneo, G., Faruolo, P., e Petrillo, U. F. (2004). Providing privacy for web services by anonymous group identification. In *International Conference on Web Services (ICWS'04)*. IEEE.

[22] [de Mello et al. 2005, de Mello e da Silva Fraga 2005, Wangham et al. 2005, Wangham et al. 2006]

[Chang et al. 2003] Chang, S., Chen, W., e Hsu, M. (2003). Managing security policy in a large distributed web services environment. In *27th International Computer Software and Applications Conference (COMPSAC'03)*. IEEE.

[Charfi e Mezini 2005] Charfi, A. e Mezini, M. (2005). Using aspects for security engineering of web service compositions. In *Proceedings of the 2005 IEEE International Conference on Web Services, Volume I*, pages 59–66.

[Daemen e Rijmen 2002] Daemen, J. e Rijmen, V. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag.

[Damiani et al. 2003] Damiani, E., di Vimercati, S. D. C., e Samarati, P. (2003). Managing multiple and dependable identities. In *IEEE Internet Computing*, pages 29–37. IEEE.

[de Mello e da Silva Fraga 2005] de Mello, E. R. e da Silva Fraga, J. (2005). Mediation of trust across web services. In *3rd IEEE International Conference on Web Services (ICWS'05)*, pages 515–522, Orlando, Flórida - EUA.

[de Mello et al. 2005] de Mello, E. R., Wangham, M., da Silva Fraga, J., e Rabelo, R. (2005). A secure model to establish trust relationships in web services for virtual organizations. In Camarinha-Matos, L. M., Afsarmanesh, H., e Ortiz, A., editors, *6th IFIP Working Conference on Virtual Enterprises (PRO-VE'05)*, pages 183–190, Valência, Espanha. Springer.

[Demchenko et al. 2005] Demchenko, Y., Gommans, L., de Laat, C., e Oudenaarde, B. (2005). Web services and grid security vulnerabilities and threats analysis and model. `http://www.uazone.org/demch/analytic/draft-grid-security-incident-04.pdf`.

[Dierks e Allen 1999] Dierks, T. e Allen, C. (1999). *The TLS Protocol – Version 1.0*. IETF RFC 2246.

[Eastlake e Jones 2001] Eastlake, D. e Jones, P. (2001). *US Secure Hash Algorithm 1 (SHA1)*. Internet Engineering Task Force RFC 3174.

[Ellison et al. 1999] Ellison, C. M., Frantz, B., Lampson, B., Rivest, R., Thomas, B. M., e Ylonen, T. (1999). *SPKI Certificate Theory*. Internet Engineering Task Force RFC 2693.

[Ferraiolo et al. 2001] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., e Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3):224–274.

[Freier et al. 1996] Freier, A. O., Karlton, P., e Kocher, P. C. (1996). *The SSL protocol - v.3*. Internet Draft.

[Goldwasser et al. 1989] Goldwasser, S., Micali, S., e Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208.

[Hallam-Baker e Mysore 2005] Hallam-Baker, P. e Mysore, S. H. (2005). *XML Key Management Specification (XKMS 2.0)*. W3C – Proposed Recommendation.

[Hopcroft e Ullman 1979] Hopcroft, J. e Ullman, J. (1979). *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.

[Housley et al. 2002] Housley, R., Polk, W., Ford, W., e Solo, D. (2002). *Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF RFC 3280.

[Hung et al. 2004] Hung, P. C. K., Ferrari, E., e Carminati, B. (2004). Towards standardized web services privacy technologies. In *International Conference on Web Services (ICWS'04)*. IEEE.

[IBM e Microsoft 2002] IBM e Microsoft (2002). *Security in a Web Services World: A Proposed Architecture and Roadmap*. IBM Corporation and Microsoft Corporation. `http://msdn.microsoft.com/ws-security/`.

[Imamura et al. 2002] Imamura, T., Dillaway, B., e Simon, E. (2002). *XML Encryption Syntax and Processing*. W3C. `http://www.w3.org/TR/xmlenc-core`.

[Jøsang et al. 2005] Jøsang, A., Fabre, J., Hay, B., Dalziel, J., e Pope, S. (2005). Trust requirements in identity management. In *CRPIT '44: Proceedings of the 2005 Australasian workshop on Grid computing and e-research*, pages 99–108, Darlinghurst, Australia. Australian Computer Society, Inc.

[Jøsang e Pope 2005] Jøsang, A. e Pope, S. (2005). User centric identity management. In *AusCERT Asia Pacific Information Technology Security Conference 2005*.

[Kohl e Neuman 1993] Kohl, J. e Neuman, C. (1993). *The Kerberos Network Authentication Service (v5)*. Internet Engineering Task Force RFC 1510.

[Landwehr 2001] Landwehr, C. E. (2001). Computer Security. In *International Journal of Information Security*, volume 1, pages 3–13. Springer-Verlag Heidelberg.

[Liberty 2003a] Liberty (2003a). *Introduction to the Liberty Alliance Identity Architecture*. Liberty Alliance.

[Liberty 2003b] Liberty (2003b). *Privacy and Security Best Practices*. Liberty Alliance.

[Lorch et al. 2003] Lorch, M., Proctor, S., Lepro, R., Kafura, D., e Shah, S. (2003). First experiences using xacml for access control in distributed systems. In *ACM Workshop on XML Security*.

[OASIS 2002] OASIS (2002). *Universal Description, Discovery and Integration v2 (UDDI)*. Organization for the Advancement of Structured Information Standards (OASIS).

[OASIS 2004a] OASIS (2004a). *Introduction to UDDI: Important features and functional concepts*. Organization for the Advancement of Structured Information Standards (OASIS). `http://uddi.org/pubs/uddi-tech-wp.pdf`.

[OASIS 2004b] OASIS (2004b). *Universal Description, Discovery and Integration v3.0.2 (UDDI)*. Organization for the Advancement of Structured Information Standards (OASIS).

[OASIS 2004c] OASIS (2004c). *Web Services Security: SOAP Message Security 1.0*. OASIS. `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf`.

[OASIS 2005a] OASIS (2005a). *eXtensible Access Control Markup Language (XACML) version 2.0*. Organization for the Advancement of Structured Information Standards (OASIS). `http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf`.

[OASIS 2005b] OASIS (2005b). *SAML Executive Overview*. Organization for the Advancement of Structured Information Standards (OASIS).

[OASIS 2005c] OASIS (2005c). *Security Assertion Markup Language (SAML) 2.0 Technical Overview*. Organization for the Advancement of Structured Information Standards (OASIS).

[OMG 2002] OMG (2002). *The Common Object Request Broker Architecture v3.0.2*. Object Management Group (OMG).

[OpenGroup 1997] OpenGroup (1997). *DCE 1.1: Remote Procedure Call*. Open Group Technical Standard, AE Specification C309.

[Papazoglou 2003] Papazoglou, M. P. (2003). Service-oriented computing: Concepts, characteristics and directions. In *Fourth International Conference on Web Information systems Engineering (WISE'03)*.

[Parr e Villars 2001] Parr, B. e Villars, R. (2001). Digital identity: The coming struggle for the future of the net. Boletim 24929, IDC.

[Rannenberg 2000] Rannenberg, K. (2000). Multilateral security a concept and examples for balanced security. In *Workshop on New security paradigms (NSPW'00)*, pages 151–162, New York, NY, USA. ACM Press.

[Rivest e Lampson 1996] Rivest, R. L. e Lampson, B. (1996). SDSI – A simple distributed security infrastructure. Presented at CRYPTO'96 Rumpsession.

[RSA 2002] RSA (2002). *PCKS#1 v2.1: RSA Cryptography Standard*. RSA Laboratories. `ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf`.

[RSS 2005] RSS (2005). *Really Simple Syndication*. `http://www.rssboard.org/rss-specification`.

[Russell e Gangeni 1991] Russell, D. e Gangeni, G. (1991). *Computer Security Basics*. O'Reilly Associates Inc.

[Santis et al. 1998] Santis, A. D., Crescenzo, G. D., e Persiono, G. (1998). Communication-efficient anonymous group identification. In *5th A.C.M. Conference on Computer and Communications Security (ACM CCS'98)*, pages 73–82, San Francisco, California, U.S.A.

[Shibboleth 2005] Shibboleth (2005). *Shibboleth Architecture*. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.

[Skogsrud et al. 2003] Skogsrud, H., Benatallah, B., e Casati, F. (2003). Modelo-driven trust negotiation for web services. In *IEEE Internet Computing*, pages 45–52. IEEE Computer Society.

[Skogsrud et al. 2004] Skogsrud, H., Benatallah, B., e Casati, F. (2004). Trust-serv: model-driven lifecycle management of trust negotiation policies for web services. In *WWW 2004*, pages 53–62. ACM.

[Sun 2002] Sun (2002). Java remote method invocation specification. Revision 1.8 Java 2 SDK.

[Vogels 2003] Vogels, W. (2003). Web services are not distributed objects. *Internet Computing*, 7(6):59–66.

[W3C 2001] W3C (2001). *Web Services Description Language 1.1*. W3C Working Group.

[W3C 2002] W3C (2002). *The Platform for Privacy Preferences 1.0 (P3P1) Specification*. W3C Recommendation. http://www.w3c.org/TR/P3P.

[W3C 2003] W3C (2003). *SOAP 1.2 – W3C Recommendation*. W3C. http://www.w3.org/TR/soap12.

[W3C 2004a] W3C (2004a). *Web Services Architecture*. W3C Working Group. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211.

[W3C 2004b] W3C (2004b). *Web Services Architecture Requirements*. W3C Working Group. http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211.

[Wangham et al. 2006] Wangham, M., da Silva Fraga, J., de Mello, E. R., e Milanez, J. (2006). Um modelo para o gerenciamento federado do spki/sdsi através do serviço xkms. In *VI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG'06)*, Santos, SP - Brasil.

[Wangham et al. 2005] Wangham, M. S., Mello, E., Rabelo, R., e da Silva Fraga, J. (2005). Provendo garantias de segurança para formação de organizações virtuais. In Gerrini, F. M., editor, *Gestão Avançada de Manufatura*, volume 2, pages 75–84. Editora Novos Talentos.

[Weerawarana et al. 2005] Weerawarana, S., Curbera, F., Leymann, F., Storey, T., e Ferguson, D. F. (2005). *Web Services Plataform Architecture*. Prentice Hall.

[Wege 2002] Wege, C. (2002). Portal server technology. *IEEE Internet Computing*, 6(3):73–77.

[Westbridge 2003] Westbridge (2003). *Securing and Managing XML Web Services – Guide to XML Web Services Security*. Westbridge Technology Inc.

[Winslett et al. 2002] Winslett, M., Yu, T., Seamons, K. E., Hess, A., Jacobson, J., Jarvis, R., Smith, B., e Yu, L. (2002). Negotiating trust on the web. In *IEEE Internet Computing*, number 6 in 6, pages 30–37. IEEE Computer Society.

[WS-Federation 2003a] WS-Federation (2003a). *Web Services Federation Language*. `http://msdn.microsoft.com/ws/2003/07/ws-federation`.

[WS-Federation 2003b] WS-Federation (2003b). *WS-Federation: Active Requestor Profile*. `ftp://www6.software.ibm.com/software/developer/library/ws-fedact.pdf`.

[WS-Federation 2003c] WS-Federation (2003c). *WS-Federation: Passive Requestor Profile*. `ftp://www6.software.ibm.com/software/developer/library/ws-fedpass.pdf`.

[WS-I 2005] WS-I (2005). *Basic Security Profile Version 1.0*. Web Services Interoperability Organization. `http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2005-08-29.html`.

[WS-Policy 2004] WS-Policy (2004). *Web Services Policy Framework*. `http://msdn.microsoft.com/ws/2004/09/policy/`.

[WS-PolicyAttachment 2004] WS-PolicyAttachment (2004). *Web Services Policy Attachment*. `http://msdn.microsoft.com/ws/2004/09/policyattachment`.

[WS-SecureConversation 2005] WS-SecureConversation (2005). *Web Services Secure Conversation Language*.

[WS-SecurityPolicy 2005] WS-SecurityPolicy (2005). *Web Services Security Policy Language*.

[WS-Trust 2005] WS-Trust (2005). *Web Services Trust Language (WS-Trust)*. `http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Trust.asp`.

[Wu 1998] Wu, T. (1998). The secure remote password protocol. In *Internet Society Network and Distributed System Security Symposium*, pages 97–111.

[Yavatkar et al. 2000] Yavatkar, R., Pendarakis, D., e Guerin, R. (2000). *A Framework for Policy-based Admission Control*. IETF RFC 2753.

[Zimmerman 1994] Zimmerman, P. (1994). *PGP User's Guide*. Massachusetts Institute of Technology.

Capítulo 2

Ataques e Mecanismos de Segurança em Redes Ad Hoc

Natalia C. Fernandes, Marcelo D. D. Moreira, Pedro B. Velloso,
Luís Henrique M. K. Costa e Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação – GTA*
COPPE-Poli – Universidade Federal do Rio de Janeiro

Abstract

Wireless ad hoc networks have specific vulnerabilities related to the wireless transmission, the lack of infrastructure, and the collaborative forwarding of messages. In ad hoc networks, besides the conventional attacks against wireless networks, collaborative routing allows new security threats, and the absence of infrastructure is an obstacle for creating simple and efficient security mechanisms. This work addresses the main attacks for ad hoc network, classifying them according to their consequences and to the aimed protocol layer. Then, the main security mechanisms used to prevent these attacks are analyzed, as well as ad hoc main specific secure protocols. The use of the trust paradigm to improve security is also mentioned, and a new proposal of trusting mechanism based on the evaluation of neighboring nodes behavior is presented.

Resumo

As redes ad hoc sem fio possuem vulnerabilidades específicas, associadas principalmente à transmissão pelo ar como meio de comunicação, à ausência de infra-estrutura e ao encaminhamento colaborativo das mensagens. Nas redes ad hoc, além dos ataques convencionais às redes sem fio, o roteamento colaborativo possibilita novas ameaças de segurança e a ausência de infra-estrutura dificulta a criação de mecanismos de defesas simples e eficientes. Este trabalho apresenta os principais ataques às redes ad hoc, classificando-os segundo os efeitos que causam e a camada de protocolos na qual eles atuam. São também apresentados e analisados os principais mecanismos de segurança utilizados para a proteção aos ataques, assim como os principais protocolos seguros específicos para redes ad hoc que foram propostos. O uso do paradigma de confiança para prover maior segurança também é abordado e apresenta-se uma nova proposta de mecanismo de confiança baseado na avaliação do comportamento dos nós vizinhos.

*Apoiado pelos recursos da CAPES, CNPq, FAPERJ, FINEP, FUJB, RNP e FUNTTEL.

2.1. Introdução

As redes ad hoc móveis (*Mobile Ad hoc NETworks - MANETs*) são constituídas por dispositivos móveis que utilizam comunicação sem fio. A principal característica dessas redes é a ausência de infra-estrutura, como pontos de acesso ou estações-base, existentes em outras redes locais sem fio ou ainda nas redes de telefonia celular. A comunicação entre nós que estão fora do alcance de transmissão do rádio é feita em múltiplos saltos através da colaboração de nós intermediários. Além disso, a topologia da rede pode mudar dinamicamente devido à mobilidade dos nós.

As redes ad hoc sem fio possuem como grande vantagem o baixo custo de instalação e facilidade de configuração. Por outro lado, o meio de comunicação sem fio, a ausência de infra-estrutura e o roteamento colaborativo em múltiplos saltos as tornam alvos potenciais de diversos tipos de ataques. Assim, a segurança é um ponto crucial das redes ad hoc.

A utilização do ar como meio de transmissão torna a rede susceptível a diversos ataques vão desde uma simples escuta clandestina (espionagem) passiva das mensagens até interferências ativas com a criação, modificação e destruição das mensagens. As redes cabeadas são consideradas mais seguras, pois um atacante tem maior dificuldade para obter um acesso ao meio físico e também para transpor as barreiras formadas pelos *firewalls*. Os ataques às redes sem fio, podem vir de várias direções e alvejar qualquer nó da rede, basta que o nó atacado esteja no alcance da transmissão do nó atacado. Dessa maneira, é possível que um nó malicioso tenha acesso a informações sigilosas, possa alterar mensagens em trânsito ou ainda tentar se passar por outros nós da rede. Portanto, o preço que se paga pelas facilidades oferecidas pela comunicação sem fio é a ausência de uma barreira de defesa clara. Assim, cada nó da rede deve estar preparado para lidar direta ou indiretamente com ações maliciosas.

Outro aspecto importante a ser considerado nas redes ad hoc é a ausência de centralização e de infra-estrutura. Portanto, não existem dispositivos dedicados a tarefas específicas da rede como, por exemplo, prover algumas funcionalidades básicas. Apesar de a descentralização ter como vantagem a robustez, devido a inexistência de pontos únicos de falha, a ausência de infra-estrutura dificulta a aplicação das técnicas convencionais de autorização de acesso e de distribuição de chaves. Isto dificulta a tarefa de distinguir os nós confiáveis dos nós não-confiáveis, pois nenhuma associação segura prévia pode ser assumida.

Devido a ausência de infra-estrutura, as redes ad hoc exigem a colaboração distribuída dos nós da rede para o encaminhamento das mensagens. Nas redes ad hoc, todos os nós participam do protocolo de roteamento, pois também desempenham a função de roteador. Além disso, estes nós roteadores estão sob o controle dos usuários da rede, e não de administradores. Isso possibilita a criação de novos ataques que visam as vulnerabilidades dos algoritmos cooperativos. Ou seja, as principais particularidades das redes ad hoc estão na camada de rede. Desta forma, os protocolos de roteamento das redes ad hoc devem ser robustos a novos tipos de ataques.

As redes ad hoc móveis introduzem outros obstáculos importantes à implementação de mecanismos de segurança devido às constantes alterações na topologia da rede.

Esta dinamicidade implica novos nós que se tornam vizinhos e antigos nós que deixam de ser vizinhos e pode até causar o particionamento da rede. Assim, os mecanismos de segurança devem se adaptar dinamicamente às mudanças na topologia da rede e ao movimento dos nós entrando e saindo da rede. Além disso, as redes ad hoc móveis são em geral compostas por dispositivos portáteis, portanto com restrições de energia, processamento e memória. Com isso, as MANETs estão sujeitas a diferentes ataques de negação de serviço que visam esgotar os recursos dos nós a fim de prejudicar o funcionamento da rede.

Desta forma, as redes ad hoc móveis possuem vulnerabilidades específicas ligadas principalmente ao meio de comunicação sem fio, à ausência de infra-estrutura e o roteamento colaborativo. A maior parte dos novos ataques concentra-se na camada redes e, conseqüentemente, também a maioria dos mecanismos de defesa específicos das redes ad hoc.

Este capítulo está organizado da seguinte forma. A Seção 2.2 revisa as ferramentas básicas utilizadas na implementação de segurança em redes de computadores. A Seção 2.3 descreve as principais formas de ataques às redes ad hoc móveis. A Seção 2.4 apresenta os principais mecanismos de segurança que podem ser utilizados para combater estes ataques, enquanto a Seção 2.5 apresenta os protocolos especialmente projetados para prover segurança em redes ad hoc. Finalmente, a Seção 2.7 analisa as tendências futuras na área de segurança em redes ad hoc.

2.2. Fundamentos de Segurança em Redes

Para o projeto de protocolos seguros, é necessário definir os objetivos que os mecanismos de segurança a serem implementados na rede devem buscar. Os requisitos de segurança clássicos que devem ser observados são a autenticação, a confidencialidade, a integridade, o não-repúdio e a disponibilidade. A autenticação garante que uma dada entidade é realmente quem ela diz ser, enquanto que o não-repúdio impede que o emissor de uma mensagem negue a sua autoria. A confidencialidade garante o sigilo das informações trocadas por dois nós e a integridade permite afirmar que as informações recebidas por um nó não foram alteradas durante o trânsito ao longo da rede. A disponibilidade trata de garantir que os recursos da rede estarão disponíveis quando forem necessários.

A criptografia é uma ferramenta fundamental para prover segurança, pois por meio dela, é possível atender a todos os requisitos clássicos. A maioria dos ataques a redes poderia ser solucionada pela utilização de um mecanismo criptográfico seguro.

Tradicionalmente, a criptografia é separada em dois ramos: simétrica e assimétrica. A criptografia simétrica é caracterizada pela existência de um segredo, chamado de chave secreta, compartilhado entre os nós que desejam se comunicar. Esta chave é utilizada em operações que alteram os dados a transportar, enviando um texto criptografado ao invés de um texto em aberto. As principais operações realizadas pelos algoritmos simétricos são o ou-exclusivo, a troca de colunas, a troca de linhas, a permutação, a rotação e a expansão, que são operações de baixo custo computacional. Apesar de serem simples, as combinações dessas operações devem ser capazes de tornar difícil a descoberta da mensagem para quem não possui a chave secreta. Por essa razão, a eficiência desses algoritmos é medida pelo seu custo computacional e pela capacidade de modificar a

saída dada uma pequena mudança na entrada. Os algoritmos simétricos mais conhecidos são o DES [National Bureau of Standards, 1977] e o AES [Daemen e Rijmen, 2002]. Na criptografia assimétrica existem duas chaves, a chave pública e privada. A chave pública deve ser distribuída aos membros da rede, enquanto que a privada deve ser mantida em segredo pelo nó. Esse tipo de criptografia possui maior custo computacional que a simétrica, por fazer uso de operações como o logaritmo discreto, curva elíptica e fatoração de inteiros, aliadas as considerações de segurança da Teoria dos Números. O objetivo principal é que, a partir de uma das chaves, não seja possível encontrar a outra, o que é obtido quando se usa para o cálculo funções que são simples de calcular, mas quase impossíveis de se reverter computacionalmente. Outras funcionalidades, como a distribuição de chaves de forma segura e a assinatura de mensagens são possíveis com o uso de criptografia assimétrica. Os algoritmos mais conhecidos são o RSA [Kaliski e Staddon, 1998], o Diffie-Hellman [Rescorla, 1999], que utilizam números primos entre si muito grandes para gerar as suas chaves, e mais recentemente a Criptografia de Curva Elíptica (*Elliptic Curve Cryptography* (ECC)) [IEEE, 2000], considerada a mais segura.

Em redes cabeadas, é comum utilizar as características dos dois tipos de criptografia para garantir uma comunicação segura, o que é conhecido como criptografia híbrida. Primeiramente, é trocado um segredo entre os nós por intermédio das chaves públicas. Este segredo servirá como chave secreta para criptografar a comunicação posterior usando criptografia simétrica, de menor custo computacional. A Figura 2.1 mostra como funciona a criptografia híbrida. Primeiramente, ambos os nós trocam suas chaves públicas. Em seguida, o nó A gera uma chave secreta, a criptografa com chave pública de B e a envia. O nó B, então, decriptografa a mensagem com sua chave privada e gera uma mensagem contendo a chave secreta, criptografada com a chave pública de A, para confirmar que conseguiu obter o segredo. É importante notar que um esquema como esse não é suficiente para garantir a autenticação e confiabilidade das mensagens, pois um nó malicioso poderia realizar o Ataque do Homem do Meio (*Man in the Middle Attack*), forjando a comunicação para os dois nós.

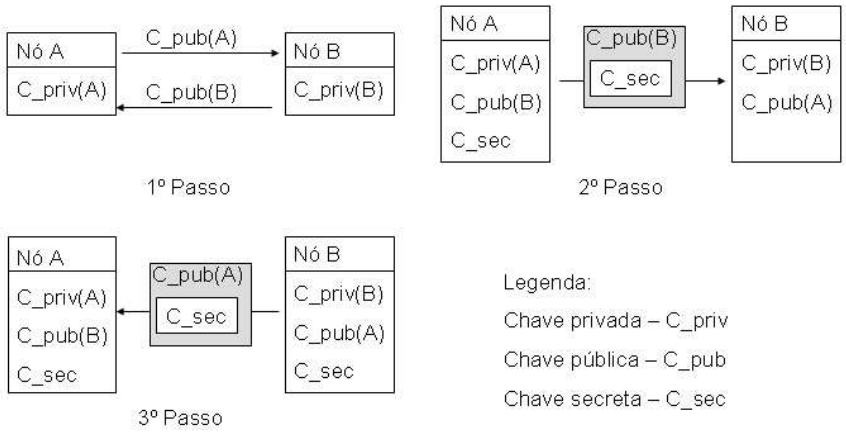


Figura 2.1. Criptografia híbrida.

O uso de uma Infra-Estrutura de Chave Pública (*Public-Key Infrastructure*-PKI) e assinatura digital permite solucionar o problema de interceptação, além de garantir a

autenticação e o não-repúdio na rede. No PKI, existe uma terceira entidade, chamada autoridade certificadora (AC), capaz de garantir a quem pertence realmente uma chave pública, através da emissão, validação e revogação de certificados. Um certificado deve conter a identificação e a chave pública do nó, criptografados com a chave privada da AC. A assinatura digital será feita criptografando a mensagem com a chave privada do emissor e enviando junto com ela um certificado emitido pela AC para aquele emissor. Assim, ao receber a mensagem, o destino decriptografa o certificado com a chave pública da AC, que deve ser conhecida por todos os nós da rede, obtendo a chave pública do emissor, que será a única chave capaz de decriptografar a mensagem. Assim, é possível garantir a autenticação e o não-repúdio, pois a chave privada deve ser mantida sempre em segredo pelos nós. É importante ressaltar, que apesar dessas características, tal mecanismo não garante a privacidade dos dados, o que poderia ser feito criptografando todo o pacote com a chave pública do receptor ou ainda com uma chave secreta obtida por criptografia híbrida.

Uma outra entidade importante para o funcionamento do PKI é a Autoridade Registradora, que realiza o cadastro dos usuários que desejam ser certificados. Um usuário sem registro não pode assinar mensagens, pois a AC não será capaz de emitir certificados para ele.

Uma outra ferramenta importante que pode ser utilizada para garantir integridade dos dados são as funções *hash*. Uma função *hash* é definida como uma função H que mapeia uma seqüência de bits de tamanho arbitrário em uma de tamanho fixo. O conceito de função *hash* unidirecional foi introduzido em [Diffie e Hellman, 1976]. De maneira informal, uma função *hash* unidirecional deve ser simples de calcular, porém computacionalmente impossível de ser invertida. Para o uso em criptografia simétrica, ainda se exige que a função *hash* seja resistente a colisões, ou seja, é computacionalmente impossível encontrar duas seqüências distintas x e y tais que $H(x) = H(y)$. Os famosos algoritmos MD5 [Rivest, 1992] e SHA-1 [National Institute of Standards, 2000] foram projetados para possuírem essas propriedades. Além dessas propriedades básicas, as funções *hash* criptográficas geralmente possuem propriedades aleatórias, como a uniformidade dos valores de saída ao longo do conjunto imagem, a independência entre a entrada e a saída, a impossibilidade de inferência da saída ainda que partes da seqüência de entrada sejam conhecidas, etc. Devido a essas propriedades, essas funções também são conhecidas como funções de espalhamento.

A disponibilidade não é tratada diretamente com o uso de criptografia. De fato, para garantir esse conceito, é necessário evitar ataques de negação de serviço. A autenticação dos usuários permite que seja dado acesso à rede ou aos serviços disponíveis apenas aos nós autorizados, o que já ajuda a reduzir o acesso dos atacantes à rede.

2.2.1. Criptografia Simétrica x Assimétrica: Vantagens e Desvantagens

A criptografia assimétrica trouxe inúmeros avanços, resolvendo questões como a autenticação e o não-repúdio através do PKI, que eram problemas em aberto para algoritmos simétricos. Mesmo os mecanismos que utilizam autenticação com chaves secretas e funções *hash* não são capazes de garantir quem foi o gerador da mensagem, pois o segredo é conhecido por todos os nós que fazem parte da mesma comunicação.

Outro ponto é que para se manter uma comunicação segura utilizando apenas criptografia simétrica, é necessário que cada par de nós possua uma chave para comunicação. Dessa forma, em uma rede com n nós, seriam necessários $(n*(n-1))/2$ pares de chaves em cada nó. Utilizando criptografia assimétrica, o número de valores armazenados cai para n. Cabe ressaltar que o tamanho da chave utilizada com a criptografia assimétrica tradicional é muito superior ao tamanho da chave da criptografia simétrica, o que pode ser um problema para nós com restrições de memória. Neste caso, é aconselhado o uso de criptografia elíptica, pois suas chaves são menores que a dos demais algoritmos assimétricos devido à alta complexidade da inversão da função elíptica.

Apesar de todas as vantagens do PKI, o seu uso em redes ad hoc não é simples. A primeira razão são as fortes restrições de processamento e memória dos nós, o que torna muito complicado o cálculo para criptografar e decriptografar mensagens. Mesmo em redes com mais recursos, como as cabeadas, não se utiliza criptografia assimétrica para criptografar todas as informações, devido ao seu alto custo. Em redes de sensores sem fio, essa restrição é ainda maior, pois seus recursos são muito reduzidos, devendo ser poupados ao máximo. Medidas com sensores MICA mostraram que estes se tornam entre 100 e 1000 vezes mais lentos utilizando criptografia assimétrica [Kulkarnia et al., 2006]. Medidas utilizando *hardwares* específicos para os algoritmos RSA e DES mostraram um desempenho 1500 vezes mais lento do RSA [Nichols e Lekkas, 2002]. Os recursos das redes ad hoc costumam ser menos restritos que os das redes de sensores, mas, ainda assim, o uso de criptografia assimétrica deve ser evitado ao máximo, tentando buscar formas de equilibrar suas vantagens com a facilidade da criptografia simétrica.

Outro ponto importante a ser observado é que a utilização de PKI exige uma terceira entidade para certificar a comunicação, o que é, geralmente, feito por um servidor. No entanto, redes ad hoc não possuem pontos centrais. Assim, um outro problema a ser solucionado é como distribuir as tarefas da AC pelos nós da rede e como fazer o registro das chaves públicas em uma rede que deve ser auto-configurável.

2.3. Principais Formas de Ataques

Os ataques a redes ad hoc móveis podem ser divididos em passivos ou ativos [Murthy e Mano, 2004]. Os ataques passivos não afetam a operação da rede, sendo caracterizados pela espionagem dos dados sem alterá-los. Por outro lado, os ataques ativos são aqueles em que o atacante cria, altera, descarta ou inviabiliza o uso dados em trânsito. Os ataques ativos são os mais numerosos, podendo atuar em diferentes camadas do modelo OSI.

Os atacantes podem ser classificados como internos ou externos. Atacantes internos são aqueles que conseguem de alguma forma se passar por membros da rede, enquanto que os externos são aqueles que influenciam, mas não participam da rede. De fato, a eficiência e as possibilidades de ataques variam de acordo com o acesso que o atacante tem à rede. Se de alguma forma ele conseguir obter chaves ou for incluído na lista de vizinhos válidos, passando a ser um atacante interno, poderá causar mais problemas.

2.3.1. Ataques Passivos

Nos ataques passivos, o atacante não interfere no funcionamento da rede, mas pode escutá-la e analisar o seu tráfego. O atacante tem acesso à informação, porém não a altera ou destrói. Os ataques passivos são de difícil detecção por não influírem no comportamento da rede.

Espionagem

A espionagem (*eavesdropping*[2]) caracteriza-se pela escuta do tráfego sem modificação dos dados. O atacante aproveita-se do meio inseguro apenas para roubar informações.

Quando o atacante utiliza o tráfego observado para aprender a localização dos recursos críticos da rede, o ataque é chamado de Revelação de Informações Críticas (*Homing*/*Information Disclosure*) [Wood e Stankovic, 2002]. Uma vez que esses pontos são encontrados, essas informações são passadas para outros nós maliciosos que poderão realizar ataques ativos. Protocolos de roteamento que utilizam encaminhamento geográfico são ainda mais expostos a esse ataque, pois a posição exata dos nós críticos é passada para os atacantes ativos, facilitando a localização e ataque ao nó.

A proteção contra espionagem costuma ficar sob responsabilidade das camadas superiores, que, em geral, cuidam do sigilo das informações. No entanto, como a espionagem de informações de roteamento pode levar à exposição da topologia da rede para o atacante, o sigilo através da criptografia passa a ser uma necessidade do roteamento [Karlof e Wagner, 2003].

2.3.2. Ataques Ativos

Os ataques ativos, em sua maior parte, têm como alvo a vulnerabilidade de alguma camada específica do modelo OSI. Esta seção considera cada uma das camadas, descrevendo os seus principais ataques.

2.3.2.1. Camada Física

Na camada física estão os ataques de tratamento mais difícil, pois eles exercem uma utilização indevida do meio sem fio por dispositivos que não participam da rede. Os ataques são mais fáceis que nas redes cabeadas, pois não há a necessidade de conexão "física" ao meio de comunicação. Cabe ressaltar que os ataques da camada física são característicos do meio físico utilizado, e não específicos das redes ad hoc.

Interferência Contínua

Esse é um ataque muito conhecido, que consiste em sujar continuamente com interferências a freqüência de comunicação da rede sem fio. Um adversário pode impedir totalmente o funcionamento de uma rede com N nós utilizando k nós maliciosos distribuídos randomicamente, onde $k \ll N$ [Wood e Stankovic, 2002]. Para redes com uma única

[2] A origem do termo vem da expressão "*hide out in the eavesdrop of a house*" [Wikipedia - The Free Encyclopedia, 2006], que tem como similar em português o "ouvir atrás da porta".

freqüência de comunicação, esse é um ataque simples e muito efetivo.

A detecção da interferência contínua também é simples, pois basta que o nó atacado ou algum nó próximo à área de interferência observe que um nível constante de energia, e não a falta de uma resposta, impede a comunicação. Se o nó está no alcance da interferência, mas não totalmente imerso nela, ele pode comunicar ao resto da rede que evite rotas por aquela área. Embora o resultado do ponto de vista da aplicação seja o mesmo devido à perda de pacotes ou devido à interferência, esta é mais grave por impedir que o nó envie ou receba pacotes, ou mesmo que se faça uma notificação para algum nó de monitoramento.

A interferência é um ataque de difícil combate. Entre algumas das propostas estão o espalhamento de espectro, descrito mais adiante, e o descobrimento de novas rotas isolando a área de interferência. Para os nós que estão dentro dessa área só resta dormir e checar periodicamente até que a interferência acabe.

Interferência Esporádica ou Exaustão por Interferência

Este tipo de interferência pode ser ainda pior que a contínua. Ela consiste em gerar interferências por curtos períodos, o que já é suficiente para impedir a comunicação. A grande eficiência dos ataques com interferência esporádica se deve a eles poderem causar grandes prejuízos no consumo de bateria do nó atacado, que deve fazer retransmissões, e ao mesmo tempo terem um custo mínimo para o nó atacante, que realiza apenas pequenas transmissões de tempos em tempos. Além disso, como o ataque é esporádico, é de mais difícil detecção. A melhor forma de minimizar os efeitos desse ataque é através do método de espalhamento de espectro.

Método de Espalhamento de Espectro

Algumas tecnologias de transmissão sem fio impõem um nível de dificuldade a mais a ataques da camada física [Stallings, 2004]. Um exemplo são as técnicas de modulação baseadas em um espectro de freqüências espalhado (*Spread Spectrum*). Técnicas de espalhamento de espectro como DSSS (*Direct Sequence Spread Spectrum*) e FHSS (*Frequency Hopping Spread Spectrum*) foram projetadas com o objetivo de maior resistência a interferências de fontes de faixa de freqüências estreita. A idéia básica é utilizar uma largura de banda maior do que a que é realmente necessária para transmitir dados a uma velocidade específica. No FHSS, o espalhamento de espectro é obtido saltando-se continuamente de uma freqüência de portadora para outra, minimizando, desta forma, interferências. Se um atacante não conhecer qual a seqüência em que as portadoras são utilizadas, ele não é capaz de obter acesso ou sujar a informação sendo transmitida. Já o DSSS literalmente aumenta a taxa de dados de um sinal, mapeando cada bit de informação em uma cadeia de bits a transmitir, chamada de seqüência de *chips*. O efeito é espalhar um bit de informação no tempo, o que aumenta a robustez a interferências. Para cada valor binário 0 ou 1 a transmitir, uma seqüência de chips é transmitida. Os códigos utilizados como seqüência de chips para transmitir os bits de dados fornecem a segurança inerente do DSSS.

É importante notar que as técnicas de modulação por espalhamento de espectro possuem um nível de segurança inerente, mas não fornecem qualquer proteção criptográfica. A segurança vem apenas do fato de manter os códigos (seqüência de portadoras ou

de chips) secretos. Uma vez que estes códigos não são protegidos, e, normalmente, são bem conhecidos ou fáceis de descobrir, o nível de segurança fornecido é mínimo. Assim, estes códigos provêem pouca ou nenhuma proteção contra ataques de negação de serviço na camada física através das interferências, embora ainda seja o melhor método para preveni-las.

2.3.2.2. Camada Enlace

A camada enlace é a responsável pela transmissão confiável de dados ponto a ponto. Desta forma, nesta camada, os ataques visam à retransmissão de quadros, a prioridade de mensagens e os códigos corretores de erro, características especificas do IEEE 802.11, a tecnologia mais usada nas redes ad hoc. A solução para esses ataques consiste de uma implementação mais robusta do protocolo de enlace, prevendo o comportamento malicioso.

Exaustão de Bateria por Colisão

Neste ataque, o nó malicioso tem como objetivo consumir a bateria do nó atacado gerando retransmissões continuamente, através de uma implementação maliciosa da camada de enlace. As retransmissões são geradas por recursos como, ao ouvir o início de uma transmissão, gerar uma colisão tardia no fim do quadro. No caso desta colisão intencional ocorrer com um pacote de ACK, isso poderia acarretar em um aumento exponencial do *back-off* em alguns protocolos MAC [Wood e Stankovic, 2002]. O uso repetido desse método culmina na exaustão da bateria do nó atacado, pois a transmissão é uma operação muito custosa, e que só deve ser feita quando estritamente necessário.

Uma variação deste ataque é conhecida como o ataque da interrogação. Neste, o nó atacante, que é um nó suicida, explora características da interação de protocolos da subcamada MAC que utilizam o *Request To Send* (RTS), *Clear To Send* (CTS) e mensagens de dado e ACK. O nó malicioso faz pedidos de alocação do canal com RTS repetidos, forçando inúmeras respostas CTS da vítima, levando ambos à morte.

A exaustão de bateria é de difícil tratamento, pois necessariamente a camada de enlace conta com certa confiança entre os nós participantes. Um nó malicioso poderia negar acesso ao canal repetidamente, impedindo o funcionamento da rede sem ter um grande gasto de energia. Soluções para essa variação do ataque são obtidas na reformulação dos protocolos, tornando-os mais robustos a comportamentos inadequados [Wood e Stankovic, 2002].

Alteração de ACK

A maioria dos algoritmos de roteamento confia de alguma forma em ACKs da camada de enlace. Uma vez que o meio é o ar, um atacante pode forjar esses ACKs com a finalidade de enganar o receptor a respeito de dados como a qualidade do canal, ou ainda dizer que um nó que já desativado ainda está ativo. Isso implicaria na escolha de rotas por enlaces inapropriados ou passando por nós que não participam mais da rede [Karlof e Wagner, 2003].

2.3.2.3. Camada Transporte

Os ataques à camada de transporte [Wood e Stankovic, 2002], visam vulnerabilidades do TCP, na fase de sincronização e na retransmissão de pacotes.

Inundação de *Sync*

Para realizar a comunicação utilizando o TCP, é necessário um período de tempo para o estabelecimento da conexão. Cada processo de conexão ocupa um espaço de memória no nó até que seja concluído. Este ataque visa explorar essa característica, gerando vários pedidos de conexão para a vítima. Cada um desses pedidos, que nunca é completado, provoca a alocação de mais recursos, até o momento que acontece um estouro de memória. A limitação do número de conexões pode impedir que o ataque consiga a exaustão de recursos, mas não pode impedir que conexões reais com nós legítimos sejam perdidas devido a inúmeros pedidos de conexões falsos na fila. Outra possível solução é o uso de desafios para diminuir a velocidade que o nó malicioso gera os pedidos de conexão.

Dessincronização

Neste ataque um terceiro nó influi em uma conexão entre dois nós legítimos. O atacante envia mensagens falsas pedindo retransmissões, a partir da observação do número de seqüência que está sendo utilizado na comunicação. Para o funcionamento efetivo do ataque, é necessário que o atacante mantenha um controle preciso do momento de envio das mensagens, para evitar que os nós legítimos troquem informações úteis. O tratamento para esse ataque exige autenticação e criptografia das mensagens.

Seqüestro de Sessão (*Session Hijacking*)

O seqüestro de sessão, descrito em [Murthy e Mano, 2004], é um ataque onde o adversário toma o controle de uma sessão entre dois nós. Uma vez que a maioria dos processos de autenticação só é feita no início da sessão, após essa fase, o atacante pode se passar por uma das extremidades, se comunicando com o outro nó como se fosse o nó legítimo. Para evitá-lo deve-se utilizar criptografia ou assinatura digital em todas as mensagens trocadas.

2.3.2.4. Camada Rede

Na camada rede acontece a maior parte dos ataques, devido tanto às características críticas da rede, quanto às vulnerabilidades dos protocolos de roteamento. Muitos dos ataques a essa camada possuem soluções eficientes, com métodos preventivos capazes de reduzir bastante a interferência do atacante na rede.

Ataque Bizantino

Neste ataque, geralmente ligado a problemas de tolerância a falhas, um ou mais nós maliciosos trabalham em conluio para gerar problemas como *loops* de roteamento, pacotes de roteamento falsos, escolha de caminhos não-ótimos, entre outros, utilizando mensagens de controle dos protocolos que estão sendo utilizados. Murthy cita esse problema com o nome de mensagens de roteamento alteradas, restringindo-o apenas aos

problemas de roteamento. Além disso, os nós também podem executar um encaminhamento seletivo [Murthy e Mano, 2004]. Esse tipo de ataque é de difícil detecção, pois para os nós comuns, o funcionamento estará correto, embora, de fato, esteja apresentando anomalias.

O nome ataque bizantino tem uma origem curiosa. A idéia é baseada no problema dos generais [Lamport et al., 1982] bizantinos, distribuídos em campo com suas tropas para organizar o ataque à cidade inimiga. A comunicação entre eles é feita apenas por mensagens e isso deve ser suficiente para organizar o ataque. No entanto, um ou mais generais podem ser traidores tentando confundir os demais, o que gera a necessidade de um algoritmo capaz de garantir que os generais leais conseguirão chegar a um acordo.

Tem-se como objetivos que todos os generais leais devem decidir pelo mesmo plano de ação e um pequeno número de generais maliciosos não deve levar os generais leais a adotar um plano ruim. Para satisfazer estes dois objetivos, é necessário que todos os generais leais recebam a mesma informação, e se um general é leal, então sua informação deve ser utilizada por todos os generais leais. Como ambas as condições levam ao mesmo ponto sobre como um general envia sua ordem, é possível simplificar o problema a um general e dois tenentes que devem receber a sua ordem. No caso de apenas um general e um tenente, a solução é trivial, pois a comunicação é direta e não existem mais versões sobre o que foi dito. Assim, o problema acontece a partir de três generais. Por simplificação, a mensagem só pode ser de atacar ou bater em retirada. O problema pode ser caracterizado como na Figura 2.2, onde há um general e dois tenentes. No primeiro caso, temos um tenente traidor, e o tenente leal receberá duas mensagens, uma do general mandando atacar e uma do traidor dizendo que as ordens do general são de bater em retirada. No segundo caso, mostrado na Figura 2.3, o general é traidor, e o mesmo grupo de informações chega ao tenente leal, de forma que, com três entidades, sendo uma traidora, não há solução. Generalizando o problema, se cada entidade representasse m generais/tenentes, ainda assim, não seria possível resolver o problema, pois seria necessário mais algum testemunho para concluir qual é a informação que deve ser usada. Isto leva a regra de que são necessários pelo menos $3m+1$ generais, sendo m o número de generais mentirosos, para que os generais leais cheguem a uma solução única e verdadeira. A solução matemática para o problema é descrita em [Pease et al., 1980].

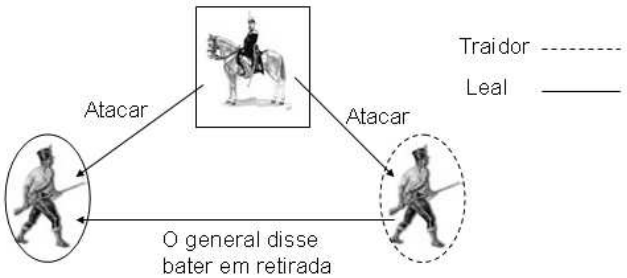


Figura 2.2. Ataque Bizantino onde um tenente é traidor.

As soluções para o ataque bizantino podem ser a assinatura digital, o uso de múltiplos caminhos e ainda a autorização, mecanismos que serão descritos na Seção 2.4.

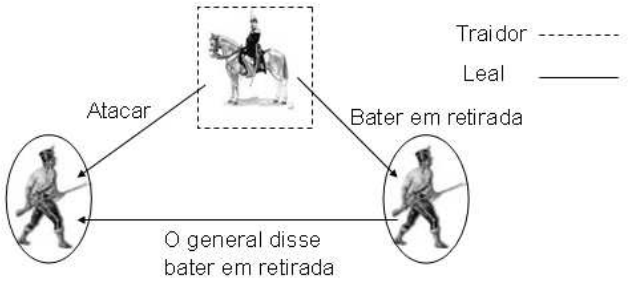


Figura 2.3. Ataque Bizantino onde o general é traidor.

Estouro (*Overflow*) da Tabela de Roteamento

Este ataque se baseia no fato de os protocolos de roteamento ad hoc pró-ativos armazenarem todas as rotas anunciadas pelos seus vizinhos. Nestes protocolos, o nó armazena em sua tabela de roteamento todas as mensagens de rota que recebe periodicamente. A estratégia deste ataque é anunciar diversas rotas para nós inexistentes, de modo a aumentar progressivamente o tamanho da tabela de roteamento, até que ela estoure e o nó não possa mais armazenar as rotas reais. Os protocolos reativos que armazenam diversas rotas para um mesmo destino também estão expostos a esse tipo de ataque, pois o nó malicioso poderia enviar rotas passando pelos nós inexistentes.

Esse ataque é grave no caso de redes ad hoc que possuem nós com escassos recursos, onde tanto o gasto de energia com a recepção de um número excessivo de mensagens, quanto o estouro de *buffer* são cruciais. Para preveni-lo, deve-se limitar o número máximo de rotas nas tabelas de roteamento, além de só aceitar entradas de nós autenticados.

Replicação de Pacotes

Este ataque possui dois objetivos principais: ocupar o meio de transmissão e levar os nós à exaustão. Esta é a versão de nível roteamento do ataque de exaustão de bateria da camada enlace. Para conseguir seus objetivos, o atacante envia réplicas de pacotes de roteamento antigos.

As soluções propostas para esse tipo de ataque são limitadas, pois ainda que se deduza que o mesmo nó envia mensagens de roteamento antigas, através da observação do número de seqüência, o máximo que poderia ser feito é retirar aquele nó das rotas, mas nada poderia impedi-lo de continuar as suas réplicas, assim como acontece no ataque da interferência na camada física.

Envenenamento de *Cache*

Muito semelhante ao ataque do estouro da tabela de roteamento, este ataque visa envenenar o *cache* de roteamento fazendo anúncios falsos de rotas para nós reais. Esse ataque se aproveita em especial de protocolos sob demanda, como o *Ad Hoc On demand Distance Vector* (AODV) [Perkins et al., 2003], que mantêm rotas para nós que foram aprendidas em um passado recente. Estes protocolos estão mais susceptíveis que os protocolos pró-ativos pelo fato de anunciarem para onde desejam mandar o pacote sempre que não tem rota, permitindo ao nó invasor anunciar a rota falsa antes do nó confiável. No caso dos protocolos pró-ativos, esse ataque também é possível, mas seria necessá-

rio mandar anúncios de rota falsos em todas as rodadas de atualização, para todos os possíveis destinos. Para evitar esse tipo de comportamento dos nós maliciosos, deve-se utilizar sistemas de confiabilidade baseados em monitoramento e punição. Outra forma de identificação deste ataque é a utilização de pacotes de investigação 2.4.

Ataque da Pressa (*Rushing Attack*)

Este ataque permite a formação de um buraco negro, se aplicando a protocolos de roteamento sob demanda que guardam apenas uma rota para cada destino em sua tabela. Ao receber um *Route Request*, o atacante o envia de forma mais rápida aos demais nós da rede, de forma que todas as respostas passem por ele. Assim, como ele será o primeiro a responder, as demais respostas provenientes dos outros vizinhos serão descartadas. Desta forma, as rotas sempre passariam pelo nó malicioso, tornando a rede vulnerável.

A detecção de tal ataque é difícil, dado que, para o protocolo de roteamento, tudo está transcorrendo de forma normal. A solução seria tentar identificar os métodos para conseguir enviar a mensagem de forma mais rápida, como por exemplo, um abuso do protocolo de enlace. De fato, existem vários métodos para acelerar o envio da mensagem, e que não exigem muitos recursos do nó malicioso [Hu et al., 2003b]. Em geral, os protocolos MAC (*Medium Access Control*) impõem atrasos entre o momento no qual o pacote é recebido e a transmissão. Um exemplo são os protocolos MAC que utilizam divisão de tempo para acesso ao meio, onde o nó precisa esperar até a sua vez de transmitir, ou ainda os protocolos de acesso ao meio que utilizam *Carrier-Sense Multiple Access* (CSMA), onde é utilizado um *backoff* para evitar colisões. Outro espaço de tempo que também pode ser burlado por um nó malicioso é aquele que pode ser usado pelo roteamento entre a recepção de um RREQ e o encaminhamento do mesmo, para evitar colisões. Assim, um atacante que deseja enviar um pacote mais rápido que outros nós pode simplesmente ignorar um ou mais destes tempos de espera. Uma outra forma de executar esse ataque seria, por exemplo, provocar a criação de filas nas interfaces dos nós vizinhos, de forma a que o nó malicioso repasse o pacote enquanto seus vizinhos estão processando os pacotes das filas. Esse tipo de atitude do nó malicioso é mais fácil em sistemas que utilizam autenticação da mensagem, pois ele poderia gerar várias mensagens com defeito, levando os vizinhos a perder tempo verificando as mensagens. No caso de autenticação por chave pública esse problema é ainda maior, devido ao alto custo computacional para realizar a verificação. Outros métodos para transmitir os pacotes mais rápido que os vizinhos também são possíveis, como a utilização de uma potência de transmissão maior, ou, ainda, através da utilização de um túnel de minhoca (*Wormhole*).

A melhor solução para este ataque é o uso de múltiplas rotas disjuntas ou trançadas, que garantiriam que mesmo que o atacante atraísse o trafego para si em uma das rotas, as outras permaneceriam seguras.

Direcionamento Falso (*Misdirection*)

O direcionamento falso consiste na fabricação de mensagens visando gerar negação de serviço para um determinado nó. Assim, são enviadas mensagens de modo a direcionar tráfego para uma determinada região que se deseja atacar. Na versão da Internet desse ataque, conhecida como Ataque *Smurf*, o atacante forja pacotes *echo*, colocando como emissor o nó vítima, que irá receber inúmeros *echo-backs*.

Esse ataque pode ser realizado por mecanismos além do uso de *echos*. O caso do protocolo *Dynamic Source Routing* (DSR) [Johnson e Maltz, 1996] é um exemplo, onde o atacante pode responder às requisições de rotas com caminhos falsos que incluem o nó que se deseja atacar.

Inundação de *Hellos*

Este ataque inicialmente foi considerado para redes de sensores. De fato, ele também se aplica as redes ad hoc, desde que o atacante possua uma potência de transmissão maior que os demais nós da rede. A Inundação de *Hello* só se aplica a protocolos que utilizam a mensagem de *hello* para identificação dos vizinhos, embora não façam a verificação de bidirecionalidade do enlace.

Para realizar o ataque, o nó malicioso envia *hellos* com alta potência, informando que o nó possui enlaces muito bons com determinados destinos. Assim, ele atinge um grande número de nós, que por terem ouvido a mensagem, o colocam na sua lista de vizinhos e podem escolhê-lo para encaminhamento de dados. No entanto, apesar dos nós ouvirem o nó malicioso, o nó malicioso não é capaz de escutá-los, de forma que vários nós da rede irão apontar suas rotas de encaminhamento para um nó inalcançável, como ilustrado na Figura 2.4.

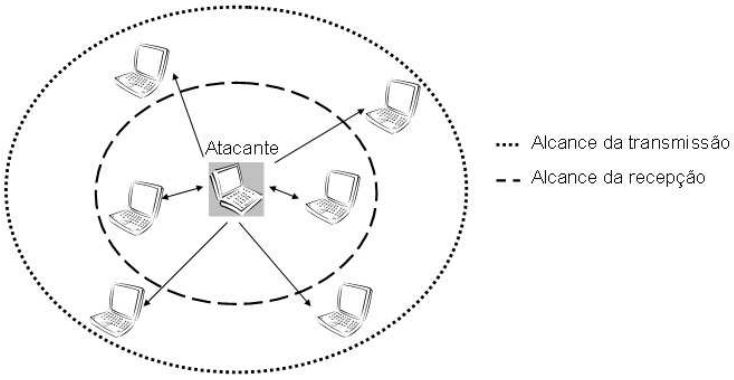


Figura 2.4. Inundação de *Hellos*.

A inundação de *hello* funciona melhor quando é realizada em conjunto com uma revelação de informações críticas, de forma a permitir que o nó malicioso descubra quais rotas ele deve tentar interceptar.

Esse ataque pode ser evitado pela verificação de bidirecionalidade do *link*. Cabe observar que apesar de ser uma solução simples, muitos protocolos de roteamento não a aplicam, assumindo que os enlaces são bidirecionais.

Ganância (*Greed*)

A Ganância é caracterizada quando um nó dá uma prioridade injusta às suas mensagens. O nome representa o fato de o nó, apesar de não prejudicar o funcionamento da rede de forma explicita, atrapalhar por tentar ter sempre a maior fatia de tempo para transmissão.

A detecção de um nó ganancioso é difícil, pois um nó que já esteja com falhas de bateria apresenta um comportamento semelhante. Assim, a melhor forma de evitar esse

ataque é a utilização de redundâncias, garantindo que, ainda que pacotes sejam perdidos por demora ao enviar, uma segunda rota poderá garantir a sua entrega.

Encaminhamento Seletivo ou Buraco Cinza

Uma das características principais das redes ad hoc é a confiança nos vizinhos para o encaminhamento de dados. No entanto, um vizinho malicioso pode encaminhar apenas alguns pacotes. Neste caso não se deseja prejudicar todos os nós, ou uma determinada área da rede. O nó malicioso pode escolher não passar alguma ou todas as mensagens pra um determinado nó alvo, ou pode optar por passar as mensagens de roteamento, mas impedir a transmissão de dados, impedindo o funcionamento da aplicação.

Um tipo especial de encaminhamento seletivo é chamado de egoísmo, no qual o nó não encaminha nenhuma mensagem dos vizinhos, passando apenas as suas próprias. O egoísmo nem sempre é um ataque, podendo ser uma escolha de um nó por um comportamento não cooperativo. Tal decisão pode ser tomada, por exemplo, em momentos nos quais o nó deseja se poupar.

Diferentemente de um buraco negro, esse ataque é de difícil detecção, assim como a Ganância, pois nós com pouca energia e perdas de pacotes normais podem gerar um quadro muito semelhante.

Buraco Negro

Este é caso extremo do Encaminhamento Seletivo, onde todos os pacotes são atraídos até o nó e são descartados. Este ataque, dependendo da posição do atacante, pode ter um efeito totalmente destrutivo na rede, impedindo todo o seu funcionamento. Por outro lado, ao contrário do Encaminhamento Seletivo, sua detecção é fácil, pois em muito pouco tempo todo um ramo da rede deixará de funcionar.

Uma segunda conseqüência do buraco negro, é que, como ele atrái muito tráfego em sua direção, ele acaba consumindo os recursos dos nós à sua volta, tanto em termos do meio, que fica excessivamente ocupado, assim como em termos de recursos dos nós. No caso de exaustão desses nós, o resultado poderia acabar particionando a rede.

Assim como o Encaminhamento Seletivo, uma premissa de funcionamento do ataque é que o nó malicioso se torne atrativo durante a escolha de rotas. Para tanto, vários métodos podem ser utilizados.

A solução mais simples para os ataques de descarte de pacotes é a utilização de múltiplas rotas. Outros métodos que ajudam a detectar e prevenir esse comportamento são a investigação e a autorização.

Túnel de Minhoca (*Wormholes*)

No ataque do túnel de minhoca, dois atacantes criam um túnel de comunicação por um enlace de baixa latência, através do qual irão trocar informações da rede, replicando-as do outro lado do túnel, de forma a tornar excepcionalmente atrativo o enlace formado pelos dois. Assim, os nós maliciosos podem convencer nós da rede que eles podem se comunicar com determinado destino por apenas um salto, ao invés de utilizar os vários saltos que existem realmente entre o nó e o destino.

O túnel é um canal seguro e de baixa latência entre os dois nós maliciosos, que per-

mite que os nós vizinhos sejam incapazes de perceber que o ataque está sendo realizado. Uma forma simples de obter esse resultado é utilizar uma conexão por fio entre os dois nós, fazendo uma transmissão mais rápida que o encaminhamento por múltiplos saltos [Hu et al., 2003b]. Outra possibilidade seria a utilização de um enlace direcional sem fio de longa distância, para conseguir maior velocidade que comunicações que normalmente utilizariam mais que um salto [Hu et al., 2003a]. Uma terceira forma seria utilizar um canal diferente do utilizado na comunicação com uma potência de transmissão superior, o que também permitiria a maior velocidade sem que os vizinhos notassem. Além disso, uma técnica que pode ser usada é o envio dos bits diretamente, sem aguardar a chegada do pacote completo para começar a transmissão.

É interessante notar que, no caso de o atacante construir o seu túnel de forma honesta e confiável, nenhum prejuízo direto é causado a rede. Pelo contrário, um serviço é prestado ao melhorar a eficiência da conexão da rede. No entanto, o ataque coloca os nós maliciosos em uma posição privilegiada, que os permite gerar, no momento que desejarem, diversos tipos de prejuízos à rede.

Ainda que a rede implemente autenticidade e confiabilidade, o ataque ainda pode ser realizado, pois, normalmente, o nó não precisa se autenticar para encaminhar um pacote. É importante observar que, para as camadas superiores, o ataque é invisível, e mesmo para a camada de roteamento, a princípio é complicado perceber a presença de um Túnel de Minhoca.

2.3.2.5. Ataques Multicamadas

Existem alguns ataques que não estão ligados a uma camada específica do modelo OSI, mas que podem afetar diversas camadas.

Exaustão de Bateria

Neste ataque, o nó malicioso tem como objetivo consumir a bateria do nó atacado, até que o nó fique inativo. De fato, esse ataque pode se aplicar a várias camadas. No caso de fazer essa atividade por meio de interferências, se trataria de um problema de camada física. Já se a interferência for gerada com o objetivo de gerar retransmissões, trata-se de um problema da camada enlace. O ataque pode também retransmitir mensagens reais da rede, dificultando sua detecção.

Tantas versões para o mesmo ataque se justificam pela importância da vida útil da bateria para dispositivos móveis, e, pela mesma razão, várias metodologias para poupar bateria já foram desenvolvidas. Por essa razão aplicativos de segurança que exigem modo promíscuo são muito criticadas, pois a ação de escutar a rede continuamente gasta muita energia do nó, sendo mais recomendado colocá-lo dormindo sempre que possível. Outros nomes dados a esse ataque são *Sleep Deprivation Attack* [Wood e Stankovic, 2002] e *Spam Attacks* [Sancak et al., 2004].

Negação de serviço

O conceito de negação de serviço é muito amplo. O ataque de negação de serviço pode ser definido como qualquer ação que reduza ou elimine a capacidade da rede de reali-

zar uma de suas funções esperadas [Wood e Stankovic, 2002]. Assim sendo, a negação de serviço não seria causada apenas por ataques, mas por qualquer evento que prejudicasse a rede, como falhas de *hardware*, defeitos de programas, exaustão de recursos intencional ou não, condições ambientais não favoráveis ou qualquer interação entre esses fatores. Dessa forma, todos os ataques ativos poderiam gerar uma negação de serviço na rede, o que dá a esse ataque a classificação de multicamadas.

Uma forma mais severa deste ataque é a negação de serviço distribuída. Nesta, vários adversários estão espalhados pela rede fazendo um conluio para impedir que usuários legítimos tenham acesso aos serviços. Este ataque tem um efeito muito mais rápido sobre a rede, podendo impedir totalmente o seu funcionamento sem grandes dificuldades.

Sybil[3]

O ataque Sybil se baseia no fato de que é praticamente impossível, em sistemas computacionais distribuídos, que nós que não se conhecem apresentem identidades distintas convincentes. Sem a existência de um ponto central para controlar a associação de uma identidade a uma entidade, é sempre possível para uma entidade desconhecida apresentar múltiplas identidades. Assim, o ataque sybil acontece quando um único *hardware* assume múltiplas identidades em uma rede [Newsome et al., 2004].

Este ataque tem grande importância por muitos sistemas utilizarem sistemas de réplicas de dados armazenados, para ter garantia contra violação de integridade, e sistemas de fragmentação de tarefas, para impedir a violação da privacidade. Em ambos os casos, a redundância, mecanismo explorado pelo ataque, é um ponto chave. Assim, devido ao nó malicioso assumir múltiplas personalidades, o sistema poderia escolher o mesmo nó para guardar todas as réplicas ou fragmentos, o que acabaria com toda a segurança adquirida com o mecanismo.

O Sybil pode ser utilizado para atacar não só armazenamentos distribuídos. Uma outra possibilidade que utiliza redundância é o roteamento com múltiplos percursos. Em geral, protocolos que utilizam essa técnica buscam escolher caminhos disjuntos ou trançados para diminuir a possibilidade de existir um atacante na rota. O ataque sybil pode ser feito de tal forma a colocar uma identidade falsa em cada rota, de forma que todos os caminhos continuarão passando pelo nó malicioso. Ainda no campo de roteamento, outro possível problema que não tem relação com redundância é o ataque ao roteamento geográfico. Neste caso, o nó malicioso anunciará sempre uma de suas identidades sybil como o nó mais próximo ao destino, fazendo com que todos os pacotes de roteamento passem por ela.

Outro ataque possível é a utilização dos nós sybils para falsificar resultados de votações na rede. Sempre que existir algum mecanismo cooperativo para tomada de decisões na rede, o nó malicioso pode gerar diversas identidades para votar sempre a seu favor. Outro ataque é a alocação injusta de recursos, que pode ocorrer em redes que fazem divisão temporal para acesso ao meio. Neste caso, o nó malicioso utiliza todas as suas

[3]A primeira descrição do ataque Sybil foi feita em [Douceur, 2002], para redes *peer-to-peer*. O nome do ataque foi inspirado em um caso famoso nos EUA, onde uma mulher sofria de múltiplas personalidades, num total de 16 diferentes personalidades. Sybil Isabel Dorsett foi o pseudônimo criado pela autora Flora Schreiber para proteger a identidade real da paciente, em seu livro "Sybil", Warner Books, 1973.

identidades falsas para obter um maior tempo de acesso. Por fim, uma outra utilização para os nós sybils acontece em redes que utilizam mecanismos de confiabilidade. Em tais redes, a índole do nó é dada pela observação de suas ações. Um nó só é considerado malicioso se cometer diversas ações consideradas ruins ou se cometer uma grande ação ruim. Assim, duas estratégias podem ser utilizadas. A primeira seria o espalhamento da culpa, na qual o nó sybil utiliza cada uma de suas identidades para fazer pequenas ações ruins, de forma que nenhuma delas possa ser considerada maliciosa. A outra estratégia seria utilizar uma identidade para realizar uma ou mais ações ruins até que ela fosse expulsa, classificada como maliciosa. Quando isso acontecesse o nó geraria uma nova identidade e a usaria para continuar atacando.

Existem diversas propostas de defesas para o Sybil, sendo a maioria delas baseadas em métodos de autenticação ou de validação de chaves distribuídos. Estes serão descritos na Seção 2.4.1.

Identidade Falsa (*Impersonating*) e Ataque da Replicação

Estes ataques se assemelham muito ao Sybil. Nestes, nós maliciosos assumirão uma ou mais identidades da rede, porém desta vez, todas as identidades são reais, e cada identidade estará ligada a um ou mais hardwares diferentes, caracterizando respectivamente a Identidade Falsa e o Ataque da Replicação. A Replicação serve para inserir vários nós maliciosos, sem ter a dificuldade de se roubar várias identidades. Desta forma, os nós maliciosos replicam alguma identidade roubada e utilizam as réplicas simultaneamente dentro da rede [Chan et al., 2003].

No caso dessas réplicas serem muito numerosas, os adversários podem dominar a rede através de um conluio para ter vantagens em casos votação, ou ainda tirar vantagem apenas por estarem participando da rede. Deve-se notar que esses ataques, que, em geral, acontecem após uma violação ou uma quebra de algoritmo criptográfico, fazem com que o atacante tenha o segredo da rede, podendo participar de todas as suas atividades como um nó legítimo. Assim, ele pode, por ter se tornado um atacante interno, executar a maioria dos ataques já descritos, com a facilidade do conluio com as outras réplicas.

Cabe ressaltar, que apesar da gravidade do efeito causado, a Replicação é de fácil detecção, devido a uma mesma identidade se anunciar em diversos pontos da rede. Algumas propostas eficientes já foram feitas, embora elas não estejam incluídas nos protocolos mais populares. No caso da identidade falsa, se o nó legítimo tiver sido destruído, a detecção é muito mais complicada, pois a identidade é única na rede.

Uma outra variação da Identidade Falsa é o Ataque do Homem no Meio (*Man-in-the-Middle*) [Murthy e Mano, 2004]. Neste, o nó malicioso intercepta uma comunicação, enganando os dois nós que deveriam se comunicar, como pode ser visto na Figura 2.5. Uma vez que ele se passa por x para y e por y para x, ele está assumindo duas identidades reais da rede. Esse tipo de ataque só pode ocorrer em redes que não possuem um terceiro ponto para autenticar a comunicação entre os dois primeiros.

Violação

Este é um dos ataques mais preocupantes para redes onde os nós ficam desprotegidos. Ele consiste da violação física dos nós com o fim de obter informações e segredos,

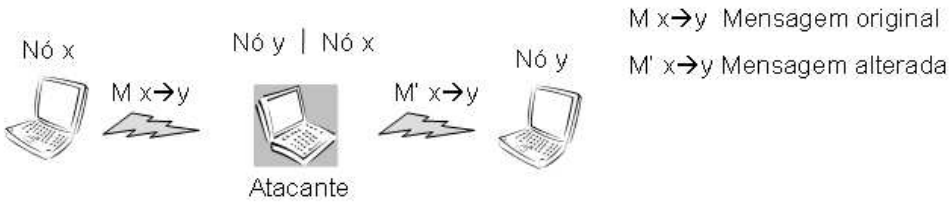


Figura 2.5. Ataque do Homem no Meio.

além de também comprometer o nó, com a inserção de códigos maliciosos ou ainda pela troca de partes do *hardware*. A maioria dos protocolos desenvolvidos para prover segurança falha em ambientes onde é possível ocorrer a violação.

De fato, não é simples garantir a segurança de todos os nós quando tratamos de redes de larga escala. Em especial pelo fato de que, em geral, existem muitas falhas de comunicação e períodos de sono, que tornam impraticável distinguir uma falha de um nó propositalmente desligado ou destruído. A proposta de defesa é a resistência à violação, descrita a seguir.

Resistência à Violação

O termo resistência à violação (*tamper proofing*), diz respeito à dificuldade imposta por um mecanismo de segurança à violação da informação, seja por software ou por hardware. O conceito pode ser estendido à capacidade da rede em resistir a ataques onde um usuário não autorizado se apossou de um dos nós da rede.

Desta forma, as técnicas de resistência à violação podem ser baseadas em hardware, em software, ou em ambos. Uma forma de aumentar a resistência à violação baseada em software é a utilização de associações de segurança temporárias. Por exemplo, toda vez que um nó for enviar uma requisição de rota em um protocolo de roteamento reativo, o sistema operacional pode exigir que o usuário entre com uma senha, ou forneça sua impressão digital para se autenticar. Se o equipamento for perdido ou roubado, o usuário não autorizado não conseguirá gerar pedidos de rota, e quando esta tentativa for feita, o sistema operacional do nó pode tomar alguma atitude adicional, como,por exemplo destruir qualquer chave armazenada no sistema, aumento sua resistência à violação [Yi et al., 2001]. Esse tipo de medida tem o problema de ser considerada como inconveniente pela maioria dos usuários.

A utilização de hardware resistente à violação possui, naturalmente, um problema de custo-benefício. Quanto mais seguro o hardware, mais caro [Anderson e Kuhn, 1996]. Além disso, como na segurança do software, não existe sistema de hardware inviolável. Considerem-se os *smart-cards*, utilizados, por exemplo, em alguns decodificadores de TV por assinatura para controlar os canais aos quais o usuário tem direito de acesso. *Smart-cards* típicos são constituídos de um micro-processador, memórias ROM, EEPROM e RAM, além de portas de entrada e saída. Normalmente, o apagamento de um bit da EEPROM exige uma voltagem alta. Ataques podem se basear no bloqueio desta voltagem mais alta, para evitar o apagamento de alguma informação. Por exemplo, em algumas empresas de TV por assinatura, o *smart-card* é gravado por padrão com todos os canais

de TV habilitados. No *set-top Box* do usuário, um sinal é emitido pela operadora de forma a "apagar"os canais não contratados do smart-card. Um ataque simples consiste em bloquear a geração de voltagem mais alta para esta operação, no set-top box, garantindo o acesso a todos os canais. Uma solução para aumentar a segurança contra este tipo de ataque é incluir o circuito de geração da voltagem alta dentro do próprio smart-card, o que aumenta o custo do cartão, e *dificulta* a ação do atacante: o circuito para gerar a voltagem mais alta utiliza um capacitor, que pode ser identificado e destruído com a ajuda de um microscópio. Para aumentar a segurança, um encapsulamento inviolável pode ser utilizado, o que aumenta ainda mais o custo do sistema.

2.4. Mecanismos de Segurança Específicos de Redes Ad Hoc

A utilização dos mecanismos de segurança deve levar em consideração a relação custo/benefício de cada solução. Nesta seção são descritos os principais mecanismos de seguranças que podem ser utilizados nas redes ad hoc.

2.4.1. Distribuição de Chaves

O principal objetivo do gerenciamento de chaves é compartilhar uma chave com um grupo de participantes. Para tanto, quatro operações podem ser necessárias: a pré-distribuição, o transporte, a arbitração e o acordo de chaves [Murthy e Mano, 2004].

A Pré-Distribuição de Chaves consiste da distribuição das chaves pelos nós interessados antes do início da comunicação. Isto exige que todos os nós da rede sejam previamente conhecidos, embora não seja exigido que todos participem sempre da rede. Uma vez que esta fase concluída, não é possível inserir novos nós ou trocar chaves. Grupos de comunicação, que devem ter uma chave própria também devem ser estabelecidos nesta fase.

No Transporte de Chaves, as entidades trocam chaves para se comunicar. O método mais simples para essa fase se chama *Key Encription Key* (KEK), e consiste em criptografar a nova chave com o segredo compartilhado, e apenas os nós que possuírem esse segredo podem obter a nova chave. No caso de não existir uma chave previamente conhecida por um grupo, mas existir uma infra-estrutura de chave pública, essa nova chave pode ser trocada criptografando-a com a chave pública do nó que irá recebê-la.

A Arbitração de Chaves utiliza um arbitrador central para criar e distribuir chaves entre os participantes, o que a torna uma especialização da fase de transporte. Em sistemas infra-estruturados, um ponto central é escolhido para exercer a função de arbitrador. No entanto, em redes ad hoc, esta função centralizada de arbitrador é proibitiva por causa da ausência de infra-estrutura e restrições de recursos. Entre esses está a necessidade do arbitrador estar sempre ativo e acessível, sob pena de negação de serviço caso o nó se mova ou saia da rede, ou ainda tornar um único ponto vulnerável a ataques. A utilização de réplicas da base de dados para resolver o problema da negação de serviço aumentaria o número de nós guardando os segredos da rede, gerando mais pontos de vulnerabilidade, além de ser uma solução mais dispendiosa em termos de recursos.

Por fim, o Acordo de Chaves corresponde à troca de chaves posterior ao início da rede. Aqui serão estabelecidos segredos entre nós através de chaves assimétricas, se elas

estiverem disponíveis. Isto é necessário para realizar uma comunicação segura dentro da rede, embora seja uma operação muito custosa.

A seguir serão descritos os principais métodos propostos na literatura para distribuição de chaves, tanto simétricos quanto assimétricos.

2.4.1.1. Criptografia de Limiar

A criptografia de limiar (*threshold criptography*) é aplicada para solucionar o problema das Autoridades Certificadoras (AC). Através desse tipo de criptografia, um segredo D é dividido em n partes $(D_1, D_2, ..., D_i, ..., D_n)$, de maneira que o conhecimento de k ou mais D_i partes facilita o cálculo de D, enquanto que o conhecimento de $k-1$ ou menos partes não permitem determinar D. Um esquema como esse é chamado de esquema de limiar (k,n), e se aplica à distribuição da função de certificação de uma autoridade certificadora por um grupo de nós. Isto é eficiente em redes ad hoc, onde a existência de uma única autoridade certificadora introduz um ponto único de falha, enquanto o uso de réplicas da autoridade certificadora para melhorar a acessibilidade produz mais vulnerabilidades. Utilizando criptografia de limiar (k,n), onde $n = 2k-1$, obtém-se um esquema de segurança robusto, pois mesmo que metade dos nós da rede fiquem comprometidos, ainda é possível reconstruir a chave k. O compromisso que se busca com a criptografia de limiar é entre segurança e conveniência, pois o ideal para a segurança é que todos os pedaços fossem necessários, embora para conveniência o ideal fosse utilizar o menor número de pedaços possível. Em outras palavras, no ambiente ad hoc se busca o equilíbrio entre a disponibilidade e tolerância à intrusão. Assim, um adversário precisa destruir $(n-k+1)$ nós para indisponibilizar o serviço, ou ainda roubar o segredo de k nós para obter a chave secreta [Kong et al., 2001].

A criptografia de limiar é ideal para aplicações nas quais um grupo de indivíduos com interesses conflitantes mutuamente suspeitos devem cooperar. O esquema de criptografia de limiar proposto em [Shamir, 1979] é baseado na interpolação polinomial de Lagrange com complexidade $O(nlog^2(n))$. Neste esquema, cada compartilhamento é calculado de um grupo de polinômios com grau k em um esquema (k,n). Usando os k valores em um sistema de equações lineares é possível encontrar o segredo, e usando menos que k equações, se obtém uma indeterminação. Um outro esquema baseado nos espaços euclidianos pode ser encontrado em [Blakley, 1979], onde o número de espaços é dado por k e o segredo compartilhado é dado por um ponto no espaço. Cada compartilhamento é um plano contendo o segredo, e a intercessão de k planos determina o ponto. Se $k = 3$, na ausência de alguns compartilhamentos, se obterá um plano ou uma linha, com infinitas possibilidades, o que torna o segredo indeterminável [Gahlin, 2004].

Uma vez que em redes ad hoc não se deseja que nenhum nó possua o segredo da autoridade certificadora, pois ele se tornaria um alvo para ataques, não é adequado que a chave possa ser recuperada, mesmo juntando k pedaços da mesma. O que se utiliza é a criptografia de limiar para fazer assinaturas. Nessa criptografia, cada nó em posse de um pedaço da chave fará uma assinatura parcial da mensagem, de forma a que, quando todas as k assinaturas parciais forem obtidas, seja possível construir a assinatura completa. Na Figura 2.6 a chave S é dividida em n pedaços, que são distribuídos entre n nós. Um

nó que deseje uma assinatura para a mensagem m deverá enviá-la aos n nós, e ainda que $n-k$ assinaturas se percam, ele será capaz de reconstituir a assinatura de m. Uma multi-assinatura de limiar através da combinação do RSA com a criptografia de limiar é proposta em [Frankel e Desmedt, 1992].

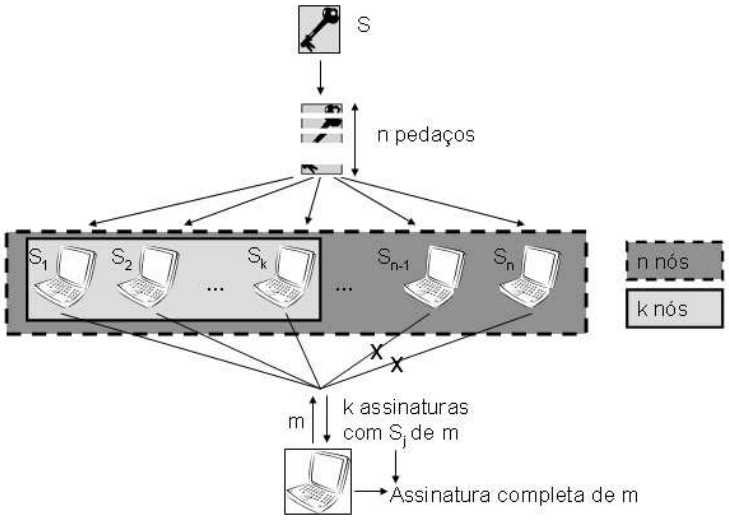


Figura 2.6. Esquema de assinatura com criptografia de limiar.

O maior problema desse tipo de criptografia é que os nós maliciosos enviarão assinaturas parciais inválidas, o que geraria uma assinatura completa inválida. Assim, o nó que desejar que sua mensagem seja validada deve ser capaz de testar todas as possibilidades de combinações de chaves até que ele obtenha uma assinatura completa válida. Esquemas mais robustos para essa composição de chaves também foram propostos, como o *Digital Signature Standard* (DSS) e são baseados em redundâncias entre as chaves [Zhou e Haas, 1999] [Gennaro et al., 1996].

O DSS é um esquema para melhorar o desempenho da criptografia de limiar. Uma primeira proposta mostrou que era possível proteger uma rede com n nós, sendo t nós maliciosos, dede que $n = t^2 - t + 1$. Isso significa que para n servidores, é possível garantir a resistência para $\sqrt{n}$ partes corruptas. Gennaro et al. mostraram um esquema de assinatura DSS, onde para obter um limiar de segurança t são necessários $2t+1$ compartilhamentos ativos durante o cálculo da assinatura, obtendo limites de até $\frac{n-1}{2}$ [Gennaro et al., 1996]. Utilizando mecanismos para detectar e corrigir assinaturas erradas, foi criado um sistema robusto a t falhas, ou seja, que suporta até t invasores. Este sistema é capaz de se proteger contra nós que se recusam a cooperar, desde que $t < \frac{n}{3}$, ou ainda contra nós que se comportam de forma arbitrária e maliciosa, e nesse caso tem-se como premissa $t \leq \frac{n}{4}$. O DSS também suporta pró-atividade, como um meio de ser mais resistente a ataques.

Os sistemas de atualização de chaves, ou pró-ativos [Zhou e Haas, 1999] têm como princípio impedir que os atacantes consigam obter k compartilhamentos da chave. Uma vez que um atacante leva um tempo até conseguir roubar um segredo compartilhado de um nó, se os segredos nunca mudarem, ele consegue, após um longo tempo, roubar todos os k segredos. Se for aplicado um sistema de atualização de chaves, isso poderá ser impedido. Nos sistemas de criptografia de limiar pró-ativos, os novos segredos são com-

putados a partir dos antigos, por meio de colaboração dos servidores sem que o sistema de certificação deixe de funcionar. Os novos segredos compõem um compartilhamento $(n, t+1)$ da chave secreta, e não podem ser descobertos ainda que o invasor consiga obter todos os compartilhamentos antigos. O processo de obtenção dos novos segredos começa com todos os n nós gerando n compartilhamentos S_{ij} do compartilhamento S_i que ele possui do segredo S. Em seguida, por meio de um canal seguro, cada nó j receberá as chaves S_{1j} até S_{nj} dos seus vizinhos, podendo gerar $S'_j = S_j + \sum_{i=1}^{n} S_{ij}$.

Pode-se, então, dizer que a técnica da criptografia de limiar trouxe melhorias às redes ad hoc, permitindo dar integridade e confidencialidade aos dados, autenticação, não-repúdio e disponibilidade do serviço de certificação.

2.4.1.2. Criptografia por ID

O sistema de criptografia por ID, proposto em [Shamir, 1984], tem como motivação inicial simplificar o gerenciamento de certificados de e-mail. A idéia é demonstrada pelo seguinte exemplo. Se Alice desejasse enviar um e-mail para Bob, ela não precisaria buscar o certificado para a chave pública de Bob, mas apenas criptografaria a mensagem utilizando a *string* bob@provedor.com. Quando Bob recebesse essa mensagem, caso ele não tivesse ainda gerado o seu segredo, ele se dirigiria ao *Private-Key Generation service* (PKG) e obteria o seu segredo. Dessa forma, Alice não seria impedida de enviar o e-mail caso Bob não tivesse obtido sua chave ou caso seu certificado já tivesse sido revogado [Boneh e Franklin, 2001].

Em um esquema de criptografia baseado em ID, os nós da rede terão como chave pública sua própria identificação, que pode ser qualquer valor arbitrário, e sua chave privada será gerada pela entidade PKG. Outra função da PKG é gerar as chaves mestras pública e privada, necessárias para criptografia e decriptografia, onde se assume que a chave mestra é conhecida por toda a rede. Para obter um nível maior de segurança, a PKG pode ser implementada utilizando criptografia de limiar. A grande vantagem deste esquema é que não é necessário que cada nó gere e divulgue a sua chave pública [Khalili et al., 2003].

As duas maneiras mais conhecidas para se realizar a criptografia de ID são o BF-IBE (Boneh and Franklin ID-Based Encription scheme) ou *Weil Paring* e o *Gap Diffie-Hellman* (GDH). O BF-IBE é um esquema baseado em mapas bilineares em curvas elípticas e é considerado o primeiro exemplo prático de uma criptografia por ID. O GDH tem sua segurança baseada na alta dificuldade de resolver o *Computational Diffie-Hellman Problem* (CDHP). Utilizando o GDH obtido pelo pareamento bilinear, obtém-se uma assinatura baseada em ID com os mesmos parâmetros do BF-IBE, com mesma eficiência, embora se saiba que o *Bilinear Diffie-Hellman Problem* (BDHP) seja mais complexo que o CDHP [Cha e Cheon, 2003].

O problema das assinaturas baseadas em ID é a dificuldade de prover a propriedade do não-repúdio e autenticação. Para tanto, é necessário proteger o PKG, o que pode ser feito através de criptografia de limiar, e possuir um sistema de arbitração de identificação. Caso a identidade do usuário não importe, bastando apenas conhecer o *hardware* que está dando origem às mensagens, a emissão de segredos pode ser feita para qualquer

ID arbitrário que faça o pedido.

2.4.1.3. Criptografia Comutativa

Para realizar o transporte de chaves na ausência de uma entidade central ou de um segredo compartilhado, existe um esquema proposto por Shamir chamado de protocolo dos três passes (*three-pass protocol*) [Shamir et al., 1978], baseado em criptografia comutativa.

Na criptografia comutativa existem duas funções, chamadas de f e g, que podem ser compostas, ou seja, existe $f(g(x))$, e que são reversíveis. Supondo que o nó X e o nó Y desejam se comunicar, o nó X irá escolher uma chave K que ele deseja utilizar para se comunicar com Y. Ele irá gerar também uma chave K_x para criptografar K com a função f, e envia o resultado para Y. O nó Y, por sua vez, ao receber essa mensagem gera uma chave K_y, criptografa tudo com K_y usando a função g e envia o valor resultante para X. O nó X, então, decriptografa a mensagem usando K_x e a inversa de f, enviando o resultado para Y. Este, por sua vez, utiliza a inversa de g e K_y para obter o valor K. Este processo pode ser visto esquematicamente na Figura 2.7.

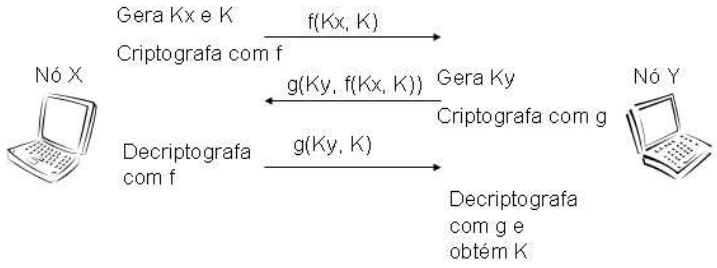


Figura 2.7. Criptografia comutativa [Murthy e Mano, 2004].

2.4.1.4. Esquemas de pré-distribuição de chaves

O princípio dos esquemas de pré-distribuição de chaves é fazer com que, a partir de um grupo de segredos pré-estabelecidos, os nós possam se comunicar baseados em segredos que possuem em comum. São necessários ainda mecanismos capazes de garantir que mesmo que dois nós não possuam segredos em comum, eles sejam capazes de se comunicar através de um caminho por pares que compartilham segredos.

Em [Eschenauer e Gligor, 2002] o primeiro esquema de pré-distribuição de chaves foi proposto. Neste esquema, uma fase de inicialização atribui um grupo de chaves S de um total de m chaves para cada nó. O número de chaves em S é escolhido de forma a garantir uma probabilidade p de encontrar pelo menos uma chave em comum entre quaisquer dois nós vizinhos. Ao entrar na rede, os nós tentam descobrir entre seus vizinhos se existem chaves em comum, através de desafios e identificadores de chaves. A chave que for compartilhada entre os dois nós se torna a chave para criptografia daquele enlace. Depois desta descoberta de chaves, um grafo conectado de enlaces seguros é formado, e

a partir deste são buscadas rotas para os nós vizinhos que não possuem uma chave em comum.

Em [Chan et al., 2003] são propostos três mecanismos para pré-distribuição de chaves, chamados de *q-composite random key predistribution scheme*, para redes de pequena escala, *multi-path key reinforcement scheme*, e *random-pairwise keys scheme*.

O esquema *q-composite random key predistribution* é uma melhoria para o esquema básico de Eschenauer e Gligor. A modificação consiste em checar a existência de q chaves em comum, ao invés de apenas uma. Aumentando o valor de q, é possível aumentar a resistência contra ataques de violação. Por outro lado, com este novo esquema é necessário buscar um equilíbrio, pois apesar do aumento de q melhorar a resistência contra o roubo de chaves, ele também força o uso de mais chaves em S para manter a probabilidade p, o que permite que o atacante consiga mais chaves com menos violações. O *multi-path key reinforcement scheme* foi outra melhoria proposta. Uma vez que a chave usada para criptografar comunicação entre dois nós faz parte do espaço de chaves, a violação de um nó da rede pode permitir obter o segredo da comunicação de diversos pares. A idéia do *multi-path key reinforcement scheme* é gerar uma chave de comunicação entre os nós a partir de um ou-exclusivo de vários números randômicos, transmitidos de uma extremidade a outra da comunicação por meio dos múltiplos caminhos existentes no grafo entre os dois nós. A chave não é passada apenas por um canal seguro porque o nó malicioso pode possuir a chave desse par e descobrir o segredo. Por outro lado, é pouco provável que o invasor possua as chaves de todas as rotas seguras. O terceiro esquema proposto, o *random-pairwise keys scheme*, traz como proposta permitir autenticação na comunicação. Com o esquema inicial, dois nós podem possuir o mesmo S, de forma que a comunicação destes com um terceiro nó pode estar utilizando a mesma chave, não permitindo que se identifique a origem exata da informação enviada. Assim, foi proposto que cada chave seja utilizada no máximo em dois grupos de chaves, de forma que ao escolher k para ser a chave criptográfica da comunicação, ela seja ligada a um ID em cada um dos lados da comunicação, garantindo a autenticação, já que nenhum outro nó possuirá aquela chave. Outra vantagem adicionada a esse método é a utilização da menor probabilidade p tal que a probabilidade de todo o grafo estar conectado ser c, permitindo que um número menor de chaves seja guardado.

Em [Newsome et al., 2004] são propostos três métodos de pré-distribuição de chaves para evitar o ataque Sybil, associando sempre grupos de chaves únicos a identidades, além de utilizar mecanismos de validação de chaves, tanto por um nó, quanto pela cooperação de vários nós da rede. O primeiro método proposto consiste de modificar o esquema básico utilizando funções pseudo-randômicas e *hash*, de forma a dificultar a criação de um ID a partir do roubo de um grupo de chaves. A segunda proposta é a adaptação do esquema de *random-pairwise keys* para utilizar grupos de chaves associados a identificações. Um terceiro esquema de *random-pairwise keys* pode ser feito utilizando uma informação pública U_i e uma privada V_i para cada nó, de forma que o segredo da comunicação do nó i com o nó j será dado por $f(V_i, U_j)$, o que é igual a $f(V_j, U_i)$, devido a características especiais da função f. Esse esquema impõe a restrição de no máximo λ nós comprometidos. O terceiro esquema proposto por Newsome et al. é o *Multi-Space Pairwise Key Distribution*, que é uma combinação dos métodos *random-pairwise keys* e básico modificado, aplicados de forma a dar uma maior segurança contra o roubo de

segredos de nós e a criação de novas identidades.

2.4.1.5. Algumas Considerações

Vários sistemas para gerenciar chaves têm sido recentemente propostos, utilizando os esquemas citados na seção anterior. No entanto, esses sistemas utilizados isoladamente não são capazes de garantir todos os requisitos de segurança da rede. Primeiramente, deve-se levar em conta que todos os sistemas que assumem a existência de um segredo pré-estabelecido supõem a atuação prévia de um administrador, o que não é condizente com o cenário de redes ad hoc, embora em muitos casos seja necessário. Além disso, a maioria dos sistemas apresentados consegue autenticar a origem das mensagens, embora não exista uma relação com o usuário que está naquela origem. Essa relação é essencial para a maioria das redes de computadores, e não existe nenhum mecanismo, senão a interferência de um administrador, de implementar estes esquemas isto em redes ad hoc.

Muitos estudos foram feitos sobre os esforços para quebrar os sistemas propostos, e a escolha por cada um deles deve levar em consideração tanto o custo de mensagens inseridas na rede quanto o custo dos recursos utilizados, além do custo de quebra, de acordo com o cenário utilizado. É evidente que o nível de segurança de um cenário militar é totalmente diferente do de uma rede ad hoc domiciliar, e isso deve ser considerado na escolha do mecanismo de gerenciamento de chaves.

2.4.2. Funções *Hash*

Dada a possibilidade de limitação de recursos dos nós de uma rede ad hoc, foram propostos mecanismos mais eficientes do que os tradicionais mecanismos de criptografia assimétrica. Esses novos mecanismos utilizam funções *hash* unidirecionais a fim de prover segurança nas tarefas de roteamento. Nessa seção, são descritos os Códigos de Autenticação de Mensagem, as cadeias de *hash* e as árvores de *hash*. Ao final, é apresentado o protocolo de autenticação TESLA, que utiliza estes mecanismos em seus esquemas de segurança.

2.4.2.1. Códigos de Autenticação de Mensagem

O uso de Códigos de Autenticação de Mensagem (*Message Authentication Code - MAC*) é uma maneira de garantir a integridade e autenticidade das informações trocadas por duas entidades através de um canal de comunicação inseguro. Tal mecanismo se baseia no compartilhamento de uma chave secreta entre as entidades. Quando uma das entidades deseja enviar uma mensagem à outra, ela anexa à mensagem um valor autenticador, denominado valor MAC ou simplesmente MAC, calculado em função da mensagem enviada e da chave secreta. Na recepção, a outra entidade, usando o mesmo procedimento e a mesma chave, recalcula o valor autenticador e compara com o valor MAC anexo à mensagem recebida. Somente se os valores forem iguais a informação recebida é considerada inalterada durante o trânsito pelo canal de comunicação. O objetivo desse mecanismo é impedir que, sem o conhecimento da chave secreta, um adversário seja capaz de forjar o valor MAC de uma nova mensagem, mesmo que muitas mensagens anteriores e seus

valores MAC correspondentes sejam conhecidos. Assim, os algoritmos MAC protegem as mensagens contra tentativas de falsificação, ou seja, tentativas de calcular o valor MAC sem o conhecimento da chave secreta.

A maioria dos algoritmos MAC foi construída usando cifradores de bloco (*block ciphers*) como o famoso DES. O algoritmo MAC desse tipo mais popular é conhecido como CBC MAC [Bellare et al., 1994]. Surgiram, porém, propostas de construção de MACs usando funções *hash* criptográficas como MD5 e SHA-1. Uma vantagem dessa abordagem é a sua simplicidade e eficiência, pois as funções *hash* populares são mais rápidas do que cifradores de bloco em implementações em *software*. Outra vantagem é que essas implementações são de domínio público. Os algoritmos que se baseiam nessa abordagem são denominados algoritmos HMAC [Bellare et al., 1996] e, devido as suas vantagens, são largamente utilizados para proteção das mensagens de roteamento em redes ad hoc.

2.4.2.2. Cadeia de *Hash*

Uma cadeia de *hash* (*hash chain*) é definida como uma seqüência, gerada a partir da aplicação sucessiva de uma função *hash* a uma semente, geralmente um número gerado aleatoriamente. Dado um dos elementos da cadeia de *hash*, pode-se garantir que os valores seguintes fazem parte da mesma cadeia, aplicando-se a função *hash* novamente sobre o elemento conhecido um número adequado de vezes. A unidirecionalidade da função *hash* impede que se obtenha os elementos anteriores da cadeia.

Para criar uma cadeia de *hash* de n elementos, um nó deve gerar uma semente aleatória h_0 e calcular a lista de valores $h_0, h_1, h_2, h_3, ..., h_n$, onde $h_i = H(h_{i-1})$ para $0 < i \leq n$. Dessa maneira, ao inicializar a cadeia de *hash*, os valores da lista são gerados da esquerda para direita. Em seguida, esses elementos podem ser utilizados para garantir a segurança da atualização das mensagens de roteamento, por exemplo. Nesse caso, para autenticar os campos atualizados da mensagem, o nó deve seguir da direita para a esquerda. Dado um elemento previamente autenticado da cadeia de *hash*, é possível verificar a pertinência dos elementos posteriores. Por exemplo, supondo conhecido o valor autenticado h_i, pode-se autenticar o elemento h_{i-3} calculando-se $H(H(H(h_{i-3})))$ e verificando-se se o valor calculado é igual ao valor previamente autenticado h_i.

2.4.2.3. Árvore de *Hash*

Outro mecanismo de segurança bastante usado pelos protocolos de roteamento seguro é o esquema de autenticação através de árvore de *hash* [Merkle, 1980]. Para autenticar os valores $v_0, v_1, ..., v_{w-1}$, estes são colocados como nós-folha de uma árvore binária, suposta balanceada por simplicidade. Em primeiro lugar, é aplicada uma função *hash* H a todos os valores v_i, isto é, $v'_i = H(v_i)$. Em seguida, é utilizada a construção de Merkle, ilustrada na Figura 2.8, na qual cada nó interno da árvore é obtido a partir de seus dois nós filhos.

Para se obter o elemento m a partir de seus nós filhos da esquerda $m_{esquerda}$ e

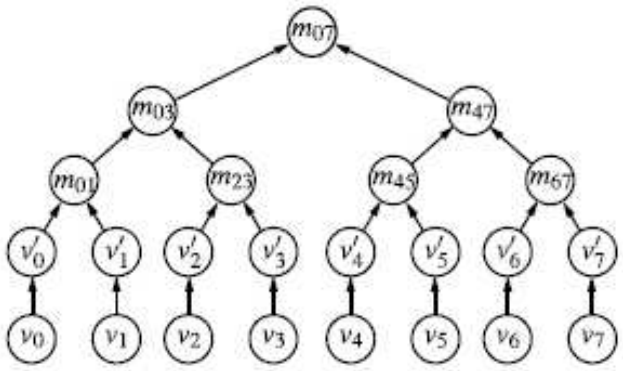


Figura 2.8. Exemplo de árvore de *hash*, extraído de [Hu et al., 2002].

da direita $m_{direita}$ deve-se calcular $H(m_{esquerda}||m_{direita})$, onde $||$ significa concatenação. Assim, os diversos níveis da árvore de *hash* são obtidos recursivamente a partir dos nós-folha da árvore. Por exemplo, na figura, $m_{01} = H(v'_0||v'_1)$ e $m_{03} = H(m_{01}||m_{02})$. O nó-raiz da árvore pode ser utilizado para autenticar os nós-folha. Para autenticar o elemento v_i, deve-se divulgar os valores i, v_i e todos os nós-irmão dos nós da rota entre v_i e o nó-raiz. Por exemplo, para autenticar o elemento v_2 da Figura 2.8, deve-se conhecer os valores v'_{03}, m_{01}, m_{47} para calcular $H(H(m_{01}||H(H(v_2)||v'_3))||m_{47})$. Caso o valor calculado seja igual a m_{07}, é assumido que o elemento v_2 é autêntico.

2.4.2.4. Autenticação TESLA

O protocolo TESLA [Perrig et al., 2002] é bastante eficiente para autenticação de mensagens e adiciona somente um único valor MAC à mensagem a ser transmitida para obter autenticação em difusão (*broadcast*). Em comunicações ponto-a-ponto, a utilização de algoritmos MAC para garantir a autenticação das mensagens é simples. Porém, em comunicações em difusão, os diversos destinatários teriam que conhecer a chave MAC, o que possibilitaria a ocorrência do ataque da Identidade Falsa. Por isso, a autenticação em difusão necessita de primitivas assimétricas. O protocolo TESLA difere dos protocolos assimétricos tradicionais, como RSA, pois a assimetria é obtida através de sincronização de relógios e atraso na divulgação da chave, ao invés de realizar operações que exigem grande poder computacional.

O protocolo TESLA determina que cada nó deve gerar uma cadeia de *hash* a partir de uma semente aleatória. Os elementos da cadeia serão utilizados como chaves para a autenticação das mensagens. O nó emissor deve divulgar o último valor da cadeia de *hash* gerada e, a partir daí, deve usar a cadeia no sentido inverso da geração para autenticar suas mensagens. Então, ao enviar uma mensagem, o nó emissor deve calcular o tempo médio que essa mensagem deve levar para chegar ao destino, divulgando a chave utilizada depois de decorrido esse tempo. Assim, os nós destinatários receberão a chave logo após terem recebido a mensagem. Com isso pode-se garantir que somente o nó emissor conhecia a chave TESLA utilizada para autenticar a mensagem recebida. Para verificar se a chave recebida está correta, deve-se aplicar a função *hash* sobre a chave um número adequado de vezes e comparar o resultado com o último elemento da cadeia de *hash*, que foi divulgado

pelo nó emissor. Se houver atraso no recebimento da mensagem ou a chave for divulgada antes que a estação de destino receba a mensagem, a mensagem deve ser descartada.

2.4.3. Um Novo Método de Confiança

As redes ad hoc dependem da colaboração dos nós para o seu bom funcionamento. No entanto, o comportamento de cada nó é dinâmico e depende dos seus objetivos e das suas limitações. Desta forma, cada nó da rede tende a decidir o que é melhor para si mesmo, sempre tentando maximizar seus objetivos. Entretanto, os nós da rede deveriam levar em consideração uma colaboração mínima, como em uma sociedade. Por isso, em uma rede ad hoc, uma ingênua dependência pode provocar baixa eficiência, alto consumo de energia e até mesmo ataques de nós maliciosos.

Existem alguns trabalhos que visam incentivar a colaboração dos nós em redes ad hoc através de sistemas de punição e incentivo [He et al., 2004] [Zhong et al., 2003] [Buttyan e Hubaux, 2000] [Buttyan e Hubaux, 2003]. O objetivo destes trabalhos é evitar a presença de nós egoístas criando um sistema de incentivo à colaboração e punição ao comportamento não colaborativo. Alguns trabalhos utilizam um sistema de crédito no qual cada nó recebe certa quantidade de unidades de crédito ao realizar uma ação que favoreça a um outro nó e o nó favorecido deve pagar pelo serviço que utilizou com seus créditos. Assim, nós egoístas seriam obrigados a colaborar a fim de receber uma quantidade suficiente de créditos que os permitam utilizar a rede. O grande problema destes sistemas é a necessidade de existir *hardwares* resistentes a alterações ou bancos virtuais em que todos os nós da rede possam confiar. Outra possibilidade para o sistema de incentivo/punição é a utilização de um esquema baseado na reputação dos nós. Neste caso, os nós devem ter mecanismos para avaliar e propagar a reputação dos outros nós da rede. Um sistema de reputação pode ser visto como um sistema de confiança. Apesar do estímulo à cooperação ser um ponto importante, não é suficiente para maximizar a eficiência da rede, por que os nós continuam dependendo de seus vizinhos de maneira ingênua.

Dentro deste contexto, a confiabilidade aparece como uma importante alternativa para viabilizar uma rede mais eficiente. A idéia é prover aos nós um mecanismo de confiança que os torne capaz de avaliar o grau de confiabilidade de seus vizinhos. Assim, os nós poderão sempre interagir com os vizinhos mais confiáveis, ignorando os vizinhos menos confiáveis.

A confiança é um conceito abrangente que engloba diversas definições. McKnight e Chervany [McKnight e Chervany, 2000] apresentam uma classificação conceitual da confiança. Neste caso, são abordados dois tipos de confiança. O primeiro está relacionado com o encaminhamento de pacotes, que permitirá aos nós decidir qual vizinho tem maior probabilidade de entregar um pacote corretamente, dado um determinado nó destino. O segundo é relacionado com a veracidade das informações recebidas de terceiros, que irá viabilizar a troca de informações entre os nós da rede de maneira eficiente e consistente.

Existem diversos trabalhos [Liu et al., 2004] [Pirzada e McDonald, 2004] que tratam da questão da confiança em redes ad hoc. No entanto, a maioria deles está focada apenas nos problemas de roteamento e de identificação de nós maliciosos.

Um novo modelo de confiança para redes ad hoc foi proposto por Velloso et al. O modelo visa simular as relações humanas de confiança e é baseado no aprendizado dos nós [Velloso et al., 2006]. A abordagem do modelo difere de outros trabalhos preocupados apenas com aspectos convencionais de segurança da rede, como a detecção de nós maliciosos, entre outros. O principal objetivo do modelo proposto por Velloso *et al.* é proporcionar aos nós de uma rede ad hoc uma maneira de avaliar e manter uma opinião sobre seus vizinhos, que servirá de base para a interação e a tomada de decisões entre eles. Assim, proporcionar um ambiente confiável não é um dos objetivos, mas sim capacitar os nós a reconhecer o ambiente ao qual pertencem. Isto é alcançado através da avaliação da confiabilidade de seus vizinhos. A informação de confiança será utilizada não apenas para o aprendizado e tomada de decisões, mas também poderá ser utilizada para a detecção e o isolamento de nós maliciosos.

O sistema proposto é distribuído e baseia-se na confiança que diferentes nós da rede possuem sobre um determinado nó sendo avaliado. O processo de avaliação do grau de confiança considera não somente o grau de confiança entre os nós adjacentes, mas também a precisão do grau de confiança e a maturidade do seu relacionamento. A fim de viabilizar a troca de recomendações, também foi proposto o protocolo REP (*Recommendation Exchange Protocol*), que simplifica a troca de informações de confiança na rede.

O modelo proposto pode ser representado por duas entidades distintas, como mostra a Figura 2.9. A entidade de Aprendizado é responsável por coletar e converter informações em conhecimento. A entidade de Confiança define como avaliar a confiança de um nó vizinho de acordo com o conhecimento adquirido pela entidade de aprendizado. Ambas as entidades interagem com todas as camadas. A entidade de Aprendizado considera o contexto do nó, que inclui o estado atual, as condições da rede, o lugar, a mobilidade e as ações de nós vizinhos para ajustar os parâmetros do modelo de confiança.

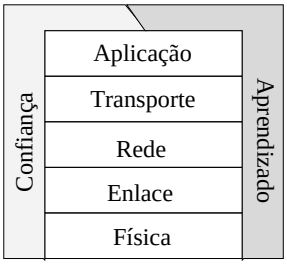


Figura 2.9. O modelo de confiança.

Velloso et al. consideram que os nós de uma rede realizam determinadas ações, como por exemplo, o envio de pacotes de dados, o encaminhamento de mensagens de roteamento, o descarte de pacotes de terceiros, entre outras possibilidades. O modelo parte do princípio que a ação de um nó tem sempre como objetivo maximizar o seu grau de satisfação. Desta forma, o grau de satisfação de um determinado nó é definido como o quão perto ele está do seu objetivo. O conjunto de ações de um nó determina o seu comportamento.

O grau de confiança é baseado nas experiências anteriores e na contribuição dos

nós vizinhos. As experiências anteriores resultam do julgamento das ações dos outros nós, realizado pela entidade de Aprendizado. Uma ação pode produzir um impacto positivo, negativo, ou nenhum impacto nos nós adjacentes. Os dois primeiros tipos geram uma reação que poderá disparar uma atualização do grau de confiança e eventualmente provocar uma mudança de comportamento. A capacidade de percepção das ações dos nós está diretamente relacionada com a eficiência da entidade de Aprendizado. Por exemplo, nós com sérias restrições de consumo de energia podem apresentar uma baixa eficiência por não poderem operar em modo promíscuo.

A contribuição dos nós vizinhos pode ser considerada no cálculo do grau de confiança. A contribuição é definida como o conjunto das recomendações dos nós vizinhos. Uma recomendação inclui o grau de confiança, a precisão do grau e a maturidade do relacionamento, que reflete a duração do relacionamento de confiança entre dois nós. Este conceito permite aos nós atribuir maior importância a recomendações baseadas em relacionamentos de mais longa duração. A precisão do grau de confiança pode ser vista como a confiabilidade na medida realizada, ou seja, a precisão representa a variação do grau de confiança sobre um determinado nó ao longo do tempo. Assim, para considerar as recomendações dos vizinhos, os nós devem utilizar o protocolo de troca de recomendações (*Recomendation Exchange Protocol* - REP).

Cada nó computa um grau de confiança para cada vizinho, o qual é atualizado sempre que necessário. Os nós são inteiramente responsáveis pelos próprios processos de avaliação do grau de confiança. Assim, o cálculo do grau de confiança é dividido em duas parcelas, como mostra a Equação 1, onde o valor do grau de confiança é uma variável continua limitada no intervalo $[0,1]$, onde o valor 1 representa o grau mais confiável.

$$T_a(b) = (1-\alpha)Q_a(b) + \alpha C_a(b), \tag{1}$$

onde α permite escolher o fator mais relevante. A primeira parcela, $Q_a(b)$, representa a capacidade de um nó de avaliar o grau de confiança baseado nas suas próprias informações, ou seja, utilizando apenas informações locais. A segunda parcela representa a contribuição dos nós vizinhos. A Equação 2 mostra como obter $Q_a(b)$, no modelo proposto.

$$Q_a(b) = \beta E_T + (1-\beta)T_a(b), \tag{2}$$

onde E_T representa um valor de grau de confiança obtido através do julgamento das ações de terceiros. O parâmetro $T_a(b)$ é o valor antigo de grau de confiança armazenado na tabela de confiança. O parâmetro β permite escolher o termo mais relevante. Isto significa que o parâmetro β depende de qual evento desencadeou a atualização do grau de confiança. Por exemplo, supondo que o nó a começou uma atualização sobre o nó b, desencadeada por uma recomendação do nó vizinho c, mas o nó a não notou nada de estranho no comportamento do nó b. Neste caso, o nó a pode ignorar o primeiro termo da Equação 2. Por outro lado, caso a atualização tenha sido desencadeada por uma ação, o nó a pode escolher $\alpha 0$ e $\beta = 1$, ignorando a contribuição dos nós vizinhos (Equation 1) e o valor antigo para o grau de confiança sobre o nó b (Equação 2).

A contribuição é o conjunto das recomendações de todos os nós vizinhos. Assim, a parcela $C_a(b)$ representa a contribuição de todos os nós $i \in K_a$ sobre o nó b ponderada

pelo grau de confiança do nó a ($T_a(i)$) sobre o nó i, como mostra a Equação 3.

$$C_a(b) = \frac{\sum_{i \in K_a} T_a(i) M_i(b) X_i(b)}{\sum_{j \in K_a} T_a(j) M_j(b)}. \quad (3)$$

K_n define o grupo de nós do qual as recomendações serão consideradas no cálculo da contribuição de outros nós. Assim, K_n é um subgrupo dos vizinhos de n (N_a) que inclui todos os nós que satisfaçam uma determinada condição. Dentre as possíveis condições para seleção de K_n, duas são consideradas neste trabalho:

$$K_n = \{\forall i \in N_a | T_a(i) \geq T_{th}\}. \quad (4)$$

onde T_{th} representa o valor de confiança limiar a partir do qual um vizinho será considerado nas contribuições. Uma outra opção seria selecionar os r primeiros nós pertencentes à N_a de acordo com o grau de confiança. A seleção de K_n é uma decisão importante para o processo de avaliação da confiança, que depende de muitos parâmetros. A relevância da recomendação de cada nó ($T_i(b)$) é fortemente relacionada à seleção de K_a. Quanto mais confiável for K_a mais útil será a recomendação dos nós vizinhos. A contribuição inclui não somente o grau de confiança ($T_a(b)$), como também a precisão desta medida e a maturidade da relação, que representa há quanto tempo os nós se conhecem. A maturidade do relacionamento ($M_i(b)$) é representada em segundos por uma variável contínua e X é uma variável aleatória de distribuição normal que pode ser expressa por

$$X_i(b) = N(T_i(b), \sigma_i(b)) \quad (5)$$

onde σ representa a precisão e é definida como o desvio padrão, similar ao trabalho de Theodorakopoulos e Baras [Theodorakopoulos e Baras, 2004].

Cada valor na tabela de grau de confiança do nó i ($T_i(b)$) está associado a um valor de desvio padrão ($\sigma_i(b)$), que se refere à variação do valor do grau de confiança que o nó i observou. Assim, após uma atualização do grau de confiança do nó i sobre o nó b, o nó i deve atualizar o valor de $\sigma_i(b)$, que é definido como:

$$\sigma_i(b) = \sqrt{\frac{\sum_{j=1}^{k} (\overline{S}_k - S_j)^2}{k-1}}, \quad (6)$$

onde S_k representa o conjunto dos k últimas amostras de grau de confiança sobre o nó b, dado $k \in \mathbb{N} \mid 2 \leq k \leq 10$. $\overline{S}_k$ é o valor médio. O parâmetro σ expressa a confiabilidade da medida do grau de confiança. Um valor grande de σ pode demonstrar a dificuldade do nó de avaliar o grau de confiança ou a instabilidade do comportamento do nó que está sendo avaliado.

A recomendação do nó i sobre o nó b é ponderada pela maturidade $M_i(b)$. Isto significa que quanto maior for o tempo que os nós se conhecem, maior será a relevância da sua opinião para o valor da contribuição final de todos os nós vizinhos. Nós maliciosos podem tentar falsificar graus de confiança por diversas razões. Por exemplo, um nó pode querer difamar um outro vizinho, ou pode querer convencer os outros vizinhos que um determinado nó malicioso é, na verdade, um nó de boa índole, ou ainda apenas querer confundir os outros nós. Assim, bastaria colocar um valor alto para a maturidade do

relacionamento de um grau de confiança forjado, para que esta recomendação tivesse um grande peso no processo de atualização dos nós vizinhos. Para minimizar este efeito, cada nó deve definir um limiar para o valor de maturidade da relação (M_{max}) de tal forma que a maturidade pode ser expressa por:

$$M_i(b) = \begin{cases} M_i(b), & \text{if } M_i(b) < M_{max} \\ M_{max}, & \text{if } M_i(b) \geq M_{max}. \end{cases} \quad (7)$$

O valor de (M_{max}) deve ser baseado na média dos valores de maturidade de relacionamento de todos seus vizinhos.

Para viabilizar a troca de recomendações foi concebido o protocolo de troca de recomendações (REP - *Recomendation Exchange Protocol*). Quando dois nós se encontram pela primeira vez, isto é, quando nós não vizinhos tornam-se vizinhos, uma mensagem de pedido de grau de confiança (*Trust Request - TREQ*) deve ser enviada em difusão. Por exemplo, quando o nó a e o nó b se encontram pela primeira vez, o nó a enviará uma TREQ com o identificador de nó igual a b ($TREQ_a^b$). O nó b, por sua vez, enviará uma $TREQ_b^a$. Os nós que recebem uma TREQ e possuem grau de confiança sobre o nó requisitado devem responder com uma mensagem de resposta de confiança (Trust Reply - TREP). Uma mensagem TREP possui a recomendação do nó remetente sobre o nó cujo grau de confiança foi requisitado. Após o envio de uma TREQ, o nó requerente deve esperar as respostas (TREP) de seus vizinhos por uma determinada quantidade de tempo (*timeout*). Caso o nó requerente não receba nenhuma TREP de seus vizinhos, a parcela da Equação 1 relativa à contribuição dos vizinhos deve ser ignorada, igualando o parâmetro α a 0.

Por último, a mensagem de anúncio de grau de confiança (*Trust Advertisement* - TA) é uma recomendação não solicitada. Uma mensagem TA inclui uma recomendação do nó que está anunciando sobre um de seus vizinhos. O envio de uma TA acontece sempre que uma atualização de grau de confiança gera um novo grau cuja diferença para a última TA enviada for maior que um determinado limiar ($TA_{threshold}$). A recepção de uma TA não implica necessariamente uma atualização do grau de confiança, mas apenas a atualização da recomendação recebida, para o caso em que o nó em questão seja também seu vizinho.

Um caso específico da Equação 1 acontece quando um nó toma conhecimento de um novo vizinho, para o qual não existe ainda nenhuma informação armazenada. Neste caso, a primeira parcela da equação, que representa as informações locais é substituída por um valor (F_a) previamente definido. Este valor está associado com a estratégia de atribuição do primeiro grau de confiança. Desta forma, reescrevendo a Equação 1 obtém-se:

$$T_a(b) = (\alpha - 1) \cdot F_a + \alpha \cdot C_a(b), \quad (8)$$

O valor de F_a dependerá das condições da rede, da mobilidade, do lugar e do estado atual do nó que irá atribuir.

A atualização do grau de confiança é um processo que pode ser desencadeado a qualquer momento, desde que o primeiro grau de confiança já tenha sido atribuído previamente. O modelo proposto considera que uma atualização é sempre desencadeada por um evento, no entanto, a ocorrência de um evento não implica necessariamente a

atualização do grau de confiança. A definição de um evento consiste na recepção de uma mensagem de recomendação ou na percepção de uma ação realizada por um dos vizinhos.

2.4.4. Outros Métodos

Existem alguns métodos simples para aumentar a segurança que podem ser implementados no projeto de protocolos de roteamento. Entre estes métodos estão a utilização de redundância, a verificação de bidirecionalidade do enlace, a autorização e a investigação.

Redundância

A redundância pode ser utilizada para prevenir diversos ataques em redes ad hoc, devido a sua característica de manter alternativas seguras, dando maior robustez à rede. O único ataque que pode inviabilizar a robustez gerada por redundância é o Sybil.

A utilização de múltiplas rotas é uma forma de se obter redundâncias no encaminhamento de dados. Embora as múltiplas rotas sejam usadas para balanceamento de carga e prevenção contra congestionamentos, elas também podem prover segurança, através do envio repetido dos dados pelos múltiplos caminhos. Assim, ainda que existam nós maliciosos em um caminho, descartando ou alterando pacotes, a mensagem real conseguirá ser entregue através de outra rota que não inclua esse nó. Para tanto, a escolha das rotas redundantes deve ser cuidadosa, escolhendo rotas disjuntas, ou seja, totalmente independentes, de forma a garantir que mesmo que um nó malicioso esteja em uma rota, ele não estará nas outras. O grande problema da utilização de rotas disjuntas é a dificuldade de encontrá-las em redes ad hoc, devido à baixa conectividade. Para resolver esta questão existem duas propostas. A primeira é ao invés de buscar rotas disjuntas, buscar os nós que são confiáveis na rede e tentar traçar múltiplas rotas passando por esses nós, independente de elas se sobreporem ou não [Ye et al., 2003]. A segunda proposta é a de utilizar caminhos trançados [Ganesan et al., 2001], que podem ter nós em comum, mas não possuem enlaces em comum, o que pode prover proteção probabilística contra o encaminhamento seletivo [Karlof e Wagner, 2003].

Apesar das vantagens trazidas pelo uso de múltiplos caminhos, é preciso observar também os contrapontos dessa medida. O uso de múltiplos caminhos pode sobrecarregar a rede, pois o tráfego total irá ser multiplicado pelo número de rotas entre pares, o que restringe o uso dessa técnica a redes que não trabalham próximas a saturação.

A utilização de armazenamento distribuído na rede é outro tipo de redundância [Newsome et al., 2004]. Este mecanismo, quando utilizado para prover segurança, constitui-se de distribuir várias réplicas de uma base de dados por diversos nós, de forma que a falha de um nó não comprometa a disponibilidade das informações.

Verificação de Bidirecionalidade do Enlace

Esta verificação está diretamente ligada à forma como o protocolo de roteamento determina quais são os seus vizinhos. Apesar de freqüentemente o roteamento executar acima da camada de enlace IEEE 802.11, nem sempre a informação gerada por ele através do ACK de enlace sobre quem são os vizinhos reais é utilizada. É comum que protocolos de roteamento assumam que todos os enlaces são bidirecionais, o que não é verdade em redes sem fio. Assim, eles determinam que os vizinhos de um nó são todos os nós que

ele é capaz de ouvir. Isso gera vulnerabilidades que permitem, por exemplo, o ataque da Inundação de *Hello*.

A verificação do estado do enlace é simples e não traz grandes custos ao funcionamento da rede. Basta que todos os nós enviem mensagens periodicamente, anunciando quem são os nós que eles são capazes de escutar. Qualquer nó que escute uma mensagem em que ele esteja listado como um possível vizinho pode considerar o emissor da mensagem como seu vizinho real.

Autorização

A autorização [Wood e Stankovic, 2002] é uma defesa contra os ataques do Direcionamento Falso, Encaminhamento Seletivo e Buraco Negro, que se baseia em escolher nós que serão os únicos autorizados a encaminhar informações de roteamento. Para tanto, é necessária a existência de autoridades certificadoras ou *Private-Key Generation Services*, para que seja possível autenticar os nós autorizados. Assim, qualquer pacote de roteamento que chegue por meio de um nó não-autorizado deve ser imediatamente descartado.

Para o bom desempenho da Autorização é necessário um bom sistema de confiabilidade, para que apenas nós que possam ser considerados de boa índole sejam escolhidos para essa função.

Investigação

A investigação é um teste que qualquer nó pode realizar para avaliar a conectividade da rede. Em particular em redes onde se tem o conhecimento da estrutura física, é possível enviar pacotes que atravessem o diâmetro da rede, funcionando como sondas. Essas sondas realizarão uma investigação sobre a rede, avaliando a existência ou não de áreas desconectadas. As sondas não são capazes de distinguir regiões de ataques de regiões de falhas, mas já trazem boas contribuições para a determinação das rotas a serem utilizadas.

É importante notar que um pacote de sonda deve ser indistinguível de pacotes normais, para que os nós que descartam pacotes maliciosamente não o encaminhem apenas para reforçar a validade da rota que o inclui. Para um bom desempenho, esses pacotes devem ser enviados periodicamente, tanto para manter um mapa das rotas disponíveis atualizado, quanto para aumentar a probabilidade de descobrir quais são os nós que realizam encaminhamento seletivo, egoísmo e ganância.

Compressão

Técnicas de compressão podem ser utilizadas para aumentar a segurança durante a atualização de bases de dados entre dois nós. Um tipo de compressão especial para essa função é chamado de Compressão Delta, e funciona através remoção de redundâncias das diferenças entre as duas bases de dados que serão atualizadas, de forma a diminuir ao máximo o tráfego na rede. A conseqüência disto é que a previsibilidade dos dados que estão sendo transmitidos cai muito, dificultando a ação de espiões (*eavesdroppers*), mesmo na ausência de algoritmos criptográficos [Bing, 2006].

2.5. Protocolos de Roteamento Seguros

Os protocolos de roteamento inicialmente propostos para redes ad hoc sem fio consideravam apenas cenários onde os nós eram confiáveis, premissa que compromete a sua eficiência em ambientes hostis. Para suprir a necessidade de segurança, foram projetados novos protocolos, que serão descritos nessa seção.

Cabe ressaltar aqui a diferença entre roteamento seguro e transmissão segura. As mensagens encaminhadas em uma rede ad hoc podem ser divididas em dois tipos: mensagens de controle e mensagens de dados. Entre as mensagens de controle, destacamos as de roteamento, trocadas pelos nós da rede a fim de estabelecer uma rota entre os nós que desejam se comunicar. Para isso, essas mensagens devem ser processadas em cada nó e, possivelmente, modificadas antes de serem encaminhadas, de acordo com as regras definidas pelo protocolo de roteamento. Já as mensagens de dados só podem ser trocadas após o estabelecimento da rota e seu conteúdo deve ser mantido, preferencialmente em sigilo, até o destino. Assim, as mensagens de roteamento e as de dados possuem naturezas e propósitos distintos, exigindo requisitos de segurança diferentes. Em geral, para ambas as mensagens se deseja garantir a integridade e autenticação, pois cada nó deseja receber dados corretos e precisa confiar nas informações de roteamento recebidas.

Os protocolos de roteamento de redes ad hoc móveis existentes estão expostos tanto a ataques passivos como ativos (Seção 2.3). Os ataques ativos mais freqüentes são os ataques de fabricação, alteração e modificação de pacotes, Identidade Falsa (*Spoofing* ou *Impersonating*) e o túnel de minhoca (*wormhole*). Com a publicação destes ataques, tornou-se evidente a fragilidade dos protocolos, e percebeu-se que propostas de segurança como IPSec (*IP Security*) [Atkinson, 1995] somente são válidas para autenticação fim-a-fim e para prover segurança entre entidades que já possuem roteamento estabelecido entre si, não garantindo roteamento seguro. Então, foram propostos novos protocolos que visam garantir a integridade, a autenticidade e o não-repúdio das mensagens de roteamento. Os novos protocolos projetados para prover segurança, expostos a seguir, utilizam os mecanismos descritos na Seção 2.4 a fim garantir segurança no roteamento.

2.5.1. ARAN

O ARAN (*Authenticated Routing for Ad hoc Networks*) [Sanzgiri et al., 2002] é um protocolo de roteamento seguro reativo que garante autenticidade, integridade e não-repúdio das mensagens de roteamento, baseado em criptografia assimétrica. Para isso, é suposta a existência de um servidor certificador confiável, cuja chave pública deve ser conhecida por todos os nós da rede, responsável pela manutenção e distribuição dos certificados de todos os nós da rede.

Após um processo preliminar de certificação, pelo qual todos os nós devem passar antes de ingressarem na rede, o protocolo ARAN prevê dois procedimentos. O primeiro é um procedimento obrigatório que garante autenticação fim-a-fim no processo de descoberta de rota. O segundo procedimento é opcional e tem por objetivo garantir de maneira segura que o caminho obtido no processo de descoberta de rota é o menor caminho até o destino.

O primeiro procedimento é iniciado quando o nó de origem envia um pacote de

descoberta de rota (*Route Discovery Packet - RDP*) por difusão (*broadcast*) para os seus vizinhos a fim de determinar uma rota para um dado destino. O pacote RDP possui o endereço IP do nó de destino, o certificado do nó de origem, o número de seqüência e um *timestamp*. Antes de ser enviado, todo o pacote RDP é assinado com a chave privada do nó de origem. O próximo nó que receber o pacote RDP deve validar a assinatura usando o certificado do nó de origem, e, em seguida, atualiza sua tabela de roteamento com os endereços dos nós de origem e destino, assina a mensagem com sua chave privada e a envia para seus vizinhos, juntamente com o seu certificado. As próximas estações que receberem a mensagem devem repetir o mesmo procedimento, retirando sempre a assinatura do nó anterior antes de assinar a mensagem e reenviá-la para os nós vizinhos. Dessa maneira, as estações intermediárias vão encaminhando o pacote até o destino final. O encaminhamento só não é feito se alguma condição de segurança não for respeitada, como a assinatura do nó anterior não ser válida, indicando um ataque de identidade falsa. Outra condição é que os nós não devem encaminhar mensagens que tiverem o mesmo número de seqüência e o mesmo destino de uma mensagem já encaminhada, evitando ataques de replicação. O número de seqüência do pacote RDP é incrementado monotonicamente toda vez que um nó inicia um processo de descoberta de rota. O nó de destino responde somente ao primeiro pacote RDP recebido, descartando os demais que possuam a mesma origem e o mesmo número de seqüência, o que não garante o melhor caminho. O pacote de resposta (*Reply Packet - REP*), que contém o endereço do nó de origem, o certificado do nó de destino, o mesmo número de seqüência e o mesmo *timestamp* do pacote RDP recebido, deve ser assinado antes de ser enviado ao nó de origem. A resposta, então, segue a rota reversa usando um processo similar ao utilizado na descoberta de rota, exceto que agora a comunicação entre os nós é ponto-a-ponto. O nó de origem, ao receber o pacote de resposta, verifica sua autenticidade usando o número de seqüência e a assinatura do nó de destino.

O segundo procedimento previsto pelo protocolo ARAN visa assegurar que a descoberta do caminho ótimo. Assim como no procedimento anterior, o nó de origem envia por difusão um pacote de confirmação de caminho ótimo (*Shortest Path Confirmation - SPC*) assinado para os seus vizinhos, contendo os mesmos campos do RDP mais dois campos adicionais para acomodar o certificado do nó de destino e a mensagem criptografada usando a chave pública do nó de destino. É utilizada uma espécie de roteamento cebola (*onion routing*) [Syverson et al., 1997], onde cada nó que receber a mensagem deve assiná-la, incorporar o seu certificado e criptografar novamente a mensagem usando a chave pública do nó de destino. Quando os pacotes SPC chegam ao nó de destino, este verifica todas as assinaturas e responde ao primeiro pacote SPC e a todos os pacotes posteriores que tiverem percorrido uma rota menor.

O protocolo ARAN ainda possui esquemas que garantem o não-repúdio das mensagens de erro e um processo seguro de revogação de certificados. Assim, em conjunto, os mecanismos de defesa do protocolo oferecem alta proteção contra a maioria dos ataques, embora o uso de criptografia assimétrica exija um alto poder computacional e o ataque do túnel de minhoca não tenha sido solucionado.

2.5.2. Ariadne

O Ariadne [Hu et al., 2005] é um protocolo de roteamento seguro baseado no protocolo reativo DSR (*Dynamic Source Routing*) [Johnson e Maltz, 1996]. O Ariadne se baseia em criptografia simétrica, tendo como vantagem a alta eficiência e simplicidade desse mecanismo. O protocolo provê ainda autenticação ponto-a-ponto das mensagens de roteamento usando um Código de Autenticação de Mensagem (*Message Authentication Code - MAC*) e uma chave secreta compartilhada pelas duas entidades. No entanto, para autenticar mensagens enviadas por difusão é utilizado um esquema de autenticação TESLA similar ao descrito na Seção 2.4.2.

Para garantir a autenticação fim-a-fim do esquema de descoberta de rota, o nó de origem calcula um MAC da mensagem usando a chave secreta que somente os nós de origem e destino conhecem. Dessa maneira, assegura-se que a mensagem veio realmente do nó de origem e que as informações de roteamento não foram alteradas. Antes de enviar a mensagem, o nó de origem estima um tempo máximo para o atraso fim-a-fim e inclui esta informação na mensagem, juntamente com uma lista de nós e uma lista de MACs, ambas inicialmente vazias. Após o término do tempo estimado, o nó de origem irá divulgar sua chave TESLA. Assim, quando uma estação intermediária recebe a mensagem, ela verifica se o tempo de divulgação da chave já expirou. Se o resultado for positivo, ela descarta a mensagem, e, se não for, insere seu endereço na lista de nós. A integridade da lista de endereços é obtida através do mecanismo de cadeias de *hash*, usando um esquema idêntico ao utilizado pelo SAODV. Assim, nesse momento é calculado um novo *hash* sobre o campo *Hash Chain*. A estação intermediária ainda utiliza sua chave TESLA atual para computar o MAC da mensagem, que é inserido na lista de MACs. Finalmente, a mensagem modificada é reenviada para os vizinhos, como no DSR, procedimento que é repetido recursivamente pelas estações intermediárias até chegar ao nó de destino. Quando o nó de destino recebe a mensagem, ele verifica se o valor final da cadeia de *hash* está correto e se as chaves TESLA já foram divulgadas. Caso a mensagem recebida seja válida, o nó de destino calcula o MAC da resposta usando a chave secreta compartilhada com o nó de origem e envia a mensagem de resposta para a origem. Ao final da rota reversa, a origem autentica a resposta antes de aceitá-la.

O protocolo ainda prevê o uso da autenticação TESLA para impedir a fabricação de mensagens de erro falsas e um mecanismo para autenticação da descoberta de rota que permite aos nós limitar a taxa de requisições de descoberta de rota vindas de outros nós. Além disso, uma versão avançada do protocolo possui proteção contra o ataque do túnel de minhoca (Seção 2.3.2), usando o protocolo TIK (*TESLA with Instant Key disclosure*) [Hu et al., 2003a]. Apesar de tantas vantagens, o Ariadne exige alguns pré-requisitos para ser utilizado, como algum mecanismo seguro de distribuição das chaves TESLA dos nós, um esquema para o estabelecimento das chaves secretas compartilhadas pelos nós comunicantes e ainda é requerida uma sincronização de tempo fraca entre os nós que possibilite estimar o tempo de transmissão fim-a-fim para outro nó da rede.

2.5.3. SRP

O SRP (*Secure Routing Protocol*) [Papadimitratos e Haas, 2002] foi projetado para manter o roteamento correto em redes ad hoc onde ocorrem mudanças freqüentes e onde

pode haver nós maliciosos, porém que não agem em conluio. O protocolo foi concebido como uma extensão, que pode ser aplicada em diversos protocolos de roteamento reativos existentes, em particular o DSR [Johnson e Maltz, 1996] e o IERP (*Interzone Routing Protocol*) [Haas et al., 2001]. No SRP, o nó iniciador do procedimento de descoberta de rota é capaz de identificar e descartar respostas contendo informações de roteamento falsas ou ainda evitar recebê-las, garantindo a obtenção de informações topológicas corretas. Para isso, é suposta a existência de uma Associação de Segurança (*Security Association - SA*) entre os nós comunicantes, como uma chave simétrica compartilhada. Além disso, é suposto que os nós possuem uma única interface de rede, com uma correspondência biunívoca entre os endereços IP e o da interface. Sob essas hipóteses, os autores provam que o protocolo é robusto.

Ao iniciar o procedimento de descoberta de rota, o nó de origem deve gerar um MAC, usando uma função *hash* com chave que recebe como argumentos de entrada o cabeçalho IP, os campos básicos da mensagem de roteamento e a chave secreta compartilhada entre os nós de origem e destino. As estações intermediárias são responsáveis por encaminhar a mensagem até o seu destino final. Quando o nó de destino recebe a mensagem de roteamento, ele não somente verifica a integridade da mensagem, como também assegura a autenticidade da origem, uma vez que o MAC só pode ser calculado pelos nós que possuem a chave secreta, garantindo que somente a origem poderia ter computado o MAC recebido. Caso a mensagem recebida seja autêntica e íntegra, o nó destino envia uma mensagem de resposta ao nó de origem realizando o mesmo procedimento feito pelo nó de origem. O nó de origem, ao receber a mensagem de resposta, verifica sua integridade usando o MAC computado pelo nó de destino e descarta a resposta se ela não tiver o mesmo identificador da mensagem inicial.

O protocolo ainda possui um interessante mecanismo de regulação das requisições de descoberta de rota. Cada nó mede as freqüências de requisições realizadas pelos seus vizinhos e mantém uma fila na qual a prioridade de atendimento às requisições é inversamente proporcional à freqüência com que elas são feitas. Assim, fica caracterizado um mecanismo de *feedback* negativo que controla a freqüência de requisições realizadas pelos nós vizinhos, impedindo ataques nos quais o nó malicioso inunda a rede com requisições de descoberta de rota, já que o atacante será atendido por último ou ignorado, dada a sua baixa prioridade de atendimento.

Uma das principais vulnerabilidades do SRP é a ausência de autenticação das mensagens de erro, embora no esquema proposto o nó malicioso só consiga prejudicar rotas às quais ele pertence. Outra desvantagem é que, como o protocolo não previne ações maliciosas em conluio, ele não está imune aos ataques de atração e descarte de pacotes. Apesar disso, o SRP possui a grande vantagem da imunidade aos ataques que modificam a origem do pacote ou simulam identidades. Isso se deve ao protocolo NLP (*Neighbor Lookup Protocol*) de descoberta de vizinhos, integrante do SRP, que mantém um mapeamento dos endereços da subcamada de acesso ao meio (*Medium Access Control*) e da camada de rede dos nós da rede.

2.5.4. SEAD

Os mesmos autores do Ariadne propuseram também mecanismos para garantir a segurança de protocolos de roteamento pró-ativos, desenvolveram o SEAD (*Secure Efficient Ad hoc Distance vector*) [Hu et al., 2002], que é baseado no protocolo de vetor distância DSDV (*Destination-Sequenced Distance Vector routing*) [Perkins e Bhagwat, 1994]. O SEAD foi projetado para suportar a existência de nós com capacidade de processamento limitada e possui defesas contra ataques de negação de serviço nos quais o atacante visa esgotar recursos da vítima, como banda passante e processamento. Para isso, são utilizados mecanismos de segurança reconhecidamente eficientes, como funções *hash* unidirecionais e criptografia simétrica. Foi provada a robustez do protocolo contra ataques ativos não coordenados ou efeitos causados por nós comprometidos.

O SEAD utiliza a técnica de cadeia de *hash* para autenticar os campos número de saltos e número de seqüência. A cadeia é criada ao se aplicar repetidas vezes uma função *hash* a um valor inicial aleatório, e é assumida a existência de algum mecanismo que permita que um nó distribua um elemento autenticado da cadeia para seus vizinhos. Um elemento autenticado da cadeia de *hash* é utilizado para garantir uma atualização segura das mensagens de roteamento. Os elementos posteriores da cadeia podem ser autenticados aplicando-se sobre eles a função *hash* um número adequado de vezes. A distribuição do elemento autenticado não é parte do protocolo, embora os autores sugiram o uso de uma autoridade certificadora confiável para distribuição das chaves públicas dos nós. Dessa maneira, um dado nó poderia usar sua chave pública para assinar o elemento da cadeia de *hash* e distribuí-lo.

São propostos ainda dois mecanismos para autenticação dos nós vizinhos. O primeiro mecanismo utiliza autenticação TESLA (Seção 2.4.2), exigindo sincronização entre os nós da rede. O outro mecanismo supõe a existência de uma chave secreta compartilhada para cada par de nós que desejam se comunicar. Essa chave secreta é usada com um código de autenticação de mensagem MAC para garantir a autenticidade dos nós vizinhos.

O protocolo SEAD não possui proteção contra o ataque do túnel de minhoca (Seção 2.3.2). Porém, como no Ariadne, os autores sugerem o uso do protocolo TIK (*TESLA with Instant Key disclosure*) [Hu et al., 2003a] para detectar esse tipo de ataque. Além disso, dada a natureza unidirecional da função *hash*, o uso da cadeia de *hash* impede que nós maliciosos diminuam o campo de número de saltos, tentando forjar rotas melhores do que elas realmente são. Outra vantagem provém do uso de mecanismos de autenticação de vizinhos, que protegem contra ataques de Identidade Falsa.

2.5.5. SAODV

O *Secure AODV (SAODV)* [Zapata, 2002] é uma extensão do protocolo AODV (*Ad Hoc On-Demand Distance Vector Routing Protocol*) [Perkins et al., 2003], que garante segurança no processo de descoberta de rota. Zapata e Asokan propõem adicionar algumas mensagens chamadas de "Extensão de Assinatura" (*Signature Extension*), no intuito de acrescentar funcionalidades ao protocolo que permitam assegurar a integridade, a autenticação e o não-repúdio das informações de roteamento.

O SAODV possui dois mecanismos de proteção das mensagens de roteamento:

assinatura digital e cadeias de *hash*. A assinatura digital garante autenticação fim-a-fim dos campos imutáveis da mensagem, que devem ser assinados pelo nó de origem antes do envio da mensagem. No entanto, o campo Número de Saltos (*Hop Count*) deve ser decrementado a cada salto pelas estações intermediárias, o que impossibilita a adoção da abordagem anterior. Nesse caso, a cadeia de *hash* é usada para autenticar o campo Número de Saltos a cada salto, garantindo a integridade do vetor de distância. Ao enviar uma mensagem, o nó de origem deve inicializar o campo *Hash* com uma semente aleatória, o campo Número Máximo de Saltos com o valor desejado e o campo *Top Hash* com o resultado da aplicação da função *hash* sobre a semente um número de vezes igual a Número Máximo de Saltos . Quando um nó intermediário receber a mensagem, poderá autenticar o campo Número de Saltos, aplicando a mesma função *hash* um número de vezes igual a diferença entre Número Máximo de Saltos e Número de Saltos sobre o campo *Hash* e comparando o resultado com o campo *Top Hash*. Se os campos foram iguais, pode-se assegurar que o campo Número de Saltos não foi alterado e, então, o nó intermediário deverá incrementar o campo Número de Saltos em 1 e aplicar a função *hash* no campo *Hash* antes de reencaminhar a mensagem. Caso contrário, a mensagem deverá ser descartada. Dessa maneira, o mecanismo de cadeias de *hash* impede que um nó malicioso diminua o Número de Saltos da mensagem de roteamento, uma vez que não se pode obter os valores anteriores da seqüência, dada a propriedade unidirecional da função *hash*.

O SAODV ainda prevê extensões de assinaturas duplas (*Double Signature Extension*), que permitem que nós intermediários respondam imediatamente à origem quando eles já possuírem uma rota atualizada para o destino, assim como é previsto no AODV original. Além disso, o SAODV propõe que todos os nós devem possuir um meio de armazenar os seus números de seqüência, mesmo quando são reiniciados, a fim de evitar ataques de número de seqüência que atrairiam mais tráfego para o nó malicioso, por suas rotas parecerem mais atualizadas. Finalmente, o protocolo ainda propõe um mecanismo de autenticação das mensagens de erro.

2.5.6. SLSP

O protocolo seguro baseado em estado do enlace denominado SLSP (*Secure Link State Routing Protocol*) [Papadimitratos e Haas, 2003] foi proposto pelos mesmos autores do SRP. É suposta a existência de um par de chaves assimétricas para cada interface de rede de um nó e de um sistema de gerenciamento das chaves públicas dos nós. O SLSP é composto por três procedimentos: distribuição de chaves públicas, descoberta de vizinhos e atualização dos estados dos enlaces.

O procedimento de distribuição das chaves públicas dos nós é feito de forma distribuída, a fim de evitar o uso de um servidor central de gerenciamento de chaves. Cada nó envia por difusão (*broadcast*) um pacote de distribuição de chaves públicas (*Public Key Distribution packet - PKD*) assinado para os vizinhos da sua zona. Esse pacote contém o certificado de chave pública do nó de origem que poderá ser usado posteriormente para autenticar os pacotes oriundos daquela fonte.

Assim como no SRP, é utilizado o protocolo NLP (*Neighbor Lookup Protocol*) de descoberta de vizinhos. Cada nó envia aos seus vizinhos uma mensagem de *hello* assinada contendo seus endereços MAC (*Medium Access Control*) e IP. Ao receberem a

mensagem, os vizinhos validam sua assinatura e, em seguida, a aceitam. Dessa maneira, o protocolo NLP mantém um mapeamento dos endereços MAC e IP dos nós da rede e fica responsável por enviar notificações no caso de ocorrência de discrepâncias, como o mesmo MAC possuir dois ou mais endereços IP ou um nó tentando obter o endereço MAC de um nó existente.

O procedimento de atualização dos estados dos enlaces utiliza pacotes denominados LSU (*Link State Update*), identificados pelo endereço IP do nó de origem e um número de seqüência. Assim como no SEAD e no SAODV, o número de saltos, indicado pelo campo *hops_traversed*, é autenticado usando a técnica de cadeia de *hash*. Assim, ao receber um pacote LSU, os nós devem atualizar o campo *hops_traversed*, substituindo-o pelo seu valor *hash* e, em seguida, reencaminhá-lo. Antes disso, porém, a assinatura do pacote recebido deve ser verificada usando a chave pública obtida no procedimento de distribuição de chaves públicas.

Os três procedimentos e o protocolo NLP, em conjunto, garantem segurança no processo de descoberta de topologia e proteção contra ataques que modificam a origem do pacote ou simulam identidades. O protocolo SLSP também possui um mecanismo de regulação da taxa de recepção de pacotes idêntico ao do SRP, oferecendo proteção contra ataques de negação de serviço. Apesar de todas essas vantagens, o SLSP é vulnerável a ataques em conluio nos quais os atacantes fabricam enlaces inexistentes entre eles e inundam a rede com as informações falsas.

2.5.7. Extensões a Protocolos de Roteamento

Esta seção apresenta algumas extensões para os protocolos de roteamento existentes. Apesar de nem todas as propostas constituírem um protocolo completo, elas podem ser aplicadas aos protocolos que já existem de forma a garantir segurança nas tarefas de roteamento.

SOLSR

O SOLSR (*Secure Optimized Link State Routing Protocol*) [Hafslund et al., 2004] é uma versão segura do protocolo de roteamento OLSR [Jacquet et al., 2001], cuja idéia é assinar, usando chaves simétricas, cada pacote de controle do OLSR a fim de garantir a autenticidade das mensagens.

Uma vantagem do SOLSR é que a autenticação é feita salto a salto. Isso significa que é garantida a segurança até de campos que devem ser atualizados pelos nós intermediários, como o número de saltos e o campo TTL (*time-to-tive*). Além disso, somente é necessária uma assinatura por salto, já que muitas mensagens de roteamento são encapsuladas em um único pacote do OLSR. Em compensação, a abordagem salto a salto não garante assinaturas fim-a-fim, já que um pacote recebido por um nó não terá sido assinado pelo nó de origem, mas pelo nó anterior. Apesar disso, o protocolo determina que os nós só devem encaminhar pacotes oriundos de nós confiáveis. Conseqüentemente, os nós de uma dada rota serão confiáveis, dois a dois. O processo de assinatura digital utiliza uma função *hash* com chave, de forma que um nó que não tenha acesso à chave secreta não poderá reproduzir a assinatura do nó emissor.

O SOLSR possui mensagens próprias para acomodar as assinaturas, de forma a

garantir a compatibilidade com nós que não estejam operando a versão segura do OLSR. Além disso, para evitar ataques de replicação, o SOLSR utiliza a técnica de *timestamp*. Foi proposto um mecanismo bidirecional de troca de *timestamps* que possui a vantagem de não exigir sincronização dos nós.

SAR

O protocolo SAR (*Security-aware Ad hoc Routing*) [Yi et al., 2001] torna seguros os processos de descoberta e manutenção de rotas de protocolos de roteamento reativos como o AODV e o DSR. A idéia básica da proposta é incorporar métricas de segurança às mensagens de roteamento no intuito de estabelecer um nível de "qualidade de segurança" que será usado na comunicação entre os nós da rede ad hoc. O SAR trata os requisitos de segurança de forma semelhante àquela com que os requisitos de qualidade de serviço (*Quality of Service - QoS*) são tratados.

As métricas de segurança do protocolo podem ser especificadas por níveis hierárquicos de confiança entre os nós ou por requisitos de segurança como autenticidade e não-repúdio. Os níveis de confiança podem ser definidos a partir de um sistema de distribuição de chaves ou de compartilhamento de uma chave secreta entre os nós. Assim, somente os nós que pertencem a um determinado nível de confiança podem trocar mensagens entre si, uma vez que nós de outros níveis de confiança não terão como decriptografar as mensagens daquele nível. Por outro lado, os requisitos de segurança podem ser implementados usando técnicas conhecidas como certificação e números de seqüência.

Uma implementação do protocolo SAR baseada no AODV também foi proposta pelos autores. O *Security-aware AODV* acrescenta às mensagens de roteamento do AODV campos que definem a métrica de segurança utilizada. Assim, somente os nós que possuem o nível de segurança especificado podem ser incorporados durante o processo de descoberta de rota. Dessa forma, ao receber um pacote de requisição de rota, o nó de destino pode ter a certeza de que existe uma rota para o nó de origem e que os componentes dessa rota podem respeitar o nível de segurança estabelecido pelo nó de origem.

TIARA

A proposta chamada TIARA (*Techniques for Intrusion Resistant Ad hoc Routing Algorithms*) [Ramanujan e Edin, 2000] consiste em um conjunto de princípios de projeto e técnicas que podem ser aplicadas aos protocolos de roteamento atuais, em especial aos protocolos reativos como o AODV e o DSR, para prover resistência a ataques de negação de serviço.

O primeiro princípio, denominado controle de acesso a rotas baseado em fluxo (*Flow-based Route Access Control - FRAC*), utiliza uma lista de controle de acesso, armazenada em cada nó, que identifica os fluxos de mensagens que possuem autorização para serem encaminhados. O roteamento por múltiplos caminhos é outra técnica utilizada pelo TIARA. Nesse caso, os procedimentos de descoberta e manutenção de rota compulsoriamente trabalham para que todas as rotas para um dado fluxo sejam encontradas e mantidas, o que assegura tolerância a falhas causadas por nós intrusos. O terceiro princípio, denominado roteamento de fluxo iniciado pela origem (*Source-Initiated Flow Routing*), diz que a origem de cada pacote deve inserir um rótulo que indique qual dos múltiplos caminhos deve ser tomado. O TIARA também utiliza um mecanismo de mo-

nitoramento de fluxo, que exige que o nó de origem transmita periodicamente mensagens que indiquem o estado do fluxo de pacotes. Com isso, o nó de destino, que deve monitorar os fluxos ativos dos quais ele participa, pode armazenar os pacotes recebidos entre as mensagens de estado do fluxo. Caso a diferença entre o número de pacotes recebidos pelo destino e o número de pacotes enviados pela origem seja muito grande ou o tempo de espera da mensagem de estado do fluxo ultrapasse um limite estipulado, é assumido que houve uma falha na rota. Para autenticação dos pacotes, o protocolo utiliza um mecanismo denominado autenticação rápida (*Fast Authentication*), que obriga os nós a colocar o rótulo do caminho numa localização secreta em cada pacote transmitido. Essa localização secreta é determinada no estabelecimento da rota entre os nós comunicantes e deve ser diferente para cada nó. O TIARA utiliza também números de seqüência a fim de evitar ataques de replicação. O último mecanismo proposto, denominado mecanismo de alocação de recursos baseado em referência, determina a quantidade máxima de recursos que cada nó pode alocar para a transmissão de um determinado fluxo. Alocações adicionais de recursos só são permitidas no caso de o nó de origem apresentar recomendações oriundas de nós confiáveis que garantam a autenticidade do seu pedido.

BISS

O protocolo BISS (*Building secure routing out of an Incomplete Set of Security associations*) [Capkun e Hubaux, 2003] é um conjunto de otimizações que podem ser aplicadas aos protocolos de roteamento existentes. A proposta dos autores é a construção de um protocolo de roteamento seguro onde não haja necessidade de associações seguras entre todos os pares de nós, mas apenas entre uma fração deles. Para atingir tal objetivo, é proposto que a autenticação dos nós intermediários ao longo do processo de descoberta de rota deve ser realizada não somente através das associações de segurança pré-existentes, mas também através da troca de certificados de chave pública entre os nós.

Em uma aplicação do BISS ao DSR, os pacotes de requisição de rota devem conter o certificado da origem assinado por uma autoridade certificadora e chave pública do nó de origem, que deve assinar o pacote antes de enviá-lo aos seus vizinhos. Os nós intermediários devem verificar a assinatura do nó anterior e, usando uma associação segura pré-existente, autenticar o destino. Assim como no Ariadne, ao reencaminhar o pacote, os nós intermediários devem calcular o MAC do pacote usando a chave secreta que compartilham com o destino, permitindo a este a verificação da autenticidade dos nós da rota. Em seguida, o nó de destino deve enviar um pacote de resposta ao nó de origem. Se o destino compartilhar uma chave secreta com a origem, a resposta será protegida com um MAC calculado com essa chave. Caso contrário, o pacote de resposta deve ser assinado para que o nó de origem possa autenticá-lo. Os nós intermediários devem proceder da mesma forma. Assim, o nó de origem é capaz de autenticar o pacote de resposta, verificando os MACs e as assinaturas incluídas pelo nó de destino e os nós intermediários. De forma análoga, as mensagens de erro serão autenticadas por um MAC, se os nós compartilharem uma chave secreta, ou usando assinatura digital, caso não possuam uma associação de segurança pré-existente, mas possam usar os certificados de chave pública correspondentes.

O BISS possui a vantagem de aumentar o número de associações seguras da rede, pois chaves e certificados anteriormente desconhecidos podem ser distribuídos ao longo do processo de descoberta de rota. Esse método permite também o estabelecimento de

chaves simétricas entre os nós, que podem ser utilizadas para futuras verificações de mensagens.

SMT

O projeto de um sistema completo de segurança deve abranger as tarefas de roteamento e encaminhamento de dados em uma rede ad hoc. Apesar de ser uma condição necessária, apenas o roteamento seguro não é suficiente para atingir tal objetivo, pois os protocolos de roteamento não garantem que os nós de uma rota corretamente descoberta irão encaminhar as mensagens de dados posteriores da forma esperada. Um nó malicioso pode enganar o sistema de segurança, fornecendo informações corretas durante o processo de descoberta de rota, mas se comportando indevidamente durante a fase de encaminhamento de dados.

O SMT (*Secure Message Transmission protocol*), proposto por Papadimitratos e Haas, também é uma extensão que se aplica aos protocolos de roteamento, mas tendo como principal objetivo garantir o encaminhamento seguro dos dados, após o estabelecimento da rota entre a origem e o destino, utilizando informações de roteamentos para determinar quais são as rotas seguras. O SMT aproveita algumas vantagens do ambiente ad hoc para conseguir um encaminhamento de dados seguro e tolerante a falhas, utilizando as redundâncias topológicas características da rede ad hoc e criptografia simétrica para alcançar tal objetivo. Além disso, é suposta uma associação segura somente entre os nós de origem e destino.

O SMT utiliza as informações do protocolo de roteamento seguro operante para determinar um conjunto de caminhos que ligam o nó de origem ao de destino. Em seguida, a mensagem a ser transmitida é dividida em partes, segundo o algoritmo de inserção de redundância limitada de Rabin [Rabin, 1989], de forma que mesmo a recepção de apenas uma fração da mensagem original possibilitará sua reconstrução pelo nó de destino. Cada parte da mensagem original é transmitida por um caminho diferente e possui um cabeçalho com informações de criptografia que permitem ao nó de destino garantir a autenticidade e a integridade dos dados recebidos, além de assegurar proteção contra ataques de replicação. Após receber algumas partes da mensagem original, o nó de destino envia mensagens de reconhecimento ao nó de origem informando quais partes permaneceram intactas e quais rotas são confiáveis. Para garantir a robustez desse mecanismo de *feedback*, as mensagens de reconhecimento são protegidas da mesma forma que as mensagens de dados e são uniformemente distribuídas ao longo das rotas reversas confiáveis. Dessa maneira, o retorno de uma única mensagem de reconhecimento é suficiente para o correto funcionamento do protocolo. Caso o nó de destino não receba um número suficiente de partes para reconstruir a mensagem original, o nó de origem deve retransmitir as partes restantes através das rotas confiáveis. Além disso, se o protocolo determinar que as rotas inicialmente escolhidas não foram suficientemente boas, um novo conjunto de rotas deve ser escolhido.

O protocolo SMT tem como vantagem o fato de não ser exigido nenhum processamento adicional dos nós intermediários, que devem simplesmente reencaminhar os pacotes recebidos. Além disso, os autores mostram que o protocolo consegue 100% de sucesso de recepção das mensagens transmitidas, ainda que haja um ambiente altamente hostil, como o caso em que 20% dos nós da rede são maliciosos. Por outro lado, o uso de

múltiplos caminhos e, conseqüentemente, o grande número de nós envolvidos na transmissão de uma única mensagem é o preço pago para se obter a robustez desejada.

2.6. Protocolos Baseados em Reputação

Uma defesa contra ações egoístas e maliciosas é a adoção de um sistema de reputação. O objetivo dessa abordagem é incentivar o comportamento que leve a uma confiança crescente entre os nós. Para isso, são realizadas decisões baseadas na confiança entre as entidades. Em geral, as propostas utilizam monitoramento passivo das transações, troca de recomendações entre nós e algum mecanismo de geração de mensagens de alarme. A seguir, são descritos os principais protocolos baseados em reputação encontrados na literatura.

2.6.1. OSRP

O OSRP (*On-demand Secure Routing Protocol Resilient to Byzantine Failures*) [Baruch Awerbuch e Rubens, 2002] é um protocolo reativo que provê resistência contra falhas bizantinas (Seção 2.3.2) causadas por nós individuais ou em conluio. A idéia básica do protocolo é atribuir pesos crescentes aos caminhos onde houver detecção de falhas bizantinas, de forma que se possa descobrir um caminho livre de falhas, caso exista, ainda que haja nós maliciosos agindo em conluio na rede. Assim, ao invés de identificar o nó malicioso em si, o OSRP visa atribuir um peso alto aos seus enlaces. O protocolo é dividido em três partes: descoberta de rota livre de colisões, detecção de falhas bizantinas e gerenciamento de peso dos enlaces.

A primeira fase é responsável por encontrar a rota de menor peso entre os nós de origem e de destino. Para isso, é usada a técnica de assinatura digital, que requer a existência de uma infra-estrutura de chave pública que certifique a autenticidade das chaves públicas dos nós da rede. Ao iniciar o processo de descoberta de rota, o nó de origem envia, por difusão (*broadcast*), uma mensagem de requisição de rota assinada para todos os seus vizinhos. Essa mensagem contém os endereços dos nós de origem e destino, um número de seqüência e uma lista de pesos. O próximo nó que receber a mensagem irá verificar sua assinatura, adicionar a requisição de rota a sua lista e reencaminhar a mensagem, procedimento que será repetido por todos os nós intermediários até que a mensagem atinja seu destino final. Quando receber a mensagem de requisição de rota, o nó de destino irá verificar sua assinatura e, em seguida, enviar uma mensagem de resposta assinada aos seus vizinhos. A mensagem de resposta irá percorrer a rota reversa e cada nó intermediário deverá calcular o peso total do caminho até o nó em questão. Caso o valor calculado seja menor que aquele encontrado em uma mensagem de resposta anterior com o mesmo número de seqüência, então o nó intermediário deverá verificar todas as assinaturas, anexar o seu endereço, assinar a mensagem e reenviá-la em difusão. O nó de origem deve realizar esse mesmo procedimento ao receber uma mensagem de resposta, atualizando o destino da rota caso o caminho seja menor que o atual.

A fase de detecção de falhas bizantinas tem como objetivo descobrir enlaces com falhas no caminho obtido na fase anterior. Para isso, em cada pacote de dados é inserida uma lista de nós, denominados nós de teste (*probe nodes*). Os nós de teste devem enviar ao nó de origem um reconhecimento para cada pacote recebido. Caso o número

de pacotes sem reconhecimento ultrapassar um limite estipulado, uma falha é registrada. Para garantir a autenticidade e a integridade dos reconhecimentos, o protocolo requer a existência de chaves secretas compartilhadas entre o nó de origem e cada nó de teste.

A terceira fase do OSRP é responsável por manter uma lista de pesos dos enlaces de acordo com o resultado da fase anterior. Essa lista de pesos é utilizada pela primeira fase do protocolo para evitar os caminhos com falhas. Nessa fase, o peso dos enlaces com falha é dobrado, de forma a penalizar o comportamento bizantino do nó daquele enlace.

As três fases do protocolo OSRP são executadas seqüencialmente e a resposta de uma fase é utilizada como entrada para outra. Sob as hipóteses impostas, os autores provam que o protocolo é resistente a falhas bizantinas. Porém, o OSRP é incapaz de prevenir ataques de atração de tráfego, caso eles sejam realizados sem um comportamento bizantino.

2.6.2. CONFIDANT

O CONFIDANT (*Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks*) [Buchegger e Boudec, 2002] é outro protocolo baseado em reputação encontrado na literatura recente. O protocolo é composto por quatro módulos e foi proposto como uma extensão de segurança para o protocolo de roteamento DSR [Johnson e Maltz, 1996].

O primeiro módulo é responsável pelo monitoramento passivo dos reconhecimentos de cada mensagem que um nó encaminha. Quando um nó transmite uma mensagem, ele tenta monitorar, usando o modo promíscuo da sua interface de rede, a transmissão do próximo nó a fim de assegurar que a mensagem foi reencaminhada devidamente. O próximo módulo, denominado gerenciador de confiança (*trust manager*), é responsável pelo envio e pelo recebimento de mensagens de alarme. Essas mensagens são trocadas por nós que foram pré-definidos como confiáveis e visam informar comportamentos indevidos na rede. O terceiro módulo consiste em um sistema de reputação que reúne as informações geradas pelos módulos anteriores e atribui notas aos nós, mantendo uma tabela de reputações para cada nó da rede. As notas são atribuídas de acordo com um critério que dá maior importância às experiências locais do que as mensagens de alerta recebidas de outros nós. Se a nota de certo nó ultrapassar um limite de segurança, o último módulo, denominado de gerenciador de rotas, é chamado para remover os caminhos que contiverem o nó com comportamento suspeito. Para isso, ele utiliza a tabela de reputação dos nós, escolhendo o melhor caminho a ser tomado. Além disso, são ignoradas as mensagens de roteamento oriundas de nós maliciosos e são gerados alertas para os nós legítimos avisando quando é feito um pedido de requisição de rota que usa um caminho comprometido.

É importante notar que o protocolo CONFIDANT suporta somente experiências negativas, que são apagadas após um determinado tempo sem detecção de ações suspeitas. Além disso, os nós da rede ad hoc devem operar os quatro módulos do protocolo para o seu correto funcionamento. Dessa maneira, o CONFIDANT é capaz de detectar, alertar e reagir a ataques no encaminhamento das mensagens de dados e de roteamento, embora a utilização do modo promíscuo esteja sujeita a críticas, devido às limitações de energia dos nós da rede.

2.7. Tendências Futuras

Este capítulo apresentou as principais vulnerabilidades e ataques às redes ad hoc, classificando-os segundo os efeitos que causam e as camadas da pilha de protocolos nas quais eles atuam. Foram então apresentados os mecanismos de segurança propostos na literatura, ressaltando-se as vantagens e desvantagens de cada um. Destaca-se entre os mecanismos um novo método de confiabilidade, que permite que um grau maior de segurança seja obtido através da avaliação do comportamento dos nós vizinhos. Por fim, foram selecionados e descritos alguns dos principais protocolos recentemente propostos. Essa seção conclui este trabalho descrevendo as linhas de pesquisa atuais e as tendências futuras da segurança de redes ad hoc.

Antes de apresentar as tendências futuras da segurança de MANETs, deve-se salientar algumas limitações das técnicas atuais. Em primeiro lugar, os mecanismos de segurança propostos recentemente foram projetados para proteger a rede ad hoc somente contra ataques conhecidos. No entanto, espera-se que surjam novos ataques que explorem as vulnerabilidades das técnicas existentes. Dessa maneira, não há uma solução completa, sendo necessários novos esforços cada vez que novas vulnerabilidades foram descobertas. Em segundo lugar, cada mecanismo de segurança acrescenta *overhead* e complexidade. Conseqüentemente, a limitação de recursos dos nós da MANET impede a ativação simultânea de todos os mecanismos existentes. Por último, as técnicas que usam chaves criptográficas necessitam de um sistema distribuído, robusto e seguro que garanta a geração, a distribuição e o gerenciamento dessas chaves. Essa é outra questão de segurança ainda aberta, podendo-se encontrar diversos trabalhos publicados recentemente nessa área de pesquisa.

O rápido crescimento das redes ad hoc móveis nos últimos anos, provendo soluções satisfatórias para os problemas técnicos a que se propunham, leva a crer num futuro promissor para essa tecnologia. Entretanto, há questões de segurança ainda não completamente solucionadas, em especial às que se referem as características do meio físico e do IEEE 802.11. Entre os ataques específicos de redes ad hoc, as soluções propostas são eficientes, embora a premissa de existir um método de autenticação automático e seguro ainda seja um problema. Além disso, os requisitos de segurança são dependentes das aplicações e dos ambientes, que podem variar significativamente. Por exemplo, as MANETs podem ser utilizadas tanto para prover conexão sem fio flexível e de curto alcance para estabelecimentos comerciais, como para conectar dispositivos militares capazes de realizar operações criptográficas de alto desempenho. Enquanto que aplicações militares podem exigir altos níveis de segurança, outras aplicações precisam de requisitos de segurança simples. Além disso, acredita-se que, no futuro, diversos dispositivos e usuários terão de coexistir numa rede ad hoc grande, aberta, ubíqua e autônoma. Dessa forma, uma linha de pesquisa atual visa encontrar soluções eficientes para esse problema.

Outra linha de pesquisa recente procura mecanismos que consigam proteger a rede ad hoc contra nós comprometidos ou com comportamento indesejável. Esses mecanismos não devem exigir alto poder computacional, devem fazer suposições limitadas acerca das relações de confiança entre os nós e ainda agregar um baixo *overhead*. Nesse sentido, destacam-se os sistemas de confiança, como o novo método apresentado na seção 2.4.3.

Outra questão ainda aberta é o roteamento seguro (seção 2.5). Mecanismos exis-

tentes, como o IPSec, não oferecem uma solução completa para o problema. A literatura apresenta duas abordagens para resolver essa questão: esquemas pró-ativos de combate aos ataques existentes ou sistemas de detecção de intrusão. Porém as propostas existentes agregam bastante *overhead* e computação aos protocolos, sem conseguir oferecer uma solução satisfatória para todos os ataques conhecidos.

Alguns estudos que vêm sendo feitos atualmente visam inserir redundâncias nos canais de comunicação de forma a reduzir o impacto de ataques de negação de serviço e combater comportamentos maliciosos, usando múltiplos caminhos entre a fonte e o destino do fluxo de dados.

Outros pontos em aberto estão ligados ao controle de acesso aos serviços da rede (nível de aplicação) e aos grupos de encaminhamento de pacotes (nível de transporte), ao estabelecimento de chaves de grupo para uma rede com topologia arbitrária, ou ainda à aplicação das técnicas atuais aos campos emergentes das redes de sensores ad hoc e das redes ad hoc veiculares (*Vehicular Ad hoc NETworks - VANETs*).

Finalmente, conforme argumentado na Seção 2.1, a natureza das MANETS possui vulnerabilidades intrínsecas que não podem ser removidas. Assim, continua sendo um campo interessante a busca por soluções que possam reunir as propostas atuais de forma a garantir um nível mínimo de segurança que cada aplicação exige, sem gerar uma sobrecarga da rede.

Referências

[Anderson e Kuhn, 1996] Anderson, R. e Kuhn, M. (1996). Tamper resistance - a cautionary note. Em *Second USENIX Workshop en Electronic Commerce*.

[Atkinson, 1995] Atkinson, R. (1995). *Security architecture for the internet protocol*. RFC 1825.

[Baruch Awerbuch e Rubens, 2002] Baruch Awerbuch, David Holmer, C. N.-R. e Rubens, H. (2002). An on-demand secure routing protocol resilient to byzantine failures. Em *ACM Workshop on Wireless Security (WiSe)*, Atlanta, Georgia.

[Bellare et al., 1996] Bellare, M., Canetti, R. e Krawczyk, H. (1996). Keying hash functions for message authentication. *Lecture Notes in Computer Science*, 1109.

[Bellare et al., 1994] Bellare, M., Kilian, J. e Rogaway, P. (1994). The security of cipher block chaining. *Lecture Notes in Computer Science*, 839:341–358.

[Bing, 2006] Bing, B. (2006). A fast and secure framework for over-the-air wireless software download using reconfigurable mobile devices. *IEEE Communications Magazine*, 44(6).

[Blakley, 1979] Blakley, G. R. (1979). Safeguarding cryptographic keys. Em *National Computer Conference (AFIPS)*, volume 48, páginas 313–317.

[Boneh e Franklin, 2001] Boneh, D. e Franklin, M. (2001). Identity-based encryption from the weil pairing. Em *21st Annual International Cryptology Conference on Advances in Cryptology - CRYPTO '01*, páginas 213–229.

[Buchegger e Boudec, 2002] Buchegger, S. e Boudec, J. L. (2002). Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). Em *The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing*, páginas 226–236.

[Buttyan e Hubaux, 2000] Buttyan, L. e Hubaux, J. P. (2000). Enforcing service availability in mobile ad-hoc wans. Em *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, USA.

[Buttyan e Hubaux, 2003] Buttyan, L. e Hubaux, J. P. (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5):579–592.

[Capkun e Hubaux, 2003] Capkun, S. e Hubaux, J. (2003). BISS: building secure routing out of an incomplete set of secure associations. Em *2nd ACM Wireless Security (WiSe'03)*, páginas 21–29.

[Cha e Cheon, 2003] Cha, J. C. e Cheon, J. H. (2003). An identity-based signature from gap diffie-hellman groups. Em *Practice and Theory in Public Key Cryptography (PKC 2003)*, volume 2567, páginas 18–30.

[Chan et al., 2003] Chan, H., Perrig, A. e Song, D. (2003). Random key predistribution schemes for sensor networks. Em *IEEE Symposium on Security and Privacy*, páginas 197–213.

[Daemen e Rijmen, 2002] Daemen, J. e Rijmen, V. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag.

[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.

[Douceur, 2002] Douceur, J. R. (2002). The sybil attack. Em *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, páginas 251–260.

[Eschenauer e Gligor, 2002] Eschenauer, L. e Gligor, V. D. (2002). A keymanagement scheme for distributed sensor networks. Em *9th ACM Conference on Computer and Communication Security*, páginas 41–47.

[Frankel e Desmedt, 1992] Frankel, Y. e Desmedt, Y. (1992). Parallel reliable threshold multisignature. TR 92-04-02, University of Wisconsin.

[Gahlin, 2004] Gahlin, C. (2004). Secure ad hoc networking. Master's thesis, University of Umeå. Work in progress.

[Ganesan et al., 2001] Ganesan, D., Govindan, R., Shenker, S. e Estrin, D. (2001). Highly resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review*, 5.

[Gennaro et al., 1996] Gennaro, R., Jarecki, S., Krawczyk, H. e Rabin, T. (1996). Robust threshold DSS signatures. Em *Advances in Cryptology - Eurocrypt '96*, páginas 354–371.

[Haas et al., 2001] Haas, Z., Pearlman, M. e Samar, P. (2001). *The Interzone Routing Protocol (IERP) for Ad Hoc Networks*. IETF MANET working group.

[Hafslund et al., 2004] Hafslund, A., Tønnesen, A., Rotvik, R. B., Andersson, J. e Øivind Kure (2004). Secure extension to the olsr protocol. Em *OLSR Interop and Workshop*, páginas 1–4, San Diego, California.

[He et al., 2004] He, Q., Wu, D. e Khosla, P. (2004). SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. Em *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Atlanta, E.U.A.

[Hu et al., 2002] Hu, Y.-C., Johnson, D. B. e Perrig, A. (2002). SEAD: Secure efficient distance vector routing in mobile wireless ad hoc networks. Em *Fourth IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA '02), páginas 3–13.

[Hu et al., 2003a] Hu, Y.-C., Perrig, A. e Johnson, D. B. (2003a). Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. Em *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, volume 3, páginas 1976–1986.

[Hu et al., 2003b] Hu, Y.-C., Perrig, A. e Johnson, D. B. (2003b). Rushing attacks and defense in wireless ad hoc network routing protocols. Em *Second ACM Workshop on Wireless Security (WiSe 03)*, páginas 30–40.

[Hu et al., 2005] Hu, Y.-C., Perrig, A. e Johnson, D. B. (2005). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11(1–2):21–38.

[IEEE, 2000] IEEE (2000). *IEEE Standard Specifications for Public-Key Cryptography*. IEEE Std 1363-2000.

[Jacquet et al., 2001] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A. e Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. Em *5th IEEE Multi Topic Conference (INMIC 2001)*, páginas 62–68.

[Johnson e Maltz, 1996] Johnson, D. B. e Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks, mobile computing. Em *Kluwer Academic Publishers*, volume 353, páginas 153–181. Mobile Computing (ed. T. Imielinski and H. Korth).

[Kaliski e Staddon, 1998] Kaliski, B. e Staddon, J. (1998). *PKCS #1: RSA Cryptography Specifications Version 2.0*. RFC 2437.

[Karlof e Wagner, 2003] Karlof, C. e Wagner, D. (2003). Secure routing in wireless sensor networks: attacks and countermeasures. *IEEE International Workshop on Sensor Network Protocols and Applications 2003*, páginas 113–127.

[Khalili et al., 2003] Khalili, A., Katz, J. e Arbaugh, W. A. (2003). Toward secure key distribution in truly ad-hoc networks. Em *Applications and the Internet Workshops (SAINT'03 Workshops)*, páginas 342–346.

[Kong et al., 2001] Kong, J., Zerfos, P., Luo, H., Lu, S. e Zhang, L. (2001). Providing robust and ubiquitous security support for mobile ad-hoc networks. Em *Ninth International Conference on Network Protocols (ICNP'01)*, páginas 251–260.

[Kulkarnia et al., 2006] Kulkarnia, S. S., Goudab, M. G. e Arora, A. (2006). Secret instantiation in ad hoc networks. Em *Computer Communicatons 29*, páginas 200–215, Oakland, California.

[Lamport et al., 1982] Lamport, L., Shostak, R. e Pease, M. (1982). The byzantine generals problem. Em *ACM Transactions on Programming Languages and Systems (TOPLAS)*, volume 4 of *3*, páginas 382–401.

[Liu et al., 2004] Liu, Z., Joy, A. W. e Thompson, R. A. (2004). A dynamic trust model for mobile ad hoc networks. Em *IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)*, Suzhou, Chine.

[McKnight e Chervany, 2000] McKnight, D. H. e Chervany, N. L. (2000). What is trust? A conceptual analysis and an interdisciplinary model. Em *Americas Conference on Information Systems (AMCIS 2000)*, Long Beach, USA.

[Merkle, 1980] Merkle, R. C. (1980). Protocols for public key cryptosystems. Em *IEEE Symposium on Security ad Privacy*, páginas 122–133.

[Murthy e Mano, 2004] Murthy, C. e Mano, B. (2004). *Ad Hoc wireless networks: architetures and protocols*. Prentice Halll Professional Technical Reference.

[National Bureau of Standards, 1977] National Bureau of Standards (1977). *Data Encryption Standard*. FIPS-Pub.46.

[National Institute of Standards, 2000] National Institute of Standards (2000). *Secure hash standard*. FIPS 180-2.

[Newsome et al., 2004] Newsome, J., Shi, E., Song, D. e Perrig, A. (2004). The sybil attack in sensor networks: Analysis & defenses. Em *3rd IEEE/ACM Information Processing in Sensor Networks 2004 - IPSN 04*, páginas 259–268.

[Nichols e Lekkas, 2002] Nichols, R. K. e Lekkas, P. C. (2002). *Wireless Security Models, Threats, and Solutions*. McGraw-Hill.

[Papadimitratos e Haas, 2002] Papadimitratos, P. e Haas, Z. (2002). Secure routing for mobile ad hoc networks.

[Papadimitratos e Haas, 2003] Papadimitratos, P. e Haas, Z. (2003). Secure link state routing for mobile ad hoc networks. Em *IEEE CS Workshop on Security and Assurance in Ad hoc Networks*, páginas 379–38.

[Pease et al., 1980] Pease, M., Shostak, R. e Lamport, L. (1980). Reaching agreement in the presence of faults. Em *Journal of ACM 27*, volume 2, páginas 228–234.

[Perkins e Bhagwat, 1994] Perkins, C. E. e Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *Sigcomm 94*.

[Perkins et al., 2003] Perkins, C. E., M.Belding-Royer, E. e Das, R. S. (2003). *Ad Hoc On-Demand Distance Vector Routing*. Request for Comments: 3561.

[Perrig et al., 2002] Perrig, A., Canetti, R., Tygar, D. e Song, D. (2002). The TESLA broadcast authentication protocol. *Cryptobytes*, 5(2):2–13.

[Pirzada e McDonald, 2004] Pirzada, A. A. e McDonald, C. (2004). Establishing trust in pure ad-hoc networks. Em *27th Australasian Computer Science Conference (ACSC'04)*, Dunedin, New Zealand.

[Rabin, 1989] Rabin, M. O. (1989). Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of ACM*, 36(2).

[Ramanujan e Edin, 2000] Ramanujan, R. e Edin, R. (2000). TIARA: Techniques for intrusion-resistant ad hoc routing algorithms. *21st Century Military Communications Conference Proceedings(MILCOM 2000)*, 2:660–664.

[Rescorla, 1999] Rescorla, E. (1999). *Diffie-Hellman Key Agreement Method*. RFC 2631.

[Rivest, 1992] Rivest, R. L. (1992). *The MD5 Message-Digest Algorithm*. RFC 1321.

[Sancak et al., 2004] Sancak, S., Cayirci, E., Coskun, V. e Levi, A. (2004). Sensor wars: Detecting and defending against spam attacks in wireless sensor networks. Em *IEEE International Conference on Communications*, páginas 20–24.

[Sanzgiri et al., 2002] Sanzgiri, K., Dahill, B., Levine, B. N. e Belding-Royer, E. M. (2002). A secure routing protocol for ad hoc networks. Em *International Conference on Network Protocols*.

[Shamir, 1979] Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

[Shamir, 1984] Shamir, A. (1984). Identity-based cryptosystems and signature schemes. Em *Advances in Cryptology (Crypto '84) and Lecture Notes in Computer Science*, volume 196, páginas 47–53.

[Shamir et al., 1978] Shamir, A., Rivest, R. e Adleman, L. (1978). Mental poker. TM-125 178184, MIT Laboratory for Computer Science.

[Stallings, 2004] Stallings, W. (2004). *Business Data Communications*. Prentice-Hall, 5th edição.

[Syverson et al., 1997] Syverson, P. F., Goldschlag, D. M. e Reed, M. G. (1997). Anonymous connections and onion routing. Em *IEEE Symposium on Security and Privacy*, páginas 44–54, Oakland, California.

[Theodorakopoulos e Baras, 2004] Theodorakopoulos, G. e Baras, J. S. (2004). Trust evaluation in ad-hoc networks. Em *ACM Workshop on Wireless Security (WiSE'04)*, Philadelphia, USA.

[Velloso et al., 2006] Velloso, P. B., Laufer, R. P., Duarte, O. C. M. B. e Pujolle, G. (2006). HIT: A human-inspired trust model. Em *VIII IFIP/IEEE International Conference on Mobile and Wireless Communications Networks (MWCN 2006) - a ser publicado*, Santiago, Chile.

[Wikipedia - The Free Encyclopedia, 2006] Wikipedia - The Free Encyclopedia (2006). *Eaves*. http://en.wikipedia.org/wiki/Eaves.

[Wood e Stankovic, 2002] Wood, A. e Stankovic, J. (2002). Denial of service in sensor networks. *Computer*, 35(10):54–62.

[Ye et al., 2003] Ye, Z., Krishnamurthy, S. V. e Tripathi, S. K. (2003). A framework for reliable routing in mobile ad hoc networks. Em *INFOCOM 2003*.

[Yi et al., 2001] Yi, S., Naldurg, P. e Kravets, R. (2001). Security-aware ad hoc routing for wireless networks. Em *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*, Long Beach, CA.

[Zapata, 2002] Zapata, M. G. (2002). Secure ad hoc on-demand distance vector (SAODV) routing. *ACM Mobile Computing and Communications Review*, 6(3):106–107.

[Zhong et al., 2003] Zhong, S., Chen, J. e Yang, Y. R. (2003). Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. Em *IEEE INFOCOM*, San Francisco, USA.

[Zhou e Haas, 1999] Zhou, L. e Haas, Z. J. (1999). Securing ad hoc networks. *IEEE Network*, 13(6):24–30.

Capítulo 3

Introdução à Biometria

Luciano R. Costa, Rafael R. Obelheiro e Joni S. Fraga
Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Email: {luciano, rro, fraga}@das.ufsc.br

Abstract

Biometric authentication, based on intrinsic personal traits, has long been an object of interest to the computer security community. However, until some time ago its adoption was restricted to highly secure environments and criminal identification applications. With the recent technological improvements and reduction in device costs, biometrics has become more disseminated, being frequently touted as a promising solution for authentication problemas. This chapter provides an overview of biometric authentication, examining the most used technologies and discussing its benefits and limitations. We also address architectural aspects of biometric systems, as well as open problems that require further research.

Resumo

A autenticação biométrica, baseada em características pessoais intrínsecas, há muito tem sido objeto do interesse da comunidade de segurança computacional. Entretanto, até pouco tempo atrás a sua adoção se restringia a ambientes de alta segurança e aplicações de identificação criminal, por razões de natureza econômica e tecnológica. Com o aperfeiçoamento da tecnologia e a redução no custo dos dispositivos verificados recentemente, a biometria vem se popularizando, sendo freqüentemente apontada como uma solução promissora para problemas de autenticação. Este capítulo apresenta uma visão geral da autenticação biométrica, examinando as principais tecnologias utilizadas e discutindo seus benefícios e limitações. São considerados ainda aspectos arquiteturais de sistemas biométricos, bem como problemas em aberto que precisam ser melhor pesquisados.

3.1. Introdução

O conceito de segurança em um sistema computacional está relacionado à manutenção de três propriedades fundamentais: a *confidencialidade*, que garante que a informação somente seja revelada com autorização apropriada; a *integridade*, que garante que a informação somente seja alterada com autorização apropriada; e, a *disponibilidade*, que garante que a informação seja acessível aos legítimos usuários, quando requerida. Tem se tornado comum acrescentar duas outras propriedades, a *autenticação*, que garante que cada entidade seja aquela que alega ser, e o *não-repúdio*, que garante que uma terceira parte neutra possa ser convencida de que uma transação ou evento em particular ocorreu, ou não ocorreu [Landwehr 2001]. A autenticação possui importância fundamental, pois em geral a autorização é concedida ou negada com base na identidade associada à entidade que solicita acesso ao recurso ou em algum atributo que depende dessa identidade.

Uma credencial é uma evidência fornecida por uma entidade, ao requisitar acesso a um recurso. O protocolo de autenticação decide se as credenciais apresentadas constituem prova suficiente de identidade para autorização da entidade a acessar recursos. As credenciais apresentadas podem ser de três tipos [Miller 1994]:

- **Posse** - Qualquer detentor da posse de um objeto é capaz de utilizar o recurso. Por exemplo, o possuidor da chave do carro possui o privilégio de utilizá-lo.
- **Conhecimento** - Indivíduos possuidores de certo conhecimento são elegíveis para utilizar um recurso. Neste caso, a autenticação é baseada em um conhecimento secreto,[1] compartilhado entre o usuário e a aplicação.
- **Biometria** - Os traços das pessoas podem ser medidos e computados na forma de um identificador biométrico único, difícil de compartilhar, roubar, forjar e de ser alterado.

O objetivo deste capítulo consiste em apresentar uma visão geral sobre os sistemas biométricos e seus principais aspectos de segurança. Desta maneira, a seção 3.2 apresenta os principais conceitos envolvendo sistemas biométricos: modos de autenticação usando biometria, requisitos de características biométricas, tecnologias biométricas existentes, aplicações de biometria, modelo conceitual de sistemas biométricos, erros, critérios de seleção de tecnologias biométricas e padrões em biometria.

Dentre as diversas tecnologias biométricas enumeradas na primeira seção, existem algumas — impressões digitais, íris, faces, formato das mãos e assinaturas — que se encontram em um estágio de desenvolvimento bastante satisfatório. Essas tecnologias estão amadurecidas, estando disponíveis comercialmente em implementações de boa qualidade a um custo razoável. Em vista disso, a seção 3.3 examina mais detidamente cada uma dessas tecnologias, detalhando o seu modo de funcionamento e analisando quais os seus

[1] Entretanto, podemos distinguir conhecimento com vários graus de *segredo*. Um ID de usuário de computador ou um número de conta bancária são freqüentemente solicitados para autenticação, embora tal conhecimento não seja segredo. Mesmo assim, não são universalmente conhecidos, o que ajuda a prevenir ataques de impostação superficiais. De fato, podemos distinguir uma faixa de segredo que vai do universalmente conhecido ao segredo completo.

principais benefícios e desvantagens. São apresentados ainda os recursos existentes para pesquisa, como bancos de dados e ferramentas.

Embora essencialmente todos os sistemas biométricos se encaixem em um mesmo modelo conceitual, a implementação desse modelo pode diferir de um sistema para o outro. A seção 3.4 apresenta as arquiteturas de armazenamento e segurança de sistemas biométricos. A arquitetura de armazenamento considera as formas com que os vários processos que compõem o modelo conceitual são distribuídos no sistema, e onde são armazenados os diferentes dados biométricos usados no modelo. A arquitetura de segurança discute as vantagens dos sistemas biométricos do ponto de vista de segurança e suas vulnerabilidades específicas (notadamente a facilidade de obtenção de características biométricas sem o consentimento do usuário e a irrevogabilidade dessas características), bem como contramedidas que podem ser usadas para contornar ou minimizar essas vulnerabilidades (multibiometria, biometria cancelável e autenticação multifatores).

Finalmente, a seção 3.5 discute os principais problemas abertos na área de biometria, e a seção 3.6 apresenta as conclusões do capítulo e algumas considerações finais.

3.2. Conceitos

3.2.1. Verificação e identificação

Os sistemas biométricos são usados para a **autenticação de pessoas**. Nestes sistemas, existem dois modos de autenticação: a verificação e a identificação [Bolle et al. 2004, p. 25]. Na **verificação**, a característica biométrica é apresentada pelo usuário juntamente com uma identidade alegada, usualmente por meio da digitação de um código de identificação. Esta abordagem de autenticação é dita uma busca 1:1, ou busca fechada, em um banco de dados de perfis biométricos. O princípio da verificação está fundamentado na resposta à questão: "O usuário é quem alega ser?". Na **identificação**, o usuário fornece apenas sua característica biométrica, competindo ao sistema "identificar o usuário". Esta abordagem de autenticação é dita uma busca 1:N, ou busca aberta, em um banco de dados de perfis biométricos. O sistema busca todos os registros do banco de dados e retorna uma lista de registros com características suficientemente similares à característica biométrica apresentada. A lista retornada pode ser refinada posteriormente por comparação adicional, biometria adicional ou intervenção humana. Basicamente, a identificação corresponde a responder à questão: "Quem é o usuário?".

A identificação também é utilizada em aplicações conhecidas como aplicações de varredura (*screening*), que somente podem ser executadas com alguma forma de biometria. Estas são aplicações de busca com política negativa, pois procuram estabelecer se um indivíduo está em alguma lista de pessoas de interesse, como a lista dos mais procurados, ou um banco de dados de algum tipo de benefício. O propósito de uma varredura é prevenir o uso de múltiplas identidades. Por exemplo, se A já recebe algum benefício e agora alega ser B e gostaria de receber de novo o benefício, o sistema pode estabelecer que B já está no banco de dados.

3.2.2. Tecnologias Utilizadas

Qualquer característica fisiológica ou comportamental humana pode ser usada como característica biométrica desde que ela satisfaça alguns requisitos básicos [Clarke 1994]:

- **Universalidade**: toda a população (a ser autenticada) deve possuir a característica. Na prática, temos pessoas que não possuem impressões digitais, por exemplo.
- **Unicidade**: uma característica biométrica deve ser única para cada indivíduo, ou seja, a possibilidade de pessoas distintas possuirem características idênticas, deve ser nula ou desprezível. Assim, a altura de uma pessoa não é uma boa característica para autenticação, já que várias pessoas podem possuir a mesma altura. Na prática, as características biométricas podem apresentar maior ou menor grau de unicidade, mas nenhuma delas pode ser considerada absolutamente única para cada indivíduo.[2]
- **Permanência**: a característica deve ser imutável. Na prática, existem alterações ocasionadas pelo envelhecimento, pela mudança das condições de saúde ou mesmo emocionais das pessoas e por mudanças nas condições do ambiente de coleta.
- **Coleta**: a característica tem que ser passível de mensuração por meio de um dispositivo. Na prática, todos as características biométricas utilizadas comercialmente atendem a este requisito.
- **Aceitação**: a coleta da característica deve ser tolerada pelo indivíduo em questão. Na prática, existem preocupações com higiene, com privacidade e questões culturais que diminuem a aceitação da coleta.

Na prática, porém, nenhuma característica biométrica consegue atender com perfeição aos requisitos de uma característica biométrica ideal.

Ao longo do tempo, diversas tecnologias biométricas foram desenvolvidas. As tecnologias biométricas existentes são classificadas, por conveniência, em dois grupos (figura 3.1). O primeiro grupo está baseado em características chamadas de **fisiológicas** ou **estáticas**. Essas características são traços fisiológicos, originários da carga genética do indivíduo, e essencialmente variam pouco (ou nada) ao longo do tempo. As principais características estáticas são a aparência facial, o padrão da íris, a geometria das mãos e as impressões digitais, que serão apresentadas com maior detalhamento na seção 3.3.

Outras características estáticas também são utilizadas em menor grau ou estão em estágios inicias de pesquisa, como a impressão palmar [Zhang and Shu 1999, Lu et al. 2003], o DNA [Bolle et al. 2004, p. 52], o formato das orelhas [Burge and Burger 2000, Victor et al. 2002], o padrão vascular da retina [Hill 1999], o odor do corpo [Korotkaya 2003], o padrão da arcada dentária [Chen and Jain 2005] e o padrão de calor do corpo ou de partes dele [Prokoski and Riedel 1999].

[2] A quantidade de variação devida à genética e ao ambiente muda de biometria para biometria. Cada pessoa é única, se analisada com suficiente detalhe. É próximo do impossível que duas pessoas diferentes tenham a mesma, idêntica, representação biométrica em qualquer sistema razoável. Contudo, ao lidar com tecnologias práticas de autenticação, encontramos limites na resolução das imagens extraídas, na capacidade de armazenamento e na habilidade de comparação entre dados extraídos. Na prática, isto extermina a noção de unicidade absoluta para todas as características biométricas.

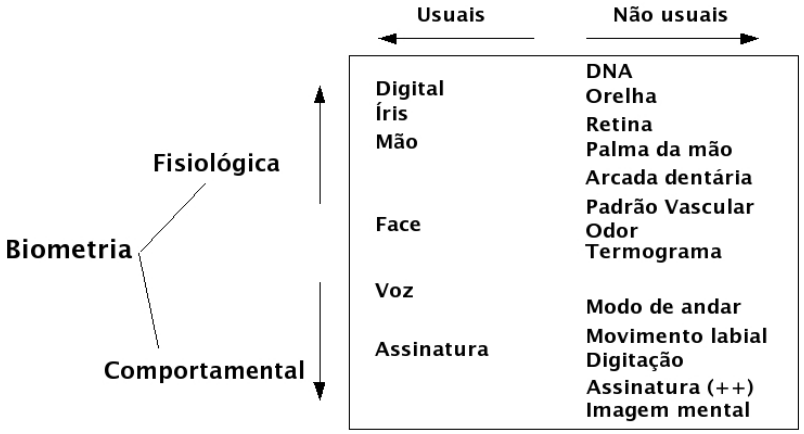


Figura 3.1. As seis características biométricas mais comuns e outras, que são usadas com menor frequência ou que estão em estágios iniciais de pesquisa. As características fisiológicas (estáticas) dependem principalmente da carga genética e as comportamentais (ou dinâmicas) dependem ainda fortemente do aprendizado e da experiência.

O segundo grupo de tecnologias biométricas está baseado em características chamadas de **comportamentais** ou **dinâmicas**. São características aprendidas ou desenvolvidas ao longo da utilização constante, e que podem variar fortemente ao longo do tempo. Além disso, podem ser facilmente alteradas pela vontade ou estado do usuário. Assim, até mesmo duas amostras consecutivas podem mudar bastante. As principais características dinâmicas utilizadas são o padrão de voz e a dinâmica da assinatura, que também serão detalhadas na seção 3.3.

Outras características dinâmicas também são utilizadas em menor grau ou estão em estágios inicias de pesquisa, como dinâmica de digitação (*keystroke dynamics*) [Bergadano et al. 2002], modo de andar [Phillips et al. 2002], movimento labial [BioID 2005] [Valid 2005], som da assinatura[3], vídeo da assinatura [Fink et al. 2001] e imagens mentais (*pass-thoughts*) [Thorpe et al. 2005].

3.2.3. Aplicações

As tecnologias biométricas podem ser utilizadas em uma ampla variedade de aplicações, para proporcionar (1) controle de acesso físico e lógico e (2) fornecimento de unicidade. Existe uma taxonomia genérica de aplicações, segundo a qual todas aplicações podem ser particionadas em sete categorias, pelo menos [Wayman 1999b]. De uma maneira prática, as aplicações dos nichos Governamental, Comercial e Forense (classificação vertical) podem ser classificadas por finalidade (classificação horizontal). Dentre os diversos conjuntos possíveis, dependendo do refinamento, um exemplo é a classificação de alto nível de sete grupos usada no relatório BITE Market Report [BITE 2005], mostrada na tabela 3.1.

[3]Informações sobre a pesquisa fornecidas pelo prof. Lee Luan Ling (FEEC/UNICAMP) e notícia publicada em `http://www.unicamp.br/unicamp/unicamp_hoje/ju/setembro2003/ju229pg8b.html`.

Finalidade	Utilização
Identificação Criminal	28 %
Controle de acesso e atendimento	22 %
Identificação Civil	21 %
Segurança de redes e de computadores	19 %
Autenticação em pontos de vendas, ATM's e varejo	4 %
Autenticação telefônica e comércio eletrônico	3 %
Vigilância e filtragem	3 %

Tabela 3.1. Distribuição horizontal (por finalidade) das principais aplicações biométricas [BITE 2005]

3.2.4. Sistema biométrico típico

Seja qual for a característica biométrica utilizada, ela deve estar enquadrada em um **sistema biométrico**. Um sistema biométrico pode ser encarado como um sistema de reconhecimento de padrões de propósito específico [Bolle et al. 2002]. O modelo conceitual simples de sistema biométrico, apresentado na figura 3.2, leva em consideração os dados e processos básicos comuns a qualquer sistema biométrico. Num sistema biométrico, o usuário é previamente registrado e seu perfil biométrico fica armazenado. Quando da utilização posterior do sistema, o processo de aquisição obtém os dados biométricos apresentados. Características particulares dos dados são extraídas para comparação com o perfil armazenado. O processo de comparação decide se os dados apresentados são suficientemente similares ao perfil registrado.

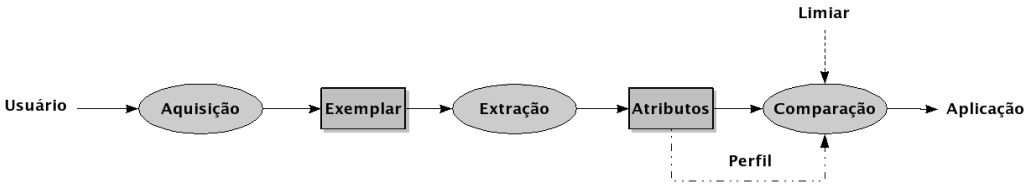


Figura 3.2. Um modelo simples de sistemas biométricos

- *Aquisição e exemplar* - O processo de aquisição ou apresentação é o processo de obtenção dos dados da característica biométrica oferecida. Normalmente a dificuldade deste processo é balancear adequadamente a qualidade da amostra sem causar excesso de inconveniência para o usuário. Neste módulo é geralmente embutido um controle da qualidade da amostra adquirida (viabilidade de processamento). O exemplar ou amostra (*sample*) é o resultado do processo de aquisição.

- *Extração e atributos* - O processo de extração produz uma representação computacional do exemplar obtido, que chamaremos de atributos, ou características extraídas (*features* ou *trial template*). A extração de características é a redução de um conjunto de medidas formado por uma grande quantidade de dados que contêm uma pequena quantidade de informação útil para um conjunto que contém menos dados mas praticamente a mesma quantidade de informação [Patrick 1972].

- *Registro e perfil* - O processo de registro, ou *enrollment*, obtém previamente os dados biométricos do usuário para cadastramento no sistema. O perfil biométrico obtido, ou *template*, é armazenado para uma comparação posterior. A linha pontilhada na figura 3.2 significa que o processo de registro, embora realizado raramente, é necessário para o estabelecimento do perfil para posterior comparação.

- *Comparação, limiar e decisão* - O processo de comparação, ou *matching*, verifica qual é o grau de similaridade entre as características extraídas da amostra do usuário e o perfil armazenado previamente. Este processo fornece um escore representativo da similaridade entre os dois conjuntos de dados. Caso a similaridade seja superior a um certo limite previamente determinado, conhecido como limiar, ou *threshold*, a decisão é aceitar o usuário, ou seja, uma autenticação válida. Caso a similaridade seja inferior ao limiar, a decisão é não aceitar o usuário, e então temos um usuário não autenticado.

3.2.5. Erros

De uma maneira geral, a comunidade biométrica diferencia vários tipos de erros, conforme a localização lógica de sua ocorrência. As diferentes aplicações biométricas podem ter diferentes definições dos erros associados. Consequentemente, há muita terminologia para expressar a precisão de uma aplicação [Bolle et al. 2004, p. 65]. O que é bastante claro e aceito por toda a comunidade biométrica é que qualquer sistema biométrico cometerá erros e que o verdadeiro valor associado às diversas taxas de erro não pode ser estabelecido teoricamente, por cálculo, mas somente por estimativas estatísticas dos erros, que são expressos em taxas e percentagens.

Há dois tipos de erros nos quais o comparador pode incorrer [Wayman 1997, Wayman 1999a].

- *False Match* (FM) - Erro do tipo I - Decidir que os exemplares são similares, enquanto na realidade eles pertencem a diferentes indivíduos. A freqüência com a qual este erro ocorre é chamada *False Match Rate* (FMR).

- *False Non-Match* (FNM) - Erro do tipo II - Decidir que dois exemplares não são do mesmo indivíduo enquanto na realidade eles pertencem ao mesmo indivíduo. A freqüência com a qual este erro ocorre é chamada *False Non-Match Rate* (FNMR).

A terminologia FM e FNM é aplicada geralmente a algoritmos de comparação ou módulos comparadores. Na prática, para os sistemas biométricos considerados como um todo, é utilizada a terminologia convencional de reconhecimento de padrões FA (*False Accept*) e FR (*False Reject*).

- *False Accept* (FA) - Erro do tipo I - Decidir que uma identidade alegada é legítima quando na realidade ela é falsa. A freqüência de ocorrências de erros deste tipo é chamada *False Accept Rate* (FAR).

- *False Reject* (FR) - Erro do tipo II - Decidir que uma identidade alegada é falsa quando na realidade ela é legítima. A freqüência de ocorrências de erros deste tipo é chamada *False Reject Rate* (FRR).

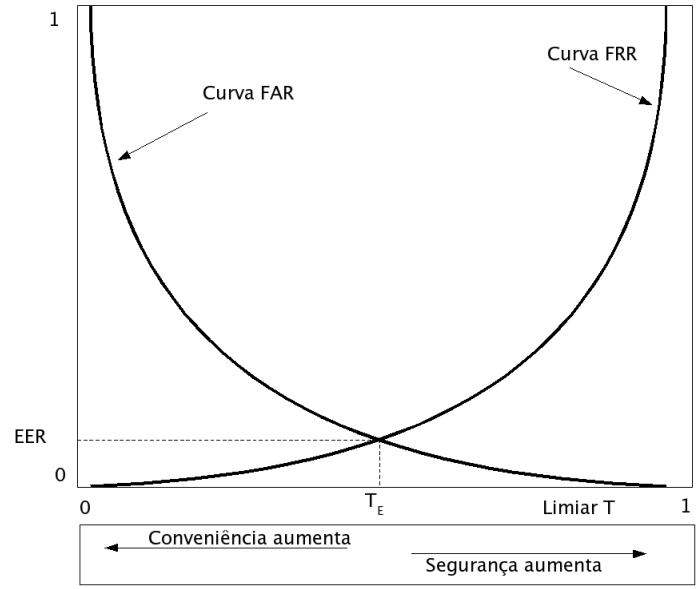


Figura 3.3. As curvas típicas das taxas de erro FAR e FRR, plotadas uma ao lado da outra, em relação ao limiar T configurado para o sistema. As curvas se cruzam num ponto notável de operação EER$(T_E) \rightarrow$ (FAR$(T_E) =$ FRR(T_E)). O sistema pode operar nas faixas de "conveniência" ou de "segurança", conforme a calibração do limiar.

Devido à possibilidade de calibrar o sistema por meio do ajuste do limiar, as taxas de erros possuem conseqüências opostas. FA resulta em brechas na segurança, com a admissão de usuários não autorizados. Por outro lado, FR resulta em problemas de conveniência, já que usuários genuínos terão acesso negado até uma verificação posterior. As taxas de erro FAR e FRR podem ser plotadas *uma ao lado da outra*, como apresentado na figura 3.3. Para avaliar de forma sumária a qualidade das curvas FAR e FRR e, por conseqüência, a precisão de operação de um dado sistema, é possível a explicitação de um ponto notável, onde as taxas são iguais, ou seja, o limiar $T = T_E$ para o qual FAR$(T) =$ FRR(T). Este ponto é conhecido como ponto de operação EE (*Equal Error*), ao qual também está associado uma taxa EER (*Equal Error Rate*).

As taxas FAR(T) e FRR(T) também podem ser comparadas *uma contra a outra* para produzir uma curva bi-dimensional característica conhecida por *Receiver Operating Characteristic* (ROC). Um exemplo hipotético pode ser apreciado na figura 3.4. Embora a curva ROC represente uma boa descrição da precisão de um sistema, sua real utilidade vem à tona quando queremos confrontar dois sistemas. É claro que não é uma tarefa trivial, pois as curvas podem não ser tão bem comportadas como a curva da figura 3.4. De fato, as curvas podem se cruzar, e podem indicar diferentes desempenhos em diferentes regiões. Assim, deve ser levado em consideração em que região de T (limiar) desejamos efetuar o confronto.

Existem outros conceitos úteis para avaliação mais delicada de comparadores, como a separação das densidades de probabilidade [Daugman and Williams 1996] e o conceito de Erro Total Esperado, com seu refinamento associado a Funções de Custo para cada tipo de erro [NIST 2003] [Bolle et al. 2004, seção 16.3]. Estas Funções de Custo levam em consideração a vocação do sistema. Por exemplo, em dado sistema, onde seja

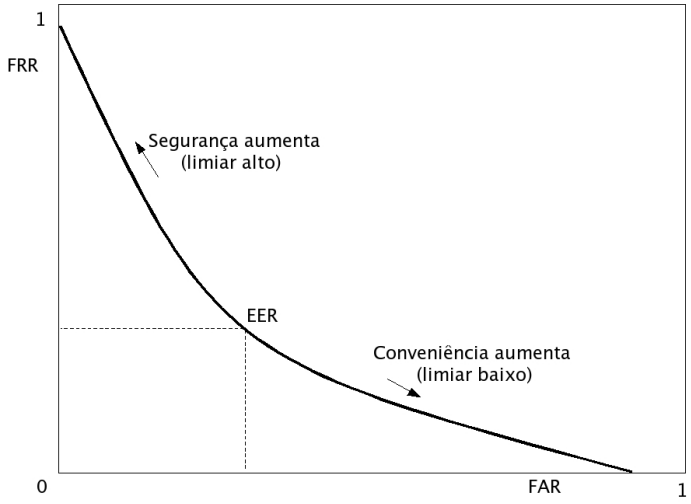


Figura 3.4. *Receiver Operating Characteristic*. As taxas de erro FAR e FRR podem ser plotadas uma contra outra numa curva bi-dimensional. Aqui tentamos mostrar a solução de compromisso entre segurança e conveniência.

necessária alta segurança, os problemas advindos de FRs são aborrecimentos rotineiros, enquanto os problemas advindos de FAs são desastrosos. Por outro lado, podem existir sistemas com maior necessidade de conveniência. Por exemplo, máquinas de autoatendimento de um banco, no qual FRs não são aceitáveis por falta de pessoal de suporte, mas FAs podem ser tolerados, já que existiria uma segunda fase de autenticação por senha.

3.2.6. Seleção

Selecionar uma tecnologia biométrica adequada para uma dada aplicação específica é um processo que envolve muitos fatores. A precisão é um fator importante, mas de maneira alguma é o fator mais importante. De uma maneira simplista, fatores de seleção são extraídos dos requisitos da aplicação. Estes fatores de seleção orientam a escolha da tecnologia biométrica mais adequada. Estes fatores, embora não sejam diretamente quantificáveis, são extremamente úteis no processo de seleção. Este processo é ilustrado na figura 3.5.

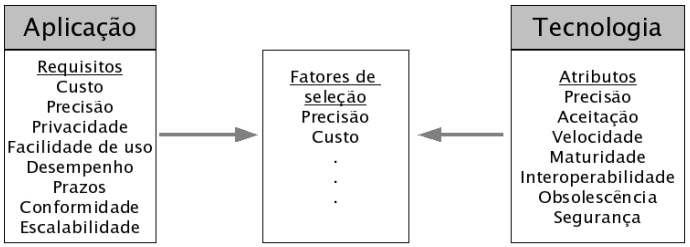


Figura 3.5. Fatores de seleção são extraídos dos requisitos da aplicação para orientar a escolha de tecnologias biométricas com atributos mais adequados.

Tendo em mente os fatores de seleção, uma primeira análise pode ser efetuada com base nos pontos fortes e pontos fracos de cada tecnologia biométrica. Como o processo de seleção pode se tornar complexo, ferramentas para orientação da escolha podem ser utilizadas. Uma **ferramenta preliminar** de análise pode ser utilizada pela construção de uma matriz de comparação baseada em pesos de atributos. A idéia básica é construir

uma matriz de avaliação. De um lado, as tecnologias biométricas disponíveis possuem atributos, aos quais podem ser vinculados valores numéricos. De outro lado, a aplicação possui requisitos. Também podem ser atribuidos valores numéricos para a importância de tais requisitos. O "casamento" entre requisitos e atributos resulta em valores de avaliação para cada tecnologia. A interpretação dos pesos simbólicos como fatores numéricos pode ser ajustada arbitrariamente. Esta matriz de avaliação é especialmente útil em estágios preliminares de análise, para apontar as sensibilidades críticas do suposto sistema [Bolle et al. 2004, p. 138].

Entretanto, um sistema biométrico pode ser suficientemente grande para incorrer em grandes investimentos. Nestes casos, uma **avaliação** mais consistente se mostra necessária. Biometria é uma tecnologia emergente com forte competição de mercado e é desejável a existência de métricas precisas e procedimentos de teste bem definidos. A tecnologia biométrica automatizada ainda é suficientemente emergente para produzir definições duvidosas de precisão e desempenho [Phillips et al. 2000]. Normalmente, as avaliações são implementadas por meio de uma competição entre os interessados (fabricantes ou grupos de pesquisa). Existem três metodologias proeminentes de avaliação: (1) avaliação de tecnologia, (2) avaliação de cenário e (3) avaliação operacional.

O objetivo da **avaliação de tecnologia** [Mansfield and Wayman 2002] é a comparação dos algoritmos competidores de uma tecnologia única. Os testes são realizados sobre um banco de dados padronizado de perfis biométricos. Os resultados dos testes são repetíveis. Neste tipo de avaliação, é concedido aos competidores um certo período de tempo para treinar seus algoritmos de verificação. Um banco de dados de perfis biométricos é disponibilizado pelos organizadores, ou seja, são usados bancos de dados de perfis biométricos previamente construídos. Os módulos de comparação competidores recebem estes dados e têm direito a um certo tempo para o treinamento de seus algoritmos. Esta é a fase de treinamento. Na outra fase, a fase de teste, são definidas as maneiras de obtenção das estatísticas de desempenho. Então, é disponibilizada aos competidores, uma partição do banco de dados de perfis biométricos. A avaliação, portanto, consiste em duas fases, uma fase de treinamento e uma fase de competição. A avaliação de tecnologia permite obter estimativas das taxas de erro dos comparadores (FMR e FNMR). O ponto fraco desta avaliação é que apenas módulos de comparação são avaliados contra bancos de dados, sem controle do ambiente de registro.

O objetivo da **avaliação de cenário** [Mansfield and Wayman 2002] é determinar o desempenho geral do sistema numa aplicação prototipada ou simulada. Os testes englobam o sistema completo num ambiente que modela a aplicação real. É fornecida uma mesma coleção de dados biométricos para os sistemas participantes da avaliação. Os resultados dos testes são repetíveis. Este tipo de avaliação ocorre em uma instalação especial, um ambiente de teste que simula um ambiente de produção. Neste ambiente, são instalados os dispositivos biométricos de verificação (1:1) usados nos testes. Um grupo de voluntários utiliza os sistemas durante um certo período de tempo (idealmente meses ou até mesmo anos), enquanto as estatísticas são coletadas. Podem ser comparados diferentes fabricantes ou até mesmo diferentes tecnologias ao mesmo tempo. Além disso, tal avaliação cria como subproduto um banco de dados de perfis biométricos que pode ser utilizado posteriormente para avaliações operacionais. São obtidas estimativas de FAR e FRR. O ponto fraco desta avaliação fim-a-fim é que os dispositivos não são realmente

atacados, o que leva a valores irreais de FAR.

O objetivo da **avaliação operacional** [Bolle et al. 2004, p. 111] é determinar o desempenho do sistema biométrico como um todo, inserido num ambiente específico de aplicação, atuando sobre uma população-alvo específica. Os resultados geralmente não são repetíveis, já que dependem de características — às vezes desconhecidas ou não documentadas — do ambiente de aplicação. Este tipo de avaliação é realizado, tanto quanto possível, sob circunstâncias reais, ou seja, no ambiente empresarial. Embora seja a avaliação mais realista, não pode medir a verdadeira FAR, já que os eventos de falsa aceitação serão de conhecimento exclusivo dos fraudadores. No entanto, ainda há a possibilidade de estimativa da verdadeira FAR por intermédio de complemento a esta avaliação, por meio da utilização de algo parecido com a contratação de testes de invasão, a exemplo do que é feito com segurança de redes de computadores. Este ainda é um campo aberto para pesquisas.

Para aliviar a dificuldade da tarefa de seleção de sistemas biométricos, existem alguns importantes **documentos de apoio** publicados por instituições dedicadas a sistemas biométricos. Por exemplo, o BWG (*Biometrics Working Group*) publicou um documento contendo um conjunto de conselhos práticos, úteis para gestores envolvidos em projetos de utilização de sistemas biométricos. O documento procura suplementar, e não substituir, metodologias e práticas de gerenciamento de projetos [Mansfield et al. 2002]. Um teste de avaliação pode ser caracterizado por cinco passos: planejamento, aquisição dos dados, análise, estimativa das incertezas e relatório final de desempenho. Regras básicas práticas para levar este trabalho a bom termo estão disponíveis no relatório publicado também pelo BWG [Mansfield and Wayman 2002] e nas especificações publicadas pelo instituto *American National Standards Institute* [ANSI 2005].

3.2.7. Padronização

A padronização é necessária para a ampla aceitação de tecnologias biométricas. Atualmente, os dispositivos não possuem **interoperabilidade**. Padrões internacionais relativos a tecnologias biométricas têm sido propostos e estão em fase de amadurecimento. Estes padrões pretendem dar suporte à troca de dados entre aplicações e sistemas e tentam evitar os problemas e custo oriundos dos sistemas proprietários. Alguns dentre os mais importantes são mostrados na figura 3.6 e descritos resumidamente nos parágrafos a seguir.

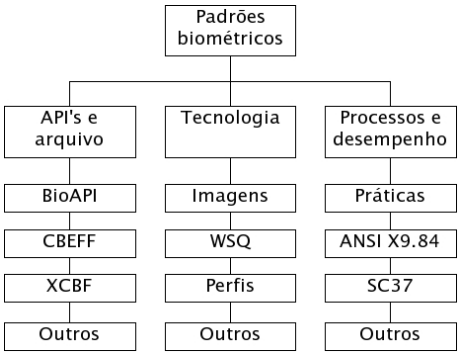


Figura 3.6. Principais esforços de padronização relacionados a sistemas biométricos

BioAPI O consórcio BioAPI[4] foi fundado para desenvolver uma API (*Application Programming Interface*) para proporcionar independência de dispositivo e de plataforma. O consórcio é formado por cerca de 120 companhias (pelo menos uma delas brasileira) interessadas em promover o crescimento do mercado biométrico. A BioAPI é a API mais popular na área biométrica. Suas primitivas se referem a tarefas de registro, identificação e verificação numa plataforma cliente/servidor e aquisição do sinal numa plataforma cliente. No nível mais alto, é definido um BSP (*Biometrics Service Provider*), que lida com todos os aspectos do processamento do sinal. Os diversos componentes se registram durante a instalação. O módulo de registro pode ser usado pelas aplicações para verificar os BSPs instalados e suas funcionalidades. Baseado na BioAPI, foi também definida uma API específica para Java Cards,[5] para dar suporte a funcionalidades biométricas em *smart cards*, principalmente quanto à segurança dos algoritmos e do perfil biométrico eventualmente armazenado no cartão.

CBEFF CBEFF (*Common Biometric Exchange File Format*) é um padrão que procura lidar com os dados biométricos, em sua forma inicial de amostra adquirida ou na forma de características extraidas [NIST 2001]. O padrão procura facilitar a troca de dados entre diferentes processos do mesmo sistema ou até mesmo entre sistemas diferentes. Os dados descritos incluem segurança (assinaturas digitais e cifragem dos dados), processamento da informação (identificação dos tipos biométricos e informação sobre a amostra) e os dados biométricos em si.

ANSI X9.84 Este padrão [ANSI 2003], desenvolvido para utilização na indústria financeira, é compatível com o padrão CBEFF. Ele define requisitos para gerenciamento e proteção da informação biométrica nas fases de coleta, distribuição e processamento dos dados. O padrão inclui especificações para a segurança do equipamento usado, o gerenciamento dos dados, a utilização da tecnologia biométrica para verificação/identificação de clientes e empregados, a aplicação da tecnologia para controle de acesso físico e lógico e técnicas para transmissão e armazenamento seguros dos dados biométricos.

XCBF Desenvolvido sob orientação de um comitê do OASIS, o *XML Common Biometric Format* (XCBF) [OASIS 2003] fornece a codificação XML para o formato padrão CBEFF. A intenção é incrementar a interoperabilidade entre aplicações biométricas baseadas em XML, como aplicações baseadas na Internet. Este padrão também procura ser compatível com as especificações ANSI X9.84.

ISO/JTC1/SC37 SC37 é um subcomitê da ISO (*International Organization for Standardization*) criado na década de 80 para padronização de aspectos ligados a sistemas biométricos. Os grupos de trabalho vinculados atuam em áreas como terminologia, interfaces, formatos de troca de dados, arquitetura funcional, teste e avaliação.

[4] `http://www.bioapi.org/`.

[5] A API completa é descrita no documento disponível em `http://www.javacardforum.org/Documents/JCFBIOApiVIA.pdf`.

WSQ Para arquivar o enorme banco de dados de impressões digitais do FBI, foi proposto um algoritmo de compressão eficiente, que mantém a fidelidade dos detalhes das linhas. As imagens de impressão digital, de resolução de 500 dpi (8 bits de escala de cinza) são comprimidos com o uso do algoritmo WSQ (*Wavelet Scalar Quantization*),[6] proporcionando taxas de compressão de cerca de 15:1.

3.3. Tecnologias

3.3.1. Impressão Digital

A formação das impressões digitais se inicia no sétimo mês de gestação, com a diferenciação da pele das pontas dos dedos. O fluxo de fluidos amnióticos em volta do feto e a posição do feto dentro do útero, mudam durante o processo de diferenciação. Então, as células das pontas dos dedos crescem em um micro-ambiente, que é ligeiramente diferente de mão para mão e de dedo para dedo. Os detalhes finos das impressões digitais são determinados por este micro-ambiente em constante mudança.

Em estudos dermatológicos, a máxima diferença entre impressões digitais tem sido encontrada entre indivíduos de diferentes raças. Pessoas da mesma raça, porém sem grau de parentesco, possuem similaridade muito pequena nas digitais. Pai e filho possuem alguma similaridade, por compartilharem metade dos genes. Gêmeos monozigóticos (idênticos) possuem a máxima similaridade. Estima-se que 95% das características das digitais de gêmeos idênticos sejam iguais [Maltoni et al. 2003].

O processo de **aquisição** da impressão digital obtém a imagem em preto e branco das linhas dos dedos. A impressão digital pode ser estampada em papel, pressionando o dedo previamente preparado com tinta. Esta imagem pode ser posteriormente digitalizada por meio de um *scanner*. Um tipo especial de imagens é o das impressões digitais latentes encontradas em cenas de crimes, que podem ser recuperadas por meio de um procedimento especial. Uma imagem ao vivo, por outro lado, é obtida por meio de dispositivos eletrônicos especiais. O princípio básico de todos é a detecção das rugosidades dos dedos que estão em contato com o dispositivo. A aquisição de imagens ao vivo está baseada em quatro tecnologias: ótica, capacitiva, térmica e ultrasônica.

Na tecnologia **ótica**, FTIR (*Frustrated Total Internal Reflection*) e outros métodos óticos são a maneira mais antiga de obtenção de imagens ao vivo. A superfície de aquisição de 1” $\times$ 1” é convertida em imagens de cerca de 500 dpi. A luz refletida depende das condições da pele e imagens saturadas ou difusas podem ser obtidas de peles molhadas e secas, respectivamente.

Na tecnologia **capacitiva**, as cristas e vales da pele da ponta de um dedo, criam diferentes acumulações de carga quando o dedo toca uma rede de chips CMOS. Com a eletrônica adequada, a carga é convertida num valor de intensidade de um pixel. A superfície de aquisição de $0,5$” $\times$ $0,5$” é convertida em uma imagem de cerca de 500 dpi. Tais dispositivos são sensíveis e a qualidade das imagens também é suscetível à pele molhada e seca.

A tecnologia **térmica** se baseia no fato de que a pele é um condutor de calor

[6] As especificações do codificador/decodificador WSQ podem ser encontradas em `http://www.itl.nist.gov/iad/894.03/fing/cert_gui.html`.

melhor que o ar. O contato com as cristas da pele causa uma alteração observável na temperatura da superfície do sensor. A tecnologia supera os problemas de pele seca e molhada e é bastante robusta. A imagem de 500 dpi obtida, no entanto, não é rica em tons de cinza.

Na tecnologia **ultrasônica**, um feixe ultrasônico é dirigido através da superfície do dedo, para medir diretamente a profundidade dos sulcos com base no sinal refletido. As condições de oleosidade da pele não afetam a imagem obtida, que reflete bastante bem a topologia dos sulcos. Contudo, estas unidades tendem a ser grandes e tendem a requerer um tempo de leitura bem maior que os leitores óticos.

A imagem resultante do processo de aquisição pode ser processada na ponta cliente da aplicação ou transmitida ao servidor para processamento. Esta transmissão e armazenamento da imagem envolve compressão e descompressão da mesma, geralmente usando WSQ (seção 3.2.7).

O processo de **extração** de características é o ponto central dos sistemas de autenticação baseados em impressões digitais, com implicações para o projeto do restante do sistema. As abordagens existentes são classificadas em três níveis: global, local e fina.

A abordagem **global** descreve a formação geral das linhas. Geralmente, podem ser observados um núcleo e mais de dois deltas. Estas formações singulares são usadas como pontos de controle, em volta dos quais as linhas são organizadas. A orientação geral das linhas é útil para classificação e indexação em grandes grupos, embora não seja suficiente para comparação precisa.

A abordagem **local** está relacionada com detalhes marcantes das próprias linhas, conhecidos como **minúcias** (*minutiae*). Embora exista mais de uma centena de tipos de detalhes catalogados, os mais utilizados em sistemas automatizados são a terminação de linha e a bifurcação de linha, conforme mostrado na figura 3.7. A extração destas características locais depende fortemente da qualidade da amostra adquirida. Os perfis biométricos obtidos por meio da extração de características de minúcias possuem um tamanho de 250 a 700 bytes.

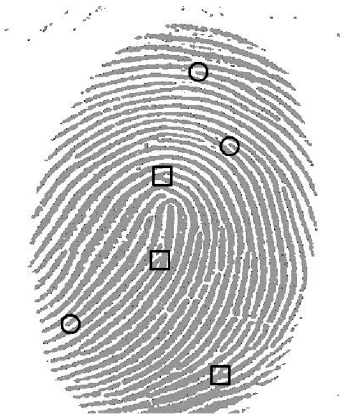


Figura 3.7. Exemplo de dois tipos de minúcias em impressões digitais: bifurcações e terminações de linha.

A abordagem **fina** está baseada nos detalhes intra-linhas, que nada mais são que a

posição e formação geral dos poros de suor, que medem cerca de 60 microns. Embora tais características sejam altamente distintivas, a sua extração somente é viável em imagens de alta resolução (cerca de 1.000 dpi) obtidas de impressões digitais de boa qualidade. A maioria dos sensores fornece imagens de resolução em trono de 500 dpi, assim este tipo de representação não é prático para a maioria das aplicações.

O processo de **comparação** é amplamente baseado nos métodos desenvolvidos por especialistas humanos. Os especialistas avaliam três fatores para declarar que duas impressões digitais pertencem ao mesmo dedo: (1) concordância na configuração global do padrão, ou seja, na distribuição do núcleo e dos deltas, o que implica em que as impressões são do mesmo tipo; (2) concordância qualitativa, ou seja, os detalhes de minúcias devem ser idênticos; e, (3) suficiência quantitativa, que especifica que ao menos um certo número de detalhes de minúcias deve ser encontrado — um mínimo de 12, segundo as orientações legais nos Estados Unidos, também aceitas no Brasil [Kazienko 2003]. A comparação por meios automatizados não segue, necessariamente, os mesmos detalhes de tais orientações, embora esteja baseada nelas de uma maneira estrutural.

Idealmente, a similaridade entre duas impressões digitais obtidas do mesmo dedo deve ser invariante quanto a (1) translação, (2) rotação, (3) pressão aplicada e (4) distorção elástica da pele. As abordagens de comparação foram estudadas por décadas, e duas classes de técnicas podem ser distinguidas:

1. Técnicas baseadas em **imagens** - Esta classe inclui técnicas de correlação de imagem tanto óticas quanto numéricas. As imagens das impressões digitais são superpostas, e a correlação no nível de intensidade entre os pixels correspondentes é computada para diferentes localizações e rotações.

2. Técnicas baseadas em **características** - A comparação baseada em minúcias é o método mais conhecido e mais largamente usado para comparação, graças à analogia com a maneira pela qual os especialistas comparam impressões digitais em aplicações forenses e graças à aceitação legal como prova de identidade na maioria dos paises. Os algoritmos de comparação mais comuns consideram cada minúcia como uma tripla $m = (x, y, \theta)$, contendo a informação de localização espacial 2D (x, y) e de orientação θ. Os detalhes extraídos são então armazenados como conjuntos de pontos, e a comparação consiste em encontrar o alinhamento para o qual os conjuntos de pontos da amostra e do perfil forneçam o máximo número de pares suficientemente coincidentes.

Os pontos fortes da tecnologia de autenticação biométrica baseada em impressão digital são:

- $\oplus$ Esta tecnologia pode proporcionar bastante precisão;[7]

[7]Na prática, a precisão obtida pelos algoritmos não deve ser avaliada pela apreciação da EER. Por exemplo, para a impressão digital, a EER obtida nas competições internacionais pode se mostrar frustrante. O resultado obtido pelo melhor algoritmo na última competição internacional (FVC2004), se aproxima de uma EER de 2,1% [Cappelli et al. 2006]. No entanto, a tecnologia de impressão digital pode trabalhar em outras faixa de operação que proporcionam excelentes resultados de precisão com um pequeno sacrifício da taxa da falsa rejeição.

- ⊕ Existe uma longa tradição legal no uso da impressão digital como identificador imutável;
- ⊕ Existem grandes bancos de dados legados de impressões digitais;
- ⊕ A impressão digital pode ser colhida facilmente a baixo custo.

Quanto aos pontos fracos, podemos citar:

- ⊖ Em algumas culturas, impressões digitais não são bem aceitas por estarem ligadas a criminosos, pessoas iletradas ou por questões de higiene;
- ⊖ A qualidade das impressões digitais varia enormemente dentro de uma população;
- ⊖ Os sensores mais baratos podem ser comprovadamente fraudados.

A tecnologia baseada em impressão digital possui vários recursos associados, como bancos de dados e aplicativos. Por exemplo, o NIST disponibiliza um banco de dados com 2.000 imagens de impressões digitais, para auxiliar pesquisas de classificação, para desenvolvimento de algoritmos e para teste e treinamento de sistemas [NIST 2005]. A Universidade de Bolonha (Itália) disponibiliza as imagens obtidas nas competições por ela organizadas em 2000, 2002 e 2004. Além disso, a mesma universidade disponibiliza um gerador automatizado de impressões digitais [BIOLAB 2005], que pode ser usado para criar imagens para uso em teste e otimização de algoritmos de reconhecimento, bem como para a execução de massa de testes para avaliações desta tecnologia.

O NIST também disponibiliza um pacote utilitário com funções de segmentação, extração e comparação de imagens de impressões digitais.[8] O algoritmo de segmentação pode ser usado para remover espaços em branco das imagens. Outro algoritmo classifica a forma geral da imagem em seis grupos diferentes. O detetor de minúcias pode localizar as terminações e bifurcações de linhas. O algoritmo de comparação pode ser executado nos modos de verificação ou identificação. Além disso, também está disponível uma grande coleção de utilitários para imagens, como codificadores e decodificadores JPEG e WSQ.

Outro exemplo bastante útil é o *FingerCode*,[9] um *software* aberto para comparação de impressões digitais implementado em MATLAB.

3.3.2. Aparência da Face

A aparência da face é uma característica biométrica particularmente convincente, pois é usada rotineiramente como primeiro método de reconhecimento entre pessoas. Por sua naturalidade, é a mais aceitável das biometrias. Devido a esta natureza amigável para o usuário, o reconhecimento de face surge como uma ferramenta poderosa, a despeito da existência de métodos mais confiáveis de identificação de pessoas, como impressão digital e íris.

O processo de **aquisição** de imagens da face possui abordagens que podem ser divididas em quatro grupos: imagem 2D, imagem 3D, seqüência de imagens e termograma.

[8] `http://fingerprint.nist.gov/NFIS/`.
[9] `http://utenti.lycos.it/matlab/speed.htm`.

1. **Imagem 2D** - A obtenção de imagens digitalizadas de fotos de documentos é importante, pois muitos dados legados estão na forma de fotografias, seja em cores, seja em preto-e-branco. Esta é a obtenção estática de imagens. Já para a obtenção de imagens ao vivo, câmeras digitais e analógicas podem ser usadas. As imagens são geralmente captadas com a cooperação do fotografado, e em condições de iluminação controladas. Qualquer câmera de baixo custo, como uma *webcam*, é utilizável para obtenção de imagens 2D. Entretanto, os melhores resultados são obtidos com câmeras que possuem foco automático e lentes apropriadas. Tanto quanto possível, câmeras com características similares devem ser utilizadas nas fases de registro e utilização. O tamanho de um arquivo contendo a imagem da face pode variar de 1 KB a 100 KB, dependendo da compressão utilizada.

2. **Imagem 3D** - Muitas técnicas modernas de reconhecimento de face estão baseadas na geometria da cabeça e exigem imagens tridimensionais. Os modelos 3D contêm mais informações da face e são invariantes à pose. Uma desvantagem ainda presente é que os modelos tratam a face como um objeto rígido, não sendo capazes de tratar expressões faciais. Embora o reconhecimento de face 2D ainda supere os métodos 3D, este cenário pode mudar num futuro próximo [Scheenstra et al. 2005]. A combinação multimodal de abordagens 2D e 3D pode incrementar a precisão total do sistema [Chang et al. 2003]. Um experiência relata uma taxa de EER de 1,9% para uma abordagem multimodal 2D+3D, contra uma taxa EER de 4,5% para as abordagens 2D e 3D separadas [Kyong I. Chang and Flynn 2005]. Para a obtenção de imagens 3D da face, podemos utilizar (1) técnicas baseadas em imagens simultâneas, onde duas câmeras 2D, cujos campos de visão são separados por um ângulo entre 8° e 15°, obtêm imagens independentes para montagem posterior; (2) técnicas baseadas em projeção de um padrão de luz conhecido, cuja distorção pode ser capturada para reconstruir a aparência 3D da face; e (3) técnicas baseadas em varredura a laser, que proporciona um mapa tridimensional pela amostragem de cada ponto da superfície da face.

3. **Seqüência de imagens** - Câmeras de vigilância gravam seqüências de vídeo, com a freqüente inclusão de imagens de faces. No entanto, devido à baixa amostragem (1 a 4 quadros por segundo), a resolução das imagens da face é de baixa qualidade, tornando difícil sua utilização em sistemas automatizados de reconhecimento. Técnicas de seguimento, em conjunção com a utilização de câmeras com *zoom* podem ser usadas para melhoria da resolução, por meio do aumento focado em faces suspeitas. É claro que o custo aumenta bastante, bem como a perda do campo de visão.

4. **Termograma da face** - Um dos problemas na aquisição de imagens da face está relacionado às condições de iluminação. Iluminação infra-vermelha de baixa potência, invisível ao olho humano, pode ser usada para suplementar o processo de detecção da face. Termogramas faciais baseados em radiação infra-vermelha oferecem atrativos, como a independência da iluminação ambiente e a habilidade de resistência a disfarces, mas o alto custo da implementação e a influência de fontes de calor pode afetar esta modalidade de biometria [Prokoski and Riedel 1999].

A figura 3.8 mostra alguns exemplos de imagens de face.

Figura 3.8. Imagem da face 2D (esquerda), 3D (centro) e infravermelho (direita).

O processo de **extração** de características da face possui como primeiro passo a detecção, ou seja, descobrir que existem uma ou mais faces em uma determinada imagem. A detecção, também conhecida como segmentação, é um processo crítico para o sucesso do reconhecimento facial. Métodos baseados em distâncias matemáticas e redes neurais alcançam cerca de 85% de taxa de detecção correta [Zhao et al. 2003]. Existem duas abordagens para a extração de características das imagens da face.

1. **Abordagem global - Aparência da Face** - A idéia básica é reduzir uma imagem de milhares de pixels para um conjunto de números. A distintividade da face pode ser capturada, independentemente do "ruído" produzido pelas variações de luminosidade, textura da pele, reflexos e outros fatores. Para isto, a imagem da face é transformada, dentro de um espaço composto por funções básicas de imagens. Falando simplesmente, as funções básicas de imagens, conhecidas como *eigenfaces*,[10] são usadas ponderadamente para compor a imagem da face em questão [Turk and Pentland 1991]. Pesquisas posteriores introduziram outras transformações similares para a representação e compressão de imagens da face. A transformação fundamental, conhecida como Transformada de Karhunen-Loéve, é agora conhecida pela comunidade biométrica como PCA (*Principal Component Analysis*).

2. **Abordagem local - Geometria da Face** - A idéia é modelar a face em termos da localização geométrica relativa de características particulares tais como olhos, boca, nariz, bochechas, etc. Assim, o reconhecimento de face se resume a comparar os sistemas geométricos obtidos.

Assim como o sistema de percepção humana usa tanto características globais como locais, um sistema de reconhecimento automatizado poderia usar ambos. Pode-se dizer que os métodos híbridos oferecem o melhor dos dois métodos.

O processo de **comparação** está baseado em três tipos de métodos: holísticos, estruturais e híbridos.

[10] *Eigenfaces* são ingredientes padronizados de face, derivados da análise estatística de muitas imagens de face. Qualquer face humana pode ser considerada como uma combinação destas faces padronizadas. A face de uma pessoa em particular poderia ser composta de 8% da face 1, 5% da face 2, e assim por diante. Isto significa que é necessário muito menos espaço para registrar uma face do que a imagem real da mesma necessita.

1. *Métodos holísticos*, que usam toda a região da face. Dentre as várias técnicas existentes, a PCA, baseada em *eigenfaces*, é a mais utilizada.

2. *Métodos estruturais*, contendo técnicas mais recentes que se utilizam de medidas geométricas (ângulos e distâncias) relativas entre diversos pontos notáveis da face, como olhos, nariz, boca e bochechas.

3. *Métodos híbridos*, que tentam oferecer o melhor dos dois métodos, na tentativa de se aproximar do sistema de percepção humano, que se utiliza tanto da aparência global da face quanto das características locais.

Estes métodos possuem em comum a dificuldade de comparação quando a aparência das características muda de forma significativa, como por exemplo, olhos fechados, olhos com óculos ou boca aberta. Em condições de laboratório, os algoritmos de reconhecimento de face podem apresentar taxas de erros bastante aceitáveis. Na prática, o desempenho dos sistemas de reconhecimento de face é muito dependente da aplicação, e bons resultados relatados em especificações de vendas ou campanhas de avaliação não significam necessariamente um bom desempenho em campo, no cenário real de uma aplicação prática [Zhao et al. 2003]. A solução encontrada tem sido restringir os problemas de captura de imagens pelo fornecimento de condições controladas. Mesmo assim, as taxas de erro ainda precisam ser bastante melhoradas.

Os pontos fortes da tecnologia de autenticação biométrica baseada na aparência da face são:

⊕ Existe larga aceitação pública para este identificador biométrico, já que fotos de faces são usadas rotineiramente em documentos.

⊕ Os sistemas de reconhecimento de face são os menos intrusivos, não exigindo qualquer contato e nem mesmo a colaboração do usuário.

⊕ Os dispositivos de aquisição de imagens 2D são de baixo custo.

Quanto aos pontos fracos, podemos citar:

⊖ Em sistemas automatizados de autenticação por meio da face, as condições de iluminação precisam ser controladas. Outros desafios técnicos ainda precisam ser vencidos.

⊖ É uma tecnologia biométrica suficientemente boa para aplicações de verificação de pequena escala. No entanto, é uma biometria pobre para aplicações de identificação de larga escala.

⊖ Uma maneira óbvia e fácil de fraudar o sistema, em aplicações de *screening*, é a utilização de disfarces.

A tecnologia baseada na aparência da face possui vários recursos associados, como bancos de dados e aplicativos. Muitos bancos de dados de imagens de face 2D estão publicamente disponíveis. Os três mais importantes são os mesmos utilizados nas competições internacionais:

- BANCA - O projeto BANCA (*Biometric Access control for Networked and e-Commerce Applications*) oferece para a comunidade de pesquisas, a oportunidade de testar seus algoritmos em um banco de dados grande e realista. Os dados de face e voz foram capturados de 208 indivíduos (metade de cada sexo), por meio de dispositivos de qualidade alta e baixa, em três diferentes cenários (controlados, degradados e adversos) [Bailly-Bailliére et al. 2003].

- FERET - O banco de dados do programa FERET (*FAcial REcognition Technology*),[11] do NIST, possui imagens neutras e naturais da face de 1.200 usuários.

- XM2VTS - Este banco de dados foi coletado durante o projeto M2VTS (*Multi Modal Verification for Teleservices and Security applications*),[12] e consiste de imagens frontais coloridas de 295 usuários em diversas posições de rosto, com fundo uniforme.

Ao contrário das imagens 2D, somente poucos bancos de dados estão disponíveis para reconhecimento facial 3D. O Max Planck Institute for Biological Cybernetics criou um banco de dados adquirido com um *laser scanner* contendo 200 indivíduos. O banco de dados XM2VTS também disponibiliza modelos 3D adquiridos de cerca de 300 indivíduos.

Competições internacionais envolvendo reconhecimento de face também são costumeiras. Existem competições documentadas desde 1995, com base nos três bancos de dados citados (BANCA, FERET e XM2VTS). A competição FVC2004 (*Face Verification Contest* 2004) foi baseada no banco de dados BANCA. A competição FRVT2006 (*Face Recognition Vendor Test* 2006)[13] foi baseada no banco de dados FERET. A competição *ICBA 2006 Face Verification*[14] teve como base o XM2VTS.

Existem vários sistemas abertos de reconhecimento de face. Por exemplo, o OSCVL (*Intel Open Source Computer Vision Library*),[15] contém algoritmos de detecção e reconhecimento de faces. A iniciação em experimentos de avaliação de sistemas de reconhecimento de face também não é difícil. Um sistema completo de avaliação é fornecido pela Colorado State University,[16] compreendendo implementações de quatro algoritmos de reconhecimento que servem como ponto de partida.

3.3.3. Padrão da Íris

A idéia do valor da íris como fonte de informação biométrica confiável, única para cada indivíduo, veio à tona em 1965. A íris contém um rico padrão composto de fibras colágenas, rugas, sulcos, estrias, veias, sardas, fendas, buracos e cores. Embora a tecnologia biométrica de reconhecimento pelo padrão da íris seja relativamente nova, ela tem se mostrado bastante precisa e estável. Dentre poucos sistemas descritos na literatura, o mais conhecido é o IrisCode [Daugman 1999].

[11] `http://www.nist.gov/humanid/feret/`.
[12] `http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/`.
[13] Competição de janeiro/2006, sob condução do NIST. `http://www.frvt.org/`.
[14] Conferência internacional em janeiro/2006, em Hong Kong.
[15] `http://www.intel.com/technology/computing/opencv/index.htm`.
[16] `http://www.cs.colostate.edu/evalfacerec/`.

Para o processo de **aquisição** das imagens da íris, os sistemas comerciais utilizam câmeras monocromáticas, já que os métodos de extração de características não se utilizam da cor. A maioria dos sistemas requer que o usuário posicione os olhos dentro do campo de visão de uma câmera de foco estreito. O posicionamento correto é obtido por meio de um *feedback* visual proporcionado por um espelho. Sistemas melhorados, com a utilização de mais de uma câmera, podem ser construídos para uso público e privado [Negin et al. 2000].

O processo de **extração** das características da íris para a criação de um *IrisCode* funciona simplificadamente da seguinte maneira (figura 3.9): (1) é localizada a imagem da íris na imagem adquirida, pela estimativa do centro da pupila; (2) o padrão da íris é isolado da pupila; (3) o padrão é demodulado para extração de sua informação de fase, quando são computados 256 bytes para a imagem da íris e outros 256 bytes representando a máscara para as áreas de ruído, para melhorar a precisão do comparador, perfazendo então um perfil de 512 bytes. Assim, um *IrisCode* é construído pela demodulação do padrão da íris. O processo utiliza uma transformada de Gabor (*complex-valued 2D Gabor wavelets*) para extrair, da estrutra da íris, uma seqüência de fasores (vetores no plano complexo), cujos ângulos de fase são quantizados em bits para compor o código final. A quantização leva em consideração apenas a que quadrante pertence o fasor. O processo é executado num sistema de coordenadas polares, que é invariante à alteração de tamanho da imagem e também invariante à alteração do diâmetro da pupila dentro da íris.

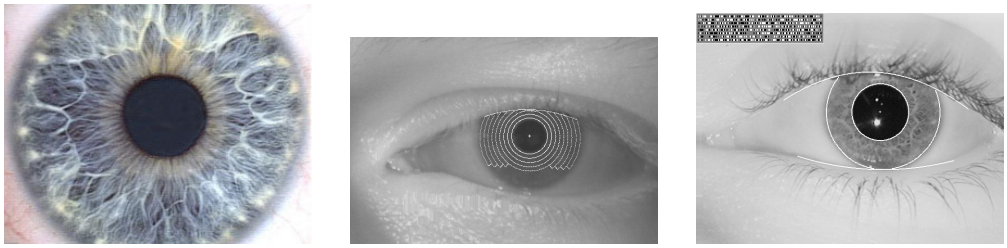


Figura 3.9. Imagem da íris adquirida sob condições ideais (esquerda). Fase de aplicação do algoritmo de extração de características (centro). Íris com seu *IrisCode* associado (direita).

O processo de **comparação** calcula uma medida da similaridade por meio da distância de Hamming normalizada, um método que simplesmente calcula a quantidade da divergência de bits entre as codificações. A chave para o reconhecimento da íris é a falha de um teste de independência estatística [Daugman 1993]. Este teste é implementado por um simples operador booleano *XOR* (OU EXCLUSIVO), aplicado aos vetores codificados dos padrões de íris. Os vetores são mascarados por meio do operador booleano *AND* (E lógico), para prevenir a influência de ruído produzido por lentes, distorções e iluminação.

A simplicidade do teste de comparação é um fator que proporciona alto desempenho. O desempenho do algoritmo é citado como sendo de 100.000 usuários por segundo numa CPU de 300MHz. A precisão dos sistemas biométricos baseados em íris também é um importante fator, que permite que a tecnologia baseada em íris seja adequada tanto para verificação como para identificação. Recente relatório de conclusão de avaliação conduzida pelo *International Biometric Group* cita o melhor ponto de ope-

ração (FMR,FNMR), de um sistema basedo em íris, como sendo $(0,00129\%, 0,583\%)$ [IBG 2005].

Os pontos fortes da tecnologia de autenticação biométrica baseada no padrão da íris são:

- $\oplus$ Dentre as seis principais tecnologias relacionadas neste trabalho, atualmente a íris é considerada como a biometria mais precisa, especialmente quanto a taxas de falsa aceitação (FAR), um importante aspecto de segurança. Portanto, poderia ser uma boa tecnologia para fins puramente de identificação.
- $\oplus$ Possui alto desempenho no processo de verificação. A codificação, comparação e tomada de decisão são computacionalmente tratáveis, com média de tempo de um segundo para a análise da imagem e codificação. Para o processo de identificação, o desempenho é muito bom, com velocidade de comparação de 100.000 registros por segundo numa CPU de 300 MHz.

Quanto aos pontos fracos, podemos citar:

- $\ominus$ A íris não é um alvo fácil. É um alvo pequeno (1 cm) para ser adquirido a uma distância de cerca de um metro. É um alvo móvel, localizado atrás de uma superfície refletora úmida e curvada, parcialmente oculta por pálpebras que piscam freqüentemente e que pode ser obscurecida por óculos, lentes e reflexos e é deformada com a dilatação da pupila. Portanto, exige a colaboração do usuário para a sua coleta.
- $\ominus$ Embora seja uma boa tecnologia para identificação, o desenvolvimento em larga escala é impedido por falta de base instalada. Ademais, criminosos não deixam traços da íris na cena do crime, o que enfraquece a possibilidade de sua utilização em aplicações de investigação criminal.

A maioria dos bancos de dados existentes foi criada para uso comercial e não está disponível publicamente. No entanto, pelo menos quatro bancos de dados estão disponibilizados para propósitos de pesquisa:

- CASIA - Um instituto de pesquisa da China (*Chinese Academy of Sciences, Institute of Automation*) disponibiliza um banco de dados contendo cerca de 3.000 imagens de íris pertencentes a cerca de 230 indivíduos diferentes.[17]
- UBIRIS - A Universidade de Beira Interior (Portugal) disponibiliza um banco de dados com cerca de 1.900 imagens da íris, contendo ruído e que simulam colaboração mínima do usuário.[18]
- CUHK - A Chinese University of Hong Kong oferece cerca de 250 imagens de íris para fins de pesquisa.[19]

[17] `http://nlpr-web.ia.ac.cn/english/irds/resources.htm`.
[18] `http://iris.di.ubi.pt`.
[19] `http://www2.acae.cuhk.edu.hk/~cvl/main_database.htm`.

- UPOL - Finalmente, 384 imagens de íris são disponibilizadas pela UPOL (Univerzita Palackého v Olomouci), da República Tcheca.[20]

Existe pelo menos um sistema de reconhecimento baseado em íris de código-fonte aberto. O sistema, implementado em MATLAB, basicamente usa como entrada uma imagem do olho e devolve como saída um perfil biométrico em código binário [Masek and Kovesi 2003].

3.3.4. Geometria da Mão

Várias tecnologias de verificação com base na geometria da mão evoluíram durante o último século, de dispositivos eletromecânicos para eletrônicos. Foi concedida, em 1960, a primeira patente para um dispositivo que media a geometria da mão, e registrava características para identificação posterior (uma máquina baseada em mecânica, projetada e construída por Robert P. Miller, sob o nome de *Identimation*). Nos anos 70 e 80, várias outras companhias lançaram esforços de desenvolvimento e implementação de dispositivos similares, pressionados pelas oportunidades de mercado. Atualmente, modernos leitores de mão executam funções de controle de acesso, registro de ponto de empregados e aplicações de pontos de venda [Zunkel 1999].

O processo de **aquisição** é baseado na geometria da mão. O comprimento, largura, espessura e curvatura dos dedos e da palma da mão, e a localização relativa destas características, distingue as pessoas entre si. O dispositivo leitor de geometria da mão usa uma câmera para capturar imagens em preto e branco da silhueta da mão (figura 3.10). Não são registrados detalhes de textura, impressões digitais, linhas e cores. Em combinação com um refletor e espelhos laterais, duas imagens distintas são produzidas, uma de cima e uma de lado. Este método é conhecido como **orto-leitura**.

Figura 3.10. Medidas típicas da geometria da mão. O modelo esquemático (esquerda) pode ser apreciado na imagem real (centro) obtida de um dispositivo (direita).

A imagem é obtida com a colaboração do usuário, que coloca a mão numa plataforma especial, contendo pinos para contenção e localização da mão. Estes pinos, que se projetam da plataforma, posicionam a mão do usuário para assegurar uma captura de imagem mais precisa, com melhor qualidade [Sanchez-Reillo et al. 2000]. Uma câmera, localizada acima da plataforma, é ativada quando sensores de pressão localizados próximos aos pinos da plataforma são ativados, indicando que o objeto de interesse está

[20] `http://phoenix.inf.upol.cz/iris/`.

corretamente posicionado. A fotografia é tomada mostrando a silhueta e imagem lateral da mão.

O processo de **extração** trabalha sobre a imagem adquirida. A imagem obtida é convertida para preto e branco, caso seja colorida, e pequenos desvios eventuais são corrigidos. Para estes ajustes, são úteis as imagens dos pinos existentes na plataforma. Um algoritmo de detecção de bordas é aplicado para extrair o contorno da mão. O processamento dos dados extraídos pode fornecer um perfil de apenas 9 bytes de dados, suficientemente pequeno para ser armazenado com facilidade em dispositivos dedicados e também é adequado para trânsito em redes de banda limitada.

No processo de **comparação**, a representação obtida é comparada com o perfil armazenado. A comparação pode envolver, por exemplo, acumulação de diferenças absolutas nas características individuais, entre a representação de entrada e o perfil armazenado. Para o cálculo da similaridade entre os dois vetores, são utilizados algoritmos baseados em distância euclidiana, distância de Hamming, modelos de mistura gaussiana (GMM—*Gaussian mixture models*) ou redes neurais. Os melhores resultados são apresentados pelos algoritmos baseados em GMMs [Sanchez-Reillo et al. 2000]. Para a acomodação dos fatores naturais e ambientais que alteram o formato da mão das pessoas, os dispositivos leitores podem possuir um processo de atualização dos perfis armazenados. Este processo é executado sob certas condições, durante o processo de comparação. Esta acomodação do perfil atualiza a descrição matemática armazenada quando a diferença entre a amostra e o perfil atinge um limite pré-determinado.

As características individuais da mão não são muito descritivas e este método de autenticação possui taxas de erro relativamente altas. Apesar disso, os sistemas de verificação com base na geometria da mão são bastante difundidos. Uma avaliação de cenário efetuada em 2001 pelo BWG relata uma taxa de erros de cruzamento de FAR$\times$FRR (ou seja, a EER) em torno de 1,5% para esta tecnologia [Mansfield et al. 2001].

Os pontos fortes da tecnologia de autenticação biométrica baseada no formato da mão são:

- $\oplus$ A coleta das características é fácil e não intrusiva.
- $\oplus$ A computação é bastante simples e os perfis são pequenos, o que torna fácil a construção de sistemas dedicados isolados. O pequeno tamanho do perfil (9 a 35 bytes) reduz as necessidades de armazenamento.
- $\oplus$ Adequado para integração com outras biometrias, em particular impressão digital e impressão palmar
- $\oplus$ Não relacionado a registros policiais e criminais.

Quanto aos pontos fracos, podemos citar:

- $\ominus$ Assim como na tecnologia de impressões digitais, a geometria da mão é medida quando o usuário pressiona uma superfície. Este contato pode despertar preocupações públicas com higiene.

⊖ Não é suficientemente distintiva para identificação, sendo adequada apenas para aplicações de verificação.

A tecnologia baseada no formato da mão é utilizada essencialmente em pequenos sistemas, uma vez que tal característica biométrica não fornece unicidade suficiente para identificação em larga escala.

3.3.5. Dinâmica da Assinatura

A assinatura pode ser ***off-line*** ou **estática**, aquela impostada em documentos de papel, escrita por meio convencional e posteriormente adquirida por meio de uma câmera ou scanner. Pode ser ainda ***on-line*** ou **dinâmica**, aquela efetuada num dispositivo eletrônico preparado para capturar, com alto grau de resolução, as característica dinâmicas temporais da assinatura, como a trajetória da caneta, a pressão, direção e elevação do traço.

O processo de **aquisição** pode ser baseado numa abordagem estática ou dinâmica. A abordagem estática data de 1975. Várias abordagens de análise automatizada são baseadas em características como número de contornos interiores e número de componentes de inclinação. Entretanto, a falta de informação dinâmica torna o processo automatizado de verificação estática bastante vulnerável a fraudes. O problema da verificação automática de assinaturas estáticas atraiu grande atenção nos últimos anos, mas os resultados não têm fornecido a precisão requerida por muitos problemas de segurança. As técnicas de abordagem criadas nos últimos 20 anos incluem transformadas 2D, histogramas de dados direcionais, curvatura, projeções verticais e horizontais do traço da assinatura, abordagens estruturais, medidas locais no traço, posição de pontos característicos. Um dos melhores resultados tem sido fornecido pela análise baseada no tamanho das distribuições granulométricas locais [Sabourin et al. 1997].

A abordagem dinâmica é bem mais interessante. A verificação da dinâmica da assinatura está baseada nas características do processo de assinatura em si. Um modo temporal de representação da assinatura contém mais informação, o que pode tornar o processo mais preciso. Contudo, este modo necessita de dispositivos especiais. Os dispositivos normalmente podem ser divididos em três tipos, de acordo com com a parte do dispositivo responsável pela aquisição: aquisição por meio da caneta, aquisição por meio da superfície e aquisição por meio de ambas.

O processo de **extração** de características se baseia principalmente na componente temporal. Na análise dinâmica, são introduzidas as noções de tempo e pressão, além do espaço bidimensional do papel. Os dispositivos utilizados podem, por exemplo, registrar um fluxo de vetores penta-dimensionais colhidos em pontos temporais equidistantes. Esses vetores poderiam, por exemplo ser compostos por $A = (x, y, p, \theta_x, \theta_y)$, onde x e y correspondem à posição, p corresponde à força axial exercida pela caneta e θ_x e θ_y registram os ângulos da caneta em relação ao plano xy. Esta informação adicional é bastante útil na prevenção de fraudes. Um arquivo de assinaturas contendo funções temporais de posição, pressão, azimute e elevação possui normalmente um tamanho entre 5 KB e 10 KB. Formatos mais eficientes e compressão na razão 3:1 permitem o armazenamento em arquivos de 1 KB a 2 KB.

Na análise de assinaturas dinâmicas, as abordagens de **comparação** incluem as

medidas de distâncias euclidianas entre as trajetórias de canetas, medidas de correlação regional e reconhecimento temporal-probabilístico como as cadeias de Markov ocultas. Afinal, o problema pode ser reduzido à classificação temporal. Durante os últimos 30 anos, numerosos algoritmos e modelos foram desenvolvidos. O conjunto de características no qual o processo de decisão está baseado, é constituido de funções temporais como pressão, posição, velocidade e aceleração, representadas por conjuntos de valores discretos periódicos e representadas por valores paramétricos obtidos com base no processamento de tais funções. Os métodos podem ser acomodados em quatro grupos:

1. *Classificadores probabilistas* - Estes métodos são baseados nas distribuições da densidade de probabilidades do conjunto de características genuíno e do conjunto de características em geral. Uma distância entre estas duas distribuições é determinada para fixar o grau de importância de dada característica. A decisão é baseada na distância Euclidiana, computada sobre um conjunto de características.

2. *Classificadores elásticos* - Esta técnica mais antiga, obscurecida desde o advento das cadeias de Markov ocultas, é baseada na utilização de DTW (*Dynamic Time Warping*) [Myers and Rabiner 1981]. Esta técnica computa as distâncias temporais mínimas entre um vetor de entrada e os vetores-modelo. Existem diferenças de tempo não-lineares entre as características das assinaturas produzidas pela mesma pessoa. O objetivo é encontrar o alinhamento temporal ótimo entre a assinatura de referência e a assinatura sob verificação.

3. *Redes neurais* - Esta ferramenta de Inteligência Artificial tem sido explorada pra a verificação dinâmica de assinaturas, mas o desempenho registrado tem sido inferior aos outros métodos.

4. *Cadeias de Markov ocultas* - Cadeias de Markov ocultas (HMM—*Hidden Markov Models*) são o meio mais popular de classificação temporal, com aplicações em áreas como reconhecimento de discurso, escrita e gesticulação. Informalmente, uma cadeia de Markov oculta é uma variante de uma máquina de estados finita e não-determinística, onde os estados e transições possuem associações probabilistas [Rabiner and Juang 1986]. Inspirada pelo sucesso da aplicação de HMMs ao reconhecimento de caracteres, este agora é o modelo com melhor desempenho na verificação de assinatura. A vantagem para esta tarefa advém da possibilidade de aceitar variabilidade, ao mesmo tempo em que se captura características individuais da assinatura.

Os pontos fortes da tecnologia de autenticação biométrica baseada na dinâmica da assinatura são:

- $\oplus$ A assinatura dinâmica é uma combinação de informação e biometria. O conteúdo e modo da escrita podem ser escolhidos e até mesmo alterados pelo usuário.

- $\oplus$ Possui grande aceitação por parte do usuário.

- $\oplus$ A assinatura dinâmica é bastante difícil de ser fraudada. A comunidade interessada em autenticação por meio da dinâmica de assinatura define o nível de sofisticação do

fraudador em categorias, como *zero-effort forgery*, *home-improved forgery*, *over-the-shoulder forgery* e *professional forgery*. Esta divisão em categorias por nível de sofisticação ainda não existe em outras tecnologias biométricas.

Quanto aos pontos fracos, podemos citar:

- ⊖ O custo dos dispositivos de aquisição é alto.

- ⊖ Esta característica biométrica possui alta variabilidade. Existem, ainda, muitas pessoas com assinaturas inconsistentes. Assim, os sistemas de verificação podem ser exigidos a apresentar a possibilidade de configuração de limiares de decisão por usuário.

Embora esta não seja uma das soluções biométricas mais seguras, ainda se justifica o uso da mesma nas práticas negociais, pois trata-se de um método *de facto* para verificação da identidade de uma pessoa. Esta tecnologia, quando utilizada para verificação (busca 1:1), ao invés da identificação (busca 1:N), possui um futuro bastante promissor. Por este motivo, várias pesquisas vêm sendo desenvolvidas, baseadas nesta tecnologia. Por exemplo, um protótipo de sistema de autenticação baseado em assinaturas dinâmicas foi construído na UNISINOS usando redes neurais do tipo *cascade-correlation* como mecanismo de comparação, relatando bons resultados de precisão, com um ponto de operação (FAR, FRR) estimado em (2,6%, 3,6%) [Heinen and Osório 2004].

Abordagens para localização da caneta e estimativa de orientação usando luz visível foram desenvolvidas, o que pode finalmente baixar o custo de aquisição de assinatura e pode até mesmo levar a assinaturas tridimensionais [Munich and Perona 1998].

O projeto BISP[21] visa desenvolver canetas multi-sensoriais para registro e análise de biometria comportamental e características neuromotoras, ambas baseadas na cinemática e na dinâmica da escrita em geral e da assinatura em particular [Hook et al. 2003].

Resultados relatados na primeira competição internacional de verificação por dinâmica da assinatura (SVC 2004)[22] relatam taxas de EER entre 2,89% e 16,34% para o melhor e pior algoritmo. Estão disponíveis no *site* da competição, arquivos de assinatura adquiridos de 40 usuários. Cada usuário contribuiu com 20 assinaturas. Por razões de privacidade, os usuários foram alertados para não contribuir com suas assinaturas reais, mas sim com assinaturas "inventadas". Para cada assinatura, existe uma assinatura forjada, perpetrada por falsários aos quais foi permitido assistir a uma exibição da impostação da assinatura. Existem assinaturas no estilo chinês (ideogramas) e no estilo latino (alfabeto latino da esquerda para a direita). Os arquivos de dados contêm vetores de dados de posição, pressão, azimute, elevação, registro de caneta em contato e registro de tempo. Este banco de dados pode ser bastante útil para a avaliação de algoritmos em desenvolvimento [Yeung et al. 2004].

[21] `http://www.bisp-regensburg.de/`.
[22] `http://www.cs.ust.hk/svc2004/`.

3.3.6. Padrão de voz

A autenticação por meio da voz tem sido uma área de pesquisa bastante ativa desde os anos 70. Atualmente, os sistemas podem ser divididos em classes, de acordo com o protocolo estabelecido:

1. *Texto fixo* - O usuário pronuncia uma palavra ou frase pré-determinada, secreta, gravada durante a fase de registro.
2. *Dependente do texto* - O usuário é solicitado, pelo sistema de autenticação, a pronunciar algo específico, dentre as diversas opções previamente registradas no sistema. Neste caso, a fase de registro é bastante longa. É similar ao protocolo de texto fixo, com um número maior de opções.
3. *Independente do texto* - O usuário pronuncia frases conforme seu desejo. O sistema processa qualquer discurso do usuário.
4. *Conversacional* - O usuário é interrogado, pelo sistema de autenticação, com perguntas cujas respostas são secretas, tornando-se um protocolo misto de conhecimento e biometria. É um protocolo similar ao dependente de texto, sendo que as frases previamente gravadas possuem um certo grau de segredo.

Para auxiliar o processo de **aquisição**, existem numerosos transdutores para transformar as ondas acústicas de voz em ondas eletromagnéticas. A quantidade de espaço de armazenamento necessária para os dados de voz sem tratamento dependem da taxa de amostragem, níveis de quantização e número de canais (mono-canal na maioria das vezes). Por exemplo, um sinal de voz amostrado a uma taxa de 16 kHz, com um nível de quantização de 16 bits, utiliza cerca de 31 KB por segundo de sinal.

Para a aplicação de ferramentas matemáticas, sem perda de generalidade, o sinal de voz deve ser representado por uma seqüência de vetores de características. O processo de **extração** pode se basear: (1) na abordagem tradicional, por meio de PCA (*Principal Component Analysis*) e FA (*Factor Analysis*); (2) na abordagem de estimativa de médias e covariâncias; e (3) na estimativa de divergências [Campbell 1997].

O processo de **comparação** das características extraídas pode ser suportado por vários métodos. Os principais métodos de abordagem para comparação dos dados de voz estão listados a seguir. Existem trabalhos que comparam algumas destas abordagens, como por exemplo [Yu et al. 1995].

- *DTW - Dynamic Time Warping* - Permite a compensação da variabilidade humana inerente ao padrão de voz. Método mais usado para verificação dependente do texto. Atualmente pouco utilizado como algoritmo *per se*, mas sim como um suplemento ao processo de decisão.
- *Métodos Estatísticos (HMM e GMM)* - Recaem na modelagem paramétrica do sinal de voz. A modelagem pode ser dependente do tempo, por meio da utilização de cadeias de Markov ocultas (HMM), ou não dependentes do tempo, por meio da utilização de modelos de mistura gaussiana (GMM). Os valores dos parâmetros

devem ser obtidos de dados de treinamento, o que é um ponto crítico nos métodos estatísticos: dados suficientes precisam ser obtidos para "treinamento". O método HMM é bastante comum para sistemas dependentes de texto. No entanto, o método GMM é agora o modelo dominante para reconhecimento de voz, freqüentemente em combinação com um provedor de informação de alto nível, como DTW.

- *VQ - Vector Quantisation* - Raramente usado, pois somente consegue superar os métodos estatísticos quando existem poucos dados disponíveis.
- *Redes Neurais* - Redes neurais têm sido usadas em pesquisas de reconhecimento de voz independente de texto, treinadas com dados de usuários genuínos e usuários impostores.
- *SVM - Support Vector Machines* - Esta abordagem tem sido proposta em pesquisas recentes (desde 1996). Os resultados relatados têm sido superiores aos resultados de GMMs.

Os pontos fortes da tecnologia de autenticação biométrica baseada no padrão de voz são:

⊕ A voz, assim como a face, é uma biometria usada instintivamente pelas pessoas para autenticação mútua.

⊕ Sistemas com infra-estrutura telefônica constituem o principal alvo do reconhecimento de voz. A fala com o objetivo único de autenticação (autenticação ativa), pode ser um tanto quanto anti-natural, mas em situações onde o usuário já tem mesmo de falar, o protocolo de autenticação se torna passivo, amigável e não-intrusivo.

⊕ Esta tecnologia utiliza dispositivos baratos, e além disso é facilmente desenvolvida sobre uma infra-estrutura já existente e amplamente espalhada, como o sistema telefônico.

⊕ Permite protocolos de autenticação de segurança incremental. Por exemplo, quando maior confiança é necessária, o sistema pode esperar por mais dados de voz. Outro exemplo, pode ser utilizado um protocolo de biometria conversacional, combinado com verificação de conhecimento. Outro exemplo, o protocolo pode verificar a identidade continuamente durante a conversação.

⊕ Em aplicações de texto independente e aplicações conversacionais, os usuários não necessitam de um processo separado de autenticação, o que torna o processo totalmente integrado.

Quanto aos pontos fracos, podemos citar:

⊖ É possível a imitação por pessoas habilidosas ou a utilização de gravações da voz do usuário legítimo para fraudar o sistema. Além disso, existem sistemas de síntese que podem ser treinados para imitar a voz de pessoas.

⊖ A tecnologia *text-to-speech* torna possível a criação de identidades não existentes, em sistemas de registro e autenticação remotos.

⊖ A qualidade do sinal de áudio é suscetível ao ruído do ambiente. Além disso, são introduzidas distorções na captação do sinal pelo microfone e na transmissão do sinal através do canal.

⊖ O padrão de voz é bastante frágil, pois pode ser alterado pelo estado do usuário (saúde, emoção, pressa, sono, preguiça, entre outros).

A tecnologia baseada no padrão de voz possui vários recursos associados, como bancos de dados e aplicativos. A utilização de bancos de dados padronizados para desenvolvimento e avaliação, mostrou seu valor no progresso das pesquisas de reconhecimento de voz e reconhecimento de discurso. Uma visão geral dos diversos bancos de dados disponíveis é proporcionada por [Campbell and Reynolds 1999], do qual foram extraídos os exemplos mais comuns:

- LDC - *Linguist Data Consortium* (EUA) - Dá suporte à pesquisa, por meio da criação e compartilhamento de recursos linguísticos, como dados, ferramentas e padrões. Mantém vários bancos de dados, inclusive o *YOHO Speaker Verification*, útil para experimentos com reconhecimento de voz dependente de texto.[23]

- ELRA - *European Language Resources Association* (Luxemburgo) - Mantém vários bancos de dados em línguas européias.[24]

O pacote LIA_RAL, da Université d'Avignon, na França, é um software de reconhecimento de voz, de código fonte aberto, implementado em C++.[25] É capaz de reconhecer vários tipos de características e tem sido usado nas avaliações do NIST. Pode servir como base de comparação com outros sistemas.

A principal competição em reconhecimento de voz é a série de avaliações conduzida pelo NIST. A série, iniciada em 1996, é focada fortemente no reconhecimento de voz por meio telefônico [Reynolds et al. 2000].

As taxas de erro para sistemas de autenticação por meio da voz são muito dependentes da aplicação. Isto quer dizer que bons resultados obtidos em competições de avaliação ou publicados em especificações de fabricantes, não significam necessariamente que os mesmos serão obtidos na prática, nas aplicações específicas. Esta tecnologia está amadurecida pelas pesquisas, mas alguns problemas permanecem ainda não resolvidos. São problemas relacionados ao usuário, ao ambiente e ao canal. O desempenho depende muito das condições de aquisição e teste. Mesmo assim, competições internacionais tentam estabelecer taxas de erros aproximadas que permitam comparações com outras tecnologias. Por exemplo, em competição aberta conduzida pelo NIST em 2003 foi obtida uma taxa EER de 5,3% [Przybocki and Martin 2004].

[23] `http://www.ldc.upenn.edu/`.
[24] `http://www.elra.info/`.
[25] `http://www.lia.univ-avignon.fr/heberges/ALIZE/LIA_RAL/index.html`.

3.3.7. Comparativo sumário

Uma comparação entre as seis tecnologias biométricas apresentadas nesta seção é mostrada na tabela 3.2 [Jain et al. 2004]. Esta comparação avalia o grau (alto, médio ou baixo) com que cada tecnologia satisfaz as propriedades desejáveis de características biométricas discutidas na seção 3.2.1; embora resumida, ela permite obter um panorama geral dessas tecnologias.

Dentre as características biométricas apresentadas, a impressão digital e a íris são as mais estáveis ao longo do tempo. A íris pode fornecer a maior precisão, embora a impressão digital seja a mais utilizada. A tecnologia baseada no formato da mão já tem seu nicho de mercado bastante consolidado. As tecnologias de face e assinatura possuem a aceitação do usuário e são de fácil coleta.

A aplicação de uma determinada tecnologia biométrica depende fortemente dos requisitos do domínio da aplicação. Nenhuma tecnologia pode superar todas as outras em todos ambientes de operação. Assim, cada uma das tecnologias é potencialmente utilizável em seu nicho apropriado, ou seja, não existe tecnologia ótima.

Biometria	Universalidade	Unicidade	Permanência	Coleta	Aceitação
Digital	Média	Alta	Alta	Média	Média
Face	Alta	Baixa	Média	Alta	Alta
Íris	Alta	Alta	Alta	Média	Baixa
Mão	Média	Média	Média	Alta	Média
Assinatura	Baixa	Baixa	Baixa	Alta	Alta
Voz	Média	Baixa	Baixa	Média	Alta

Tabela 3.2. Comparativo sumário entre as características de alguns identificadores biométricos

3.4. Arquiteturas

3.4.1. Armazenamento

Existem várias possibilidades de **distribuição dos processos** componentes de um sistema biométrico. Num caso extremo, podemos ter todos os processos localizados no dispositivo de aquisição, como é o caso de pequenos sistemas de acesso físico. Neste caso, os processos de aquisição, extração e comparação, bem como o banco de dados de perfis biométricos, estão todos localizados no mesmo equipamento ou, no máximo, limitados a uma rede local.

Noutro extremo, podemos ter um ampla distribuição dos processos. Vamos supor um sistema de larga escala, com centenas de milhares de perfis registrados e diversos locais de aquisição de biometria, como é o caso de um sistema de autenticação de clientes bancários em máquinas de auto-atendimento. O processo de aquisição pode se dar em diversos pontos do sistema. O armazenamento dos perfis pode se dar em *smart cards* em poder do usuário. Uma cópia do perfil pode ou não ser armazenado em servidor central para o caso de uma reemissão de cartões extraviados. Os processos de extração e comparação também podem estar distribuídos, dependendo da conveniência para a arqui-

tetura do sistema. Estes processos podem ser locais (junto ao dispositivo de aquisição) ou remotos (em servidor ou até mesmo no próprio *smart card*).

A forma de **armazenamento dos perfis** depende do tipo de aplicação para qual o dispositivo biométrico será utilizado e do tamanho dos perfis. Os perfis, como visto inicialmente, são os dados armazenados que representam a medida biométrica de um usuário cadastrado, utilizados pelo sistema biométrico para posterior comparação com outras amostras submetidas. De uma forma geral, os perfis podem ser armazenados de forma local, remota ou distribuída.

O armazenamento **local** corresponde ao armazenamento no próprio dispositivo de aquisição, ou em computador a ele acoplado por meio da rede local. Esta forma de armazenamento não é adequada para o caso de aplicações com um grande número de usuários ou quando o usuário precisa ser verificado em diversos locais diferentes. Quanto à segurança, os riscos de comunicação são eliminados, uma vez que não é necessária a transmissão dos perfis biométricos, e o impacto de um possível comprometimento é reduzido em extensão, pois somente atinge os dados locais. Por exemplo, os pequenos e médios sistemas de controle de acesso físico geralmente se valem de armazenamento local. O sistema armazena os perfis dos usuários candidatos a acesso a determinado local. A quantidade de usuários pode variar de unidades, no caso de acesso a uma residência, ou centenas, no caso de controle de acesso a academias, ou milhares, para controle de acesso a grandes prédios ou instalações.

O armazenamento **remoto** corresponde ao armazenamento em um servidor, o que quase sempre quer dizer uma base de dados centralizada. Esta solução é adequada para aplicações onde o número de usuários é grande ou quando é necessária verificação remota. Este processo pode ser comprometido quando a segurança dos dados é ameaçada por sistemas de comunicação ou redes vulneráveis ou por abuso de privilégios na manipulação da base de dados. Os sistemas de identificação (busca 1:N) de larga escala geralmente se utilizam da modalidade de armazenamento remota. Este sistemas geralmente comportam milhões de usuários e possuem requisitos mais refinados de precisão e desempenho. Os sistemas de verificação (busca 1:1) de larga escala podem ou não se valer desta modalidade de armazenamento.

O armazenamento **distribuído** corresponde ao armazenamento em dispositivos que ficam em poder do usuário, normalmente sob a forma de *smart cards*. O método de armazenamento de perfis utilizando cartões magnéticos permite que o usuário carregue seu próprio perfil para a utilização nos dispositivos de verificação, sendo indicado para aplicações onde o grupo de usuários seja numeroso demais para ser armazenado numa base de dados central, quando é necessário que os usuários sejam verificados remotamente ou quando há necessidade de uma transmissão rápida dos perfis.

A **entidade armazenadora** dos perfis biométricos possui sérias responsabilidades derivadas das preocupações com privacidade e possibilidade de mau uso dos dados. Os pioneiros na adoção da tecnologia de autenticação baseada em biometria normalmente estão baseados nos próprios recursos para implementação e gestão da infra-estrutura necessária para dar suporte à autenticação. Este cenário pode sofrer alterações, dependendo da entidade armazenadora e da portabilidade do dispositivo de aquisição.

Quanto à entidade armazenadora, podemos considerar dois tipos de entidades, que chamamos de agentes autorizados e agentes de confiança. Um **agente autorizado** é uma organização que adota a autenticação biométrica e assume a responsabilidade por registrar e administrar os perfis biométricos de seus usuários conforme os requisitos dessa autenticação. Um **agente de confiança**, por sua vez, é uma organização à qual é delegada a responsabilidade pelos dados biométricos: ela se encarrega do registro e administração de perfis de usuários e presta um serviço de verificação de credenciais biométricas a entidades que desejam utilizar essa forma de autenticação. Um exemplo de agente autorizado seria um banco que decide usar biometria para autenticar seus próprios clientes, e um exemplo de agente de confiança seria um órgão governamental responsável por gerenciar dados biométricos que seriam usados para fins de autenticação em vários setores do serviço público.

Já quanto ao dispositivo de aquisição, vamos considerar que ele pode ser administrado ou livre. O dispositivo de aquisição **administrado** é um equipamento que está localizado em pontos específicos de acesso ao sistema, e que pode servir a vários usuários, cada um por sua vez. Já o dispositivo de aquisição **livre** é um equipamento que está em poder do usuário, como por exemplo, um *palmtop* ou um telefone celular.

A tabela 3.3 mostra os cenários derivados das diferentes combinações possíveis entre agentes de armazenamento e dispositivos de aquisição. No cenário chamado "pioneiro" (com agente autorizado e dispositivo administrado), temos a figura de um agente autorizado que registra cada usuário e armazena o perfil para uso posterior, quando o usuário desejar fazer uma transação. Além disso, instala e gerencia a infra-estrutura necessária para os dipositivos de aquisição. Neste cenário, o início de novos projetos é facilitado, pois o agente pode decidir-se pelo uso da biometria unilateralmente, ou seja, não depende de infra-estrutura oficial. A integridade fim a fim do sistema também pode ser controlada pela entidade. É claro que este agente autorizado tem que suportar todo o custo e o risco de implementar e gerenciar o sistema. Outro ponto fraco é que o usuário deve se registrar novamente a cada novo agente ou organização, talvez usando outras características biométricas. Isto pode levar a preocupações com privacidade, uma vez que o usuário pode relutar em confiar sua característica biométrica a diversas organizações diferentes.

Entidade	Dispositivo administrado	Dispositivo livre
Agente autorizado	Cenário Pioneiro	Cenário Temerário
Agente de confiança	Cenário Organizado	Cenário Audacioso

Tabela 3.3. Cenários possíveis ao se combinar os dispositivos de aquisição (administrados ou livres) e agentes de armazenamento (autorizados ou de confiança)

No cenário dito "temerário" (com agente autorizado e dispositivo livre), temos a figura de um agente autorizado que registra cada usuário e cada dispositivo. Este é um cenário considerado ainda improvável atualmente, pois o agente autorizado fica responsável pelas condições de segurança do sistema, embora não possua o gerenciamento completo dos dispositivos de aquisição. Um exemplo atual seria um sistema de uma instituição financeira, que oferece acesso aos usuários por meio de seus telefones celulares.

No cenário considerado "organizado" (com agente de confiança e dispositivo ad-

ministrado), temos a figura de um agente de confiança que fornece um *smart card* com *status* oficial, a ser usado em múltiplas entidades integrantes do sistema. O cartão contém o perfil biométrico e permite autenticação local em dispositivos de aquisição fixos, espalhados entre várias entidades integrantes do sistema. O agente de confiança mantém cópia do perfil para nova emissão de cartão, quando necessário. Este cenário assume que um número pequeno de dispositivos de aquisição é adotado como padrão. O custo do sistema diminui bastante para os agentes autorizados, já que ele pode ser compartilhado por todos. No entanto, as entidades integrantes dependem da infra-estrutura do agente de confiança. A aceitação do sistema pelo usuário é aumentada, pois ele passa a ser o detentor de seu próprio perfil biométrico, armazenado em cartão. Todavia, o usuário pode relutar em utilizar os dispositivos de aquisição, com a suspeita de que os mesmos poderiam ter sido adulterados para capturar as características biométricas para utilização posterior.

No quarto cenário, batizado de "audacioso" (com agente de confiança e dispositivo livre), o agente de confiança distribui o perfil associado ao usuário. Além disso, o usuário se utiliza de dispositivos de aquisição que estão em seu próprio poder. Temos um cenário onde a sensação de privacidade do usuário é aumentada, pois agora o usuário carrega consigo o seu próprio dispositivo de aquisição e o seu próprio perfil biométrico. Os agentes autorizados mantêm a diminuição de seus custos pelo compartilhamento dos mesmos, mas o usuário tem o inconveniente de ter de carregar consigo o dispositivo.

3.4.2. Segurança

A segurança de sistemas biométricos pode ser diferenciada em, ao menos, três importantes aspectos: a precisão do sistema, representada pelas medidas clássicas estatísticas de taxas de falsa aceitação e falsa rejeição; a arquitetura do sistema e implementação do sistema em si, representada pela interconexão física e lógica entre suas diversas partes componentes e a aplicação; e, a robustez do sistema, representada pela sua capacidade de resistência à fraude e falsificação intencionais.

A precisão pode ser avaliada por meio de bancos de dados representativos e um conjunto básico de medidas aceitas. Com respeito à arquitetura do sistema, existem procedimentos, embora mais complexos, para avaliar a segurança de um projeto e sua implementação de uma maneira padronizada. No entanto, a robustez é a mais difícil de ser avaliada, pois é fácil mostrar que um sistema biométrico pode ser fraudado, mas é muito mais difícil mostrar que um sistema biométrico não pode ser fraudado. Assim, independentemente de quão preciso é o sistema e de quão bem projetada é a sua arquitetura, não se pode enunciar de antemão conclusões sobre a sua resistência a ataques. Esta seção se concentra em considerações sobre as vulnerabilidades de sistemas biométricos.

Vários padrões de **taxonomia de ataques** a sistemas biométricos foram apresentados. Os mais importantes são:

1. *Biometric Device Protection Profile* (BDPP) [DIN 2003].

2. *Department of Defense & Federal Biometric System Protection Profile for Medium Robustness Environments* (DoDPP) [Kong et al. 2002].

3. *U.S. Government Biometric Verification Mode Protection Profile for Medium Ro-*

bustness Environments (USGovPP) [Kong et al. 2003].

Os padrões de taxonomia listados são bastante similares em várias maneiras, mas mesmo assim não é trivial comparar a nomenclatura dos ataques entre eles. De qualquer modo, a abordagem de análise de ameças e contramedidas por meio do auxílio da construção sistemática de uma árvore de possibilidades, ou árvore de ataques, permanece como ferramenta útil de projeto. Um estudo para os sistemas focados em defesa contra FRR (*False Rejection Rate*), tenta cobrir os claros existentes, acrescentando outros níveis de hierarquia à árvore de ataques [Buhan et al. 2006]. Outra abordagem, que integra conceitos de gerenciamento e de segurança, propõe uma metodologia estruturada (*BASS model*) bastante abrangente na análise de vulnerabilidades [Leniski et al. 2003]. A lista a seguir apresenta apenas alguns exemplos de vulnerabilidades:

- *Vulnerabilidades no processo de aquisição* - Um ataque pode ser implementado de várias maneiras. Num ataque de coerção, os dados biométricos verdadeiros são apresentados usando a força ou outros métodos ilegais de persuasão. Num ataque de personificação, um usuário não autorizado altera seus dados biométricos para aparecer como um indivíduo autorizado, por exemplo por meio do uso de disfarces ou imitação. Num ataque de impostação, dados verdadeiros são apresentados por um usuário não autorizado. Por exemplo, os passos necessários para criar uma impressão digital sem colaboração do seu proprietário são descritos em [Putte and Keuning 2000]. Outro exemplo seria a apresentação de partes do corpo extraídas de seus usuários. Uma análise de ataques para o caso da impressão digital pode ser apreciada em [Uludag and Jain 2004].

- *Vulnerabilidades nos processos de extração e comparação* - A utilização de um cavalo de Tróia (*Trojan horse*) pode permitir um ataque que consiste em alterar o módulo de extração. Por exemplo, a corrupção do processo de extração pode ser programada para produzir um conjunto de caractarísticas favoráveis à aceitação do impostor. A corrupção do processo de comparação pode permitir a produção de escores superiores ao escore real e pode, ainda, permitir a modificação da decisão final produzida no módulo de comparação. Outros ataques interessantes podem ser executados [Schneier 1999]. O ataque *hill-climbing* envolve a submissão repetida de dados biométricos, com pequenas modificações entre cada repetição, com a preservação das modificações que resultem num escore melhorado. O ataque *swamping* é similar ao ataque de força bruta, e consiste na submissão de dados em abundância, na esperança de que seja alcançado pelo menos o escore necessário para autenticação.

- *Vulnerabilidades no processo de registro* - A segurança do processo de registro é de extrema importância, pois uma vez que um fraudador consiga colocar seu perfil biométrico no sistema, passará a ser tratado como usuário válido. Até mesmo possíveis ataques em conivência com o administrador do sistema devem ser analisados neste processo. Outro ataque poderoso é aquele dirigido ao banco de dados dos perfis biométricos armazenados (centralizado ou distribuído), para leitura ou modificação não autorizada dos perfis.

- *Vulnerabilidade nos canais entre os processos* - Em muitos sistemas reais, alguns módulos do sistema podem estar fisicamente distantes entre si. Em tais sistemas, os canais entre os processos podem constituir vulnerabilidades importantes. Ataques de repetição (*replay*) são os mais comuns.

Assim como outros mecanismos de segurança, qualquer sistema biométrico pode ser fraudado com um adequado investimento em tempo e dinheiro. Do ponto de vista do gerenciamento de riscos, a tarefa do projetista de segurança é fazer com que o custo para se violar a segurança do sistema seja superior ao benefício obtido com a violação. A única coisa que pode ser feita a favor da segurança é o incremento dos custos envolvidos para a consecução da fraude. A vantagem do projetista é que ele pode investir tempo e dinheiro previamente para tentar proteger o sistema contra todo ataque possível e imaginável. A vantagem do impostor é que ele apenas necessita usar a criatividade para encontrar um ataque ainda não pensado. Esta luta entre ataques e contramedidas pode ser bem exemplificado por meio de uma coletânea de ataques e contramedidas referente a um sistema hipotético baseado no padrão da íris [Ernst 2002]. Além dos mecanismos tradicionais de cifragem e estampilha de tempo, a lista a seguir apresenta algumas das principais contramedidas de caráter geral e outras que ainda estão em fase de pesquisa.

- Suporte na área de aquisição - Em aplicações biométricas nas quais a supervisão está presente quando os sujeitos estão submetendo seus dados biométricos, a probabilidade de um indivíduo ludibriar o sistema é substancialmente reduzida. Algumas aplicações simplesmente não permitem tal supervisão, como é o caso de autenticação de usuários via Web. Em outras aplicações pode existir uma solução de compromisso entre custo e segurança, como seria o caso de uma aplicação de autenticação de usuários em terminais de auto-atendimento de bancos.

- Detecção de repetição - O sistema pode se valer de uma propriedade das características biométricas como ferramenta de segurança. Afinal, é desprezível a possibilidade de dois exemplares biométricos serem exatamente iguais. O sistema poderia então descartar qualquer exemplar idêntico a um dos exemplares anteriores. O preço a pagar por tal ferramenta é custo do espaço de armazenamento e capacidade de processamento extra. Mesmo assim, uma solução econômica pode manter em histórico os códigos *hash* dos últimos exemplares colhidos de cada usuário. Uma coincidência exata em nova amostra indica um ataque de repetição. Outro método poderia ser a solicitação de reapresentação da biometria. Por exemplo, em sistemas baseados em dinâmica da assinatura, o usuário pode ser solicitado a assinar mais de uma vez, devendo o sistema certificar-se de que os exemplares de assinatura não sejam idênticos.

- Detecção de perfeição - A mesma propriedade do item anterior serve para a criação de outra contramedida aplicável a sistemas biométricos. Caso o exemplar apresentado seja idêntico ao perfil armazenado, é certo que houve vazamento do perfil biométrico.

- Resposta sumária - As respostas sumários ou ocultação dos dados (*hiding data*), servem para evitar ataques *hill-climbing*. Assim, o sistema deve fornecer apenas uma

resposta ao usuário não autenticado (NÃO), abstendo-se de explicar qual o motivo da recusa e abstendo-se, é claro, de informar qualquer valor de escore obtido.

- Desafio e resposta - Medida bastante apropriada contra ataques de repetição, o desafio e reposta envolve o envio de um desafio ao usuário, que deve responder apropriadamente para obter autorização. Em sistemas de voz, pode ser usada a verificação independente de texto ou a verificação conversacional.
- Detecção de vitalidade (*liveness detection*) - A detecção de vitalidade (ou detecção de vivacidade) num sistema biométrico de autenticação deveria assegurar que somente características reais, pertencentes a pessoas vivas, fossem aceitas como válidas. Isto tornaria o sistema mais seguro e aumentaria também o poder de não-repudiação. No entanto, até mesmo pequenos esforços podem levar à fraude em sensores biométricos atuais. Trabalhos descrevendo fraudes em impressões digitais [Sandstrom 2004], íris e imagens da face demonstram isto claramente. A detecção de vitalidade pode se dar no processo de aquisição ou no processo de extração de características.

Além das citadas, outras três contramedidas podem vir a se tornar importantes ferramentas de segurança: a utilização conjunta de várias biometrias, a aplicação de transformações irreversíveis sobre os dados biométricos (para aumentar a privacidade) e a combinação de biometria e *smart cards*. Vejamos mais detalhes quanto a estas contramedidas.

Multibiometria

Algumas limitações dos sistemas biométricos podem ser superadas com a utilização sistemas biométricos multimodais. A proposta de tais sistemas é aumentar a confiabilidade e atender os requisitos impostos por várias aplicações [Ross et al. 2006]. A obtenção de multiplicidade pode se dar em diversos pontos do sistema, conforme ilustrado na figura 3.11:

1. Múltiplas biometrias podem ser utilizadas (voz e face, por exemplo) ou múltiplas unidades da mesma biometria (dedos diferentes ou olhos diferentes, por exemplo).
2. Múltiplos sensores, como sensores óticos e capacitivos para impressão digital.
3. Múltiplas amostras da mesma biometria; por exemplo, múltiplas impressões do mesmo dedo.
4. Múltiplos comparadores, ou seja, diferentes abordagens para a representação de características e diferentes algoritmos de comparação.

O processo de fusão também pode se dar em diversos pontos do sistema (figura 3.11):

1. Fusão na amostra, ou seja, os diversos dados obtidos são concatenados em um único vetor de características com maior poder de diferenciação.

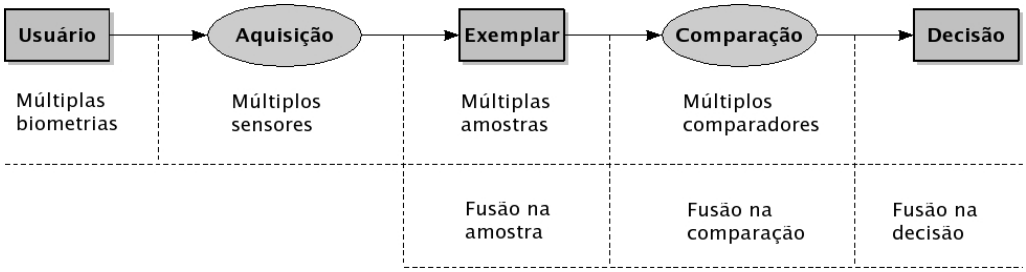


Figura 3.11. Dentro do processo genérico de um sistema biométrico, há diversos pontos para obter multiplicidade e há diversos pontos para efetuar a fusão.

2. Fusão na comparação, ou seja, os diversos escores de similaridade obtidos são combinados por meio de médias ponderadas.

3. Fusão na decisão, ou seja, as diversas decisões obtidas são combinadas para produzir uma única decisão.

O aumento de custo e a maior inconveniência para o usuário são as maiores barreiras para a utilização de sistemas biométricos multimodais em aplicações comerciais. No entanto, em aplicações de alta segurança, em aplicações de identificação de larga escala e em aplicações de varredura a utilização de tais sistemas é bastante adequada [Jain et al. 2004].

Biometria cancelável

Uma técnica conhecida como **biometria cancelável** pode aliviar as preocupações com privacidade e segurança. Trata-se de uma distorção intencional efetuada sobre os dados biométricos, por meio de uma transformação escolhida. Geralmente, as transformações não são reversíveis, de modo a proteger a característica biométrica original. Em caso de comprometimento, o perfil transformado pode ser cancelado, e outra variante pode ser criada por meio de outra transformação. As transformações podem ser aplicadas no domínio do sinal adquirido ou no domínio das características extraídas. As **distorções no domínio do sinal** se referem às transformações aplicadas aos dados biométricos adquiridos por meio do sensor. Exemplos de transformações neste domínio são a grade de deformação e a permutação de blocos. Na grade de deformação, a imagem original é estruturada dentro de uma grade alinhada com as características marcantes da mesma. Um algoritmo de deformação qualquer é então aplicado, com diferentes parâmetros para cada porção da grade. Já na permutação de blocos, uma estrutura de blocos é superposta à imagem, alinhada com pontos característicos da mesma. Os blocos da imagem original são então misturados de uma forma aleatória, mas repetível. Estas transformações são mais comumente aplicáveis a imagens 2D de face, impressão digital, íris e mão.

As **distorções no domínio das características** extraídas atuam sobre o perfil biométrico, geralmente por meio de um mapeamento irreversível. Assim, o sinal adquirido é processado da forma usual, e os atributos extraídos é que sofrem uma transformação. Por exemplo, seja um perfil biométrico representativo de uma impressão digital, representado por um conjunto de pontos de minúcias $M = (x_i, y_y, \theta_i); i = 1, \ldots, n$. As coordenadas

dos pontos podem ser transformadas através de um mapeamento baseado em polinômios. Como mostra a figura 3.12, cada coordenada x_i é transformada para uma nova coordenada X_i por meio de uma função polinomial de, digamos, terceira ordem $X = F(x)$. As coordenadas y e θ podem ser transformadas de modo similar por meio de $Y = G(y)$ e $\Theta = H(\theta)$.

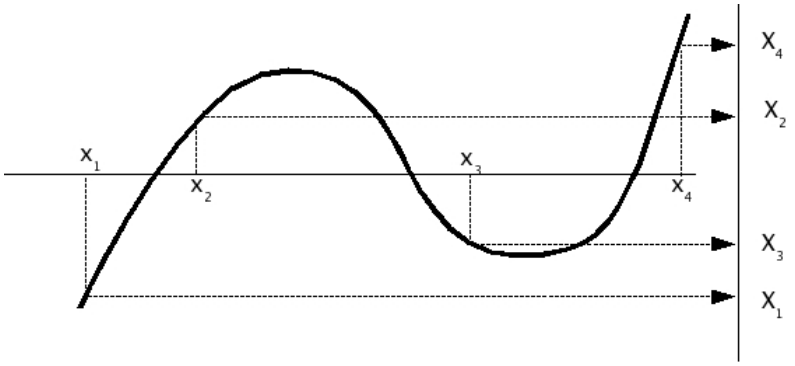


Figura 3.12. Um exemplo de mapeamento de uma das coordenadas de um conjunto de pontos de minúcias em um novo conjunto de coordenadas, por meio de uma transformação irreversível.

Outro exemplo seria o sistema proposto para tornar revogáveis os perfis biométricos de impressões palmares, por meio do armazenamento de vários códigos, compostos pelo *hash* da impressão palmar em conjunto com uma chave pseudo-aleatória [Connie et al. 2005].

Utilização de *smart cards*

Em muitas soluções de segurança, é usada uma infra-estrutura de chaves públicas, cuja confiabilidade repousa na existência de chaves privadas de conhecimento exclusivo dos seus proprietários. A criptografia assimétrica permite a criação de uma par de chaves, a chave pública D_k e a chave privada E_k. Ora, a chave privada proporciona alta resistência à fraude dificulta o ataque de força bruta, devido ao tamanho da chave. A chave privada (E_k) deve ser de conhecimento exclusivo do usuário e deve permanecer sempre em poder deste. Na prática, este requisito de segurança causa ao usuário um certo grau de inconveniência, pois a chave privada é muito grande para ser memorizada. Geralmente ela é armazenada em algum dispositivo, como um disco magnético, um *pen drive* ou um *smart card*.

No caso de armazenamento em cartão inteligente, é necessário que o usuário forneça um código para acessar o *smart card*, o que geralmente é feito pelo fornecimento de um número PIN ou uma senha. Isto leva a um ponto desvantajoso no armazenamento da chave privada. Para a proteção da mesma, é utilizado um código fornecido pelo usuário. Se o usuário se vale de um código forte (longo e complicado) não é prático memorizá-lo e se o usuário se vale de um código apenas guardado na memória, provavelmente será um código fraco. Isto reduz a segurança proporcionada pela chave privada para o nível de segurança proporcionado pelo código usado para acessá-la. Idealmente, a chave privada deveria ser protegida por um método tão seguro quanto a segurança que ela proporciona. Assim, estamos de volta ao impasse código forte (difícil de memorizar) $\times$ código fraco

(fácil de memorizar). Esta situação leva naturalmente ao desejo de utilização de biometria como código de acesso ao dispositivo que armazena uma chave privada. Esta combinação valiosa poderia proporcionar excelente nível de segurança.

As questões a serem consideradas nesta união de *smart card* e biometria envolvem a capacidade computacional dos cartões e o projeto de algoritmos eficientes de processamento de sinais, adequados ao ambiente proporcionado pela estrutura dos cartões. Além disso, para tratar as ameças de segurança proporcionadas pela comparação *off-card* de amostras e perfis biométricos, é necessário que o algoritmo de comparação seja implementado dentro do *smart card*. Atualmente, para algumas tecnologias biométricas, é possível também embutir o dispositivo de aquisição no próprio cartão, como é o caso da impressão digital e da voz. Alguns algoritmos específicos para reconhecimento de impressões digitais *on-card* foram desenvolvidos. Por exemplo, uma parceria entre companhias francesa e sueca desenvolveu um cartão com acesso por meio de impressão digital [Carlson 2003]. [26]

Embora existam *smart cards* com sensores de impressão digital ou microfones embutidos, a inserção de sensores de outras tecnologias biométricas no corpo de um *smart card* é assunto para o futuro próximo. O problema maior para a larga utilização desta facilidade é a diversidade de sistemas operacionais e ambientes de desenvolvimento. Uma alternativa promissora é a utilização de *Java Cards*, mesmo com a penalidade ao desempenho imposta por uma linguagem interpretada [Osborne and Ratha 2003]. Por meio do armazenamento, aquisição e comparação no *smart card*, o perfil biométrico fica circunscrito ao cartão. Este método é normalmente visto como o meio mais seguro de proteção biométrica em segurança da informação. Na prática, o *smart card* é tornado pessoal, posto que não pode ser acessado sem a autenticação biométrica apropriada. Os perfis biométricos nunca são expostos a ambientes não confiáveis e o usuário carrega consigo seus próprios perfis biométricos, o que soluciona várias questões relativas à privacidade das características biométricas.

A introdução de biometria para o acesso ao cartão que armazena uma chave privada também introduz um problema. O que acontece se a característica biométrica muda? Por exemplo, suponhamos que apenas o polegar direito seja usado para acesso e o usuário sofre um acidente que altera a sua impressão digital? Este problema da irrevogabilidade é o mesmo do usuário que esquece a senha de acesso a um certo recurso. É necessário alterar a senha. No caso descrito, é necessário que o usuário utilize os serviços da mesma entidade que registrou o seu perfil biométrico atualizar o cartão.

É razoável supor, então, que uma determinada aplicação possa contar com a existência de chaves privadas de usuários armazenada em *smart cards* e somente acessíveis por meio de dispositivos de aquisição embutidos no cartão. Mesmo assim, será necessário um dispositivo de leitura para interagir com o cartão. Outra camada necessária é uma camada de *software* localizada no computador que hospeda a aplicação principal, que geralmente toma a forma de um *driver* de dispositivo. Desta maneira, considerando a necessidade de segurança em sistemas, remanesce a pergunta: quanta segurança foi

[26] Segundo alegação de um fabricante específico de *smart card* acessível por meio de impressão digital, o sistema embutido no cartão suporta níveis de precisão desde 1% FAR até 0.0001% FAR, e testes independentes mostraram que a precisão de EER fica em torno de 0.1% [Nordin 2004].

adicionada à aplicação, ou, em outras palavras, quão poderosa é a ferramenta descrita?

3.5. Problemas Abertos

Além da larga utilização em investigação criminal, as tecnologias biométricas estão sendo rapidamente sendo adotadas numa grande variedade de aplicações de segurança, como controle de acesso físico e lógico, comércio eletrônico, gestão digital de direitos autorais, segurança de prédios e residências e bloqueio de equipamentos. Em geral, essas aplicações requerem, dos subsistemas biométricos, alta precisão, alto desempenho e baixo custo.

Entretanto, embora tenha havido grandes avanços recentes, ainda é necessário um vigoroso esforço de pesquisa para resolver muitos problemas desafiadores. Um trabalho recente organiza os principais obstáculos à ampla disseminação de sistemas biométricos [Chandra and Calderon 2005]. Seis grandes classes abrangem cerca de 30 problemas atuais ainda não resolvidos concernentes a tecnologias biométricas. Companhias fabricantes e usuárias que planejam implementar a tecnologia de sistemas biométricos automatizados devem refletir sobre as principais questões que desfiam tais sistemas. Tais desafios necessitam de uma solução abrangente que satisfaça às legítimas preocupações dos usuários. Existe um campo aberto para pesquisas sobre o assunto, do qual elaboramos uma lista não exaustiva:

- Unicidade - Métodos e métricas para estimar a quantidade de informação contida nos diversos identificadores biométricos, o que está relacionado diretamente com a unicidade dos mesmos.
- Avaliação - Padrões, métodos e métricas para avaliação estatística da precisão e do desempenho dos diversos tipos de sistemas biométricos.
- Escala - Análise de diversas características específicas de sistemas de verificação e de identificação de larga escala.
- Cifragem - Técnicas eficientes de proteção dos perfis biométricos, dos dados que transitam entre os processos e técnicas de proteção de privacidade.
- Multibiometria - Fusão da informação em diversos níveis.
- Certificação - Proposta de entidade(s) certificadora(s) de sistemas biométricos. Natureza da entidade e escopo da certificação.
- Desenvolvimento de sensores, considerando aspectos desejáveis como baixo custo, detecção de vitalidade, portabilidade, entre outros.
- Processos - Melhoria ou criação de métodos ou algoritmos de aquisição, extração e comparação de características.
- Protocolos e arquiteturas - Análise dos protocolos e arquiteturas existentes e efetivação de novas propostas com foco no reforço de segurança e privacidade dos dados.

- Detecção de vitalidade - Equipamentos e métodos de detecção de vitalidade ou, até mesmo, de detecção de não-vitalidade.

- Revogação ou cancelamento - Criação ou melhoria de métodos e técnicas para prover a revogação das características biométricas.

- Novas tecnologias - Criação de novos identificadores biométricos.

- *Smart cards* - Conjugação de biometria e *smart cards*.

3.6. Conclusão

Este capítulo buscou apresentar uma visão geral sobre sistemas de autenticação biométrica. Os tipos de autenticação biométrica levam à diferenciação dos sistemas em sistemas de identificação (busca 1:N) e de verificação (busca 1:1), sendo que cada um destes tipos possui características específicas e aplicações mais adequadas. Existem numerosas características físicas e comportamentais do ser humano que podem ser usadas como identificadores biométricos. Dentre elas, os mais utilizados atualmente foram apresentados com um pouco mais de detalhe. Cenários de armazenamento de perfis foram levantados e, finalmente, questões de segurança foram abordadas.

Mostramos que não existe uma tecnologia "melhor", mas sim a tecnologia mais adequada perante cada aplicação. Mostramos ainda que a biometria possui grande utilidade. Para sistemas de identificação, a utilização de biometria já está bastante consolidada, sendo a impressão digital a tecnologia biométrica mais utilizada, embora haja espaço para outras tecnologias. Para sistemas de verificação, consideramos que, no estágio atual de desenvolvimento tecnológico, a utilização de biometria deve ser cuidadosamente analisada. No caso de haver risco para o usuário, a biometria deve ser utilizada como acessório.

Não é demais relembrar à exaustão que a biometria não é cem por cento precisa. Esta é uma característica que permite configurar um sistema para ser mais rigoroso ou mais permissivo, dependendo do limiar de comparação. Os pontos fortes das tecnologias biométricas em geral são: (1) a biometria é fortemente vinculada a uma identidade e (2) a biometria não precisa ser memorizada, nem pode ser esquecida ou emprestada. No entanto, estes pontos fortes levam também a fraquezas correspondentes, que são: (1) a biometria não é revogável e (2) a biometria não é segredo. Pesquisas têm sido levadas a cabo no sentido de eliminar ou amenizar os pontos fracos.

Uma mensagem final sobre a utilização de sistemas biométricos não pode deixar de lado a questão principal deste trabalho, que é o reforço de segurança. A segurança de sistemas biométricos se traduz na proteção da aplicação e é alcançada pela eliminação de vulnerabilidades nos pontos de ataque aos ativos da aplicação. A introdução de biometria em um sistema não deve criar novas vulnerabilidades e aberturas. Em outras palavras, a introdução de biometria para incrementar segurança deve ser convenientemente analisada e justificada. A autenticação biométrica deve ser um aspecto integrado da segurança da aplicação como um todo, o que inclui a identificação e prevenção de brechas de segurança do próprio sistema biométrico.

Até mesmo o reforço de segurança proporcionado por sistemas biométricos necessita ser cuidadosamente avaliado, devido ao efeito da *mudança do elo fraco*. Um sistema qualquer possui pontos de vulnerabilidades quanto à segurança. Os pontos mais vulneráveis são os "elos fracos". Ao reforçarmos a segurança em um elo mais fraco, outro ponto do sistema vai se tornar o elo mais fraco. Um exemplo particularmente alarmante é o do homem que teve a extremidade de um dedo amputada por ladrões para que estes pudessem roubar seu carro, protegido por um sistema biométrico [BBC 2005]. Neste caso, a contramedida causou uma mudança de tática do atacante, mudando o elo fraco para o próprio usuário.

Referências

[ANSI 2003] ANSI (2003). Biometric information management and security for the financial services industry. ANSI X9.84-2003, American National Standards Institute.

[ANSI 2005] ANSI (2005). *ANSI INCITS 409 - Information Technology - Biometric Performance Testing and Reporting - Part 1: Principles and Framework - Part 2: Technology Testing and Reporting - Part 3: Scenario Testing and Reporting*. American National Standards Institute.

[Bailly-Bailliére et al. 2003] Bailly-Bailliére, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Mariéthoz, J., Matas, J., Messer, K., Popovici, V., Porée, F., Ruiz, B., and Thiran, J.-P. (2003). The BANCA database and evaluation protocol. In *4th International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, volume 2688 of *Lecture Notes in Computer Science*, pages 625–638, Guildford, UK. Springer-Verlag.

[BBC 2005] BBC (2005). Malaysia car thieves steal finger. `http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm`. Acessado em julho/2006.

[Bergadano et al. 2002] Bergadano, F., Gunetti, D., and Picardi, C. (2002). User authentication through keystroke dynamics. *ACM Transactions on Information and System Security*, 5(4):367–397.

[BioID 2005] BioID (2005). Humanscan. `http://www.bioid.com`. Acessado em julho/2006.

[BIOLAB 2005] BIOLAB (2005). Synthetic FINgerprint GEnerator. Biometric Systems Lab - `http://bias.csr.unibo.it/research/biolab`. Acessado em julho/2006.

[BITE 2005] BITE (2005). Global biometric market and industry report. Technical report, Biometric Identification Technology Ethics. `http://www.biteproject.org/`.

[Bolle et al. 2004] Bolle, R. M., Connell, J. H., Pankanti, S., Ratha, N. K., and Senior, A. W. (2004). *Guide to Biometrics*. Springer Professional Computing, 1st edition.

[Bolle et al. 2002] Bolle, R. M., Connell, J. H., and Ratha, N. K. (2002). Biometric perils and patches. In *Pattern Recognition*, volume 35, pages 2727–2738. Elsevier Science.

[Buhan et al. 2006] Buhan, I., Bazen, A., Hartel, P., and Veldhuis, R. (2006). A false rejection oriented threat model for the design of biometric authentication systems. *Proceedings of the International Conference on Biometrics 2006 (Hong Kong, China)*, 3832:728–736.

[Burge and Burger 2000] Burge, M. and Burger, W. (2000). Ear biometrics in computer vision. In *International Conference on Pattern Recognition*, volume 2, pages 2822–2826, Los Alamitos, CA, USA. IEEE Computer Society.

[Campbell 1997] Campbell, J. P. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462.

[Campbell and Reynolds 1999] Campbell, J. P. and Reynolds, D. A. (1999). Corpora for the evaluation of speaker recognition systems. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2:829–832.

[Cappelli et al. 2006] Cappelli, R., Maio, D., Maltoni, D., Wayman, J. L., and Jain, A. K. (2006). Performance evaluation of fingerprint verification systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):3–18.

[Carlson 2003] Carlson, L. (2003). Match on card system for IT security. In *Biometric Technology Today*, volume 11, pages 3–4. Elsevier Science.

[Chandra and Calderon 2005] Chandra, A. and Calderon, T. (2005). Challenges and constraints to the diffusion of biometrics in information systems. *Communications of the ACM*, 48(12):101–106.

[Chang et al. 2003] Chang, K., Bowyer, K., and Flynn, P. (2003). Multimodal 2D and 3D biometrics for face recognition. *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 187–194.

[Chen and Jain 2005] Chen, H. and Jain, A. K. (2005). Dental biometrics: Alignment and matching of dental radiographs. *IEEE Tansactions on Pattern Analysis and Machine Intelligence*, 27(8):1319–1326.

[Clarke 1994] Clarke, R. (1994). Human identification in information systems: management challenges and public policy issues. *Information Technology & People*, 7(4):6–37.

[Connie et al. 2005] Connie, T., Teoh, A., Goh, M., and Ngo, D. (2005). Palmhashing: a novel approach for cancelable biometrics. *Information Processing Letters*, 93(1):1–5.

[Daugman 1999] Daugman, J. (1999). Recognizing persons by their iris patterns. In Jain, A. K., Bolle, R. M., and Pankanti, S., editors, *Biometrics: Personal Identification in Networked Society*, chapter 5. Kluwer Academic Publishers, Boston, MA, USA.

[Daugman 1993] Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161.

[Daugman and Williams 1996] Daugman, J. G. and Williams, G. O. (1996). A proposed standard for biometric decidability. In *Proceedings of CardTech/SecureTech*, pages 223–234, Atlanta, GA, USA.

[DIN 2003] DIN (2003). Information Technology - security techniques - a framework for security evaluation and testing of biometric technology. ISO/IEC JTC 1/SC 27 N 3806, Deutsches Institut fur Normung, Berlin, Germany.

[Ernst 2002] Ernst, J. (2002). Iris recognition: Counterfeit and countermeasures. `http://www.iris-recognition.org/counterfeit.htm`. Acessado em julho/2006.

[Fink et al. 2001] Fink, G. A., Wienecke, M., and Sagerer, G. (2001). Video-based online handwriting recognition. In *International Conference on Document Analysis and Recognition*, pages 226–230, Los Alamitos, CA, USA. IEEE Computer Society.

[Heinen and Osório 2004] Heinen, M. R. and Osório, F. S. (2004). Biometria comportamental: Pesquisa e desenvolvimento de um sistema de autenticação de usuários utilizando assinaturas manuscritas. *Infocomp Revista de Ciência da Computação*. ISSN 1807-4545 volume 3 fasciculo 2 pgs 31 a 37 Lavras MG Brasil.

[Hill 1999] Hill, R. B. (1999). Retina identification. In Jain, A. K., Bolle, R. M., and Pankanti, S., editors, *Biometrics: Personal Identification in Networked Society*, chapter 6. Kluwer Academic Publishers, Boston, MA, USA.

[Hook et al. 2003] Hook, C., Kempf, J., and Scharfenberg, G. (2003). New pen device for biometrical 3d pressure analysis of handwritten characters, words and signatures. In *WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications*, pages 38–44, New York, NY, USA. ACM Press.

[IBG 2005] IBG (2005). Independent testing of iris recognition technology. Technical Report NBCHC030114/0002, International Biometric Group.

[Jain et al. 2004] Jain, A. K., Ross, A., and Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Tecnhology*, 14(1):4–20.

[Kazienko 2003] Kazienko, J. F. (2003). Assinatura digital de documentos eletrônicos através da impressão digital. Dissertação de mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina.

[Kong et al. 2002] Kong, A., Griffith, A., Rhude, D., Bacon, G., and Shahs, G. (2002). Department of Defense federal biometric system protection profile for medium robustness environments. Technical report, U.S. Department of Defense.

[Kong et al. 2003] Kong, A., Griffith, A., Rhude, D., Bacon, G., and Shahs, G. (2003). US Government biometric verification mode protection profile for medium robustness environments. Technical report, The Biometrics Management Office and National Security Agency.

[Korotkaya 2003] Korotkaya, Z. (2003). Biometric person authentication: Odor. Inner report in Department of Information Technology, Laboratory of Applied Mathematics, Lappeenranta University of Technology. in "Advanced Topics in Information Processing: Biometric Person Authentication".

[Kyong I. Chang and Flynn 2005] Kyong I. Chang, K. W. B. and Flynn, P. J. (2005). An evaluation of multimodal 2D+3D face biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4).

[Landwehr 2001] Landwehr, C. E. (2001). Computer security. *International Journal of Information Security*, 1(1):3–13.

[Leniski et al. 2003] Leniski, A. C., Skinner, R. C., McGann, S. F., and Elliott, S. J. (2003). Securing the biometric model. In *IEEE* 37^{th} *Annual 2003 International Carnahan Conference on Security Technology*, pages 444–449.

[Lu et al. 2003] Lu, G., Zhang, D., and Wang, K. (2003). Palmprint recognition using eigenpalms features. *Pattern Recognition Letters*, 24(9-10):1463–1467.

[Maltoni et al. 2003] Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. (2003). *Handbook of Fingerprint Recognition*. Springer Verlag, New York, USA.

[Mansfield and Wayman 2002] Mansfield, A. and Wayman, J. (2002). Best practices in testing and reporting performance of biometric devices, version 2.0.1. Technical report, Biometrics Working Group, `http://www.afb.org.uk/bwg/bestprac.html`.

[Mansfield et al. 2001] Mansfield, T., Kelly, G., Chandler, D., and Kane, J. (2001). Biometric product testing final report. Technical Report CESG contract X92A/4009309, UK Biometrics Working Group.

[Mansfield et al. 2002] Mansfield, T., Kelly, G., Chandler, D., and Kane, J. (2002). Biometrics for identification and authentication - advice on product selection. Technical report, UK Biometrics Working Group.

[Masek and Kovesi 2003] Masek, L. and Kovesi, P. (2003). MATLAB source code for a biometric identification system based on iris patterns. Master's thesis, The School of Computer Science and Software Engineering, The University of Western Australia. Código-fonte disponível em `http://www.csse.uwa.edu.au/~pk/studentprojects/libor/sourcecode.html`. Acessado em julho/2006.

[Miller 1994] Miller, B. (1994). Vital signs of identity. *IEEE Spectrum*, 31(2):22–30.

[Munich and Perona 1998] Munich, M. E. and Perona, P. (1998). Camera-based ID verification by signatures tracking. *Lecture Notes in Computer Science*, 1406:782.

[Myers and Rabiner 1981] Myers, C. S. and Rabiner, L. R. (1981). A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409.

[Negin et al. 2000] Negin, M., Chmielewski(Jr.), T. A., Salganicoff, M., Camus, T. A., von Seelen, U. M. C., Venetianer, P. L., and Zhang, G. G. (2000). An iris biometric system for public and personal use. *IEEE Computer Society*, 33(2):70–75.

[NIST 2001] NIST (2001). CBEFF - Common Biometric Exchange File Format. Technical Report NISTIR 6529, National Institute of Standards and Technology, USA.

[NIST 2003] NIST (2003). NIST year 2003 speaker recognition evaluation plan. Technical report, NIST Speech Group. `http://www.nist.gov/speech/tests/spk/2003/doc/2003-spkrec-evalplan-v2.2%.pdf`.

[NIST 2005] NIST (2005). NIST special database 4 - NIST 8-bit gray scale images of fingerprint image groups (FIGS). `http://www.nist.gov/srd/nistsd4.htm`. Acessado em julho/2006.

[Nordin 2004] Nordin, B. (2004). *Match-on-Card Technology*. Precise Biometrics Inc., /urlhttp://www.precisebiometrics.com. Acessado em julho/2006.

[OASIS 2003] OASIS (2003). XCBF - XML Common Biometric Format. Technical report, Organization for the Advancement of Structured Information Standards. `http://www.oasis-open.org/committees/xcbf/`.

[Osborne and Ratha 2003] Osborne, M. and Ratha, N. K. (2003). A JC-BioAPI compliant smart card with biometrics for secure access control. *Lecture Notes in Computer Science*, 2688:903–910.

[Patrick 1972] Patrick, E. A. (1972). *Fundamentals of Pattern Recognition*. Prentice-Hall Inc.

[Phillips et al. 2000] Phillips, P., Martin, A., Wilson, C., and Przybocki, M. (2000). An introduction to evaluating biometric systems. *IEEE Computer*, 33(2):56–63.

[Phillips et al. 2002] Phillips, P. J., Sarkar, S., Robledo, I., Grother, P., and Bowyer, K. (2002). The gait identification challenge problem: Data sets and baseline algorithm. In *International Conference on Pattern Recognition*, volume 01, pages 385–388, Los Alamitos, CA, USA. IEEE Computer Society.

[Prokoski and Riedel 1999] Prokoski, F. J. and Riedel, R. (1999). Infrared identification of faces and body parts. In Jain, A. K., Bolle, R. M., and Pankanti, S., editors, *Biometrics: Personal Identification in Networked Society*, chapter 9. Kluwer Academic Publishers, Boston, MA, USA.

[Przybocki and Martin 2004] Przybocki, M. and Martin, A. (2004). NIST speaker recognition evaluation chronicles. Technical report, Speech Group, Information Access Division, Information Technology Laboratory National Institute of Standards and Technology, USA. Published in the Odissey 2004 Conference.

[Putte and Keuning 2000] Putte, T. and Keuning, J. (2000). Biometrical fingerprint recognition: don't get your fingers burned. In *Proceedings of IFIP TC8/WG8.8 Fourth Working Conference on Smart Card Research and Advanced Applications*, pages 289–303.

[Rabiner and Juang 1986] Rabiner, L. R. and Juang, B. H. (1986). An introduction to Hidden Markov Models. *IEEE Magazine on Accoustics, Speech and Signal Processing*, 3(1):4–16.

[Reynolds et al. 2000] Reynolds, D. A., Doddington, G. R., Przybocki, M. A., and Martin, A. F. (2000). The NIST speaker recognition evaluation - overview methodology, systems, results, perspective. *Speech Communications*, 31(2-3):225–254.

[Ross et al. 2006] Ross, A. A., Nandakumar, K., and Jain, A. K. (2006). *Handbook of Multibiometrics*. International Series on Biometrics. Springer.

[Sabourin et al. 1997] Sabourin, R., Genest, G., and Preteux, F. J. (1997). Off-line signature verification by local granulometric size distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):976–988.

[Sanchez-Reillo et al. 2000] Sanchez-Reillo, R., Sanchez-Avila, C., and Gonzalez-Marcos, A. (2000). Biometric identification through hand geometry measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1168–1171.

[Sandstrom 2004] Sandstrom, M. (2004). Liveness detection in fingerprint recognition systems. Linkoping University, Department of Electrical Engineering, Eletronic Press, Student Thesis.

[Scheenstra et al. 2005] Scheenstra, A., Ruifrok, A., and Veltkamp, R. C. (2005). A survey of 3D face recognition methods. In *5^{th} International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA)*, volume 3546 of *Lecture Notes in Computer Science*, pages 891–899, Rye Brook, NY, USA. Springer-Verlag.

[Schneier 1999] Schneier, B. (1999). Inside risks: the uses and abuses of biometrics. *Communications of the ACM*, 42(8):136.

[Thorpe et al. 2005] Thorpe, J., van Oorschot, P., and Somayaji, A. (2005). Passthoughts: Authenticating with our minds. *Proceedings of the New Security Paradigms Workshop*.

[Turk and Pentland 1991] Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Maui, HI, USA.

[Uludag and Jain 2004] Uludag, U. and Jain, A. K. (2004). Attacks on biometric systems: A case study in fingerprints. *Proc. SPIE-EI*.

[Valid 2005] Valid (2005). Visual audio lip-motion identification. `http://www.validbiometrics.com`. Acessado em julho/2006.

[Victor et al. 2002] Victor, B., Bowyer, K., and Sarkar, S. (2002). An evaluation of face and ear biometrics. In *International Conference on Pattern Recognition*, volume 1, pages 429–432, Quebec City, Canada. IEEE Computer Society.

[Wayman 1997] Wayman, J. L. (1997). A scientific approach to evaluation biometric systems using mathematical methodology. In *Proceedings of CardTech/SecureTech*, Orlando, FL, EUA.

[Wayman 1999a] Wayman, J. L. (1999a). Error rate equations for the general biometric system. *IEEE Robotics & Automation Magazine*, 6(1):35–48.

[Wayman 1999b] Wayman, J. L. (1999b). National biometric test center collected works. Technical report, National Biometric Test Center, San Jose, California, USA.

[Yeung et al. 2004] Yeung, D.-Y., Chang, H., Xiong, Y., George, S., Kashi, R., Matsumoto, T., and Rigoll, G. (2004). SVC2004: First international signature verification competition. In *1st International Conference on Biometric Authentication (ICBA)*, volume 3072 of *Lecture Notes in Computer Science*, pages 16–22, Hong Kong, China. Springer-Verlag.

[Yu et al. 1995] Yu, K., Mason, J., and Oglesby, J. (1995). Speaker recognition using Hidden Markov Models, Dynamic Time Warping and Vector Quantisation. *IEE Proceedings – Vision, Image and Signal Processing*, 142:313–318.

[Zhang and Shu 1999] Zhang, D. and Shu, W. (1999). Two novel characteristic in palmprint verification: Datum point invariance and line feature matching. *Pattern Recognition*, 32(4):691–702.

[Zhao et al. 2003] Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458.

[Zunkel 1999] Zunkel, R. L. (1999). Hand geometry based verification. In Jain, A. K., Bolle, R. M., and Pankanti, S., editors, *Biometrics: Personal Identification in Networked Society*, chapter 4, pages 87–101. Kluwer Academic Publishers, Boston, MA, USA.

Capítulo

4

A Nova Geração de Modelos de Controle de Acesso em Sistemas Computacionais

Luiz Otávio Botelho Lento[1], Joni da Silva Fraga[2] e Lau Cheuk Lung[3]

1 Depto. Automação de Sistemas, Universidade Federal de Santa Catarina,– otavio@das.ufsc.br
2 Depto. Automação de Sistemas, Universidade Federal de Santa Catarina - fraga@das.ufsc.br
3 PPGIA, Pontifícia Universidade Católica do Paraná - lau@ppgia.pucpr.br

Abstract

The access control is a security service and is used to manage the access to the resources (ex data, process and devices) from a computer system, making the actions or operations of a valid user may execute limited. In the last few years, the access control has been improving itself, your representation and functions. This short course has the objective to present a few models of access control, such as DAC, MAC, DRM, RBAC and UCON, approaching the meaning characteristics and proprieties, and also a brief comparison between then.

Resumo

O controle de acesso é um serviço de segurança, e tem como função gerenciar o acesso aos recursos (dados, processos, dispositivos, etc.) de um sistema computacional, limitando as ações ou operações que um usuário válido possa executar. Nos últimos anos, o controle de acesso vem evoluindo quanto a sua representação e funcionalidades. Este mini-curso visa apresentar alguns modelos de controle de acesso, como DAC, MAC, DRM, RBAC e UCON, abordando as suas principais características e propriedades, como também um breve comparativo entre eles.

4.1. Introdução

O controle de acesso que teve no modelo Matriz de Acesso a sua principal expressão formal quando introduzido em 1971 por Butler W. Lampson [Lampson 1971], tem sido mantido como definitivo na descrição de políticas simples. Grande parte das políticas discricionárias são descritas na simplicidade deste modelo formando os controles conhecidos na literatura como controles discricionários (DAC – *Discretionary Access Controls*). Alguns autores identificam ainda estes controles como baseados em identidades [Karp, A. H. (2006)] (IBAC – *Identity-Based Access Control*) devido ao fato que os controles que implementam as políticas, dependem fortemente da autenticação dos sujeitos que solicitam os acessos controlados.

Os mesmos modelos que descrevem políticas obrigatórias (Modelos MAC – *Mandatory Access Control*) incorporam a simplicidade do Matriz de Acesso nas verificações em um mesmo nível de segurança. Modelos como o Bell-LaPadula [Bell e LaPadula 1976], que trata da confidencialidade, e o Biba, que se fundamenta na verificação da integridade dos acessos, são exemplos deste grupo de modelos. Estes modelos são também identificados como baseados em regras (*Rule-Based Access Model*) porque envolvem a concretização de políticas globais em um sistema ou organização. As regras de acesso definem a perda do caráter essencialmente discricionário dos criadores dos objetos acessados, e possibilitam a verificação de propriedades mais globais, que devem valer no sistema como um todo.

Nos últimos anos, o controle de acesso vem experimentando uma evolução acentuada com novas formas de representação e também pela necessidade de adequação a novas aplicações e tecnologias. Um exemplo, destes novos modelos é o RBAC (*Role-Based Access Control* - controle de acesso baseado em papel) [Sandhu 1997] que simplifica a gerência de direitos por não mais concentrar sob identidades os mesmos, mas sim em papéis ou funções. O controle de acesso motivado pelos direitos de propriedade e o controle de uso também provocaram o aparecimento de modelos como o DRM (*Digital Right Management*) [Ku e Chi 2004]. Recentemente, Sandhu e Park propuseram o UCON (*Usage Control Model*) [Sandhu. e Park 2004] como expressão máxima de modelos e políticas que pode englobar todos os modelos citados, tradicionais ou não. Este modelo geral é uma ferramenta poderosa tanto para a formalização como para a implementação de todos os tipos de controles em um sistema.

Este mini-curso tem como objetivo central apresentar os modelos citados em suas principais características e propriedades e verificar suas utilidades na formalização e implementação dos diferentes controles de acesso identificados na literatura. Este documento está dividido em uma parte que trata com modelos mais convencionais, onde são apresentados modelos como DAC, MAC, DRM e RBAC. Uma boa parte deste texto explora ainda o modelo de controle de acesso UCON. Uma outra parte que trata alguns modelos de controle de acesso desenvolvido por pesquisadores. Por fim, o texto termina com uma análise comparativa entre estes modelos descritos onde procuramos evidenciar a importância de cada modelo na expressão dos controles de sistemas e aplicações.

4.2. Aspectos Básicos sobre Segurança em Sistemas Computacionais

Na seqüência foram colocadas algumas definições envolvendo Segurança, Política e Modelos de Segurança que usamos no texto. Não são objetos do mini-curso políticas e controles externos que visam à proteção dos equipamentos e sistemas computacionais. No final desta seção caracterizamos ainda controle de acesso.

4.2.1. Conceitos

Segurança

A segurança em sistemas computacionais não é formada exclusivamente por meios que visam proteger informações ou recursos computacionais, mas é, antes de tudo, uma disciplina que através de seus conceitos, metodologias e técnicas, tenta manter propriedades de um sistema, evitando ações danosas no mesmo. Na literatura, existem várias definições para Segurança (*security*) e, em quase todas, a mesma é caracterizada como a qualidade de serviço que visa manter no sistema um conjunto de propriedades [Landwehr 2001, Denning 1982]:

- A **Confidencialidade** garante a revelação da informação só a sujeitos autorizados. [1]
- A **Integridade** assegura a não modificação indevida – seja acidental ou intencionalmente – das informações e recursos no sistema.
- A **Disponibilidade** garante que as informações e recursos num sistema computacional estarão desimpedidos e prontos para serem usados quando requisitados por sujeitos autorizados.

Alguns autores ainda juntam às citadas, as propriedades de Autenticidade e de Não Repúdio [Landwehr 2001]. A legitimidade de informações e de principais é explicitada pela propriedade de autenticidade. O não repúdio garante, em protocolos e transações, as proteções contra comportamentos omissos ou maliciosos onde participantes neguem ações realizadas.

As Violações de segurança em sistemas computacionais correspondem a burlar de alguma forma a segurança de um sistema computacional de modo a não se verificarem uma ou mais propriedades de segurança. A Tabela 4.1 ilustra os tipos de violação em contraposição às propriedades de segurança não verificadas.

Tipo de Violação		Propriedade de Segurança Violada
1	Revelação Não Autorizada	Confidencialidade
2	Modificação Não Autorizada	Integridade
3	Negação de Serviço	Disponibilidade

Tabela 4.1. Tabela de relação de referências.

As violações de segurança são decorrências de **vulnerabilidades** (*vulnerability*), **ameaças** (*threat*) e **ataques** (*attack*) em sistemas computacionais. Entende-se por vulnerabilidades, as fraquezas ou imperfeições em procedimentos, serviços ou sistemas,

[1] Entende-se por sujeito uma entidade ativa como um humano, sistema ou máquina.

oriundas de falhas de concepção, implementação ou de configuração dos mesmos. Uma ameaça é a caracterização de um possível conjunto de ações que explore as vulnerabilidades e o conhecimento sobre um sistema que possa por em risco as propriedades de segurança. Uma ameaça, quando concretizada na execução de suas ações, é identificada como um ataque à segurança do sistema.

Políticas de Segurança

O termo *política de segurança* pode ter significados diferentes dependendo do nível em que se aplica. Em ambientes computacionais, política de segurança é entendida normalmente como um conjunto de regras que especificam como um sistema provê os seus serviços, mantendo as propriedades de confidencialidade, integridade e de disponibilidade [Lendweir2001]. Os sistemas computacionais fazem então uso de regras através de *controles*, estabelecendo os limites de operação dos usuários no sistema e protegendo seus dados e recursos da ação de intrusos[2]. Uma política sempre se aplica a um sistema específico, e não a uma classe geral de sistemas.

As políticas de segurança são classificadas em duas categorias: as discricionárias e obrigatórias. Nas discricionárias os acessos a cada recurso ou informação são manipulados livremente pelo proprietário ou responsável pelo mesmo, segundo a sua vontade (à sua discrição). Já nas obrigatórias (não discricionárias) as autorizações de acesso são definidas através de um conjunto incontornável de regras que expressam algum tipo de organização envolvendo a segurança das informações no sistema como um todo [Mackenzie 1997]. Neste texto, serão ainda tratadas as políticas discricionárias e não discricionárias.

Modelos de Segurança

Os modelos de segurança correspondem a descrições formais do comportamento de um sistema atuando segundo regras de uma política de segurança. Estes modelos são representados na forma de um conjunto de entidades e relacionamentos [Goguen 1982]. A definição de políticas de segurança é normalmente orientada por modelos de segurança, que fornecem na representação abstrata o funcionamento seguro do uso no sistema alvo de um conjunto de regras de segurança.

Os modelos se apresentam, na literatura, divididos em três tipos básicos [Sandhu, e Samarati 1996]:

- Controles baseados em identidade ou discricionários (*Discretionary Access Control*: DAC): por expressarem as políticas discricionárias, baseiam-se na idéia de que o proprietário do recurso deve determinar quem tem acesso ao mesmo.

[2] As regras definidas pelas políticas de segurança determinam as entidades autorizadas e responsáveis pelas ações executadas sobre informações mantidas no sistema, normalmente identificadas como **principal** (sujeitos autorizados). O nível de aplicação desta política pode caracterizar um principal como um usuário, um processo ou ainda uma máquina em uma rede de computadores. A entidade (usuário, processo ou máquina) que ganha acesso a recursos de um sistema computacional violando a política de segurança é normalmente denominada de **intruso**.

- Controles baseados em regras gerais ou obrigatórios (*Mandatory Access Control*: MAC): baseiam-se em uma administração centralizada de segurança, na qual são ditadas regras incontornáveis de acesso à informação. A forma mais usual de controle de acesso obrigatório é o controle de acesso baseado em reticulados (*lattice-based access control*), que confina a transferência de informação a uma direção em um reticulado de rótulos de segurança (vide seção 4.3.2).
- Controles baseados em papéis (*role*). (*Role-Based Access Control*—RBAC): requer que permissões de acesso sejam atribuídas a papéis e não a usuários, como no DAC; os usuários obtêm estes direitos através de papéis alocados a si.

Mecanismos de Segurança

Os mecanismos de segurança são responsáveis pela concretização das políticas de segurança nos sistemas computacionais. Estas políticas, cujos comportamentos são expressos através de modelos de segurança, são implantadas por mecanismos que exercem os controles necessários para manter as propriedades de segurança. Os controles executados internamente em sistemas computacionais que gerenciam os acessos a recursos são identificados como **Controles de Acesso**. Controles usados na proteção das informações que são disponíveis através de dispositivos de entrada e saída (memórias secundárias, suportes de comunicação, etc.), envolvem o que é normalmente identificado como **Controles Criptográficos**. Outros controles ainda podem ser identificados em sistemas computacionais. Estes controles são chamados de Serviços de Autenticação, importantes na identificação de principais (sujeitos autorizados), e Controles de Inferência, que normalmente envolvem as semânticas das aplicações. O estudo aqui apresentado está centrado em Controle de Acesso.

4.2.2. Controle de Acesso

O controle de acesso limita as ações ou operações que um sujeito de um sistema computacional pode executar, restringindo o que ele pode fazer diretamente, como também os programas que podem ser executados em seu nome. A Figura 4.1 apresenta o esquema básico do controle de acesso exercido através de mecanismos em um sistema computacional.

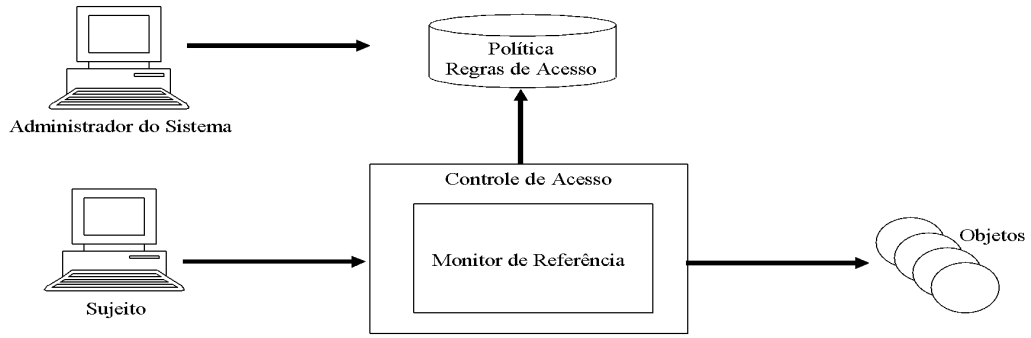


Figura 4.1. Controle de acesso

Na efetivação do controle de acesso são usados mecanismos que tomam o nome de **Monitor de Referências** [Anderson 1972], e que atuam em vários níveis de um sistema. As referências a segmentos de memória são validadas nas camadas inferiores do sistema, através do *hardware*. O sistema operacional, por sua vez, através do seu serviço de arquivos valida os acessos a arquivos no sistema. O monitor de referência é o mediador de toda a tentativa de acesso de sujeitos aos objetos do sistema, consultando as regras da política para verificar se as solicitações de acesso são permitidas. As regras são mantidas pelo administrador de segurança (ou de sistema), tendo como base uma política de segurança.

O monitor de referência como responsável na intermediação de todas as requisições de acesso a objetos de um sistema resultou na definição de **núcleo de segurança** [Landwehr 1983]. Este núcleo que envolve um conjunto de mecanismos de hardware e software, permitindo a concretização da noção de um monitor de referências, deve possuir algumas propriedades: ser inviolável, incontornável (sempre ativado nas requisições de acesso) e pequeno o suficiente para permitir a verificação de sua correção. A idéia de núcleo de segurança deu origem mais tarde as ***TCBs***[3] (*Trusted Computing Base*) introduzidas pelos critérios de avaliação do DoD [DOD85] (também conhecidos como *Orange Book*).

Os aspectos envolvendo a implementação destes monitores de referência (ou na sua versão de núcleo de segurança) não são muito evidentes. Por exemplo, na definição das permissões de acesso, as abordagens existentes tentam atingir soluções comuns que atendam a seu grupo de principais e recursos controlados. O grau de refinamento e a simplicidade de gerenciamento são metas a serem também consideradas. Todavia, a abordagem usada, quando da definição das permissões, certamente não conseguirá levar em consideração todos os aspectos existentes, e prestigiará sempre um determinado grupo dos recursos controlados.

Os sistemas de controle de acesso podem ser diferenciados via suas políticas e seus mecanismos de acesso. As políticas de acesso são direcionamentos de alto nível, baseadas nas necessidades dos proprietários dos recursos ou ainda das organizações. A partir destas definições de alto nível devem ser geradas permissões que determinam como os acessos serão controlados em todos os níveis do sistema.

4.3. Modelos de Segurança

A pesquisa na área de modelos de segurança computacional começou no início da década de 70. Ao longo desses anos, inúmeros modelos foram propostos, com as mais variadas premissas e os mais diversos objetivos. Esta seção apresenta detalhes de modelos de segurança considerados clássicos: Matriz de Acesso, Bell-LaPadula, Biba e RBAC. Na seqüência destes, são apresentadas as novas proposições de modelos presentes na literatura.

[3] Uma *Trusted Computing Base* também deve ter as mesmas propriedades de um núcleo, consistindo na concentração da totalidade dos mecanismos de segurança de sistema computacional (incluindo hardware, software). A combinação destes mecanismos é responsável em cumprir a política de segurança.

4.3.1. Matriz de Acesso Segurança

O modelo de segurança utilizado em boa parte dos sistemas atuais é o chamado controle de acesso discricionário (DAC), que delega aos usuários a tarefa de proteger seus recursos no sistema. Neste modelo, cada usuário é quem determina quais os direitos de acesso que outros usuários ou aplicações do sistema possuem sobre as informações que são de sua responsabilidade. Modelos discricionários são considerados inadequados para diversas aplicações, uma vez que são relativamente fracos e demasiadamente flexíveis: basta um equívoco por parte de um usuário inocente (ou um ato deliberado de um usuário malicioso) e informações importantes podem ser indevidamente reveladas, alteradas ou destruídas.

O modelo de **Matriz de Acesso** [Sandhu e Samarati 1994] é o modelo conceitual subjacente ao controle de acesso discricionário. Neste modelo, o estado de segurança do sistema é representado pela tripla (S, O, A), onde: S é o conjunto de sujeitos s_i que podem exercer privilégios; O é um conjunto de objetos o_j nos quais os privilégios ou direitos podem ser exercidos; e A é a matriz de acesso onde linhas correspondem aos sujeitos em S e as colunas aos objetos em O. Uma célula A_{ij} da matriz representa os direitos de acesso do sujeito s_i sobre o objeto o_j. É importante ressaltar que sujeitos podem ser também objetos. Por exemplo, um dos acessos representados na matriz pode ser o envio de um sinal a um processo; neste caso, os sujeitos correspondentes a processos deveriam ser incluídos também nas colunas da matriz.

A Figura 4.2 apresenta um exemplo da matriz de acesso com três objetos (três arquivos) e três sujeitos (usuários Waldir, Nanda e Luiz). Os direitos sobre arquivos são os usuais: dono (D), leitura (L), escrita (W) e execução (E). A entrada *A[Waldir, Arquivo1]* representa os privilégios D, L, W e E de Waldir sobre o Arquivo 1.

	Arquivo 1	**Arquivo 2**	**Arquivo 3**	**Arquivo 4**
Waldir	D W, L, E		D W, L, E	L
Nanda	E	D W, L	E	
Luiz	E	L		D W, L

Figura 4.2. Matriz de Acesso

No controle de acesso discricionário, a concessão e a revogação dos direitos de acesso a um objeto são feitas pelo usuário que é dono desse objeto (à sua *discrição*). Isso fornece ao usuário uma grande flexibilidade na proteção de seus objetos, o que é uma vantagem deste modelo de controle de acesso. Entretanto, o controle de acesso discricionário não permite controlar a disseminação da informação. No exemplo da figura 4.3, Nanda permite que Luiz leia o seu arquivo 2, mas proíbe que estas informações sejam lidas por Valdir. Entretanto, não há nada no modelo que possa impedir que Luiz aja maliciosamente, copiando as informações do arquivo 2 para 0 arquivo 4, o que possibilitaria que Valdir lesse tais informações mesmo contra a vontade da usuária Nanda.

A matriz de acesso proposta por Lampson [Lampson 1971] é um modelo relativamente informal. O modelo de matriz de acesso é definido em termos de estados de segurança do sistema. A matriz corresponde ao estado atual de segurança do sistema. As mudanças de estado de segurança do sistema são realizadas usando **regras de transição do modelo** [Landwehr 1981]. Estas correspondem a permissões para mudar o objeto "matriz" e são tipicamente **retirar sujeito/objeto**, **criar sujeito/objeto**, **transferir direitos** e **suprimir direitos.** A aplicação destas regras determina situações de acessos especiais:

> Um sujeito pode acessar um objeto porque possui o direito no estado de segurança atual do sistema.
>
> Ou, um sujeito pode acessar um objeto porque pode obter o direito necessário através de mudanças de estado da matriz.

Um estado não autorizado ou estado de fuga é aquele onde um direito pode ser obtido por um sujeito não autorizado. Portanto um sujeito não autorizado a aceder um objeto pelo seu proprietário pode por mudanças de estado da matriz vir a obter o direito de acesso necessário ao objeto. Este aspecto de possíveis evoluções em tempo de execução da matriz e, por conseqüência, do estado de segurança do sistema, determinaram o aparecimento de várias extensões formais e gráficas do modelo matriz de acesso para verificar este comportamento dinâmico do modelo. Exemplos destas extensões são os modelos formais *HRU* [Harrison 1976] e *Take-Grant* [Snyder 1981] que fundamentados no modelo de matriz de acesso, estudam a disseminação de direitos devido a evolução dinâmica do estado de segurança de um sistema.

A implementação da Matriz de Acesso na forma original em grandes sistemas torna-se inviável devido ao seu tamanho, e a grande quantidade de células em branco. Existem duas abordagens tradicionais para a implementação do modelo Matrizes de acesso: as listas de controle de acesso (*ACLs*) e listas de competências (*lists of capabilities*).

Listas de controle de acesso (*Access Control Lists*: *ACLs*)

Esta é, talvez, a abordagem mais popular de implementação da matriz de acesso. Cada objeto é associados uma ACL que indica os sujeitos no sistema com acessos autorizados ao objeto considerado. Uma ACL corresponde no armazenamento da matriz por colunas. A Figura 4.3 apresenta um exemplo de ACL.

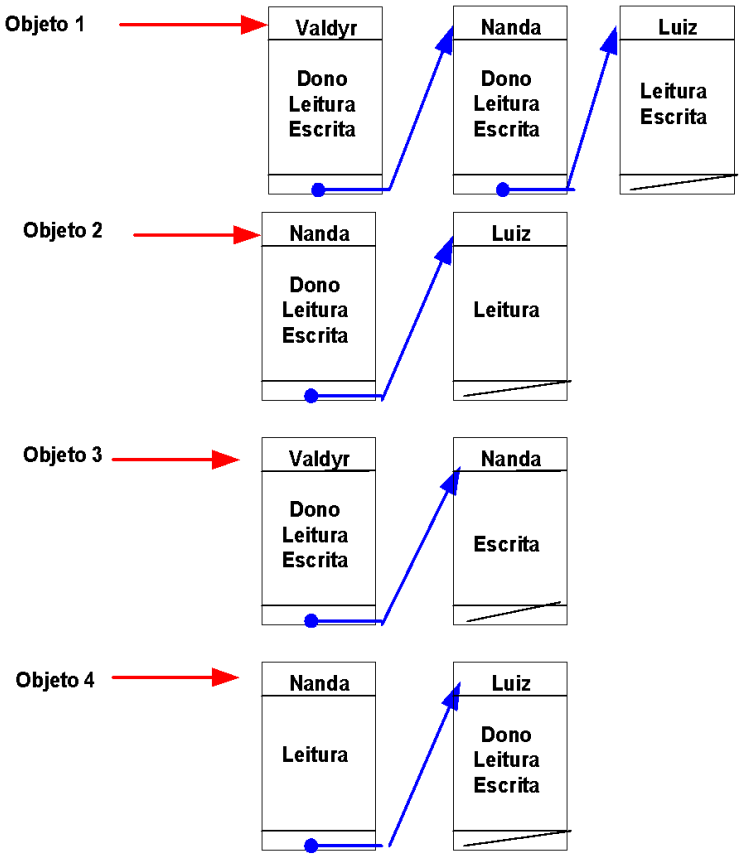


Figura 4.3. ACL

A *ACL* facilita determinar quais os modos de acesso que os sujeitos estão autorizados em um objeto, provendo uma forma fácil de rever ou revogar os modos de acessos aos objetos. Porém, é difícil determinar todos os acessos que um sujeito possui, porque seria necessário verificar a ACL de cada objeto. As listas de acesso são mecanismos normalmente usados em níveis altos de um sistema (em nível de usuário).

Listas de Competências (*Capabilities*)

Nesta abordagem, a cada sujeito está associado uma **lista de competências** (*capability list*) que indica, para cada objeto no sistema, quais as permissões de acessos o sujeito possui (armazenamento da matriz de acesso por linhas). Listas de competências permitem fácil verificação e revogação dos acessos autorizados para um determinado sujeito. As vantagens e desvantagens de ACLs e *capabilities* são, como as próprias estratégias, ortogonais entre si.

Uma *capability* corresponde a um identificador protegido (imutável) que identifica o objeto e especifica os direitos de acesso a serem atribuídos ao sujeito possuidor da mesma. Duas propriedades são fundamentais no mecanismo de *capability*:

O *capability* pode ser passada de um sujeito a outro; e

> Nenhum sujeito possuidor de uma *capability* (identificador) pode alterá-la ou construir novas sem uma negociação prévia com TCB (Trusted Computing Base) do sistema.

Capabilities são vantajosas em sistemas distribuídos. A posse de uma *capability* é suficiente para que um sujeito obtenha o acesso autorizado por esta *capability*. Em um sistema distribuído, isso possibilita que um sujeito se autentique uma vez, obtenha a sua lista de *capabilities* (ou privilégios) e apresente as mesmas quando necessário para obter os acessos desejados; os servidores precisam apenas verificar a validade da *capability* apresentada para liberar o acesso desejado [Sandhu 1994].

A Figura 4.4 apresenta um exemplo de uma lista de ca*pabilities*.

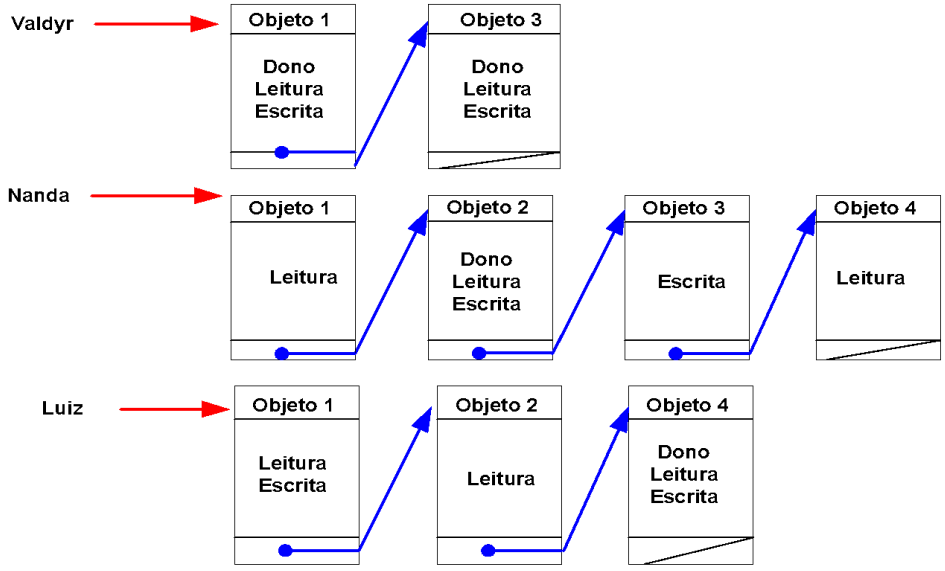


Figura 4.4. Lista de Competências

4.3.2. Modelos Não Discricionários (*Mandatory Access Control* - MAC)

As pesquisas que levaram aos modelos MAC na década de 70, foram financiadas pelo Departamento de Defesa (DoD) dos Estados Unidos. Desta forma, estes modelos iniciais foram baseados em práticas de segurança utilizadas em áreas ligadas à segurança nacional. Em que pese esta origem, os modelos e seus conceitos subjacentes são perfeitamente aplicáveis a ambientes não-militares.

Os controles definidos pelos modelos MAC seguem políticas que valem para todo o sistema e, portanto, que definem regras e estruturas aplicáveis no âmbito de todo o sistema. Estas políticas, normalmente, especificando envolvem algum tipo de classificação multinível (*multilevel policy*) de informação. Os acessos dos sujeitos aos objetos são submetidos a um tipo de controle baseado na classificação usada.

Um dos conceitos importantes nestas classificações usadas em MACs é o de **níveis de sensibilidade**[4]. Como há custos associados à proteção da informação e nem todas as informações são igualmente importantes (ou **sensíveis**), definem-se diferentes níveis de sensibilidade, ordenados segundo uma hierarquia. Os níveis mais usuais são, em ordem crescente de "sensibilidade": NÃO-CLASSIFICADO, CONFIDENCIAL, SECRETO e ULTRA-SECRETO. De maneira similar, uma universidade poderia adotar os níveis ALUNO, FUNCIONÁRIO e PROFESSOR — os níveis devem refletir a necessidade de proteção da informação (dificilmente um ambiente acadêmico classificaria as informações da mesma maneira que um ambiente militar).

Entretanto, a simples associação de níveis de sensibilidade à informação não atende a um princípio clássico de segurança conhecido como *need-to-know*. Este princípio diz que o controle da disseminação da informação está diretamente ligado à quantidade de pessoas que têm acesso a essa informação; desta forma, quanto menos pessoas conhecerem um segredo, mais fácil será garantir que o segredo não será revelado. Para que isso seja viabilizado, são definidas **categorias** (*category*), ou compartimentos de segurança, que correspondem a diferentes projetos ou setores confinando suas informações. Assim, por exemplo, professores do Departamento de Física provavelmente não devem ter acesso a informações classificadas com o nível PROFESSOR pertencentes ao Departamento de Geografia. Os indivíduos podem ter acesso a diferentes categorias na medida em que as suas incumbências demandem este acesso.

Para transpor estes conceitos ao contexto computacional, são definidos **rótulos de segurança** (*security labels*) que agregam os níveis de sensitividade e categorias. Os rótulos de segurança associados as informações de um sistema correspondem ao produto vetorial entre o conjunto de níveis de sensitividade e o conjunto representado por *Category* que definem todos os compartimentos destas informações (*Security label* = *Sensitivity level* x *Category*).

Em sistemas que fazem uso deste mecanismo, todas as entidades recebem um rótulo de segurança; o rótulo de um objeto define a sua **classificação**, e o rótulo de segurança de um sujeito é chamado de **autorização** ou **habilitação** (*clearance*) do sujeito.

Uma das formas de viabilizar a implementação de políticas é construir **reticulados** (*lattice*) com **rótulos de segurança** (*security labels*). Estes reticulados são construídos a partir de uma relação de ordem parcial sobre o conjunto R de rótulos do sistema. Esta relação que permite comparações entre os *security labels* de sujeitos e objetos nestes modelos de políticas multi-nível, é conhecida como relação de dominância e é definida como:

> **Relação de Dominância**: um *Security Level* de um objeto o_1 domina outro de um objeto o_2, se as seguintes condições são verificadas:
>
> $$Sensitivity\_Level(o_1) \geq Sensitivity\_Level(o_2) \wedge Category(o_1) \supseteq Category(o_2)$$

É importante notar que, em conjuntos ordenados parcialmente, existem elementos que são ditos não-comparáveis, e que também, é verificado sempre possui um ínfimo (limite

[4] A tradução mais apropriada para *Sensitivity level* é "nível de sensitividade". Porém acreditamos que "nível de sensibilidade" descreve melhor em Português o aspecto semântico por traz destas classificações e o manteremos neste texto.

inferior) e um supremo (limite superior) segundo a relação de dominância nos reticulados de rótulos de segurança [Landwehr 1981]

Entre os modelos MACs, que fazem uso destas classificações que fazem uso de rótulos de segurança está o modelo Bell-La Padula (BLP) de David Bell e Leonard LaPadula [Bell e LaPadula. 1976] que é descrito a seguir.

O Modelo Bell-LaPadula

Dois cientistas da MITRE Corporation, David Bell e Leonard LaPadula, desenvolveram um modelo baseado nos procedimentos usuais de manipulação de informação em áreas ligadas à segurança nacional americana. Este modelo ficou conhecido como **modelo Bell-LaPadula**, ou modelo BLP [Bell 1976]. Existem diversas outras descrições do modelo Bell-LaPadula disponíveis na literatura, como [Amoroso 1994, Landwehr 1981, Sandhu 1993], algumas delas apresentando pequenas variações em relação ao modelo original. O modelo Bell-LaPadula trata exclusivamente com a confidencialidade das informações no sistema.

Na apresentação original do modelo [Bell 1976], um sistema é descrito através de uma máquina de estados finitos. As transições de estados no sistema obedecem a determinadas regras. Bell e LaPadula demonstram indutivamente que a segurança do sistema é mantida se ele parte de um estado seguro e as únicas transições de estado permitidas são as que conduzem o modelo a um outro estado seguro.

No modelo um sistema é descrito em termos de sujeitos que acessam objetos, onde cada sujeito possui uma habilitação e cada objeto possui uma classificação. A cada sujeito está associado também um **rótulo corrente de segurança**, que representa a classificação mais alta dentre as informações já consultadas pelo sujeito no sistema até um determinado instante, sendo, portanto, uma classificação flutuante (dinâmica). A habilitação de um sujeito deve sempre dominar o seu rótulo corrente de segurança.

A **propriedade de segurança simples**, também conhecida como propriedade-ss ou regra *no read up* (NRU)[5], diz que um sujeito só pode observar informações para as quais esteja habilitado; em outras palavras, a **leitura** de um sujeito s_i sobre um objeto o_j é autorizada se, e somente se, *rótulo*(s_i) deve dominar *rótulo*(o_j). Por exemplo, uma informação classificada como SECRETO só pode ser lida por sujeitos com habilitação SECRETO ou ULTRA-SECRETO.

A propriedade-ss não é suficiente para garantir a segurança desejada do sistema: ela não evita que um sujeito malicioso coloque informações privilegiadas em um recipiente com classificação inferior à das informações, o que constitui claramente um fluxo não-autorizado de informação. Assim, torna-se necessário adicionar outra propriedade a ser satisfeita pelo sistema.

[5] *No read up* vem do fato de um sujeito não poder ler objetos localizados acima dele no reticulado de rótulos de segurança.

A **propriedade-*** (propriedade estrela), também chamada de regra *no write down*[6] (NWD), é satisfeita se, quando um sujeito tem simultaneamente um acesso de leitura sobre um objeto o_1 e um acesso de escrita sobre um objeto o_2. Sendo assim, o *rótulo*(o_1) deve dominar o *rótulo*(o_2), isto é, o acesso do sujeito s_i sobre um objeto o_j é autorizado se:

rótulo(o_j) domina o *rótulo-corrente*(s_i) quando o acesso for de escrita;

rótulo(o_j) é dominado pelo *rótulo-corrente*(s_i) quando o acesso for de leitura.

Por exemplo, se um sujeito está lendo um objeto SECRETO, ele só pode alterar um objeto SECRETO ou ULTRA-SECRETO simultaneamente.

Existem duas observações importantes a se fazer respeito da propriedade-*.

1. Ela não se aplica a sujeitos de confiança - um sujeito de confiança é aquele em quem se confia a não transferir informação de modo a quebrar a segurança, mesmo que esta transferência seja possível;
2. Vale a pena lembrar que a propriedade-ss e a propriedade-* devem ser ambas satisfeitas; nenhuma delas garante, por si só, a segurança desejada.

Dinâmica do Modelo Bell-LaPadula

O rótulo corrente de segurança de um sujeito é conceituado como uma classificação flutuante, e define a propriedade-* em termos do rótulo corrente de segurança de um sujeito, sem explicitar como este rótulo efetivamente flutua dentro do sistema. Esta flutuação está ligada ao comportamento dinâmico do modelo BLP, isto é, o rótulo corrente de segurança de um sujeito evolui durante a evolução do próprio sistema.[7]

Quando um usuário entra no sistema, ele recebe um rótulo corrente de segurança que seja dominado pela sua habilitação. Este rótulo pode ser escolhido pelo usuário ou atribuído automaticamente pelo sistema; a abordagem adotada não interfere no comportamento dinâmico. Os sujeitos criados em nome de um usuário herdam tanto a habilitação como o rótulo corrente de segurança do usuário. Os acessos destes sujeitos aos objetos do sistema devem observar a propriedade-ss e a propriedade-*.

Bell e LaPadula [Bell 1976] fornecem um conjunto de regras para a operação de um sistema seguro.[8] Uma destas regras dita que o rótulo corrente de segurança de um sujeito só é modificado mediante uma requisição explícita deste sujeito; isto significa que o rótulo corrente de segurança não flutua de maneira automática no sistema, e, também, que esta flutuação ocorre por iniciativa do próprio sujeito. A regra especifica

[6] Assim chamada porque impede que um sujeito escreva em objetos localizados abaixo dele no reticulado de rótulos de segurança.

[7] O **princípio da tranqüilidade** estabelece que nenhuma operação pode alterar a classificação de objetos ativos no sistema [Landwehr 1981]. Entretanto, implementações baseadas no modelo BLP tipicamente lançam mão de sujeitos de confiança para a reclassificação de objetos.

[8] Evidentemente, a noção de sistema seguro, no contexto do modelo BLP, corresponde a um sistema a salvo de ameaças de revelação não-autorizada.

também que a alteração do rótulo corrente de segurança só é autorizada se ela não violar a propriedade-*.

Por exemplo, seja a seguinte situação: um sujeito s_i, com rótulo corrente NÃO CLASSIFICADO e habilitação (estática) SECRETO, deseja ler um objeto o_1, que é CONFIDENCIAL. A propriedade-ss permite que s_i leia o_1, pois *rótulo*(s_i) domina *rótulo*(o_1). Entretanto, essa operação não satisfaz a propriedade-*, pois *rótulo*(o_1) domina *rótulo-corrente*(s_i). Logo, s_i precisa solicitar a atualização de seu rótulo corrente de segurança para (pelo menos) CONFIDENCIAL. Entretanto, se s_i, ao solicitar a atualização de seu rótulo corrente para CONFIDENCIAL possuir um acesso de escrita para o_2, onde o_2 é igualmente NÃO-CLASSIFICADO, ele deve ter esta solicitação negada pelo sistema, uma vez que a sua aceitação violaria a propriedade-*.

A regra que governa a atualização do rótulo corrente de segurança não impõe qualquer restrição além da satisfação da propriedade-* e da condição de que o rótulo corrente seja dominado pela habilitação do sujeito.

Limitações do Modelo Bell-LaPadula

A adoção do modelo *BLP* pode acarretar problemas se o sistema tiver que lidar também com ameaças de integridade. Quando um sujeito escreve em um objeto com uma classificação superior à sua habilitação (o que satisfaz a propriedade-*), ele não pode observar os efeitos desta operação de escrita (o que violaria a propriedade-ss); por esse motivo, tal operação é chamada de **escrita cega** [Amoroso 1994, Sandhu 1993].

O cenário de escritas cegas torna-se uma preocupação na medida em que o mesmo sujeito considerado inadequado para ver o conteúdo de um objeto possui permissão para fazer modificações arbitrárias neste mesmo objeto. Isto pode causar problemas de integridade que só podem ser resolvidos através de alterações nas regras do modelo BLP. Por exemplo, escritas em objetos com níveis mais altos de segurança podem ser proibidas; um sujeito só poderia escrever em um objeto que tivesse o mesmo nível de segurança. Entretanto, tal modificação restringe, de certa forma, o modelo BLP e muda o seu enfoque, que deixa de ser exclusivamente a ameaça de revelação não-autorizada e passa a ser uma combinação de revelação e integridade. Por outro lado, a adoção da propriedade-* revisada é bastante comum em implementações de sistemas computacionais que seguem o modelo BLP.

O modelo Bell-LaPadula inclui a noção de **sujeitos de confiança** (*trusted subjects*) [Bell 76, Landwehr 1981]. Um sujeito de confiança é aquele em quem se confia a não quebrar a segurança mesmo que alguns dos seus acessos atuais violem a propriedade-*. Neste caso, a propriedade-* só se aplica aos demais sujeitos do sistema. Por exemplo, o conceito de sujeitos de confiança pode ser usado para qualificar os processos relacionados com a manutenção do sistema, pois se o administrador do sistema tiver que obedecer estritamente às regras do modelo BLP ele dificilmente conseguirá realizar qualquer tarefa significativa de administração. Outra classe de processos que faz uso da noção de sujeitos de confiança é a dos subsistemas mais críticos do sistema operacional, como gerência de memória e *drivers* de dispositivos [Amoroso 1994].

Um dos principais problemas do modelo Bell-LaPadula reside no aspecto extremamente restritivo da propriedade-*.[9] Por exemplo, se um sujeito com rótulo corrente de segurança SECRETO deseja copiar um arquivo CONFIDENCIAL, a propriedade-* impõe que a cópia tenha classificação SECRETO, mesmo que as informações ali contidas possuam classificação CONFIDENCIAL. Ao longo do tempo, isso faz com que as informações subam no reticulado de rótulos de segurança, recebendo classificações sucessivamente maiores. Este fenômeno é conhecido como **superclassificação da informação** [Landwehr 1981]. A superclassificação da informação provoca a necessidade de reclassificações periódicas dos objetos (através de sujeitos de confiança) apenas para garantir a usabilidade de sistemas baseados no modelo BLP.

Modelo Biba

O modelo Bell-LaPadula tem por objetivo conter ameaças de revelação não-autorizada; não obstante, os próprios criadores do modelo BLP discutem como ele poderia ser adaptado para conter ameaças de integridade [Bell 1976]. Embora as idéias de Bell e LaPadula careçam de maior consistência, elas serviram de base para que Ken Biba desenvolvesse um modelo de segurança com o propósito de garantir a integridade da informação; conhecido como **modelo de integridade Biba** [Biba 1977].

O modelo *Biba* é definido como o dual do *BLP*. Suas regras são similares ao do modelo anterior, mas tem como objetivo a preservação da integridade das informações classificadas, evitando alterações não autorizadas. O modelo define níveis hierárquicos de integridade para os sujeitos (s_i's) e para os objetos (o_j's) similares aos níveis de sensitividade definidos no *BLP*. A **propriedade simples de integridade** define que um sujeito só pode ler um objeto se o seu nível de integridade for dominado pelo do objeto. A **propriedade estrela de integridade** especifica que um sujeito pode ter direito de escrita sobre um objeto, se e somente se o seu nível se sensibilidade for dominado pelo do objeto.

Por ser o dual do modelo *BLP*, este modelo apresenta limitações similares às descritas no modelo anterior. No modelo *Biba* ocorre uma degradação do nível de integridade, de maneira análoga a superclassificação da informação do modelo de *BLP*. Existe também há a necessidade de *sujeitos de confiança* no modelo Biba, utilizados para alterar a integridade de sujeitos e objetos, mantendo o sistema viável.

Existem alguns outros modelos mandatórios além do *BellLapadula* e do *Biba* citados em literatura. O modelo *Clark-Wilson* (*CW*) é um exemplo, baseia-se na idéia que a integridade é mais importante que a confidencialidade [Clark. e Wilson. 1987] em operações comerciais. Porém, diferente dos modelos *Bell-LaPadula* e *Biba*, o *CW* assume **transações bem-formadas** (todos os passos de uma seqüência de atividades são executados corretamente) e a **separação de tarefas** (cada sujeito desempenha um papel

[9] Segundo Landwehr [30], a provisão de sujeitos de confiança é um reconhecimento de que a propriedade-* impõerestrições de acesso mais rigorosas do que aquelas usadas extracomputacionalmente em ambientes de segurança militar,uma vez que o seu propósito é evitar que programas malcomportados causem vazamentos de informação.

distinto na seqüência de atividades que formam uma transação) como essência de sua definição.

4.3.3. Modelos Baseados em Papéis (RBAC: Role Basic Access Control)

Os modelos baseados em papéis regulam o acesso dos usuários à informação com base nas atividades que os usuários executam no sistema. Estes modelos necessitam a identificação de **papéis** no sistema, onde um papel pode ser definido como um conjunto de atividades e responsabilidades associadas a um determinado cargo ou função. Logo, ao invés de especificar um conjunto de acessos autorizados para cada usuário do sistema, as permissões são conferidas aos papéis. Por conseguinte, um usuário que exerce um papel pode realizar todos os acessos para os quais o papel está autorizado.

Os modelos baseados em papéis possuem diversas características importantes, tais como: [Sandhu e Samarati (1994)]:

> **Gerência de autorizações mais simples** - a especificação de autorizações é dividida em duas partes, associação de direitos de acesso a papéis e associação de papéis a usuários. Isso simplifica bastante a gerência da segurança, facilitando tarefas como ajustar os direitos de acesso de um usuário em função de uma promoção ou transferência de setor na organização.
>
> **Suporte a hierarquias de papéis** - em muitas aplicações existe uma hierarquia natural de papéis baseada nas noções de generalização e especialização. Isto permite que permissões sejam herdadas e compartilhadas através da hierarquia.
>
> **Suporte a privilégio mínimo** - os papéis permitem que um usuário trabalhe com o mínimo privilégio exigido para uma determinada tarefa. Usuários autorizados a exercer papéis poderosos só precisam exercê-los quando forem absolutamente necessários, minimizando a possibilidade de danos por causa de erros inadvertidos.
>
> **Suporte a separação de tarefas** - os modelos baseados em papéis suportam separação de tarefas. Nestes modelos, a separação de tarefas é obtida através de restrições à autorização e/ou à ativação de papéis considerados mutuamente exclusivos.
>
> **Delegação da administração de segurança** - modelos baseados em papéis permitem que a administração da segurança seja descentralizada de maneira controlada. Isto significa que o administrador de segurança pode delegar parte de suas atribuições de acordo com a estrutura organizacional ou com a arquitetura do sistema computacional, permitindo, por exemplo, que administradores regionais gerenciem a segurança dos subsistemas locais.

O modelo RBAC é **independente de política**, diferente do que acontece com os modelos tradicionais de controle de acesso.[10] A independência da política possibilita uma grande flexibilidade e facilidade do ajuste do controle de acesso à medida em que ocorram mudanças no ambiente. Apesar da independência de política, o RBAC garante três princípios de segurança: o princípio de privilégio mínimo, separação de tarefas

[10] Tanto o controle obrigatório quanto o discricionário impõem uma política de segurança. No MAC, fluxos de informação contrários a um determinado sentido no reticulado de rótulos de segurança são proibidos; no DAC, a política imposta é que o dono do objeto é quem determina os seus direitos de acesso.

(restrito aos papéis) e abstração de dados (não há restrições quanto à natureza das permissões, podendo ser abstratas, tais como débito e crédito em um objeto conta).

O Modelo RBAC-NIST[11] [Sandhu e Park 2004]

O modelo RBAC-NIST reflete a compreensão e a modelagem do RBAC por parte de dois grupos de pesquisa: o grupo do NIST e o grupo liderado por Ravi Sandhu, da George Mason University. O modelo NIST-RBAC é um excelente componente para uma padronização do conhecimento na área de controle de acesso baseado em papéis.

Sendo o RBAC um conceito bastante amplo e aberto, bem como complexo de ser representado, a utilização de um modelo único para tratá-lo torna-se um tanto quanto restritiva e complexa. Uma abordagem mais realista seria a definição de uma família de modelos, que a partir de um modelo básico que contempla as características fundamentais do RBAC, modelos adicionais podem ser criados com mais funcionalidades e requisitos em relação ao básico.

O modelo RBAC-NIST é definido por quatro modelos:

Modelo RBAC Básico (Core)

O modelo RBAC básico define um conjunto de elementos e relações para ativar o sistema RBAC completamente. Isto inclui as relações usuário-papel e permissão-papel. O RBAC básico também introduz o conceito de ativação do papel como parte da sessão do usuário dentro do sistema computacional. Ele é necessário em qualquer sistema RBAC, mas os outros componentes são independentes entre si e podem ser implementados separadamente

Este modelo inclui os conjuntos de 5 elementos básicos de dados chamado usuários (pessoas, hosts, etc), papéis, objetos operações e permissões, e também as relações entre eles. A Figura 4.5 (RBAC básico) apresenta estes elementos junto com as suas relações.

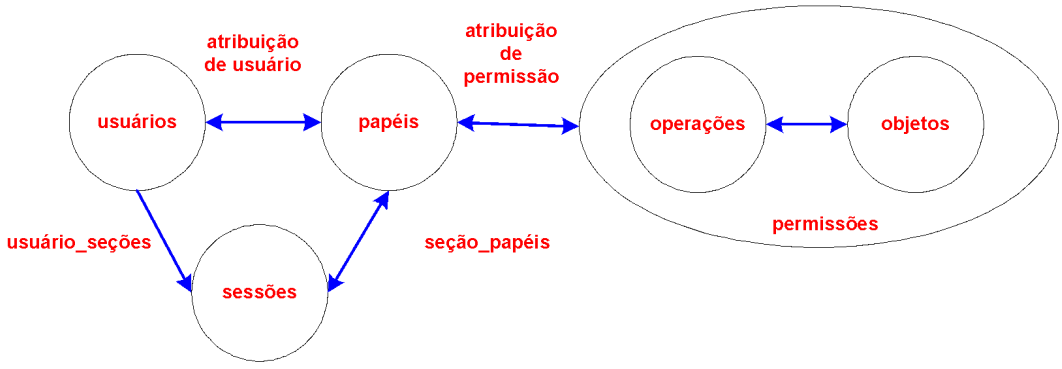


Figura 4.5. RBAC Básico (Core).

[11] National Institute of Standards and Technology

O modelo RBAC como um todo é definido basicamente pelos seus usuários relacionados a papéis (autoridade ou responsabilidade atribuída a um usuário) e as permissões (aprovação para executar uma operação em um ou mais objetos protegidos) relacionadas a estes papéis. Assim, um papel admite que sejam estabelecidos relacionamentos de muitos para muitos entre usuários e permissões. O modelo RBAC básico inclui um conjunto de seções onde cada uma delas é o mapeamento entre um usuário e o subconjunto de papéis a ele relacionado.

A Figura 4.5 mostra as relações da atribuição do usuário e a atribuição da permissão, onde as setas representam os relacionamentos de muitos para muitos (um usuário pode estar relacionado a um ou mais papéis e um papel pode estar relacionada a um ou mais usuários). Esta estrutura possibilita uma grande flexibilidade e possibilidades de atribuições de permissões a papéis e usuários, evitando que o usuário possa ter acesso a recursos desnecessários (o controle de acesso é limitado ao tipo de acesso que pode estar associado a usuários e recursos).

Cada sessão é um mapeamento de um usuário para alguns dos possíveis papéis que ele pode assumir durante um determinado período de tempo. Cada sessão está associada a um único usuário e cada usuário associado a uma ou mais sessões.

Modelo RBAC Hierárquico

O modelo RBAC hierárquico adiciona as relações (Figura 4.6), ao modelo básico, que suportam a hierarquia de papéis. As hierarquias são meios naturais de estruturar os papéis, representando os aspectos de autoridade e responsabilidade dentro de uma organização.A hierarquia é matematicamente uma ordem parcial definindo a relação de superioridade entre papéis, pelo qual os papéis superiores adquirem as permissões dos seus papéis subordinados e os papéis subordinados adquirem usuários dos seus papéis superiores.

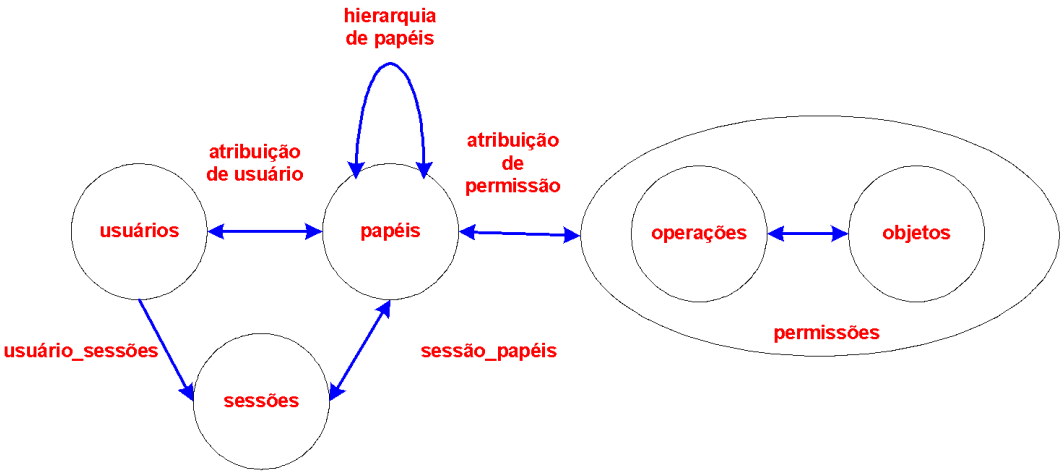


Figura 4.6. RBAC hierárquico.

Apesar das hierarquias arbitrárias estejam mais próximas de uma realidade, a larga utilização de hierarquias limitadas levou a uma subdivisão do RBAC Hierárquico:

- **RBAC Hierárquico Geral** - uma hierarquia de papéis pode constituir qualquer tipo de ordem parcial existente. suporte para uma determinada ordem parcial arbitrária que serve como uma hierarquia de papéis, para incluir o conceito de múltiplas heranças de permissões e usuários entre os papéis existentes
- **RBAC Hierárquico Limitado** - quando existe qualquer restrição em relação à estrutura da hierarquia de papéis. Na maioria das vezes, as hierarquias estão limitadas a estruturas simples como árvores ou árvores invertidas.

A hierarquia de papéis define a relação de herança entre os papéis. A herança é descrita em termos de permissões, de forma que: r1 herda o papel de r2 se todos os privilégios de r2 são também privilégios de r1.

O padrão NIST reconhece tanto as hierarquias geral e limitada, como apresentado anteriormente. A Figura 4.7 apresenta um exemplo de uma estrutura hierárquica de papéis. Pode-se observar que os usuários no topo da árvore além de possuírem as suas permissões, eles herdam as permissões dos usuários que estão abaixo dele (o diretor possui as suas permissões mais as do chefe de Depto. e Encarregado de Divisão).

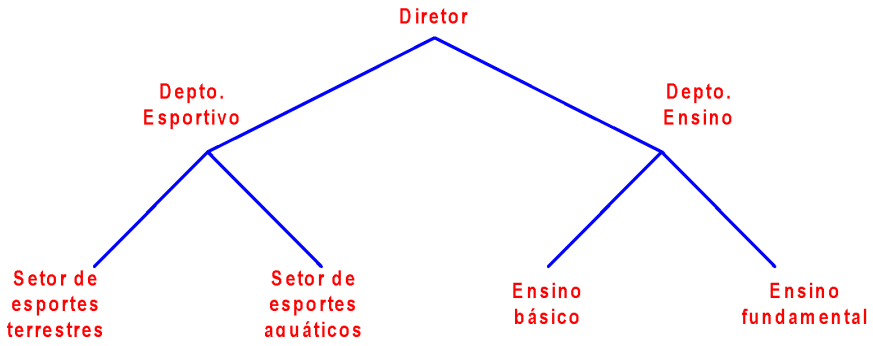


Figura 4.7. Estrutura hierárquica de papéis.

Modelo RBAC com Restrições

A separação de tarefas é utilizada para aplicar políticas de conflito de interesses, de forma que estas políticas previnam que usuários excedam a sua autoridade em suas posições de trabalho.

O princípio de separação de tarefas tem como objetivo garantir que as falhas por omissão e delegação de poderes dentro de uma organização possam ser causados somente como resultado da conivência entre indivíduos. Para minimizar a possibilidade destas conivências, indivíduos de diferentes habilidades ou conhecimento profissional ou interesses são atribuídos a diferentes tarefas necessárias no desempenho da função no negócio da Instituição. A motivação é garantir que a fraude e maiores erros não venham ocorrer sem conivência deliberada de vários usuários. Dois modelos são apresentados a seguir:

Separação Estática de Tarefas (SET) – Static Separation Duty (SSD)

O modelo de separação estática de tarefas adiciona relações exclusivas entre os papéis com respeito às atribuições do usuário (Figura 4.8). Isto significa que ele aplica restrições de associações de usuários a funções, de forma que quando um usuário está associado a um papel, ele não poderá assumir outro. Por exemplo, se um usuário exerce o papel de comparador, ele também não pode exercer o papel de executor de pagamentos (confeccionar os cheques). Este tipo de procedimento evita possíveis fraudes, tornando estes papéis mutuamente exclusivos. Normalmente, as restrições estáticas são colocadas em operações administrativas que possuem um potencial para questionar as políticas de separação de tarefas nos altos escalões administrativos.

Os modelos RBAC definem relações de separação estática de serviços com respeito a restrições nas associações usuário-papel (ex: um usuário só pode estar associado a um único papel por vez). Esta definição é bastante restritiva em dois aspectos importantes: o tamanho do conjunto de papéis e a combinação de papéis no conjunto para que a atribuição do usuário é restrita. Assim, o modelo define a separação estática de tarefas em dois argumentos: o conjunto de papéis e a maior cardinalidade que indica uma violação da separação estática de tarefas (ex: uma organização prescreve que nenhum usuário do setor de compras pode estar associado a três papéis dos quatro existentes). Porém, deve-se ter cuidado que a herança de usuários não questione as políticas de separação de tarefas.

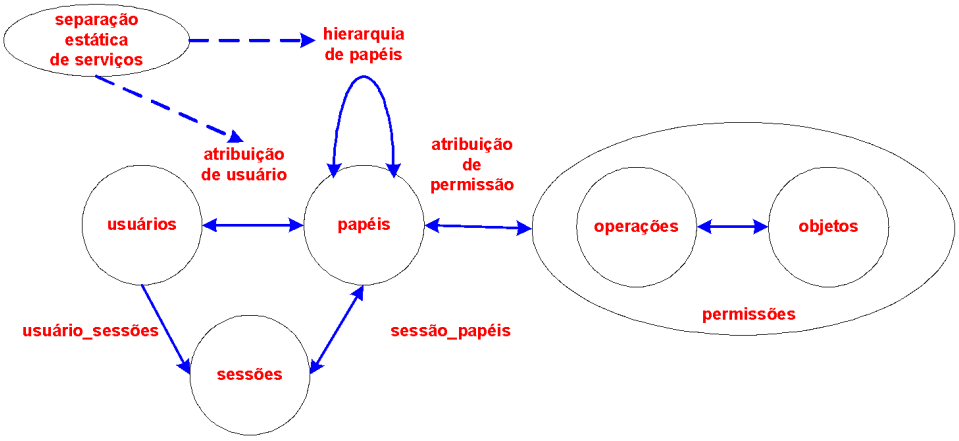


Figura 4.8. Separação estática de tarefas.

Separação Dinâmica de Tarefas (SDT) – Dynamic Separation Duty (DSD)

A separação dinâmica de tarefas define relações exclusivas com respeito a papéis que são ativados como parte da sessão do usuário (Figura 4.9).

A separação dinâmica de tarefas coloca as restrições somente nos papéis que podem ser ativados dentro ou via as sessões dos usuários, enquanto a estática coloca as restrições em todo o espaço de permissões do usuário.

A separação dinâmica de tarefas possui a capacidade de encaminhar as questões de conflito de interesses no momento que o usuário é associado a um papel. Esta política admite que um usuário seja autorizado a exercer dois ou mais papéis que não criem

conflitos de interesse quando atuando de forma independente, mas deve atender aos interesses da política quando ativados simultaneamente. Por exemplo: um usuário pode ser autorizado para exercer as funções de caixa e de supervisor de caixa, onde o supervisor é utilizado para reconhecer as correções do dinheiro da gaveta aberta do caixa. Caso a pessoa esteja exercendo somente o papel de caixa, ao mudar para o papel de supervisor de caixa, esta deverá inicialmente fechar o seu caixa antes de assumir o novo papel (supervisor de caixa).

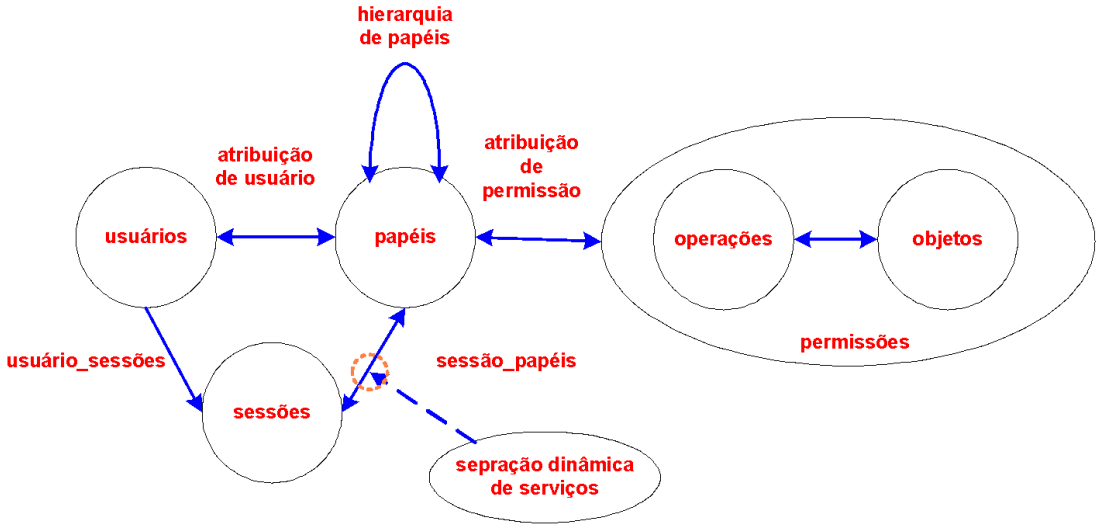


Figura 4.9. A separação dinâmica de tarefas.

Sendo assim, viu-se que o RBAC é um modelo com características diferenciadas do DAC e MAC, que faz uso do papel como aspecto principal de sua abordagem, e que possibilita uma fácil adaptação de novos requisitos em políticas de controle de acesso centralizadas de forma bastante flexível.

4.4 Digital Rights Mangement - DRM

Os direitos digitais são hoje um alvo de diversas atividades como: permissão para reprodução, transferência ou empréstimo de arquivos digitais, impressão, uso, extração e edição de informações disponibilizadas de forma digital, bem como a inserção e a obtenção de cópias de publicações digitais. No ambiente virtual, a possibilidade de gerenciamento e o controle e segurança na proteção do bem é um dos maiores problemas para os seus produtores. Os dispositivos tecnológicos buscam o controle e objetivam minimizar a disseminação ou a distribuição não autorizadas destes produtos, garantindo os direitos autorais de seus donos (ex: copiar o trabalho, emitir cópias do trabalho para o público, executar apresentações em público). [Kaminsky 2004]

A gerência dos direitos autorais (DRM - Digital Right Management) busca minimizar estes problemas, apresentando soluções compatíveis com as solicitações dos proprietários de bens, e busca via recursos tecnológicos disponíveis, com base numa política de controle de acesso a softwares, músicas, filmes ou outros dados digitais

restringir o uso destes dispositivos atendendo os interesses de direitos de cópia de seus proprietários. [Kaminsky 2004]

A necessidade de se utilizar o DRM está ligada às diferenças existentes entre o mundo digital e não digital levantando questões como: [Duncan, Barker, Douglas, Morrey e Waelde 2004 e Bechtold 2001]:

- O pronto acesso aos recursos na Internet cria dificuldades em estabelecer a fonte destes, enquanto é mais fácil incluir uma declaração de direitos de cópia em uma mídia de papel;
- A tranqüilidade com que os recursos digitais podem ser modificados encoraja a modificação sem verificar se esta é permitida (em mídias como papel esta questão está limitada a cópias com alterações de pequenos pedaços); e
- Qualquer pessoa pode publicar material na Internet sem as devidas permissões. O monitoramento destes aspectos é complicado, porque a maioria das pessoas não é capaz de definir legalmente a licença de publicação (em mídias como papel este controle é mais efetivo).

4.4.1 Aspectos Gerais do funcionamento de um Sistema DRM

Um típico sistema DRM leva em consideração basicamente três componentes na sua configuração [Ku e Chi 2004]:

- Proprietário do conteúdo – normalmente possui todos os direitos do conteúdo;
- Gerente – manipula todas as transações em nome do proprietário do conteúdo, trata as questões da licença especificando exatamente as permissões para um usuário fazer uso do conteúdo; e
- Usuário – neste caso, refere-se ao hardware ou software confiável, servindo de proxy para o consumidor do produto. Este hardware ou software é considerado confiável porque não admite que consumidores não autorizados acessem o conteúdo.

A Figura 4.10 apresenta uma visão geral de um típico sistema DRM.

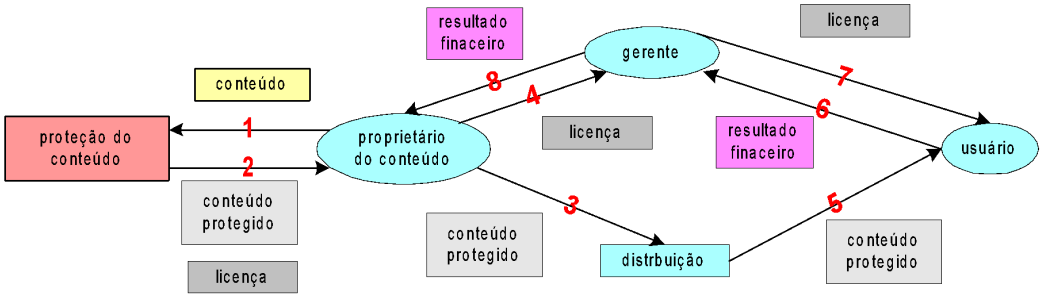


Figura 4.10. Sistema DRM.

1 – o proprietário entra com o conteúdo para ser protegido. Em algumas situações o conteúdo pode ser codificado em algum determinado formato. Por exemplo, o sistema DRM da Microsoft requer o formato Window Media (.wma ou .wmv). O proprietário do conteúdo pode desejar inserir uma maca d'água digital no conteúdo com o propósito de identificação. O sistema DRM pode então criptografar (na maioria das vezes fazendo

uso de técnicas de criptografia proprietária) e empacotar para a distribuição. O proprietário do conteúdo necessita especificar, utilizando uma Linguagem de Expressão de Direitos (REL), todos os direitos ou regras de uso que se aplicam ao conteúdo. Pode ser necessário que as regras sejam divididas em conjuntos, no qual cada conjunto está ligado a um determinado contexto, apesar de ser um mesmo conteúdo.

2 – o sistema DRM retorna o conteúdo protegido e a licença (ou um conjunto de licenças). A licença contém todos os direitos aplicáveis, termos e condições para o uso do conteúdo. Ele também contém a chave que é necessária para decriptografar o conteúdo protegido.

3 – o proprietário do conteúdo dissemina o conteúdo protegido via vários canais de distribuição, como a Internet, CDs, DVDs, email, P2P filesharing, entre outros. Os dois últimos meios de distribuição formam o conceito de superdistribuição. Este conceito se refere a possibilidade dos usuários redistribuir o conteúdo protegido de forma livre sem nenhuma restrição.

4 – o proprietário do conteúdo envia a licença/licenças para o gerente. O gerente é uma entidade confiável que pode manipular todas as solicitações de transações para acessar o conteúdo. Ele libera alguns recursos e possibilita que o proprietário do conteúdo se concentre no desenvolvimento do conteúdo, como também pode realimentar o perfil de consumo do usuário.

5 – o usuário recupera o conteúdo protegido do canal de distribuição. Ele examina o seu meta-data para identificar a licença necessária para acessar o conteúdo e a localização do gerente, que irá prover a licença.

6 – caso o usuário (consumidor) não possui a licença ou não é válida, ele pode contactar o gerente para solicitar uma licença e realizar o pagamento necessário.

7 – após o usuário realizar o pagamento, o gerente pode emitir a licença. O tipo de pagamento realizado determina os direitos de acesso ao conteúdo.

8 – o gerente remete ao proprietário do conteúdo o resultado financeiro das transações (após a dedução do seu serviço). Ele também pode prover alguma informação proveitosa de cada transação.

4.4.2 Necessidades do Sistema DRM

Implementação de hardware e software

O DRM é inicialmente disponibilizado somente em PC. Sendo o PC o sistema do usuário final, quando conectado a Internet, torna-se uma ferramenta de fácil liberação de conteúdo, realização de atualizações e *downloads* de software de segurança e DRM. O PC é considerado uma implementação de software.

A proteção do conteúdo também pode vir na forma de implementação de hardware. Por exemplo, o contorno da infraestrutura para a área de armazenamento de dados em discos óticos, protegido contra acessos não autorizados. Esta área só pode ser acessada por hardware em concordância com o acesso.

Interoperabilidade e Mobilidade

A Internet é quase sempre acessível para todas as formas de elementos computacionais, onde a natureza heterogênea do cenário computacional não limita o acesso à Internet. O DRM poderia ter a mesma acessibilidade, mas eles, na sua maioria, são sistemas proprietários empregando formato de dados proprietários e técnicas de criptografia confiáveis. Estes aspectos proporcionam uma grande ausência de interoperabilidade entre sistemas DRM diferentes, limitando os usuários no seu uso.

Pode-se ainda citar que como a maioria das licenças DRM está limitada a dispositivos e não a usuários, obtendo o acesso ao conteúdo somente ao dispositivo liberado.

Segurança

A segurança é um requisito fundamental no DRM. Necessidades essenciais de segurança nos sistemas DRM incluem: confidencialidade e integridade do conteúdo (obtida via o uso de criptografia, assinaturas e certificados digitais), identificação única do usuário para o controle de acesso (pode ser verificado no agente) e mecanismos de proteção à falsificação (preocupação com duas importantes áreas: o conteúdo protegido e o player do usuário final) para processar o conteúdo protegido e aplicar as regras de uso do conteúdo.

Na maioria dos sistemas DRM existentes, a preocupação com a segurança é com o conteúdo liberado (que pode prover confidencialidade e integridade) no canal ao invés do conteúdo propriamente dito. Logo, este tipo de decisão é um aspecto de fraqueza na segurança, porque não evita a cópia e a redistribuição ilimitada do conteúdo protegido. Com isso, a proteção do conteúdo tem que ser persistente, evitando que estes tipos de ações possam ser realizados.

Privacidade do Usuário

Os usuários desejam opções para consumir o conteúdo de forma anônima e não ter o seu comportamento de consumo estabelecido em um perfil. O sistema DRM deve estabelecer parâmetros para que este tipo de exigência de seus usuários seja cumprida.

4.4.3 Componentes do Sistema DRM

Apresentada as funcionalidades e as necessidades de um sistema DRM, necessita-se conhecer os componentes de um sistema DRM. Esta seção visa apresentar os componentes básicos do DRM. A Figura 4.11 apresenta os principais componentes dos sistemas DRM, onde a proteção do conteúdo é representada por uma caixa que pode ser visualizada da seguinte forma [Ku e Chi 2004]:

1 – o conteúdo é rotulado com um identificador único, junto com o meta-dada (dado sobre o dado);

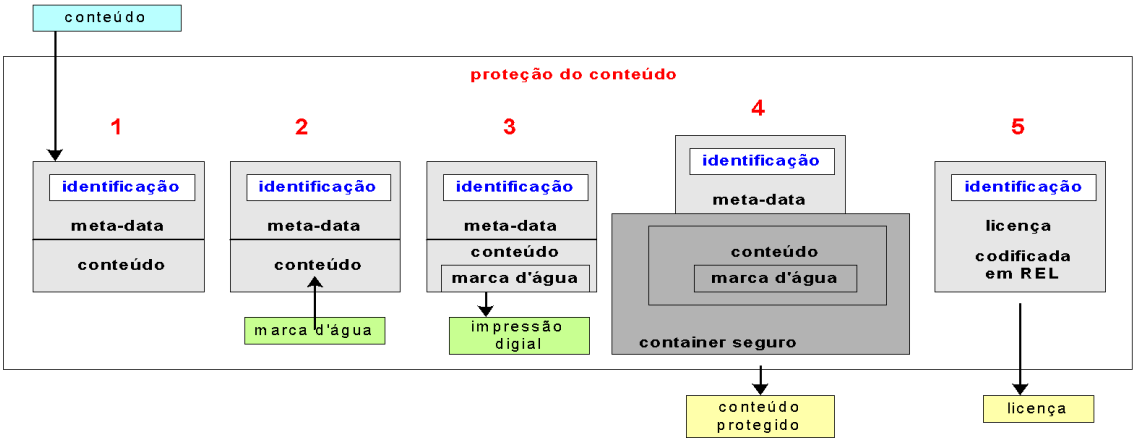


Figura 4.11. Componentes do sistema DRM.

2 – a marca d'água digital é inserida dentro do conteúdo para servir como uma prova de identidade do dono no evento de uma disputa.

3 – a impressão digital é gerado do conteúdo. É utilizado na aplicação para a autenticação, como identificação automática do conteúdo.

4 – o conteúdo está cercado por um container seguro, prevenindo acessos não autorizados.

5 – a licença, os direitos e as condições do uso do conteúdo são codificados em uma REL.

4.4.3.1 Identificação do Conteúdo e Meta-data

Antes dos direitos do conteúdo estarem definidos, ele tem de ser identificado para que os usuários que desejam acessa-lo possam comprar os seus direitos de uso. O meta-data do conteúdo pode prover alguma informação não sensível, como tipo de mídia, tamanho do arquivo, etc. Ele pode descrever alguma informação de como fazer uso do identificador do conteúdo.

Identificação do conteúdo

O identificador do conteúdo deve ser único para ser persistente. Apesar de mudanças no conteúdo, o identificador deve ser mantido. São utilizados esquemas de numeração padrão no DRM como: ISBN, ISSN, ISAN e DOI (Digital Object Identifier). A ACM utiliza o sistema DOI para itemizar cópias digitais de vários *proceedings* na sua biblioteca digital [DOI 2004].

O identificador do conteúdo também pode ser utilizado para localizar recursos ou derivações do conteúdo. Isto pode ser útil por exemplo, quando o usuário está interessado em localizar a versão do conteúdo para ser utilizado em seu dispositivo.

Meta-data

O meta-data complementa o uso do identificador do conteúdo (string alfanumérica). Se a infraestrutura do identificador do conteúdo (por exemplo DOI) é conhecido, então o identificador do conteúdo pode agir como um ponteiro para mais informações. Por outro

lado, o meta-data atenderia a esta capacidade e o contexto do DRM poderia prover mais informação para acessar o conteúdo.

4.4.3.2 Identificação/Autenticação do Usuário

Esta questão é importante porque deseja-se que somente usuários autorizados sejam habilitados a acessar o conteúdo. O problema está na dificuldade de identificar o usuário na maioria dos sistemas DRM, porque o conteúdo está relacionado ao dispositivo ao invés do usuário. Por exemplo, o DRM Microsoft admite relacionar o conteúdo de áudio exatamente a uma máquina via o uso do número serial do hardware como entrada. Isto quer dizer que o usuário não pode acessar facilmente o conteúdo que ele pagou via o uso de seus dispositivos (ex: PC caseiro ou do seu trabalho), ele está autorizado a acessar somente de um dispositivo designado.

A identificação/autenticação do usuário pode ser delegada ao gerente que pode utilizar tecnologias como SSL para superar este problema. Existe também o conceito de *Single Sign-On* (SSO) pelo qual os usuários somente fazem *login* para acessar os serviços via múltiplas plataformas. O Microsoft DRM necessita que os usuários DRM registrem antes o seu serviço SSO para poder acessar o conteúdo.

4.4.3.3 Marca D'água Digital

A tecnologia de marca d'água pode ser utilizada para controle de cópia, identificação de conteúdo e cópia. A maioria das técnicas de marca d'água usa uma abordagem que é a inserção de um sinal ruidoso com pequena intensidade dentro do conteúdo. Esta marca d'água pode ser detectada via a utilização de métodos específicos e relacionados com a chave secreta, responsável em detectar e remover a marca d'água pelas partes autorizadas [Katzenbeisser e Veith 2003].

No DRM, o conteúdo é tipicamente vulnerável a ataques nos sistemas dos usuários finais. O conteúdo pode ser capturado durante a sua reprodução ou ter os seus mecanismos de proteção removidos pelos ataques diretos. A marca d'água pode ser utilizada para detectar cópias ilegais de conteúdo que estão desprotegidos quanto a este tipo de ataques. Esta detecção é realizada pelos sistemas de usuários finais que detectam a marca d'água e a ausência de um mecanismo de proteção associado que é suposto para vir com o conteúdo. O sistema do usuário final pode também reportar e assistir na trilha destas cópias ilegais.

As necessidades básicas da marca d'água são:

Imperceptível – a marca d'água não deve afetar a qualidade do conteúdo;

Segurança – a marca d'água deve ser somente acessível pelos parceiros autorizados; e

Robustez – a marca d'água deve ser persistente e resistente a ataques.

4.4.3.4 Identificação baseada no Conteúdo (Impressão digital)

A identificação baseada no conteúdo usa de características existentes no conteúdo com base nas suas representações (sinais e características) e as compara com entradas existentes no banco de dados. O termo impressão digital tem sido utilizado em conjunto

com a marca d'água. A impressão digital é diferente da marca d'água, sendo vista na seguinte comparação (Tabela 4.2):

Tabela 4.2. Tabela comparativa entre marca d`água e impressão digital.

Marca d'água	Impressão digital
Embute o sinal no conteúdo, alterando-o	Não embute o sinal no conteúdo.
Não é em função do conteúdo.	É em função do conteúdo.
Requer acesso prioritário ao conteúdo.	Não requer esta prioridade e pode ser usado para a legalidade do conteúdo.
Deve ser refeito para todas as cópias no caso de novas tecnologias.	Não possui esta necessidade.
Nenhum tratamento adicional para novos conteúdos.	Existe a necessidade de armazenar as impressões digitais dos novos conteúdos em um banco de dados.

4.4.3.5 Containers Seguros

Os containers seguros são implementados com algoritmos de criptografia tais como DES (Data Encryption Standart) e AES (Advanced Encryption Standart) [Buenett e Paine 2002]. Junto com certificados e assinaturas digitais o container seguro oferece um conteúdo com confidencialidade e integridade. A integridade pode ir mais adiante com a utilização de mecanismos de autenticação para o conteúdo.

Um aspecto interessante é que o uso do conceito de container seguro foge ao conceito da transação comercial eletrônica tradicional, pelo qual a chave do conteúdo e o conteúdo protegido são transmitidos juntos na mesma transação. No DRM, a proteção do conteúdo é realizada offline e a chave do conteúdo é obtida separadamente. A mesma chave pode ser utilizada para várias transações (distribuição do conteúdo protegido), tornando-se uma vulnerabilidade.

As regras de uso do conteúdo podem ser codificado em seu meta-data ou em licenças. Codificando nas licenças, possibilita uma maior flexibilidade na determinação específica das regras de uso, porque as regras irão basear-se nas necessidades do usuário.

4.5. The $UCON_{ABC}$ Usage Control Model

O UCON (Usage Control) é um novo modelo de controle de acesso, diferente dos modelos tradicionais, em que a autorização pode ser feita também em tempo de requisição. O UCON é um modelo que estende os modelos de controle de acesso tradicionais em vários aspectos. O acesso pode ser uma ação instantânea, ou pode ser uma ação contínua, durante um determinado período de tempo, com várias ações seqüenciais e próximas. A decisão de acesso pode ser realizada antes, durante o processo de acesso, ou em ambos os casos, e as ações durante o período de acesso podem resultar em alterações de atributos [Sandhu e Park 2004]. Esta seção irá apresentar as características básicas deste modelo, mostrando o quanto ele modifica alguns conceitos de controle de acesso já conhecidos e o quanto ele melhora o controle propriamente dito.

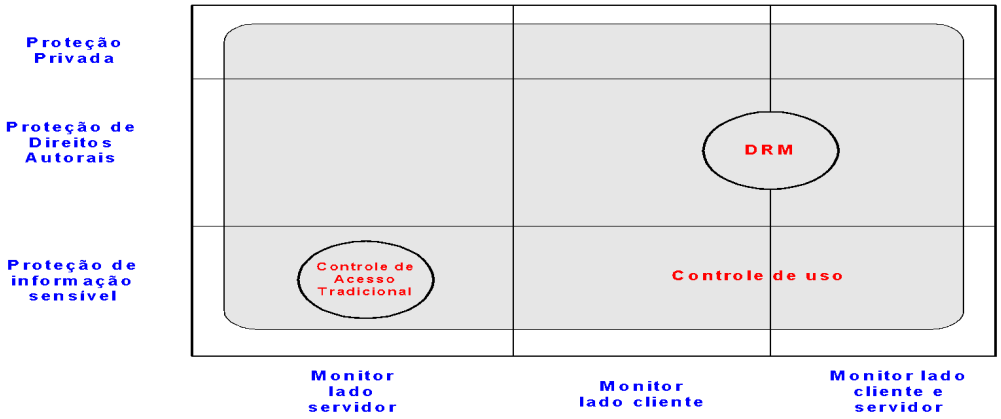


Figura 4.12. Cobertura do modelo UCON.

As decisões de uso no UCON são realizadas por políticas de autorização, obrigação e condição - UCONABC.

> As decisões de autorização são determinadas por políticas, utilizando atributos do sujeito, objeto e direitos.
>
> As obrigações são ações que tem que ser executadas por sujeitos, antes ou durante, o processo de acesso.
>
> As condições são necessidades pertinentes ao sistema ou ao ambiente em que opera que tem que ser satisfeitas antes ou durante o acesso.

A Figura 4.12 mostra a cobertura do UCON sobre os controles de acesso e seus relacionamentos com outros modelos.

4.5.1 Aspectos gerais do UCON

No UCON, os objetos estão relacionados com consumidores, provedores e identificadores. O consumidor busca o acesso ao objeto oferecido pelo provedor. O objeto pode conter informações privadas de elementos. Estes elementos são chamados de identificadores e guardam certos direitos no objeto. A decisão de uso é baseada nos relacionamentos entre estes diferentes componentes (consumidor, provedor e identificador) em relação ao objeto, e não mais tomada numa só direção. A parte do núcleo do UCON negocia com os aspectos relativos à tomada de decisão em relação da utilização dos objetos pelos consumidores.

Tradicionalmente, o controle de acesso tem negociado com autorizações como a base para o seu processo de tomada de decisão. No modelo UCON_{ABC}, o processo de tomada de decisão utiliza os atributos do objeto e do sujeito. Os atributos podem ser: identidades, rótulos de segurança, propriedades, capacidades, etc. Os predicados de segurança, obrigações e condições, podem ser avaliados antes ou durante o exercício de uma requisição. Em adição, o uso de um objeto pode necessitar de atualizações nos atributos do usuário (sujeito) ou do objeto antes, durante ou após o exercício do uso do objeto/recurso.

As seguintes propriedades distinguem o UCON dos modelos de controle de acesso tradicionais [Sandhu e Park 2004]:

A continuidade do processo de decisão de acesso; e

A mutabilidade dos atributos do sujeito e do objeto.

A continuidade e mutabilidade no UCON introduzem os conceitos de interatividade e concorrência, onde o acesso resulta na atualização dos atributos do sujeito ou do objeto. Estas mudanças, por outro lado, resultarão alterações durante ou em futuros acessos pelo mesmo sujeito, objeto ou algum acesso que esteja implicitamente relacionado. Logo, estas mudanças podem alterar não só o estado do acesso, mas também dos que estiverem relacionados.

4.5.2 Componentes do modelo $UCON_{ABC}$ [Sandhu e Park 2004]

O modelo $UCON_{ABC}$ possui oito componentes: sujeitos (usuários), atributos dos sujeitos objetos (recursos), atributos dos objetos, direitos, autorizações, obrigações e condições. As autorizações, obrigações e condições são predicados funcionais que existem para serem avaliados em uma decisão de uso. Os sujeitos, objetos, e direitos podem ser divididos em vários componentes, detalhados com diferentes perspectivas.

O controle de acesso tradicional utiliza somente autorizações para o processo de decisão. As obrigações e condições são os novos conceitos, os quais podem resolver certas deficiências encontradas em modelos de controle de acesso tradicionais. Outro aspecto significante do $UCON_{ABC}$ é que os atributos do sujeito e do objeto podem ser mutáveis, isto é, são mudados como conseqüência do acesso. Por exemplo: políticas que exigem limites no número de acessos pelos usuários, podem ser facilmente especificadas usando atributos mutáveis.

A Figura 4.13 apresenta os relacionamentos entre os componentes. O processo de decisão é mostrado como o relacionamento entre sujeitos, objetos e direitos que necessitam autorizações, obrigações e condições.

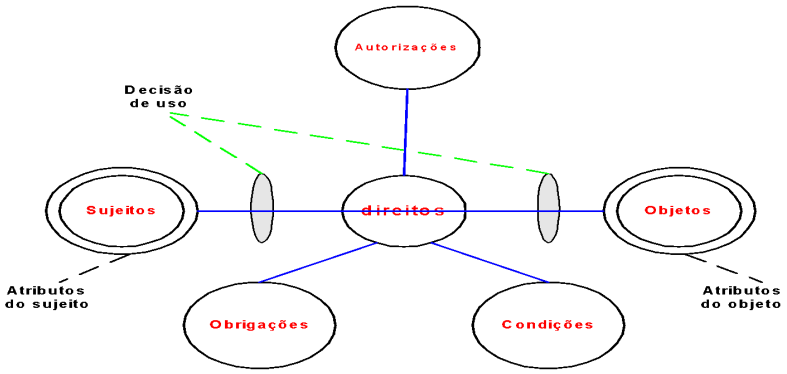


Figura 4.13. .O relacionamento dos componentes UCON.

Sujeitos e seus atributos

O sujeito (usuário) é uma entidade com atributos e determinados direitos de execução nos objetos (recursos). Os atributos do sujeito são propriedades ou capacidades que podem ser utilizadas no processo de decisão do uso. Exemplos de atributos: identidades, nomes de grupos (conjunto de usuários que possuem os mesmos direitos), papéis, etc. Os atributos no sujeito podem ser: [Sandhu, Park e Zhang 2004]

Imutável - ele não pode ser mudado pela atividade do usuário, somente ações administrativas podem mudá-lo; e

Mutável - pode ser modificado como efeito do acesso do usuário ao objeto/recurso (ex: atributo mutável - crédito).

Objetos e seus Atributos

Os objetos são basicamente os recursos oferecidos pelo sistema. Os objetos são também associados com atributos, e estes possuem certas propriedades que podem ser utilizadas nas tomadas de decisão. No caso do objeto classe, por exemplo, ele pode ser usado para estabelecer categorias de objetos, possibilitando que a autorização possa ser realizada não somente em um objeto, mas no conjunto de objetos que pertençam àquela classe. Alguns exemplos de atributos para objetos podem ser citados: valor, permissão, função, etc. Imagine a seguinte aplicação do atributo valor: o livro "A Volta dos que não Foram" necessita de R$ 100,00 para ler e mais R$ 50,00 para imprimir. Conforme o pagamento realizado, o sujeito poderá ler e ou imprimir o livro.

Direitos

São privilégios que um sujeito pode manter e exercer em um objeto. Os direitos consistem em um conjunto de funções de uso que habilita o acesso de um sujeito a objetos. Os direitos podem ou não ter hierarquia e podem ser divididos em: direitos do consumidor, direitos do provedor e direitos do identificador.

No UCON_{ABC}, o conceito de direito é semelhante ao do controle de acesso (direito de leitura e escrita, por exemplo). Porém, existe uma diferença: o UCON_{ABC} não visualiza os direitos como na matriz de acesso, independente da atividade do sujeito, eles estão relacionados com os atributos do sujeito, atributos do objeto, autorizações, obrigações e condições (ex. de direitos – uso de objetos, delegação de direitos e direitos para administração de acesso, etc). Imagine no caso do direito de ler um livro pela Internet, esse direito foi dado a:

Um determinado sujeito – atributo do sujeito – identificação;

Atributo do objeto livro – leitura e tempo indeterminado;

Obrigação – pagamento do valor estipulado; e

Condição – a leitura seja realizada somente no ambiente Windows.

Autorizações

As autorizações são predicados funcionais utilizados na avaliação da decisão de uso. As autorizações avaliam os atributos do sujeito, atributos do objeto e direitos requisitados junto com o conjunto de regras de autorização para a decisão de uso.

As autorizações podem ser:

Pré-autorização – é executada antes de um direito requisitado seja exercido.

Autorização em andamento – é executada enquanto o direito é exercido. A autorização em andamento pode ser executada continuamente ou periodicamente durante o tempo de acesso.

De forma geral, as políticas de controle de acesso tradicionais, incluindo MAC, DAC e RBAC utilizam, de alguma forma, a pré-autorização para as suas decisões, bem como o DRM em alguns casos. Algumas autorizações podem necessitar atualizações nos atributos dos objetos e sujeitos. Estas atualizações podem ser feitas antes, durante ou após (ex: créditos pré-pagos para utilização de recursos).

Obrigações

As obrigações são predicados funcionais que verificam as necessidades mandatórias que um sujeito tem que desempenhar antes ou durante o exercício do uso. As obrigações podem ser:

> Pré-obrigação – é um predicado que utiliza algum tipo de histórico de funções para verificar se certas atividades tenham sido realizadas ou não e retorna verdadeiro ou falso (ex: um usuário deve ter preenchido alguns dados antes de ler um documento de uma Empresa).
>
> Obrigação em andamento – é um predicado que tem que ser satisfeito continuamente ou periodicamente enquanto os direitos admitidos estão em uso (ex: um usuário tem que verificar certos avisos enquanto "logado").

As obrigações podem ou não possuir atributos. Os atributos podem ser utilizados para determinar quais tipos de obrigações são necessárias para a aprovação do uso. Pode-se dizer que os atributos não são usados para tomadas de decisão, com respeito a obrigações, mas são utilizados somente na escolha do que as obrigações aplicam-se.

Condições

São fatores de decisão com base no sistema ou no ambiente. O predicado condição avalia o status do sistema ou ambiente para verificar se as necessidades são satisfeitas ou não (*true* ou *false*).

Os atributos de objetos e sujeitos podem ser usados para escolher que condições têm que ser usadas para uma requisição. Portanto, nenhum atributo é incluído dentro de suas próprias requisições. Diferente das obrigações e autorizações, as variáveis das condições não podem ser mutáveis, porque as condições não estão diretamente sob o controle de sujeitos (ex: status de segurança de um sistema, carga do sistema, variação de fuso horário para transações comerciais).

4.5.3 Os principais modelos da família $UCON_{ABC}$ [Sandhu e Park 2004]

Apresentados os conceitos básicos do modelo UCON, esta seção visa apresentar o conjunto de modelos da família UCON. Os modelos que serão apresentados são considerados os principais, porque visam o processo de aplicação, não incluindo construtores administrativos. A classificação dos modelos é baseada nos seguintes critérios:

> Fatores de decisão – que consiste de autorizações, obrigações e condições;
>
> Continuidade de decisão – ou pré ou em andamento, com respeito ao acesso em questão; e

Mutabilidade – que pode admitir atualizações nos atributos dos sujeitos e objetos.

Se todos os atributos são imutáveis, nenhuma atualização é possível no processo de decisão. Este caso é denotado como '0'. Com atributos mutáveis, atualizações são possíveis antes, durante ou após o direito ser exercido. Denota-se 1, 2 e 3 respectivamente (Tabela 4.3).

Tabela 4.3. Os 16 modelos $UCOM_{ABC}$ básicos.

	0 (immutable)	1 (pre-update)	2 (ongoing-update)	3 (post-update)
preA	S	S	N	S
onA	S	S	S	S
preB	S	S	N	S
onB	S	S	S	S
preC	S	N	N	N
onC	S	N	N	N

Exemplo: Suponha que Alice é membro de uma biblioteca musical, e que ela pague R$ 1,00 por hora de música tocada. Este exemplo pode ser tratado como uma pré-autorização com atualização posterior, não existindo a necessidade de realizar alguma atualização durante a execução da música. Se o fator de decisão é durante, a atualização pode ocorrer antes, durante e após a execução. Explica as quatro primeiras linhas da tabela. Para as duas linhas restantes, o fator de decisão é a condição.

A Figura 4.14 mostra a continuidade de decisões, com as possibilidades de mutação de atributos que podem ocorrer antes ou durante a execução.

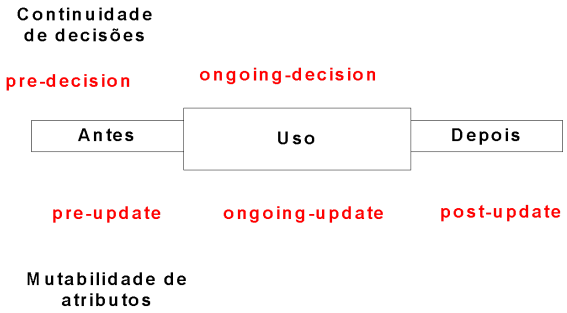


Figura 4.14. Continuidade de decisões.

A Figura 4.15 apresenta as possíveis combinações do modelo $UCON_{ABC}$ e suas relações, considerando-se que cada A, B e C estão na base do modelo. O próximo nível possui as combinações de dois deles e assim por diante. Desta forma, foi apresentada de forma sucinta que combinações de A, B e C podem ser utilizadas em um contexto.

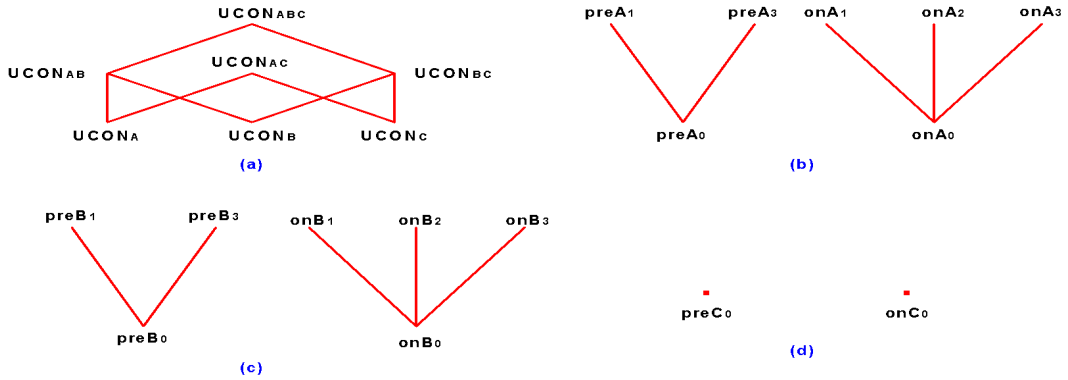


Figura 4.15. Combinações do modelo UCON.

A seguir serão apresentados alguns dos 16 modelos da família UCON. Esta apresentação visa mostrar a validação destes modelos como exemplificá-los.

Modelo de pré-autorização - $\text{UCON}_{\text{preA}}$

As autorizações têm sido consideradas como o núcleo do controle de acesso, como o uso da pré-autorização na tomada de decisão. O modelo $\text{UCON}_{\text{preA}}$ utiliza a pré-autorização no seu processo de decisão de uso, onde o processo de decisão de autorização é realizado antes do uso ser admitido. Existem três modelos detalhados baseados nas variações de mutabilidade.

> $\text{UCON}_{\text{preA0}}$ – modelo de pré-autorização imutável que não requer atualização. Ele possui os seguintes componentes.

S;O;R;ATT(S);ATT(O) and preA (subjects, objects, rights, subject attributes, object attributes, and pre-authorizations respectively);

> allowed(s; o; r) => preA(ATT(s);ATT(o); r).
>
> $\text{UCON}_{\text{preA1}}$ – é um modelo com procedimento de pré atualização opcional. Ele inclui funções de atualização que modificam os atributos antes do seu uso ser iniciado. O modelo é idêntico ao $\text{UCON}_{\text{preA0}}$ exceto pela adição dos seguintes processos de pré-atualização.

preUpdate(ATT(s)); preUpdate(ATT(o)), um procedimento opcional para executar operações de atualização em ATT(s) e ATT(o), respectivamente. Veja que o preUpdate pode incluir operações não determinísticas.

> $\text{UCON}_{\text{preA3}}$ - é um modelo com procedimento de pós-atualização opcional. A atualização posterior utiliza funções para modificar certos atributos após o uso ter encerrado. O modelo é idêntico ao $\text{UCON}_{\text{preA0}}$ exceto pela adição dos seguintes processos de pós-atualização.

postUpdate(ATT(s)); postUpdate(ATT(o)), um procedimento opcional para executar operações de atualização em ATT(s) e ATT(o), respectivamente. Veja que o preUpdate pode incluir operações não determinísticas.

Exemplo 1 MAC policies, UCONpreA0 :

> L é uma lattice de rótulos de segurança com relação de dominância¸

clearance : S → L

classification : O → L

ATT(S) = {clearance}

ATT(O) = {classification}

allowed(s; o; read) ⇒ clearance(s) ≥ classification(o)

allowed(s; o;write) ⇒ clearance(s) ≤ classification(o)

Neste exemplo, os rótulos de segurança (*clearance* e *classification*) são usados como um atributo do sujeito e do objeto e as propriedades de segurança são utilizadas para pré-autorizações. Se a *clearance* do sujeito s domina a *classification* do objeto o, a solicitação de *read* é admitida. O processo é semelhante no *write*.

Exemplo 2 DAC utilizando políticas fechadas ACL com um ID individual, UCONpreA0 :

N é um conjunto de nomes identificadores

id : S → N, mapeamento de um para um

ACL : O → 2^{NxR}

ATT(S) = {id}

ATT(O) = {ACL}

allowed(s; o; r) => (id(s); r) ∈ ACL(o)

No exemplo do DAC, identidades individuais ou de grupo e a ACL são atributos do sujeito e do objeto respectivamente. A ACL é um mapeamento funcional do objeto para múltiplos ids e direitos. Se a identidade de um sujeito junto com o direito à requisição existe na ACL, a requisição é admitida.

No RBAC, a função do usuário e as permissões pertinentes a sua função podem ser considerados como atributos do sujeito e do objeto.

Modelos ongoing-obrigações - $UCON_{onB}$

São modelos similares ao UCONpreB, exceto pelas obrigações que tem serem realizadas enquanto os direitos são exercidos. Eles podem ser realizados de forma periódica ou de forma contínua. Para isto foi introduzido o parâmetro tempo T como partes das obrigações onOBL. O parâmetro T é definido como intervalos de tempo baseados em períodos ou eventos.

Por exemplo, um sujeito pode clicar em um aviso para ser executado em 20 dias (deseja se registrar agora ou mais tarde).

O modelo $UCON_{onB}$ possui os seguintes componentes:

O modelo UCONonB0 possui os seguintes componentes:

S;O;R;ATT(S);ATT(O);OBS;OBO; e OB não são mudados pelo UCONpreB;

T, é um conjunto de valores de tempo ou elementos deu m evento;

onB and onOBL, (os predicados ongoing-obligations e os elementos ongoing-obligation, respectivamente);

onOBL $\subseteq$ OBS x OBO x OB x T;

getOnOBL : S x O x R $\rightarrow 2^{onOBL}$, a função escolhe um ongoing-obrigations para uma resquisição de uso

onFulfilled : OBS x OBO x OB x T $\rightarrow$ {true; false};

onB(s; o; r) = $\bigwedge$(obs_i; obo_i; ob_i;t_i)$\in$getOnOBL(s;o;r) ^{onFulfilled}(obs_i; obo_i; ob_i;t_i);

onB(s; o; r) = true por definição se getOnOBL(s; o; r) = $\varnothing$;

allowed(s; o; r) $\Rightarrow$ true;

stopped(s; o; r) $\Leftarrow \neg$onB(s; o; r).
O modelo UCONonB1 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos pré-atualizações:

preUpdate(ATT(s)); preUpdate(ATT(o)): um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.
O modelo UCONonB2 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos ongoing-atualizações:

onUpdate(ATT(s)); onUpdate(ATT(o)): um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.
O modelo UCONonB3 é idêntico ao UCONonB0 exceto pela adição dos seguintes processos pós-atualizações:

postUpdate(ATT(s)); postUpdate(ATT(o um procedimento opcional para alterar alguns atributos como conseqüência das pré-obrigações.

Mostra-se um simples exemplo do UCONonB0 .

Visualiza-se um aviso no windpws enquanto s exerce r, UCONonB0 :

OBS = S

OBO = {ad window}

OB = {keep active}

T = {always}

getOnOBL(s; o; r) = {(s; ad window; keep active; always)}

allowed(s; o; r) $\Rightarrow$ true

stopped(s; o; r) $\Leftarrow \neg$onFulfilled(s; ad window; keep active; always)

Aqui, somente uma obrigação ongoing é requisitada. Suponha um provedor de serviço livre na Internet que solicita que os usuários vejam os avisos enquanto conectados ao servidor. Neste caso, não existe solicitação que tenha que ser completada antes do uso do serviço. Tão logo o aviso é ativado, o serviço é liberado.

Modelo pré-condições - $\mathbf{UCON_{preC}}$

Este modelo inclui algumas restrições ambientais que não estão diretamente relacionadas aos sujeitos e objetos. O ambiente atual e o status do sistema é retornado, e a cada vez que isto acontece a condição é avaliada. Via a utilização de condições no processo de decisão de uso, o UCONc pode prover controles finos de uso. Ao contrário dos modelos de autorização e obrigação, o de condição não pode ser mutável.

As seguintes definições formalizam o modelo $UCON_{preC:}$

O modelo UCONpreC0 possui os seguintes componentes:

S;O;R;ATT(S); and ATT(O) are not changed from UCONpreA;

preCON (a set of pre-conditions elements);

getP reCON : S x O x R $\rightarrow 2^{preCON}$;

preConChecked : preCON $\rightarrow$ {true; false};

preC(s; o; r) = $\wedge$ preCon$_i \in$ getP reCON(s;o;r) preConChecked(preCon$_i$)

allowed(s; o; r) $\Rightarrow$ preC(s; o; r).

No $UCON_{preC0}$, o preC é utilizado no processo de decisão de uso junto com S, O e R. O conjunto relevante de elementos de condição preCON é escolhido com base na possível requisição, usando os atributos do sujeito e do objeto. Para admitir uma requisição, todas as restrições de condição devem ser avaliadas.

Por exemplo, suponha que existam requisições para restringir localizações onde o uso pode ser exercido. Isto pode ser realizado, por exemplo, verificando o endereço IP antes do uso ser admitido ($UCON_{preC0).}$

Exemplo: limite de localização, UCONpreC0 :

studentAREA; facultyAREA (admite códigos de área para student e faculty) curArea is código de área de um dispositivo atual

ATT(s) = fmemberg

preCON = {(curArea $\in$ studentAREA); (curArea $\in$ facultyAREA)}

getP reCON(s; o; r) = (curArea 2 studentAREA); if member(s) =`student';

(curArea 2 facultyAREA); if member(s) =`faculty'.

allowed(s; o; r) $\Rightarrow$ preConChecked(getP reCON(s; o; r))

O exemplo verifica a localização corrente de um usuário em tempo de solicitação. As localizações admitidas pelo estudante e pela Universidade podem ser diferentes e tenha que se chegar a um acordo. Este exemplo assume que não existe mudança de localização enquanto a requisição é exercida ou não exista restrição de mudanças de localização durante o uso se a localização original tenha sido aprovada.

4.6 Modelos de pesquisa

A comunidade de segurança no âmbito mundial vem buscando melhorias nos procedimentos de controle de acesso. Esta seção apresenta alguns modelos desenvolvidos por pesquisadores que apresentam novas abordagens sobre o assunto.

4.6.1 Or-BAC (Organization Based Access Control)

O Or-BAC é um modelo de controle de acesso que tem o conceito de organização como a sua principal linha de ação. Este modelo não está restrito somente a garantir ou não permissões, mas possibilita estabelecer proibições, obrigações e recomendações.

O conceito de organização é visto como um grupo organizado de sujeitos representando um papel. Isto significa que o papel exercido por um sujeito corresponde a alguma relação de concordância entre os sujeitos na formação da organização.

4.6.1.1 O Modelo Or-BAC [Kalam, et. al. 2003]

Vários modelos são utilizados para compor o modelo Or-BAC como um todo. Todos possuem a sua representação sob a forma do modelo E-R (entidade – relacionamento), onde as suas entidades são relacionadas entre si dentro de uma organização. Por exemplo, o modelo sujeitos-papéis.

Sujeitos e papéis

Os sujeitos no modelo são entidades ativas, que podem ser representados por usuários (p. ex. Luiz, Ana, professor, aluno) ou organizações (p. ex. Departamento de ensino de uma Faculdade). Os papéis estruturam uma ligação entre os sujeitos e as organizações (professores horistas). A representação do relacionamento emprega no modelo Or-BAC é mostrado na Figura 4.16.

O modelo possibilita, via a entidade papel, estruturar os sujeitos e atualizar de forma simples a política de segurança quando novos sujeitos são adicionados ao sistema.

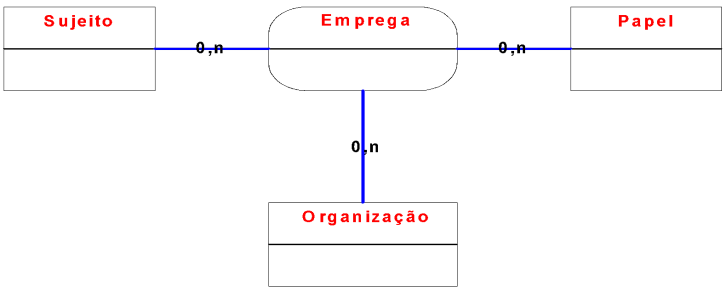


Figura 4.16. Modelo via entidade papel.

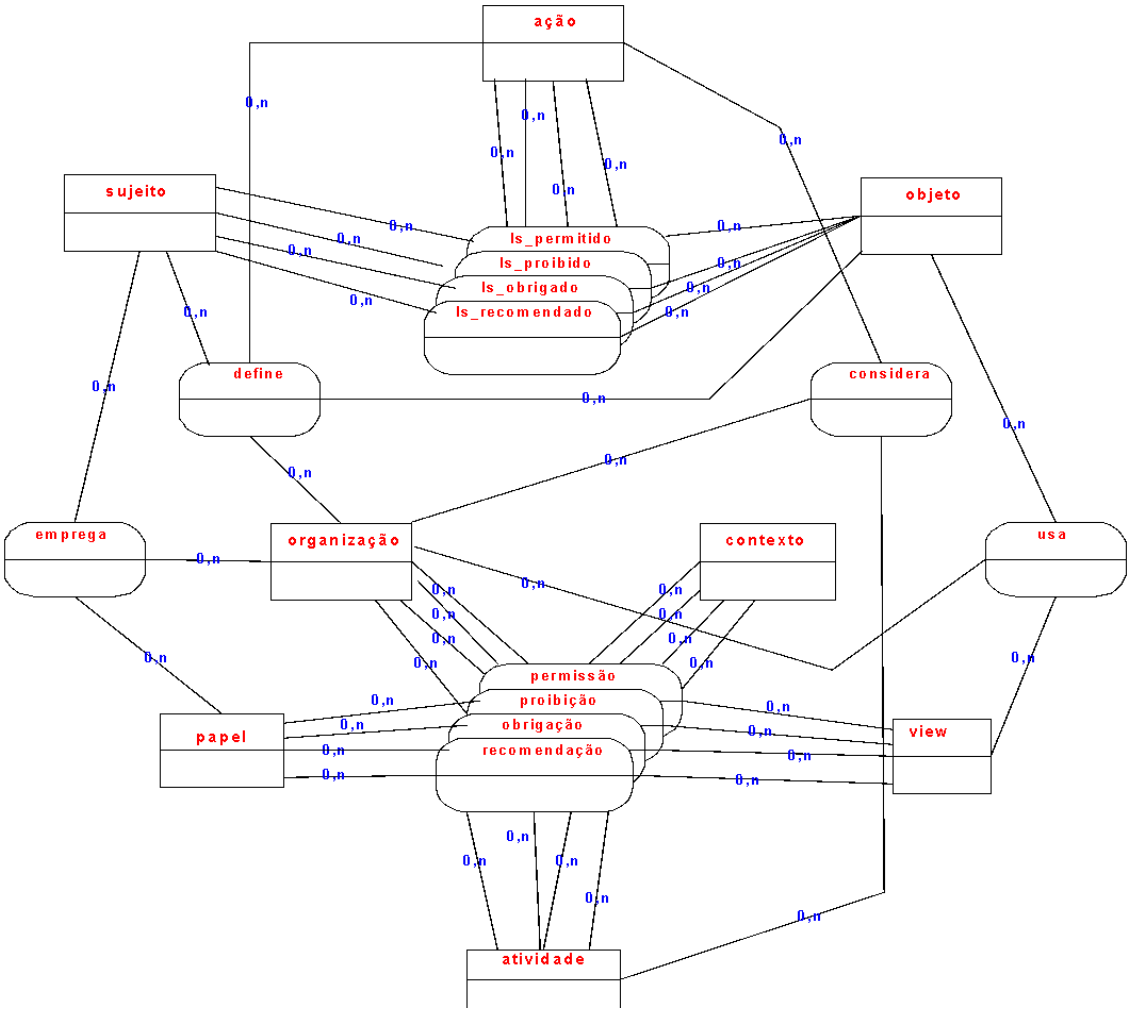


Figure 4.17. Modelo Or-BAC.

Autorização concreta

Apresentado todos os modelos que compõe o modelo Or-BAC (Figura 4.17), existe agora a possibilidade de modelar as permissões concretas. As permissões são chamadas desta forma porque foram introduzidos novos relacionamentos. Por exemplo, o conceito de Is_permitido como um relacionamento entre sujeitos (s), objetos (o) e ações (α) – Is_permitido(s, o, α), significando que um sujeito s pode realizar uma ação α em um objeto o. Os relacionamentos Is_proibido, Is_obrigado e Is_recomendado também são modelados de forma semelhante.

Cada instância do relacionamento Is_permitido é logicamente derivada das permissões garantidas para papéis, views e atividades obtidas do relacionamento Permissão. Sendo assim, este modelo faz da organização a sua abordagem, e o contexto a referência para a garantia do acesso do sujeito a um objeto na realização de uma ação.

4.6.2 TBAC – Tasked-based Authorization Controls

O TBAC é um modelo de controle de acesso que possui características diferentes dos controles de acesso tradicionais e modelos de segurança. Ele aborda a modelagem e a aplicação de segurança no nível dos serviços executados na empresa. Em função desta

nova abordagem, o TBAC cria um novo fundamento, o modelo de segurança ativo. Este fundamento trabalha a modelagem e aplicação de segurança sob a perspectiva de atividades ou serviços, provendo abstrações e mecanismos para ativar o gerenciamento de segurança, em tempo de execução, conforme a evolução dos serviços realizados. Outra característica do modelo de segurança ativo é a capacidade de gerenciamento ativo das permissões, elas são constantemente monitoradas, ativadas e desativadas conforme a evolução do contexto, isto é, em função dos serviços que estão sendo realizados [Thomas e Sandhu 1997, Thomas e Sandhu 1994].

Esta nova visão, segurança baseada na evolução de serviços, representa uma ruptura dos modelos de segurança clássicos, tal como aqueles baseados em uma ou mais variações da visão de segurança de controle de acesso entre sujeito e objeto e o controle centralizado das decisões de permissões. A flexibilidade de se obter autorizações baseadas em serviços (na sua evolução) possibilita uma maior agilidade nos processos de obtenção das permissões, não somente pela necessidade de se automatizar o processo de autorização e os controle de acesso relacionados, mas também pela ausência da figura do administrador de segurança na gerência durante a sua realização [Thomas e Sandhu 1997, Thomas e Sandhu 1993].

Para que se alcance realmente a agilidade e automatização proposta pelo modelo, a sua aplicabilidade deve ser direcionada a sistemas que provêem permissões curtas e em intervalos de tempo pré-determinados, principalmente em ambientes baseados em transações e *workflows*. Outro aspecto importante é o direcionamento para sistemas que tenham capacidade auto-administrativa, reduzindo a sobrecarga associada com a administração de segurança existente entre sujeito e objeto.

A utilização do TBAC em sistemas *workflows* possibilita que as permissões sejam garantidas, utilizadas e revogadas automaticamente, e coordenadas conforme a evolução dos diversos serviços realizados. Desta forma, se evita que as permissões sejam ativadas antes ou após surgir a necessidade dos serviços, ou que as mesmas estejam ainda ativas após os serviços terem sido encerrados, criando vulnerabilidades nos sistemas. O TBAC descarta também a responsabilidade do administrador de segurança manter um controle constante da evolução dos serviços e com isso as permissões compatíveis para as suas execuções. É claro que as permissões de controle de acesso, apesar de automatizadas, seguem a lógica da aplicação, bem como uma política de controle de acesso pré-estabelecida.

4.6.2.1 TBAC como modelo de segurança ativo

O conceito de modelo de segurança ativo caracteriza os modelos que reconhecem todo o contexto em que as requisições de segurança aparecem, e aplica a atividade de gerenciamento de segurança conforme a evolução do contexto, com base no progresso das atividades realizadas.

A Figura 4.18 apresenta o progresso da autorização (*Authorization-step*), considerada a mais fundamental abstração do TBAC, isto porque a autorização é fornecida passo a passo conforme a evolução da execução das tarefas (a autorização é fornecida conforme a necessidade de uso). Ela representa o progresso do processamento inicial da autorização, agrupando sujeitos (*executor*) confiáveis a um conjunto de permissões, semelhante no mundo do papel (ex: RBAC), onde um usuário ou um grupo de usuários

pode ser agrupado conforme o papel exercido e estar ligado a um conjunto de funções que podem ser exercidas (ex: o gerente de vendas Marco pode autorizar a ordem de venda X) [Thomas e Sandhu 1997].

O progresso da autorização no TBAC está associado com o grupo de *trustee* (depositário) chamado de *trustee-set*. Um membro deste conjunto (*executor-trustee* daquele passo) irá eventualmente permitir o progresso da autorização quando ele for instanciado. As permissões necessárias ao *executor-trustee* para invocar e permitir o progresso de autorização compõe um conjunto de permissões chamada de permissões do *executor*. As permissões que são habilitadas no progresso de autorização formam o conjunto de permissões habilitadas, e a união das permissões do *executor* e das permissões habilitadas é conhecida como estado de proteção do progresso de autorização. O período de validade e o ciclo de vida da atividade estão associados com toda o progresso de autorização.

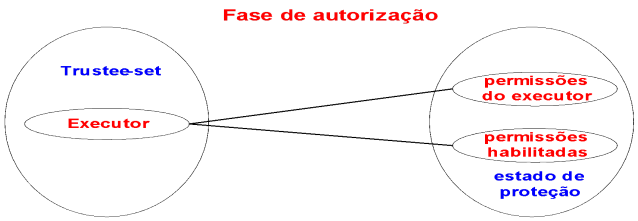


Figura 4.18. Authorization-step.

O TBAC difere dos modelos de controle de acesso tradicionais pela inclusão do domínio de progressos de autorização e do domínio de estimativa de validade e uso, que embutem a informação contextual baseada em tarefas.

A Figura 4.19 [Thomas e Sandhu 1997] apresenta os conceitos, características e componentes que fazem o TBAC um modelo de segurança ativo, como:

> Modelagem de autorização em serviços e workflows, bem como o monitoramento e gerenciamento do processamento da autorização e ciclos de vida como progresso das tarefas;
>
> O uso do controle de acesso baseado no uso e no tipo;
>
> A manutenção de proteções separadas para cada progresso de autorização; e
>
> Execução dinâmica dos procedimentos de entrada e saída das permissões dos estados de proteção como progressos de autorização são processadas.

Todo progresso da autorização mantém o seu próprio estado de proteção. O valor inicial do estado de proteção é o conjunto de permissões que são ativadas como resultado da validação do progresso da autorização (*authorization-step*). Portanto, o conteúdo deste conjunto de permissões sofrerá mudanças conforme o progresso da autorização é processado e as permissões forem sendo utilizadas. Para cada permissão atrela-se uma estimativa de uso. Quando se alcança esta estimativa, a permissão associada é desativada e a atividade correspondente não é mais permitida. A constante verificação automática dos procedimentos de entrada e saída das permissões como autorizações é a principal característica que torna o modelo TBAC ativo. Além desta questão, existe o

aspecto de que toda a permissão no estado de proteção é única, e mapeada para uma instância do progresso da autorização e para uma tarefa que está invocando a autorização.

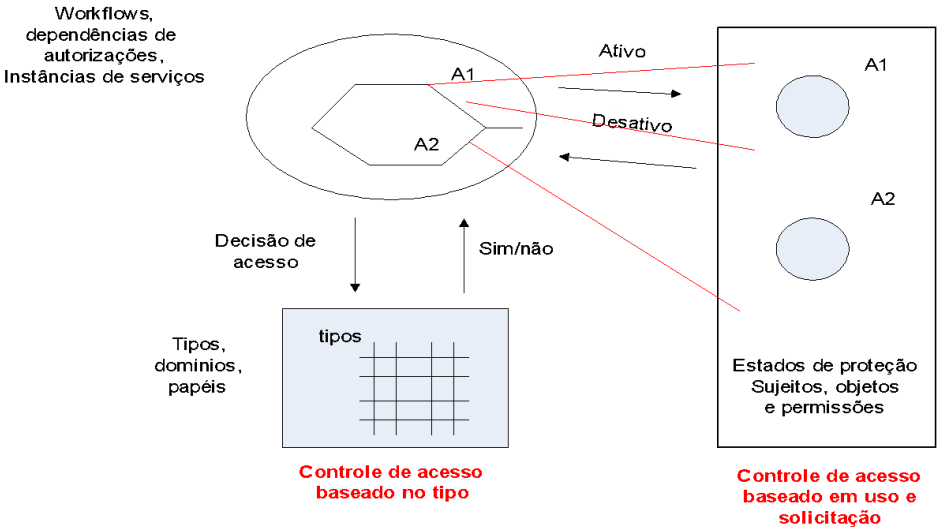


Figura 4.19. TBAC como modelo de segurança ativo.

As diferenças entre os controles de acesso baseado em uso e solicitação, e baseado em tipo é também uma característica significante do modelo TBAC. O controle de acesso baseado em tipo é utilizado para encapsular restrições de controle de acesso especificadas pela política de controle de acesso e aplicadas a tipos. Controle de acesso baseado em uso e solicitação é usado para modelar e gerenciar os detalhes do controle de acesso e as permissões das solicitações de autorização individuais, incluindo a manutenção do uso de permissões.

4.6.2.2 A família dos modelos TBAC

A família dos modelos TBAC possui como base o modelo $TBAC_0$, sendo um modelo que oferece algumas facilidades para modelar serviços, progressos da autorização e dependências, relatando vários progressos da autorização. É considerado o modelo mais genérico e flexível. Os modelos avançados $TBAC_1$ e $TBAC_2$ herdam as características de $TBAC_0$, mas também incorporam as suas próprias características. O $TBAC_1$ incorpora a noção de autorizações compostas e o $TBAC_2$ a questão de restrições. O $TBAC_3$ incorpora as funcionalidades dos três modelos anteriores.

4.6.3 TMAC – Team-based Access Control [Thomas 1997]

O TMAC é uma abordagem para aplicação do controle de acesso baseado em papel em ambientes colaborativos tais como aqueles que envolvem *workflow*. A abordagem em atividades colaborativas deve-se ao fato de se obter a perfeição via o uso de grupos organizados. Logo, o aspecto central da abordagem TMAC é a noção de grupo como uma abstração que encapsula uma grupo de usuários com o objetivo de executar um determinado serviço ou alcançar uma determinada meta.

O trabalho desenvolvido por esta abordagem de controle de acesso visa criar um paradigma de segurança, apresentando melhorias de segurança para um *workflow* de uma Empresa. Três objetivos são buscados nesta abordagem:

- Criar um ambiente seguro que não possa ser invadido pelo quadro de funcionários;
- Prover um sistema rígido quanto ao acesso a determinada informação de um segmento de negócio em um período de tempo; e
- Criar uma infra-estrutura de segurança que não necessite de um grande volume administrativo, isto é, possa ser difícil de administrar na sua maioria.

Estas questões podem ser mais bem entendidas via a apresentação dos modelos de segurança passivo e ativo. O modelo de segurança ativo é aquele que tem inicialmente a função da manutenção das atribuições de permissão, tal como no RBAC, onde as permissões são atribuídas a papéis, e distingue as ativações de permissão baseada em contexto e tarefas. Sendo assim, após a permissão ser atribuída, ela pode ser ativada ou desativada várias vezes de acordo com o contexto o qual está associado com a evolução de execução das tarefas.

No modelo passivo, a permissão é atribuída somente ao usuário. Assume-se sempre que ela pode ser ativada independente de qualquer outra consideração como o contexto. Este é o típico caso da obtenção da permissão com base na lista de controle de acesso (ACL).

4.6.3.1 Apresentação do Modelo

O TMAC trabalha com controle de acesso baseado em papel em ambientes colaborativos (*workflow*). Dois aspectos são necessários para o controle de acesso em atividades colaborativas:

1. a necessidade da permissão baseada em papel; e
2. a necessidade de se ter a ativação da permissão no nível de usuário e objetos individualmente.

O TMAC cria a abordagem de controle de acesso com as duas atividades, acima citada, trabalhando de forma concorrente. Para que isto aconteça necessita-se de:

> Uma abstração para limitar e modelar um conjunto de usuários e os seus respectivos papéis;
>
> Registro de memória de todo o contexto de colaboração para um conjunto de usuários.

Nos modelos RBAC, um grupo de usuários está relacionado a um papel, limitando o trabalho em equipe. O TMAC, por visar o conceito de euipe, introduz o conceito de contexto de colaboração. Este contexto contém a informação sobre toda as tarefas que serão executadas. Do ponto de vista de controle de acesso, o contexto de colaboração da equipe pode conter dois tipos de informação: contexto do usuário - os usuários que compõe a equipe; e contexto do objeto – o conjunto de instância de objetos necessária pela equipe para realizar a sua tarefa. Portanto, conhecendo a estrutura básica da equipe

em termos de seus vários papéis, encontra-se o aspecto 1, citado anteriormente, e conhecendo o contexto de colaboração encontra-se o aspecto 2.

A Figura 4.20 apresenta os principais conceitos do TMAC, e a interação entre eles.

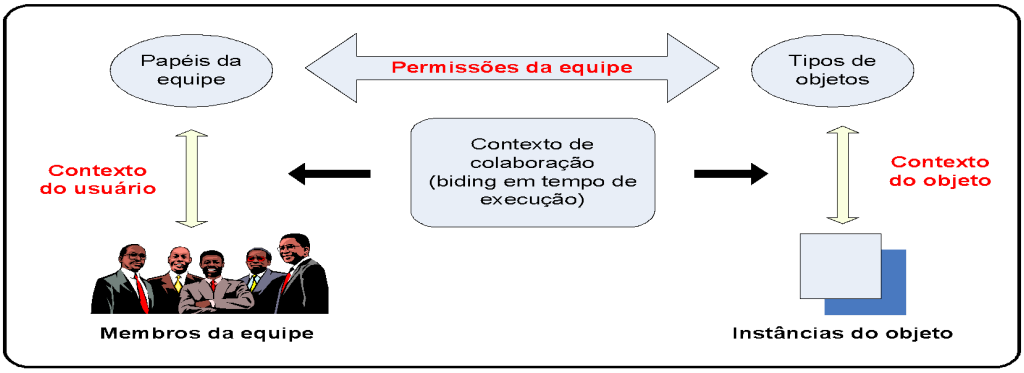


Figura 4.20. Principais conceitos do TMAC.

Uma equipe na abordagem TMAC consiste de:

O nome da equipe, T;

O conjunto de membros/usuários da equipe, TU;

O conjunto de papéis da equipe, TR – TR $\subseteq$ R, onde R é o conjunto total de papéis no sistema de informação;

O papel chefe da equipe (h), onde h $\subset$ TR. Somente um usuário pode ser o chefe da equipe a cada vez;

O conjunto dos tipos de objetos, OT;

O conjunto de instâncias de objetos, O;

O conjunto de permissões da equipe, TP, definida via TR e OT, isto é, TP $\subseteq$ TR x OT;

O contexto de colaboração consiste de dois componentes:

O contexto do usuário (UC), onde UC:TR x TU;

O contexto do objeto (OC), onde OC:OT x O.

A idéia básica no TMAC é usar o RBAC para definir o conjunto de permissões P via os domínios de R e OT. Equipes individuais de mesma estrutura (tipo/classe) englobará o mesmo subconjunto de papéis, TR de R e assim herdará o mesmo subconjunto TP e P. Porém, o TMAC invoca em tempo de execução a ligação do TP de cada equipe para os conjuntos TU e O da equipe. Isto admite a ativação em tempo de execução das permissões no nível de usuários e objetos individuais.

Sob o ponto de vista operacional e de implementação, o TMAC poderia suportar as seguintes primitivas para habilitar o controle de acesso na equipe como um todo:

- Usuário_atribuição (usuário, equipe): atribuição de um usuário a uma equipe;
- Usuário_desabilita (usuário, equipe): o usuário deixa de ser da equipe;

- Equipe_ativa (equipe): amarra as permissões da equipe aos membros da equipe e os objetos necessários (TU e O); e
- Equipe_desativa (equipe): desativas as permissões para toda a equipe.

Nas aplicações a ativação e desativação das permissões são realizadas por papel, mas podem existir casos onde elas são ativadas e desativadas usuário por usuário. Neste caso as primitivas de ativa_usuário e desativa_usuário podem ser habilitadas.

Para finalizar o TMAC pode ser auto-administrado via chamadas básicas emitidas pelo sistema de informação, atribuindo ou não os membros de uma equipe, em tempo de execução conforme a evolução do *workflow*. Esta ativação pode ser sincronizada com as primitivas de atribuição e ativação do usuário para automatizar a administração de segurança. Para preservar o controle de acesso a objetos individual via equipes, o contexto do objeto pode ser passado de uma equipe para outra.

Sendo assim, o TMAC possibilita formular um modelo de segurança que trata a natureza dos acessos baseado em equipe e trabalha de forma colaborativa. Ele possui a vantagem de ser capaz melhor administrar e modelar o sistema, via um maior refinamento no controle sobre ativação das permissões para objetos e usuários individuais.

4.7 Aspectos comparativos

Como visto durante o mini-curso, todos os modelos de controle de acesso visam minimizar os problemas de segurança estabelecendo o que, quando e como acessar, com base em uma política de controle de acesso, mas cada um com a sua abordagem. Esta seção busca apresentar uma pequena comparação entre estas diversas formas de abordagens, focando os seus aspectos positivos e ou negativos.

Os modelos de controle de acesso tradicionais como RBAC, MAC e DAC consideram inicialmente as decisões de autorizações estáticas baseadas nas permissões de sujeitos em objetos. Eles também possuem sistemas de gerenciamento de autorização baseados em políticas, possuidores de um monitor de referência centralizado ou distribuído, com administração centralizada, que verifica as permissões de cada sujeito quando este requisita o acesso. A permissão é garantida ao sujeito conforme a política de segurança em tempo de requisição de acesso, e não existe um limite de tempo de uso do objeto.

O DAC é um controle de acesso simples de ser implementado, pois faz uso de ACLs, e é implementado na maioria dos sistemas operacionais existentes, tornando a sua utilização bastante divulgada na comunidade de segurança. Porém, este modelo possui falhas quanto aos aspectos de segurança. Estas falhas são causadas devido a possibilidade da passagem de permissão sem um conhecimento prévio de seu dono ou do gerente de segurança, provendo vulnerabilidades de acesso aos objetos do sistema. Arquivos que possuam informações sensíveis podem tramitar entre sujeitos que não estejam qualificados a acessá-los. Portanto, o DAC não é um controle de acesso que possui características que garantam de forma efetiva o acesso somente a sujeitos que realmente possuam direito de acesso.

O MAC pode estender ou substituir o DAC por permissões de sistemas de arquivos os conceitos de usuários e grupos. É uma estrutura mais complexa de ser implementada,

vista em alguns sistemas (Free BSD), porém mais segura que o DAC. Esta maior segurança deve-se ao aspecto do sujeito não possuir mais o controle completo do acesso aos recursos por ele criado, e a política de segurança do sistema (especificado pelo administrador de segurança) determina inteiramente o acesso que deve ser concedido aos objetos pelos sujeitos. A possibilidade de classificar o sujeito e o objeto quanto ao grau de sensibilidade da informação foi um grande avanço para o controle de acesso. Esta classificação viabilizou o princípio da distribuição dos sujeitos e objetos em categorias, isto é, relacionou o acesso de sujeitos a um conjunto de categorias de objetos conforme o seu grau de confidencialidade. Sendo assim, o MAC pode ser considerado um controle de acesso rígido, sendo aplicado em organizações onde informações são classificadas como sensíveis. A sua grande desvantagem de da capacidade de classificação da política de segurança, que está atrelada diretamente a capacidade de seu criador.

O RBAC, apesar de ser uma boa opção de controle de acesso, ele não atende as questões de ambientes colaborativos como:

> a necessidade de um controle de acesso híbrido que incorpore as vantagens de se ter um amplo leque de permissões baseado em papéis via tipos de objetos e um controle refinado de determinados usuários em determinados papéis e em instâncias individuais de objetos;
>
> a necessidade de reconhecer o contexto associado com as tarefas colaborativas e a habilidade para aplicar este contexto em decisões na ativação de permissões.

O RBAC é um modelo de extrema aplicabilidade em organizações, pois o uso do conceito de papel facilita e simplifica a criação e gerenciamento da política de segurança. Junto com o conceito de papel a propriedade de hierarquia também colabora com estes aspectos. Porém, a facilidade do uso do papel e da hierarquia pode trazer problemas quanto a segurança. A herança obtida por papéis subordinados pode causar problemas de segurança, isto é, ter acesso a objetos não adequados ao sujeito. O administrador de segurança deve estar atento, em conjunto com a política de segurança especificada, quanto às questões de limitação de acesso pelos sujeitos. Mesmo sendo a característica de herança um fator positivo, ela deve ser cuidadosamente analisada de forma que acessos indesejados a objetos sejam a sujeitos.

Controles de acesso como o DRM também seguem a mesma linha de ação dos controles DAC, MAC e RBAC. Possuem os seus monitores de referências distribuídos conforme a distribuição dos produtos pelos fornecedores, mas possuem problemas de segurança. O modelo de controle de acesso DRM busca cada vez mais limitar o acesso não autorizado a objetos remotos. Porém, o DRM apresenta normalmente um grande problema: os usuários dos objetos podem fazer uso de forma indiscriminada tanto a sua distribuição quanto ao seu tempo de uso. O DRM procura garantir a segurança nos servidores provedores de produtos, mas na sua maioria não tem condições de garantir que os usuários possuam ambientes seguros ou que estes façam bom uso dos produtos. Sendo assim, estes fatores são aspectos limitadores no DRM, que devem ser trabalhados pelos seus desenvolvedores.

Com o desenvolvimento da tecnologia da informação, principalmente no comércio eletrônico, alguns aspectos adicionais tornaram-se necessários ao controle de acesso. Os sistemas de informação, que usam objetos digitais, passaram a se preocupar com os aspectos temporais de utilização de um objeto (ex: o tempo de utilização de um objeto é consumido conforme a sua utilização). Logo, a permissão de um sujeito a um objeto deve ser atualizada em tempo de execução conforme a sua utilização, até expirar totalmente, além poder revogá-la durante o uso do objeto. O UCON é um modelo que veio revolucionar e resolver diversos problemas já citados. Tem como principais vantagens o controle dinâmico do uso do objeto (nenhum outro modelo possui) via a capacidade de mutabilidade de atributos. Esta característica acaba com o problema do uso eterno do objeto, passando a estabelecer limites de tempo na sua utilização. Vale também ressaltar a capacidade de operar com os modelos DAC, MAC, RBAC e DRM, por exemplo, possibilitando uma maior versatilidade das suas funcionalidades.

Os demais modelos citados no mini-curso buscam novas abordagens não mais restritas em garantir ou não a permissão, ou mesmo preocupados com a relação sujeito, objeto e permissão. O Or-BAC baseia-se no conceito de organização, e trabalha com os aspectos de proibições, obrigações e recomendações. Usa o conceito de contexto como a sua grande arma. A sua abordagem estabelece que o papel exercido por um sujeito corresponde a alguma relação de concordância entre os sujeitos na formação da organização. Sendo assim, todos os aspectos de estabelecimento de permissões ou proibições está baseado no contexto em que o sujeito ou grupo se encontram na organização e na execução das suas tarefas. O Or-BAC consegue estabelecer um controle mais amplo dos papéis dentro da organização, mas deixa de existir um controle efetivo de cada sujeito em relação a um objeto, ocasionando possíveis problemas de gerenciamento da política de controle de acesso pelo administrador de segurança.

Os controles de acesso TMAC e TBAC, como citado no parágrafo anterior, também fogem ao escopo da permissão direta entre sujeito e objeto. Eles trazem uma grande vantagem em relação aos demais e ambos são auto-administráveis, possibilitando um maior dinamismo e rapidez nas tomadas de decisão. O gerente de segurança programa a política de controle de acesso uma única vez, e esta é aplicada conforme com a evolução das tarefas dentro do sistema.

Esta vantagem pode talvez proporcionar problemas não quanto ao desempenho, mas sim quanto ao gerenciamento de um incidente de segurança. Este problema deve-se a necessidade de se criar pontos de controle ou sistemáticas que verifiquem ou que garantam que a política de controle de acesso aplicada é adequada as necessidades da organização.

Após esta breve análise dos modelos de controle de acesso apresentados neste minicurso, pode-se observar que o controle de acesso sofreu uma evolução no decorrer do tempo, principalmente em função das novas necessidades apresentadas também pela evolução dos sistemas computacionais. Todos eles podem ser utilizados, sozinhos ou em conjunto, conforme a necessidade de segurança das organizações, mas seja cuidadoso quanto a determinação da sua política de segurança e quanto a escolha do modelo, porque mal utilizados podem reverter a sua função de segurança a uma ferramenta de ajuda a uma invasão ao seu sistema computacional.

4.8 Conclusão

O controle de acesso é um serviço de segurança e tem como função gerenciar o acesso aos objetos de seu sistema computacional. Este mini-curso apresentou as características fundamentais do controle de acesso e alguns dos seus principais modelos. Iniciou-se com os modelos mais tradicionais como DAC e MAC, apresentando as suas principais características e funcionalidades. O RBAC inclui o conceito de papel, que possibilitou uma maior simplicidade e facilidade quanto a modelagem e a administração do controle de acesso dentro da organização. A possibilidade de estabelecer o acesso aos objetos com base no papel foi uma revolução frente ao DAC e MAC. O DRM mostrou como podem ser realizados os acessos aos recursos digitais providos por um fornecedor garantindo os direitos autorais do sistema.

Fugindo aos modelos tradicionais de controle de acesso o UCON veio como o grande modelo inovador, trazendo um maior dinamismo em relação aos controles de acesso anteriores (mutabilidade de atributos, aspectos temporais). A possibilidade de incorporar as funcionalidades dos modelos como DAC, MAC, DRM e RBAC deu ao UCON um escopo maior de abrangência e hoje pode ser considerado o modelo mais completo.

Foram também apresentados os modelos desenvolvidos pela comunidade de pesquisa na área de segurança. Alguns destes modelos trabalham de forma que o administrador do sistema não atue diretamente no processo administrativo do controle de acesso. Os modelos são auto-administrados, no qual o administrador de segurança atua somente no início do processo (TMAC e TBAC). O Or-BAC traz o contexto como a sua grande virtude, e com isso possibilita limitar o acesso a determinadas situações dentro das organizações. Com base no que foi apresentado pode-se dizer que cada um dos modelos apresentados possui os seus valores e as suas aplicabilidades dentro de um sistema computacional. Não se pode dizer que eles irão resolver todos os problemas de controle de acesso aos recursos de um sistema, mas com certeza eles reduzirão sensivelmente.

4.10 Referências

Amoroso, Edward G. (1994) “Fundamentals of Computer Security Technology”, Prentice Hall PTR, Upper Saddle River, NJ.

Anderson, James P. (1972) ”Computer Security Technology Planning Study” Report ESD-TR-73-51.Electronic Systems Division.

Bechtold S. (2001) “Implications of Digital rights management, security and privacy in Digital rights management”, Proceedings of ACM - Workshop DRM p. 213 – 232.

Bell, D. E, e LaPadula, Leonard J. (1976) “Secure Computer Systems: Unified Exposition and Multics Interpretation”, MITRE Technical Report MTR-2997 Rev. 1, MITRE Corporation.

Biba, Kenneth J. (1977) “Integrity Considerations for Secure Computer Systems”, MITRE Technical Report MTR-3153, MITRE Corporation, Bedford, MA.

Bishop, M. (2003) “Computer Security Art and Sciense”, ed. Addison Wesley

Buenett, S. e Paine, S. (2002) “Criptografia e segurança”, Ed. Campus.

Camelot (2001) “Differentiating Between Access Control Terms” Network Security Library :: Auth. & Access Control.

Clark, David D. e Wilson, David R. (1987) "A Comparison of Commercial and Military Computer Security Policies", In Proceedings of the IEEE Symposium on Security and Privacy, p. 184– 194, Oakland, CA.

Cuppens, F. e Miège, A. (2003) "Administration Model for Or-BAC", Workshop on Metadata for Security (WMS).

Cuppens, F. e Miège, A. (2003) "Modelling Contexts in the Or-BAC Model", Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003), IEEE Press.

Curphey, M., Endler, D., Hau, W. e Taylor S. (2002) "A Guide to Building Secure Web Applications - Mandatory Access Control – Chapter 8". Access Control and Authorization, The Open Web Application Security Project (OWASP).

Denning, D. E. R. (1982) "Cryptography and data security", Addison-Wesley.

Departament of Defense (1985). "Trusted Computer System Evaluation Criteria", DOD 5200.28-STD.

Duncan C., Barker E., Peter D., Morrey M. e Waelde C. (2004) "Digital Rights Management", JISC DRM Study – Final Report.

El Kalam, A. A., El Baida, R., Balbiani P., Benferhat S., Cuppens F., Deswarte Y., Miège A., Saurel C. e Trouessin, G. (2003) "Organization based access control", Proceedings oh the 4th International on Policies for Distributed Systems and Networks, IEEE Press.

Ferraiolo, David F., Sandhu, Ravi S., Gavrila, S., Kuhn, D. R. e Chandramouli, R. (2001) "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol. 4, No. 3, p. 224–274.

Goguen J. A. e Mesajuer J. (1982) "Security Policies And Security Models", Proceedings of IEEE symposium on Reseach in Security and Privacy.

Harrison, Michael A. e Ruzzo, Walter L. (1976) "Protection in Operating Systems", Communications of the ACM, Vol. 19, No 8.

Harrison, Michael A. Harrison, Ruzzo, Walter L. e Ullman, Jeffrey D. (1976) "Protection in Operating Systems", Communications of the ACM.

ISO/IEC 27001 (2005)"Tecnologia da Informação – Técnicas de Segurança – Sistemas de Gerenciamento de Segurança da Informação – Necessidades", ISO/IEC.

Jagadeesan, R. e Saraswat, V. (2005) "Timed Constraint Programing: A decarative Approach to Usage Control", Principles and Practice of Declarative Programming (PPDP'05).

Jordan, Carole S., Downs D., Wagner G., LaFountain, S. e Baker, Dixie B. (1987) "A Guide to Understanding Discrentionary Access Control in Trusted Systems", National Computer Security Center.

Kaminsky, Omar (2004) "Introdução à Gestão de Direitos Digitais", www.cem.itesm.mx/verba-iuris/articulos/080203.htm.

Karp, A. H. (2006) "Authorization-Based Access Control for the Services Oriented Architecture", 4th ICCC, IEEE Press

Katzenbeisser, Adelsbach, S. e Veith, H. (2003) "Watermarking schemes provably secure against copy and ambiguity attacks", Proceedings of the 2003 ACM workshop on Digital rights management, p. 111-119.

Ku, W. e Chi, Chi-Hung (2004) "Survey on the technological aspects of Digital Rights Management", Proceeding of the 7th Information Security Conference.

Lamport, L. (1994) "Transactions on Programming Languages and Systems - The Temporal Logic of Actions", ACM, Vol. 16 Issue 3.

Lampson, Butler W. (1971) "Protection"; Proceedings of the 5th Princeton Conference on Information Sciences and Systems, Princeton, p.437.

Landwehr, Carl E. (1981) "Formal Models for Computer Security", ACM Computing Surveys, 13(3): p. 247–278.

Landwehr Carl E, (1983) "Best available technologies for computer security", IEEE Comput, p.86-100

Landwehr, Carl E. (2001) "Computer security" Publicado por Springer-Verlag.

Mackenzie, D. e Pottinger, G. (1997) "Mathematics, Technology, and Trust: Formal Verification, Computer Security, and the U.S. Military", IEEE Annals of the History of Computing, Vol. 19, nr 3.

McLean, John (1990). "The Specification and Modeling of Computer Security". IEEE Computer, 23(1): p. 9–16.

Nicomette, Vincent (1996). "La Protection dans les Systèmes à Objets Répartis". Thèse de doctorat, Institut National Polytechnique de Toulouse, France.

Osborn, S. (1997) "Mandatory Access Control and Role-Based Access Control Revisited", Proceedings RBAC97.

Russel, Deborah e Gangemi, G. T. (1991) "Computer Security Basics", Ed. O' Reilly.

Samarati, Pierangela e Capitani di Vimercati, S. (2001)"Access Control: Policies, Models, and Mechanisms", Eds. R. Focardi and R. Gorrieri : FOSAD 2000, LNCS 2171, pp. 137–196.

Sandhu, Ravi S – Role (1997) "Based Access Control", SBC97.

Sandhu, Ravi S. (1993) "Lattice-Based Access Control Models", IEEE Computer, 26(11):p.9–19.

Sandhu, Ravi S. e Park, J. e Zhang X (2004) "Attribute Mutability in Usage Control", www.list.gmu.edu/confrnc/ifip/IFIP04-mutability.pdf.

Sandhu, Ravi S. e Park, Jaehong (2004) "The $UCON_{ABC}$ Usage Control Model", ACM Transactions on Information and System Security, Vol. 0, No. 0.

Sandhu, Ravi S. e Samarati, P. (1994) "Access Control: Principles and Practice", IEEE Communications Magazine.

Sandhu, Ravi S. e Samarati, P. (1996) "Authentication, Access Control, and Audit", ACM Computing Surveys, Vol. 28, No. 1.

Snyder, L. (1981) "Theft and Conspiracy in the Take-Grant Protection Model", Journal of Computerand System Sciences, p. 333–347.

The International DOI Foundation (2004) "DOI. The Digital Object Identifier system", http://www.doi.org/about_the_doi.html

The SCO Group (2004) "UnixWare 7 Documentation – Managing system security", http://ou800doc.caldera.com/en/SEC_admin/_Access_Control.html.

Thomas, R. K. e Sandhu, Ravi S (1993) “Towards a task-based paradigm for flexible and adaptable access control in distributed applications”, Proceedings of the Second New Security Paradigms Workshop, Little Compton, Rhode Island, IEEE Press.

Thomas, R. K. e Sandhu, Ravi S. (1997) “Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management”, Proceedings of the IFIP, Workshop on Database Security.

Thomas, R.K. and Sandhu, R.S. (1994) “Conceptual Foundations for A Model of Taskbased Authorizations”, Proceedings of the IEEE Computer Security Foundations Workshop, New Hampshire, IEEE Press.

Thomas, Roshan K. (1997) “Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments”, RBAC97

Thomas, T. (1988) “A Mandatory Access Control Mechanism for the Unix File System”, IEEE Press.

Yi C., Zhi-rong, Z. e Chang-xiang, S.(2002) “Design and Implementation MAC in Security Operating System”, Proceedings of IEEE TECON 02.

Capítulo 5

Técnicas de Defesa Contra *Spam*

Danilo Michalczuk Taveira[1], Igor Monteiro Moraes[1],
Marcelo Gonçalves Rubinstein[2] e Otto Carlos Muniz Bandeira Duarte[1]

[1]Grupo de Teleinformática e Automação - GTA
COPPE/Poli - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro

[2]Programa de Pós-Graduação em Engenharia Eletrônica
Departamento de Engenharia Eletrônica e Telecomunicações - FEN
Universidade do Estado do Rio de Janeiro

Abstract

Spams, or unsolicited electronic messages, represent more than half the e-mail traffic carried nowadays in the Internet and there is no evidence which points out the reduction of sending these messages. This situation increases the operational cost of service providers and also reduces the users trust in the e-mail application. The countermeasure to spams is the adoption of a set of techniques and procedures called anti-spam systems. These systems comprise all the process to fight spams from the prevention of e-mail harvesting and the inhibition of spamming to the messages characterization and filtering. This chapter presents the motivation and the mechanisms used to send spams and the techniques used to classify and filter them. Furthermore, different anti-spam systems proposed in the literature are analyzed in detail. At last, new proposals to inhibit the act of sending spams are discussed.

Resumo

Os spams, ou mensagens eletrônicas não solicitadas, já representam mais da metade do tráfego de correio eletrônico que circula atualmente na Internet e não há indícios que apontem para uma redução do envio destas mensagens. Tal situação aumenta o custo de operação dos provedores de serviço e diminui a credibilidade dos usuários na aplicação de correio eletrônico. A contramedida aos spams é a adoção de um conjunto de técnicas e procedimentos denominados sistemas anti-spam. Estes sistemas procuram atuar em todo o ciclo do processo de combate aos spams, desde a prevenção à construção da lista de destinatários, passando pela coibição do envio, até a caracterização e a filtragem das mensagens. Neste capítulo, são apresentados a motivação e os mecanismos utilizados para enviar os spams e as técnicas usadas para classificá-los e filtrá-los. Além disso, diferentes sistemas anti-spam encontrados na literatura são caracterizados. Por fim, novas propostas para coibir o envio de spams são discutidas.

5.1. Introdução

O combate ao envio de *spams*, é um dos grandes desafios na Internet. O *spam*, de forma simplificada, é toda mensagem eletrônica enviada sem a autorização do destinatário. Devido à simplicidade do protocolo SMTP (*Simple Mail Transfer Protocol*), o correio eletrônico é a aplicação mais afetada pelos *spams*. As estatísticas mostram que os *spams* já correspondem a pelo menos dois terços de todo o tráfego de correio eletrônico transportado pelos provedores de serviço, causando prejuízos da ordem de milhões de dólares [Pfleeger e Bloom, 2005]. Algumas previsões mais pessimistas estimam que, em poucos anos, as mensagens não solicitadas serão responsáveis por 95% do tráfego de correio eletrônico na Internet [Hoanca, 2006]. No Brasil esse problema também é bastante grave. Atualmente, o país é apontado como o quinto maior receptor e também como o quarto maior gerador de *spams* do mundo [Agência Globo, 2005, Spammer-X et al., 2004].

Além de causarem enormes prejuízos aos provedores de serviço, devido ao consumo de recursos tais como banda passante, memória e processamento, os *spams* também consomem inutilmente o tempo dos destinatários e reduzem a credibilidade dos usuários na Internet. A insatisfação entre os usuários é cada vez maior tanto pela perda de tempo na recepção e leitura das mensagens quanto pela possibilidade de disseminação de vírus e de outros programas que causam a perda de dados e o comprometimento da segurança de seus computadores.

A partir da popularização da Internet, um número cada vez maior de usuários tem acesso aos serviços de correio eletrônico, mensagens instantâneas e voz sobre IP (*Voice over IP* - VoIP). Em virtude de tal fato, o envio de *spams* é visto como uma atividade lucrativa. Como os *spams* em sua maioria possuem conteúdo comercial, grande parte dos custos de divulgação do anunciante são transferidos para os provedores de serviço, que são os responsáveis pelo encaminhamento das mensagens até os destinatários. Além disso, um único *spam* pode atingir milhares de destinatários. Um estudo da America Online [Krim, 2003] conclui que apenas dois *spammers* foram responsáveis por dois bilhões de *spams* que resultaram em oito milhões de reclamações de usuários. Um *spammer* é o indivíduo responsável por gerar e/ou enviar *spams*. O envio de *spams* também é estimulado pelo retorno obtido com as mensagens enviadas. Um estudo [Cukier et al., 2006] aponta que 39% dos usuários de correio eletrônico clicam em *spams* e que 11% dos usuários já compraram algum produto anunciado por *spams*. Outro estudo mostra que taxa de resposta às malas-diretas enviadas através de mensagens eletrônicas é doze vezes superior à taxa de resposta das malas-diretas impressas [Cullen, 2002]. Devido a este sucesso, já existem *spams* em serviços de mensagens instantâneas e de voz sobre IP. Além disso, já começam a aparecer os *spams* de vídeo que prometem ter efeitos nocivos ainda mais devastadores que os das mensagens de texto e de voz.

A adoção de sistemas anti-*spam* é a principal contramedida ao envio de mensagens não solicitadas. Os sistemas anti-*spam* são compostos por técnicas e procedimentos que buscam atuar na prevenção e na coibição de todas as etapas do processo de envio de *spams*. Estes sistemas tentam evitar a coleta de endereços de correio eletrônico para construção das listas de destinatários, coibir o envio e caracterizar e filtrar as mensagens. Para tentar classificar e reduzir o número de mensagens não solicitadas, diversos mecanismos foram propostos. A idéia básica destes mecanismos é tentar classificar as

mensagens como *spams* para, então, filtrá-las. As técnicas de combate ao *spam* existentes e os mecanismos propostos na literatura podem ser classificados em três grupos: os sistemas baseados em filtragem simples, os sistemas baseados na verificação da origem e os sistemas com auto-aprendizado. Nos sistemas por filtragem simples, novos dados ou regras são inseridos de forma manual no classificador de mensagens. A principal crítica a sistemas dessa natureza é a baixa eficiência, uma vez que tais sistemas dependem de constante atualização manual. O segundo grupo corresponde aos sistemas de verificação da origem das mensagens. Tais sistemas são essenciais uma vez que o endereço de origem do remetente pode ser facilmente falsificado, dificultando o rastreamento dos *spammers*. Desta forma, o objetivo dos mecanismos baseados na verificação da origem é confirmar a autenticidade do endereço de origem e determinar se o remetente não é um programa de envio automático de mensagens. Outra classe de sistemas anti-*spam* são os sistemas com auto-aprendizado, que são capazes de aprender sozinhos com as mensagens recebidas e, portanto, aumentar a sua eficiência no combate aos *spams*.

Apesar dos esforços para reduzir e até mesmo regulamentar o envio de *spams*, não existe hoje nenhum indício que permita inferir que tal atividade diminuirá nos próximos anos. Ao contrário, os *spammers* vêm se especializando e usando técnicas cada vez mais elaboradas para burlar os sistemas anti-*spam*. Vale ressaltar que os sistemas anti-*spam* estão em constante evolução já que para cada novo mecanismo criado, novas técnicas são desenvolvidas pelos *spammers* para enganá-los e permitir a passagem das mensagens não solicitadas. Também é fato que a maioria dos usuários da Internet não tem formação técnica em computação com capacidade para gerenciar e configurar seus computadores.

O objetivo principal deste capítulo é apresentar os conceitos e as técnicas usadas para classificar e filtrar as mensagens eletrônicas não solicitadas, os *spams*. Primeiramente, na Seção 5.2, discute-se quais características uma mensagem deve apresentar para ser considerada como um *spam*. Em seguida, são abordados alguns aspectos que motivam o envio de *spams*, bem como aspectos legais e iniciativas para regulamentar e coibir o envio de mensagens não solicitadas. Na Seção 5.3, são descritas técnicas usadas pelos *spammers* para obter endereços de correio eletrônico, enviar as mensagens e para burlar os sistemas anti-*spam*. Por sua vez na Seção 5.4, o funcionamento dos sistemas anti-*spam* é detalhado. São apresentados sistemas baseados em filtragem simples, como as listas negras, os sistemas com auto-aprendizado, como os que utilizam características de padrões sociais, e os sistemas baseados na verificação de origem da mensagem, como a verificação do DNS reverso. Por fim, na Seção 5.4.4, são apresentadas as novas propostas e as direções futuras no combate e na regulamentação do envio de *spams* na Internet.

5.2. Mensagens eletrônicas não solicitadas

Definir o que é uma mensagem eletrônica não solicitada é importante tanto para os sistemas anti-*spam*, que devem classificar tais mensagens, quanto para o desenvolvimento de leis que inibam ou até regulamentem o envio de *spams*. Nesta seção, são apresentadas definições e versões sobre a origem do termo *spam*. Também são apresentados alguns tipos de *spams* e as conseqüências e prejuízos causados por essas mensagens. São abordados ainda aspectos que tornam o envio de *spams* lucrativo e quais as medidas legais estão em discussão para regulamentar essa atividade.

5.2.1. Definição e classificação de *spams*

Alguns autores consideram *spam* toda mensagem comercial não solicitada (*Unsolicited Commercial E-mail* - UCE). Essa definição não inclui, por exemplo, mensagens que contêm fraudes e tentativas de golpe e que também são não solicitadas. Outros autores definem um *spam* como uma mensagem não solicitada enviada em batelada (*Unsolicited Bulk E-mail* - UBE), pois, na maior parte das vezes, inúmeras réplicas do mesmo *spam* são enviadas. Para muitos, um *spam* é simplesmente uma mensagem não desejada por um usuário. Entretanto, esta definição de *spam* é bem geral, conflitante e também possui um caráter subjetivo. Uma mensagem que pode ser considerada como um *spam* para um determinado usuário pode não ser para outro. Portanto, a definição do que pode ser considerado um *spam* já é um primeiro desafio. Devido à dificuldade para se afirmar o que é um *spam*, as regulamentações em vigor e as propostas de lei em discussão definem um conjunto de regras baseado em características comuns encontradas em mensagens classificadas como *spams*. Dessa forma, quando uma mensagem não atende às regras definidas, ela é considerada um *spam*.

Figura 5.1. A embalagem do SPAM extraída de http://www.spam.com.

Tão controversas quanto as definições são as explicações para a origem do termo *spam*. A palavra SPAM, escrita em letras maiúsculas, é uma marca registrada pela Hormel Foods LLC [Hormel Foods, 2000]. A Hormel Foods LLC é uma empresa de alimentos e o nome SPAM pode ter surgido de uma contração das palavras "*SPiced hAM*" que batizou um dos seus produtos. O SPAM, um presunto enlatado como mostra a Figura 5.1, se tornou conhecido em 1937 durante uma campanha publicitária e em seguida pela utilização durante a segunda guerra mundial pelo exército americano. Como a carne era racionada às tropas, o SPAM era o alimento largamente consumido. Ao retornar aos EUA, todos os soldados americanos recebiam uma medalha que ficou conhecida como medalha *spam*. Como a condecoração foi recebida por diversas pessoas, a palavra *spam* ficou associada a algo comum [Holmes, 2005]. Anos mais tarde, o grupo de comédia Monty Python filmou um esquete em que um cliente entra em um restaurante e pergunta à garçonete quais são os pratos do cardápio. A garçonete, então, cita cada um dos pratos e todos contêm SPAM. Dessa forma, a palavra *spam* é repetida muitas vezes em pouco tempo. Por isso, *spam* se tornou sinônimo de algo repetitivo e sem sentido como a maioria das mensagens eletrônicas não solicitadas [Hambridge e Lunde, 1999].

O registro histórico do primeiro *spam* é de 1978. Um anúncio de uma demonstração de produtos foi enviado na Arpanet por um funcionário do departamento de vendas

da Digital Equipment Corporation (DEC), uma fabricante de computadores. Devido à limitação de espaço destinado ao endereço dos destinatários dos programas de correio eletrônico usados na época, a mensagem foi encaminhada para "apenas" 320 destinatários. Na época, o anúncio da DEC causou surpresa e gerou um debate sobre se era correto ou não utilizar o correio eletrônico para tal finalidade. O segundo caso emblemático de envio de *spams* ocorreu em março de 1994 e foi uma propaganda enviada, de forma automatizada, para seis mil grupos de discussão de um fórum da Internet. Um casal de advogados enviou uma mensagem anunciando que o prazo de inscrições na loteria de vistos de trabalho americanos estava próximo e ofereciam seus serviços a imigrantes interessados. Tal mensagem é conhecida como o *spam* do Green Card [Canter e Siegel, 1994]. O fato de uma propaganda ter sido enviada para um fórum sem nenhuma relação com o tema em discussão revoltou grande parte dos usuários. Segundo alguns relatos, foi durante o debate sobre o anúncio dos advogados que surgiu a primeira associação entre o tipo de mensagem enviada e a palavra *spam*.

Atualmente, existem diversos tipos de *spams* [Cukier et al., 2006] que vão desde simples anúncios de produtos até tentativas de golpes financeiros contra os destinatários das mensagens. Os mais comuns são os de conteúdo comercial, que representam 74% do volume total de *spams*, como mostra a Figura 5.2. As mensagens comerciais anunciam, por exemplo, a venda de medicamentos sem prescrição médica, tratamentos estéticos, oportunidades de enriquecimento rápido e sítios de conteúdo pornográfico. Embora existam *spams* enviados por empresas conhecidas e que contêm ofertas verdadeiras, grande parte das mensagens tem origem e conteúdos suspeitos.

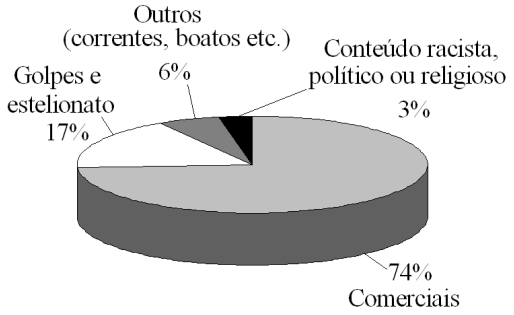
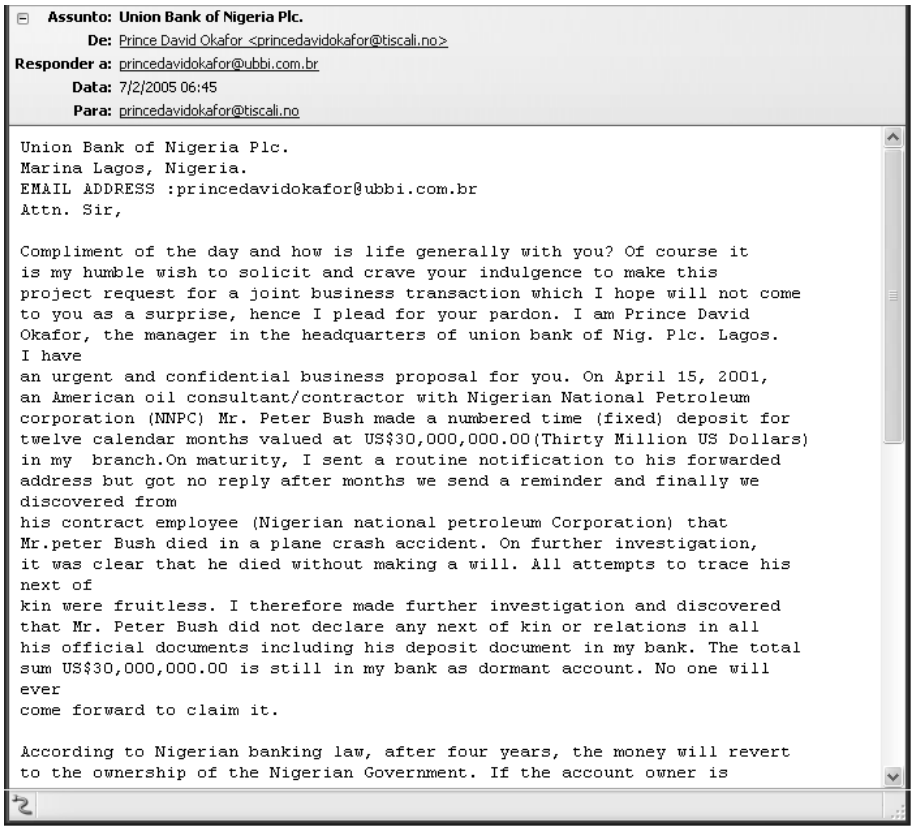


Figura 5.2. Estatísticas sobre os tipos de *spams* (adaptação) [Spammer-X et al., 2004].

Outro tipo de *spam* bastante comum é o que contém correntes ou boatos. As correntes são mensagens que prometem algum tipo de benefício, como dinheiro e saúde, a quem encaminhá-las para um determinado número de destinatários em um dado intervalo de tempo. Do contrário, se o usuário não encaminhar a mensagem sofrerá as conseqüências. Os boatos são *spams* que buscam impressionar os usuários com as falsas histórias que contêm. Essas histórias tratam, por exemplo, da busca por crianças desaparecidas, de ameaças de vírus de computador e da difamação de empresas e pessoas. O objetivo de quem envia tanto correntes quanto boatos é divulgar o conteúdo dessas mensagens para o maior número pessoas em um curto espaço de tempo e, conseqüentemente, torná-la uma lenda urbana.

Os *spams* com os efeitos mais nocivos para os destinatários são os que contêm ten-

tativas de golpes e fraudes e os que tentam disseminar vírus e outros códigos maliciosos. Toda mensagem não solicitada que contém alguma fraude ou tentativa de golpe é chamada de *scam*. Tais mensagens contêm histórias que servem de pano de fundo para que o destinatário execute uma determinada ação desejada pelo *scammer*. As histórias descritas nos *scams*, em sua maioria, tratam de ofertas de produtos que prometem resultados enganosos, oferecem oportunidades de negócios miraculosos ou até mesmo informam ao usuário que ele acabou de "ganhar" na loteria. Em troca, é solicitada alguma recompensa ao destinatário. Um dos golpes mais conhecidos da Internet, ilustrado na Figura 5.3, é a mensagem enviada por um suposto cidadão nigeriano que, por razões políticas e pessoais, está disposto a transferir uma grande quantidade de dinheiro para o destinatário. A condição para que o destinatário seja beneficiado é depositar uma pequena quantia em dinheiro como garantia em uma conta bancária a ser indicada. As mensagens com golpes semelhantes são classificadas como 419 em alusão ao número da lei nigeriana sobre a prática de fraudes.



Assunto: Union Bank of Nigeria Plc.
De: Prince David Okafor <princedavidokafor@tiscali.no>
Responder a: princedavidokafor@ubbi.com.br
Data: 7/2/2005 06:45
Para: princedavidokafor@tiscali.no

Union Bank of Nigeria Plc.
Marina Lagos, Nigeria.
EMAIL ADDRESS :princedavidokafor@ubbi.com.br
Attn. Sir,

Compliment of the day and how is life generally with you? Of course it
is my humble wish to solicit and crave your indulgence to make this
project request for a joint business transaction which I hope will not come
to you as a surprise, hence I plead for your pardon. I am Prince David
Okafor, the manager in the headquarters of union bank of Nig. Plc. Lagos.
I have
an urgent and confidential business proposal for you. On April 15, 2001,
an American oil consultant/contractor with Nigerian National Petroleum
corporation (NNPC) Mr. Peter Bush made a numbered time (fixed) deposit for
twelve calendar months valued at US$30,000,000.00(Thirty Million US Dollars)
in my branch.On maturity, I sent a routine notification to his forwarded
address but got no reply after months we send a reminder and finally we
discovered from
his contract employee (Nigerian national petroleum Corporation) that
Mr.peter Bush died in a plane crash accident. On further investigation,
it was clear that he died without making a will. All attempts to trace his
next of
kin were fruitless. I therefore made further investigation and discovered
that Mr. Peter Bush did not declare any next of kin or relations in all
his official documents including his deposit document in my bank. The total
sum US$30,000,000.00 is still in my bank as dormant account. No one will
ever
come forward to claim it.

According to Nigerian banking law, after four years, the money will revert
to the ownership of the Nigerian Government. If the account owner is

Figura 5.3. Um exemplo de um *spam* 419.

O estelionato também é um tipo de mensagem de golpe, em que é utilizada uma isca para roubar dados pessoais e/ou bancários do destinatário. Por utilizarem uma isca para "pescar" os dados do destinatário, essa atividade é chamada de *phishing* em referência ao verbo *fishing* do inglês. As iscas geralmente são mensagens com solicitações de recadastramento de dados em bancos, em administradoras de cartão de crédito e até mesmo em órgãos públicos, como a receita federal. Para que o usuário execute o que é pe-

dido, as iscas tentam se aproximar ao máximo de uma mensagem legítima, supostamente enviada por uma instituição acima de qualquer suspeita. Um exemplo de estelionato é uma mensagem enviada em nome do Banco do Brasil oferecendo ao destinatário um seguro contra fraudes. De acordo com o conteúdo da mensagem, o destinatário deve clicar em um atalho contido no *spam* para que o seguro seja ativado. O atalho o leva a uma página com um formulário que contém campos, como senha e conta corrente, a serem preenchidos. Para induzir o usuário a preencher os campos, o formulário falso é bastante semelhante ao formulário real utilizado pelo banco, como mostra a Figura 5.4. Uma vez que o destinatário preenche os campos e clica no botão de submissão, seus dados são enviados ao *spammer*.

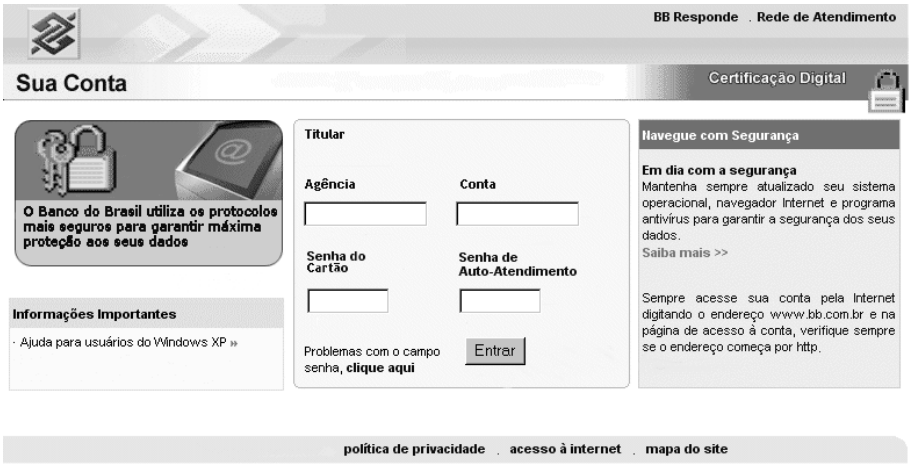
Os *spams* também são usados para difundir programas maliciosos como vírus, vermes e cavalos de tróia. Assim como nas mensagens de golpe e estelionato, os *spams* com programas maliciosos possuem algum tipo de isca para disfarçar o seu real conteúdo. Tenta-se com isso, induzir o destinatário a executar o programa enviado junto à mensagem ou fazer com que ele clique em um atalho que o leve a executar o programa hospedado em um dado sítio da Internet. Um dos objetivos da disseminação de programas maliciosos é recrutar máquinas zumbis para enviar cada vez mais *spams*, como será visto na Seção 5.3.4. Os programas maliciosos também são usados para capturar dados do destinatário. Um dos exemplos de *spam* dessa natureza é a mensagem que informa ao destinatário que ele está sendo traído, reproduzida na Figura 5.5. Para que o destinatário veja as fotos que comprovam a suposta traição conjugal, ele deve clicar no atalho indicado na mensagem. Ao clicar, o destinatário executa um cavalo de tróia que deixa o seu computador vulnerável às ações dos *spammers*. Deve ser observado na barra inferior à esquerda da Figura 5.5 que o atalho contido no *spam* é um programa executável do sistema operacional Windows.

Existem ainda mensagens com conteúdo racista, político ou religioso que correspondem a 3% do total das *spams* enviados.

5.2.2. Motivação para o envio de *spams*

Existem três razões para a proliferação dos *spams* na Internet: a facilidade para se obter endereços de potenciais consumidores, o baixo custo para enviá-los e o número de destinatários alcançados com apenas uma mensagem.

O correio eletrônico se tornou uma aplicação de grande popularidade por facilitar a comunicação entre pessoas e pelo baixíssimo custo para se enviar uma mensagem. Em virtude do sucesso dessa aplicação, a divulgação de endereços de correio eletrônico em sítios pessoais, de empresas e de instituições de ensino se tornou uma prática comum. Aproveitando-se da forma como estes endereços são divulgados eletronicamente, os *spammers* constroem a lista de destinatários de suas mensagens. Para isso, são utilizados programas, denominados robôs, que vasculham de forma automatizada os sítios da Internet em busca de endereços de correio eletrônico. Utilizando um desses programas, é possível se obter, em poucas horas e com um custo muito baixo, milhares de endereços de correio eletrônico. Mais detalhes sobre a aquisição de endereços são apresentados na Seção 5.3.2. Devido a eficácia dos *spams*, criou-se um grande comércio de listas de endereços de correio eletrônico. Uma lista com milhões de endereços pode custar de US$



(a) O formulário falso.

(b) O formulário verdadeiro.

Figura 5.4. Um exemplo de um *spam* de estelionato.

100,00 a US$ 1000,00 [Spammer-X et al., 2004]. Dessa forma, a criação e a comercialização dessas listas se tornaram uma fonte de renda atrativa, devido à sua lucratividade e ao fato de que nenhum ato ilegal está sendo cometido por quem pratica tal atividade.

Em comparação com outros meios usados na divulgação de propagandas, o correio eletrônico é o que possui o menor custo e o maior alcance geográfico. Os anúncios impressos em papel são realizados por correio convencional ou através de mensageiros e requerem gastos com a criação, a reprodução e a distribuição das mensagens. A TV e o rádio são os veículos de massa mais utilizados por anunciantes para divulgarem produ-

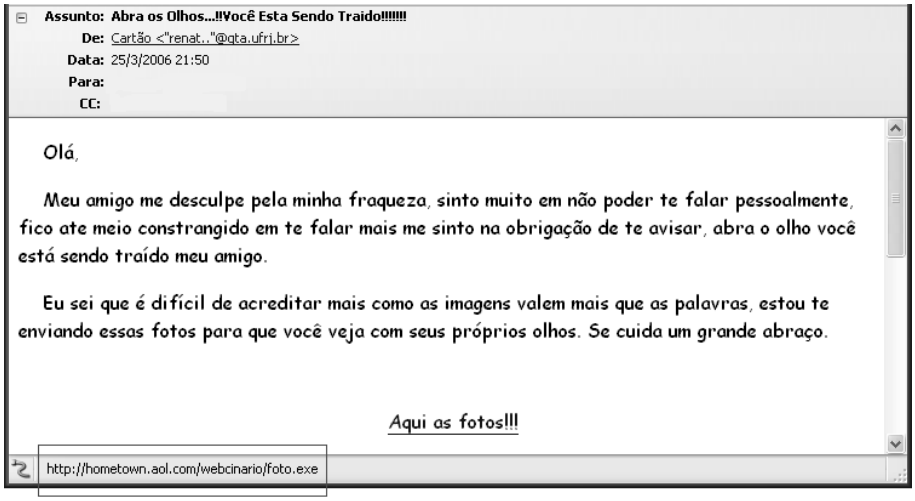


Figura 5.5. Um *spam* que induz o destinatário a executar um cavalo de tróia.

tos, serviços, ofertas e novas idéias. O impacto sonoro do rádio e o audiovisual da TV são imbatíveis. Porém, apesar do grande poder de penetração desses veículos, as propagandas na TV e no rádio estão limitadas geograficamente a regiões, estados e países e, além disso, requerem que o espectador esteja fisicamente em frente ao aparelho receptor durante a apresentação do anúncio. Deve-se considerar ainda as elevadas taxas pagas às agências e emissoras de radiodifusão pela produção e pela veiculação dos anúncios. Na busca por novos meios de divulgação, a Internet se apresenta como um veículo de grande alcance de potencias consumidores, de baixo custo e que cada vez mais possui impacto audiovisual devido às novas tecnologias. Por isso, mesmo não tendo sido criado para tal finalidade, o correio eletrônico se tornou um poderoso veículo de divulgação, em virtude da simplicidade do protocolo SMTP usado para enviar as mensagens. O protocolo SMTP foi desenvolvido pressupondo-se que as mensagens eletrônicas seriam trocadas entre remetentes e destinatários que confiavam uns nos outros e que as mensagens trocadas entre estes seriam relevantes para ambos. Essa situação foi logo explorada pelos *spammers*. Como são os responsáveis pelo encaminhamento das mensagens de correio eletrônico até os destinatários, os provedores de serviço herdam boa parte dos custos de divulgação das propagandas contidas nos *spams*. Dessa forma, a um custo reduzido, um único *spam* com uma propaganda pode alcançar milhares de destinatários, sem limites geográficos e sem grande força de trabalho. Estima-se que para enviar um milhão de *spams*, um *spammer* gaste US$ 250,00 [Pfleeger e Bloom, 2005].

O baixo custo de produção e de divulgação e o enorme número de destinatários alcançados são, sem dúvida, grandes estimulantes para o envio de *spams*. Porém, nenhuma característica motiva mais o envio de *spams* do que a efetividade dessas mensagens. Pode parecer surpreendente, mas, segundo uma pesquisa de uma empresa de consultoria em segurança da informação, 39% dos usuários de correio eletrônico entrevistados já clicaram em um atalho contido em *spams* [Cukier et al., 2006]. Foram entrevistados tanto usuários corporativos quanto domésticos. Dos usuários corporativos, 13% já compraram algum produto anunciado por um *spam*. Entre os usuários domésticos esse número é de

11%. Outro estudo [Pfleeger e Bloom, 2005] mostra que os catálogos de produtos enviados através de mensagens eletrônicas geram doze vezes mais respostas do que os catálogos impressos enviados através do correio tradicional. Além disso, em virtude do grande volume de mensagens enviadas, mesmo uma taxa de resposta baixa já representaria uma grande possibilidade de lucro para os anunciantes.

Recentemente, o envio de *spams* também tem sido motivado pela possibilidade de enriquecimento ilícito por parte dos *spammers*. Visto que muitos usuários seguem as determinações indicadas nos *spams*, o número de mensagens contendo tentativas de golpes cresce a cada dia e essas mensagens se tornam mais sofisticadas. É comum se ter notícia de pessoas que foram lesadas financeiramente após terem clicado em atalhos para sítios suspeitos contidos em mensagens eletrônicas. Sem saber, essas pessoas forneceram dados pessoais e financeiros aos *spammers* que de posse desses dados podem efetuar compras em cartões de crédito e transferências bancárias. Na maioria das vezes, quadrilhas bem organizadas estão por trás dessas fraudes, mas há alguns anos já é possível se notar a ação da polícia no combate a esse tipo de crime.

5.2.3. Prejuízos causados pelo envio de *spams*

O envio de mensagens eletrônicas não solicitadas acarreta em prejuízos para os usuários de correio eletrônico e para os provedores de serviço da Internet. Do ponto de vista dos usuários, os *spams* incomodam tanto pelo conteúdo quanto pela quantidade de mensagens recebidas. Estima-se que atualmente um usuário receba milhares de mensagens não solicitadas por ano [Pfleeger e Bloom, 2005]. Além do incômodo, os *spams* podem afetar a produtividade e a segurança dos usuários. Para cada mensagem não solicitada recebida, o usuário tem que baixá-la para o seu computador, abri-la e identificá-la como um *spam*. Isto provoca desperdício de tempo e gastos desnecessários, uma vez que o usuário paga para acessar a Internet. Em um ambiente de trabalho, os *spams* aumentam o tempo gasto pelos funcionários na leitura de mensagens eletrônicas. Estima-se que as empresas gastem cerca de US$ 1300,00 por ano com cada empregado por causa dos *spams* [Cukier et al., 2006]. Um usuário também pode sofrer com o não recebimento de mensagens legítimas em função dos *spams*. Na tentativa de reduzir o volume de mensagens não solicitadas, os provedores de serviço utilizam filtros para bloqueá-las. Como será visto em detalhes na Seção 5.4, esses filtros podem classificar mensagens legítimas como *spams* e, conseqüentemente, o usuário deixa de recebê-las. Além disso, em virtude do volume de *spams* recebidos, um usuário pode ter a sua caixa de correio completamente ocupada, o que o impede de receber novas mensagens. Porém, a ameaça mais grave aos usuários de correio eletrônico são as mensagens com tentativas de golpe. De acordo com uma pesquisa [Cukier et al., 2006], cerca de 4% dos entrevistados que utilizam correio eletrônico no trabalho e 11% dos que são usuários domésticos já perderam dinheiro com golpes enviados através de *spams*. Todos esses fatores contribuem para que a credibilidade dos usuários na aplicação de correio eletrônico e também na Internet diminua.

Para os provedores de serviço, os *spams* representam um prejuízo de bilhões de dólares. A cada um milhão de *spams* enviados, os provedores perdem US$ 2800, o que anualmente provoca um prejuízo de US$ 8,9 bilhões para as empresas americanas e US$ 2,5 bilhões para as empresas européias [Emery, 2003, Pfleeger e Bloom, 2005]. Os provedores de serviços são os responsáveis por encaminhar as mensagens eletrônicas até os

seus destinatários, inclusive as não solicitadas que atualmente correspondem a mais da metade do tráfego total de correio eletrônico. Além disso, muitos provedores oferecem gratuitamente serviços de correio eletrônico que são utilizados pelos *spammers* para enviarem suas mensagens. Assim, os provedores arcam com custos desnecessários de banda passante, memória e processamento para receber, armazenar e processar as mensagens eletrônicas não solicitadas que, ao serem recebidas, serão simplesmente movidas para a lixeira pela grande maioria dos destinatários. Portanto, além do aumento de custos com infra-estrutura, os provedores de serviço também têm que investir em soluções para combater os *spams* e, conseqüentemente, aumentam seus gastos com pessoal. É necessário aumentar o número de funcionários do serviço de atendimento ao cliente, que recebe inúmeras reclamações relativas ao recebimento de *spams*, e também formar uma equipe de manutenção para a partir das reclamações realimentar o sistema de combate aos *spams*. Para se ter uma idéia do custo de pessoal, a America Online estima que apenas dois *spammers*, que enviaram dois bilhões de *spams*, foram responsáveis por oito milhões de reclamações de usuários [Krim, 2003].

5.2.4. Legislação atual

Uma das etapas do processo para coibir o envio de mensagens não solicitadas na Internet é a definição de uma legislação sobre o tema. O estabelecimento de leis é fundamental para que se possa determinar quais mensagens podem ser consideradas como *spams* e também quais as punições devem ser aplicadas aos responsáveis pelo envio dessas mensagens.

Os Estados Unidos são pioneiros na discussão de leis anti-*spam*, provavelmente, porque cerca de 35% dos *spams* são originados no país [Commtouch, 2006]. Para tentar controlar e definir regras para o envio de *spams*, em 2004 entrou em vigor o estatuto CAN-SPAM (*Controlling the Assault of Non-Solicited Pornography and Marketing Act*). Este estatuto, válido em todo o território americano, foi definido pelo FTC (*Federal Trade Commission*), órgão responsável pela legislação anti-*spam* nos EUA, com base em leis estaduais já existentes. Para o FTC, um *spam* é qualquer mensagem eletrônica de conteúdo comercial enviada, geralmente em batelada, para um consumidor sem a requisição ou consentimento prévio desse consumidor [FTC, 2005]. Essa definição é a base das regras estabelecidas pelo CAN-SPAM. De acordo com o estatuto:

- uma mensagem eletrônica deve conter informações verdadeiras a respeito da sua origem;
- o campo de assunto da mensagem deve estar relacionado com o próprio conteúdo;
- caso seja uma propaganda, uma mensagem eletrônica deve indicar claramente o seu propósito;
- o endereço físico do remetente deve estar presente na mensagem para que o destinatário possa enviar reclamações ou denúncias;
- uma mensagem eletrônica deve fornecer ao destinatário a opção de não receber mais mensagens semelhantes do mesmo remetente.

O CAN-SPAM prevê penas criminais para os remetentes que não respeitarem as regras definidas no estatuto. As penas vão desde pagamentos de multa, para quem viola alguma das regras do estatuto, até a prisão do remetente, no caso, por exemplo, em que um *spammer* utiliza, sem autorização, o computador de terceiros para enviar as mensagens não solicitadas. As multas podem chegar a até US$ 11000. O CAN-SPAM também estipula multas para quem realizar ataques de dicionário para gerar endereços de possíveis destinatários dos *spams* e para quem coleta endereços de correio eletrônico em sítios da Internet (Seção 5.3.2). Ainda segundo o estatuto, os sítios devem conter mensagens explícitas informando sobre a proibição do uso dos endereços disponibilizados para o envio de *spams*. Também estão sujeitos a multa os indivíduos que usam programas para automatizar o registro de múltiplos endereços em sítios de serviço gratuito de correio eletrônico com o objetivo de enviar *spams*. O pagamento de multa também é previsto para quem se aproveita, sem autorização, de uma falha de configuração de servidores de correio eletrônico para enviar *spams*. O estatuto também pune com prisão quem tenta iludir e despistar destinatários e provedores de serviços sobre a verdadeira origem das mensagens não solicitadas, quem falsifica informações no cabeçalho de múltiplas mensagens e inicia a transmissão dessas mensagens, quem utiliza endereços IP falsos para enviar *spams* e quem, com o mesmo objetivo, cria contas de correio eletrônico e registra nomes de domínios usando informações falsas. De acordo com o CAN-SPAM, o termo "múltiplas" significa mais de 100 mensagens eletrônicas em um período de 24 horas, mais de 1000 mensagens em um período de 30 dias ou mais de 10000 mensagens durante o período de um ano.

A principal crítica ao estatuto CAN-SPAM é que ele mostra claramente aos *spammers* os limites nos quais eles podem atuar. Isso ocorre, pois o estatuto define legalmente o que é um *spam*. Dessa forma, é possível criar um *spam* que esteja de acordo com a regras definidas no estatuto e, conseqüentemente, dentro da lei [Spammer-X et al., 2004]. Além disso, mesmo que uma mensagem eletrônica viole as regras do estatuto CAN-SPAM, a identificação de quem a enviou é difícil, pois uma mensagem pode ser enviada com um remetente falso, como é visto na Seção 5.3.3. Outro ponto criticado no estatuto CAN-SPAM é a tentativa de balancear a manutenção da liberdade de expressão com a inibição ao envio de *spams*, o que deixou enormes brechas para que os *spammers* continuem suas atividades sem serem perturbados [Hoanca, 2006]. Também há quem diga que as medidas legais serão sempre ineficientes, pois não são ágeis o suficiente para acompanhar a evolução da tecnologia.

Por todos esses fatores, não houve redução no envio de *spams* após a entrada em vigor da legislação anti-*spam* americana. O volume de mensagens não solicitadas continua a crescer, o que comprova que as medidas legais ainda são pouco efetivas no combate aos *spams*.

No Brasil, ainda não há nenhuma legislação anti-*spam* em vigor. O que existem são projetos de lei em debate na Câmara e no Senado Federal, como o projeto 021/04 que recebeu parecer favorável na Comissão de Constituição, Justiça e Cidadania (CCJ) do Senado. Este projeto define um *spam* como sendo uma mensagem eletrônica com conteúdo comercial ou publicitário enviada a mais de 500 destinatários durante um período de 96 horas. Além disso, esta mensagem só pode ser enviada aos destinatários que autorizarem previamente o seu recebimento, deve deixar claro o seu objetivo, deve conter a verdadeira

identidade do remetente e deve possuir algum mecanismo para que o destinatário possa optar por não receber mais mensagens semelhantes do mesmo remetente. Como não há nenhuma regulamentação sobre o assunto em vigor no país, o objetivo de qualquer mensagem não solicitada que mencione estar em conformidade com as leis brasileiras é enganar os destinatários. Além do debate sobre projetos de lei, existem alguns grupos de discussão [CGI.BR, 2006, Grupo Brasil AntiSPAM, 2006b] que propõem cartilhas para informar aos usuários os perigos relativos aos *spams* e normas de conduta para os usuários de correio eletrônico. Um desses grupos é o Brasil AntiSPAM, que define um conjunto de regras para definir o que é uma mensagem não solicitada [Grupo Brasil AntiSPAM, 2006a]. Para este grupo, um *spam* é toda mensagem eletrônica com pelo menos duas das características a seguir:

- o remetente é inexistente ou possui identidade falsa;
- o destinatário não autorizou previamente o envio da mensagem;
- o destinatário não pode optar em não receber mais a mensagem;
- o assunto não condiz com o conteúdo da mensagem;
- a sigla NS (Não Solicitado) está ausente no campo de assunto de uma mensagem que não foi previamente requisitada;
- o remetente não pode ser identificado;
- uma mensagem semelhante foi recebida anteriormente em menos de dez dias apenas com os campos de remetente ou de assunto diferentes.

A proposta de estabelecer leis específicas para o combate aos *spams* no Brasil pode ter efeitos tão ineficientes quanto o estatuto CAN-SPAM nos EUA. Por isso, existe quem defenda a adaptação de artigos do código penal brasileiro [Decreto-lei n^o 2.848, 1940] para regulamentar a prática de enviar *spams* e aplicar punições aos eventuais infratores. Os artigos que tratam da usurpação (Artigo 161 do Capítulo III do Título II), de danos (Artigo 163 do Capítulo IV do Título II) e do estelionato (Artigo 171 do Capítulo VI do Título II) podem ser aplicados aos *spammers* que invadem máquinas de terceiros para enviar, sem autorização, mensagens não solicitadas e também aos que enviam mensagens com tentativas de golpe. A pena pode ir de detenção de 1 a 6 meses e multa no caso da usurpação, detenção de 6 meses a 3 anos e multa no caso de danos e, por fim, reclusão de um a cinco anos e multa no caso de estelionato. O texto destes artigos está reproduzido na íntegra no Apêndice A. Há também quem defenda a utilização dos Artigos 36 e 37 da Seção III do Capítulo V do Título I do código de defesa do consumidor [Lei n^o 8.078, 1990] que punem, respectivamente, a publicidade velada e a prática da propaganda abusiva. A reprodução destes artigos está no Apêndice B.

Como visto, as medidas legais não são suficientes para coibir a prática do envio de mensagens não solicitadas. Um dos fatores que contribuem para essa situação é a simplicidade do sistema de correio eletrônico da Internet, que possibilita o envio de mensagens

sem a confirmação da autenticidade do remetente. Dessa forma, a identificação de indivíduos que praticam atos ilícitos através de *spams* é bastante difícil e, por isso, a cada dia surgem novas técnicas para enviar mensagens não solicitadas. Algumas dessas técnicas são apresentadas na seção seguinte.

5.3. Técnicas para o envio de *spams*

O envio de *spams* engloba três fases principais. A primeira fase corresponde à obtenção de uma grande quantidade de endereços de correio eletrônico para a elaboração de uma lista de destinatários. A segunda compreende a criação da mensagem que será enviada. Por fim, é necessário um meio para enviar as mensagens. Nesta seção são apresentados o sistema de correio eletrônico da Internet, os aspectos de cada uma das fases de envio dos *spams* e como os *spams* evoluíram ao longo do tempo. Esta evolução é observada a partir de uma base de dados construída pelos autores com mais de oito mil mensagens não solicitadas recebidas nos últimos três anos. Vários exemplos de *spams* apresentados a seguir foram retirados dessa base de dados.

5.3.1. Sistema de correio eletrônico da Internet

O sistema de correio eletrônico da Internet é composto de agentes de usuário (*User Agents* - UAs), de servidores de correio ou agentes de transferência de mensagens (*Message Transfer Agents* - MTAs), de um protocolo simples de transferência de correio (*Simple Mail Transfer Protocol* - SMTP) e de protocolos de acesso a correio. A Figura 5.6 mostra os componentes do sistema de correio eletrônico e ilustra o envio de uma mensagem. Os agentes de usuário permitem que usuários leiam, respondam, reencaminhem, salvem e editem mensagens. Alguns dos principais agentes de usuário são o Outlook, o Eudora, o Thunderbird e o Mutt. Os servidores de correio armazenam as mensagens e se comunicam com outros servidores para realizar a transferência das mensagens. O protocolo SMTP transfere as mensagens entre servidores de correio e pode ser usado também nas comunicações entre o agente do usuário e o servidor de correio do usuário. O protocolo SMTP é normalmente executado em segundo plano como um *daemon* do sistema. Por último, os protocolos de acesso ao correio transferem mensagens do servidor de correio do usuário para o agente do usuário. O POP (*Post Office Protocol*), o IMAP (*Internet Message Access Protocol*) e o HTTP (*HyperText Transfer Protocol*) são exemplos destes protocolos.

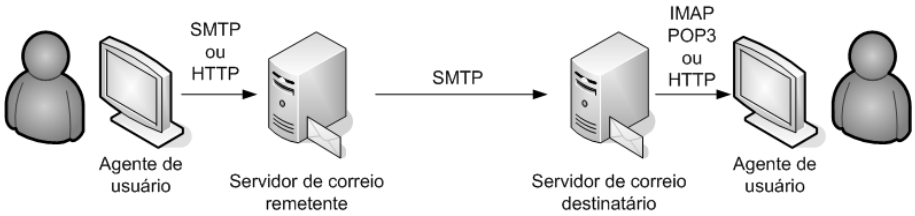


Figura 5.6. Os componentes do sistema de correio eletrônico da Internet.

Dentre os componentes de um sistema de correio eletrônico da Internet, o protocolo SMTP [Klensin, 2001] é o principal envolvido no envio de *spams*. O SMTP utiliza

o protocolo TCP (*Transmission Control Protocol*) e a porta 25 para enviar mensagens em ASCII de 7 bits[1] entre um cliente SMTP (transmissor) e um servidor SMTP (receptor).

O protocolo SMTP utiliza uma série de comandos para fazer a comunicação entre servidores de correio. Os principais comandos utilizados são o `HELO`, o `MAIL FROM`, o `RCPT TO`, o `DATA`, o `QUIT` e o `VRFY`. Os comandos, em sua maioria, são simples e auto-explicativos. O comando `VRFY` é utilizado para verificar a existência de um usuário ou de uma caixa de correio. Um exemplo de interação fictícia entre um cliente SMTP (C) e um servidor SMTP (S) é apresentado na Figura 5.7.

```
S: 220 servidor.br
C: HELO cliente.br
S: 250 Hello cliente.br, pleased to meet you
C: MAIL FROM: <usuario@cliente.br>
S: 250 usuario@cliente.br... Sender ok
C: RCPT TO: <usuario@servidor.br>
S: 250 usuario@servidor.br ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: From: usuario@cliente.br
C: To: usuario@servidor.br
C: Subject: Teste
C:
C: Teste de envio de correio.
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 servidor.br closing connection
```

Figura 5.7. Um exemplo de troca de mensagens entre um cliente e um servidor SMTP.

Como no correio convencional, um correio eletrônico é formado por um envelope e uma mensagem. O envelope encapsula a mensagem e contém todas as informações necessárias para o transporte da mensagem do remetente até o destinatário. Os servidores de correio utilizam os envelopes para o transporte das mensagens.

Uma mensagem é composta de um cabeçalho e de um corpo. Os principais campos de cabeçalho são: `From:`, `To:`, `Subject:` e `Received:`. Um exemplo de uma mensagem fictícia simplificada é apresentado na Figura 5.8. Nesse exemplo, pode-se perceber que os campos `Received:` indicam a máquina de origem da mensagem (maquina.cliente.br) e os servidores SMTP pelos quais a mensagem passou, do último para o primeiro. Além disso, o cabeçalho é separado do corpo por uma linha em branco. O corpo da mensagem só diz respeito ao destinatário e em nada importa para os servidores.

```
Received: from cliente.br by servidor.br; 16 Jul 06 10:30:01 GMT
Received: from maquina.cliente.br by cliente.br; 16 Jul 06 10:29:58 GMT
From: usuario@cliente.br
To: usuario@servidor.br
Subject: Teste

Teste de envio de correio.
```

Figura 5.8. Um exemplo de mensagem eletrônica simplificada.

[1]Para transmitir dados binários ou em ASCII de 8 bits é necessário o uso de codificação.

Um dos principais problemas do uso do SMTP para enviar correios eletrônicos é a falta de um mecanismo de autenticação para assegurar a verdadeira identidade do remetente. Isso faz com que *spammers* possam facilmente fazer uma mensagem parecer que tenha sido originada de qualquer endereço de correio eletrônico. Para evitar isso, alguns provedores de serviço exigem o uso da extensão SMTP-AUTH [Myers, 1999], que permite a verdadeira identificação do remetente do correio eletrônico. Contudo, essa extensão não garante a autenticidade do remetente do envelope ou do campo `From:` da mensagem. Um usuário autenticado pode falsificar esses valores.

5.3.2. Coleta de dados

A coleta de dados é o primeiro passo a ser realizado antes do envio de um *spam*. É nessa etapa que os *spammers* constroem as listas de destinatários de suas mensagens. Quanto maior o número de endereços válidos conseguidos nessa etapa melhor será o resultado das etapas seguintes. As principais técnicas de obtenção de endereços de correio eletrônico se baseiam na varredura de lugares onde são divulgados endereços, na invasão ou exploração de sítios para obter o cadastro dos seus usuários, na utilização de ataques de força bruta, no uso de ofertas ou concursos e na compra de listas prontas de endereços.

A forma mais simples de obtenção de endereços eletrônicos corresponde à varredura de lugares onde endereços de correio eletrônico são divulgados, como grupos de notícias (*newsgroups*), listas de distribuição (*mailing lists*), salas de bate-papo, *bulletin boards*, bases de dados livres e páginas *web*. Essa técnica surgiu no fim da década de 90 e é conhecida como coleta de endereços de correio eletrônico (*harvesting*). Inicialmente a busca era realizada manualmente, através da procura pelo símbolo "@" e da extração do endereço correspondente. Com o passar do tempo, foram introduzidos robôs automatizados para realizar essa tarefa repetitiva com maior rapidez e menor esforço. O grande número de sítios e de listas de mensagens verificados em um curto espaço de tempo fizeram com que os desenvolvedores de páginas *web* começassem a utilizar uma série de artifícios para dificultar a coleta dos endereços. Alguns trocam o símbolo "@' por "at", enquanto outros procuram colocar espaços em branco entre os caracteres do endereço de correio eletrônico ou até mesmo utilizam figuras que contêm o endereço ao invés de texto. Contudo, em páginas *web* que contém uma forma de contato na qual basta ao usuário clicar sobre um atalho com o comando `mailto:`, o trabalho de busca pelos robôs é facilitado. De modo a evitar esses problemas, pode-se substituir o comando `mailto:` por um código em JavaScript com a mesma funcionalidade. As listas de discussão também são bastante usadas pelos *spammers* que podem procurar endereços em listas abertas ou se inscrever em uma lista fechada de modo a ter acesso às mensagens trocadas. As mensagens, pertencentes a listas de discussão ou não, com diversos endereços no campo `To:` facilitam a atuação dos *spammers*. Esse fato é agravado quando uma mensagem é encaminhada (*forwarded*) e, conseqüentemente, vários endereços são expostos para outros destinatários. Nesse caso, aconselha-se a utilização do campo `Bcc:` no lugar do `To:`, principalmente no caso do encaminhamento de uma mensagem com um número grande de destinatários [Walker, 2005]. A base de dados *whois* com dados de responsáveis por domínios também é explorada por programas automatizados que capturam endereços. Para os *spammers*, um dos principais problemas destas buscas em lugares onde endereços são divulgados é a falta de correlação dos usuários listados com os produtos oferecidos pelos

spammers, exceto quando a coleta é feita em uma lista sobre um determinado assunto e a oferta está relacionada ao tema.

Os ataques ou invasões de sítios também têm sido bastante explorados com o intuito de obter endereços eletrônicos. Por mais surpreendente que possa parecer, às vezes determinadas informações como números de cartões de crédito são deixadas de lado quando o objetivo principal é a obtenção de endereços de correio eletrônico e de informações relacionadas. Essas invasões procuram explorar falhas de segurança existentes nos sistemas operacionais e nos aplicativos. Um dos principais alvos dos *hackers* são as lojas *on-line*. Essas lojas contêm bases de dados que geralmente utilizam comunicação criptografada com o servidor, mas armazenam os dados dos clientes em texto simples (não formatado) [Spammer-X et al., 2004]. A invasão de servidores específicos, como de um sítio de um clube de futebol, pode dar ao *spammer* listas de usuários identificados com o produto a ser anunciado. Programas maliciosos como cavalos de tróia, vírus e vermes também são usados para obter a lista de endereços dos usuários do computador afetado. Esses programas acessam os dados, criam as listas de endereços e enviam as informações para os *spammers*.

Outra técnica bastante utilizada pelos *spammers* consiste no uso de dicionários para realizar ataques de força bruta ou verificação de massa, através da adivinhação dos nomes dos usuários em domínios da Internet. Domínios muito populares, principalmente dos provedores gratuitos de correio eletrônico, contêm milhares de usuários que geralmente utilizam seus nomes, apelidos e nomes de seus personagens favoritos do cinema e dos quadrinhos como identificadores de contas de correio eletrônico. Certamente um dos nomes mais comuns é jose@meudominio.com.br. Em função dessa previsibilidade dos nomes, pode-se usar dicionários de nomes ou de palavras mais gerais para gerar potenciais endereços de correio eletrônico. Essa idéia pode ser explorada de modo a usar combinações aleatórias de letras e números para buscar endereços incomuns, como por exemplo abcd@meudominio.com.br. Após a geração desses endereços, as mensagens eletrônicas são enviadas e aquelas que não retornarem com erro (usuário inexistente) terão o endereço de seus destinatários validados. A busca pode ser dificultada pelo limite de tentativas mal sucedidas de descoberta de destinatários (através do comando RCPT do SMTP) configurado em muitos servidores de correio. Outra forma de realizar a verificação consiste no uso do comando VRFY do SMTP. Em função disso, atualmente diversos administradores de rede desabilitam o uso do comando VRFY. Essa técnica de força bruta é a principal responsável pelo fato de uma pessoa receber *spams* mesmo sem divulgar o seu endereço de correio eletrônico [Levine et al., 2004].

Concursos ou ofertas gratuitas de produtos também têm sido usados com o intuito de obter endereços de correio eletrônico, através de uma solicitação de dados para participação no concurso ou na oferta. Na maioria das vezes, os anúncios são falsos. Existem alguns anúncios que em letras miúdas dizem que se o usuário concordar em participar da oferta, os dados do usuário podem ser repassados para parceiros (talvez *spammers*) e o usuário pode receber ofertas comerciais. Muitos afirmam que essa idéia é utilizada por grandes empresas de software para repassar os contatos dos usuários para parceiros [Spammer-X et al., 2004].

A compra de listas prontas de endereços de correio eletrônico também é utilizada

pelos *spammers*. Algumas são negociadas diretamente entre *spammers* enquanto outras são oferecidas livremente através de *spams*. A maioria das listas contém endereços já utilizados anteriormente de forma abusiva por *spammers*. Desse modo, a maioria dos usuários pertencentes às listas já se encontra em um estado de tolerância máxima para com os *spams*, geralmente apagando-os sem ler.

Após obter os potenciais endereços dos destinatários, é necessário saber se eles são válidos. Para saber se um usuário é ativo existem diversas técnicas utilizadas pelos *spammers*. Uma delas consiste em enviar no *spam* uma referência a uma figura armazenada remotamente com um atalho contendo o endereço do destinatário, como no trecho fictício apresentado na Figura 5.9. Nesse caso, ao abrir o *spam*, a figura é automaticamente buscada do sítio, permitindo a validação do endereço do destinatário através da consulta a arquivos de registro (*log*). Por isso alguns agentes de usuário como o Thunderbird não abrem automaticamente figuras que são referenciadas em mensagens eletrônicas. Ainda assim, basta o usuário clicar no atalho para o *spammer* validar o endereço. Nesse caso, o *spammer* pode ainda montar um perfil do usuário em função do tipo de figura baixada. A opção de remoção do endereço eletrônico de uma lista também pode ser usada para validar endereços. Nesse caso, é enviado um atalho dentro do *spam* com a opção de remover o endereço da lista, conforme o trecho de um *spam* real mostrado na Figura 5.10. Assim, regras estabelecidas por legislações anti-*spam*, como o CAN-SPAM, podem ser utilizadas em benefício do próprio *spammer*. Por último, outra forma de confirmar os endereços consiste no uso do comando VRFY como citado anteriormente.

```
<img src="http://dominiospammer.com.br/figs/sexo.jpg?email=usuario@meudominio.com.br"
width=60 height=40>
```

Figura 5.9. Um trecho de um *spam* com um atalho usado para identificar o destinatário.

```
<p align=3D"center"><font face=3D"Verdana, Arial, Helvetica, sans-serif" s=
ize=3D"1">
Caso deseje remover seu nome desta lista, <u><font color=3D"#0000ff">
<a href=3D"http://dominiospammer.com.br/out/outlist.asp?email=3D=
\%25TO_EMAIL">
clique aqui</a></font></u><a href=3D"http://www.dominiospammer.com.br/sair/out/o=
utlist.asp?email=3Dusuario@meudominio.com.br">.</a></font></p>
```

Figura 5.10. Um trecho de um *spam* modificado com a opção de remoção de uma lista.

Devido à natureza da atividade de enviar *spams*, os *spammers* tentam obter além dos endereços de correio eletrônico algo que identifique que um usuário é um potencial cliente para o produto anunciado pelo *spam*. O perfil de um usuário pode ser obtido facilmente quando o endereço é coletado em listas ou através da invasão de servidores específicos como o servidor de uma loja *on-line*, de um sítio de jogos etc. [Laufer et al., 2005].

5.3.3. Formato das mensagens

As mensagens não solicitadas eram inicialmente construídas sem nenhuma preocupação com o texto ou com as palavras utilizadas. Com o avanço das técnicas anti-*spam*, a criação das mensagens eletrônicas passou a ser mais elaborada de tal forma que determinadas expressões reconhecidas pelos filtros anti-*spam* não são mais utilizadas ou são

então criados artifícios para "escondê-las" dos mecanismos anti-*spams*. Os principais formatos utilizados para o envio de *spams* são o texto simples e o texto formatado em HTML (*HyperText Markup Language*). Grande parte dos *spams* enviados atualmente ainda utiliza o texto simples. Uma das principais razões para isso, é a maior dificuldade dos filtros anti-*spam* em identificá-los. Além disso, alguns programas de correio eletrônico mais simples, que funcionam em modo texto, não aceitam outros formatos como o HTML. Contudo, os *spams* com texto simples não costumam atrair muito a atenção dos usuários, levando a uma baixa taxa de retorno.

O formato HTML é o preferido dos *spammers* atualmente. Nesse formato, o *spammer* pode colocar textos que piscam ou figuras de modo a chamar a atenção dos leitores. No entanto, o uso incorreto da linguagem HTML permite que os anti-*spams* possam identificar mais facilmente essas mensagens como *spams* [Spammer-X et al., 2004].

Diversos artifícios vêm sendo utilizados atualmente para enganar os programas anti-*spams*. Ao verificar um grande número de mensagens que continham palavras muito utilizadas nos *spams* como Viagra e Cialis, os programas anti-*spams* passaram a filtrar essas mensagens. A reação efetuada pelos *spammers* foi trocar a maneira de escrever essas palavras fazendo com que o destinatário ainda percebesse a palavra. Por exemplo, existem várias variações utilizadas para a palavra Viagra, como Vi@gr@, V-i-@-g-r-@, \/iagra etc. Conforme essas variações vão se tornando mais comuns, os mecanismos anti-*spam* passam a filtrá-las. Com o uso do HTML para compor as mensagens, torna-se mais fácil iludir os programas anti-*spam*. Uma das formas empregadas consiste no uso de caracteres invisíveis ao olho nu, por exemplo figuras de 1 x 1 pixel, ou de comentários em determinadas palavras mais comuns em *spams*. A Figura 5.11 mostra um trecho de um *spam* que usa essa técnica. Na realidade, qualquer caracter entre os sinais "<" e ">" é considerado um comentário em HTML [Costales e Flynt, 2005].

```
Having <font>prob</font><font>lems</font> maintaining a full erection or=20
one at all?<br>
<font>\/l</font><font>agra</font> works excellently for your=20
<font>pro</font><font>blem.</font>
```

Figura 5.11. Um exemplo do uso de comentários em HTML para enganar anti-*spams*.

A codificação de entidade caracter (*character-entity encoding*), na qual os caracteres são descritos por uma palavra-chave ou por um "#" seguido de um número decimal de três dígitos, também é usada para que os *spammers* disfarcem o conteúdo das páginas. A codificação URL (*Uniform Resource Locator*) também pode ser usada para os mesmos fins. A técnica *snowflaking messages* [Gregory e Simon, 2005] também é usada pelos *spammers* para enganar os programas anti-*spam*, mesmo aqueles baseados em filtros bayesianos (Seção 5.4.1.3). A técnica consiste em incluir uma grande quantidade de texto aleatório, de modo a fazer que com o programa anti-*spam* pense que o correio eletrônico seja pessoal e individual. Essa técnica é utilizada nas várias versões do *spam* apresentado na Figura 5.12. Além disso, servidores de correio reportam a listas negras em tempo real (Seção 5.4.1.1) quando recebem uma grande quantidade de mensagens iguais [Spammer-X et al., 2004]. Com a aleatoriedade, fica muito difícil identificar as várias "cópias" das mensagens. A modificação dos campos `From:` e `Received:` também

é utilizada pelos *spammers*, de modo a dificultar a obtenção de suas identidades verdadeiras e iludir os programas anti-*spams*.

```
Hi,

CIjALIS
VALjIUM
AMBjIEN
VjIAGRA

http://www.notrogores.com

in a thick fence of them all round him-that at least was the spiders
idea. Standing now in the middle of the hunting and spinning insects
Bilbo plucked up his courage and began a new song:
```

Figura 5.12. Um exemplo de um *spam* com texto aleatório para enganar anti-*spams*.

5.3.4. Envio das mensagens

Em função dos *spams* poderem conter esquemas de fraudes, estelionato, ofertas de produtos inexistentes etc., os *spammers* procuram se manter no anonimato. Os *spammers* que praticam esses atos ilícitos não enviam suas mensagens através de seus servidores, pois mesmo utilizando campos falsos de `From:` e `Received:`, um usuário com algum conhecimento do formato de mensagens eletrônicas poderia chegar ao servidor do *spammer*. Além disso, para enviar mensagens para um grande número de usuários, é necessária uma grande banda disponível. Em função disso, servidores ou máquinas alheias são utilizados para que os recursos computacionais do *spammer* não fiquem indisponíveis e também para dificultar o rastreamento do verdadeiro remetente das mensagens. Com a mesma finalidade, contas são abertas em serviços de *webmail* gratuitos. As principais formas de envio de *spams* são apresentadas a seguir.

A primeira técnica de envio de mensagens não solicitadas utilizada pelos *spammers* consiste no uso *relays* abertos de correios eletrônicos. Um *relay* aberto é um servidor de correio que está configurado para encaminhar mensagens enviadas para ele de qualquer lugar, para qualquer receptor, sem a verificação do remetente. No protocolo SMTP original, essa era a configuração padrão. De modo a solucionar esse problema, o sendmail e outros servidores de correio eletrônico passaram a não permitir o *relay* de fora da rede. Contudo, às vezes surgem algumas falhas de segurança no sendmail que permitem o uso do *relay* [Spammer-X et al., 2004].

Por ter sido amplamente utilizada, a exploração dos servidores com *relay* aberto se tornou conhecida pelos administradores de rede que passaram a corrigir as falhas na configuração desses servidores. Além disso, atualmente, o envio de *spams* usando os *relays* abertos diminuiu. Esta técnica se tornou pouco utilizada pelos *spammers*, pois testes são proativamente realizados para verificar se servidores de correio eletrônico estão agindo como *relays* abertos e se for esse o caso, colocam o servidor em uma lista negra (Seção 5.4.1.1).

Em virtude da redução do número de servidores com *relay* aberto e do uso de listas negras, os *spammers* passaram a usar outras técnicas de envio como os servidores *proxy* abertos. Um servidor *proxy* é um servidor usado dentro de uma rede que outros compu-

tadores utilizam como um *gateway* para a Internet. Os *proxies* mais comuns usados na Internet são o Wingate e o Squid. Esses *proxies* utilizam protocolos como o Socks v4 e v5. Um servidor *socks* padrão permite que qualquer "cliente" utilize o servidor para encaminhar mensagens eletrônicas para servidores SMTP. Um dos problemas do uso de um servidor *proxy* vem do fato que assim que algum servidor de correio eletrônico detecta um número grande de mensagens vindas de um mesmo servidor (*proxy*), ele pode rejeitar as mensagens e reportar o endereço IP do servidor *proxy* para uma lista negra. Em função disso, os *spammers* procuram utilizar um grande número de servidores *proxy* que ainda não estejam em alguma lista negra. Outro artifício utilizado envolve servidores em países com línguas diferentes para dificultar a comunicação entre o servidor SMTP atacado e o responsável pelo *proxy* utilizado. Como a maioria dos servidores *proxy* pertencem a usuários de modem a cabo ou de linhas de assinantes digitais (*Digital Subscriber Lines* - DSLs), recentemente um grande provedor americano de acesso via cabo começou a bloquear todo o tráfego de saída na porta 25 (SMTP) para reduzir a quantidade de máquinas inseguras que enviam *spams* [Spammer-X et al., 2004]. Outro problema em usar os servidores *proxy* tem origem no compartilhamento dos servidores entre vários *spammers*, o que acelera a detecção e a inclusão em listas negras.

Computadores também podem ser invadidos de modo a serem utilizados para o envio de *spams*. Dependendo do número de computadores comprometidos, os *spammers* podem formar redes de máquinas zumbis (*zombie networks*), também chamadas redes de robôs (*botnets*). Essas máquinas estão sob total controle dos *spammers* e são usadas para enviar suas mensagens. As redes zumbis são coordenadas por máquinas mestras que também são controladas pelos *spammers*. A função das máquinas mestras é disparar o envio de *spams*. O grande problema de usar uma rede de máquinas zumbis para enviar *spams* é o risco de uma punição severa. Nessa situação, os *spammers* estão se apropriando de bens de terceiros para utilizá-los em benefício próprio, o que pode levá-los à prisão. Apesar dos riscos, a maioria dos *spams* é enviado dessa forma. Outro método popular de envio de *spams* utiliza o seqüestro (*hijacking*) de interfaces CGI (*Common Gateway Interface*). *Scripts* de CGIs são modificados através da adição e controle de variáveis de configuração ou de campos de entrada pelo usuário. Redes sem fio também podem ser invadidas com o propósito do envio de *spams*. Por último, o protocolo de roteamento BGP (*Border Gateway Protocol*) também pode ser usado para o envio das mensagens. Essa técnica é conhecida como injeção de rota BGP ou seqüestro de sistema autônomo. O *spammer* seqüestra faixas de endereços IP válidos que não estejam sendo utilizados e invade um roteador com falhas de segurança para ser responsável por essa faixa de endereços. Depois disso, o *spammer* manipula os pedidos de atualização de rotas e anuncia aos roteadores vizinhos que o roteador invadido é a única rota para a faixa de endereços IP adquirida maliciosamente. A partir daí, o *spammer* pode começar a enviar as mensagens usando um endereço IP da faixa. Quando um endereço IP é adicionado a uma lista negra, o *spammer* passa a usar outro endereço da faixa. Essa técnica necessita um bom conhecimento de roteadores e de protocolos de roteamento, porém é uma das mais efetivas no envio de *spams*. O envio dessa forma é muito difícil de ser rastreado e bloqueado.

Contas em serviços gratuitos de correio eletrônico via *web* são utilizadas para enviar *spams* ou receber respostas de usuários interessados nas ofertas [Wiki-Spam, 2006]. Em função da grande quantidade de mensagens enviadas pelos *spammers*, várias contas

são criadas por robôs. De modo a tentar evitar a ploriferação dessas contas, vários serviços de correio eletrônico via *web* passaram a exigir a identificação de uma palavra através de um gráfico sobre um fundo difícil de ser lido (Seção 5.4.3.3). Isso não impede que seres humanos identifiquem a palavra, mas torna difícil a leitura para os robôs. Com isso os *spammers* resolveram usar seres humanos para fazer essas leituras. Uma das maneiras de fazer isso é enviar mensagens a usuários pedindo que eles entrem com a palavra do gráfico e disponibilizando acesso a material pornográfico supostamente exclusivo.

Por último, existem também empresas especializadas no envio de *spams*, voltadas principalmente para os *spammers* que não possuem conhecimento técnico suficiente para enviar as mensagens. Essas empresas podem usar servidores hospedados em países que não proíbem o envio de *spams*.

5.4. Técnicas de combate aos *spams*

O objetivo de um sistema anti-*spam* é reduzir o número de *spams* recebidos por um usuário, classificando as mensagens para, então, filtrá-las. Esses sistemas estão em constante evolução já que para cada novo sistema, tenta-se criar técnicas para enganá-lo e permitir a passagem dos *spams*. As principais propriedades de um sistema anti-*spam* são a sua taxa de falsos positivos e de falsos negativos, ou seja, a taxa de mensagens legítimas classificadas como *spams* e vice-versa. Em geral, a taxa de falsos positivos tem um valor mais importante, já que uma mensagem legítima acaba sendo filtrada, o que pode gerar grandes transtornos e atrasos no processo de comunicação. Já os falsos negativos têm um impacto menor, já que o usuário irá receber o *spam*, mas provavelmente acabará apagando-o. Outro aspecto importante de um sistema anti-*spam* é a sua interferência com o usuário, seja ela por necessidade de configuração, manutenção, atualização ou desafios que são feitos ao usuário e que devem ser respondidos. Quanto maior o nível de interação com o usuário, mais complexo e menos amigável o sistema acaba se tornando, dificultando sua adoção em grande escala. Nesta seção, os sistemas anti-*spam* atuais e novos mecanismos propostos na literatura são classificados e analisados.

5.4.1. Sistemas baseados em filtragem simples

Nesses sistemas, novos dados ou regras são inseridos de forma manual. A principal crítica a sistemas dessa natureza é a baixa eficiência, uma vez que tais sistemas dependem de constante atualização manual, tornando o processo custoso para o usuário. O alto grau de evolução das técnicas de envio de *spam* também irão fazer com que esses sistemas precisem de uma grande taxa de atualização, o que aumenta ainda mais o trabalho do usuário. Alguns exemplos desse tipo de sistema são apresentados nesta seção.

5.4.1.1. Listas negras

Os primeiros sistemas anti-*spam* a surgirem baseavam-se na utilização de listas negras, que são listas com endereços de origem ou endereços IP de remetentes que reconhecidamente são fontes de *spam*. Já nas listas brancas, são colocados endereços de pessoas ou servidores confiáveis. Dessa forma, quando uma mensagem é recebida as duas listas são analisadas. Se a presença do endereço de origem for detectada na lista

negra, a mensagem é diretamente classificada como *spam*, sem sequer ser analisada por quaisquer outros mecanismos que porventura estejam sendo utilizados. Por outro lado, caso o endereço esteja na lista branca, a mensagem é aceita diretamente. Caso o endereço não esteja em nenhuma das duas listas, a mensagem pode ser aceita ou então são utilizados outros mecanismos anti-*spam* que estejam disponíveis, para tentar classificar a mensagem.

Inicialmente, essas listas eram feitas pelos próprios usuários e cada um ficava responsável por adicionar e remover os endereços das duas listas. Isso demandava um grande esforço do usuário, pois ele tinha que separar as mensagens e determinar se o endereço deveria ser colocado na lista branca na negra ou, em caso de dúvidas, em nenhuma das duas listas. A evolução natural foi tornar o sistema centralizado, onde entidades centrais controlam a entrada e saída de endereços das listas. A primeira implementação desse tipo de sistema distribuído foi chamada de *Real-time Blackhole List* (RBL) e foi criada por Paul Vixie. Na RBL eram listados os endereços IP de servidores utilizados para enviar *spam*. Contudo, esses endereços eram adicionados manualmente. Em seguida a RBL, surgiu outra proposta chamada ORBS (*Open Relay Behavior-modification System*) cuja principal diferença para a RBL é a realização automática de testes para identificar servidores que permitem o envio de mensagens sem nenhum controle. Os servidores com essa característica são adicionados automaticamente na lista negra, bloqueando as mensagens originadas por eles. O processo de remoção dessa lista, no entanto, é realizado de forma manual, através do contato do administrador do servidor listado devia com a entidade responsável pela manutenção da lista.

A consulta a estas listas distribuídas é geralmente feita através do protocolo DNS (*Domain Name System*), fazendo com que elas sejam chamadas genericamente de DNSBLs (*Domain Name Sysytem Black Lists*). Para realizar a consulta a essas listas, o servidor verifica se o endereço IP do cliente ou de outro servidor que se conectou a ele está presente na lista DNSBL anteriormente configurada. A verificação utilizando o protocolo DNS é realizada fazendo uma consulta DNS ao endereço formado invertendo-se os bytes do endereço IP do cliente e adicionando o nome do domínio da entidade responsável pela DNSBL. Um exemplo desse processo é mostrado na Figura 5.13, onde o servidor A quer enviar uma mensagem eletrônica para o Servidor B. Nesse caso, o Servidor B verifica se o Servidor A está presente ou não na lista negra. Essa verificação é realizada através do envio de um pedido DNS para um servidor DNS que fará a consulta à lista DNSBL. Caso a resposta do pedido de DNS seja positiva, isso significa que o endereço testado está presente na lista negra. Essas listas são também um alvo freqüente de ataques de negação de serviço por parte de *spammers* para tentar neutralizar esse mecanismo de proteção anti-*spam*.

As listas negras podem ser efetivas no bloqueio de servidores utilizados por *spammers*. Estudos mostram que até 80% dos *spams* podem ser evitados por meio do uso desses mecanismos [Jung e Sit, 2004]. Essas listas, no entanto, sofrem o problema de falsos positivos, quando um endereço é incorretamente adicionado à lista. O processo de retirada da lista pode demorar, fazendo com que mensagens legítimas acabem sendo perdidas. Máquinas contaminadas por vírus também podem ser afetadas por esse sistema. O vírus pode aproveitar-se dos recursos da máquina e instalar um servidor de correio eletrônico para ser usado por *spammers*. Essas máquinas podem acabar nas listas negras, fazendo

com que as mensagens do usuário da máquina contaminada acabem sendo recusadas.

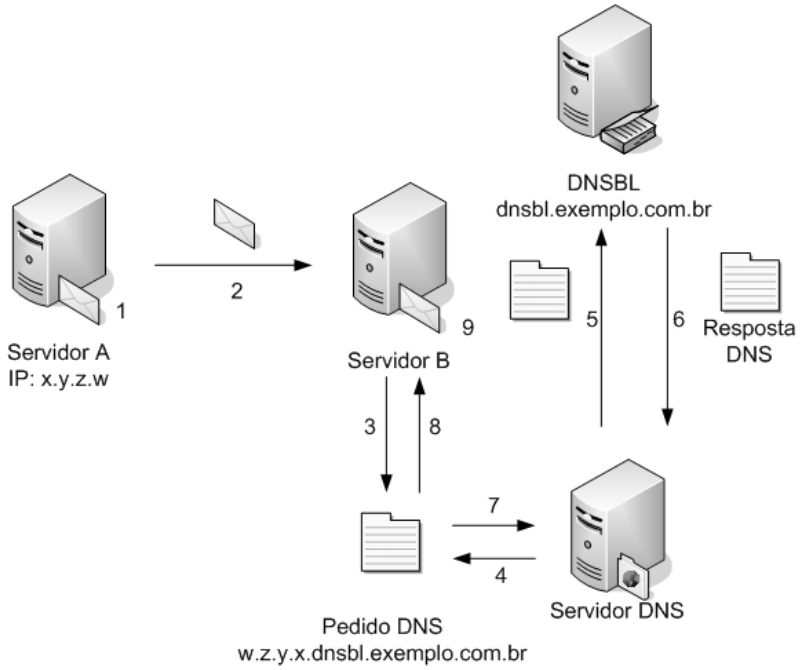


Figura 5.13. Verificação de listas negras.

5.4.1.2. Uso de pesos e regras

Nesses sistemas são utilizadas regras lógicas para a classificação das mensagens como *spam* ou não. Cada uma das regras define um teste que deve ser realizado na mensagem. Caso o resultado de um teste seja positivo, esse resultado é usado de forma ponderada para determinar a probabilidade da mensagem ser ou não *spam*. Os pesos de cada um dos testes podem ser positivos ou negativos, indicando uma maior ou uma menor probabilidade da mensagem ser *spam*. Quando uma mensagem é recebida, o mecanismo realiza todos os testes previamente definidos, somando os pesos de todos os testes cujo resultado foi positivo. Com base no valor final da soma de todos os pesos, são tomadas ações, podendo a mensagem ser encaminhada para o destinatário, marcada como sendo provavelmente um *spam* ou então descartada, o que geralmente acontece se o valor da soma de todos os pesos for muito alta.

Uma das principais e mais delicadas etapas desse mecanismo é a construção de regras, que ao mesmo tempo tem que balancear a generalidade para a adaptação a novos tipos de *spam* e a especificidade, para não classificar mensagens legítimas como *spam*. O processo de criação de regras é geralmente feito extraindo-se frases ou palavras características de mensagens de *spam*. Em seguida é feita uma análise utilizando uma grande base de mensagens manualmente classificadas como *spam* ou não, para avaliar a taxa de falsos positivos e negativos. Se alguma das taxas for alta, a regra é refinada, até que as duas taxas sejam baixas o suficiente. A determinação do peso da regra geralmente é baseada na percentagem das mensagens de *spam* onde a regra apresentou um resultado

positivo, sendo maior quanto maior for essa percentagem. Um dos principais sistemas anti-*spam* baseados nessa proposta é o *SpamAssassin* [Apache, 2006] que atualmente é um dos mecanismos mais utilizados. O *SpamAssassin*, no entanto, também utiliza outros métodos como listas negras, filtros bayesianos, verificação do DNS reverso e verificação SPF, criando regras especiais que representam esses outros tipos de teste e não são regras estáticas.

As soluções baseadas apenas em regras estáticas, no entanto, tem o inconveniente da construção das regras, tornando o processo muito complexo para usuários não experientes. Além do problema da construção das regras, elas também devem ser atualizadas freqüentemente, já que os *spams* estão em constante evolução. Esse problema da construção manual de regras levou ao surgimento de sistemas adaptativos, que são capazes de se adaptar a mudanças nas características dos *spams* ao longo do tempo como serão mostrados na Seção 5.4.2. Como a maioria dos usuários não tem conhecimento e/ou tempo suficiente para criar suas próprias regras, geralmente um grupo de pessoas fica responsável por criar as novas regras. Esse processo, no entanto pode causar problemas em situações específicas, principalmente para pessoas que trabalham com assuntos que geralmente estão presentes em mensagens de *spam*, fazendo com que as mensagens legítimas acabem sendo classificadas como *spam*. Analisando as regras [Project, 2006] utilizadas pelo sistema anti-*spam* SpamAssassin, se uma mensagem contiver nomes de certos medicamentos, a probabilidade da mensagem ser classificada como *spam* já será bem alta, o que pode causar grandes transtornos para pessoas que atuam no segmento de farmácias *on-line*.

5.4.1.3. Filtros bayesianos

Os sistemas que utilizam filtros bayesianos funcionam com base em métodos estatísticos que levam em conta a freqüência de ocorrência de determinadas palavras ou frases em mensagens classificadas ou não como *spam*. Um usuário de correio eletrônico quando recebe uma nova mensagem *spam* faz a sua leitura e acaba identificando e memorizando algumas palavras que fizeram com que ele decidisse que aquela mensagem é um *spam*. Da próxima vez que este usuário ler e identificar essas mesmas palavras em outra mensagem já terá uma suspeita maior que a nova mensagem recebida também seja uma mensagem *spam*. Dessa forma, algumas palavras-chave acabam se constituindo em características dos *spams*. Os filtros bayesianos têm por finalidade repetir este mesmo procedimento humano de identificação de *spam* realizado pelo usuário, só que de forma automatizada. Para a identificação automatizada de *spam* um filtro bayesiano deve ser construído. O primeiro passo é um processo de aprendizagem, onde o usuário identifica manualmente mensagens legítimas e mensagens *spam*, para a construção de um filtro que permita classificar mensagens futuras como legítimas ou *spams*. Uma característica importante desses filtros é que como eles são geralmente feitos com base nas informações de identificação de mensagens *spams* passadas pelo usuário, eles se adaptam ao padrão de *spams* e de mensagens legítimas recebidas pelo próprio usuário.

Para construir os classificadores usados para filtrar os *spam*, são geralmente utilizadas redes bayesianas. Uma rede bayesiana é um grafo acíclico, direcionado que re-

presenta uma distribuição de probabilidade [Pearl, 1988]. Nesse grafo, cada variável aleatória X_i é representada por um nó. Uma aresta entre dois nós indica a probabilidade de influência do nó pai para o nó filho. Além disso, cada nó X_i da rede é associado a uma tabela de probabilidade condicional, que determina a distribuição de X_i dados os valores dos seus pais. Um classificador bayesiano utiliza uma rede bayesiana onde existe um nó C representando uma das possíveis classes c_k que representam as possíveis classificações que o filtro pode realizar e vários filhos X_i, para cada uma das características testadas. Dessa forma, dado um certo conjunto de valores de X_i, pode-se calcular a probabilidade de cada classe c_k de acordo com a Equação 1, onde o termo $P(X = x|C = c_k)$ é dado pela Equação 2. Os filtros bayesianos podem permitir a dependência entre as características X_i ou não, nesse caso tornando a Equação 2 mais fácil de ser resolvida.

$$P(C = c_k|X = x) = \frac{P(X = x|C = c_k)P(C = c_k)}{P(X = x)} \tag{1}$$

$$P(X = x|C = c_k) = \prod_i P(X_i = x_i|C = c_k) \tag{2}$$

Para a utilização dos filtros bayesianos na classificação de mensagens, é necessário que, em primeiro lugar, as mensagens sejam representadas como vetores de características X_i a serem testadas, chamadas de símbolos. Esses vetores são construídos com base na separação das palavras da mensagem em vários símbolos. Além de considerar as palavras para a classificação das mensagens, podem ser utilizadas outras características que geralmente estão presentes em *spams*, como o uso exagerado de exclamações e outras informações características, como o uso de *tags* html. Após a separação da mensagem em vários símbolos, cada um deles é comparado com sua freqüência de ocorrência em mensagens *spams* e não *spams* anteriores. Se o símbolo apareceu predominantemente em mensagens *spams*, é atribuída uma alta probabilidade dele representar uma característica de *spam*. A probabilidade do símbolo é geralmente calculada como sendo a proporção entre o percentual de aparição do símbolo em mensagens *spams* e legítimas. Para economizar espaço no filtro, geralmente somente os símbolos com probabilidade muito alta e muito baixa são armazenados, já que eles podem determinar as principais características. O processo de divisão da mensagem em vários símbolos é de extrema importância no processo de classificação. Os primeiros mecanismos separavam as palavras utilizando-se sinais de pontuação, fazendo com que *spammers* explorassem essa vulnerabilidade, criando frases como *C/A/L/L/ N-O-W - I/T/S F_R_E_E*, onde as letras são separadas por caracteres que geralmente são utilizados para separar frases ou palavras. Se for utilizado um método convencional de separação em palavras, a frase acima produzirá apenas símbolos que são compostos de uma letra, dificultando a determinação se o símbolo corresponde a uma característica de *spam* ou não. Atualmente, no processo de separação de símbolos das mensagens são utilizados processos mais complexos, como a junção de símbolos formados por apenas uma letra, a agregação de símbolos, e a tentativa de reduzir novos símbolos a símbolos que tenham uma similaridade com os já conhecidos, com o objetivo de reduzir o número de símbolos diferentes que são armazenados e tratar da mesma forma símbolos que tenham grande similaridade.

Outro mecanismo que pode ser utilizado juntamente com os filtros bayesianos é

a redução bayesiana de ruído (*Bayesian Noise Reduction* - BNR) [Zdziarski, 2004]. Essa técnica tem como objetivo remover palavras fora do contexto, que muitas vezes são utilizadas por *spammers* para tentarem enganar os filtros bayesianos, conforme foi mostrado na Seção 5.3.3. Para analisar o contexto, é verificada inicialmente a probabilidade de cada símbolo encontrado na mensagem. Em seguida, é utilizada uma janela deslizante que agrupa as probabilidades de cada um dos símbolos serem característicos de *spam*. Cada um desses agrupamentos é chamado de contexto artificial. O contexto artificial é então analisado utilizando um outro filtro bayesiano, fornecendo uma probabilidade do contexto artificial criado estar presente em mensagens *spam*. Se o contexto for identificado como sendo característico de *spam*, os símbolos que fazem parte do contexto e tem uma baixa probabilidade são removidos, por se tratarem provavelmente de palavras aleatórias que não são freqüentemente encontradas em *spams* e que foram adicionadas para tentar enganar o filtro.

5.4.2. Sistemas com auto-aprendizado

Estes sistemas são capazes de aprenderem sozinhos e de aumentar sua eficiência no combate aos *spams*. Um sistema com auto-aprendizado possui como principal característica a falta de necessidade de intervenção do usuário, pois o próprio sistema aprende com o seu passado e presente, para se adaptar as mudanças do futuro. Em função de não necessitar de nenhuma intervenção do usuário, geralmente suas taxas de falsos positivos e falsos negativos podem ser maiores. Atualmente, a maior parte das pesquisas na área se concentra nesse tipo de sistema. Alguns exemplos desse tipo de sistema são apresentados a seguir.

5.4.2.1. Caracterização de tráfego de *spams*

Um dos mecanismos mais básicos de caracterização do tráfego de *spams* é monitorar a quantidade de mensagens enviadas pelos usuários. Se um determinado usuário estiver enviando uma quantidade muito grande de mensagens, provavelmente trata-se de um *spammer* e medidas podem ser tomadas, como proibir o envio de novas mensagens. Esse mecanismo, no entanto, não impede que o *spammer* utilize seu próprio servidor que não irá impor nenhum limite ao envio de mensagens. A verificação do tráfego excessivo de mensagens entre servidores é difícil de ser realizada, pois mensagens legítimas podem gerar um tráfego muito grande.

O tráfego de *spam* tem várias características que o tornam diferente do tráfego gerado por mensagens legítimas, já que os *spams* são geralmente enviados por um mecanismo automatizado que visa enviar o máximo de mensagens possíveis para o máximo de destinatários. Essa característica torna possível a identificação de padrões e anomalias no tráfego de mensagens, que podem ser utilizados para caracterizar uma mensagem ou um fluxo de mensagens como *spam*. Mecanismos baseados nesse sistema ainda não estão disponíveis comercialmente. Por enquanto essa é apenas uma área de pesquisa que busca encontrar e entender as diferenças entre os *spams* e mensagens legítimas.

Uma das características de um *spam* corresponde a sua distribuição constante ao longo dos dias de semana e horas do dia [Gomes et al., 2004]. Essa distribuição já não

ocorre para as mensagens legítimas, cujos envios se concentram nos horários entre a manhã e a tarde ao longo do dia e durante os dias da semana, excluindo-se o final de semana. Essa característica tem origem no próprio comportamento humano, já que a grande parte da população trabalha e troca mensagens mais freqüentemente durante esse período. Já para os *spammers*, quanto mais *spams* enviados, maiores serão seus lucros, fazendo com que essas mensagens sejam enviadas sempre que possível, incluindo os períodos da madrugada e os finais de semana, em que a grande maioria das pessoas não trabalha.

Outra característica temporal do tráfego de *spam* é o intervalo entre as mensagens enviadas, que geralmente é muito curto, novamente devido à forma pela qual são enviados. Para mensagens legítimas, esse tempo geralmente é maior, já que é necessário ler a mensagem, tomar alguma ação, caso necessário, e escrever uma resposta. Todo esse processo dura, em geral, um tempo maior do que o tempo de envio de dois *spams*.

A análise do número e da freqüência dos remetentes e dos destinatários das mensagens também pode fornecer outra característica das mensagens de *spam*. Um remetente que envie mensagens para um número muito grande de destinatários é candidato a ser classificado como *spammer*, já que esse comportamento não é geralmente encontrado entre usuários legítimos. Esse caso geralmente ocorre quando um determinado endereço está em uma lista utilizada por *spammers*, fazendo com que um mesmo usuário receba uma quantidade de mensagens muito grande de diferentes origens. Outra situação onde esse processo pode ocorrer é quando *spammers* utilizam ataques de dicionário, onde pessoas com nomes comuns acabam recebendo *spams* de vários *spammers* que utilizem esse método.

O tamanho das mensagens também é outro aspecto que pode ser utilizado para caracterização de *spam*, já que estudos [Gomes et al., 2004] mostram que apenas 1% das mensagens de *spam* tem mais do que 60Kb. O principal motivo para que os *spammers* utilizem mensagens pequenas e não enviem anexos nas mensagens é que quanto menor for a mensagem, menos banda ela irá utilizar, permitindo então que mais mensagens sejam enviadas em menos tempo, um dos principais objetivos dos *spammers*.

As características do comportamento humano se refletem na característica do tráfego gerado pelas mensagens legítimas, diferenciando-o do tráfego gerado por *spams*. As características do tráfego, no entanto, são muito pessoais e nem sempre podem ser generalizadas. Embora o comportamento humano na média possibilite identificar as anomalias em tráfegos de *spams*, diferenças em relação ao padrão podem ser resultados de casos especiais. Um exemplo dessa situação é uma pessoa que trabalha com grupos de pessoas em diferentes fusos horários, fazendo com que a distribuição temporal não siga o comportamento da média, tendo picos em horários não convencionais, como as madrugadas. Essas limitações poderiam ser contornadas com o aprendizado específico para cada usuário, ao invés de serem utilizadas médias de comportamento de tráfego.

5.4.2.2. Listas cinzas

A proposta dos sistemas baseados em listas cinzas é criar uma lista intermediária, entre as listas brancas e as listas negras, apresentadas na Seção 5.4.1.1. Sua eficácia baseia-se no fato de que *spams* geralmente não são retransmitidos quando o servidor noti-

fica algum erro no envio das mensagens. Esse comportamento visa aumentar a capacidade dos *spammers* enviarem mensagens com sucesso, uma vez que se um determinado endereço reportou um erro quando a mensagem foi enviada, um *spammer* não tentará enviar a mensagem uma próxima vez, para não perder tempo. Endereços para os quais o *spam* não foi entregue na primeira vez podem ser excluídos das listas de *spam*, assumindo que esses endereços não existem mais ou são inválidos.

O mecanismo de listas cinzas trabalha com duas listas, uma lista branca e uma lista cinza. Quando uma mensagem é recebida, o sistema consulta se o par destinatário/origem encontra-se na lista branca. Caso positivo, a mensagem é encaminhada para o destinatário sem restrições. Por outro lado, quando o par destinatário/origem não se encontra na lista branca, o servidor reporta um erro para o remetente da mensagem informando que houve um problema temporário no envio da mensagem. Servidores de mensagem legítimos devem seguir a RFC821 [Postel, 1982] e tentar reenviar a mensagem após um tempo, recomendado pela norma de quatro horas. Esse par destinatário/origem que teve a mensagem recusada é inserido na lista cinza juntamente com a informação temporal de quando foi adicionado. Quando o servidor do remetente tenta realizar novamente a entrega da mensagem, o servidor do destinatário irá verificar que o par destinatário/origem já foi adicionado à lista cinza. Se o par estiver na lista cinza por um período maior que um determinado valor configurado, denominado "quarentena", a mensagem é aceita e encaminhada para o destinatário. Nesse caso, a entrada que estava na lista cinza é retirada e adicionada à lista branca, fazendo com que futuras comunicações entre esse destino e origem não precisem passar novamente pelo processo de inclusão e exclusão da lista cinza. No caso da mensagem retransmitida ser recebida antes de se esgotar o período de quarentena especificado, a mensagem é recusada novamente. O período de quarentena é adotado para não permitir que um servidor retransmita logo em seguida à recepção de uma mensagem de erro temporário causado pelo processo de listas cinzas e a mensagem acabe sendo aceita. Isto dificulta e aumenta os custos de envios automáticos de mensagens.

A origem é identificada não apenas pelo endereço do remetente da mensagem, mas também pelo endereço do servidor que enviou a mensagem. O objetivo desse mecanismo é amenizar o problema da falsificação de endereços do remetente nos *spams*. Se a informação do servidor de origem não fosse guardada junto com o endereço eletrônico, um *spammer* poderia enviar *spams* de outros servidores se passando por um usuário que já se encontra na lista branca, não precisando passar pelo processo da lista cinza. Entretanto, essa característica do mecanismo, entretanto, pode causar problemas para servidores de correio eletrônico que utilizam vários servidores para enviar as mensagens. Quando a mensagem for enviada para o servidor de destino pela primeira vez, ela será enviada por algum dos servidores de correio que estejam disponíveis. Supondo o caso do servidor do destinatário não ter ainda recebido nenhuma mensagem desse remetente e servidor, sua informação será armazenada na lista cinza e a mensagem não será aceita. O servidor do remetente irá receber o erro e esperar um determinado tempo para reenviar a mensagem. Quando a mensagem for reenviada, um servidor diferente pode ser utilizado, fazendo com que o servidor do destinatário coloque essa nova combinação de origem/servidor novamente na lista cinza, atrasando ainda mais o envio da mensagem. Quanto maior o número de servidores de correio eletrônico que o provedor dispuser, mais grave esse problema se tornará, já que a cada vez a mensagem pode ser enviada por um servidor diferente.

Esse caso pode chegar até o extremo da mensagem não ser entregue devido a um número excessivo de tentativas de retransmissão.

Esse mecanismo tem a vantagem de necessitar de poucos recursos computacionais para ser executado, já que não é necessário realizar nenhum procedimento complexo. Além disso, a mensagem é recusada na primeira vez antes de ser recebida, fazendo com que os custos com armazenamento e banda passante acabem sendo menores, uma vez que se a mensagem não for retransmitida, trata-se provavelmente de um caso de *spam* e nesse caso a mensagem não é recebida ou armazenada.

A principal desvantagem desse método consiste no atraso de mensagens legítimas. Isso pode causar descontentamento dos usuários em função do tempo muito grande de espera para receber mensagens de certos remetentes, principalmente daqueles que usem provedores maiores que geralmente contam com um grande número de servidores de envio de correio eletrônico. Servidores mal configurados que tentam retransmitir mensagens muito rápido ou um pequeno número de vezes, podem gerar falsos positivos, fazendo com que mensagens legítimas não cheguem aos destinatários.

5.4.2.3. Potes de mel

A proposta de utilizar potes de mel (*honeypots*) não tem como objetivo principal atuar diretamente na filtragem de *spam*, mas servir como um mecanismo que auxilie o aprendizado sobre o comportamento de *spammers* e também para detectar padrões utilizados por *spams*. A partir deste aprendizado elaborar técnicas de combate aos *spams*.

Os potes de mel são compostos por estruturas que têm como principal finalidade enganar de alguma forma os *spammers* [Andreolini et al., 2005]. Os três principais componentes de um pote de mel para *spams* são:

- Servidores de Internet Falsos - Esse componente simula um servidor de páginas de internet falso, gerando páginas aleatórias com um grande conjunto de endereços eletrônicos criados aleatoriamente, mas cujo domínio pertence ao mesmo domínio do pote de mel. Esse procedimento faz com que robôs que automaticamente vasculham páginas da internet à procura de endereços de correio eletrônico adicionem uma grande quantidade de endereços à sua lista de envio de *spams*. Alguns destes endereços podem ser usados como recipientes de *spams* e, desta forma, aprender as características dos *spams*. O objetivo de se colocar uma enorme quantidade de endereços falsos é de diminuir a eficácia da ação do *spammer* uma vez que se diminui taxa de sucesso do *spam*.chegar a usuários legítimos. A construção das páginas geralmente é feita com base em vários textos aleatórios, de forma que torne mais difícil a detecção de um pote de mel. Uma técnica possível é criar textos com os endereços falsos com a mesma cor do fundo da página, fazendo com que uma pessoa que visite a página não visualize esses endereços, mas que os robôs os adicionem às suas listas;

- Servidores de Envio de Mensagens - Para enganar os *spammers* são criados servidores de *email* que aparentemente podem enviar mensagens sem nenhuma restrição,

passando-se por servidores mal configurados ou, por exemplo, máquinas infectadas por vírus que passam a ser servidores de correio eletrônico. Essas máquinas acabam sendo encontradas por *spammers* que fazem ataques de varredura à procura desse tipo de servidor. Assim que uma máquina desse tipo seja descoberta por *spammers*, ela pode servir para envio de *spams*. Este servidor é configurado para reportar ao remetente que a mensagem foi enviada com sucesso, quando na verdade as mensagens não são transmitidas. Dessa forma, esses servidores podem obter várias informações sobre os *spammers* como, por exemplo: a sua origem, os tipos de mensagens enviadas e características do tráfego gerado;

- Servidores para Receber Mensagens - Esses servidores são responsáveis por receber as mensagens que foram enviadas para os endereços divulgados com o intuito de serem adicionados às listas de *spam*. Já que esses endereços não são utilizados para nenhum outro propósito, todas as mensagens recebidas são *spams*, que podem ser utilizadas nos mecanismos de aprendizado de outros sistemas anti-*spam*.

Os potes de mel podem ser compostos de um ou mais componentes apresentados anteriormente. Podem ser utilizadas várias máquinas virtuais, representando cada um dos servidores, mas todas sendo executadas na mesma máquina física. É importante que se tome o cuidado de separar o pote mel do resto da rede para que o comprometimento do pote de mel não atinja toda a rede. Uma solução para esse problema consiste em utilizar uma arquitetura na qual os servidores do pote de mel estão localizados em uma zona desmilitarizada (DMZ) e separados dos servidores públicos por um roteador e um *firewall* (Figura 5.14) responsável por limitar o acesso dos servidores do pote de mel à Internet, não permitindo, por exemplo, que as mensagens enviadas por *spammers* através dos falsos servidores sejam realmente enviadas para o destino final.

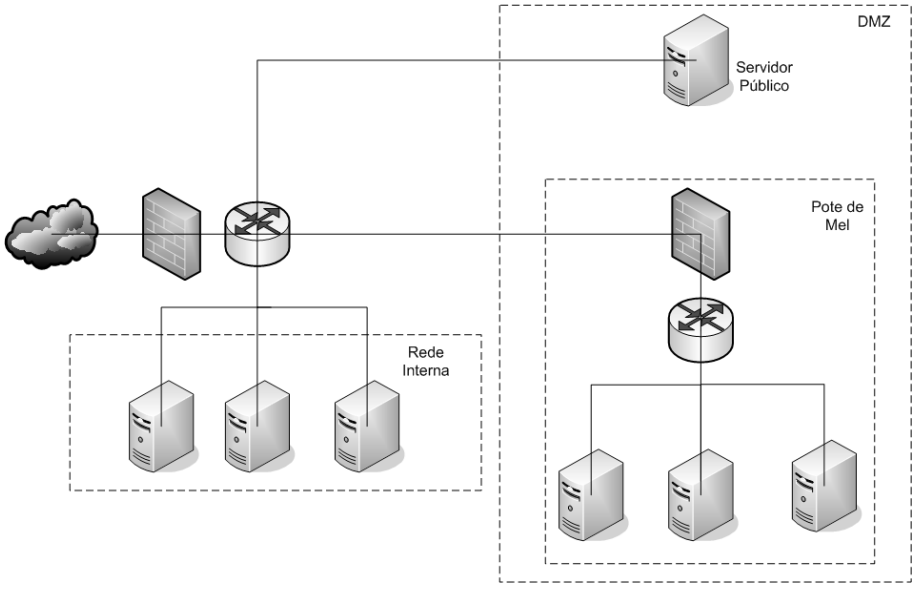
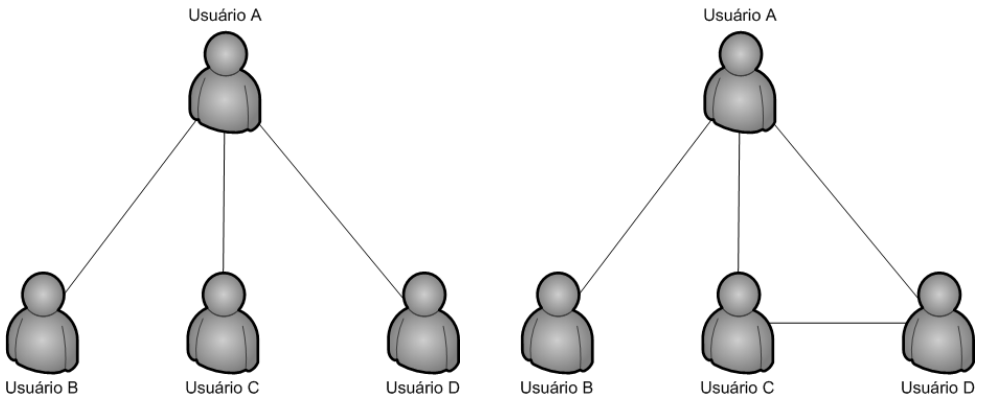


Figura 5.14. A estrutura de um pote de mel.

5.4.2.4. Padrões sociais

Uma característica importante que pode ser utilizada para a classificação das mensagens como *spam* é o uso de padrões de redes sociais. Devido ao instinto humano de formação de grupos na sociedade, a análise da rede social de determinado indivíduo pode auxiliar no processo de classificação de *spams*. Em grupos em que as pessoas se conhecem, a probabilidade de uma mensagem enviada por um indivíduo desse grupo para outro indivíduo do grupo ser um *spam* é baixa. Portanto, o objetivo desse mecanismo é determinar quais são as redes sociais formadas pelos usuários, classificando como *spam* as mensagens que não pertencem a remetentes participantes da rede social.

Uma forma de estabelecer uma rede social é construir um grafo onde cada nó representa um endereço de correio eletrônico e cada aresta representa a ocorrência de uma comunicação prévia entre os dois usuários correspondentes aos nós na extremidade da aresta. Na Figura 5.15(a) é mostrado o grafo que se forma ao se considerar uma mensagem que foi enviada do usuário A para os usuários B, C e D. Considerando que C envia uma mensagem para D, é adicionada mais uma aresta ao grafo ligando os nós C e D 5.15(b). Repetindo o processo de analisar os remetentes e destinatários das mensagens e atualizando o grafo, é criada uma representação das relações entre os diversos usuários. A construção da rede de relacionamentos pode ser feita apenas levando-se em consideração as mensagens recebidas por um usuário [Boykin e Roychowdhury, 2005] ou então utilizando as mensagens recebidas por todos os usuários do domínio [Gomes et al., 2006].



(a) Rede inicial. (b) Rede final.

Figura 5.15. Um exemplo de rede social.

Depois da rede de relacionamentos ter sido construída, ela é dividida em várias componentes, que são subgrafos do grafo completo que não apresentam arestas entre si. Porém, antes dessa divisão em componentes, o nó do qual se deseja analisar as relações é removido do grafo, já que ele acabaria ligando todas as componentes e não poderia ser feita a separação. Cada componente é então analisada para descobrir se ela corresponde a um relacionamento social ou corresponde a uma componente formada pelo processo de envio de mensagens de *spam*. A principal característica usada para classificar as compo-

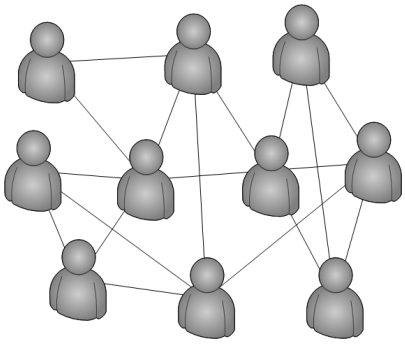
nentes nessas duas classes é o grau de agrupamento [Boykin e Roychowdhury, 2005], que é definido pela Equação 3, onde N_2 é o número de nós com mais de dois vizinhos, k_i é o número de vizinhos do nó i e E_i é o número de arestas que existem entre os k_i vizinhos do nó i. O significado do coeficiente de agrupamento é medir o nível de ligações entre usuários diferentes. Se cada nó do grafo possui uma ligação com cada um dos outros nós, pode-se observar pela Equação 3 que o grau de agrupamento é um. Já no caso de não existirem interligações entre usuários, o termo E_i será zero, fazendo com que o coeficiente de agrupamento também seja zero.

$$C = \frac{1}{N_2} \sum_i \frac{2E_i}{k_i(k_i - 1)} \tag{3}$$

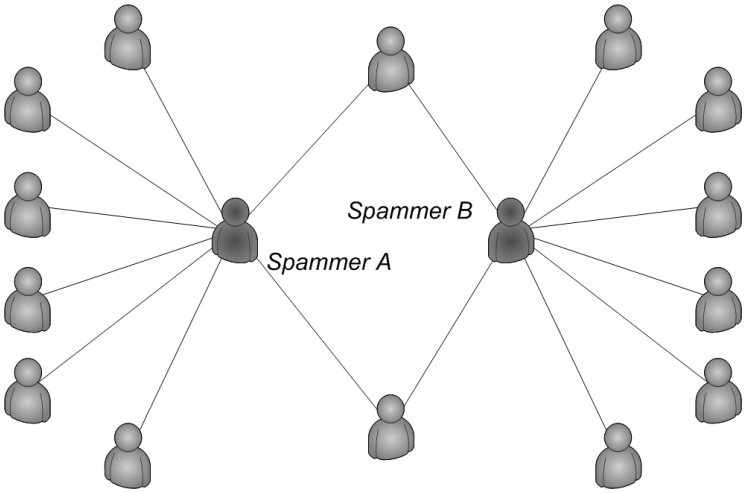
Analisando as relações sociais, geralmente um grupo de pessoas que se conhecem acaba trocando mensagens entre si, criando várias ligações entre os nós que representam cada pessoa. Já um *spammer* envia mensagens para uma grande quantidade de pessoas que não respondem ao *spammer* e, em geral, não possuem nenhuma relação entre si, fazendo com que existam várias ligações entre o nó que representa o *spammer* e os nós representantes dos usuários que receberam o *spam*, mas sem ligações entre diferentes nós. Analisando o grau de agrupamento de cada componente, se o mesmo for alto, pode-se classificar a componente como formada por uma relação social. Já no caso do coeficiente de agrupamento ter um valor baixo, a componente trata-se de uma componente anti-social que foi formada por um processo de envio de *spams*. Na Figura 5.16(a) é mostrado o exemplo de uma rede social representando a relação social entre as pessoas. Pode-se notar o grande número de arestas entre os nós. Já na Figura 5.16(b) é mostrado um exemplo de rede formada por dois *spammers* que enviam mensagens para um certo número de usuários em comum e para grupos distintos. Observando a figura, pode-se notar que não existe nenhuma ligação entre dois nós através de um terceiro, já que nenhum dos nós tem relações em comum. Já que os *spammers* enviam mensagens para um grande número de destinatários, podem acabar enviando uma mensagem para duas pessoas que possuem uma relação social, fazendo com que as duas componentes representando a rede social e a anti-social se unam em uma só componente. Essa união pode reduzir o grau de agrupamento dessa nova componente, dificultando a classificação como uma rede social ou anti-social. Para resolver esse possível problema, no processo de separação das componentes é utilizado um processo de remoção de arestas que fazem a ligação entre muitos nós, já que as arestas que ligam duas componentes que anteriormente eram separadas, são os únicos caminhos entre nós das duas componentes.

Após a classificação das componentes em redes sociais ou anti-sociais, os indivíduos pertencentes às redes sociais são colocados em uma lista branca, ajudando a reduzir a taxa de falsos positivos, que é o parâmetro mais importante de um sistema anti-*spam*. Já os *spammers* das componentes anti-sociais são colocados em uma lista negra, bloqueando suas mensagens. Uma contramedida que pode ser adotada pelos *spammers* é utilizar um endereço de origem diferente para cada *spam* enviado, impossibilitando dessa forma a formação de uma componente que pode ser classificada como sendo anti-social. É importante observar que os *spammers*, se servindo deste artifício, reduzem a eficácia das listas negras mas a rede social continua eficiente na formação das listas brancas, que reduzem os falsos positivos. Uma outra contramedida que pode ser adotada pelos *spammers* é a

obtenção da lista branca dos usuários. Neste caso, se os *spammers* enviarem os *spams* com remetentes forjados iguais aos endereços que se encontram na lista branca a eficácia do mecanismo fica totalmente anulada, pois o *spammer* passou a pertencer ao grupo social dos destinatários. Para a defesa contra tipo de ataque podem ser usados mecanismos de verificação da origem, como descrito na Seção 5.4.3. Se os *spammers* invadem e se servem de uma máquina de algum usuário pertencente a uma rede social, este mecanismo é ineficiente para identificar *spams* enviados desta máquina para outros usuários pertencentes a esta mesma rede social.



(a) Rede social.



(b) Rede anti-social.

Figura 5.16. Um exemplo de uma rede social e de uma rede anti-social.

5.4.3. Sistemas baseados na verificação da origem

O endereço de origem do remetente pode ser facilmente falsificado, dificultando o rastreamento da fonte de envio de *spams*, já que o protocolo SMTP não tem nenhum mecanismo de autenticação ou de verificação da origem. Os métodos baseados na verificação da origem geralmente buscam dois objetivos que podem ser concorrentes ou não.

Esses mecanismos podem buscar confirmar a autenticidade do endereço de origem e/ou determinar se o remetente não é um programa de envio automático de mensagens, que geralmente é utilizado por *spammers* para enviar uma grande quantidade de mensagens. Nesta seção serão apresentados os principais sistemas.

5.4.3.1. Verificação do endereço DNS reverso

A verificação do endereço de DNS reverso foi um dos primeiros mecanismos anti-*spam* baseados na verificação da origem que surgiram. O objetivo da verificação do endereço de DNS reverso é tornar mais difícil a falsificação do endereço de origem, que pode ser facilmente forjado no protocolo SMTP, como mostrado na Seção 5.3.1. O sistema DNS tem dois principais tipos de registro, os registros A e os registros PTR. Os registros A são utilizados para fazer o mapeamento entre nomes de domínio e endereços IP. Quando é feita uma consulta a um servidor de DNS para descobrir o endereço IP de um determinado domínio, o registro A é consultado. Por outro lado, pode ser feita uma requisição do registro PTR de um determinado endereço IP, para descobrir o nome de domínio registrado para ele. Esse tipo de consulta é chamado de consulta reversa, pois funciona de forma contrária ao mecanismo normal de resolução de nomes para endereços IP.

Uma maneira de verificar a autenticidade da origem é fazer uma consulta reversa ao endereço IP do servidor que está tentando enviar uma mensagem. Quando o servidor do destinatário recebe uma conexão para receber uma mensagem, ele em primeiro lugar faz uma consulta DNS reversa do endereço IP do servidor que se conectou a ele. Se o endereço IP não possuir um nome associado, a mensagem é então descartada. Caso exista um nome registrado, é feita uma consulta DNS desse nome, para verificar se o endereço IP desse nome realmente corresponde ao endereço IP original. Caso os endereços IP sejam correspondentes, diz-se que o endereço de DNS reverso é valido. Em seguida, o servidor espera pelos comandos `HELO` e `MAIL FROM` do protocolo SMTP e em seguida compara se os domínios informados nesses dois tipos de mensagem estão de acordo com o domínio que foi obtido pela verificação reversa do endereço IP. Caso os domínios sejam diferentes, a mensagem é recusada. Com essa medida, apenas os servidores cujo endereço IP tem como endereço reverso um nome que pertença ao domínio podem enviar mensagens do domínio. Na verificação do DNS reverso, muitas vezes esse segundo passo não é executado, pois podem acontecer situações onde o servidor de correio eletrônico está em um domínio diferente do qual está enviando as mensagens, como é o caso de servidores que apenas encaminham mensagens de outros domínios.

Essa verificação de DNS é feita, pois os *spammers* geralmente não configuram o endereço reverso de seus servidores, já que se forem configurados, pode-se obter o nome do domínio que o endereço IP pertence. A partir do nome do domínio, podem-se descobrir informações sobre a pessoa que registrou o domínio, aumentando as chances de rastreamento. Provedores de serviço da Internet, em geral, também não registram o endereço reverso de seus clientes, fazendo com que máquinas funcionando como zumbis também não passem no teste de verificação do DNS reverso.

O teste de DNS reverso tem uma baixa taxa de falsos negativos, já que ele elimina grande parte dos *spams* gerados por máquinas zumbis sem DNS reverso e por servidores

que não possuem registro de DNS reverso de forma proposital para dificultar o rastreamento. Em contrapartida, sua taxa de falsos positivos é geralmente bem alta, acima da média dos outros sistemas anti-*spam*, pois muitos provedores legítimos de correio eletrônico não configuram corretamente seus DNSs reversos, fazendo com que as mensagens de todos os seus usuários sejam descartadas por servidores que utilizam a verificação do DNS reverso.

5.4.3.2. *Sender Policy Framework* (SPF)

Esse mecanismo também tem como objetivo dificultar a falsificação do endereço de origem das mensagens. Seu funcionamento se baseia na publicação de informações sobre quais servidores tem permissão de enviar mensagens de um determinado domínio. Dessa forma, cada domínio fica sendo responsável por determinar quais máquinas podem enviar mensagens utilizando o domínio no endereço do remetente. Para determinar as máquinas autorizadas a enviar mensagens, o domínio determina uma série de testes que devem ser realizados por outros servidores que recebam uma mensagem com o domínio do remetente igual ao domínio em questão.

As informações sobre as máquinas autorizadas a enviar mensagens são publicadas em registros no servidor DNS do domínio, utilizando um registro de DNS de modo texto também chamado TXT. O registro do SPF publicado é composto de uma parte inicial, identificada pela seqüência "v=" que especifica a versão utilizada do SPF. Atualmente somente a versão 1 está definida, sendo utilizado o identificador spf1 para a mesma. Em seguida à informação de versão, são definidos os mecanismos que são conjuntos de testes que podem retornar um resultado positivo ou negativo. Para cada um dos mecanismos, podem ser atribuídos modificadores, que irão determinar a ação a ser tomada caso o teste feito pelo mecanismo forneça um resultado positivo. Os mecanismos são avaliados da esquerda para a direita e caso um deles forneça um resultado positivo é tomada a ação definida pelo modificador e os outros mecanismos não são testados.

Os mecanismos definidos na RFC4408 [Wong e Schlitt, 2006] para o SPF são apresentados na Tabela 5.1 e os modificadores utilizados com esses mecanismos são apresentados na Tabela 5.2.

Na Figura 5.17 é mostrado um exemplo de registro SPF permitindo que o servidor cujo endereço IP está associado no nome do domínio e os servidores de correio eletrônico do domínio enviem mensagens. Além disso, todas as regras utilizadas no domínio dominio.com.br também serão verificadas, permitindo que as máquinas autorizadas por ele também sejam aceitas. Todas as outras máquinas que não atendam às características anteriores são impedidas de enviar mensagens como sendo do domínio.

```
v=spf1 a mx include:dominio.com.br -all
```

Figura 5.17. Um exemplo de registro SPF.

O mecanismo SPF, embora não seja utilizado diretamente para filtrar *spams*, pode ajudar a reduzi-los a partir do momento que torna mais difícil o envio de mensagens com

Tabela 5.1. Mecanismos do SPF.

Mecanismo	Descrição
all	Esse teste sempre retorna verdadeiro e é geralmente utilizado como o último mecanismo a ser executado, para definir uma ação padrão caso nenhum dos mecanismos anteriores tenha retornado um resultado positivo.
include	Utilizado para incluir os mecanismos SPF definidos em um outro domínio especificado no parâmetro nome de domínio. Esse mecanismo é geralmente utilizado quando um servidor de um domínio também aceita que as suas mensagens sejam enviadas através de outros domínios, dessa forma incluindo também as restrições impostas pelo outro domínio. A sintaxe desse mecanismo é `include:<nome domínio>`.
a	Realiza uma consulta DNS ao nome do domínio para verificar se o endereço IP do servidor que está enviando a mensagem é um dos endereços IP associados ao nome do domínio.
mx	Através do protocolo DNS, consulta quais são os servidores de correio eletrônico do domínio, através dos registros MX do DNS. Após descobrir os servidores registrados de correio eletrônico, verifica se o endereço do servidor que está tentando enviar a mensagem corresponde a um dos endereços dos servidores registrados.
ptr	Esse mecanismo realiza o teste do DNS reverso, apresentado na Seção 5.4.3.1.
ip4	Define uma faixa de endereços IPv4 que estão autorizados a enviar mensagens. A sintaxe desse mecanismo é `ip4:<endereço de rede>/<máscara de sub-rede>`.
ip6	Similar ao mecanismo anterior só que utilizado para testar faixas de endereços IPv6. Sua sintaxe é a mesma do mecanismo ip4.
exists[2]	Esse mecanismo permite a utilização de macros para criar um determinado nome de domínio baseado em informações da mensagem, como o endereço IP, domínio do remetente e endereço utilizado no comando `HELO` do protocolo SMTP. Com base no nome de domínio que foi criado, verifica-se se o domínio possui um registro de DNS válido. Caso o registro DNS seja válido, o mecanismo retorna um resultado positivo.

endereços falsos, caso os provedores utilizem o SPF para determinar de forma precisa as máquinas autorizadas a enviar mensagens. O uso do SPF, entretanto, não impede que um *spammer* crie vários domínios e publique registros SPF permitindo que qualquer máquina envie mensagens utilizando como remetente esses domínios. Nessa situação os servidores que utilizem o mecanismo do SPF irão concluir que o servidor que está tentando enviar mensagem tem autorização, aceitando a mensagem. Os *spammers*, no entanto, não conseguirão falsificar endereços de provedores legítimos que usem o SPF, já que os mesmos

[2]A principal utilização desse mecanismo é em conjunto com as listas negras DNSBL, apresentadas na Seção 5.4.1.1. Com base em macros utilizando o endereço IP do remetente da mensagem, pode-se construir o nome de domínio necessário para verificar se o endereço IP que está tentando enviar a mensagem está na lista negra.

Tabela 5.2. Modificadores do SPF.

Modificador	Descrição
+	A mensagem é classificada como em conformidade com a política definida e o remetente é autorizado pelo domínio a enviar mensagens. Esse é o modificador padrão, tornando opcional sua definição explícita.
-	A mensagem não está de acordo com a política e o remetente não está autorizado a enviar mensagens como sendo do domínio.
?	O resultado da análise é neutro.
$\sim$	Define que o remetente provavelmente não está autorizado a enviar mensagens como sendo do domínio, mas não é feita uma afirmação da sua autenticidade ou não, permitindo que o servidor trate esse caso de forma diferente dos casos em que o servidor garante que o remetente está autorizado a enviar mensagens ou não.

podem definir políticas permitindo que somente seus servidores enviem mensagens como sendo do seu domínio. Embora um registro SPF devidamente configurado garanta que apenas os servidores de correio eletrônico do domínio especificados na política do SPF sejam aceitos como remetentes de mensagens do domínio, devem-se utilizar métodos para a autenticação dos clientes que enviam mensagens para os servidores do domínio. Se não for utilizado um mecanismo de autenticação, um *spammer* pode enviar uma mensagem com o endereço forjado para o servidor do domínio que ele está utilizando no endereço forjado e quando esse servidor reenviar a mensagem para o servidor do destinatário, ela será aceita já que está sendo enviada por uma máquina autorizada.

5.4.3.3. Desafio e resposta

A idéia desta proposta é limitar o envio de *spams*, usando como artifício o aumento do custo por mensagem enviada, fazendo com que esse custo seja maior que o lucro por mensagem. Estima-se que o lucro obtido por mensagem seja no mínimo de 0,01 centavos de dólar por mensagem [Detroit Free Press, 2002]. Uma das primeiras propostas nesse sentido é criar um custo inicial para a criação de uma nova conta no servidor de correio eletrônico. Esse custo pode ser imposto através de uma CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*), de desafios computacionais ou do pagamento de quantias financeiras.

Um exemplo de CAPTCHA é requisitar que o usuário digite letras que são dispostas aleatoriamente em uma imagem, como a imagem mostrada na Figura 5.18. Para uma pessoa, o processo de reconhecimento de caracteres é realizado de forma fácil. No entanto, para uma máquina reconhecer os caracteres da imagem, são necessários complexos algoritmos de processamento de imagens [Mori e Malik, 2003]. O reconhecimento pela máquina se torna ainda mais complicado quando são utilizados outros artifícios, como adicionar linhas e fundos de cores diferentes na imagem. Esse mecanismo, embora seja eficiente no sentido de diferenciar uma máquina de uma pessoa, pode causar problemas

para indivíduos com deficiência visual. Para resolver esse problema podem ser utilizados testes baseados em sons.

Figura 5.18. Um exemplo de CAPTCHA.

Desafios computacionais são procedimentos ou algoritimos com um alto custo computacional. Eles são projetados de tal forma que um máquina gaste um tempo grande para sua resolução mas que possa ser resolvido precisamente, diferente das CAPTCHAs que são projetadas para não serem resolvidas por uma máquina. Como o poder de processamento das máquinas é em geral subutilizado, esses desafios computacionais podem ser resolvidos em segundo plano, fazendo com que o usuário praticamente não perceba que eles estão sendo resolvidos. A principal característica desses desafios é que eles devem ter um custo computacional elevado para sua resolução, mas um custo baixo para verificação. Dessa forma, o usuário irá gastar um tempo grande para resolvê-lo, mas o servidor pode verificar facilmente a resposta. Uma proposta de desafio computacional que atende essas características é um desafio onde deve-se encontrar uma determinada seqüência que adicionada ao começo do cabeçalho da mensagem faça com que o seu *hash* tenha um determinado número de zeros nos bits mais significativos. Para resolver esse desafio, são necessárias tentativas de um grande número de seqüências e, para cada uma, que seja calculado o *hash* da mensagem. Já para a verificação do resultado, só é necessário adicionar a seqüência informada ao cabeçalho e calcular o seu *hash*, para verificar se os bits iniciais realmente são zero. O número de tentativas para resolver o desafio vai aumentar conforme a restrição do número de bits iniciais que devem ser iguais a zero, de forma que ajustando esse valor, pode-se definir em média em quanto tempo o desafio será realizado.

Os dois últimos tipos de custo apresentados podem ser convertidos em custos financeiros, levando-se em conta que é necessária certa quantia financeira para manter um computador e uma pessoa e/ou uma pessoa que trabalhe resolvendo as CAPTCHAs. Para resolver as CAPTCHAs, um *spammer* terá que passar parte do seu tempo resolvendo os desafios, ou então contratar uma pessoa para resolvê-los. Nos dois casos, o tempo perdido ou o salário pago irão representar um custo financeiro, estimado da ordem de 2 centavos de dólar por cada teste que tenha que ser resolvido. Os desafios computacionais também representam um custo para o *spammer*, já que para resolvê-los, o *spammer* precisará utilizar a capacidade de processamento de um ou mais computadores, o que gera custos de aquisição e manutenção.

A proposta de realizar a cobrança de um determinado custo apenas inicialmente é a mais utilizada atualmente. Na maioria dos provedores, para um usuário criar uma nova conta ele precisa resolver uma CAPTCHA. Depois do desafio ter sido resolvido, ele não precisa resolver nenhum desafio e passa a ser limitado apenas pelo número de mensagens diárias que pode enviar. Goodman [Goodman e Rounthwaite, 2004] faz uma análise desse método levando-se em consideração o custo inicial (C), a probabilidade de uma pessoa que recebeu um *spam* notificar o provedor de correio eletrônico que originou

o mesmo (p), o número médio de dias entre o recebimento do *spam* e a reclamação (L) e, por fim, o limite diário de mensagens que podem ser enviadas (D). Nos primeiros L dias, o *spammer* conseguirá enviar D mensagens todos os dias, que é o limite máximo de mensagens, já que mesmo se a primeira pessoa que recebeu uma mensagem sua faça uma reclamação, ela só será feita, em média, após os L dias. Dessa forma, durante esse tempo, ele poderá continuar enviando *spams*. A partir do dia $L+1$, a probabilidade de reclamação para cada dia, será dada pela probabilidade de qualquer uma das D mensagens enviadas serem reportadas como *spam*. Como a probabilidade de cada mensagem não ser reportada é $(1-p)$, a probabilidade de nenhuma mensagem ser reportada no dia será $(1-p)^D$. Por fim, a probabilidade q de alguma mensagem ser reportada a cada dia será justamente o complemento da probabilidade de nenhuma mensagem ser reportada, o que leva a $q = 1-(1-p)^D$. Como nos primeiros L dias o *spammer* poderá enviar D mensagens diariamente e depois disso poderá continuar enviando em média por $1/q$ dias, o número total de mensagens que serão enviadas será $LD + D/q$. Para valores pequenos de D, q será aproximadamente pD, então o número de mensagens aproximado será $LD + 1/p$. A probabilidade de reclamação é, em geral, bem pequena, da ordem de $1/1000$. Já o termo L tem ordem de grandeza de dias e D de dezenas de mensagens, fazendo com que o termo $1/p$ seja maior que o termo LD. Esse fato leva a aproximação final de $1/p$ mensagens que podem ser enviadas, não tendo o parâmetro D grande importância. Como somente foi pago um custo C inicialmente e foram enviadas aproximadamente $1/p$ mensagens, o custo por mensagem será de C/p. Supondo o custo de dois centavos de dólar do desafio, e a probabilidade p sendo $1/1000$, o custo por mensagem será de 0,002 centavos, bem abaixo do lucro por mensagem. Essa é a razão pela qual esse método de cobrança inicial não conseguiu deter os *spammers*.

Uma evolução natural desse método seria impor um custo em cada mensagem. Essa abordagem, no entanto, afetaria de forma negativa os usuários legítimos, que poderiam acabar desistindo de utilizar o serviço. Como uma possível solução, Goodman propõe a cobrança de custos a cada *n* mensagens [Goodman e Rounthwaite, 2004], só que essa cobrança só é realizada *k* vezes, depois disso o usuário só é limitado pelo número de mensagens por dia que podem ser enviadas. A princípio, essa proposta não parece ser boa, mas os resultados mostram que o custo por mensagem utilizando-se o esquema de cobrança apenas nas *k* primeiras vezes tem um resultado bastante similar ao obtido caso a cobrança fosse feita indefinidamente, mostrando que esse método não diminui de forma considerável o custo para o *spammer*, mas é vantajoso para o usuário, que não tem que pagar para sempre os custos.

Outro sistema baseado em desafios e repostas que é utilizado atualmente funciona de forma similar às listas cinzas, apresentadas na Seção 5.4.2.2. Diferentemente do método de cobrança de um determinado custo por mensagem ou por grupo de mensagens, nesses sistemas os usuários precisam resolver um desafio que é enviado pelo servidor do destinatário na primeira vez que enviarem uma mensagem ao destinatário. O servidor, ao receber uma mensagem para um dado par destinatário/origem que ainda não foi confirmado, guarda a mensagem e envia um desafio para o remetente da mensagem, geralmente uma CAPTCHA. Caso o remetente responda o desafio, a mensagem que tinha sido guardada é entregue ao destinatário e nas próximas vezes que o par se comunicar não será mais necessária essa verificação. Como os *spammers* utilizam mecanismos automatiza-

dos de envio de mensagens com endereços de origem forjados, acabarão não recebendo as mensagens contendo os desafios e, por conseqüência, suas mensagens não chegaram ao destino. Devido a essas características, esse método é bastante eficaz em relação à taxa de falsos negativos. Entretanto, usuários legítimos podem não responder aos desafios enviados. Dessa forma, mensagens legítimas acabarão sendo descartadas, gerando uma alta taxa de falsos positivos.

5.4.4. Perspectivas futuras

Não existe hoje nenhum indício que permita inferir que a atividade de enviar *spams* diminuirá nos próximos anos. Ao contrário, os *spammers* vêm se especializando e usando técnicas cada vez mais elaboradas para burlar os sistemas anti-*spam*. Vale ressaltar que os sistemas anti-*spam* estão em constante evolução já que para cada novo mecanismo criado novas técnicas são desenvolvidas pelos *spammers* para enganá-los e permitir a passagem das mensagens não solicitadas.

Uma nova forma de mensagem não solicitada que está surgindo é o *spam* através de serviços de voz sobre IP (VoIP) [MacIntosh e Vinokurov, 2005]. Os *spams* de VoIP, também chamados de SPITs (*Spam over Internet Telephony*), consistem de mensagens, em sua maioria de conteúdo publicitário, enviadas em difusão através de serviços de telefonia IP. Espera-se que, em um curto espaço de tempo, o volume de *spams* dessa natureza cresça em virtude do aumento do número de usuários dos sistemas de telefonia sobre IP. Assim como as mensagens não solicitadas de texto, os *spams* de VoIP têm como atrativo para o *spammer* um custo bem menor do que o custo do envio de mensagens através da rede telefônica convencional. O telemarketing e as gravações publicitárias são formas de *spam* usando a rede telefônica convencional. Pela própria característica da Internet, que utiliza a técnica de comutação de pacotes, um *spammer* pode enviar simultaneamente um grande número de mensagens utilizando apenas um acesso à Internet. Por outro lado, na rede convencional de telefonia isto seria inviável, pois para se enviar várias mensagens de telemarketing e gravações publicitárias ao mesmo tempo seriam necessárias diversas linhas telefônicas. Para combater os *spams* de VoIP, a maioria dos mecanismos anti-*spam* deve ser modificada, já que muitos desses mecanismos se baseiam no conteúdo do texto da mensagem para filtrá-las. A análise por conteúdo se torna muito difícil nos *spams* de VoIP, pois exige a execução de algoritmos de reconhecimento de voz. Atualmente, esses algoritmos demandam um alto custo computacional e não são muito eficientes. Além disso, ao contrário de uma mensagem de texto que é recebida por um servidor e depois encaminhada ao destinatário para que ele a leia quando quiser, uma chamada telefônica ocorre em tempo real, ou seja, é necessário que o usuário atenda a chamada para que ela se inicie. Sendo assim, a análise do conteúdo de um *spam* de VoIP tem que ser feita no momento da comunicação, o que torna o mecanismo ineficiente, uma vez que o usuário já atendeu à ligação e escutou um determinado trecho da propaganda. Por esse fato, os *spams* de VoIP tendem a se tornar ainda mais incômodos do que os *spams* de texto para os usuários, que precisam parar suas atividades para atender a chamada e só depois descobrir que era uma mensagem não solicitada.

As mensagens não solicitadas em forma de vídeo também estão surgindo e assim como os *spams* de VoIP são difíceis de ser classificadas, já que para analisar o conteúdo das mensagens de vídeo são necessárias técnicas de processamento e reconhecimento de

padrões de imagem. Os *spams* de vídeo estão começando a aparecer em sítios da Internet que, sem a permissão do usuário, carregam e mostram um determinado vídeo, que na maioria das vezes possui um conteúdo publicitário. Dessa forma, durante a exibição do vídeo, a atenção do usuário é desviada para o anúncio, devido aos sons e aos movimentos. A identificação dos *spams* de vídeo é complexa, pois em muitas ocasiões um sítio contém um vídeo que é do interesse do usuário e, portanto, não deve ser considerado como um *spam*.

Outra forma indireta de *spam* corresponde à manipulação do resultado de mecanismos de busca na Internet [Gyongyi e Garcia-Molina, 2005], como o Google. A finalidade desses *spams* é fazer com que um determinado produto ou sítio apareça como uma das primeiras referências retornadas pelos mecanismos de busca, quando é realizada a busca de determinadas palavras. Para manipular os mecanismos de busca, os *spammers* se aproveitam do fato de que esses mecanismos, geralmente, dão mais importância a sítios que são referenciados por outros sítios da Internet [Brin e Page, 1998]. Dessa forma, um *spammer* cria vários sítios que contém apenas atalhos para um determinado sítio do seu interesse. Neste sítio, que na maioria das vezes tem conteúdo pornográfico ou comercial, são inseridas palavras com as quais se deseja manipular o resultado dos mecanismos de busca. Geralmente, estas palavras estão camufladas e não têm qualquer relação com o conteúdo do sítio. Assim, como o sítio do interesse do *spammer* contém a palavra buscada e é muito referenciado por outros sítios, ele acaba tendo um resultado de destaque na busca e, conseqüentemente, atrai um grande número de usuários que somente após acessar o sítio descobrem o seu real conteúdo.

Na luta contra os *spams*, deve-se levar em consideração que a maioria dos usuários da Internet não tem formação técnica em computação, possuindo uma capacidade limitada para gerenciar e configurar seus computadores. Portanto, é fundamental que se construam sistemas o mais independentemente possível da intervenção humana. Uma das propostas nesse sentido é a proposição de sistemas autônomos [Kephart e Chess, 2003] para combater os *spams*. A idéia é fazer com que os sistemas anti-*spam* sejam capazes de se adaptar, sem a intervenção humana, às novas técnicas que vão sendo criadas pelos *spammers*. Tais sistemas devem possuir, além da característica de auto-aprendizado já encontrada em alguns sistemas atuais, as propriedades de auto-gerenciamento, auto-manutenção, auto-configuração e auto-recuperação. Neste novo paradigma, já existem propostas que utilizam mecanismos bio-inspirados [Oda e White, 2003] criando um sistema imunológico artificial capaz de combater os *spams*. O sistema imunológico humano se baseia na identificação e na destruição de agentes que causem mal ao sistema, chamados de patógenos, que podem ser vírus, bactérias ou fungos. O sistema imunológico para detectar e neutralizar os patógenos utiliza linfócitos, que são células capazes de identificar os patógenos e destruí-los. A idéia do mecanismo bio-inspirado proposto por Oda e White é utilizar linfócitos virtuais, que de forma análoga aos linfócitos do sistema imunológico humano, possuem receptores capazes de se ligar a antígenos, que representam as características do *spam* [Oda e White, 2003]. Um antígeno é uma partícula capaz de iniciar a produção de um anticorpo específico. A proposta de adoção desse modelo se baseia no fato de que um *spam* é similar a um resfriado. Ele não causa efeitos graves, mas é um mal que incomoda as pessoas e está em constante evolução.

Está claro que o envio de *spams* se tornou uma atividade financeira atrativa para

os *spammers* tanto para anunciar produtos e serviços quanto pela possibilidade de enriquecimento ilícito através de fraudes. Para tanto, os *spammers* estão se especializando cada vez mais. Neste sentido, prevê-se uma batalha interminável entre os *spammers* e os desenvolvedores de sistemas anti-*spam* e a criação de medidas legais para punir os infratores. A solução mais eficaz passa pela destruição da base do modelo de negócios dos *spammers*, que se baseia na idéia de que se mesmo um percentual muito reduzido de usuários comprarem os produtos anunciados por eles, seus anunciantes irão obter lucro e continuaram os financiando. A verdadeira base do problema está na conscientização dos usuários em não comprarem produtos anunciados através de *spams* que, na prática, é muito difícil de ser atingida.

Agradecimentos

Este trabalho foi realizado com recursos da CAPES, CNPq, FAPERJ, FINEP, RNP e FUNTTEL.

Referências

[Agência Globo, 2005] Agência Globo (2005). Brasil é 5^o maior receptor de spam; spywares representam 22% das infecções. http://www.certisign.com.br/certinews/banconoticias/2005/agosto/agosto_15_Brasil_e_5_maior_receptor_de_spam.jsp.

[Andreolini et al., 2005] Andreolini, M., Bulgarelli, A., Colajanni, M. e Mazzoni, F. (2005). Honeyspam: Honeypots fighting spam at the source. Em *International Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'05)*, páginas 77–83.

[Apache, 2006] Apache (2006). Spamassassin. http://spamassassin.apache.org/.

[Boykin e Roychowdhury, 2005] Boykin, P. O. e Roychowdhury, V. P. (2005). Leveraging social networks to fight spam. *IEEE Computer Magazine*, 38(4):61–68.

[Brin e Page, 1998] Brin, S. e Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Seventh International World-Wide Web Conference*.

[Canter e Siegel, 1994] Canter, L. A. e Siegel, M. S. (1994). Green card lottery- final one? http://www.bio.net/bionet/mm/dros/1994-April/000326.html.

[CGI.BR, 2006] CGI.BR (2006). Comitê gestor da Internet no Brasil - Antispam.br. http://www.antispam.br/.

[Commtouch, 2006] Commtouch (2006). Spam lab online statistics. http://www.commtouch.com/Site/ResearchLab/statistics.asp.

[Costales e Flynt, 2005] Costales, B. e Flynt, M. (2005). *sendmail Milters A Guide for Fighting Spam*. Addison Wesley Professional, 1^a edição.

[Cukier et al., 2006] Cukier, W. L., Cody, S. e Nesselroth, E. J. (2006). Genres of spam: Expectations and deceptions. Em *Hawaii International Conference on System Sciences (HICSS)*, páginas 1–10.

[Cullen, 2002] Cullen, L. T. (2002). Some more spam, please. *Time*, 160(20):58–59.

[Decreto-lei n^o 2.848, 1940] Decreto-lei n^o 2.848 (1940). Código penal. http://www.planalto.gov.br/ccivil_03/Decreto-Lei/Del2848compilado.htm.

[Detroit Free Press, 2002] Detroit Free Press (2002). Spam king lives large off others' e-mail troubles. http://www.freep.com.

[Emery, 2003] Emery, T. (2003). MIT conference takes aim at spam emails. *Associated Press*.

[FTC, 2005] FTC (2005). FTC - spam - home page. http://www.ftc.gov/spam/.

[Gomes et al., 2006] Gomes, L. H., Bettencourt, L. M. A., Almeida, V. A. F., Almeida, J. M. e Castro, F. D. O. (2006). Quantifying social vs. antisocial behavior in email networks. *ArXiv Physics e-prints*.

[Gomes et al., 2004] Gomes, L. H., Cazita, C., Almeida, J. M., Almeida, V. e Wagner Meira, J. (2004). Characterizing a spam traffic. Em *ACM SIGCOMM conference on Internet measurement (IMC'04)*, páginas 356–369. ACM Press.

[Goodman e Rounthwaite, 2004] Goodman, J. T. e Rounthwaite, R. (2004). Stopping outgoing spam. Em *ACM conference on Electronic commerce (EC'04)*, páginas 30–39. ACM Press.

[Gregory e Simon, 2005] Gregory, P. e Simon, M. A. (2005). *Blocking Spam & Spyware for Dummies*. Wiley Publishing, Inc.

[Grupo Brasil AntiSPAM, 2006a] Grupo Brasil AntiSPAM (2006a). Código de Ética AntiSPAM e melhores práticas de uso de mensagens eletrônicas. http://brasilantispam.locaweb.com.br/main/codigoopt.htm.

[Grupo Brasil AntiSPAM, 2006b] Grupo Brasil AntiSPAM (2006b). Página Brasil AntiSPAM.org. http://brasilantispam.locaweb.com.br.

[Gyongyi e Garcia-Molina, 2005] Gyongyi, Z. e Garcia-Molina, H. (2005). Web spam taxonomy. Em *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

[Hambridge e Lunde, 1999] Hambridge, S. e Lunde, A. (1999). *DON'T SPEW: A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*)*. RFC 2635.

[Hoanca, 2006] Hoanca, B. (2006). How good are our weapons in the spam wars? *IEEE Technology and Society Magazine*, 25(1):22–30.

[Holmes, 2005] Holmes, N. (2005). In defense of spam. *IEEE Computer Magazine*, 38(4):86–88.

[Hormel Foods, 2000] Hormel Foods (2000). Your use of our trademark SPAM on your "Page-O-SPAM" website. http://www.rsi.com/spam/.

[Jung e Sit, 2004] Jung, J. e Sit, E. (2004). An empirical study of spam traffic and the use of DNS black lists. Em *ACM SIGCOMM conference on Internet measurement (IMC' 04)*, páginas 370–375. ACM Press.

[Kephart e Chess, 2003] Kephart, J. O. e Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1):41–52.

[Klensin, 2001] Klensin, J. (2001). *Simple Mail Transfer Protocol*. RFC 2821.

[Krim, 2003] Krim, J. (2003). Lawsuits by AOL escalates fight against junk e-mail. *The Washington Post*, 15:A1.

[Laufer et al., 2005] Laufer, R. P., Moraes, I. M., Velloso, P. B., Bicudo, M. D. D., Campista, M. E. M., de O. Cunha, D., Costa, L. H. M. K. e Duarte, O. C. M. B. (2005). *Livro Texto dos Mini-cursos do V Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, chapter Negação de Serviço: Ataques e Contramedidas, páginas 1–63. Sociedade Brasileira de Computação.

[Lei n^o 8.078, 1990] Lei n^o 8.078 (1990). Código de defesa do consumidor. http://www.planalto.gov.br/ccivil_03/LEIS/L8078compilado.htm.

[Levine et al., 2004] Levine, J. R., Young, M. L. e Everett-Church, R. (2004). *Fighting Spam For Dummies*. John Wiley & Sons, 1^a edição.

[MacIntosh e Vinokurov, 2005] MacIntosh, R. e Vinokurov, D. (2005). Detection and mitigation of spam in ip telephony networks using signaling protocol analysis. *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, páginas 49–52.

[Mori e Malik, 2003] Mori, G. e Malik, J. (2003). Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. Em *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 134–141.

[Myers, 1999] Myers, J. (1999). *SMTP Service Extention for Authentication*. RFC 2554.

[Oda e White, 2003] Oda, T. e White, T. (2003). Developing an immunity to spam.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc.

[Pfleeger e Bloom, 2005] Pfleeger, S. L. e Bloom, G. (2005). Canning spam: Proposed solutions to unwanted email. *IEEE Security & Privacy Magazine*, 3(2):40–47.

[Postel, 1982] Postel, J. B. (1982). *Simple Mail Transfer Protocol*. RFC 821.

[Project, 2006] Project, A. S. (2006). Spamassassin tests performed: v3.1.x. http://spamassassin.apache.org/tests_3_1_x.html.

[Spammer-X et al., 2004] Spammer-X, Posluns, J. e Sjouwerman, S. (2004). *Inside the SPAM Cartel: Trade Secrets from the Dark Side*. Syngress Publishing, 1^a edição.

[Walker, 2005] Walker, A. (2005). *Absolute Beginner's Guide to: Security, Spam, Spyware & Viruses*. Que Publishing.

[Wiki-Spam, 2006] Wiki-Spam (2006). http://en.wikipedia.org/wiki/E-mail_spam.

[Wong e Schlitt, 2006] Wong, M. e Schlitt, W. (2006). *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL, Version 1*. RFC 4408.

[Zdziarski, 2004] Zdziarski, J. A. (2004). Bayesian noise reduction: Contextual symmetry logic utilizing pattern consistency analysis. http://bnr.nuclearelephant.com/BNR%20LNCS.pdf.

A. Artigos do código penal brasileiro

TÍTULO II
DOS CRIMES CONTRA O PATRIMÔNIO

CAPÍTULO III
DA USURPAÇÃO

Alteração de limites

Art. 161 - Suprimir ou deslocar tapume, marco, ou qualquer outro sinal indicativo de linha divisória, para apropriar-se, no todo ou em parte, de coisa imóvel alheia:

Pena - detenção, de um a seis meses, e multa.

§ 1º - Na mesma pena incorre quem:

Usurpação de águas

I - desvia ou represa, em proveito próprio ou de outrem, águas alheias;

Esbulho possessório

II - invade, com violência a pessoa ou grave ameaça, ou mediante concurso de mais de duas pessoas, terreno ou edifício alheio, para o fim de esbulho possessório.

§ 2º - Se o agente usa de violência, incorre também na pena a esta cominada.

§ 3º - Se a propriedade é particular, e não há emprego de violência, somente se procede mediante queixa.

CAPÍTULO IV
DO DANO

Dano

Art. 163 - Destruir, inutilizar ou deteriorar coisa alheia:

Pena - detenção, de um a seis meses, ou multa.

Dano qualificado

Parágrafo único - Se o crime é cometido:

I - com violência à pessoa ou grave ameaça;

II - com emprego de substância inflamável ou explosiva, se o fato não constitui crime mais grave

III - contra o patrimônio da União, Estado, Município, empresa concessionária de serviços públicos ou sociedade de economia mista; (Redação dada pela Lei nº 5.346, de 3.11.1967)

IV - por motivo egoístico ou com prejuízo considerável para a vítima:

Pena - detenção, de seis meses a três anos, e multa, além da pena correspondente à violência.

CAPÍTULO VI
DO ESTELIONATO E OUTRAS FRAUDES

Estelionato

Art. 171 - Obter, para si ou para outrem, vantagem ilícita, em prejuízo alheio, induzindo ou mantendo alguém em erro, mediante artifício, ardil, ou qualquer outro meio fraudulento:

Pena - reclusão, de um a cinco anos, e multa.

§ 1º - Se o criminoso é primário, e é de pequeno valor o prejuízo, o juiz pode aplicar a pena conforme o disposto no art. 155, § 2º.

§ 2º - Nas mesmas penas incorre quem:

Disposição de coisa alheia como própria

I - vende, permuta, dá em pagamento, em locação ou em garantia coisa alheia como própria;

Alienação ou oneração fraudulenta de coisa própria

II - vende, permuta, dá em pagamento ou em garantia coisa própria inalienável, gravada de ônus ou litigiosa, ou imóvel que prometeu vender a terceiro, mediante pagamento em prestações, silenciando sobre qualquer dessas circunstâncias;

Defraudação de penhor

III - defrauda, mediante alienação não consentida pelo credor ou por outro modo, a garantia pignoratícia, quando tem a posse do objeto empenhado;

Fraude na entrega de coisa

IV - defrauda substância, qualidade ou quantidade de coisa que deve entregar a alguém;

Fraude para recebimento de indenização ou valor de seguro

V - destrói, total ou parcialmente, ou oculta coisa própria, ou lesa o próprio corpo ou a saúde, ou agrava as conseqüências da lesão ou doença, com o intuito de haver indenização ou valor de seguro;

Fraude no pagamento por meio de cheque

VI - emite cheque, sem suficiente provisão de fundos em poder do sacado, ou lhe frustra o pagamento.

§ 3º - A pena aumenta-se de um terço, se o crime é cometido em detrimento de entidade de direito público ou de instituto de economia popular, assistência social ou beneficência.

B. Artigos do código de defesa do consumidor

TÍTULO I
Dos Direitos do Consumidor

CAPÍTULO V
Das Práticas Comerciais

SEÇÃO III
Da Publicidade

Art. 36. A publicidade deve ser veiculada de tal forma que o consumidor, fácil e imediatamente, a identifique como tal.

Parágrafo único. O fornecedor, na publicidade de seus produtos ou serviços, manterá, em seu poder, para informação dos legítimos interessados, os dados fáticos, técnicos e científicos que dão sustentação à mensagem.

Art. 37. É proibida toda publicidade enganosa ou abusiva.

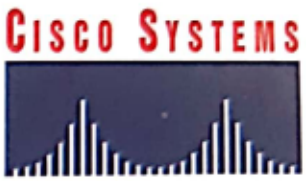
§ 1° É enganosa qualquer modalidade de informação ou comunicação de caráter publicitário, inteira ou parcialmente falsa, ou, por qualquer outro modo, mesmo por omissão, capaz de induzir em erro o consumidor a respeito da natureza, características, qualidade, quantidade, propriedades, origem, preço e quaisquer outros dados sobre produtos e serviços.

§ 2° É abusiva, dentre outras a publicidade discriminatória de qualquer natureza, a que incite à violência, explore o medo ou a superstição, se aproveite da deficiência de julgamento e experiência da criança, desrespeita valores ambientais, ou que seja capaz de induzir o consumidor a se comportar de forma prejudicial ou perigosa à sua saúde ou segurança.

§ 3° Para os efeitos deste código, a publicidade é enganosa por omissão quando deixar de informar sobre dado essencial do produto ou serviço.

§ 4° (Vetado).

patrocínio



apoio

promoção

organização