UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA DA COMPUTAÇÃO

Fabiano Castro Pereira

Estudo e Implementação de Redes de Comunicação Anônima e aplicação ao Sistema de Votação Digital OSTRACON

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis, Janeiro de 2005

Estudo e Implementação de Redes de Comunicação Anônima e aplicação ao Sistema de Votação Digital OSTRACON

Fabiano Castro Pereira

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Raul Sidnei Wazlawick, Dr.
Coordenador do Curso

Banca Examinadora

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Prof. Jeroen Antonius Maria van de Graaf, Dr.

Osvaldo Catsumi Imamura, Dr.

Prof. Carla Merkle Westphall, Dr.

Prof. Luiz Carlos Zancanella, Dr.

"Tesouros mal adquiridos de nada servem,
mas a justiça livra da morte."
Provérbios 10,2

Ofereço este trabalho a todas as pessoas que, em suas vidas, buscam o exercício da justiça, seja qual for a área do conhecimento a que se dedicam e investem seu tempo.
A todas as pessoas que buscam garantir os direitos dos demais cidadãos, principalmente os que foram escolhidos por estes para serem seus representantes no Governo de nossa nação, e que o fazem com honestidade, com justiça.

Agradecimentos

Agradeço primeiramente a **Deus**, por conceder-nos o Dom da Vida e garantir-nos a certeza da Vida Eterna;

Agradeço a meu pai Osni, minha mãe Eliane, e demais familiares, que me deram a primeira formação e os primeiros ensinamentos;

Agradeço à minha esposa Simone pela compreensão e incentivo durante todo este período;

Agradeço a todos os meus mestres e professores ao longo destes anos de estudo, principalmente àqueles que me instruíram na vida acadêmica;

Agradeço em especial ao professor Ricardo Felipe Custódio por ter me orientado desde o curso de graduação e por ter me despertado o interesse por esta área da Ciência da Computação que é tão desafiante, a Segurança em Computação;

Agradeço também a todos os colegas e amigos que têm me ajudado e incentivado ao longo destes anos de estudo.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Fragmentos de Código

Lista de Siglas

API	*Application Program Interface* - Interface para Programas Aplicativos.
ASN.1	*Abstract Syntax Notation One* - Notação Abstrata de Sintaxe versão Um.
B2B	*Business-to-Business* - Empresa-para-Empresa.
FTP	*File Transfer Protocol* - Protocolo de Transferência de Arquivo.
HTTP	*HyperText Transfer Protocol* - Protocolo de Transferência de Hipertexto.
HTTPS	*Secure HyperText Transfer Protocol* - Protocolo Seguro de Transferência de Hipertexto.
IETF	*Internet Engineering Task Force* - Força-tarefa de Engenharia da Internet.
IRC	*Internet Relay Chat* - Conversa pela Internet.
NIST	*National Institute of Standards and Technology* - Instituto americano que define padrões e tecnologias naquele país.
NNTP	*Network News Transfer Protocol* - Protocolo de Transferência de Notícias em Rede.
P2P	*Peer-to-Peer* - Colega-para-Colega.
RCA	Rede de Comunicação Anônima.
RFC	*Request For Comments* - Série de documentos contendo propostas de atualização de padrões, ou sugestão de novos, para a Internet.
SSL	*Secure Socket Layer* - Camada de Socket Seguro.
TLS	*Transport Layer Security* - Segurança da Camada de Transporte.

Lista de Símbolos

K_a	Chave simétrica escolhida pela entidade a
$K_a(X)$	Cifração do conteúdo X utilizando a chave simétrica escolhida pela entidade a
KU_a	Chave pública da entidade a
KR_a	Chave privada da entidade a
$KU_a(X)$	Cifração do conteúdo X utilizando a chave pública da entidade a
$KR_a(X)$	Cifração do conteúdo X utilizando a chave privada da entidade a
M	Uma mensagem a ser enviada
A	Endereço do receptor de uma mensagem
R_x	Dados aleatórios de preenchimento em uma mensagem
$\rightarrow$	Operação de decifração feita por um servidor

Resumo

Em muitos casos de uso de sistemas em rede o anonimato da comunicação apresenta-se como um requisito desejado. Este trabalho tem como tema principal as técnicas para comunicação anônima. Para um melhor conhecimento do problema do anonimato foram pesquisadas formas de ataque ao anonimato, e mecanismos de defesa para tais ataques. As técnicas para comunicação anônima foram estudadas e avaliadas quanto a eficácia no combate aos ataques, e quanto ao provimento de comunicação anônima. Com base nas técnicas pesquisadas, foi proposta uma implementação de uma rede para comunicação anônima. Com a implementação realizada foi possível medir a performance da rede com ênfase nas operações criptográficas necessárias, e avaliar a aplicação prática da técnica escolhida no Sistema Ostracon, um sistema de votação digital desenvolvido no Laboratório de Segurança em Computação da Universidade Federal de Santa Catarina.

Palavras-chave: comunicação em rede, criptografia, anonimato, votação digital.

Abstract

Many network systems use cases have communication anonymity as a desired requirement. This work has as subject the techniques to achieve anonymous communication. For a better realization of the anonymity issue, a research for many anonymity attacks and defense techniques for such attacks was made. Techniques for anonymous communication were studied and evaluated about their efficiency on defending against attacks, and about their anonymous communication provisioning. Based on studied techniques, an anonymous communication network implementation proposal was made. Such implementation allowed to place measures of implemented network performance, with emphasis on cryptographic operations needed, and to evaluate the use of the chosen technique into Ostracon System, a digital voting system developed in the Computer Security Laboratory of Santa Catarina Federal University.

Keywords: network communication, cryptography, anonymity, digital voting.

Capítulo 1

Introdução

Em algumas aplicações para redes de computadores o anonimato é desejado como característica básica do sistema. Entende-se por anonimato a possibilidade de realizar uma comunicação sem que seja necessária a identificação das pessoas que estão realizando tal comunicação. Ou ainda, a realização de uma comunicação entre duas pessoas que conhecem uma a outra, sem que uma terceira pessoa saiba da existência daquela comunicação.

A necessidade do anonimato fica evidente quando se faz uso, por exemplo, de sistemas digitais para a realização de votações, onde deseja-se que o voto não seja associado ao votante. Entretanto, para a garantia desta característica são necessários mecanismos bem elaborados. No caso dos sistemas de votação estes são mecanismos criptográficos, tais como protocolos criptográficos. Estes protocolos devem definir claramente como será atingido o anonimato, quando desejado.

Este trabalho trata especificamente das propostas de técnicas para comunicação anônima conhecidas atualmente para serem utilizadas em redes de computadores. Foi feita uma análise da segurança apresentada por algumas técnicas, e também a verificação da adequação das mesmas para uso em sistemas de comunicação em rede. Este trabalho também propõe uma implementação de um sistema para anonimato, baseada em algumas técnicas de anonimato estudadas, e uma aplicação prática desta implementação

em um sistema de votação digital[1].

1.1 Motivação

A principal motivação para este trabalho foi a possibilidade de contribuir com a pesquisa em segurança de redes de computadores, especificamente na questão da garantia do anonimato da comunicação. A pesquisa nesta área dispõe de muitas possibilidades, podendo trazer benefícios para a sociedade.

Um instrumento bastante conhecido e utilizado no dia-a-dia das pessoas, e que claramente provê benefícios para a sociedade por utilizar comunicação anônima, é o chamado **disque-denúncia**. Através de uma ligação telefônica anônima um cidadão pode auxiliar o trabalho da polícia, inclusive ajudando na solução de crimes. Sem a garantia do anonimato nesta ligação telefônica as pessoas não se sentiriam seguras para fazer suas denúncias, pois a identificação poderia trazer riscos.

Instrumentos como este já contam inclusive com versões para a Internet, como a página de denúncia anônima da Polícia Civil do Estado de Santa Catarina (POLÍCIA CIVIL DE SANTA CATARINA, 2004). Apesar de permitir um certo nível de anonimato para a realização da denúncia, esta página apresenta um problema de segurança por tratar-se de uma conexão HTTP comum, sem a utilização do protocolo SSL. A pesquisa por técnicas de comunicação anônima permite trazer melhorias para instrumentos como este.

A utilização de comunicação anônima também contribui para um problema social que se tornou mais grave com o crescimento do número de usuários da Internet, a **privacidade**. Garfinkel (2000, p. 4) ressalta que a privacidade não diz respeito apenas a manter dados pessoais escondidos, mas também diz respeito à autonomia e integridade da pessoa; o direito à privacidade deve permitir que a pessoa tenha controle sobre quais dados são mantidos em segredo, e quais podem ser livremente informados.

[1] Neste trabalho, o termo **votação digital** é usado para designar votações realizadas exclusivamente através da Internet, ou seja, sem a necessidade de equipamentos específicos para votação. E o termo **votação eletrônica** é usado para designar votações que exigem equipamentos específicos, como é o caso das eleições brasileiras, que fazem uso da urna eletrônica.

Violações à privacidade do usuário da Internet são feitas quando se faz observações do comportamento do usuário com fins lucrativos. Uma empresa de publicidade, ou que tenha interesse na venda de produtos diversos pode ter seus lucros aumentados ao ter acesso aos dados de navegação, e às preferências dos usuários, invadindo sua privacidade (CHAUM, 1985, p. 1). Os hábitos de acesso a sítios da Web podem ser obtidos caso os projetistas e operadores dos sítios não tomem medidas de segurança com relação à privacidade de seus usuários (GARFINKEL; SPAFFORD, 1997, p. 69).

Outro problema de privacidade que ocorre na Internet, e que também é bastante discutido, são as mensagens eletrônicas não solicitadas que um usuário recebe (prática conhecida como *spam*), e que têm geralmente objetivos comerciais. Com os usuários da Internet tendo acesso facilitado a ferramentas de comunicação anônima, os problemas com a privacidade dos seus dados também são atenuados. Goldberg, Wagner e Brewer (1997, p. 1) relatam as possibilidades de invasão de privacidade que a monitoração e o registro de dados do usuário pode ocasionar, tais como o conhecimento de mensagens de correio eletrônico, mensagens enviadas para grupos de notícia (*newsgroups*), e a relação de sítios visitados.

Em um trabalho mais recente de Goldberg (2002) são enumeradas diversas tecnologias que têm como principal objetivo a melhoria da privacidade no uso da Internet. A proteção de informações pessoais é considerada um dos objetivos mais importantes dos protocolos de comunicação em redes como a Internet (HUGHES; SHMATIKOV, 2004, p. 15).

1.2 Justificativa

Estando este trabalho inserido em um projeto maior, cujo tema é votação digital, também é importante esclarecer os benefícios do uso de votação digital, e destacar a importância da aplicação de comunicação anônima em votação digital.

Tendo-se cada vez mais pessoas conectadas à Internet, surgem novas oportunidades de aplicações, muitas delas dispondo de benefícios inviáveis sem a pre-

sença de um meio de comunicação deste porte. Um exemplo é a realização de votações de forma digital.

No seu dia-a-dia as pessoas realizam escolhas, sendo que muitas destas escolhas são feitas por um grupo de pessoas com um mesmo interesse, seja numa empresa, numa assembléia de sindicato, ou até mesmo numa reunião de um condomínio. Estas escolhas, ou votações, podem ocorrer em grupos com pequeno número de participantes, como as citadas, ou com grande número de pessoas, sendo que dentre as mais conhecidas pode-se citar as eleições oficiais para cargos públicos, como ocorre no Brasil.

Tanto para votações com pequeno número de pessoas quanto para grandes eleições com milhares de participantes, o uso de sistemas informatizados para a realização das votações apresenta vantagens. A primeira vantagem, facilmente identificada, é a velocidade na apuração e publicação dos resultados, que são feitas com tempo reduzido. Conforme Nazário (2003, p. 72), esta vantagem pôde ser observada nas eleições brasileiras quando passaram a ser realizadas de forma eletrônica, com o uso das urnas eletrônicas, onde o resultado da eleição de cada urna é publicado imediatamente após o encerramento da votação, através do boletim de urna.

Uma vantagem advinda do uso de votações digitais é a redução dos custos na realização de uma votação, não sendo mais necessários gastos com alguns ítens da infra-estrutura física tradicionalmente usada, tais como cédulas, urnas e demais materiais.

Com votações digitais outra grande vantagem é a mobilidade, que não se observa no sistema eletrônico, o qual obriga o votante a comparecer à sua seção eleitoral. Com o uso da Internet, para exercer o direito de voto é preciso apenas utilizar um computador conectado à grande rede. Conforme ressalta Devegili (2001, p. 28), em uma votação digital a mobilidade fica dependente apenas das restrições da rede de comunicação, em vez de ficar dependente de locais físicos previamente determinados.

Esta mobilidade que o votante possui em votar é uma grande contribuição dos sistemas digitais para o processo democrático pelo qual se fundamenta a existência de votações ou eleições. Além do benefício de exercer a democracia, passa-se a poder exercê-la em praticamente qualquer lugar.

No ano de 1999 o presidente americano Bill Clinton solicitou à **Fun-**

dação Americana de Ciência (NSF - *National Science Foundation*) um estudo sobre a viabilidade da realização de votações através da Internet. De acordo com o resultado da pesquisa realizada pela fundação, sistemas de votação digital oferecem benefícios e são viáveis pois, apesar de ainda existirem problemas a serem resolvidos, é provável que os mesmos sejam solucionados em pouco tempo (IPI, 2001, p. 34). Este estudo também afirma que as falhas em sistemas de votação, tais como as que ocorreram nas eleições presidenciais americanas no ano de 2000, se devem à falta de pesquisa e análise dos processos de votação (IPI, 2001, p. 35); e que estas pesquisas devem envolver tanto cientistas sociais quanto oficiais de votação e especialistas em tecnologia da informação.

A pesquisa em votação digital também é incentivada pelo governo brasileiro, pois, como ressaltou o Secretário de Informática do Tribunal Superior Eleitoral (TSE), Paulo César Camarão, *"Quem não pensar nisso está perdendo tempo."* (CARVALHO, 2000, p. 39), que também se referiu à Infra-Estrutura de Chaves Públicas Brasileira (ICP-Brasil): *"Demos o primeiro passo e isso é muito importante para concretizarmos, um dia, o sonho do voto via Internet."* (CARVALHO, 2000, p. 39).

A utilização de comunicação anônima em sistemas de votação digital pode trazer benefícios para a sociedade ao gerar novas tecnologias que auxiliem na garantia do sigilo do voto, no processo de votação. Exemplos destes benefícios podem ser encontrados no trabalho realizado por Nazário (2003).

Na garantia do sigilo do voto ocorre uma forma particular de anonimato, que diz respeito ao destinatário de determinada comunicação. Ou seja, para o sigilo do voto, ninguém além do votante deve saber a quem se destina seu voto. A necessidade de comunicação anônima para a garantia do sigilo do voto em sistemas de votação digital é a principal motivação para a implementação prática realizada, parte do presente trabalho.

Ataques a sistemas computacionais põem em risco a segurança das informações que trafegam em uma rede de computadores, podendo ocorrer alteração de informações, adição de informações falsas e captação de informações sigilosas. Em uma votação digital isto significaria a alteração de um voto trafegando na rede, a emissão de votos falsos, ou a obtenção do conteúdo de uma cédula de votação, desfazendo o sigilo do voto.

A criptografia é uma ferramenta útil para solucionar os problemas relacionados à segurança da informação que trafega em redes de computadores. Ela se propõe a dificultar ao máximo a obtenção não autorizada de informações sigilosas, nem a produção ou alteração das mensagens enviadas.

Entretanto, a criptografia tradicional por si só não é suficiente para obter a segurança desejada para uma aplicação como votação digital. O processo de votação deve apresentar requisitos de segurança que vão além daqueles desejados para redes de computadores. Para garantir estes requisitos adicionais de segurança é preciso estabelecer protocolos para os processos de votação, neste caso protocolos criptográficos, sendo esta uma grande área de pesquisa.

Um protocolo criptográfico para votação digital precisa levar em conta diversos aspectos do processo de votação. Um dos aspectos principais e mais importantes é o anonimato. Pelo fato de o anonimato ser de importância fundamental para o sucesso de um sistema de votação digital, a forma como ele é implementado é inclusive considerado um princípio básico para a classificação de sistemas deste tipo (RIERA, 1999, p. 14).

A dificuldade em conseguir anonimato em um protocolo de votação digital se dá por vários motivos. Um primeiro motivo é a necessidade de ter, no mesmo protocolo, autenticação em um dado momento, e anonimato em outro. Isto ocorre pois apenas votantes autorizados realizam seus votos, tornando necessária a autenticação por parte do votante, e a verificação da presença do mesmo na lista de votantes autorizados. Entretanto, no momento da realização do voto pode-se desejar o anonimato, exigindo que aquele votante autenticado perante o sistema não tenha mais sua identidade revelada, para se evitar a associação do voto ao votante.

Outra dificuldade para obtenção de comunicação anônima é a transformação de soluções teóricas em aplicações práticas. Ainda que uma proposta de solução teórica exista, se o esforço computacional necessário para sua implementação prática for excessivo, não haverá como obter benefícios da teoria desenvolvida.

1.3 Objetivos

1.3.1 Objetivo Principal

Este trabalho tem como objetivo principal descrever as técnicas estudadas para a solução do problema do anonimato em redes de computadores, analisando a aplicação das mesmas em sistemas existentes; e com estes resultados propor uma rede de comunicação anônima, projetada a partir dos procedimentos encontrados nas técnicas pesquisadas.

1.3.2 Objetivos Específicos

Como a questão do anonimato é complexa, as necessidades do uso de criptografia são muitas. É preciso o uso de técnicas bastante elaboradas, resultando na utilização de protocolos e outros mecanismos criptográficos. Esta necessidade resultou na pesquisa, estudo e compreensão das técnicas de criptografia comumente utilizadas para anonimato.

As técnicas que se propõem a solucionar o problema do anonimato precisam ser resistentes aos diversos ataques que uma rede de comunicação pode sofrer. Para poder analisar a eficácia destas técnicas, foi preciso conhecer os diversos tipos e o funcionamento de cada um dos ataques ao anonimato.

A realização de uma implementação prática de técnicas de anonimato possibilita uma melhor compreensão dos conceitos envolvidos em técnicas deste tipo. Com a localização deste trabalho dentro de um projeto maior, foi possível aplicar técnicas de anonimato em um sistema real, de votação digital.

Com essas considerações, pode-se destacar os objetivos específicos deste trabalho:

- Compreensão dos conceitos envolvidos no uso de anonimato;
- Conhecimento das formas de ataque ao anonimato;
- Pesquisa e estudo de técnicas para comunicação anônima encontradas na literatura;
- Análise da aplicação destas técnicas em sistemas existentes;
- Implementação de uma rede de comunicação anônima;
- Aplicação prática da rede implementada.

1.4 Trabalhos Correlacionados

Com o objetivo de realizar pesquisas na área de votação digital, o Laboratório de Segurança em Computação (LabSEC) iniciou no ano de 2000 o **Projeto Ostracon**[2]. A primeira etapa do projeto, que consistiu em uma dissertação de mestrado (DEVEGILI, 2001), resultou na proposta de um protocolo criptográfico para votação digital, o **Protocolo Farnel** (ARAÚJO; DEVEGILI; CUSTÓDIO, 2002), visando cumprir os requisitos de segurança identificados em um processo de votação digital.

A primeira implementação simplificada do protocolo Farnel foi resultado de um trabalho de conclusão de curso realizado por Pereira e Mazzi (2001), e fez uso das ferramentas criptográficas disponíveis no sistema operacional Windows para executar algumas partes do protocolo, sem abrangê-lo por completo. Este sistema foi denominado **Sistema Ostracon**. A segunda dissertação de mestrado com tema associado ao projeto (ARAÚJO, 2002) teve como objetivo fazer um estudo dos principais protocolos criptográficos para votação digital encontrados no meio acadêmico com ênfase nas implementações existentes. Esta dissertação, juntamente com um segundo trabalho de conclusão de curso (MASELLA, 2002), também resultou numa implementação prática,

[2] **Ostracon** - pedaço de pedra calcária usado na Antigüidade para desenhos ou registros.

obtendo-se a segunda versão do Sistema Ostracon. Esta versão implementou outros protocolos, permitindo ao administrador da votação escolher o protocolo mais adequado e com os requisitos de segurança desejados.

O Protocolo Farnel especifica a utilização de comunicação anônima para a garantia do sigilo do voto. Nenhuma das implementações do Sistema Ostracon realizadas faz uso da comunicação anônima especificada no Protocolo Farnel, e nenhum estudo deste tema foi feito para se identificar formas de implementar esta característica do protocolo. Os resultados apresentados neste trabalho servem como contribuições para o Protocolo Farnel, pois tratam especificamente da comunicação anônima. A implementação realizada durante o desenvolvimento deste trabalho foi aplicada ao Sistema Ostracon, e constitui um elemento importante para a obtenção de uma implementação completa do Protocolo Farnel.

Como um dos objetivos deste trabalho, com relação à aplicação em votação digital, trata apenas da utilização de comunicação anônima para a garantia do sigilo do voto, outras características também importantes não estão dentro do escopo deste trabalho, tais como a autenticação do votante.

Outras publicações foram realizadas com o desenvolvimento do presente trabalho. A primeira (PEREIRA; CUSTÓDIO, 2003), descreveu o Sistema Ostracon, incluindo a utilização de uma rede de comunicação anônima neste sistema para a garantia do anonimato do votante. A segunda publicação (DIAS et al., 2004), propôs a utilização de uma rede de comunicação anônima em conjunto com um sistema de votação por lista de discussão, mostrando como a utilização de um sistema deste tipo pode melhorar o processo democrático presente em uma votação.

1.5 Metodologia

Para alcançar os objetivos deste trabalho foram pesquisados diversos artigos e publicações cujo tema era correlacionado ao assunto tratado. Foram avaliadas as propostas que possuem contribuições relacionadas à busca de soluções para o problema

do anonimato.

Para realizar as avaliações foram levados em conta a abrangência de determinada proposta quanto ao cumprimento do tema, o anonimato. Também foi considerado a adequação às necessidades computacionais para se tornar viável a implementação prática da proposta.

A aplicação prática do mecanismo de anonimato implementado foi feita aprimorando um sistema de votação digital já existente, o Sistema Ostracon (ARAÚJO, 2002, p. 74). Ele faz uso da Internet, mais especificamente a Web (protocolos HTTP e HTTPS), para a realização de uma votação de forma digital.

Foram feitos experimentos para avaliar o desempenho da implementação de um sistema de comunicação anônima realizada, tanto do módulo servidor, quanto do módulo cliente. As características do funcionamento da rede permitiram a realização de um experimento com um fator (tamanho da mensagem) para o módulo servidor, e um experimento com dois fatores (tamanho da mensagem e número de servidores) para o módulo cliente. Para obter confiança nos valores das medições de desempenho realizadas, foi utilizado o método estatístico da **análise de variância** (também conhecido como **ANOVA**). Os detalhes dos experimentos são descritos na seção 4.6.

1.6 Organização do Texto

O capítulo 2 discute diversos pontos relacionados ao problema do anonimato em redes de computadores. No capítulo 3 são apresentadas as técnicas para comunicação anônima pesquisadas e analisadas. No capítulo 4 é apresentada a implementação realizada de um sistema para comunicação anônima. O capítulo 5 apresenta o Sistema Ostracon, um sistema digital para votações onde foi feita a aplicação prática do sistema de anonimato proposto. O capítulo 6 apresenta as considerações finais.

Capítulo 2

Anonimato em Redes

2.1 Introdução

O desejo de realizar comunicações de forma anônima pode ter diversas motivações, principalmente quando se trata da comunicação entre as pessoas de um modo geral. A primeira motivação é a possibilidade de expressar idéias que não poderiam ser colocadas de forma não-anônima, com a identificação da pessoa que está expressando determinada idéia. Seja porque isto ocasionaria uma repressão à pessoa por parte do grupo com o qual ela se comunica, ou até mesmo porque isto poderia causar danos físicos à pessoa, tal como ocorre em determinados regimes governamentais.

Uma segunda motivação para o anonimato é a possibilidade de entretenimento entre um grupo de pessoas que se comunicam de forma anônima, tal como ocorre em alguns jogos, ou então como é claramente observado nas **conversas via Internet** (**IRC** - *Internet Relay Chat*), onde as pessoas muitas vezes se divertem ao se passar por uma pessoa, fingindo ser de determinado modo, ou ter determinada característica, o que é garantido pelo fato de o IRC (IETF, 1993) não prover meios muito confiáveis para que as pessoas identifiquem com quem estão falando, ou seja, ele provê um certo grau de anonimato na comunicação.

Outra motivação que pode ser encontrada, porém não tão nobre, é a possibilidade de se intimidar pessoas através do anonimato, tal como ocorre em seqüestros

onde os seqüestradores fazem uso do anonimato em ligações telefônicas para se comunicar com a família do seqüestrado. A possibilidade da utilização mal intencionada do anonimato da comunicação deve ser considerada para a escolha de um sistema de anonimato. Esta decisão envolve questões éticas e até mesmo políticas ou governamentais (SHERWOOD; BHATTACHARJEE; SRINIVASAN, 2002, p. 13).

Esta exposição de motivações para o anonimato não é exaustiva, e com certeza existem ainda outras motivações para a realização de comunicações anônimas. O anonimato em redes de computadores pode possuir as mesmas motivações que se encontram em comunicações em geral, pois as redes de computadores podem ser tratadas, de uma forma mais genérica, como uma ferramenta para a comunicação entre as pessoas.

Este capítulo apresenta conceitos envolvidos no uso de anonimato, também mostrando como a comunicação anônima pode estar presente em aplicações básicas de comunicação em rede, tais como o correio eletrônico, e a navegação na Web. Aqui também são apresentadas diversas formas de ataque que podem ser realizados contra sistemas de comunicação em rede, com o objetivo de obter informações que revelam dados dos usuários destes sistemas.

Na seção 2.2 serão descritas as propriedades da comunicação anônima. A seção 2.3 trata do uso do anonimato no correio eletrônico. Uma técnica conhecida como "servidores procuradores" é apresentada na seção 2.4. Na seção 2.5 são descritos os ataques ao anonimato mais comuns. A seção 2.6 conclui o capítulo.

2.2 Propriedades da comunicação anônima

Para a utilização de comunicação anônima é preciso definir quais as informações se deseja manter anônimas, e quais as possibilidades de anonimato na comunicação. Pfitzmann e Waidner (1985) descrevem as três propriedades da comunicação anônima que podem ser obtidas:

- **Anonimato do remetente:** quando esta propriedade é obtida, a identidade do remetente de uma mensagem é mantida em segredo;

- **Anonimato do destinatário:** esta propriedade é similar à anterior, sendo que neste caso é a identidade do destinatário que é mantida em segredo. Outra forma pela qual esta propriedade pode se fazer presente é quando somente o destinatário sabe que ele recebeu uma mensagem, ninguém mais;

- **Impossibilidade de associação do remetente ao destinatário:** esta propriedade indica que, embora o remetente e o destinatário possam ser identificados como participantes de uma comunicação, não se pode estabelecer uma associação identificando qual remetente está se comunicando com qual destinatário.

Muitas vezes o anonimato desejado em uma comunicação não necessita obter as três propriedades. Em alguns casos as pessoas ou sistemas que estiverem se comunicando podem identificar-se, mas apenas entre si. Neste caso a propriedade desejada é a impossibilidade da associação. Por exemplo, muitos sítios da Internet exigem a autenticação por parte do cliente para que ele tenha acesso a determinadas páginas, devendo o usuário identificar-se perante o servidor. Mesmo com esta identificação, o usuário pode desejar uma comunicação anônima, no sentido de que outras pessoas não saibam que ele está acessando especificamente aquele sítio.

As propriedades apresentadas dizem respeito ao anonimato da comunicação de forma genérica. O anonimato desejado em serviços específicos pode exigir uma ou mais destas propriedades. Em especial, se considerarmos uma aplicação de votação digital, o anonimato desejado é bastante peculiar. O que ocorre neste caso é a necessidade do sigilo do voto, de forma a não associar o voto ao votante, sendo que este último não é anônimo, pois necessita se identificar para ser autorizado a realizar seu voto. A propriedade da comunicação anônima desejada neste caso é a impossibilidade de associação do remetente ao destinatário, pois se deseja que o voto (comunicação em questão) realizado por um votante (remetente) em favor de um candidato/opção (destinatário) não seja revelado.

Os sistemas para anonimato da comunicação buscam garantir uma ou mais das propriedades apresentadas. A tentativa de desfazer uma ou mais destas propriedades é que caracteriza os **ataques ao anonimato** (veja seção 2.5, na página 19).

O principal método utilizado em ataques às redes de comunicação anônima é a análise de tráfego, pelo qual um atacante busca obter informações sobre as mensagens que trafegam na rede, tais como o tamanho dos pacotes de dados, o tempo despendido durante o tráfego, a freqüência com que as mensagens são trocadas, os períodos mais intensos de comunicação, etc. Com estas informações em mãos o atacante pode estabelecer uma relação entre a origem e o destino de uma ou mais mensagens (AGRAWAL; KESDOGAN; PENZ, 2003, p. 1).

Para tanto o atacante pode monitorar as conexões da rede de comunicação (ataques passivos), ou até mesmo pode ter comprometido certas partes do sistema de comunicação utilizado (ataques ativos). Ele pode estar situado próximo do emissor, do receptor, ou dos pontos da rede por onde as mensagens trafegam, ou ainda uma combinação destes locais.

Desde os primeiros trabalhos de pesquisa sobre comunicação anônima, o problema da análise de tráfego já estava presente, o qual se agravou com o crescimento do uso do correio eletrônico (CHAUM, 1981, p. 1).

2.3 Correio eletrônico anônimo

O correio eletrônico é uma forma de comunicação muito utilizada pelos usuários da Internet. Com relação ao anonimato da comunicação, um usuário pode desejar emitir opiniões sobre assuntos polêmicos sem que se revele sua identidade (primeira propriedade da comunicação anônima). Isto pode ocorrer por exemplo em listas de discussão através de correio eletrônico. Como o protocolo SMTP, para envio de mensagens de correio eletrônico, realiza a identificação do remetente e também dos servidores intermediários por onde a mensagem trafegou, o anonimato fica comprometido.

Com o objetivo de proporcionar o anonimato do remetente de uma mensagem eletrônica, foram criados sistemas denominados **reenviadores anônimos** (*anonymous remailer*). Estes sistemas são classificados de acordo com as funcionalidades providas, e podem ser dos tipos “0”, “I”, “II” ou “III” (DANEZIS; DINGLEDINE; MATHEWSON,

2003, p. 2).

O primeiro sistema reenviador anônimo, que foi classificado como sendo do tipo 0, consistia em uma modificação de um servidor de correio eletrônico, que identificava um cabeçalho extra na mensagem, o qual continha o endereço final de envio. Ao enviar uma mensagem para este servidor, o usuário colocava este cabeçalho extra, informando o destino desejado para sua mensagem. Ao receber uma mensagem, o servidor retirava o endereço do remetente (e qualquer outra informação que pudesse identificá-lo) e a reenviava para o destino final. A mensagem então recebia um novo endereço de remetente, identificando o servidor anônimo. Desta forma, mensagens de retorno poderiam ser enviadas sem que se soubesse a identidade do remetente original.

Apesar de proporcionar anonimato na comunicação, um sistema do tipo 0 não apresentava a segurança necessária, pois não fazia uso de criptografia, mantendo a codificação original das mensagens. E por ser um sistema centralizado, uma falha no servidor tornaria o serviço indisponível, além do que a relação de remetentes originais e respectivos endereços anônimos estava disponível para o administrador. Inclusive, conforme relata Goldberg, Wagner e Brewer (1997, p. 4), um sistema real do tipo 0, mantido por Johan Helsingius, teve o anonimato de um de seus usuários desfeito através de uma ação judicial.

As desvantagens dos sistemas do tipo 0 levaram ao desenvolvimento de sistemas do tipo I. A arquitetura de um sistema deste tipo consiste de diversos servidores que fazem uso de criptografia assimétrica. Para enviar uma mensagem anônima, o usuário deve cifrar sua mensagem com a chave pública de cada servidor. Com o envio de uma mensagem, o primeiro servidor conhece apenas o remetente, o último servidor conhece apenas o destinatário, e os servidores intermediários não têm nenhuma informação útil sobre a mensagem. Este sistema também permite o envio de respostas para o remetente, sem revelar sua identidade.

Entretanto, conforme Mazières e Kaashoek (1998, p. 4), os sistemas reenviadores do tipo I são vulneráveis a ataques de análise de tráfego, pois eles procuram reenviar as mensagens assim que são recebidas. Outro ponto que auxilia na realização de um ataque é que o tamanho das mensagens enviadas depende do tamanho da mensagem

original. Desta forma um atacante poderia observar o tráfego ao longo dos pontos da rede e, de acordo com os tempos de comunicação e o tamanho das mensagens, descobrir a origem e o destino da comunicação (GOLDBERG, 2002, p. 2).

Com as vulnerabilidades dos sistemas do tipo I, surgiram os sistemas do tipo II. Neste tipo de sistema todas as mensagens enviadas possuem o mesmo tamanho. Quando uma mensagem chega a um servidor, ela é decifrada e colocada em um conjunto. Ao atingir um certo número de mensagens armazenadas, o servidor retira uma mensagem de forma aleatória e a reenvia.

Mixminion (DANEZIS; DINGLEDINE; MATHEWSON, 2003) é um sistema de reenvio do tipo III. Este sistema permite o uso de serviços de diretório para que os usuários possam conhecer a chave pública e as estatísticas de uso e performance dos servidores integrantes da rede. Os servidores de diretório são redundantes e são mantidos sincronizados para garantir que todos os usuários sempre tenham disponível a mesma relação de servidores. Para que esta relação de servidores e suas estatísticas estejam sempre corretas e completas, os servidores de diretório assinam as informações para que os usuários tenham a garantia de que possuem dados verdadeiros.

O sistema Mixminion também possui um mecanismo mais elaborado para o envio de mensagens de resposta sem que se conheça o remetente original. Neste mecanismo é utilizada a técnica dos endereços pseudônimos (veja próxima seção), e os servidores podem operar de duas formas. Na primeira, o remetente envia juntamente com a mensagem um bloco de resposta contendo um endereço pseudônimo, o qual deve ser usado pelo servidor para submeter respostas enviadas pelo destinatário. Na segunda forma de operação as respostas são mantidas no servidor, e o remetente deve verificar periodicamente a chegada de respostas.

Com relação às propriedades da comunicação anônima, os sistemas para envio de correio eletrônico anônimo buscam garantir principalmente a impossibilidade de associação do remetente ao destinatário, sendo também possível o anonimato do remetente, caso este não queira se identificar perante o destinatário de suas mensagens.

2.3.1 Endereços pseudônimos

O endereço fictício que substitui o endereço real do remetente, utilizado principalmente nos sistemas reenviadores do tipo 0, caracteriza um **endereço pseudônimo** de correio eletrônico (MAZIÈRES; KAASHOEK, 1998, p. 1). O uso de pseudônimos em sistemas de comunicação já havia sido sugerido por Chaum (1985, p. 10) como uma forma de garantir a segurança de um indivíduo sem a necessidade da sua identificação. Este conceito de endereço pseudônimo pode ser estendido, podendo ser usado em outras aplicações além de correio eletrônico, conforme propôs Goldberg (2000) ao desenvolver uma infra-estrutura de comunicação com pseudônimo para a Internet.

Mesmo sendo útil para o remetente de mensagens de correio eletrônico que deseja permanecer anônimo, o uso de pseudônimo pode não garantir o anonimato em determinadas situações, principalmente se o remetente possuir certos padrões de escrita. Rao e Rohatgi (2000) demonstram que com a utilização de determinadas técnicas lingüísticas é possível analisar a sintaxe e a semântica do texto de mensagens de correio eletrônico, podendo-se identificar determinadas mensagens como sendo de um mesmo autor. Caso este autor tenha enviado intencionalmente alguma mensagem identificada, torna-se possível relacionar sua identidade às mensagens anteriores, enviadas através de um pseudônimo. Apesar de terem utilizado apenas análise sintática e semântica, os autores propõem o uso de outras características, tais como a freqüência de palavras ortograficamente incorretas, a forma de utilização da pontuação, ou até mesmo a formatação aplicada ao texto, envolvendo espaçamentos, identação e etc.

Outra questão advinda do uso de pseudônimos é a necessidade de estabelecer relações de confiança entre portadores de pseudônimos. Apesar de desejarem permanecer anônimos, ainda assim os donos de pseudônimos necessitam estabelecer relações com os outros participantes da comunicação, seja por correio eletrônico ou outra forma. Friedman e Resnick (2001) analisam mecanismos que proporcionam confiança entre os portadores de pseudônimos, que fazem uso de dois princípios, o da repetição, e o da reputação. A repetição tem o objetivo de garantir, após várias comunicações com outro pseudônimo, que o portador do mesmo comporta-se de forma honesta. A reputação per-

mite que esta confiança estabelecida através da repetição possa ser propagada para outros portadores de pseudônimo com os quais também já se tenha confiança estabelecida.

2.4 Servidores procuradores

De acordo com Luotonen e Altis (1994, p. 1), os servidores procuradores (*proxy servers*) são softwares servidores que dão acesso às páginas Web para clientes que estão dentro de uma subrede e que têm acesso à Internet apenas através de um *firewall*.

Um procurador é implementado como um servidor HTTP especializado. Ele espera pedidos dos clientes internos, encaminha os pedidos ao servidor remoto, que está fora da rede interna. Ao receber os dados de resposta do servidor remoto, o procurador entrega os dados ao cliente que os solicitou inicialmente.

Outro uso dos servidores procuradores é para armazenamento temporário das informações solicitadas (técnica conhecida como *cache*). Isto ocorre quando dois ou mais clientes do servidor procurador solicitam as mesmas informações, por exemplo, uma imagem na Web. Neste caso, o servidor recebe a imagem apenas uma vez, na primeira solicitação, e a armazena. Quando outras solicitações forem feitas, ele entrega a imagem que foi armazenada localmente, diminuindo o tráfego externo e o tempo de resposta para os clientes.

2.4.1 Uso de servidores procuradores para anonimato

Pela característica de poder fazer solicitações em nome de outro cliente, um servidor deste tipo pode ser usado para fornecer anonimato aos clientes que o utilizam. Com o uso de servidores procuradores, as solicitações de informação são enviadas tendo como endereço de origem o endereço do servidor procurador. Desta forma, o receptor das solicitações (por exemplo um servidor Web) não tem meios de saber o endereço real do emissor.

Um serviço que tem esta proposta de anonimato é o **Anonymizer**, acessível no sítio www.anonymizer.com. Neste serviço, o usuário digita em um formulário

HTML o endereço desejado, e o sistema faz a consulta para o usuário, enviando-lhe as informações obtidas. Além de repassar as informações, o sistema faz alterações no cabeçalho das mensagens do protocolo HTTP, retirando dados que possam identificar o usuário.

Com relação às propriedades da comunicação anônima, o serviço Anonymizer busca garantir apenas o anonimato do remetente, não tomando medidas para impossibilitar a associação do remetente ao destinatário, o que pode levar à descoberta do remetente caso sejam realizados determinados tipos de ataque.

Conforme descreve Goldberg (2002, p. 6), por possuírem uma arquitetura simplificada, que nem faz uso de criptografia, a utilização de servidores procuradores para anonimato é resistente apenas a modelos fracos de ataques, em que o atacante possui poucos recursos computacionais e de comunicação a seu dispor. Para modelos de ataque mais elaborados, onde o atacante tem mais recursos a seu dispor, a utilização apenas de servidores procuradores não garante o anonimato da comunicação.

2.5 Ataques ao anonimato

Qualquer sistema que se proponha a prover anonimato pode ser genericamente considerado como uma Rede de Comunicação Anônima (RCA), que possui uma infra-estrutura voltada para a solução deste problema. Esta infra-estrutura é composta por pontos de comunicação, os quais são servidores conectados à Internet, e faz uso de mecanismos criptográficos para a obtenção do anonimato. Os ataques ao anonimato são realizados contra uma RCA e seus usuários.

Os ataques ao anonimato em redes de computadores, em geral, não visam apenas a obtenção do conteúdo das mensagens que trafegam na rede, mas principalmente a descoberta da origem e do destino das mesmas. Isto fica claro, por exemplo, quando se deseja realizar um ataque a um usuário da Web com o objetivo de obter uma relação dos sítios por ele visitados. Neste caso não se está interessado no conteúdo das mensagens enviadas, pois sabe-se que se trata de solicitações do protocolo HTTP, textos

em linguagem natural e imagens em formato binário. O objetivo num ataque como este é descobrir quais os sítios visitados pelos usuários da Internet. Com uma informação destas em mãos, empresas de conteúdo e de *marketing* poderiam oferecer produtos para um público selecionado, obtendo lucros através de tal informação. Ao descobrir quem está comunicando, desfaz-se o **anonimato** daquela pessoa, e ao estabelecer uma relação entre ela e os sítios por ela visitados, desfaz-se a **privacidade** da mesma.

Nesta seção serão mostrados os ataques mais comuns ao anonimato, baseados em análise de tráfego, e também serão apresentadas as técnicas comumente utilizadas para a proteção contra estes ataques.

2.5.1 Ataque à codificação da mensagem

O **ataque à codificação da mensagem** (*message coding attack*) é caracterizado pela observação da mensagem ao longo de diversos pontos de uma RCA. Caso a mensagem mantenha sua codificação, ou seja, a forma como os dados estão dispostos dentro do pacote, a mesma poderá ser rastreada desde a origem até o destino (BERTHOLD; FEDERRATH; KÖHNTOPP, 2000, p. 1).

Este ataque é geralmente realizado de forma passiva; o atacante monitora as ligações entre os pontos de rede por onde as mensagens trafegam. Este ataque pode ser bem sucedido mesmo que o atacante não tenha acesso aos pontos intermediários da rede de comunicação. Caso o atacante esteja monitorando diversos emissores/receptores de mensagens, e estes troquem mensagens entre si, o atacante terá meios para desfazer o anonimato na comunicação, descobrindo quem enviou mensagens e para qual destino.

A prevenção deste tipo de ataque exige o uso de criptografia assimétrica na RCA (SONG; KORBA, 2002, p. 5), da seguinte maneira: o emissor de uma mensagem cifra a mesma com a chave pública do receptor e em seguida realiza sucessivas cifrações[1] desta mensagem, tantas vezes quantos forem os pontos da RCA pelo qual a mensagem deverá passar. Por exemplo, se a mensagem for trafegar por 5 pontos antes de chegar ao

[1] De acordo com o Decreto 3.587 de 5 de setembro de 2000 (BRASIL, 2000), a tradução usada para os termos *encryption* e *decryption* devem ser, respectivamente, **cifração** e **decifração**.

receptor, o emissor deve realizar cinco cifrações da mensagem, além daquela destinada ao receptor. Em cada uma das sucessivas cifrações o emissor deve utilizar a chave pública de cada um dos pontos do caminho de rede pelo qual a mensagem trafegará. Cada ponto da RCA, ao receber a mensagem, deve realizar uma decifração da mesma, utilizando sua chave privada, e enviando a mensagem para o próximo ponto da rede. O ponto da rede que antecede o receptor entregará a mensagem com apenas a última decifração a ser realizada, a qual será feita pelo receptor, obtendo assim a mensagem original. Isto garante que em cada ponto do caminho a mensagem terá sua codificação alterada, impossibilitando a realização do ataque à codificação da mensagem.

Com o uso da criptografia assimétrica, o atacante só obteria sucesso se conseguisse ter acesso à todas as chaves privadas de cada ponto por onde a mensagem trafega, o que exigiria o comprometimento de cada servidor da rede.

2.5.2 Ataque de temporização

No **ataque de temporização** (*timing attack*) o atacante monitora apenas os emissores e receptores de mensagens, registrando a ocorrência de início e término de envio de mensagens. Como os possíveis caminhos que as mensagens podem seguir tendem a ter tempos de tráfego constantes, o atacante pode estabelecer uma relação entre o início de uma comunicação por parte do emissor, e a correspondente recepção, com o término da comunicação (RAYMOND, 2000, p. 10). Por exemplo, supondo que (1) o atacante está monitorando a atividade de dois pontos **A** e **B** em uma rede; (2) ele conhece o tempo médio de tráfego de uma mensagem entre os dois pontos; e (3) ele observou que este foi o tempo aproximado que decorreu entre o início de uma comunicação por parte de **A** e o recebimento de uma mensagem por parte de **B**; então ele pode concluir, com alta probabilidade, que o ponto **A** estabeleceu comunicação com o ponto **B**.

Com este tipo de ataque, monitorando diversos pontos terminais em uma rede, e de posse dos valores médios do tempo de tráfego entre estes pontos, o atacante pode estabelecer as relações para as comunicações entre os pontos, e desfazer o anonimato da comunicação (BACK; MÖLLER; STIGLIC, 2001, p. 8).

A prevenção contra o ataque de temporização pode ser feita com algumas técnicas, que dependerão do tipo de aplicação a que uma RCA se propõe. O primeiro ponto a se observar é que uma mensagem pode ser escondida apenas se ela estiver dentro de um grupo de mensagens trafegando pela rede. A relação da comunicação não pode ser protegida caso apenas uma mensagem esteja trafegando em determinado período de tempo.

A primeira medida que uma RCA pode tomar para evitar o ataque de temporização é fazer com que as mensagens trafegando em um ponto da rede sejam enviadas para o próximo ponto em ordem diferente de como foram recebidas (CLEMENTI; IANNI, 1996), esta técnica é conhecida como **armazena-e-encaminha** (*store-and-forward*). Isto fará com que o tempo de tráfego entre dois pontos atendidos pela RCA deixe de ser constante. O atraso no envio de uma mensagem dependerá da quantidade de usuários da RCA e do volume de tráfego por eles produzido.

Para dificultar ainda mais, cada ponto da RCA pode introduzir atrasos aleatórios para o envio de cada mensagem. Estas medidas evitam o ataque de temporização, mas elas são úteis apenas para comunicações que não necessitem de alta velocidade, como, por exemplo, o envio de correio eletrônico (*e-mail*). Neste caso o atraso na comunicação não causa incômodo, pois o emissor da mensagem não espera uma resposta imediata por parte do receptor. Entretanto, para comunicações que necessitam de alta velocidade, como as conversas por IRC (IETF, 1993), ou o acesso a páginas Web, este atraso na comunicação torna-se incômodo pois é esperada uma resposta imediata do receptor da mensagem.

Uma medida que pode ser usada quando se necessita de comunicação bidirecional sem atrasos, é o uso de **mensagens de disfarce** (*dummy messages*). Estas são mensagens enviadas apenas para aumentar o volume de tráfego na RCA; elas não possuem um significado nem carregam informação útil (LEVINE et al., 2004, p. 5). Desta forma, é preciso ter meios para identificar o recebimento de mensagens de disfarce e descartá-las. As mensagens de disfarce são geralmente geradas pelos servidores de uma RCA, mas também podem ser geradas pelos usuários da rede, o que aumenta o nível de anonimato da rede (BERTHOLD; LANGOS, 2002, p. 6).

Segundo Rennhard e Plattner (2003, p. 2), o uso de mensagens de disfarce juntamente com a técnica de armazena-e-encaminha dificulta a realização do ataque de temporização sem inserir grandes atrasos no tráfego das mensagens. Com o uso conjunto destas técnicas, uma rede de comunicação anônima de baixa latência pode ser usada inclusive para a navegação na Web (RENNHARD et al., 2002, p. 3). Entretanto, o uso de mensagens de disfarce deve ser avaliado quanto à sobrecarga de processamento e ao uso de banda, que pode consumir muitos recursos, conforme ressaltam Rackoff e Simon (1993, p. 2).

2.5.3 Ataque de volume de mensagens

Outra forma de se relacionar emissores com receptores de mensagens é através do **ataque de volume de mensagens** (*message volume attack*). Este ataque é feito observando-se a quantidade da dados transmitidos e recebidos pelos usuários de uma RCA através do número de pacotes transmitidos e o tamanho de cada pacote (BACK; MÖLLER; STIGLIC, 2001, p. 7).

Um atacante que esteja observando diversos usuários de uma RCA, ao ver que um usuário emite certo número de mensagens com um determinado tamanho, e ao perceber que um segundo usuário recebe o mesmo número de mensagens e com o mesmo tamanho anterior, pode concluir que aqueles dois usuários estabeleceram uma comunicação. Caso o atacante esteja também monitorando os pontos que constituem a RCA, ele poderá ter mais certeza da relação de comunicação, traçando inclusive o caminho utilizado pelas mensagens, pois ele pode comparar o volume das mensagens que entram em um ponto da rede com o volume daquelas que saem deste mesmo ponto. O sucesso deste ataque acaba por reduzir o anonimato da comunicação entre os usuários da RCA (ZHU et al., 2004, p. 6).

A solução para este problema está no uso de **preenchimento de mensagem** (*message padding*), que consiste em adicionar dados sem valor informativo às mensagens que trafegam na RCA, fazendo com que as mesmas tenham todas tamanho constante (SUN et al., 2002, p. 6). Desta forma torna-se impraticável a correlação entre as

mensagens que entram e as que saem de determinado ponto da rede (ZHU et al., 2004, p. 5). Algumas RCAs impedem este ataque com o uso de mensagens de disfarce, algoritmos de divisão de mensagens, e adequação de tráfego (RENNHARD; PLATTNER, 2003, p. 2).

2.5.4 Ataque de inundação

Uma mensagem pode permanecer anônima desde que esteja trafegando em meio a diversas outras mensagens também anônimas; neste caso os emissores das diversas mensagens formarão o que é conhecido como **conjunto de anonimato** (KESDOGAN; EGNER; BÜSCHKES, 1998, p. 2). Caso se tenha muito pouco tráfego, ou se o restante do tráfego for conhecido, pode-se obter informações sobre a origem e destino da mensagem. O **ataque de inundação** (*flooding attack*) é feito enchendo-se uma RCA com mensagens conhecidas, com o objetivo de separar um certo grupo de mensagens cujo destino se deseja descobrir (SERJANTOV; DINGLEDINE; SYVERSON, 2002, p. 1).

Uma possível situação para este ataque é em RCAs que utilizam a técnica de armazena-e-encaminha. Se o atacante sabe que a RCA espera até que *n* mensagens sejam armazenadas para então encaminhá-las ao próximo ponto da rede, ele pode enviar *n-1* mensagens para um ponto da rede e ficar observando qual será a última mensagem necessária para completar as *n* armazenadas. Como ele sabe o destino de suas mensagens, ao observar o destino da mensagem restante ele pode descobrir o destinatário desta.

Ao fazer a inundação de diversos pontos de uma RCA, um atacante pode detectar aumentos no tráfego entre os pontos da rede e traçar o caminho percorrido pelas mensagens observadas em cada ponto. Com esta inundação de mensagens o atacante consegue quebrar o anonimato de determinadas mensagens, obtendo a relação emissor/receptor das mesmas (RAYMOND, 2000, p. 9).

Uma primeira medida para dificultar a ação do atacante seria buscar manter o tráfego constante entre os pontos da rede com o uso de mensagens de disfarce, tornando difícil a análise do atacante em saber quais são mensagens reais e quais não são (RENNHARD et al., 2002, p. 3).

Outra medida para evitar o ataque de inundação é estabelecer limites

de envio de mensagens por usuário da rede (BERTHOLD; FEDERRATH; KÖHNTOPP, 2000, p. 2). Esta medida tornaria necessária a identificação dos usuários por parte da RCA, o que pode não ser aceitável em situações que se desejar anonimato completo. Nestes casos pode-se ainda introduzir o uso de **bilhetes** (*tickets*), os quais dariam a determinado usuário o direito de utilizar a rede para enviar um certo volume de dados por unidade de tempo. Com o uso de bilhetes pode-se limitar o tráfego dos usuários sem que seja necessária a sua identificação perante a RCA. Este limite no tráfego para os usuários dificulta a ação de um atacante que planeje realizar o ataque de inundação.

Apesar destas medidas dificultarem a realização deste ataque, ele ainda pode ser praticado caso o atacante possua muitos recursos computacionais (SONG; KORBA, 2002, p. 6). Neste caso ele poderia forjar diversas identidades, e se fazer passar por diversos usuários utilizando a rede. Além disso, o controle do tráfego realizado pela RCA tem um custo computacional elevado, principalmente em situações de alto tráfego, o que pode degradar significantemente a performance da rede.

2.5.5 Ataque de cruzamento

Um usuário da Internet pode ter um padrão de comportamento regular nas suas comunicações ao longo do tempo. Por exemplo, quando ele freqüentemente visita os mesmos sítios da Web, ou envia correio eletrônico para um mesmo grupo de pessoas com certa freqüência, ou outras formas freqüentes de comunicação (WRIGHT et al., 2003, p. 6). Este padrão de comportamento torna-se mais fácil de ser identificado quando dados específicos do usuário são transmitidos, tais como os *cookies* (IETF, 2000). Outra informação que contribui para a definição de um padrão de comportamento são os períodos de atividade e inatividade de um usuário.

Um atacante pode observar as informações enviadas pelos usuários da Internet, e observar também os períodos de atividade e inatividade dos mesmos. Ao fazer um cruzamento das informações obtidas ao longo das diversas observações feitas, em um longo período, ele pode traçar a identidade dos usuários associando-os ao tráfego analisado. É este mapeamento das informações dos usuários que caracteriza o **ataque**

de cruzamento (*intersection attack*). Este ataque não permite descobrir com certeza um usuário, mas reduz bastante as possibilidades para o número de participantes de uma comunicação anônima (BERTHOLD; FEDERRATH; KÖHNTOPP, 2000, p. 3). Entretanto, este ataque exige bastante esforço computacional, pois para realizá-lo é necessário monitorar um grande volume de dados e realizar a análise de todo o tráfego obtido.

O uso de mensagens de disfarce reduz as chances de sucesso deste ataque, pois dificulta a análise dos períodos de atividade e inatividade do usuário, que, com o uso desta técnica, estará realizando comunicações na maior parte do tempo (SONG; KORBA, 2002, p. 6). Mas a principal medida a ser tomada para solucionar este problema é o uso de criptografia. Com a cifração dos dados enviados nas mensagens, o atacante não tem meios de obter as informações necessárias para o cruzamento dos dados (SUN et al., 2002, p. 1). Aliado ao uso de mensagens de disfarce, esta medida torna o ataque de cruzamento ineficaz.

2.5.6 Ataque de marcação da mensagem

Se um atacante conseguir comprometer alguns pontos de uma RCA, ele poderá realizar o **ataque de marcação da mensagem** (*message tagging attack*). Este ataque consiste em controlar os dois pontos extremos de uma rota utilizada no envio de uma mensagem, de forma a marcar a mensagem no ponto inicial da rota, ou seja, logo que a rede recebe a mensagem de um usuário, e observar a sua saída no ponto final da rota definida (RAYMOND, 2000, p. 13). Caso o atacante esteja controlando justamente os dois pontos extremos da rota traçada, ele terá como identificar a origem e o destino da mensagem devido à marcação por ele feita na mesma ao entrar na RCA parcialmente comprometida, apesar de o roteamento ter ocorrido normalmente.

Existem duas medidas que podem ser tomadas para evitar o ataque de marcação da mensagem. A primeira seria fazer com que os pontos de uma RCA não tivessem conhecimento completo de todos os outros pontos constituintes da rede, e não obtivessem outra informação sobre a mensagem além do seu endereço de origem (SONG; KORBA, 2002, p. 6). Sabendo apenas o endereço de origem de uma mensagem, cada

ponto não tem como descobrir se ela está vindo de um outro ponto, ou de um usuário. Com esta medida, o atacante, mesmo conseguindo comprometer alguns pontos da RCA, não tem como saber quando aqueles pontos estão sendo utilizados como extremidades da rota das mensagens, ou como pontos intermediários desta rota. Isto reduz bastante as possibilidades de sucesso deste ataque, pois ao marcar mensagens e ao observar a passagem de mensagens marcadas, o atacante não pode ter certeza de que se trata de um usuário emissor/receptor ou se se trata de pontos que fazem parte da rota.

A segunda medida para evitar o ataque é criando meios para que seja impossível realizar a marcação da mensagem, ou que, ao realizar uma marcação, esta seja facilmente descoberta e descartada. Para tanto pode-se utilizar técnicas de criptografia, como sugerem Berthold, Federrath e Köhntopp (2000, p. 2). Desta maneira, com as mensagens trafegando de forma cifrada, qualquer tentativa de alteração/marcação pode ser detectada pelos pontos da rede, pois para realizar uma alteração sem que esta seja detectada, o atacante deve possuir as chaves de criptografia utilizadas no processo de cifração.

2.5.7 Ataque de repetição da mensagem

Um atacante que esteja observando um ponto da rede pode criar cópias de uma mensagem que entre naquele ponto e depois observar as cópias da mensagem na saída daquele ponto da rede, observando assim qual o próximo ponto a que elas se dirigem. Com este **ataque de repetição da mensagem** (*message replay attack*) o atacante pode obter informações da rota percorrida por uma mensagem até chegar no seu destino. Desta forma o atacante pode encontrar a relação emissor/receptor, quebrando o anonimato das mensagens (RAYMOND, 2000, p. 13).

Este ataque apresenta um risco maior porque não necessita que o atacante comprometa pontos da rede. Apenas com a monitoração dos canais de comunicação é possível realizar este ataque.

Uma prevenção para este tipo de ataque consiste em atribuir uma espécie de número serial, de uso único (*nonce*), a cada mensagem enviada (SONG; KORBA,

2002, p. 7). Desta forma, ao receber uma mensagem, cada ponto da rede registra o número serial daquela mensagem em uma lista interna antes de encaminhá-la e caso o número serial daquela mensagem já esteja registrado na lista, o ponto da rede pode então identificar que se trata de uma mensagem repetida, descartando-a.

Como no ataque de repetição da mensagem é necessário que as cópias da mensagem sejam enviadas em um curto espaço de tempo para ter sucesso (de forma que não se diluam em meio ao tráfego total), a lista de registro de números seriais de um ponto da rede pode ter um tamanho pré-determinado, de acordo com a quantidade de tráfego esperada. Com o tamanho da lista fixo, números antigos podem ser descartados, pois se referem a mensagens também antigas. Assim, a verificação dos números seriais antes do encaminhamento da mensagem não causa perdas significativas na performance da RCA, o que não seria verdade caso a lista crescesse indefinidamente.

Outra medida para evitar que uma mensagem seja processada mais de uma vez é o uso de **carimbo de tempo** (*time-stamp*). Nesta técnica, cada mensagem recebe uma marcação indicando o tempo de validade da mesma, desta forma uma mensagem é processada apenas se estiver dentro do período de tempo indicado na marcação feita ao ser enviada (SONG; KORBA, 2002, p. 7). Entretanto, esta técnica exige a sincronização dos relógios dos elementos da RCA.

2.5.8 Ataque de negação de serviço

O **ataque de negação de serviço** (*denial of service attack*) é feito tornando-se alguns dos pontos de uma RCA inoperantes através do envio de uma quantidade de informação bem maior do que cada ponto é capaz de processar. Como a quantidade de rotas possíveis diminui com um número reduzido de pontos operantes, o atacante possui mais chances de descobrir uma relação emissor/receptor (RAYMOND, 2000, p. 12). Entretanto, a realização de um ataque deste tipo demanda bastante recursos computacionais, além da necessidade do comprometimento de diversos computadores ligados a Internet, que servem como ferramentas para a realização de um ataque deste tipo.

Assim como os ataques de negação de serviço que ocorrem na Internet

em geral, este ataque ao anonimato não possui medidas eficazes que evitem a sua realização. A prevenção a um ataque deste tipo consistiria em detectar um aumento muito grande no fluxo de dados e, a partir deste ponto, começar a rejeitar pacotes de dados. Entretanto, este procedimento consome uma quantidade muito grande de recursos computacionais, o que pode inviabilizar sua realização.

2.5.9 Ataque do predecessor

O **ataque do predecessor** (*predecessor attack*) tem como objetivo descobrir quem é o emissor de uma mensagem anônima (REITER; RUBIN, 1998, p. 12). Para tanto, o atacante precisa comprometer um ou mais servidores da rede atacada. Outra situação necessária para a realização deste ataque é que a comunicação entre o emissor e o receptor das mensagens deve ocorrer durante um período de tempo suficiente para que a RCA realize diversos ciclos de entrega de mensagens (WRIGHT et al., 2002).

As técnicas de defesa contra ataques ao anonimato empregadas nos demais ataques não têm efeito contra o ataque do predecessor. Entretanto, a realização deste ataque consome uma grande quantidade de recursos, o que dificulta sua realização (WRIGHT et al., 2002, p. 1).

2.5.10 Ataque da descoberta

Conforme Agrawal, Kesdogan e Penz (2003), o **ataque da descoberta** (*disclosure attack*) consiste em se observar o conjunto de receptores de mensagens em uma RCA ao longo do tempo, verificando variações na sua composição. Fazendo a intersecção das diversas observações dos conjuntos ao longo do tempo, um atacante pode ter meios de identificar os remetentes de mensagens, e os respectivos destinatários. Entretanto, como o atacante não tem meios de observar todos os remetentes e destinatários, a possibilidade de descoberta de uma comunicação passa a ser dada por uma probabilidade. Conforme demonstra o trabalho de Agrawal, Kesdogan e Penz (2003), esta probabilidade pode ser estimada a partir do número de observações que o atacante consegue realizar.

Apesar de o ataque permitir a descoberta de comunicações através da rede, sua aplicação é bastante inviável devido ao excesso de processamento necessário (AGRAWAL; KESDOGAN; PENZ, 2003, p. 11).

2.6 Conclusão

Assim como ocorre na vida real, no mundo digital o uso do anonimato também pode trazer vantagens para os usuários da Internet, como por exemplo a melhoria da privacidade. As motivações para a obtenção do anonimato na comunicação são diversas. Entretanto, quando se deseja uma comunicação deste tipo, é preciso definir corretamente quais dados devem ser mantidos anônimos. Este capítulo apresentou as propriedades da comunicação anônima que podem ser obtidas.

Tabela 2.1: Tipos de ataque ao anonimato e mecanismos de defesa

Tipo de Ataque	**Mecanismos de defesa aplicáveis**
Ataque à codificação da mensagem	Criptografia assimétrica
Ataque de temporização	Técnica armazena-e-encaminha (*store-and-forward*) Inserção de atrasos aleatórios Mensagens de disfarce (*dummy messages*)
Ataque de volume de mensagens	Preenchimento de mensagem (*message padding*)
Ataque de inundação	Mensagens de disfarce Limitação de tráfego por usuário
Ataque de cruzamento	Mensagens de disfarce Criptografia simétrica
Ataque de marcação da mensagem	Criptografia simétrica
Ataque de repetição da mensagem	Mensagens com número serial Marcas de tempo (*time-stamp*).
Ataque de negação de serviço	Monitoramento do fluxo de dados
Ataque do predecessor	*Não há medidas viáveis*
Ataque da descoberta	*Não há medidas viáveis*

Estas propriedades são elementos fundamentais para o estudo e compreensão das técnicas para comunicação anônima encontradas atualmente. Os ataques ao anonimato apresentados auxiliam na análise destas técnicas quanto ao cumprimento dos requisitos de segurança envolvidos na garantia do anonimato. A tabela 2.1 sintetiza os ataques ao anonimato apresentados neste capítulo e os possíveis mecanismos de defesa

para estes ataques.

As formas apresentadas neste capítulo para o envio anônimo de correio eletrônico e a utilização de servidores procuradores para navegação anônima na Web mostram a necessidade de sistemas dedicados à garantia do anonimato da comunicação em rede. E como exemplificado em algumas das técnicas para correio eletrônico anônimo, a criptografia é uma ferramenta indispensável para esta garantia.

Capítulo 3

Redes de Comunicação Anônima

3.1 Introdução

Este capítulo apresenta (na seção 3.2) os conceitos envolvidos em uma Rede de Comunicação Anônima (RCA), que é a forma como se apresentam as principais técnicas para comunicação anônima em redes de computadores encontradas atualmente.

Além dos conceitos envolvidos em uma RCA, este capítulo também apresenta as técnicas para comunicação anônima estudadas. Para facilitar a compreensão das técnicas aqui expostas, são apresentadas duas RCAs que fazem uso destes procedimentos de segurança, e cujas formas de operação servem de base para outras RCAs existentes. Para cada uma das RCAs é apresentado o funcionamento do sistema, com as devidas particularidades, e é feita uma análise da segurança de cada rede. A escolha por estas duas RCAs se deu por elas diferirem em duas características importantes: a forma como as mensagens são tratadas pela rede, e a latência da comunicação inserida pela rede. Estas características também permitem classificar as técnicas para comunicação anônima, pois, por exemplo, algumas se destinam à redes de alta latência, enquanto que outras se destinam à redes de baixa latência.

A primeira rede (Roteamento de Cebolas) é descrita na seção 3.3, e a segunda (Rede de Mistura) é descrita na seção 3.4. A seção 3.5 conclui o capítulo.

3.2 Rede de Comunicação Anônima

As redes de comunicação anônima visam impedir a obtenção de informação privada, a qual pode ser obtida através do monitoramento da comunicação ou através de análise de tráfego (SONG; KORBA, 2001, p. 2). Um sistema deste tipo tem como objetivo tornar impossível a obtenção, por um atacante, de informação valiosa sobre qualquer relação de comunicação ou sobre qualquer pedido de comunicação de, e para, um usuário (BERTHOLD; FEDERRATH; KÖHNTOPP, 2000, p. 1).

3.2.1 Atuação em camadas de rede

Uma rede de comunicação anônima pode atuar em diversas camadas de rede, interagindo com os protocolos adequados, presentes em cada camada. Atualmente se encontram RCAs sendo aplicadas na **camada de rede** propriamente dita, na **camada de transporte**, e na **camada de aplicação**.

Um exemplo de atuação na **camada de rede** é o sistema **Tarzan** (FREEDMAN; MORRIS, 2002). Uma das técnicas utilizadas nesta RCA é a tradução de endereços de rede - **NAT** (IETF, 1994), que permite a reutilização de endereços IP em diferentes redes locais. As RCAs que atuam na **camada de transporte** fazem uso da técnica de servidores procuradores (veja seção 2.4, na página 18) de forma a permitir o anonimato de forma transparente, sem necessidade de alterações em aplicativos já existentes. Um exemplo de atuação na camada de transporte é o sistema **Crowds** (REITER; RUBIN, 1998), que tem o objetivo de prover navegação anônima na Web, e age como um servidor procurador para navegadores Web. As RCAs que atuam na **camada de aplicação** são dedicadas a contextos específicos, buscando fornecer anonimato para determinadas aplicações, e geralmente exigindo clientes específicos, que possam realizar as operações necessárias. Um exemplo é o sistema **Mixminion** (DANEZIS; DINGLEDINE; MATHEWSON, 2003), que é dedicado ao envio de correio eletrônico anônimo.

Especificamente para a atuação na camada de rede, uma técnica que apresentaria possibilidades de uso em redes de comunicação anônima seria a **Arquitetura**

de Segurança para o Protocolo da Internet, também conhecida como **IPSec** (IETF, 1998b). Entretanto, esta técnica não é comumente encontrada em RCAs. Um dos motivos é a incompatibilidade com algumas técnicas muito presentes em redes locais, como o **NAT** (IETF, 2004). Uma possibilidade para a solução destas incompatibilidades é a versão 6 do protocolo IP, **IPv6** (IETF, 1998a), que possui procedimentos de segurança como parte integrante do protocolo, e não apenas através de extensões, como ocorre na versão 4 do protocolo IP. Entretanto, com a lenta adoção do IPv6, ainda não existem RCAs que fazem uso desta versão do protocolo.

Outro problema que ocorre com a utilização de RCAs diretamente na camada de rede, é que neste contexto apenas o endereço IP dos usuários da rede fica anônimo, ou seja, neste caso a RCA não se preocupa com o conteúdo das mensagens enviadas, que pode conter informações que identifiquem o remetente perante o destinatário. Desta forma, é preciso que as RCAs também possuam mecanismos de verificação do conteúdo das mensagens para garantir o anonimato dos usuários. Esta característica é obtida com maior facilidade quando a RCA atua na camada de transporte ou de aplicação.

3.2.2 Cenários de utilização

Os usuários de uma RCA (emissores/receptores de mensagens) podem ser tanto clientes quanto servidores de dados. Desta forma, tem-se os seguintes possíveis cenários de utilização de uma RCA:

- **Cliente/Servidor** - Este é o cenário mais comum para o uso de uma RCA. Aqui clientes e servidores trocam mensagens entre si de forma anônima. Um exemplo deste cenário é a navegação por páginas Web, onde primeiramente um cliente de dados (um software navegador Web) envia mensagens através da RCA para um servidor de dados (um servidor de páginas), o qual responde ao cliente, também através da RCA, enviando os dados solicitados pelo mesmo;

- **Cliente/Cliente** - Neste cenário, apenas usuários clientes trocam mensagens entre si de forma anônima. Um exemplo deste cenário é a utilização dos serviços de

mensagens instantâneas, onde dois clientes (com um software de mensagens instantâneas) trocam mensagens entre si de forma anônima, fazendo uso de uma RCA. Apesar de ser menos encontrado do que o anterior, este cenário também é bastante comum no uso de RCAs;

- **Servidor/Servidor** - Este cenário prevê que apenas usuários servidores troquem mensagens entre si de forma anônima. Um caso de uso neste cenário é encontrado em sistemas B2B[1] onde usuários servidores (um software específico para determinado negócio) trocam mensagens entre si, geralmente em horários pré-estabelecidos, com objetivo de concluir operações relativas ao negócio de determinada empresa. Apesar de ainda ser menos comum, este também é um cenário que pode ser encontrado no uso de RCAs.

A grande utilidade de uma rede de comunicação anônima é que ela oferece a possibilidade de esconder alguns dados sobre o conteúdo das comunicações realizadas na Internet (SONG; KORBA, 2001, p. 9).

É importante que uma rede de comunicação anônima tenha a propriedade de que, se pelo menos um ponto da rede estiver íntegro, então toda a comunicação realizada através da rede estará segura (CHAUM, 1981, p. 2). Desta forma, para um atacante conseguir comprometer a infra-estrutura da RCA ele terá que comprometer todos os servidores que a compõem. Por esta característica, uma RCA será sempre um sistema distribuído.

3.2.3 Latência da comunicação

Uma questão a ser considerada na utilização de RCAs diz respeito ao tempo que as mensagens anônimas levam para chegar ao destino. Por fazer com que a mensagem passe por diversos servidores, que muitas vezes também realizam diversas operações sobre as mensagens, uma RCA introduz atrasos na entrega de mensagens e na

[1] *Business-to-Business* (Empresa-para-Empresa) - Denominação dada aos sistemas de informação utilizados para a realização de negócios entre empresas através da Internet.

entrega de eventuais respostas. No caso da navegação anônima este atraso deve ser o menor possível, enquanto que para o envio de correio eletrônico anônimo até mesmo atrasos de minutos são toleráveis, pois o usuário não espera resposta imediata. Esta característica das RCAs é comumente denominada **latência**, sendo utilizada como um dos parâmetros para análise dessas redes (WRIGHT et al., 2002). Assim, as RCAs podem ser classificadas como de baixa latência (*low-latency*), ou de alta latência (*high-latency*).

Este atraso na comunicação introduzido por uma RCA deve ser levado em consideração no momento de decidir a utilizar determinada rede, pois torna-se difícil a obtenção de uma sistema de baixa latência que possua um alto nível de anonimato na comunicação. Se for necessária a velocidade na troca de informações, deve-se optar por uma rede de baixa latência, sabendo que neste caso o nível de anonimato que se consegue é menor do que o obtido em uma rede de alta latência. Considera-se inclusive que a obtenção de uma RCA prática, para utilização em aplicações que exigem baixa latência, que suporte uma grande quantidade de usuários, e que ofereça anonimato total é algo extremamente difícil e pode não ser possível (RENNHARD; PLATTNER, 2003, p. 3).

3.2.4 Arquitetura da rede

Nos serviços comumente encontrados na Internet, os servidores têm autonomia no que diz respeito ao provimento do serviço para seus clientes, ou seja, raramente um servidor depende de outro para prover o serviço aos seus clientes. Isto não ocorre em uma RCA, pois as mensagens anônimas precisam passar por diversos servidores integrantes da rede. Desta forma, os servidores não são autônomos para o provimento do serviço, e dependem de outros servidores da RCA para tanto.

Quanto à conexão dos servidores, duas arquiteturas são mais utilizadas:

- **Arquitetura em cadeia**: Nesta estrutura os servidores da rede estão interconectados para prover o serviço, mas cada servidor possui no máximo duas conexões constantemente ativas, independentemente do número de servidores participantes. O primeiro servidor da cadeia conecta-se apenas à seu sucessor, o último conecta-se

apenas à seu predecessor, e os servidores intermediários são os que possuem duas conexões, com seus respectivos sucessor e predecessor. Desta forma apenas o primeiro servidor da cadeia recebe mensagens de clientes, e apenas o último servidor da cadeia entrega as mensagens aos seus destinatários. Esta é a arquitetura utilizada na proposta inicial de Chaum (1981), da Rede de Mistura (descrita na seção 3.4, na página 59);

- **Arquitetura em clique**: Nesta estrutura, para n servidores participantes de uma RCA, cada servidor possui $n - 1$ conexões constantemente ativas. Estas conexões formam uma estrutura que na Teoria dos Grafos é denominada **clique**, pois cada servidor está conectado aos demais servidores integrantes da RCA. Nesta forma de arquitetura cada servidor pode tanto receber mensagens de remetentes, quanto entregar mensagens a destinatários, além de também ser um servidor intermediário. Quando se faz uso desta arquitetura o consumo de recursos e a complexidade da implementação tornam-se maiores do que na arquitetura em cadeia. Diferentemente do que ocorre na arquitetura em cadeia, onde o caminho que as mensagens seguem entre os servidores é sempre o mesmo, na arquitetura em clique o usuário pode escolher por qual caminho sua mensagem deverá trafegar. Conforme Syverson, Goldschlag e Reed (1997), esta é a arquitetura utilizada no Roteamento de Cebolas (esta RCA é descrita na seção 3.3, na página 45).

Berthold, Pfitzmann e Standtke (2000) fazem uma análise destas duas arquiteturas, buscando descrever as vantagens e desvantagens envolvidas no uso de cada arquitetura. Segundo estes autores, existem ataques que possuem maior probabilidade de sucesso quando realizados em redes com arquitetura em cadeia.

Danezis (2003) propõe uma arquitetura intermediária, denominada **arquitetura esparsa**, na qual cada servidor comunica-se apenas com alguns dos outros servidores integrantes da rede, formando um grafo com agregados de nodos onde a comunicação entre os agregados é feita por apenas algumas arestas. Esta arquitetura é mais conexa do que a arquitetura em cadeia, mas não possui a completa conexão de todos os nodos, como ocorre na arquitetura em clique. O estudo de Danezis (2003) mostra que a

arquitetura esparsa, quando bem utilizada, apresenta as propriedades desejadas contra os ataques de análise de tráfego, e torna-se mais escalável do que as arquiteturas em cadeia e em clique (DANEZIS, 2003, p. 15).

Mesmo diferindo quanto ao número de servidores disponíveis para receber mensagens dos clientes (apenas 1 na arquitetura em cadeia, e n na arquitetura em clique), ambas as arquiteturas apresentam o mesmo nível de segurança quanto à disponibilidade do serviço. Basta que um servidor seja retirado da rede para que todo o serviço torne-se inoperante. A proteção para esta vulnerabilidade pode ser feita de duas formas:

- **Redundância de servidor**: Esta proteção pode ser aplicada à qualquer uma das arquiteturas, e consiste em se ter mais de um servidor agindo como se fosse o mesmo. Por exemplo, pode-se ter três servidores utilizando os mesmos dados de configuração e participando da rede. Caso um deles seja retirado da rede, os outros dois têm condições de assumir o seu lugar, dando continuidade ao funcionamento da rede;

- **Caminhos reduzidos**: Esta proteção é aplicável apenas à arquitetura em clique, e consiste em dar ao usuário a possibilidade de escolher o tamanho do caminho percorrido por suas mensagens. Com esta proteção, caso algum servidor saia da rede, apenas os caminhos que contiverem aquele servidor ficarão inoperantes. Se as duas formas de proteção forem aplicadas em conjunto, a disponibilidade da rede torna-se ainda maior.

Conforme Dingledine, Shmatikov e Syverson (2004), a utilização de caminhos reduzidos melhora a confiabilidade da rede na entrega das mensagens na ocorrência de falhas nos servidores componentes da rede.

Uma possibilidade para a implementação da redundância de servidor é a utilização da técnica de compartilhamento de segredo (SHAMIR, 1979). Esta técnica permite a divisão de uma informação em diversos pedaços de forma que a informação original pode ser reconstruída conhecendo-se apenas uma quantidade pré-definida dos pedaços criados. Pode-se utilizar esta propriedade para compartilhar a chave criptográfica dos servidores da rede, e caso algum deles deixe de operar corretamente, será possível

reconstruir a chave criptográfica e permitir que outro servidor atue de forma redundante, substituindo aquele que falhou.

3.2.5 Servidores dinâmicos

Seja qual for a arquitetura da rede de comunicação anônima, a situação mais comum para a operação da rede é mantê-la com servidores estáticos, ou seja, uma vez configurado o conjunto de servidores integrantes da rede, este permanece sempre o mesmo, enquanto a rede se mantiver operante. Esta configuração estática pode trazer dificuldades para a operação da rede, principalmente com relação aos custos operacionais, conforme ressalta Goldberg (2002, p. 7).

Uma alternativa de operação para as RCAs é a utilização de uma configuração com servidores dinâmicos, ou seja, ao longo do tempo, com a operação da rede, o conjunto de servidores integrantes da rede pode ter a sua composição variada. Em especial, se cada usuário tiver a oportunidade de também atuar como um servidor, ocorre uma configuração conhecida como redes P2P[2]. Nestas redes, além de não ser necessária a utilização de servidores estáticos, a existência de servidores intermediários só é necessária para informação sobre a entrada e saída de usuários na rede. O anonimato provido pela rede torna-se uma conseqüência da interação direta entre os usuários.

Um exemplo de rede de comunicação anônima que utiliza a configuração P2P é o sistema **Crowds** (REITER; RUBIN, 1998), que é utilizado para navegação na Web. Nesta RCA, o anonimato é provido através da colocação do usuário dentro de uma multidão (*crowd*) de outros usuários, onde as solicitações de informação são enviadas através de vários integrantes da multidão, de forma que não se saiba qual deles foi o remetente das mensagens. O sistema Crowds possui um servidor central que autoriza a entrada de novos usuários na multidão, e avisa os demais usuários sobre a chegada ou saída de usuários.

Rennhard e Plattner (2004, p. 16) afirmam que a utilização do servidor central, presente no sistema Crowds, consiste em uma desvantagem. Um problema

[2] *Peer-to-Peer* (Colega-para-Colega) - Denota a comunicação direta realizada entre os usuários da rede.

apresentado é que este servidor central constitui um único ponto de falha, e que pode ser atacado até mesmo por um atacante que possua poucos recursos. Outro problema apresentado por Rennhard e Plattner é que, como o servidor precisa avisar todos os usuários sobre mudanças na rede, isto prejudica a escalabilidade, pois grandes variações na quantidade de usuários degradam consideravelmente a performance do servidor central.

Outra rede de comunicação anônima que utiliza a configuração P2P foi proposta por Freedman e Morris (2002), denominada **Tarzan**. Esta RCA opera diretamente na camada de rede. Desta forma, o serviço de anonimato é provido de forma transparente para as aplicações de comunicação em rede. A operação da rede consiste em estabelecer um circuito virtual anônimo através de uma seqüência ordenada de usuários da rede. Para a escolha de quais usuários irão compor o circuito virtual, Freedman e Morris propõem um procedimento que limita esta escolha a determinados usuários, dependendo da topologia da rede no momento da construção do caminho. Esta medida visa combater a ação de nodos da rede que estejam sendo operados por atacantes. Outra medida utilizada na rede Tarzan para combate aos ataques ao anonimato é o uso de mensagens de disfarce. Conforme a análise da performance feita por Freedman e Morris (2002, p. 11), o atraso introduzido na entrega de mensagens é pequeno, possibilitando o uso desta RCA em aplicações com requisitos de baixa latência na comunicação.

Um problema que surge com a utilização de configurações P2P é a necessidade de confiança nos integrantes da rede, pois, como cada usuário pode entrar livremente e tornar-se um novo servidor, isso facilita a ação de atacantes. Uma solução para este problema é a utilização do conceito de reputação aplicado aos integrantes da rede. Dingledine, Mathewson e Syverson (2003) propõem um mecanismo para se avaliar a reputação dos servidores de uma RCA através da realização de verificações aleatórias nas operações realizadas pelos servidores. Com as provas de operação correta fornecidas por cada servidor a um remetente de mensagens, este usuário torna-se uma testemunha de que determinado servidor é honesto, podendo difundir para outros usuários esta boa reputação do servidor. Com esta difusão de informações sobre reputação, cada usuário pode estabelecer uma relação de confiança com cada servidor.

3.2.6 Quantificação do anonimato

Como o objetivo principal de uma RCA é prover comunicação com anonimato, torna-se necessária a existência de uma forma de se quantificar o anonimato provido por determinada rede, para se poder saber se o objetivo está sendo alcançado, e qual o nível de vulnerabilidade a ataques que a rede possui. Esta quantificação do anonimato é complexa, e as características das RCAs influenciam nos procedimentos utilizados para medição. Nos trabalhos sobre anonimato que tratam sobre a quantificação do anonimato, encontra-se dois principais tipos de medida: medidas discretas e medidas probabilísticas.

Reiter e Rubin (1998, p. 68-69) propõem uma medida discreta para o anonimato, denominada **graus de anonimato** (*degrees of anonymity*). Esta medida varia desde "anonimato absoluto", onde o atacante não consegue nem ao menos detectar a existência de uma comunicação, até "exposição provável", onde o atacante consegue inclusive provar quem é o remetente, o destinatário, ou a relação entre eles.

Shields e Levine (2000) aprimoraram as definições feitas por Reiter e Rubin, propondo uma medida probabilística, e adicionando intervalos de valores de probabilidade às medidas discretas definidas. Esta medida proposta consiste em calcular a probabilidade de que um usuário x, dentre todos os usuário de uma RCA, seja o iniciador de uma rota de comunicação desta RCA. Caso entre os usuários da RCA exista um ou mais atacantes, estes podem acumular informações a respeito da criação e destruição de rotas, aumentando a possibilidade de descobrir os iniciantes das rotas e, conseqüentemente, obtendo valores diferentes para as probabilidades. Realizando variações no número de usuários da rede, e no número de usuários que são atacantes, Shields e Levine definem os limites de probabilidade que podem ser associados às medidas discretas definidas por Reiter e Rubin.

Outra proposta de medida probabilística para quantificação do anonimato foi feita por Guan et al. (2002). Para a quantificação do anonimato, Guan et al. se baseiam em informações sobre a forma de operação da RCA, e em eventos observados durante a operação da rede. As informações sobre a RCA consistem em saber o algoritmo de seleção de rota e, conseqüentemente, a distribuição do tamanho da rota. Os eventos

são observados através do monitoramento do nodos comprometidos, contendo informações sobre o momento em que uma mensagem chegou ao nodo, qual o predecessor do nodo, e qual o sucessor. Com a variação do tamanho das rotas na RCA, o número de eventos observados também será variável, o que exerce influência na quantificação do anonimato. Com estas informações, pode-se calcular a probabilidade de que cada nodo da rede seja o iniciador de uma comunicação. Com os cálculos de probabilidade considerando variações do tamanho da rota, Guan et al. obtiveram uma medida probabilística para quantificação do anonimato, que foi denominada **grau de anonimato** (*anonymity degree*), que representa o anonimato médio geral obtido no sistema (GUAN et al., 2002, p. 5).

Freedman e Morris (2002) também propõem uma medida probabilística para quantificação do anonimato, considerando uma RCA que utilize servidores dinâmicos, em que cada usuário também atua como nodo da rede. Esta medida baseia-se na incerteza que um atacante tem de que seu predecessor seja o iniciante de uma rota. Como ao se iniciar uma rota, é utilizado um número aleatório de nodos para a compor, a definição de quem é o iniciante, por parte do nodo atacante, torna-se uma probabilidade a ser calculada. Para o cálculo desta probabilidade também é levada em consideração a distribuição de probabilidade associada à variação do comprimento da rota utilizada na RCA.

3.2.7 Formas de operação

Ao se analisar as propostas de redes de comunicação anônima encontradas atualmente, observa-se que muitas delas utilizam formas semelhantes de operação, e dentre estas formas existem algumas que aparecem com maior freqüência. A definição de uma forma de operação depende de alguns fatores, tais como o tipo de informação que está sendo transmitida, o meio de comunicação utilizado, e o ambiente em que a RCA está sendo utilizada. A escolha de uma forma de operação também influencia nas possibilidades de uso das técnicas de comunicação anônima, pois algumas formas de operação impedem o uso de determinadas técnicas.

Uma das principais características que diferem entre as formas de operação é o procedimento utilizado para o tratamento das mensagens enviadas através de uma RCA. Dois principais procedimentos são encontrados: a criação de circuitos virtuais, e a reordenação. Em uma RCA cuja forma de operação utiliza a criação de circuitos virtuais para tratamento das mensagens, busca-se garantir o anonimato mantendo indisponíveis as informações sobre as rotas existentes, utilizadas pelos circuitos virtuais. O segundo procedimento de tratamento de mensagens geralmente encontrado em RCAs, o da reordenação, visa garantir o anonimato de outra forma. RCAs que utilizam este procedimento realizam uma alteração na ordem das mensagens ao saírem de cada ponto da rede, fazendo com que elas sejam encaminhadas em uma ordem diferente da que foram recebidas.

Outra característica principal que difere entre as formas de operação de RCAs é a latência da comunicação presente na RCA (veja seção 3.2.3, na página 35). Dependendo da latência existente em uma forma de operação, a RCA será destinada ao uso em determinadas situações, que exijam baixa ou alta latência. Esta característica está relacionada com o procedimento utilizado para o tratamento das mensagens enviadas através de uma RCA. Em geral, uma RCA cuja forma de operação faça uso de circuitos virtuais terá baixa latência, enquanto que uma forma de operação que utilize a reordenação terá alta latência. Entretanto, existem formas de operação em que esta relação não se aplica.

Como o presente trabalho tem o objetivo principal de descrever as técnicas estudadas para a solução do problema do anonimato em redes de computadores, e como as possíveis formas de operação de uma RCA influenciam nas técnicas de anonimato que podem ser empregadas, optou-se por apresentar em detalhes neste capítulo duas RCAs que possuem formas de operação diferentes, e que permitem empregar uma grande quantidade de técnicas de anonimato. Estas RCAs são: **Roteamento de Cebolas**, apresentada na seção 3.3 (página 45); e **Rede de Mistura**, apresentada na seção 3.4 (página 59).

Além de utilizarem formas de operação diferentes, estas RCAs servem como base para outras RCAs estudadas, por terem sido as primeiras a empregar a res-

pectiva forma de operação. As seguintes RCAs possuem uma forma de operação semelhante ao Roteamento de Cebolas: Crowds (REITER; RUBIN, 1998), Tarzan (FREEDMAN; MORRIS, 2002) e Tor (DINGLEDINE; MATHEWSON; SYVERSON, 2004). RCAs que possuem uma forma de operação semelhante à Rede de Mistura são: Babel (GÜLCÜ; TSUDIK, 1996), MorphMix (RENNHARD; PLATTNER, 2002) e Mixminion (DANEZIS; DINGLEDINE; MATHEWSON, 2003).

Todas as RCAs encontradas atualmente buscam garantir o anonimato da comunicação através de um mesmo princípio: definir uma rota diferente entre o remetente e o destinatário de mensagens, de forma a esconder esta informação de atacantes. Chaum (1988) propôs uma forma diferente para obtenção de anonimato, que não se baseia no estabelecimento de rotas alternativas. Na proposta de Chaum, cada dupla de usuários deve compartilhar uma informação binária obtida de forma aleatória, semelhante ao que ocorre com o lançamento de uma moeda do tipo cara-ou-coroa. Com o anúncio dos resultados aleatórios observados por cada usuário, deve-se obter ao final um valor par. Caso um usuário deseje enviar uma mensagem, ele deve informar o valor oposto ao obtido aleatoriamente. Neste caso, o valor total dos resultados observados será ímpar, o que indicará uma comunicação de algum usuário, porém sem se saber qual deles. Além de Chaum, Waidner e Pfitzmann (1990) também fazem uma descrição detalhada do funcionamento deste procedimento. A vantagem deste procedimento sobre a definição de rotas diferentes entre o remetente e o destinatário, é a de não incluir atrasos na comunicação nem tráfego excessivo. Entretanto, ele depende de um mecanismo de propagação das informações para todos os usuários, que torna-se excessivamente dispendioso com o aumento do número de usuários. Guan et al. (2002), em sua análise de protocolos de comunicação anônima, descreve que a falta de escalabilidade deste procedimento é que faz com que ele não seja utilizado em RCAs.

3.3 Roteamento de Cebolas

Com o objetivo de limitar as vulnerabilidades de uma rede à análise de tráfego, foi proposta e implementada por Goldschlag, Reed e Syverson (1996) uma arquitetura denominada **Roteamento de Cebolas** (*Onion Routing*).

Esta rede de comunicação anônima constrói um circuito virtual anônimo e bi-direcional entre duas entidades que se comunicam, denominadas **iniciador** (*initiator*) e **respondedor** (*responder*).

A garantia de que as informações sobre a rota percorrida por uma mensagem serão mantidas em segredo baseia-se principalmente no fato de que cada nodo de roteamento em um circuito virtual conhece o endereço apenas dos nodos adjacentes, e no fato de que cada nodo cifra a mensagem ao longo do circuito virtual.

Em outro trabalho sobre o Roteamento de Cebolas, Reed, Syverson e Goldschlag (1996) ressaltam que o objetivo desta RCA é prover conexões anônimas. Deseja-se garantir que a comunicação entre o iniciador e o respondedor seja realizada sem o conhecimento de terceiros, mas com a identificação entre ambos podendo ser feita, pois os participantes não precisam necessariamente permanecer anônimos entre si. Desta forma, com relação às propriedades da comunicação anônima (veja seção 2.2, na página 12), o Roteamento de Cebolas busca apenas impossibilitar a associação do remetente ao destinatário, o anonimato do remetente e o anonimato do destinatário devem ser tratados pelo sistema que fizer uso desta RCA, limitando as informações que são disponibilizadas entre eles.

Esta RCA faz uso de servidores procuradores (veja seção 2.4, na página 18) como elementos básicos para fornecer uma conexão anônima. Como existem diversos protocolos com suporte a servidores procuradores, esta RCA pode ser utilizada pelas aplicações que fazem uso destes protocolos (GOLDSCHLAG; REED; SYVERSON, 1996, p. 1).

Para que as informações do roteamento das mensagens sejam escondidas, o fluxo de dados passa por uma rota constituída de diversos servidores até chegar ao destino. Quem define como será composto esta rota é o primeiro servidor utilizado pelo iniciador, o qual age como um servidor procurador para o serviço solicitado. Por exem-

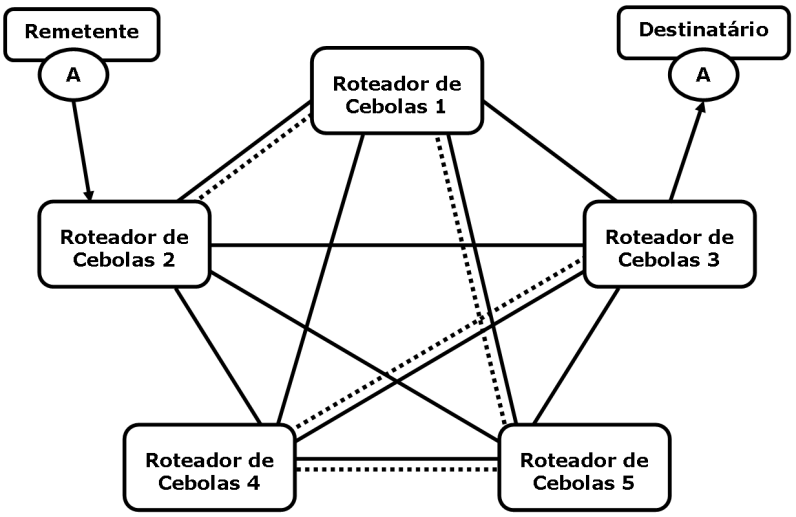

Figura 3.1: Funcionamento do Roteamento de Cebolas

plo, para o protocolo HTTP (para acesso à páginas Web), o primeiro servidor da RCA agirá como um procurador HTTP (*HTTP proxy*) para o usuário iniciador.

A figura 3.1 ilustra o funcionamento do Roteamento de Cebolas. Uma mensagem é enviada pelo remetente através do Roteamento de Cebolas com o uso de um servidor procurador (no caso o roteador de cebolas 2). Este servidor define a rota a ser percorrida pela mensagem (indicada pela linha tracejada), passando pelos servidores 1, 5, 4 e 3, nesta ordem. Ao receber a mensagem, o servidor procurador do respondedor (no caso o roteador de cebolas 3) entrega a mensagem ao destinatário.

Syverson, Reed e Goldschlag (1997) exemplificam esta utilização do roteamento de cebolas para navegação na Web e fazem uma comparação com outras RCAs que têm este propósito, apresentando como vantagem a utilização de diversos servidores, formando um serviço descentralizado. Reed, Syverson e Goldschlag (1996) exemplificam a utilização do roteamento de cebolas em outros serviços, tais como acesso remoto, correio eletrônico, e transferência de arquivos.

Uma vez que o primeiro servidor é quem define o restante da rota, este torna-se o mais sensível da rota em questão, pois o seu comprometimento faria com que o anonimato na comunicação deixasse de existir.

Quando se tem o estabelecimento de uma rota de forma segura (o servi-

dor inicial, que define a rota, não está comprometido), esta RCA tem a segurança do anonimato garantida de forma semelhante ao que propõe Chaum (1981) quanto aos servidores envolvidos: se ao menos um servidor componente da rota estiver seguro, a segurança do anonimato é garantida, mesmo que os demais estejam comprometidos (GOLDSCHLAG; REED; SYVERSON, 1996, p. 2).

3.3.1 Pacotes Cebola

Em uma rede de roteamento de cebolas os pacotes de dados trocados entre os servidores integrantes da RCA são denominados **pacotes cebola** (*onion packets*). A estrutura de dados de uma cebola é composta de diversas camadas de cifração sobre os dados que constituem a informação transmitida em uma cebola, denominada carga útil (*payload*).

As camadas de cifração que envolvem os dados da cebola dependem da rota estabelecida pelo servidor procurador do iniciador. De acordo com a rota estabelecida, o servidor procurador realiza uma primeira cifração dos dados, que será decifrada apenas pelo servidor mais próximo do respondedor. Em seguida o resultado desta cifração é também cifrado, desta vez para que apenas o penúltimo servidor na rota possa decifrar. Assim ocorre sucessivamente até que seja cifrado de forma que apenas o segundo servidor na rota possa decifrar. Este é o servidor que receberá a cebola recém criada. Syverson, Goldschlag e Reed (1997) descrevem este processo, que é feito com envelopes digitais, utilizando-se tanto de criptografia simétrica quanto assimétrica.

Quando uma cebola é recebida, cada servidor sabe de qual outro servidor a cebola se originou, e para onde ela deve ser encaminhada. Entretanto, não é possível saber algo sobre os demais componentes da rota, nem quantos servidores ainda existem na rota, e nem a sua posição na rota (a não ser que ele seja o último da rota).

Um pacote cebola pode ser representado pela seguinte expressão:

$$KU_{serv}(cabecalho, t\_exp, proximo\_nodo, F_{enc}, K_{enc}, F_{ret}, K_{ret}, carga\_util) \quad (3.1)$$

Detalhadamente, um pacote cebola contém os seguintes campos:

- **Cabeçalho**: O cabeçalho (representado por $cabecalho$ na expressão 3.1) contém duas informações: ele indica a qual circuito virtual a cebola pertence, e um comando que deve ser interpretado por cada servidor da rota. Este comando pode ser de **criação** (para estabelecer um circuito), de **destruição** (para encerrar um circuito), ou de **dados** (para transmitir informações entre os usuários). Este campo é cifrado simetricamente, de modo que apenas os servidores da rota tenham acesso ao seu conteúdo;

- **Tempo de expiração**: este campo (representado por t_exp na expressão 3.1) é utilizado para detectar reenvio de mensagens. Este reenvio pode ser utilizado por um atacante para obter informações de correlação entre as mensagens enviadas, de acordo com o ataque de repetição da mensagem (veja seção 2.5.7, na página 27). Cada nodo mantém uma cópia da cebola até ocorrer o tempo de expiração. Se o nodo recebe outra cópia daquela cebola durante este tempo, ela é ignorada. Também são ignoradas as cebolas recebidas com tempo de expiração ultrapassado;

- **Próximo nodo de roteamento**: este campo (representado por $proximo\_nodo$ na expressão 3.1) contém o endereço do próximo servidor componente da rota definida, para onde a cebola deve ser enviada. Caso o nodo que esteja recebendo seja o último da rota, este campo contém valor nulo;

- **Funções e chaves**: este campo contém dois pares de função/chave que indicam quais funções criptográficas e quais chaves devem ser utilizadas para a cifração e decifração dos dados ao longo do roteamento. Um é denominado **par de encaminhamento** (*forward pair*), e contém a função/chave (representado por F_{enc} e K_{enc} na expressão 3.1) que deve ser utilizada para cifrar as mensagens que vão no sentido do iniciador para o respondedor. O outro, denominado **par de retorno** (*backward pair*), contém a função/chave (representado por F_{ret} e K_{ret} na expressão 3.1) que deve ser utilizada para cifrar as mensagens que vão no sentido do respondedor para o iniciador;

- **Carga útil (*payload*)**: é neste campo (representado por $carga\_util$ na expressão 3.1) que se colocam os dados que representam a informação a ser enviada. Como as camadas da cebola constituem operações criptográficas, uma sobre a outra, para camadas mais externas da cebola, este campo conterá outra cebola. Apenas na cebola mais interna, a qual representa a última camada de cifração, é que este campo conterá apenas a informação útil, que se deseja transmitir.

Em cada nodo da rota pelo qual a cebola passa, é retirada a camada mais externa da cebola, o que faz com que seu tamanho diminua. Para que essa diminuição de tamanho não ocorra, o que permitiria o ataque de volume de mensagens (veja seção 2.5.3, na página 23), é feito um preenchimento (*padding*) com dados aleatórios antes que a cebola seja encaminhada. Este dados são colocados dentro da carga útil, e possuem o tamanho exato da camada que foi retirada.

Nenhum dos servidores da rota (exceto o último) sabe o que é informação útil e o que é preenchimento dentro da carga útil porque nenhum deles tem meios de saber sua posição na rota. Por não haver esta distinção, cada servidor trata o preenchimento como se fosse informação útil, realizando as operações criptográficas também sobre o preenchimento.

Com o uso do preenchimento, todas as cebolas enviadas têm sempre o mesmo tamanho. Para tanto, já na criação da cebola, feita pelo servidor procurador do iniciador, é colocado preenchimento para adequar a cebola ao tamanho padrão definido. O tamanho deste preenchimento inicial dependerá da quantidade de informação útil a ser transmitida.

3.3.2 Operação do circuito

Para a criação de um circuito virtual, uma cebola com o comando de criação é enviada através da rota definida pelo primeiro servidor procurador. Ao receber uma cebola de criação, o nodo escolhe um identificador para o circuito virtual e envia uma outra cebola de criação para o próximo servidor da rota, contendo o identificador de

circuito escolhido.

O par formado pelo identificador de circuito recebido, e o identificador de circuito escolhido por um nodo é armazenado. Até que o circuito virtual seja destruído, sempre que um servidor receber cebolas de dados com o identificador igual ao primeiro deste par, ele encaminhará a cebola utilizando o segundo identificador do par.

Mesmo com as operações criptográficas envolvidas no processo de estabelecimento do circuito virtual, e com as diversas camadas criadas em cada cebola, o processamento extra necessário para as comunicações realizadas através da rede não é muito maior do que o atraso que já existe naturalmente em comunicações pela Internet (GOLDSCHLAG; REED; SYVERSON, 1999, p. 3).

Desta forma, a troca de mensagens entre o iniciador e o respondedor através do circuito estabelecido ocorre da seguinte forma:

- **Sentido iniciador-respondedor**: Para o envio das cebolas de dados no sentido iniciador-respondedor, o servidor procurador do iniciador cifra a mensagem sucessivamente, incluindo os endereços intermediários da rota e as funções/chaves a serem utilizadas em cada camada de cifração. Durante o roteamento, cada servidor decifra a cebola recebida aplicando a função criptográfica de encaminhamento (e a correspondente chave) indicada na camada de cifração mais externa. Com esta decifração a camada mais externa é retirada, e a cebola resultante é então encaminhada para o próximo servidor da rota, de acordo com o endereço do próximo, presente na cebola obtida. O último servidor da rota então obterá a mensagem original, e a encaminhará ao respondedor;

- **Sentido respondedor-iniciador**: Ao receber dados de resposta, o servidor procurador que está se comunicando com o respondedor cifra estes dados. Esta cifração é feita aplicando-se a função criptográfica de retorno (e a correspondente chave) indicada na cebola por ele processada (que continha a mensagem original). Esta nova cebola de resposta é então enviada ao penúltimo servidor da rota, que realiza uma nova cifração sobre esta cebola. Esta nova cifração é feita utilizando-se a função criptográfica de retorno que o servidor havia obtido da camada por ele retirada da

cebola original. Este processo se repete em cada servidor da rota, agora na ordem inversa. O servidor procurador do iniciador, ao receber a cebola de resposta final, retira cada uma das camadas, utilizando a função criptográfica de retorno indicada em cada camada. Os dados obtidos da decifração da última camada são então entregues ao iniciador.

Como apenas a carga útil da cebola é cifrada, os outros campos da cebola poderiam ser visualizados por um atacante que estivesse observando a comunicação entre servidores adjacentes. Para se evitar esta visualização, as trocas de cebolas feitas entre os servidores adjacentes são feitas utilizando envelopes digitais.

Com as características obtidas através do estabelecimento de circuitos virtuais, a técnica do roteamento de cebolas pode ser utilizada inclusive na formação de **redes privadas virtuais** (*Virtual Private Network - VPN*), conforme propõem Reed, Syverson e Goldschlag (1998, p. 10).

3.3.3 Roteamento livre

Existe a possibilidade de fazer com que o roteamento dos pacotes cebola seja dinâmico, dando liberdade aos servidores para alterar a rota ao longo do encaminhamento dos pacotes. Para tanto o servidor procurador do iniciador pode instruir os integrantes da rota por ele definida para que estabeleçam rotas alternativas para a entrega da cebola ao próximo servidor definido.

Esta possibilidade de roteamento livre é útil para a garantia da segurança do anonimato, pois adiciona mais pontos à rota de entrega. O roteamento livre também é útil quando o servidor procurador do iniciador não tem conhecimento de uma rota completa até o respondedor. Ou ainda, quando existe uma falha na conexão em algum ponto da rede, da qual o servidor ainda não tem conhecimento. Nestes casos os servidores componentes da rota também são instruídos a estabelecerem rotas alternativas para se alcançar o próximo servidor.

Outra vantagem da utilização do roteamento livre é que a rota pode ter um tamanho maior mesmo mantendo-se o tamanho fixo das cebolas, necessário para a segurança do anonimato. Com o tamanho fixo, o número total de servidores em uma rota fica limitado. Ao se utilizar roteamento livre, a cebola pode ter camadas de cifração acrescentadas ao longo do caminho, e não apenas camadas retiradas. Esta possibilidade impede que o tamanho máximo da rota esteja associado ao tamanho fixo definido para as cebolas.

O roteamento livre também faz uso de um mecanismo para se prevenir o crescimento indefinido do tamanho da rota, o que é indesejável pois pode causar atrasos na entrega das cebolas. Isto é feito acrescentando-se um campo às camadas das cebolas que participam de uma rede onde o roteamento livre é utilizado. Este campo indica ao servidor que fará um roteamento livre o número máximo de camadas que ele poderá acrescentar.

Desta forma, ao decidir por realizar um roteamento livre, o servidor poderá acrescentar tantas camadas quanto for indicado no campo. Caso ele inclua o campo de limite de camada às camadas por ele acrescentadas (dando possibilidade aos outros servidores para também realizar o roteamento livre), os valores máximos de camada por ele acrescentados deverão ser levados em consideração, de forma a não ultrapassar o limite estabelecido na camada por ele retirada da cebola recebida.

Esta condição pode ser representada pela seguinte expressão:

$$|C| + \sum maxCamadas_c \text{ para todo } c \in C \leq maxCamadas_a \tag{3.2}$$

Sendo $maxCamadas_a$ o valor do campo de limite de camadas presente na camada retirada por um servidor a que decidiu realizar um roteamento livre; $maxCamadas_c$ o valor do campo de limite de camadas presente em uma camada c adicionada; e C o conjunto das camadas que este servidor decidiu acrescentar neste roteamento.

Ao se manter esta relação, garante-se que o tamanho total da rota não cresça indefinidamente, o que atrasaria consideravelmente a entrega das cebolas.

O tamanho da rota utilizada em uma rede de comunicação anônima tem influência no grau de anonimato proporcionado pela rede. Em geral, um aumento no tamanho da rota ocasiona um aumento no grau de anonimato do usuário. Entretanto, conforme um estudo feito por Guan et al. (2002), este comportamento pode ter um resultado inverso se o tamanho da rota tornar-se muito grande, diminuindo o grau de anonimato ao invés de aumentar (GUAN et al., 2002, p. 7). Esta forma como uma RCA define a rota a ser percorrida pelas mensagens é denominada **estratégia de seleção de rota**. Estratégias que utilizam tamanho variável de rota apresentam melhores resultados quanto ao grau de anonimato do que estratégias que utilizam um tamanho de rota fixo (GUAN et al., 2002, p. 2).

3.3.4 Cebolas de resposta

O roteamento de cebolas permite que o respondedor envie respostas ao iniciador mesmo após o circuito original ter sido desfeito. Isto é útil quando informações solicitadas pelo iniciador não estão prontamente disponíveis, fazendo com que o respondedor tenha que enviar estas informações num momento futuro, após a comunicação inicial.

Para utilização desta funcionalidade é preciso fazer uso das **cebolas de resposta**. O uso das cebolas de resposta faz com que o iniciador mantenha-se anônimo, o fato de o respondedor desejar enviar uma resposta tardia ao iniciador não implica em que o respondedor deva tomar conhecimento de quem é o iniciador.

Para diferenciação das cebolas de resposta, podemos denominar as cebolas vistas até o momento, usadas para o envio e recebimento de informações enquanto o circuito original permanece ativo, de **cebolas de transferência**.

De mesma forma que ocorre com a cebola de transferência, uma cebola de resposta revela a cada servidor da rede apenas o próximo nodo para onde a cebola deve ser enviada. E a estrutura de uma cebola de resposta também é a mesma de uma

cebola de transferência, o que faz com que ela seja processada da mesma forma pelos nodos da rede. Os nodos intermediários da rede não têm meios de distinguir uma cebola de resposta de uma cebola de transferência. Para os servidores procuradores do iniciador e do respondedor, o processamento também passa a ser o mesmo, uma vez que o circuito tenha sido restabelecido.

Da mesma forma que uma cebola de transferência, uma cebola de resposta também pode ser utilizada apenas uma vez. As cebolas são guardadas pelo nodo até que expirem, e todas as cebolas processadas são comparadas para evitar repetição. Como as cebolas de resposta são utilizadas apenas uma vez, caso se deseje múltiplas respostas, então deve-se enviar múltiplas cebolas de resposta diferentes.

3.3.5 Comandos de controle

A operação dos circuitos virtuais que se formam através dos nodos da rede é feita utilizando-se três possíveis comandos de controle.

O primeiro comando que um nodo processa é o de **criação** de um circuito virtual. Em cada nodo, um circuito virtual tem duas conexões, os dados que chegam por uma conexão são encaminhados para a outra conexão. O circuito é definido pelos rótulos dados às suas duas conexões.

A criação de um circuito virtual consiste no processo de definir os rótulos das conexões em cada nodo ao longo da rota. Para o primeiro nodo, uma conexão é a ligação com o iniciador, e a outra conexão é uma ligação com o próximo nodo da rota definida. O servidor procurador do iniciador cria uma cebola definindo a seqüência dos nodos intermediários de roteamento até se alcançar o servidor procurador do respondedor. A cebola é então dividida em pedaços, de acordo com o tamanho de carga útil estipulado, e enviada ao próximo nodo. Esta cebola contém uma identificação da conexão utilizada, e o comando de criação.

Cada nodo subseqüente remonta a cebola contendo o comando de criação de circuito. Antes de executar o comando de criação é feita a verificação de que não se trata de uma cebola com tempo de expiração vencido, nem de uma repetição.

Com uma cebola autêntica, ele retira uma camada da cebola, revelando o próximo nodo da rota, e o par criptográfico de função/chave que deve ser utilizado. O nodo então cria um novo rótulo da conexão com o próximo nodo, e repassa a cebola para este, de forma similar. Esta cebola enviada conta com preenchimento para compensar a camada retirada. Também é armazenado o par criptográfico de função/chave associado ao circuito estabelecido.

O servidor procurador do respondedor, por ser o último da rota, irá realizar apenas parte deste processamento. Ele precisa confirmar que se trata de uma cebola autêntica, e assim retirar uma camada da cebola. Ao identificar que não há mais nodos, ele apenas armazena o par criptográfico de função/chave que deve ser utilizado para os dados de resposta. É passado então a carga útil para o respondedor, em seu próprio formato, e não como uma cebola.

O segundo comando que pode ser processado por um nodo é o comando de **dados**, usado para transmitir as informações do iniciador para o respondedor. Estas informações são divididas em pedaços do tamanho da carga útil de uma cebola, e cada pedaço torna-se uma nova cebola contendo o comando de dados. O processamento destas cebolas ocorre conforme a operação do circuito, considerando-se os identificadores de conexão entre nodos, e os pares de função/chave para cada nodo.

O terceiro comando é o de **destruição**, e tem a função de desconectar um circuito virtual quando o mesmo não é mais necessário, ou sob alguma condição de erro. As mensagens com o comando de destruição podem ser iniciadas por qualquer nodo integrante da rota, e todos os nodos subseqüentes devem encaminhar a mensagem na direção adequada, para que todo o circuito seja desfeito. A carga útil de uma cebola que contenha um comando de destruição é constituída apenas por preenchimentos, e ainda assim é cifrada com o par função/chave definido. Ao receber uma mensagem de destruição cada nodo da rota descarta o identificador do circuito virtual em questão, desfazendo a conexão.

3.3.6 Análise da segurança

O roteamento de cebolas apresenta medidas que dificultam ou, em muitos casos, impedem a realização de diversos ataques ao anonimato. Com base nas características encontradas no roteamento de cebolas, propõe-se a seguinte análise da segurança desta RCA, considerando os tipos de ataque ao anonimato, e as possibilidades de defesa encontradas:

- **Ataque à codificação da mensagem:** Como a comunicação do iniciador com o seu servidor procurador é feita de forma aberta, sem o uso de criptografia, a realização deste ataque no momento inicial da comunicação resulta na revelação de todas as informações enviadas, desfazendo o anonimato. Se o ataque for realizado durante o tráfego das cebolas entre cada nodo, o atacante não terá sucesso, pois nestas trocas de mensagens é utilizada a técnica do envelope digital entre cada nodo, garantindo uma nova codificação a cada novo ponto da rota;

- **Ataque de temporização:** Conforme os próprios autores do roteamento de cebolas reconhecem, esta RCA é vulnerável ao ataque de temporização (GOLDSCHLAG; REED; SYVERSON, 1996, p. 12). Isto ocorre porque o roteamento de cebolas destina-se a aplicações que exigem respostas rápidas, impedindo a RCA de inserir atrasos na comunicação, que dificultam o ataque de temporização. Desta forma, o atacante pode observar a abertura simultânea de conexões no primeiro e no último servidor e estabelecer relações entre elas, revelando quem está solicitando determinada informação, e para quem a solicitação se encaminha. Como a solução para este problema (inserção de atrasos) é indesejável nesta rede, uma medida a ser tomada para se evitar o ataque de temporização seria encontrar um ponto de equilíbrio entre a rapidez desejada para troca de mensagens, e o nível de anonimato obtido com a inserção de pequenos atrasos;

- **Ataque de volume de mensagens:** Como o Roteamento de Cebolas faz uso de preenchimento de mensagem para manter as cebolas com tamanho constante, a

realização do ataque de volume de mensagens não revela informações sobre as comunicações realizadas;

- **Ataque de inundação:** Para dificultar a realização deste ataque, o roteamento de cebolas prevê a utilização de mensagens de disfarce para a manutenção do tráfego com um volume constante;

- **Ataque de cruzamento:** Conforme visto anteriormente (seção 2.5.5, na página 25), o uso de criptografia e de mensagens de disfarce torna o ataque de cruzamento ineficaz. Apesar de o roteamento de cebolas fazer uso de ambas as técnicas, esta RCA ainda é vulnerável a este ataque. Isto ocorre porque tanto a comunicação do iniciador com seu servidor procurador, quanto a do respondedor com seu servidor procurador, são feitas sem o uso de criptografia. Para evitar este ataque, os autores propõem que a comunicação entre o iniciador e seu servidor procurador seja feita em um ambiente garantidamente seguro (GOLDSCHLAG; REED; SYVERSON, 1996, p. 2);

- **Ataque de marcação da mensagem:** O uso de criptografia no envio das cebolas faz com que qualquer alteração em uma cebola seja detectada pelos servidores da rede, impedindo o ataque de marcação da mensagem;

- **Ataque de repetição da mensagem:** A medida tomada pelo roteamento de cebolas para impedir a realização deste ataque é a utilização do mecanismo de carimbo de tempo, conforme a informação presente no campo **Tempo de expiração** de uma cebola. Entretanto, como os próprios autores reconhecem, a falta de sincronia entre os relógios dos servidores pode ocasionar a perda de mensagens verdadeiras (GOLDSCHLAG; REED; SYVERSON, 1996, p. 13), sendo tratadas como falsas devido ao tempo de expiração presente não ser adequado ao tempo marcado no relógio do servidor;

- **Ataque de negação de serviço:** O roteamento de cebolas não possui medidas preventivas contra o ataque de negação de serviço;

- **Ataque do predecessor:** Conforme descrevem Wright et al. (2002, p. 8), se o ataque do predecessor for realizado em conjunto com o ataque de temporização, o roteamento de cebolas torna-se vulnerável, revelando informações a respeito do emissor das mensagens;

- **Ataque da descoberta:** Como se trata de um ataque passivo, baseado apenas em observações dos remetentes e destinatários, torna-se difícil a detecção deste ataque por parte de uma RCA. Desta forma, o roteamento de cebolas não possui medidas preventivas para este ataque.

Pelas medidas tomadas no Roteamento de Cebolas, esta RCA apresenta-se como uma opção para utilização em sistemas que necessitam de baixa latência na comunicação. Um problema para este requisito é a maior vulnerabilidade a um ataque de temporização. Em outra análise da segurança do Roteamento de Cebolas, feita por Syverson et al. (2000, p. 12), são feitas propostas para diminuir esta vulnerabilidade ao ataque de temporização. Syverson et al. propõem a adição de atrasos no tráfego a partir do servidor procurador do iniciador, e a utilização de preenchimentos de mensagem em partes da rota, além daquele que já existe entre cada servidor.

Em um trabalho recente de Dingledine, Mathewson e Syverson (2004) é apresentada a rede **Tor**, que consiste em uma segunda geração da RCA Roteamento de Cebolas. O objetivo desta rede de segunda geração é superar algumas limitações presentes no projeto original do Roteamento de Cebolas. As principais alterações realizadas consistem em: permitir a utilização de um único circuito virtual por várias conexões do usuário; arquitetura alternativa que permite a entrega de mensagens também por servidores intermediários; e controle de congestionamento. Com as melhorias implementadas nesta segunda geração é possível estabelecer relações de custo/benefício entre três características: anonimato, usabilidade, e eficiência (DINGLEDINE; MATHEWSON; SYVERSON, 2004, p. 3). Com ajustes na configuração da rede pode-se fazer um balanceamento entre as três características, definindo qual a mais interessante para os requisitos existentes.

3.4 Rede de Mistura

Uma das primeiras propostas de rede de comunicação anônima foi feita por Chaum (1981), e é conhecida como **Rede de Mistura** (*Mix Net*).

A rede de mistura é um mecanismo que visa resolver os problemas da análise de tráfego (que resultam nos ataques ao anonimato) sem que seja necessária a confiança em uma única autoridade central. Para tanto a rede de mistura faz uso da criptografia assimétrica, e distribui a segurança da rede ao longo de ***n*** servidores (denominados **misturadores**) que compõem os pontos da RCA, de forma que, mesmo tendo ***n-1*** servidores comprometidos por um atacante, o anonimato da comunicação ainda é mantido (CHAUM, 1981, p. 1).

A rede de mistura faz duas considerações para garantir a segurança do anonimato:

- (1) Nenhum participante pode determinar qualquer coisa sobre a relação entre o conjunto de mensagens cifradas e o conjunto de mensagens decifradas, nem pode criar falsificações sem o conhecimento da chave envolvida na cifração;
- (2) Qualquer participante pode saber a origem, destino, e representação de todas as mensagens no sistema de comunicação utilizado, e qualquer participante pode inserir, remover ou modificar mensagens.

A primeira consideração diz respeito à segurança das técnicas de criptografia utilizadas, como confia-se que as mesmas são invioláveis, é preciso fazer a mesma consideração para a rede de mistura, pois a mesma faz uso destas técnicas.

A segunda consideração diz respeito à forma como os dados trafegam em uma rede de computadores, em especial a Internet, a qual permite a realização das operações citadas. Um exemplo são os softwares conhecidos como *sniffers* que permitem a leitura de qualquer informação que esteja trafegando na rede em que um computador esteja conectado, mesmo que as informações não sejam destinadas a ele. Com o uso desta ferramenta um atacante também pode forjar mensagens, de forma que o destinatário pense que a mesma veio de determinado local.

3.4.1 Comunicação anônima na rede

O envio de mensagens de forma anônima através de um misturador é feito da seguinte forma: o emissor deve cifrar cada mensagem com a chave pública do receptor. Junto a esta mensagem cifrada o emissor coloca o endereço do receptor e cifra estes dados com a chave pública do misturador.

Ao receber uma mensagem, o misturador utiliza sua chave privada para decifrá-la, obtendo assim a mensagem a ser encaminhada e o endereço para onde deve ser encaminhada. Ao receber a mensagem do misturador, o receptor utiliza sua chave privada para decifrar a mensagem, obtendo assim a mensagem original, enviada de forma anônima pelo emissor.

Para evitar o ataque de volume de mensagens (veja seção 2.5.3, na página 23) a rede de mistura utiliza o preenchimento de mensagem, através da adição de dados aleatórios à mensagem que será cifrada (CHAUM, 1981, p. 1). Considerando a notação e operações presentes na Lista de Símbolos (veja página xiv), sendo KU_1 a chave pública do misturador, KU_a a chave pública do receptor, M a mensagem a ser enviada, A o endereço do receptor, R_x dados aleatórios de preenchimento em cada camada da mensagem (R_0 na camada interna, e R_1 na externa), o tratamento de mensagens cifradas feito pelo misturador pode ser representado pela seguinte expressão:

$$KU_1(R_1, KU_a(R_0, M), A) \rightarrow KU_a(R_0, M), A \tag{3.3}$$

Nesta expressão, o símbolo "$\rightarrow$" representa a decifração feita pelo misturador, utilizando sua chave privada. O lado esquerdo da expressão constitui a mensagem criptográfica gerada pelo emissor, e o lado direito da expressão representa o conteúdo obtido, que consiste na mensagem cifrada a ser enviada para o receptor, seguida do endereço A do receptor.

O propósito de um misturador é o de esconder a relação entre as mensagens que entram no servidor e as que saem (CHAUM, 1981, p. 2). A ordem de chegada das mensagens é escondida através do encaminhamento das mesmas em pacotes de dados do

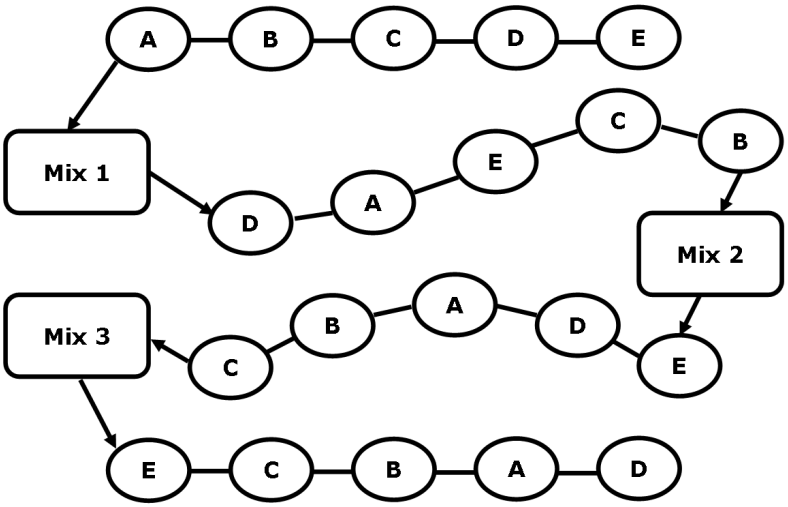

Figura 3.2: Funcionamento da Rede de Mistura

mesmo tamanho e em ordem diferente da recebida. A figura 3.2 ilustra o funcionamento de uma rede de mistura, em que cada misturador altera a ordem de envio das mensagens. Após serem processadas pelo último misturador, as mensagens são encaminhadas aos respectivos destinatários.

Esta reordenação das mensagens para envio é que aumenta o tempo necessário para que uma mensagem saia do emissor e chegue até o receptor. Este atraso na entrega resultante deste procedimento realizado é que faz com que a rede de mistura seja classificada como uma **rede de comunicação anônima de alta latência** (WRIGHT et al., 2002, p. 11).

Outra função importante do misturador é garantir que nenhuma mensagem seja processada mais de uma vez. Para tanto o misturador mantém um registro das mensagens processadas durante sua operação, descartando qualquer mensagem que tenha seu registro já presente, o que caracteriza uma repetição da mesma (CHAUM, 1981, p. 2). Outra solução proposta na rede de mistura para esta questão é o uso de **carimbo de tempo** (*time-stamp*) em cada mensagem, fazendo com que cada mensagem seja processada apenas se estiver dentro do período de tempo indicado na marcação feita ao ser enviada.

O uso de diversos misturadores na rede aumenta a garantia da obtenção do anonimato, pois, conforme discutido anteriormente, esta medida faz com que a segurança da rede esteja distribuída ao longo dos diversos servidores. Entretanto, o uso de diversos servidores faz com que o envio de mensagens necessite de alguns passos a mais: é preciso que a mensagem, juntamente com o endereço do receptor, seja cifrada com a chave pública de cada um dos misturadores da rede, na ordem inversa a que eles serão utilizados. Este procedimento forma uma estrutura similar à colocação de vários envelopes um dentro do outro.

Com o uso de diversos misturadores, e sendo n o primeiro misturador da rede, a mensagem criptográfica criada pelo emissor pode ser representada pela seguinte expressão:

$$KU_n(R_n, KU_{n-1}(R_{n-1}, \ldots, KU_3(R_3, KU_2(R_2, KU_1(R_1, KU_a(R_0, M), A))) \ldots)) \tag{3.4}$$

A decifração realizada pelo último misturador obtém conteúdo semelhante ao que ocorre na utilização de apenas um misturador, ou seja, a mensagem cifrada a ser enviada para o receptor, e o endereço do receptor: $KU_a(R_0, M), A$.

Com relação às propriedades da comunicação anônima (veja seção 2.2, na página 12), o procedimento de reordenação das mensagens utilizado na Rede de Mistura busca garantir a impossibilidade de associação do remetente ao destinatário. Caso o remetente deseje receber respostas do destinatário, e também manter-se anônimo perante este, é preciso utilizar o procedimento descrito na próxima seção.

3.4.2 Endereço de retorno não-rastreável

A rede de mistura também permite comunicação bi-direcional sem a revelação do endereço do emissor para o receptor. Para tanto o emissor (usuário A) deve enviar com suas mensagens um **endereço de retorno não-rastreável**, que consiste no seu endereço real (A_a), juntamente com uma chave simétrica por ele escolhida (K_a), ambos cifrados com a chave pública do misturador (KU_1), e também uma chave pública

escolhida por ele (KU_x), única para aquele envio. Assim, o endereço de retorno pode ser representado por $KU_1(K_a, A_a), KU_x$.

Para enviar uma resposta ao usuário A, o receptor (usuário B) deve proceder da seguinte forma: de posse do endereço de retorno enviado pelo usuário A, ele deixa intacta a primeira parte (que contém uma chave simétrica e o endereço real do usuário A cifrados com a chave pública do misturador - $KU_1(K_a, A_a)$) e utiliza a segunda parte (a chave pública escolhida pelo usuário A - KU_x) para cifrar o seguinte par: uma chave simétrica K_b escolhida pelo usuário B, e sua mensagem M de resposta cifrada com esta chave. O usuário B então envia para o misturador a primeira parte do endereço de retorno recebido do usuário A juntamente com a cifração da sua mensagem de resposta.

A operação para envio de uma mensagem de resposta para o usuário A pode ser expressa da seguinte maneira:

$$KU_1(K_a, A_a), KU_x(R, M) \rightarrow A_a, K_a(KU_x(K_b, M)) \tag{3.5}$$

O símbolo "$\rightarrow$" representa as operações feitas pelo misturador ao receber uma mensagem de resposta: o misturador utiliza sua chave privada para decifrar a primeira parte e obter o endereço real do usuário A (A_a) para onde se dirige a resposta, e a chave simétrica K_a que deve ser utilizada para cifrar a mensagem de resposta, de forma a alterar sua codificação ao passar pelo misturador. Assim o misturador envia para o usuário A, a mensagem de resposta $KU_x(R, M)$ recebida, cifrada com a chave simétrica obtida no passo anterior.

Apenas o usuário A pode obter o conteúdo da mensagem de resposta, pois foi ele quem definiu tanto a chave simétrica K_a utilizada pelo misturador para lhe enviar a resposta cifrada, quanto a chave assimétrica KU_x utilizada na cifração inicial da mensagem de resposta, realizada pelo usuário B.

Da mesma forma que as mensagens enviadas não podem ser repetidas para se evitar ataques, os endereços de retorno enviados por um emissor de mensagens também não podem se repetir. Desta forma, para cada mensagem enviada da qual o

emissor deseje receber uma resposta, ele deve gerar chaves diferentes para acompanhar cada endereço de retorno.

Para a utilização do endereço de retorno não-rastreável em conjunto com diversos misturadores, o emissor deve escolher, além da chave assimétrica KU_x, uma chave simétrica para cada misturador da rede (K_1 a K_n), as quais farão parte do endereço de retorno não-rastreável:

$$KU_1(K_1, KU_2(K_2, \ldots, KU_{n-1}(K_{n-1}, KU_n(K_n, A_a)) \ldots)), KU_x \tag{3.6}$$

Para enviar sua resposta, o usuário B procede da mesma forma como descrito anteriormente, obtendo $KU_x(K_b, M)$. Ele envia este conteúdo e a primeira parte do endereço de retorno não-rastreável para o primeiro misturador. Cada misturador, ao processar a mensagem de resposta, decifra a mensagem recebida, e utiliza a chave simétrica obtida para cifrar a mensagem de resposta, com o objetivo de alterar a codificação da mesma. O resultado desta operação feita pelo primeiro misturador pode ser expresso da seguinte maneira:

$$KU_2(K_2, \ldots, KU_{n-1}(K_{n-1}, KU_n(K_n, A_a)) \ldots), K_1(KU_x(K_b, M)) \tag{3.7}$$

Após o procedimento feito por cada misturador, o último misturador da rede terá o endereço real do usuário A, e a mensagem de resposta do usuário B cifrada com todas as chaves simétricas inicialmente definidas pelo usuário A:

$$A_a, K_n(K_{n-1} \ldots K_2(K_1(KU_x(K_b, M))) \ldots) \tag{3.8}$$

Semelhante ao que ocorre na utilização de apenas um misturador, apenas o usuário A pode obter o conteúdo da mensagem de resposta, pois foi ele quem definiu tanto as chaves simétricas K_1 a K_n utilizadas pelos misturadores para lhe enviar a resposta cifrada, quanto a chave assimétrica KU_x utilizada na cifração inicial da mensagem de resposta, realizada pelo usuário B.

3.4.3 Estratégia de agrupamento de mensagens

Na proposta original da rede de mistura, um misturador recebe as mensagens continuamente, e quando seu repositório de mensagens enche, as mesmas são encaminhadas em uma ordem diferente daquela em que foram recebidas. A forma como são tratados o recebimento e o encaminhamento (*flush*) de mensagens é denominada **estratégia de agrupamento de mensagens** (*batching strategy*). Em um misturador, a estratégia utilizada é um dos parâmetros mais importantes (DÍAZ; SERJANTOV, 2003, p. 1).

As estratégias de agrupamento encontradas na literatura fazem uso de duas principais técnicas. A primeira consiste no uso de um **limiar** (*threshold*), e a segunda consiste no uso de **temporização** (*timing*). A técnica do limiar é aquela utilizada na proposta inicial da rede de mistura, ou seja, quando o número de mensagens recebidas atinge um limiar (a capacidade do repositório), ocorre o encaminhamento. Na técnica da temporização, a cada período de tempo definido o encaminhamento é realizado.

Algumas estratégias utilizam as duas técnicas, e outras ainda introduzem um mecanismo denominado **poça** (*pool*). Este mecanismo faz com que um número determinado de mensagens recebidas sejam mantidas na poça, enquanto que as outras sejam encaminhadas. Caso o número de mensagens mantidas na poça seja diferente a cada rodada, este mecanismo recebe o nome de **poça dinâmica** (*dynamic pool*). A adição de uma poça ao misturador melhora significativamente o anonimato, e a utilização de uma poça dinâmica melhora a resistência a ataques de inundação (SERJANTOV; DINGLEDINE; SYVERSON, 2002, p. 15).

Conforme Serjantov, Dingledine e Syverson (2002, p. 9), um misturador que tenha esta estratégia de agrupamento é denominado **misturador temporizado de poça dinâmica** (*timed dynamic-pool mix*). Para esta estratégia de agrupamento é preciso a definição de três parâmetros: o período com o qual os encaminhamentos serão feitos, o tamanho mínimo da poça (indicando a quantidade mínima de mensagens que serão retidas em um encaminhamento), e a porcentagem de mensagens que devem ser enviadas (este parâmetro é que torna dinâmica a poça). A correta utilização desta estratégia melhora o anonimato da rede por fazer aumentar o tamanho do grupo de anonimato, dentre os

usuários da rede (DÍAZ; PRENEEL, 2004, p. 6).

A melhoria no anonimato proporcionada pela utilização de uma correta estratégia de agrupamento de mensagens é obtida principalmente em redes de alta latência. Entretanto, conforme demonstraram Zhu et al. (2004, p. 11), esta melhoria não ocorre de forma tão significativa em redes de baixa latência, principalmente em relação ao ataque de volume de mensagens.

Com relação à arquitetura da rede de mistura, Dingledine, Shmatikov e Syverson (2004) mostram que a utilização correta da estratégia de agrupamento de mensagens resulta na melhoria do anonimato proporcionado pela rede, tanto na arquitetura em cadeia, quanto na arquitetura em clique. Dingledine, Shmatikov e Syverson também propuseram outras métricas para comparações entre as arquiteturas, tais como vazão, capacidade, e utilização da largura de banda disponível (DINGLEDINE; SHMATIKOV; SYVERSON, 2004, p. 12).

3.4.4 Uso genérico da rede

Originalmente (CHAUM, 1981) a proposta da rede de mistura foi feita tendo o envio de correio eletrônico como a principal aplicação para se utilizar esta RCA. **Babel** é um sistema proposto por Gülcü e Tsudik (1996) que implementa a rede de mistura para utilização em correio eletrônico. Este sistema pode ser classificado como um sistema reenviador do tipo II (veja seção 2.3, na página 14). Esta implementação possui medidas para combate aos ataques de análise de tráfego, tais como utilização de mensagens com tamanho constante, e detecção de repetição de mensagens.

Entretanto, apesar da proposta inicial da rede de mistura ter o correio eletrônico como principal aplicação, a sua estrutura e procedimentos de manipulação dos dados permitem um uso mais genérico da rede, a qual pode ser aplicada a diversos sistemas que necessitem do anonimato na comunicação.

Como as mensagens enviadas podem ter tamanhos variados, para o envio de mensagens grandes a proposta da rede de mistura prevê que estas mensagens primeiro sejam cifradas para depois serem divididas em diversas partes. Para que o número

de mensagens enviadas não seja revelado, Chaum propôs o envio de mensagens de disfarce por parte do emissor. Para que também se mantenha em segredo o número de mensagens recebidas, Chaum propôs que o receptor tenha acesso a uma grande quantidade de mensagens, selecionando apenas as que são a ele destinadas, e descartando as demais, bem como as mensagens de disfarce.

A rede de mistura pode ser utilizada tanto com arquitetura em cadeia quanto com arquitetura em clique (veja seção 3.2.4, na página 36). Esta escolha pode ser baseada na infra-estrutura de comunicação disponível, ou no nível de confiança de cada ponto.

Jerichow et al. (1998) propuseram a utilização de misturadores em redes ISDN (Integrated Services Digital Network), que são utilizadas em telefonia, o que demanda baixa latência na comunicação. A proposta de Jerichow et al. é baseada em uma alteração realizada no procedimento original da rede de mistura, que consiste em permitir a mistura de um fluxo contínuo de dados, ao invés de misturar apenas mensagens em pacotes de dados. Este conceito foi denominado **canal anônimo**, e segundo os autores, pode ser utilizado em outras redes de comunicação. A implementação realizada por Jerichow et al. exigiu alterações nos protocolos existentes para redes ISDN, e obteve taxas de transmissão de dados semelhantes às que se obtém sem a utilização de misturadores.

Na trabalho feito por Berthold, Federrath e Köpsell (2001) é proposta a utilização da rede de mistura para navegação anônima na Web. Para o combate aos ataques de análise de tráfego Berthold, Federrath e Köpsell utilizam mensagens de disfarce, tamanho constante de mensagens, e um mecanismo de autenticação com bilhetes (veja seção 2.5.4, na página 24). Para fornecer aos usuários da rede uma medida aproximada do nível de anonimato que o usuário está obtendo, a arquitetura proposta mantém estatísticas a respeito do número de usuários presentes na RCA, e a duração de cada sessão estabelecida pelos usuários. Conforme aumenta o número de usuários e o tempo de duração das sessões, cada usuário é informado a respeito da variação no nível de anonimato. Como nesta proposta de navegação anônima é utilizada a rede de mistura, a propriedade da comunicação anônima de impossibilidade de associação do remetente ao destinatário também é obtida, diferente do que ocorre com o serviço Anonymizer, que obtém apenas

o anonimato do remetente em navegação na Web (veja seção 2.4.1, na página 18).

A rede de mistura também pode utilizar servidores dinâmicos, resultando em uma configuração P2P (veja seção 3.2.5, na página 39). **MorphMix** é uma rede proposta por Rennhard e Plattner (2002) que utiliza esta arquitetura, estabelecendo túneis anônimos para comunicação através dos misturadores/usuários. Um característica interessante desta rede é que ela não faz uso de mensagens de disfarce para o combate a ataques ao anonimato, uma técnica comumente encontrada em redes de comunicação anônima. Os autores justificam a não utilização de mensagens de disfarce com o problema do processamento extra desnecessário, existente nesta técnica, e buscam a garantia do anonimato através da utilização de uma grande quantidade de usuários, o que é comum em redes P2P (RENNHARD; PLATTNER, 2002, p. 5). Entretanto, com esta grande quantidade de usuários, surge o problema da confiança em cada usuário, para se ter a garantia de que não se trata de um atacante. Para tanto Rennhard e Plattner utilizaram um mecanismo de detecção de conspiração, que busca encontrar atacantes na rede. Este mecanismo toma como medida para determinar se um misturador da rede é um atacante, a quantidade de vezes que ele aparece nos túneis estabelecidos. Segundo os autores, os misturadores operados por um atacante irão buscar formar mais túneis nos quais eles são participantes para obter sucesso no ataque (RENNHARD; PLATTNER, 2002, p. 6).

Em outro trabalho de Rennhard e Plattner (2004), são apresentados novos resultados de pesquisas realizadas com a rede MorphMix. É feita uma análise do mecanismo de detecção de conspiração, e da escalabilidade proporcionada pela rede. Também são apresentados dados de simulações feitas, e comparações com outros sistemas. Ainda existem questões a serem resolvidas na proposta da rede MorphMix, uma delas é o problema da volatilidade dos servidores, que entram e saem da rede constantemente, o que impede o estabelecimento de conexões por longos períodos (RENNHARD; PLATTNER, 2004, p. 16).

Para a utilização genérica da rede de mistura, o atraso introduzido pela rede no envio das mensagens é uma questão a ser considerada na utilização desta RCA em determinada aplicação. Como cada misturador entrega as mensagens em ordem diferente da que as recebeu, as sucessivas trocas na ordem de entrega ao longo dos servidores

da rede introduz atrasos na entrega das mensagens. Conforme demonstrou o trabalho de Chaum (1981), realizando-se as adaptações necessárias, a rede de mistura pode ser utilizada também em aplicações com requisito de baixa latência na comunicação. Todavia, dependendo da configuração da rede, o atraso inserido pode não ser adequado, podendo ser mais interessante optar por outro tipo de rede de comunicação anônima.

3.4.5 Aplicação em votação digital

Uma aplicação em que pode ser utilizada a rede de mistura, mesmo com alta latência, é em sistemas de votação digital, onde os atrasos no envio das mensagens são aceitáveis, O envio do voto para o sistema de votação tem características semelhantes ao envio de correio eletrônico, ou seja, o votante não espera uma resposta imediata do sistema (o resultado da votação). Uma vez entregue o voto, o votante encerra sua participação no processo, voltando a ter acesso ao sistema apenas ao final da votação, para visualização dos resultados.

No trabalho original da rede de mistura, Chaum (1981) já propunha sua utilização em conjunto com pseudônimos (veja seção 2.3.1, na página 17) para possibilitar a realização de uma votação de forma anônima, sem que ao votante fosse associado o voto realizado.

A principal vantagem da utilização da rede de mistura em uma votação digital é que o seu funcionamento é semelhante ao que ocorre com uma urna real, onde os votos são colocados em determinada seqüência, mas ao se abrir a urna, não há garantia de que a ordem de retirada dos votos será igual à que ocorreu na colocação dos votos na urna. Desta forma, mesmo que se saiba a ordem em que os votantes realizaram seu voto, não há como relacionar o voto ao votante correspondente, pois a urna realiza naturalmente uma mistura das cédulas em papel. Este processo é o mesmo que ocorre na rede de mistura, pois as mensagens (tal como as cédulas em papel) que entram em determinada ordem, são misturadas, e ao saírem estão em outra ordem (tal como a retirada das cédulas de papel de uma urna). Isto caracteriza uma das propriedades da comunicação anônima garantidas pelo uso da rede de mistura, a propriedade de impossibilidade de associação do remetente

ao destinatário (veja seção 2.2, na página 12).

A principal questão a ser resolvida na utilização de redes de mistura em votação digital é poder verificar que a rede de mistura operou corretamente, sem ter ocorrido alterações nas mensagens processadas. Isto precisa ser feito sem se revelar o conteúdo das mensagens processadas.

Neff (2001) propôs um método de mistura que, ao utilizar criptografia assimétrica, possui a propriedade da verificabilidade universal (veja seção 5.3, na página 109), o que em uma votação permite garantir que todos os votos sejam contados corretamente, porém sem revelar os respectivos votantes. Para a garantia de operação correta dos misturadores, Neff faz uso de operações com números primos, e obtém um método computacionalmente mais eficiente que outras propostas (NEFF, 2001, p. 3).

Jakobsson, Juels e Rivest (2002) também propuseram um método para se obter a garantia de que a rede de mistura tenha operado corretamente. Para tanto, realiza-se uma verificação aleatória parcial da mistura realizada por cada servidor da rede. A idéia básica deste procedimento consiste em que cada servidor, ao invés de fornecer uma prova de operação completa e correta, forneça uma forte evidência de sua correta operação. Para a obtenção desta forte evidência, o servidor revela um subconjunto de suas relações de entrada e saída, selecionado aleatoriamente. Esta revelação parcial permite uma verificação probabilística da operação correta de cada servidor.

Com a revelação parcial, o anonimato das mensagens é garantido com a operação global da rede, e não apenas em cada servidor pois, como cada servidor revela uma parte da mistura realizada, algumas relações de mistura de mensagem estarão disponíveis. Como as posições reveladas por um servidor não são necessariamente as mesmas reveladas pelo próximo servidor, pois a mistura feita pelos servidores é independente, a possibilidade de que uma mensagem tenha sua origem descoberta ao passar por toda a rede diminui a cada servidor, tornando-se uma probabilidade calculada de acordo com a revelação parcial definida para cada servidor. Esta probabilidade também se refere à ocorrência de operação desonesta de servidores, no caso de estes estarem sendo controlados por um atacante.

Jakobsson, Juels e Rivest (2002) exemplificam o uso desta probabili-

dade em uma votação. No caso de um atacante conseguir alterar votos (mensagens da rede), a probabilidade de ele não ser detectado reduz exponencialmente com o aumento do número de votos alterados (JAKOBSSON; JUELS; RIVEST, 2002, p. 11), e é expressa por $1/2^{va}$, sendo va o número de votos alterados. Os autores usam como dados de exemplo os votos obtidos no estado americano da Flórida, nas eleições presidenciais americanas do ano de 2001, onde um candidato obteve 2.910.074 votos, e o outro candidato obteve 2.909.114. Neste exemplo, para conseguir alterar o vencedor, o atacante teria que ter conseguido alterar 480 votos, o que resultaria em uma probabilidade de não ser detectado de apenas 2^{-480}, o que é *"muito menor do que a probabilidade de se quebrar um criptosistema tipicamente parametrizado"* (JAKOBSSON; JUELS; RIVEST, 2002, p. 12).

Outra proposta para solução do problema da verificabilidade da operação da rede foi feita por Boneh e Golle (2002). Para a aplicação da rede de mistura em votação digital, Boneh e Golle sugerem a utilização de **redes de recifração**, um conceito originalmente proposto por Park, Itoh e Kurosawa (1993) e que consiste em dividir a operação do servidor em duas fases. Na primeira fase ocorre a mistura e recifração das mensagens, e na segunda fase ocorre a decifração das mensagens.

Nesta proposta, para a obtenção da prova de operação correta, o auditor deve selecionar um subconjunto das mensagens processadas pela rede, e cada misturador fornece a função de permutação utilizada na mistura das mensagens escolhidas, sem entretanto, revelar quais foram as mensagens correspondentes, apenas a permutação de posições é informada. Boneh e Golle mostram a aplicação desta prova de permutação correta utilizando o método de recifração e o criptosistema assimétrico ElGamal (ELGAMAL, 1985). A segurança da prova fornecida baseia-se em algumas propriedades do criptosistema utilizado, que faz com que a obtenção de uma prova forjada seja computacionalmente inviável, embora seja teoricamente possível. A principal vantagem desta proposta é a velocidade de processamento, que é algumas ordens de grandeza mais rápida que outras redes (BONEH; GOLLE, 2002, p. 10).

Dias et al. (2004) propõem a utilização da rede de mistura em um sistema de votação que tem como meio de comunicação as listas de discussão, ferramenta presente no uso de correio eletrônico. Nesta proposta é feita uma ampliação de um sis-

tema já existente, acrescentando a utilização de certificados digitais, e dando possibilidade de anonimato aos votantes através do uso de uma rede de mistura adaptada para correio eletrônico, o que não é possível em uma lista de discussão convencional, onde os participantes são identificados por seus endereços eletrônicos nas mensagens enviadas.

Outra proposta de utilização da rede de mistura em um sistema de votação foi feita por Sako (2004), para ser utilizado em uma empresa japonesa. A rede de mistura foi implementada para utilização em uma rede interna, e fez uso de verificação de correta operação. Como benefícios da utilização do sistema, Sako relata que o sistema recebeu poucas reclamações dos usuários, apresentou um custo operacional de apenas 10% comparado ao valor das votações feitas anteriormente em papel, e o número de votos inválidos reduziu 75%.

3.4.6 Análise da segurança

A rede de mistura apresenta medidas que dificultam ou, em muitos casos, impedem a realização de diversos ataques ao anonimato. Com base nas características encontradas na rede de mistura, propõe-se a seguinte análise da segurança desta RCA, considerando os tipos de ataque ao anonimato, e as possibilidades de defesa encontradas:

- **Ataque à codificação da mensagem:** Por fazer uso sucessivo da criptografia assimétrica em cada encaminhamento das mensagens, a rede de mistura garante que a codificação das mensagens mudará ao passar por um misturador. Assim, qualquer alteração externa em uma mensagem será identificada pelo misturador, impedindo a realização deste ataque;

- **Ataque de temporização:** Pelo fato de cada misturador realizar o encaminhamento das mensagens em uma ordem diferente da recebida, o tempo de envio de uma mensagem entre o emissor e o receptor dependerá da ordem de envio estabelecida pelo misturador, não sendo constante para mensagens sucessivas. Desta forma, a rede de mistura dificulta a realização deste ataque;

- **Ataque de volume de mensagens:** Como os misturadores da rede podem encaminhar as mensagens com um tamanho sempre fixo, através do uso de preenchimento nas mensagens, a realização de um ataque de volume de mensagens não traz informações úteis para o atacante;

- **Ataque de inundação:** A rede de mistura originalmente proposta não prevê medidas específicas para se evitar o ataque de inundação. Entretanto, existem duas medidas que podem ser utilizadas na rede de mistura para dificultar a realização deste ataque. A primeira seria o uso de mensagens de disfarce, que dificultam a inundação realizada pelo atacante, e a segunda medida seria a escolha de uma estratégia de agrupamento de mensagens mais elaborada, incluindo o uso de poças para dificultar a ação do atacante;

- **Ataque de cruzamento:** Como as mensagens cifradas utilizadas na rede de mistura são ininteligíveis para quem não possui a chave utilizada na cifração, o ataque de cruzamento não pode ser realizado em uma rede de mistura;

- **Ataque de marcação da mensagem:** O uso de criptografia no envio das mensagens faz com que qualquer alteração em uma mensagem seja detectada por um misturador. Como a marcação realizada neste ataque altera a mensagem, a detecção feita pelo misturador impede o ataque de marcação da mensagem;

- **Ataque de repetição da mensagem:** Ao manter o registro das mensagens processadas, ou alternativamente utilizando o mecanismo de carimbo de tempo, a rede de mistura impede a realização deste ataque;

- **Ataque de negação de serviço:** A rede de mistura não possui medidas preventivas para este ataque;

- **Ataque do predecessor:** Para que o ataque do predecessor tenha sucesso é preciso que o número de servidores comprometidos pelo atacante seja igual ao número de servidores que constituem um caminho de mensagem (WRIGHT et al., 2002, p. 8). Desta forma, por demandar uma grande quantidade de recursos, a rede de mistura

apresenta melhor defesa a este ataque se comparada ao Roteamento de Cebolas (WRIGHT et al., 2002, p. 3);

- **Ataque da descoberta:** Semelhante ao que ocorre com o roteamento de cebolas, a rede de mistura não possui medidas preventivas para este ataque. Entretanto, o uso de servidores dinâmicos, onde cada cliente pode agir como servidor, pode dificultar a ação do atacante, que terá maior dificuldade em distinguir entre remetentes e destinatários.

Com as proteções contra os ataques ao anonimato presentes na rede de mistura, esta RCA apresenta-se como uma alternativa viável para sistemas que tenham como requisito o anonimato da comunicação em determinado instante, desde que a velocidade na troca de mensagens não seja um requisito desejado, pois a rede de mistura apresenta alta latência na comunicação.

3.5 Conclusão

Ao se comparar a Rede de Mistura com o Roteamento de Cebolas, verifica-se que aquela possui maior segurança do que esta (WRIGHT et al., 2003, p. 4), com a desvantagem de possuir um custo de processamento maior, tanto para os servidores da rede quanto para os usuários.

Dada a complexidade que existe na obtenção do anonimato da comunicação, torna-se essencial a existência de sistemas dedicados ao propósito de garantir a comunicação anônima. Este capítulo apresentou os conceitos existentes em sistemas deste tipo, que podem ser classificados genericamente como Redes de Comunicação Anônima (RCAs).

Estes conceitos são de fundamental importância para a compreensão das propostas encontradas na literatura, as quais se destinam a obtenção de soluções para o problema do anonimato. A principal característica a ser levada em consideração para utilização de técnicas para comunicação anônima é a latência da comunicação que uma

RCA apresenta. Para usuários de serviços que exigem comunicações sem atraso, a utilização de uma RCA de baixa latência torna-se mais adequada, enquanto que para os serviços que não têm o requisito de velocidade na troca de informações, uma RCA de alta latência pode ser suficiente.

O Roteamento de Cebolas, uma RCA de baixa latência apresentada neste capítulo, possui uma arquitetura que permite a sua utilização em diversas aplicações já existentes, que fazem uso de servidores procuradores. A proposta original da Rede de Mistura, uma RCA que possui alta latência, serve como base para diversas outras RCAs encontradas na literatura, tais como **Babel** e **MorphMix**, apresentadas na seção 3.4.4 (página 66).

Outra característica a ser levada em consideração para utilização de técnicas para comunicação anônima é a forma como a RCA trata as mensagens. Em redes como o Roteamento de Cebolas, as mensagens enviadas chegam ao destinatário na mesma ordem em que foram enviadas, o que pode não ser desejável em determinados sistemas, tais como os de votação digital. Para sistemas de votação digital, uma RCA mais indicada para ser empregada é a Rede de Mistura, que possui uma forma de tratamento das mensagens adequada para votação, pois realiza a entrega das mensagens em uma ordem diferente da que foram enviadas. A alta latência existente nesta RCA não é um problema para o sistema, pois a entrega dos votos pode sofrer atrasos sem prejudicar o processo. Esta foi a técnica escolhida para ser utilizada no Protocolo Farnel (DEVEGILI, 2001), um protocolo criptográfico para votações digitais.

As características encontradas nos vários trabalhos analisados, que trazem melhorias para a rede de mistura, serviram de base para a implementação realizada como parte do presente trabalho, apresentada no capítulo seguinte.

Capítulo 4

Implementação de uma RCA

4.1 Introdução

Este capítulo apresenta a implementação de uma rede de comunicação anônima, que foi aplicada a um sistema de votação digital. Este sistema de votação, denominado Sistema Ostracon, é parte do Projeto Ostracon, e tem o objetivo de implementar diferentes protocolos criptográficos para votação digital. Um destes protocolos é o Protocolo Farnel (DEVEGILI, 2001), que também foi desenvolvido dentro do Projeto Ostracon. Neste protocolo, a rede de comunicação escolhida para proporcionar o anonimato dos votantes foi a **Rede de Mistura**.

Esta RCA foi escolhida por proporcionar a entrega das mensagens em uma ordem diferente da que foram enviadas. Conforme discutido na seção 3.4.5 (página 69), esta característica da Rede de Mistura torna-se útil para votação digital, pois o seu funcionamento assemelha-se a uma urna real, onde os votos são inseridos em uma ordem, mas quando os mesmos são retirados estão em outra ordem.

Como este trabalho também faz parte do Projeto Ostracon, a implementação de rede de comunicação anônima apresentada neste capítulo teve o objetivo, primeiramente, de ser utilizada no Protocolo Farnel. A figura 4.1 mostra as entidades existentes no Protocolo Farnel e os caminhos de comunicação existentes. A seção 5.4 (página 111) descreve de forma sucinta a relação entre as entidades. O **Cesto 1** é o ele-

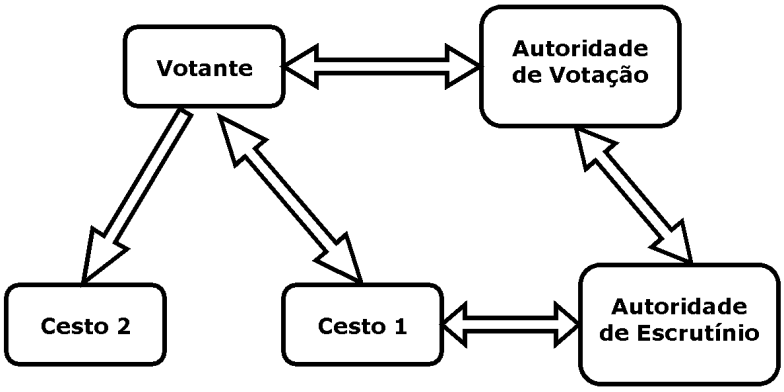

Figura 4.1: Entidades do Protocolo Farnel

mento responsável pela mistura dos votos, auxiliando no sigilo; e o **Cesto 2** é o repositório final dos votos. A implementação apresentada neste capítulo atua como a entidade **Cesto 1** no Protocolo Farnel.

Pela forma como o projeto foi realizado, esta implementação também pode ser adaptada para uso em outros sistemas em rede que necessitem de comunicação anônima. Para a implementação fez-se uso das características observadas nos trabalhos relacionados a redes de comunicação anônima estudados, que foram discutidos nos capítulos anteriores. A figura 4.2 diferencia o envio tradicional de mensagens e o envio de mensagens fazendo uso da JMixNet. Na figura 4.2(a) o remetente envia sua mensagem diretamente para o destinatário, enquanto que na figura 4.2(b) ele utiliza o cliente implementado na JMixNet, permitindo o envio da mensagem ao destinatário de forma anônima, utilizando os servidores da JMixNet.

Como no Protocolo Farnel a utilização da rede de mistura se dá apenas no momento em que o usuário deposita seu voto na urna, não havendo necessidade de tráfego intenso de informações entre o usuário e a rede, fez-se uma implementação de alta latência (veja seção 3.2, na página 33).

Esta implementação da rede de mistura tem a mesma arquitetura proposta inicialmente por Chaum (1981), com servidores interconectados e trocando mensagens cifradas entre si, fazendo uso tanto de criptografia simétrica quanto assimétrica. Por ser uma rede genérica, ela acrescenta recursos que permitem sua utilização em diversas

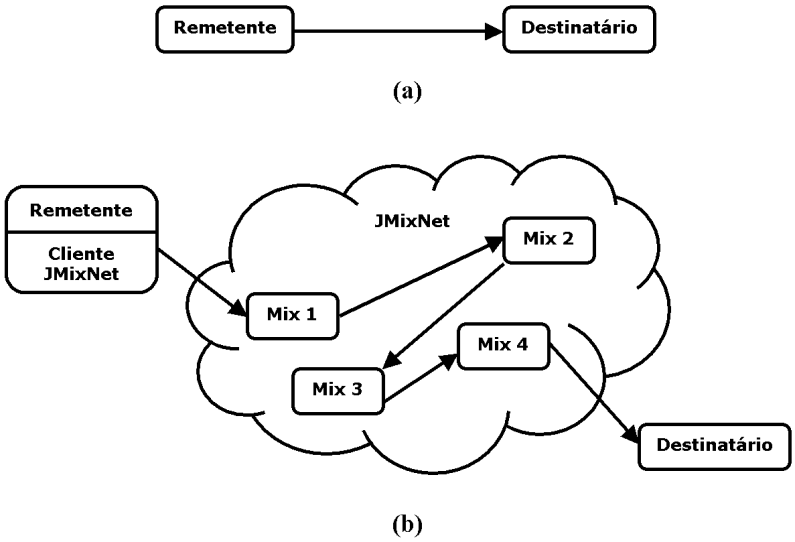

Figura 4.2: (a) Envio tradicional de mensagens, em que o remetente comunica-se diretamente com o destinatário. (b) Envio de mensagens com o uso da arquitetura presente na JMixNet, em que o remetente utiliza o cliente JMixNet para comunicação com o destinatário, através da JMixNet.

aplicações, principalmente aplicações de uso específico, desde que não possuam como requisito a rapidez da comunicação, dada a característica de alta latência.

Com a popularização de sistemas operacionais alternativos, o desenvolvimento de sistemas distribuídos deve considerar a possibilidade de que os serviços sejam executados em sistemas operacionais diferentes. Como a codificação de um mesmo sistema em plataformas diferentes demanda um esforço muito maior de desenvolvimento, a opção por linguagens de programação multiplataforma torna-se uma opção vantajosa. Desta forma, a implementação da rede de mistura foi realizada utilizando a linguagem Java. Para identificar a implementação realizada, esta implementação recebeu o nome de **JMixNet**, por se tratar de uma implementação em linguagem **J**ava de uma rede de mistura (**MixNet**). Esta implementação foi realizada com licença de código aberto (*open source*), e encontra-se disponível em um conhecido sítio de *softwares* de código aberto, o **SourceForge**. Para obter a implementação realizada pode-se acessar o seguinte endereço: `http://jmixnet.sourceforge.net`.

Aqui serão apresentados aspectos de projeto que interferem diretamente na implementação por serem baseados nos estudos realizados em outros trabalhos, sobre

redes de comunicação anônima, apresentados nos capítulos anteriores. Também serão apresentados aspectos práticos, descrevendo alguns detalhes da implementação que facilitam a compreensão da proposta. Nestes casos serão mencionados conceitos e tecnologias presentes na linguagem Java, bem como algumas de suas classes e interfaces para desenvolvimento.

Para evitar confusões com nomenclatura, é preciso diferenciar o servidor da JMixNet, o cliente da JMixNet, e o sistema que faz uso da JMixNet (o qual também age como cliente, neste caso, do serviço provido pela JMixNet). Desta forma, neste capítulo, sempre que se mencionar **módulo servidor**, será referente ao servidor implementado para a JMixNet; ao se mencionar **módulo cliente**, será referente ao cliente implementado para a JMixNet; e ao se mencionar **sistema hospedeiro**, será referente a um sistema que faça uso desta RCA.

Na seção 4.2 são apresentadas as decisões de projeto tomadas para a implementação da JMixNet. A seção 4.3 descreve a arquitetura implementada. Na seção 4.4 é apresentada a forma como as mensagens criptográficas são implementadas e utilizadas na JMixNet. A forma como a JMixNet pode ser utilizada em sistemas com requisitos de anonimato é descrita na seção 4.5. A seção 4.6 apresenta as medidas de desempenho realizadas na implementação. Na seção 4.7 é feita uma comparação da JMixNet com sistemas semelhantes. A seção 4.8 conclui o capítulo.

4.2 Decisões de projeto

Como o objetivo inicial para a JMixNet seria obter uma implementação de Rede de Mistura adaptável ao protocolo Farnel, e como sua aplicação se daria no Sistema Ostracon, o qual agiria como sistema hospedeiro; algumas decisões de projeto foram tomadas com o objetivo de se ter uma implementação funcional em tempo hábil. Para tanto, as decisões de projeto levaram em consideração principalmente a segurança, sem concentrar esforços em questões como performance e escalabilidade, que não é o foco da proposta.

Uma decisão de projeto foi a de utilizar a arquitetura em cadeia (veja seção 3.2.4, na página 36), que faz com que apenas um servidor seja o ponto de entrada de mensagens, e também apenas um (diferente do primeiro) seja o ponto de saída. Apesar de dificultar a sua utilização em sistemas com tráfego muito intenso, pois exigiria um grande consumo de recursos de processamento e de comunicação; a JMixNet apresenta-se como uma solução viável para sistemas com tráfego reduzido, como é o caso da votação digital.

Outra decisão de projeto foi a de utilizar comunicação uni-direcional, permitindo apenas a entrega de mensagens do remetente para o destinatário, sem que este possa enviar uma resposta ao primeiro. Esta característica impede o uso da JMixNet, na forma atual, em sistemas que necessitam de interação entre remetente e destinatário; mas permite a aplicação em votação digital, onde basta o envio do voto para a urna. Para a comunicação bi-direcional é necessário implementar o endereço de retorno não-rastreável (veja seção 3.4.2, na página 62).

O limite de tamanho das mensagens também foi uma decisão de projeto tomada na JMixNet. Uma vez definido o tamanho das mensagens que podem ser enviadas através da rede, o sistema hospedeiro deve prever que suas mensagens tenham o tamanho adequado para envio. O ideal seria fazer com que a rede de mistura dividisse em mensagens menores as mensagens que excedessem o tamanho definido, ficando transparente para o sistema hospedeiro esta divisão. Como em uma votação digital pode-se ter uma previsão do tamanho das informações sobre votos, esta decisão de projeto é adequada para este uso, além da vantagem de simplificar o projeto.

4.3 Arquitetura da rede

A arquitetura formada pelos servidores implementados na JMixNet é a arquitetura em cadeia, e sem redundância de servidor (veja seção 3.2.4, na página 36).

4.3.1 Tratamento de conexões cliente

A classe criada para implementar o módulo servidor na JMixNet foi denominada `JMixNetServer`, e encontra-se no pacote `br.edu.ufsc.labsec.jmixnet`. O módulo cliente criado no projeto JMixNet é implementado pela classe `JMixNetClient`, que encontra-se no mesmo pacote. Este cliente possui os procedimentos necessários para o envio de informações de forma anônima, através da JMixNet.

Como na arquitetura utilizada na JMixNet é preciso que um único servidor receba as mensagens dos usuários, as conexões recebidas precisam ser tratadas rapidamente e com eficiência. Até a versão 1.3 da linguagem Java, cada conexão a um *socket* servidor precisava ser controlada por uma *thread* de processamento em separado, pois até então a operação de leitura e escrita em conexões de rede era **bloqueante**, ou seja, cada leitura e escrita de dados precisava ser feita por completo para que a aplicação pudesse continuar seu processamento, daí a necessidade de *threads* extras, específicas para o manuseio de cada conexão.

Conforme Tanenbaum (2001, p. 83), o uso de *threads* torna necessário um consumo extra de memória para manter informações sobre a *thread*, tais como a pilha de execução e o estado da *thread*, além do consumo de processamento necessário para realização da troca de contextos entre as *threads* existentes. Desta forma, o excesso de *threads* causa um consumo de recursos considerável, o que dificulta a obtenção de um servidor com capacidade para diversas conexões simultâneas. Hitchens (2002, p. 91) descreve o uso de mecanismos denominados **canais** e **selecionadores**, presentes em algumas linguagens de programação, e que tornam desnecessário o uso de uma *thread* para o controle de cada conexão de rede existente em um servidor. Com estes mecanismo é possível fazer com que uma ou algumas *threads* controlem centenas ou até mesmo milhares de conexões, sem uma perda significativa de performance.

Na versão 1.4 do Java surgiram classes que permitem a leitura e escrita de dados de forma **não-bloqueante**, tornando desnecessário o uso de *threads* extra. Para tanto faz-se uso dos mecanismos de **canais** e **selecionadores**. Um canal representa uma ligação com uma entidade que é capaz de realizar operações de leitura e escrita (por

exemplo, uma conexão de rede), e outras operações específicas. Um selecionador funciona como um controlador de canais, sendo capaz de verificar cada canal e informar se cada um deles está pronto para leitura, escrita, ou outra operação (estas operações são denominadas **chaves de seleção**). Para que um canal tome conhecimento de qual selecionador o está controlando, é preciso fazer um procedimento de **registro** do selecionador no canal.

```
1  try {
2      serverChannel = ServerSocketChannel.open();
3      serverChannel.socket()
4          .bind(new InetSocketAddress(config.getPort()));
5      serverChannel.configureBlocking (false);
6      selector = Selector.open();
7  }
8  catch (IOException ex) {
9      if (null == serverChannel)
10         showError("Não foi possível abrir um canal servidor:\n" +
11             ex.getMessage());
12     else if (!serverChannel.socket().isBound())
13         showError("Não foi possível esperar por clientes." +
14         "A porta já está em uso:" + String.valueOf(config.getPort()));
15     throw new JMixNetException(ex);
16 }
17 serverChannel.register(selector, SelectionKey.OP_ACCEPT);
```

Fragmento de Código 1: Início de serviço não-bloqueante

O módulo servidor da JMixNet que recebe as mensagens dos módulos cliente possui um **canal de *socket* servidor** (objeto da classe `ServerSocketChannel` do pacote `java.nio.channels`). Este canal possui a operação de aceitação de conexão. O canal então registra o selecionador da JMixNet (objeto da classe `Selector` do pacote `java.nio.channels`) com a chave de seleção de aceitação, e passa a ser controlado pelo mesmo. Desta forma a aplicação servidora precisa apenas esperar até que o selecionador informe chaves de seleção disponíveis, sem a necessidade de *threads* extra. O Fragmento de Código 1 mostra a abertura do serviço não-bloqueante (linhas 2 a 6), e o registro da operação de aceitação de conexões (linha 17).

Quando ocorre uma conexão cliente, a operação de aceitação é informada pelo selecionador. A aplicação então cria um **canal de *socket*** (objeto da classe

`SocketChannel` do pacote `java.nio.channels`) para a nova conexão, este canal também registra o selecionador da JMixNet, desta vez com a chave de seleção de leitura. Como medida de defesa contra um possível ataque de negação de serviço, a aplicação aceita apenas uma conexão cliente por endereço IP. Quando um cliente envia uma mensagem, a operação de leitura é informada pelo selecionador. O Fragmento de Código 2 mostra a aceitação de uma conexão cliente com configuração não-bloqueante (linhas 2 e 8), e o registro da operação de leitura de dados (linha 9).

```
1  try {
2      channel = serverChannel.accept();
3  }
4  catch (IOException ex) {
5      showError("Não foi possível aceitar conexão de cliente:\n" +
6          ex.getMessage());
7  }
8  channel.configureBlocking (false);
9  channel.register(selector, SelectionKey.OP_READ);
10 showLog("Cliente '" + channel.socket().getInetAddress()
11     .getCanonicalHostName() + "' conectado.");
```

Fragmento de Código 2: Aceitação de conexão cliente

4.3.2 Conexão segura entre servidores

A conexão entre cada servidor da JMixNet é feita mediante autenticação, utilizando o protocolo SSL (*Secure Socket Layer*). Cada servidor, ao se conectar com seu sucessor, apresenta seu certificado de chave pública. Para tanto foi estabelecida uma infra-estrutura de chaves públicas simplificada, que possui uma Autoridade Certificadora, responsável por emitir os certificados utilizados pelos servidores da rede. O certificado de um servidor JMixNet contém no campo *common name* (CN) o seu endereço IP. Isto permite que a autenticação do servidor no momento da conexão seja feita baseada nesta informação.

A linguagem Java dispõe de pacotes de classes relacionadas à segurança para as aplicações que necessitam fazer uso de pares de chaves assimétricas e certificados digitais. Elas fazem uso de entidades denominadas **armazém de chaves** (*keystore*), que

geralmente consistem em um arquivo, para armazenar as chaves privadas e os certificados de chave pública utilizados por uma aplicação. No projeto JMixNet, cada servidor possui o seu armazém de chaves, contendo uma entrada com sua chave privada, associada ao respectivo certificado digital, assinado pela Autoridade Certificadora da JMixNet.

Os sistemas e ambientes que fazem uso de uma infra-estrutura de chaves públicas precisam manter um registro das Autoridades Certificadoras consideradas confiáveis, para que possa ser feita a validação dos certificados utilizados. No ambiente Java este registro consiste em um armazém de chaves situado no sistema de arquivos onde encontra-se instalado o Java, no diretório de configurações de segurança. Para que os certificados da JMixNet sejam reconhecidos como válidos é preciso incluir o certificado auto-assinado da AC JMixNet neste armazém de chaves.

```
1  try {
2      serverSocket = (SSLServerSocket)serverSocketFactory
3          .createServerSocket(config.getPort());
4  }
5  catch (IOException ex) {
6      showError("Não foi possível criar um socket servidor:\n" +
7          ex.getMessage());
8      throw new JMixNetException(ex);
9  }
10 showLog("Esperando na porta " + config.getPort() +
11     " pela conexão do servidor predecessor.");
12 previousServerSocket = (SSLSocket)serverSocket.accept();
```

Fragmento de Código 3: Aceitação de conexão do servidor predecessor

Quando um servidor JMixNet inicia seu serviço ele abre uma porta de rede aguardando pela conexão de seu predecessor (desde que não seja o primeiro da cadeia). Como as conexões entre servidores são feitas com SSL, o servidor utiliza classes específicas para este tipo de conexão. Um objeto da classe `SSLServerSocket` (pacote `javax.net.ssl`) é criado, o qual fica aguardando a conexão do servidor predecessor. Para que o ambiente Java saiba qual chave privada e qual certificado utilizar para as verificações de uma conexão SSL, a aplicação precisa informar qual armazém de chaves deve ser utilizado. Isto é feito através da definição de propriedades do sistema no momento da inicialização do serviço. O Fragmento de Código 3 mostra a criação do *socket* servidor

(linhas 2 e 3), e a espera pela conexão do servidor predecessor (linhas 10 a 12).

4.4 Mensagens utilizadas na rede

O formato das mensagens utilizadas na JMixNet é semelhante ao proposto na rede de mistura. Cada informação que o sistema hospedeiro precise enviar através da JMixNet deve ser cifrada com a chave pública de cada servidor integrante da rede, na ordem inversa da cadeia. Como a cifração assimétrica é mais lenta que a cifração simétrica, além de apresentar limitações quanto ao tamanho máximo do texto original, foi utilizada a técnica do envelope digital, conforme descrito na norma RFC3369 (IETF, 2002). Desta forma, é criado um envelope digital para cada servidor integrante da rede, sendo que o primeiro envelope é colocado dentro do segundo, e assim sucessivamente.

Para a formação dos envelopes digitais de cada servidor, foi utilizado o algoritmo simétrico AES (NIST, 2001), para a cifração dos dados contidos pelo envelope; e o algoritmo assimétrico RSA (RIVEST; SHAMIR; ADLEMAN, 1983) para a cifração da chave simétrica utilizada no passo anterior. O algoritmo de resumo criptográfico utilizado foi o SHA-1 (NIST, 1993).

4.4.1 Tratamento dos envelopes digitais

Com a criação dos envelopes aninhados, o tamanho da mensagem vai aumentando à medida que uma nova cifração é realizada. Se a mensagem final fosse enviada desta forma, a cada envelope retirado em cada servidor da rede, o tamanho da mensagem diminuiria de forma constante, o que possibilitaria um ataque de volume de mensagens (veja seção 2.5.3, na página 23). Para evitar este ataque a JMixNet utiliza o preenchimento de mensagem.

O preenchimento de mensagem implementado faz com que as mensagens enviadas através da rede tenham sempre o mesmo tamanho. O tamanho de mensagem a ser utilizado pode ser definido no arquivo de configuração da rede, e assume um valor padrão caso esta informação não esteja presente na configuração. Com esta medida

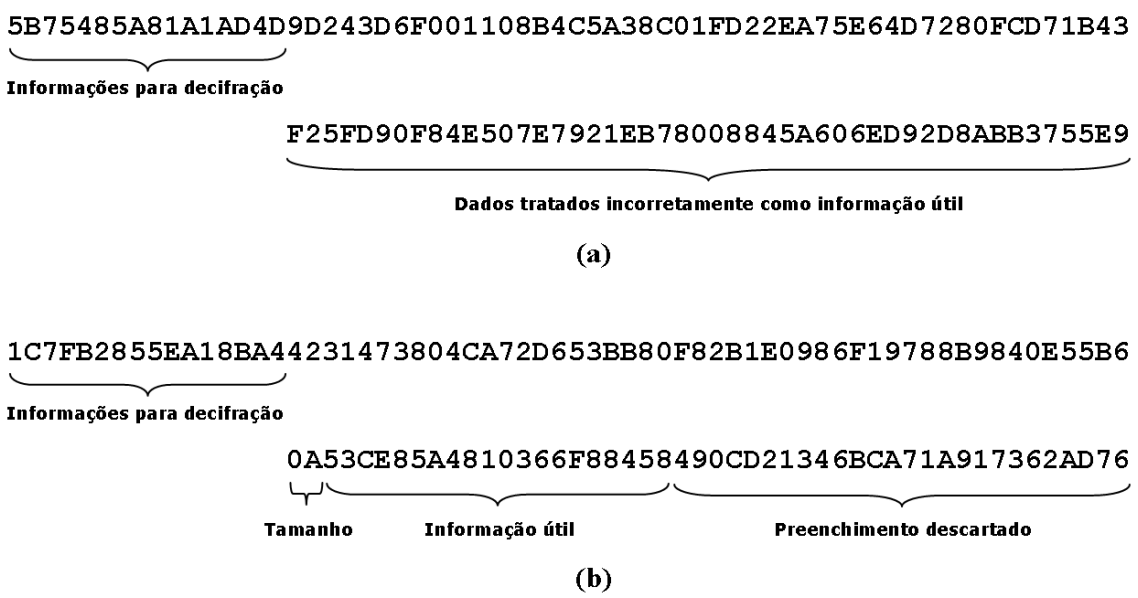

Figura 4.3: (a) Quando não há informações para tratamento da mensagem, todo o resultado da decifração é tratado incorretamente como informação útil. (b) Quando há informações para tratamento da mensagem, apenas uma parte do resultado da decifração é tratado como informação útil, é o restante consiste em preenchimento, que é descartado.

é preciso diferenciar o que é preenchimento, e o que é informação útil que está sendo enviada de forma anônima. Para tanto, ao se criar o primeiro envelope, o módulo cliente da rede acrescenta ao começo da informação a ser enviada, o tamanho da mesma, permitindo ao último servidor da rede fazer a diferenciação ao abrir seu envelope.

A figura 4.3 ilustra a necessidade de se acrescentar informações para o tratamento das mensagens enviadas. Na figura 4.3(a) é mostrado o problema que se tem quando não há informações para o tratamento da mensagem. Neste caso o último servidor, ao abrir o seu envelope, não pode diferenciar o que consiste em informação útil e o que consiste em preenchimento, tratando todo o resultado da decifração como se fosse a informação útil a ser entregue ao destinatário. A figura 4.3(b) demonstra o tratamento correto do resultado da decifração. Ao abrir o envelope, o último servidor pode, à partir dos dados obtidos, diferenciar a informação útil do preenchimento.

A cada envelope criado, o envelope anterior serve como conteúdo para o atual, exceto para o primeiro envelope, que tem como conteúdo a informação útil. O preenchimento é introduzido quando o último envelope é criado, neste momento o con-

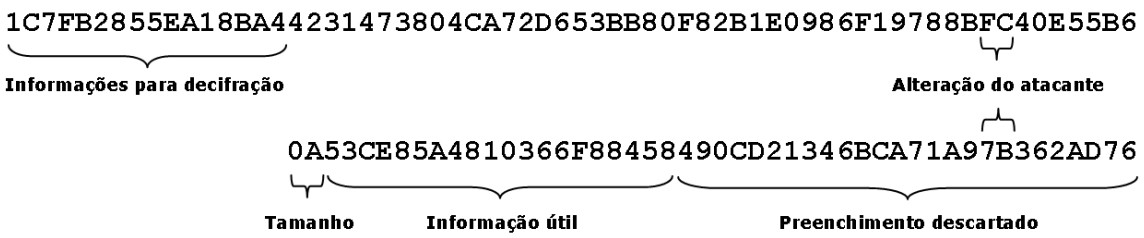

Figura 4.4: Alteração não detectada em mensagem

teúdo do último envelope passa a ser todo o espaço restante na mensagem final, e não apenas o envelope anterior. Desta forma, cada servidor também tratará toda a mensagem recebida como sendo o conteúdo do seu envelope. Isto faz com que toda a mensagem tenha sua codificação alterada ao passar por cada servidor.

O fato de se fazer uso do preenchimento de mensagem facilita a ocorrência do ataque de repetição da mensagem (veja seção 2.5.7, na página 27), pois se um atacante realizar alterações na mensagem no ponto onde se encontra o preenchimento, o ataque não será descoberto, pois o preenchimento é descartado pelo último servidor. Assim, o atacante poderia realizar o ataque sem ter que repetir a mensagem de forma exata, byte por byte.

A figura 4.4 demonstra o problema da não detecção de mensagem repetida que pode ocorrer quando se faz uso do preenchimento de mensagem. Após a decifração o servidor considera como informação útil apenas a quantidade de dados informada pelo campo correspondente, descartando o restante (preenchimento). Caso um atacante tenha feito uma alteração na mensagem no local do preenchimento, o resultado da decifração será diferente, porém a mensagem estará repetida. Isto ocorre pois a parte alterada pelo atacante será tratada como preenchimento, sendo descartada.

A primeira medida tomada na JMixNet para se evitar este ataque consiste no uso de funções resumo, no momento da criação do último envelope pelo módulo cliente. Ao ser criado o último envelope, são submetidos à função resumo o envelope anterior, juntamente com o espaço restante na mensagem final. Desta forma, o conteúdo do último envelope consiste no resultado da função resumo, seguido do envelope anterior, e do espaço restante. Ao receber esta mensagem, o primeiro servidor decifra o envelope,

obtendo o resumo e o restante dos dados. Ele submete o restante dos dados à mesma função resumo utilizada pelo módulo cliente e compara o resumo obtido com o resumo presente no envelope. Caso os resumos sejam iguais, trata-se de uma mensagem verdadeira; do contrário, trata-se de uma mensagem alterada por um atacante, que é descartada. O Fragmento de Código 4 mostra a verificação de resumo de mensagem: primeiramente a mensagem é decifrada (linhas 2 e 3), em seguida o resumo é calculado (linhas 5 e 6), e comparado com o resumo recebido (linha 8).

```
1 //decifrar os dados
2 symCrypto.decrypt(newMsg.array(), sessionKeyFieldSize,
3     messageSize - sessionKeyFieldSize, plainData);
4 //computar e comparar os resumos
5 digester.update(plainData, hashFieldSize,
6     messageSize - sessionKeyFieldSize - hashFieldSize);
7 System.arraycopy(plainData, 0, receivedHash, 0, hashFieldSize);
8 isGoodMessage = MessageDigest.isEqual(digester.digest(), receivedHash);
```

Fragmento de Código 4: Verificação de resumo de mensagem

Desta forma, ainda restaria ao atacante realizar o ataque repetindo as mensagens de forma exata, byte por byte, sem realizar alterações. Neste caso, a proteção utilizada na JMixNet é a utilização de números identificadores únicos (*nonces*). Para tanto, o próprio valor resumo presente na mensagem é utilizado como o número identificador único de uma mensagem. Os resumos são guardados em uma lista, e cada mensagem recebida tem o seu resumo comparado com a lista. Caso ocorra uma repetição, a mensagem é descartada, do contrário a mesma é processada e o seu resumo é incluído na lista. Periodicamente a lista é apagada para se evitar problemas de performance. O Fragmento de Código 5 mostra a verificação de repetição de mensagem: se a lista de resumos recebidos já contiver o resumo atual (linha 1), então a mensagem é tratada como inválida (linha 5).

```
1 strReceivedHash = new String(receivedHash);
2 if (receivedMsgTrack.containsKey(strReceivedHash)) {
3     showLog("Mensagem repetida recebida.");
4     badMsgCount++;
5     isGoodMessage = false;
6 }
```

Fragmento de Código 5: Verificação de repetição de mensagem

4.4.2 Estratégia de agrupamento das mensagens

A estratégia de agrupamento das mensagens (veja seção 3.4.3, na página 65) escolhida para ser utilizada na JMixNet foi a poça dinâmica, aliada à técnica de temporização. O arquivo de configuração da JMixNet permite definir os três parâmetros da estratégia de encaminhamento, os quais assumem valores padrão caso não sejam definidos na configuração.

Assim, a estratégia de agrupamento das mensagens da JMixNet tem o seguinte funcionamento (mostrado no Fragmento de Código 6): as mensagens válidas recebidas são colocadas no repositório, e periodicamente ocorre a verificação do número de mensagens disponíveis para envio (linhas 3 e 4). Caso haja um número de mensagens maior do que o tamanho mínimo definido para a poça, então ocorre a escolha de mensagens para encaminhamento (linha 9 e 10). Antes da obtenção das mensagens que serão encaminhadas, o repositório é reordenado (caracterizando a mistura - *mix*) para que não haja relação na ordem de entrada de mensagens com a ordem de saída (linha 14). Apenas uma parte das mensagens que excedem o número mínimo para a poça são efetivamente encaminhadas, de acordo com a porcentagem configurada.

A classe que implementa o repositório de mensagens da JMixNet, e que estabelece a estratégia de agrupamento de mensagens, é denominada `MessageBatch`, contida no pacote `br.edu.ufsc.labsec.jmixnet`.

```
1  //verificar se a lista possui mensagens suficientes para serem enviadas
2  //ou se já decorreu o tempo máximo de espera
3  if (messageBatch.isReadyToFlush() &&
4      ((new Date()).getTime() - lastFlush > flushPeriod)) {
5      deliverMessages(messageBatch.getFlushList());
6      lastFlush = (new Date()).getTime();
7  }
8  //número de mensagens escolhidas para serem enviadas
9  numChosenMsg = (new Double(flushPercent *
10     (fullBuffers.size() - poolSize) / 100.0)).intValue();
11 //pelo menos uma mensagem deve ser enviada
12 if (numChosenMsg == 0) numChosenMsg = 1;
13 //misturar e deixar apenas as mensagens escolhidas
14 Collections.shuffle(fullBuffers);
15 Object obj;
16 for (i = 0; i < numChosenMsg; i++) {
17     obj = fullBuffers.removeFirst();
18     flushBuffers.add(obj);
19 }
20 return flushBuffers;
```

Fragmento de Código 6: Funcionamento da estratégia de agrupamento das mensagens

4.4.3 Criptografia das mensagens

Na arquitetura de criptografia da linguagem Java, todos os algoritmos de criptografia (cifração, decifração, resumo de mensagem, geração de chaves, assinatura digital, certificados e etc.) possuem uma mesma forma de uso, e são representados por **classes-motor** (*engine classes*). Estas classes definem as funcionalidades de cada algoritmo. As implementações destes algoritmos são feitas com o uso dos chamados **Provedores de Serviços de Criptografia** (*Cryptographic Service Provider - CSP*). Desta forma, um mesmo algoritmo pode ter implementações diferentes, desde que exista mais de um provedor configurado que forneça a implementação do algoritmo desejado.

O ambiente Java possui um provedor padrão, que acompanha a linguagem. Durante os primeiros desenvolvimentos do projeto JMixNet encontrou-se dificuldades devido a limitações presentes neste provedor. Os únicos algoritmos de criptografia assimétrica implementados por este provedor são o de assinatura DSA - *Digital Signature Algorithm* (NIST, 1994), e o de geração de par de chaves DSA. Este provedor não

dá a possibilidade, por exemplo, de se realizar apenas cifrações ou decifrações com o algoritmo DSA, e nenhuma operação com o algoritmo RSA. A realização de cifrações e decifrações assimétricas é justamente a necessidade para a criptografia das mensagens enviadas através da rede de mistura.

Fez-se então uma pesquisa por provedores de criptografia alternativos disponíveis na Internet, e optou-se pelo provedor desenvolvido pelo grupo *Legion of the Bouncy Castle* (www.bouncycastle.org), que é disponibilizado gratuitamente pelo grupo, e com código fonte aberto. Este provedor apresenta um número muito maior de implementações de algoritmos, tanto simétricos quanto assimétricos, do que o provedor padrão do Java.

Atualmente já existem padrões para a troca de mensagens criptográficas entre sistemas distribuídos, sendo um dos mais utilizados a **CMS** (*Cryptographic Message Syntax* - Sintaxe para Mensagem Criptográfica) que é definida pela RFC 3369 (IETF, 2002). No projeto JMixNet optou-se por não utilizar este padrão, pois o mesmo causa um aumento desnecessário no tamanho das mensagens enviadas, pois em cada mensagem dados do certificado digital do destinatário (neste caso um servidor da JMixNet) também são inseridos. Foi feito um teste com alguns certificados digitais, utilizando o padrão CMS, e comparando-o com o formato definido para a JMixNet. Os certificados utilizados continham chaves públicas de 1024 bits, e ocasionaram um aumento médio no tamanho das mensagens de 430 bytes. Para o formato definido na JMixNet, o aumento no tamanho das mensagens é de 128 bytes utilizando-se certificados de 1024 bits.

Além do problema do aumento desnecessário do tamanho da mensagem, o provedor padrão do Java não possui suporte ao CMS. O formato definido para a JMixNet segue o procedimento de criação de um envelope digital.

O Fragmento de Código 7 mostra o procedimento utilizado pelo módulo cliente da JMixNet para a criação de mensagens a serem enviadas pela rede. A informação a ser enviada é cifrada simetricamente com uma chave definida exclusivamente para aquela operação (linhas 3 e 4), criada pelo provedor de criptografia (linha 1). Para o algoritmo AES foram utilizadas chaves de 128 bits, e no modo CBC (*Cipher Block Chaining* - Encadeamento do Bloco de Cifração). Neste modo de operação o bloco de dados deve

ter como tamanho um múltiplo do tamanho da chave (no caso um múltiplo de 16 bytes), do contrário é preciso utilizar preenchimento de bloco. Assim, antes da cifração da informação útil, é acrescentado um número descrevendo o tamanho desta informação (o qual ocupa 2 bytes), e se o tamanho da informação somado a estes 2 bytes não for um múltiplo de 16, são acrescentados tantos bytes de preenchimento quantos forem necessários. Este preenchimento é necessário apenas para o primeiro envelope, pois para os demais o conteúdo já terá um tamanho múltiplo de 16.

```
1  symCrypto.generateKey();
2  //cifrar os dados
3  symCrypto.encrypt(srcBuf, 0, srcBufContentSize,
4      dstBuf, sessionKeyFieldSize);
5  srcBufContentSize += sessionKeyFieldSize;
6  //cifrar a chave simétrica e os parâmetros da cifra
7  asymCrypto.addDataToEncrypt(symCrypto.getEncodedKey(), 0,
8      symCrypto.getEncodedKey().length, dstBuf, 0);
9  encodedParameters = symCrypto.getParameters().getEncoded();
10 asymCrypto.encrypt(encodedParameters, 0,
11     encodedParameters.length, dstBuf);
```

Fragmento de Código 7: Criação de mensagem pelo cliente

A figura 4.5 ilustra a necessidade de preenchimento. Na figura 4.5(a) o tamanho da informação útil somado aos 2 bytes necessários para informar este tamanho não resulta em um múltiplo de 16. Desta forma, torna-se necessário o uso de preenchimento para se atingir um valor deste tipo. Na figura 4.5(b) ocorre o contrário, como o tamanho da informação útil somado aos 2 bytes necessários para informar este tamanho resulta em um múltiplo de 16, não é necessário o preenchimento .

Estes dados com preenchimento são então cifrados simetricamente. Os 16 bytes componentes da chave simétrica juntamente com outros 18 bytes que compõem parâmetros internos do cifrador são tratados como entrada para o cifrador assimétrico RSA (linhas 7 a 11). Este cifrador é inicializado no modo de cifração e recebe o certificado do último servidor da rede, para que sua chave pública seja utilizada na cifração da chave simétrica do envelope, utilizada no passo anterior. Os certificados emitidos para uso na JMixNet possuem um tamanho de chave de 1024 bits. Assim, o primeiro envelope

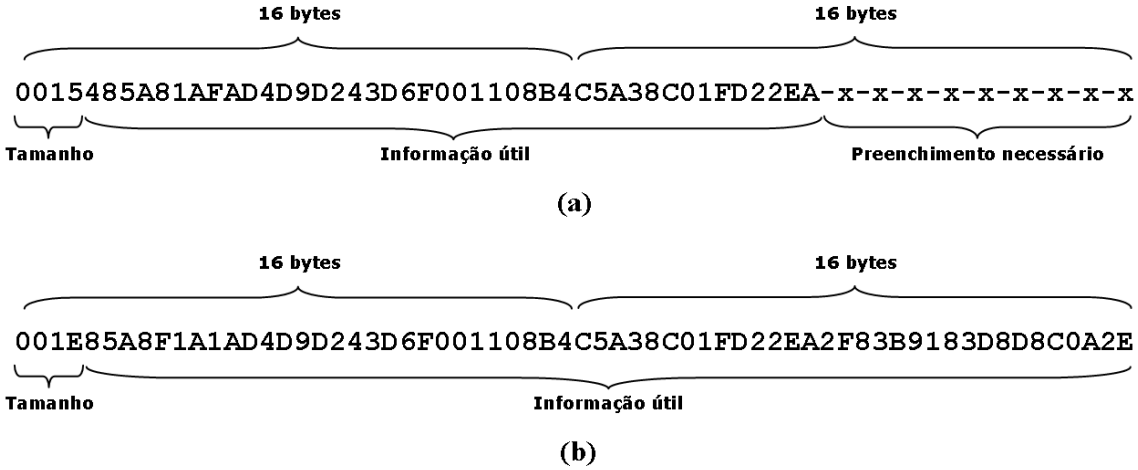

Figura 4.5: (a) Neste caso a informação útil não ocupa um espaço com tamanho adequado, sendo necessário o preenchimento. (b) Caso a informação útil possua um tamanho adequado, o preenchimento torna-se desnecessário.

digital consistirá nos dados cifrados simetricamente (com ou sem preenchimento), unido ao resultado da cifração assimétrica.

Para a formação do segundo envelope o processo é idêntico, sendo que agora o conteúdo deste envelope será o envelope anterior, uma nova chave simétrica será gerada, e o cifrador assimétrico será inicializado com o certificado do penúltimo servidor da rede. Este processo se repete até a criação do penúltimo envelope. Na criação do último envelope o processo é alterado para que se ocupe todo o espaço reservado para a mensagem anônima (considerando todo o espaço disponível como sendo o conteúdo do envelope), e para se evitar eventuais ataques ao anonimato (acrescentando o resumo dos dados). A figura 4.6 ilustra o processo de criação de mensagem anônima por um módulo cliente de uma JMixNet com quatro servidores.

Como é possível notar na figura, o espaço disponível para a informação útil a ser enviada de forma anônima consiste em apenas uma parte dos bytes disponíveis para a composição da mensagem. Este espaço é calculado quando o módulo cliente é iniciado. É considerado o número de servidores integrantes da rede, o espaço para o resumo criptográfico, e os dados sobre o tamanho da informação útil. Caso a informação útil a ser enviada seja maior que este espaço calculado, uma situação de erro é criada, informando sobre a restrição.

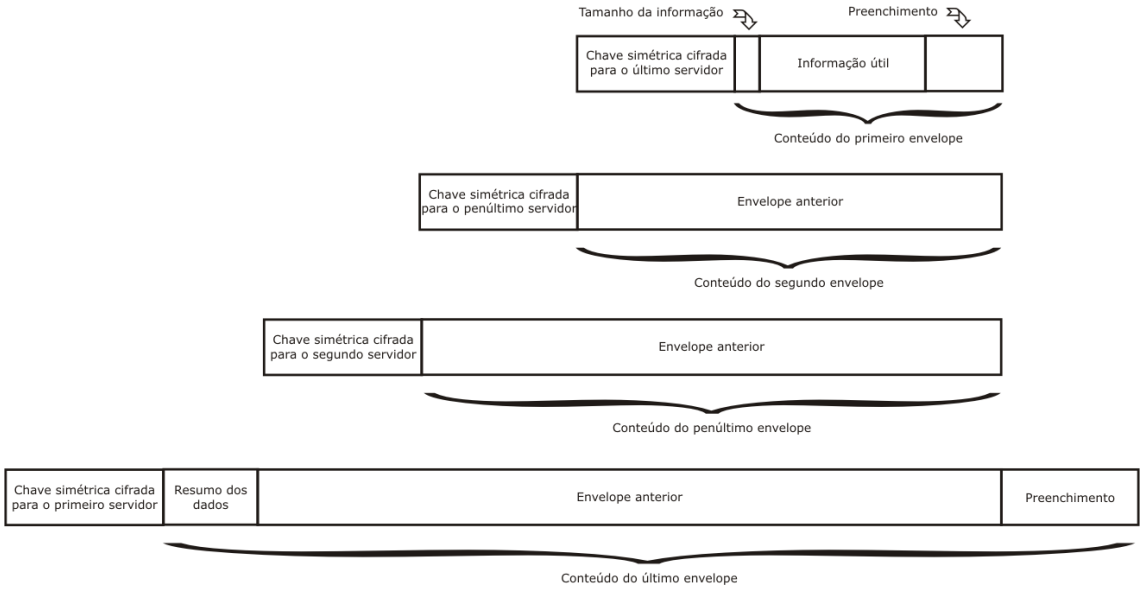

Figura 4.6: Criação de uma mensagem anônima

4.5 Uso em sistemas com requisitos de anonimato

Os sistemas hospedeiros, que possuem como requisito o anonimato da comunicação em determinados momentos da execução, enxergam a RCA utilizada como sendo um agente de entrega, o qual recebe as informações úteis do sistema hospedeiro, e faz com que as mesmas cheguem ao seu destinatário. Desta forma, o sistema hospedeiro de uma RCA precisa saber o formato da mensagem que a RCA utiliza para que suas informações sejam entregues corretamente. No caso da JMixNet, os sistemas que fizerem uso da rede devem utilizar o módulo cliente disponível, que realiza a montagem da mensagem no formato esperado. Para utilização do módulo cliente implementado basta criar um objeto da classe definida, solicitar a conexão à JMixNet, e passar as informações a serem enviadas de forma anônima sempre que necessário.

Em uma RCA também há questões que precisam ser consideradas no que diz respeito ao servidor que recebe mensagens de clientes, e ao servidor que entrega as informações úteis aos destinatários. O primeiro precisa ter meios de saber se aquele cliente está autorizado a utilizar a rede; e o segundo precisa saber como entregar as informações, ou seja, os dados resultantes da última decifração da RCA precisam informar

quem é o destinatário, e de que forma entregar a mensagem.

Com relação ao servidor que recebe mensagens de módulos clientes, a rede JMixNet permite definir quais endereços IP podem ter acesso à rede. Estes dados devem ser informados no arquivo de configuração da JMixNet. Caso estas informações não estejam presentes, o servidor permite a conexão de qualquer cliente, mantendo a restrição de apenas uma conexão por endereço IP. O Fragmento de Código 8 mostra como foi implementado este procedimento: ao se receber uma nova conexão, é feita a verificação de permissão de utilização da rede (linhas 2 e 3), e é feita a verificação de limite de conexão, apenas uma por endereço IP (linha 10).

```
1  //permitir apenas os endereços IP listados no arquivo de configuração
2  if ((allowedClients.size() > 0) && !allowedClients.containsKey(
3      channel.socket().getInetAddress().getHostAddress())) {
4      showLog("Conexão de '" + channel.socket().getInetAddress().
5          getCanonicalHostName() + "' negada.");
6      refusedConnections++;
7      channel.close();
8  }
9  //permitir apenas uma conexão por endereço IP
10 else if (!clientBuffers.containsKey(channel.socket().getInetAddress())){
11     //definir o buffer que este cliente irá usar
12     clientBuffers.put(channel.socket().getInetAddress(),
13         messageBatch.getEmptyBuffer());
14     //configuração do canal
15     channel.configureBlocking (false);
16     channel.register(selector, SelectionKey.OP_READ);
17     showLog("Cliente '" + channel.socket().getInetAddress().
18         getCanonicalHostName() + "' conectado.");
19     connectedClients++;
20 }
```

Fragmento de Código 8: Verificação de permissões de conexão cliente

Para o problema relacionado ao último servidor da rede, referente a forma de entrega das mensagens, a JMixNet permite duas formas de operação. O Fragmento de Código 9 mostra como foi implementada a primeira forma de operação: ao realizar a decifração e obter a mensagem original, o servidor considera que a primeira informação encontrada na mensagem constitui a indicação do tamanho da mensagem real, permitindo descartar o preenchimento de mensagem (linha 2). A segunda informação

consiste no endereço do destinatário (linhas 4 a 6), e o restante da mensagem constitui os dados propriamente ditos, que são entregues ao destinatário (linhas 8 a 11). Nesta primeira forma de operação o servidor não se preocupa com a semântica dos dados, apenas se preocupa em entregá-los.

```
1  //obter a quantidade de dados significativos da mensagem
2  meaningfulMsgSize = msgBuffer.getShort();
3  //obter o endereço e a porta TCP do destinatário
4  msgBuffer.get(addressBuf, 0, 4);
5  receiver = InetAddress.getByAddress(addressBuf);
6  receiverPort = msgBuffer.getShort();
7  //conectar ao destinatário e entregar a mensagem
8  receiverSocket = new Socket(receiver, receiverPort);
9  receiverSocket.getOutputStream().write(msgBuffer.array(), 8,
10     meaningfulMsgSize - 6);
11 receiverSocket.close();
```

Fragmento de Código 9: Entrega de mensagem ao destinatário

Na segunda forma de operação, a JMixNet permite utilizar um procedimento específico de entrega, que depende dos requisitos do sistema hospedeiro. Por exemplo, se o sistema hospedeiro tem como requisito a entrega de mensagens de correio eletrônico, então é necessário utilizar um procedimento específico de entrega, que faça uso dos protocolos adequados. Com esta dependência dos requisitos do sistema, seria necessário alterar o código da JMixNet, fazendo uma adaptação para cada sistema hospedeiro, de forma que a rede se preocupasse também com a semântica das informações a serem entregues. Isto tornaria a utilização da JMixNet mais trabalhosa, pois o administrador do sistema hospedeiro deveria realizar estas alterações no código fonte da JMixNet, de modo a torná-la compatível com seu sistema hospedeiro.

Para solucionar este problema, o projeto JMixNet faz uso da técnica de **reflexão computacional**, através da *Java Reflection API*. A idéia básica da reflexão consiste em se realizar inspeções em classes e objetos Java (SIMMONS, 2004, p. 210). Esta técnica permite a criação de objetos cujas classes só podem ser conhecidas em tempo de execução.

Uma questão comumente presente no uso de reflexão computacional é a segurança relacionada ao código executado, necessária no uso desta técnica. Esta questão, em especial, diz respeito ao uso empregado na JMixNet: como a classe responsável pela entrega das mensagens só é conhecida em tempo de execução, é preciso tomar medidas para garantir que o objeto criado tenha acesso apenas às informações necessárias. No caso da JMixNet esta garantia se dá pelo fato de que a rede não depende de nenhum procedimento presente no objeto criado, ela realiza apenas uma chamada ao objeto, passando os dados obtidos com a última decifração. Esta comunicação limitada impede o objeto criado de realizar qualquer interferência no funcionamento da rede. Além disso, é responsabilidade do administrador do sistema hospedeiro garantir que está utilizando uma classe adequada para o seu sistema, que satisfaça os seus requisitos. Outra possibilidade para a garantia da execução seria a utilização de assinatura de código. Desta forma a JMixNet poderia se recusar a utilizar uma classe que não estivesse assinada, cujo funcionamento tivesse sido previamente verificado e aprovado.

```
1 //obter o nome da classe que deve ser usada para entrega
2 String msgDeliverercClass = config.getMsgDeliverercClass();
3 //criar o objeto responsável pela entrega das mensagens
4 deliverer = (MessageDeliverer)(Class.forName(msgDeliverercClass))
5     .newInstance();
6 //solicitação de entrega de mensagem
7 deliverer.deliverMsg(currentBuffer);
```

Fragmento de Código 10: Entrega de mensagem utilizando reflexão computacional

O uso desta técnica permite ao módulo servidor da JMixNet saber qual procedimento de entrega utilizar em tempo de execução, de forma independente do sistema hospedeiro. Para tanto foi definida a interface `MessageDeliverer`, que possui os métodos utilizados para entrega de mensagens. Assim, o sistema hospedeiro precisa apenas possuir uma classe que implemente esta interface, e informá-la no arquivo de configuração da JMixNet. Quando o último servidor é iniciado, ele realiza chamadas à API de reflexão computacional do Java solicitando a criação de um objeto da classe configurada, e utiliza este objeto para realizar a entrega das mensagens. O Fragmento de Código

10 mostra como foi implementado este procedimento: a classe definida na configuração é lida (linha 2), o objeto responsável pela entrega das mensagens é criado (linhas 4 e 5) e torna-se disponível para entrega de mensagens (linha 7).

Com estas formas de operação dos servidores da rede, qualquer sistema que necessite de comunicação anônima pode utilizar a JMixNet como RCA, mesmo sistemas que não tenham sido desenvolvidos com a linguagem Java. Estes sistemas podem adequar a rede através do arquivo de configuração, e sistemas desenvolvidos em Java podem utilizar a JMixNet de uma forma mais ajustada às suas necessidades.

4.6 Desempenho da rede

Para observar o desempenho da JMixNet, foi considerada principalmente a capacidade de processamento de mensagens por unidade de tempo, tanto pelo módulo servidor quanto pelo módulo cliente. Os testes foram realizados em um computador com processador Intel Pentium 4, com velocidade de 2,4 GHz, e 1Gb de memória RAM. A rede utilizada foi uma rede local *Ethernet* de 100Mbps.

Com relação ao ambiente utilizado para as medições de desempenho (computadores e rede local), a rede local não possui interferência no desempenho do servidor e do cliente, pois a única possibilidade de interferência seria a ocorrência de um "gargalo" na comunicação, caso existisse um número muito grande de clientes. Neste caso, o desempenho do sistema não seria alterado, apesar de os clientes/servidores não poderem enviar/receber tantas mensagens quanto desejariam; seria apenas uma subutilização dos recursos de processamento disponíveis.

4.6.1 Desempenho do Servidor

Para o desempenho do servidor (medido em quantidade de mensagens processadas por segundo) o único fator presente no sistema que influencia na performance é o tamanho das mensagens enviadas através da rede. O número de servidores exerce pouca influência porque cada servidor realiza apenas a sua decifração assimétrica.

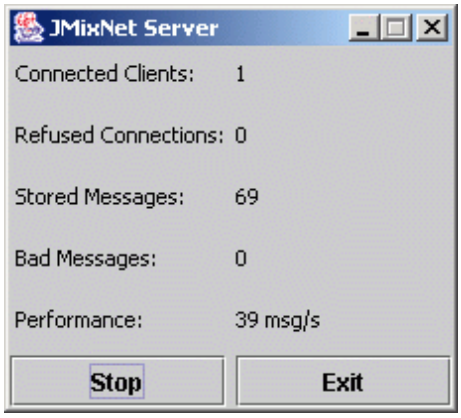

Figura 4.7: Interface gráfica do servidor JMixNet

A tabela 4.1 mostra a média de mensagens por segundo (**mgs/s**) obtidas durante o experimento em cada situação, variando-se o tamanho das mensagens enviadas através da rede (para o cálculo de cada média foram feitas cinco observações). Para facilitar a medição da performance do servidor foi criada uma interface gráfica para o servidor da JMixNet (Figura 4.7).

Tabela 4.1: Medidas de performance do servidor

Tamanho da msg	**1024 bytes**	**4096 bytes**	**16384 bytes**	**65536 bytes**
Performance (msg/s)	61	60	54	38

Com a hipótese inicial de que o aumento do tamanho das mensagens causa uma diminuição na performance do servidor, e com as medições de performance realizadas no servidor, utilizou-se o método estatístico da análise de variância - **ANOVA** (MONTGOMERY, 1996, p. 67) com um fator para a verificação da hipótese e da confiança dos dados experimentais. A tabela 4.2 mostra os dados da análise realizada, em que foi utilizado o nível de significância de 1%.

Como o **Valor p** apresentou-se muito menor do que o nível de significância utilizado (1%), a hipótese inicial de que o aumento no tamanho da mensagem causa uma diminuição na performance do servidor foi confirmada. Isto pode ser visualizado na figura 4.8, que mostra de forma gráfica a degradação da performance do servidor à medida que o tamanho da mensagem aumenta.

Tabela 4.2: Análise de variância para os dados da performance do servidor

Fonte da variação	Soma de Quadrados	Graus de Liberdade	Quadrados Médios	Estatística F	Valor p
Entre grupos	1618,2	3	539,4	719,2	2,87E-17
Dentro dos grupos	12	16	0,75		
Total	1630,2	19			

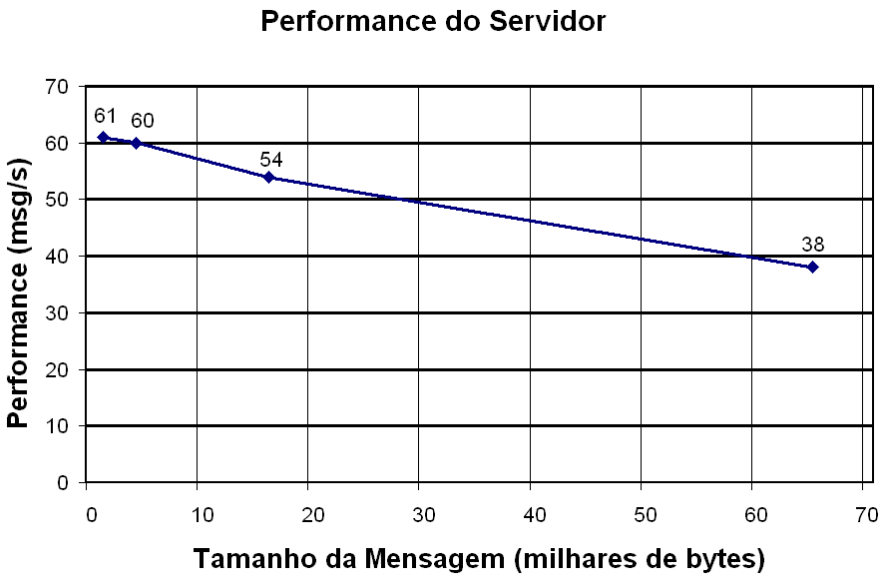

Figura 4.8: Alterações na performance do servidor com o aumento do tamanho das mensagens

4.6.2 Desempenho do Cliente

Para se medir o desempenho do módulo cliente da JMixNet é preciso construir uma situação de uso no limite. Para tanto, algumas medidas foram tomadas. A primeira foi fazer com que o sistema hospedeiro estivesse sempre com dados disponíveis para envio. Outra medida foi fazer com que todas as mensagens enviadas ocupassem o máximo de espaço disponível para informação útil. Além disso, foi desconsiderado o tempo necessário para o envio da mensagem para o servidor, concentrando a medida de desempenho apenas no tempo de processamento.

Foram feitas variações nos dois fatores presentes no sistema que influenciam no desempenho do cliente: o tamanho das mensagens, e o número de servidores

integrantes da rede. O tamanho das mensagens influencia o desempenho por tornar necessária a cifração de uma maior quantidade de dados de forma simétrica, e o número de servidores integrantes da rede também causa influência por aumentar o número de cifrações assimétricas necessárias.

A tabela 4.3 mostra a média de mensagens por segundo (**mgs/s**) obtidas durante o experimento em cada situação, variando-se tanto o tamanho das mensagens, quanto o número de servidores (para o cálculo de cada média foram feitas cinco observações).

Tabela 4.3: Medidas de performance do cliente

Performance (msg/s)	1024 bytes	4096 bytes	16384 bytes	65536 bytes
2 servidores	434	383	254	106
5 servidores	195	170	115	49
10 servidores	102	90	60	26

Com a hipótese inicial de que tanto o aumento do tamanho das mensagens quanto o aumento no número de servidores causam uma diminuição na performance do cliente, e com as medições de performance realizadas no módulo cliente, utilizou-se o método estatístico da análise de variância (**ANOVA**) com dois fatores para a verificação da hipótese e da confiança dos dados experimentais. A tabela 4.4 mostra os dados da análise realizada, em que também foi utilizado o nível de significância de 1%.

Tabela 4.4: Análise de variância para os dados da performance do cliente

Fonte da variação	Soma de Quadrados	Graus de Liberdade	Quadrados Médios	Estatística F	Valor p
Amostra	537379,2	2	268689,6	441681,6	4,4E-103
Colunas	300951	3	100317	164904,6	2,71E-96
Interações	101214,8	6	16869,13	27730,07	3,56E-83
Dentro	29,2	48	0,608333		
Total	939574,2	59			

Como os **Valores p** apresentaram-se muito menores do que o nível de significância utilizado (1%), foi confirmada a hipótese inicial de que tanto o aumento do tamanho das mensagens quanto o aumento no número de servidores causa uma diminui-

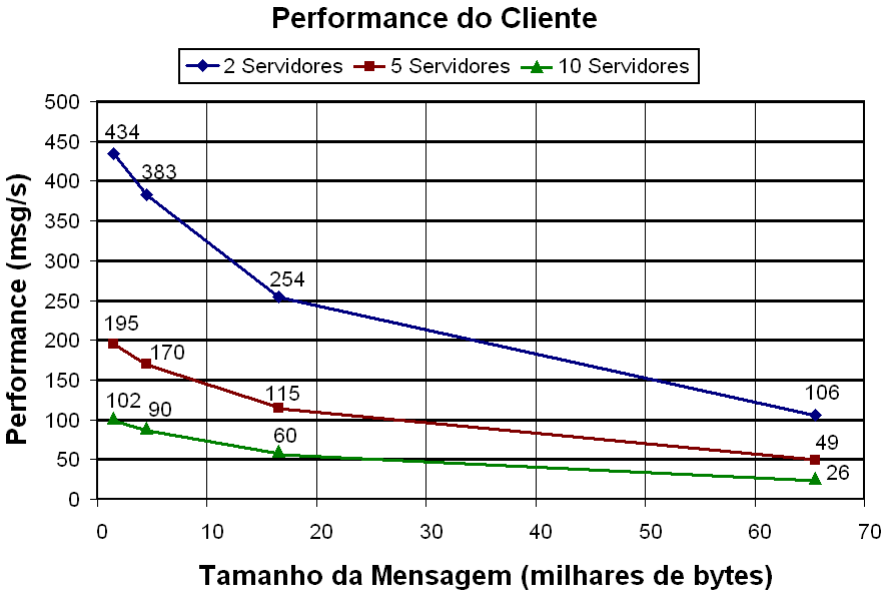

Figura 4.9: Alterações na performance do cliente com o aumento do tamanho das mensagens e do número de servidores

ção na performance do cliente. Pelo **valor p** calculado para interações, observa-se que não há interação entre os fatores.

A figura 4.9 mostra de forma gráfica a degradação da performance do cliente à medida que o tamanho da mensagem aumenta. As três linhas presentes no gráfico representam as três situações de quantidade de servidores testadas.

Comparando-se os melhores casos de ambos os fatores, observa-se que o aumento do número de servidores tem mais influência na queda do desempenho do cliente do que o aumento do tamanho da mensagem: para o caso de 2 servidores (melhor caso deste fator), um aumento considerável no tamanho da mensagem, considerando-se o primeiro valor (1024 bytes) e o último valor (65536 bytes), causou uma diminuição da performance de 75,6%; enquanto que para o caso de mensagens com 1024 bytes (melhor caso deste fator), uma diminuição semelhante (76,5%) foi observada bastando apenas um pequeno aumento no número de servidores, considerando-se o primeiro valor (2 servidores) e o último valor (10 servidores). Desta forma, com a análise de desempenho feita, observa-se que a cifração assimétrica (utilizada com maior intensidade conforme se aumenta o número de servidores) tem mais influência na degradação do desempenho do

cliente do que a cifração simétrica (utilizada com maior intensidade conforme se aumenta o tamanho das mensagens).

4.7 Comparação com sistemas semelhantes

Muitos dos sistemas para comunicação anônima apresentados nos capítulos anteriores são semelhantes à JMixNet, consistindo em implementações de redes de comunicação anônima, com objetivos e aplicações variados. Uma comparação da JMixNet com estes sistemas permite visualizar melhor as possibilidades de melhorias, e o aprimoramento das vantagens já presentes.

O sistema Mixminion, discutido na seção 2.3 (página 14), também consiste em uma implementação de uma rede de mistura, aplicada especificamente para o envio de correio eletrônico. Uma vantagem da JMixNet com relação a este sistema é a possibilidade de utilização em outras aplicações, sem limitar-se ao envio de correio eletrônico. A possibilidade de publicação das chaves públicas dos servidores em serviços de diretório, funcionalidade presente no sistema Mixminion, é uma possibilidade de melhoria para a JMixNet, permitindo um acesso facilitado às informações sobre os misturadores.

Crowds é um sistema para navegação anônima na Web (veja seção 3.2.1, na página 33) que possui um servidor central responsável por controlar a entrada e saída de usuários no grupo de anonimato. Comparando-se a JMixNet com este sistema, pode-se identificar a vantagem que a JMixNet possui em não depender exclusivamente de um servidor central, o que apresenta riscos de segurança e de estabilidade para o sistema. Com a utilização de diversos servidores e com o mecanismo de mistura, a segurança do sistema na JMixNet é distribuída entre todos os misturadores. Neste caso também se verifica a vantagem que a JMixNet possui em não ser limitada a determinada aplicação, como a navegação anônima.

Outra RCA que implementa a rede de mistura para utilização em correio eletrônico é o sistema Babel (veja seção 3.4.4, na página 66). Uma vantagem do sistema

Babel comparado à JMixNet é a possibilidade de prover anonimato mesmo para quem não possui acesso direto ao sistema, pois este pode atuar como um servidor procurador. Desta forma, clientes tradicionais de correio eletrônico podem utilizar esta RCA sem a necessidade de módulos adicionais. Isto não ocorre na JMixNet, que exige a utilização do módulo cliente para que o sistema hospedeiro possa utilizar os serviços de anonimato. Também com relação a este sistema a vantagem da JMixNet consiste na possibilidade de utilização em diversos tipos de aplicação.

A rede MorphMix, discutida na seção 3.4.4 (página 66), também permite o uso em diversas aplicações, semelhante ao que ocorre na JMixNet. A principal diferença encontrada nesta RCA, comparando-se com a JMixNet, é a utilização de servidores dinâmicos, formando uma rede P2P. Esta utilização de servidores dinâmicos apresenta vantagens e desvantagens quando comparada à arquitetura utilizada na JMixNet. Uma vantagem é que o usuário não depende de um número pré-estabelecido de servidores, podendo utilizar qualquer ponto da rede para comunicação. Entretanto, surge a desvantagem de que a rota estabelecida pode ser desfeita a qualquer momento, assim que um dos integrantes da rota decidir deixar a rede. Por ter um número pré-estabelecido de servidores, a JMixNet não apresenta esta instabilidade nas conexões em uso.

4.8 Conclusão

A implementação real de uma rede de comunicação anônima é muito importante para uma correta avaliação das vantagens e desvantagens que um sistema deste tipo pode trazer para usuários que desejam comunicar-se de forma anônima. Mesmo que uma proposta teórica resolva o problema do anonimato, se esta não for viável de forma prática, sua utilidade será apenas uma contribuição conceitual, sem trazer vantagens reais para usuários de serviço de comunicação de dados. Desta forma, uma das contribuições do presente trabalho é a implementação prática de uma rede de mistura, a **JMixNet**, que foi descrita neste capítulo.

A forma como foi feita a implementação permite um uso genérico da

rede em sistemas com requisitos de anonimato. A análise de desempenho realizada mostra a viabilidade prática do uso desta implementação, que atingiu níveis perfeitamente aplicáveis em sistemas existentes, e com equipamentos de custo aceitável. Entretanto, existem algumas limitações que impedem sua utilização em larga escala, mas que podem ser superadas com as devidas alterações na arquitetura da rede. A implementação realizada foi aplicada ao Sistema Ostracon, um sistema de votação digital. O capítulo seguinte descreve este sistema e a sua integração com a JMixNet.

Capítulo 5

Sistema de Votação com Anonimato

5.1 Introdução

Este capítulo apresenta a aplicação prática de uma técnica criptográfica de anonimato feita no **Sistema Ostracon**. Este sistema de votação digital vem sendo desenvolvido dentro do **Projeto Ostracon**, tendo como um dos objetivos a implementação do Protocolo Farnel.

Aqui será apresentado de forma mais completa o Projeto Ostracon (na seção 5.2). Serão apresentados os requisitos de segurança existentes para o sistema (seção 5.3), e de forma sucinta o Protocolo Farnel e o Sistema Ostracon, nas seções 5.4 e 5.5, respectivamente. Na seção 5.6 será apresentada a integração realizada, de uma técnica criptográfica de anonimato (Rede de Mistura - JMixNet) a um sistema de votação (Sistema Ostracon), como contribuição deste trabalho para o Projeto Ostracon. A seção 5.7 conclui o capítulo.

5.2 Projeto Ostracon

O Projeto Ostracon tem como objetivos a realização de pesquisas e o desenvolvimento de soluções de segurança para a realização de votações digitais.

Este projeto teve início no ano de 2000, com a pesquisa para o desen-

volvimento de um protocolo criptográfico para a realização de votações digitais. Esta pesquisa inicial resultou em uma dissertação de mestrado, de Augusto Jun Devegili (DEVEGILI, 2001), que propôs um novo protocolo para votação digital, o **Protocolo Farnel**. Este protocolo foi projetado visando estabelecer um processo semelhante ao que ocorre nas eleições brasileiras, e considerando os requisitos de segurança de uma votação feita de forma digital.

Também no ano de 2000 iniciou-se o desenvolvimento de protótipos de sistemas para votação digital. Estes desenvolvimentos resultaram, no ano de 2001, em um trabalho de conclusão de curso realizado por Fabiano Castro Pereira e Carlos Eduardo Mazzi (PEREIRA; MAZZI, 2001). Este trabalho implementou uma versão simplificada do Protocolo Farnel, e o sistema resultante foi denominado **Sistema Ostracon**. Esta versão inicial do sistema permitiu identificar dificuldades práticas na implementação das questões de segurança presentes no Protocolo Farnel.

Dando continuidade à pesquisa em protocolos de votação digital, no ano de 2001 foi feita uma avaliação de outros protocolos encontrados na literatura, e uma nova implementação do Sistema Ostracon. Estes trabalhos resultaram, em 2002, na dissertação de mestrado de Roberto Samarone dos Santos Araújo (ARAÚJO, 2002). Nessa dissertação, além da análise dos protocolos pesquisados, também foi feito um aprimoramento no Protocolo Farnel, e uma nova análise do cumprimento dos requisitos de segurança desejados.

Esta segunda versão do Sistema Ostracon foi desenvolvida em conjunto com outro trabalho dentro do Projeto Ostracon, que resultou, também no ano de 2002, no trabalho de conclusão de curso de Enrico Golfeto Masela (MASELLA, 2002). Nesta versão o sistema passou a trabalhar com interface Web, e foram implementados dois protocolos de votação digital:

- **Protocolo Simplista**: neste protocolo o voto é conhecido apenas pelo votante e pelo sistema de votação, no qual o votante confia. A comunicação segura entre o votante e o sistema é feita utilizando-se certificados digitais. Este protocolo é destinado a votações com requisitos de segurança pouco exigentes, pois existe a necessidade de confiança no sistema utilizado;

- **Protocolo de Assinatura Cega**: este protocolo permite que o sistema receba os votos e os assine sem ter acesso ao seu conteúdo, através do uso da técnica de assinaturas cegas. Entretanto, para não fazer a relação entre o voto e o votante, este protocolo considera não ser possível identificar o computador utilizado por um votante. Esta é uma deficiência pois os protocolos de comunicação em rede fazem esta identificação implicitamente. Apesar de poder garantir o anonimato (o que depende das condições citadas), este protocolo deixa de atender a determinados requisitos, fazendo com que também seja necessária a confiança no sistema por parte dos votantes.

Além de sistemas para votação digital, o Projeto Ostracon também trouxe contribuições para o Sistema Eleitoral Brasileiro, através do trabalho de Débora Cabral Nazário (NAZÁRIO, 2003). Neste trabalho foi feita uma análise da segurança da Urna Eletrônica Brasileira, e foram sugeridas melhorias para aumentar a segurança do processo eleitoral.

De acordo com a análise feita do Protocolo Farnel, ele atende aos requisitos de segurança necessários para uma votação digital (ARAÚJO, 2002, p. 71). Além disso, outras propostas de protocolos analisadas não possuem implementações viáveis, tendo valor apenas teórico (ARAÚJO, 2002, p. 73). Devido à segurança de atendimento aos requisitos garantida pelo Protocolo Farnel, o Projeto Ostracon vem buscando obter uma implementação funcional deste protocolo. É este um dos principais objetivos do desenvolvimento do Sistema Ostracon. Esta implementação tem importância também por ajudar na especificação formal do protocolo.

Conforme apresentado por Araújo (2002, p. 96), a implementação do Protocolo Farnel necessita principalmente de dois mecanismos criptográficos: o de assinatura cega, e uma rede de mistura adaptada ao sistema. Com a implementação da assinatura cega realizada no Sistema Ostracon, para o Protocolo de Assinatura Cega, uma adaptação pode ser feita para o seu uso de acordo com o previsto no Protocolo Farnel. Desta forma, ao se obter uma implementação Web de uma rede de mistura adaptada ao sistema, tem-se as ferramentas necessárias para uma implementação completa do Farnel.

A principal contribuição do presente trabalho para o Projeto Ostracon consiste nesta implementação de rede de mistura adaptada ao sistema. Foi feita a análise das necessidades previstas no Protocolo Farnel para se obter uma implementação funcional desta RCA, e foram feitos testes de utilização da mesma integrada ao Sistema Ostracon.

5.3 Requisitos de segurança

O Sistema Ostracon engloba os requisitos presentes no Protocolo Farnel. Para a definição teórica deste protocolo foram pesquisadas as necessidades quanto aos requisitos para uma votação digital.

Além dos requisitos funcionais e não-funcionais que podem ser encontrados para o projeto ao se fazer uma análise de requisitos tradicional, orientada por alguma técnica de engenharia de software, observa-se requisitos especificamente relacionados à segurança do processo de votação.

Partindo-se do trabalho de Riera (1999), e do trabalho de Cranor e Cytron (1997), a proposta do Protocolo Farnel chegou a uma classificação dos requisitos de segurança em uma votação digital (DEVEGILI, 2001, p. 26). Estes requisitos podem ser de: **exatidão**, que tratam da consistência dos votos; **democracia**, que definem limites quanto aos votantes; **privacidade**, os quais prezam pelo anonimato e imparcialidade; e **verificabilidade**, que visam prover meios para se garantir a apuração correta dos resultados.

Considerando esta classificação, definiu-se (ARAÚJO, 2002, p. 31-32) os seguintes requisitos de segurança:

- A cédula não pode ser alterada;
- Toda cédula válida deve ser contada na fase de apuração;
- Nenhuma cédula inválida deve ser considerada no momento da apuração;
- Apenas os votantes autorizados podem participar da votação;

- Cada votante pode emitir apenas um voto;
- Não pode ser possível associar uma cédula ao seu respectivo votante (Anonimato);
- Não pode ser permitido que um votante consiga provar o seu voto (Não-coação);
- Todos os votos devem ficar em segredo até o fim da eleição (Imparcialidade);
- Deve ser possível verificar que as cédulas foram contadas corretamente.

O último requisito apresenta-se de duas formas: uma, denominada **verificabilidade universal**, diz que qualquer entidade, independentemente, pode verificar que todas as cédulas foram contadas corretamente; e outra, denominada **verificabilidade individual**, diz que cada votante pode verificar que sua cédula foi contada corretamente.

Uma questão importante é o impasse existente entre os requisitos de **não-coação** e o de **verificabilidade individual**. Ao se garantir este último requisito, também se está fornecendo meios para o votante exibir seu voto a uma terceira pessoa, que pode estar obrigando o mesmo a fazer esta revelação (caracterizando a coação). Desta forma, estes requisitos tornam-se mutuamente exclusivos, ao se garantir um dos requisitos, tem-se a impossibilidade da garantia do outro. Assim, o requisito de verificabilidade individual pode ser garantido apenas em votações onde a possibilidade de ocorrência de coação seja inexistente. Por exemplo, em uma votação entre executivos de determinada empresa, onde os mesmos têm autonomia para não sofrer coação.

Todos estes requisitos podem ser utilizados para avaliar protocolos criptográficos de votação digital, podendo-se definir as situações em que determinado protocolo pode ser utilizado. Esta definição baseia-se na verificação do cumprimento, por parte do protocolo, de cada requisito enumerado.

5.4 Protocolo Farnel

Um processo de votação pode ser dividido em diversas fases, sendo que o Farnel é voltado especificamente para a fase de votação propriamente dita. Entretanto, as outras fases também são importantes para a segurança do processo como um todo, tais como a configuração, o alistamento, o encerramento e a apuração.

Para a operação do protocolo são definidas algumas entidades, denominadas autoridades, que interagem entre si para a realização das comunicações. As entidades definidas pelo protocolo podem ser visualizadas na figura 5.1. A **Autoridade de Votação** é a responsável por coordenar as tarefas da votação; a **Autoridade de Alistamento** emite certificados digitais para que os votantes estejam autorizados a votar, e mantém a lista de votantes; e as **Autoridades de Escrutínio**[1] têm a responsabilidade de garantir que o trabalho da Autoridade de Votação seja feito de forma honesta. Além das autoridades, existem duas outras entidades, denominadas **Cesto 1** e **Cesto 2**, que participam do processo e são responsáveis por receber os votos dos votantes e tratá-los de forma anônima. O Cesto 2 trata-se apenas de um repositório final para os votos realizados. O Cesto 1, que será o responsável pelo anonimato, trata-se de uma rede de mistura (veja seção 3.4, na página 59).

5.4.1 Fases de configuração e alistamento

Na fase de configuração da votação é emitido um certificado digital para a Autoridade de Votação. Este certificado possui uma extensão, de acordo com o padrão X.509v3, que indica à qual votação aquele certificado se destina. É definido um conjunto de Autoridades de Escrutínio, sendo que cada uma das autoridades também recebe um certificado digital para participação da votação. Os certificados, juntamente com outras informações, são publicados em um local denominado **diretório público da votação**.

Também nesta fase é gerado um conjunto de cédulas contendo todas as

[1] No protocolo Farnel, o significado do termo **escrutínio** difere da terminologia utilizada na Justiça Eleitoral, onde este termo diz respeito à contabilização precedida da interpretação/leitura do voto.

possíveis combinações de opções de votos, e caso esteja configurado para repetições do conjunto, o número de repetições deve ser o mesmo para cada opção de voto existente. Isto visa garantir o requisito de privacidade, pois durante a fase de votação é retirado um voto aleatoriamente deste conjunto e inserido o voto recebido. Como todos os possíveis votos estavam presentes, não se pode saber qual o voto inserido. Este conjunto inicial é então assinado digitalmente pelas Autoridades de Escrutínio e colocado no diretório público.

Na fase de alistamento os votantes se autenticam perante a Autoridade de Alistamento, geralmente através de certificados digitais, e recebem um número de identificação, que é incluído na lista de votantes.

5.4.2 Fase de Votação

É nesta fase do processo de votação que se concentra o protocolo Farnel. A figura 5.1 ilustra o funcionamento do protocolo.

O primeiro passo (1) desta fase consiste em se realizar a autenticação mútua entre o Votante (V) e a Autoridade de Votação (AV). O votante apresenta seu número de identificação e o certificado correspondente, presente na lista de votantes. Para ter certeza de que está se comunicando com a Autoridade de Votação correta, o votante obtém o certificado da mesma e verifica a existência do número de identificação da votação. Esta autenticação mútua pode ser feita utilizando o protocolo TLS (IETF, 1999), largamente utilizado para autenticação na camada de transporte, estabelecendo assim um canal seguro de comunicação entre o votante e a Autoridade de Votação.

No segundo passo (2) é gerada a cédula em branco, que é criada pela Autoridade de Votação contendo as opções de voto. As Autoridades de Escrutínio (AE) verificam então a cédula e a assinam, enviando-a de volta para a Autoridade de Votação (3). Ao receber da Autoridade de Votação a cédula assinada (4), o votante verifica a assinatura das Autoridades de Escrutínio, através da chave pública de cada autoridade, e obtém a cédula em branco.

A última parte desta fase consiste na emissão da cédula com votos. Pri-

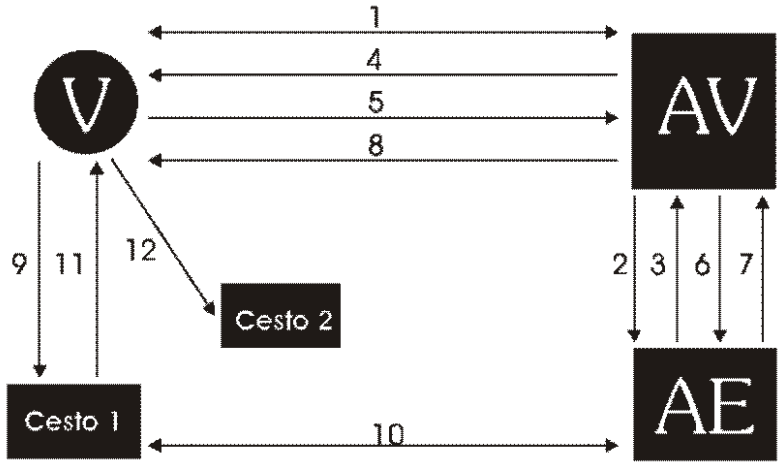

Figura 5.1: Passos do Protocolo Farnel

meiramente o votante assina a cédula em branco, que posteriormente será enviada à Autoridade de Votação como comprovante do recebimento de uma cédula em branco e de entrega de uma preenchida. Esta cédula preenchida precisa ser assinada pelas Autoridades de Escrutínio, porém sem que as mesmas vejam o voto realizado, o que quebraria o requisito de privacidade. Para tanto o Farnel faz uso do mecanismo de assinaturas cegas. O votante oculta a cédula preenchida com um fator de ocultação e depois cria um envelope digital contendo seu número de identificação, a cédula em branco assinada, e a cédula preenchida oculta.

Este envelope é então assinado pelo votante e enviado à Autoridade de Votação (5), a qual o repassa às Autoridades de Escrutínio (6). Cada Autoridade de Escrutínio verifica a assinatura do envelope, de acordo com o certificado digital presente na lista de votantes, registra a entrega do voto, e realiza a assinatura cega da cédula preenchida. Com o voto assinado cegamente por cada Autoridade de Escrutínio, a cédula é devolvida à Autoridade de Votação (7), que a envia ao votante (8). Ao recebê-la, o votante remove o fator de ocultação e obtém então a cédula preenchida e assinada, que é enviada ao Cesto 1 (9) com seu número de identificação.

Ao receber a cédula, o Cesto 1 verifica com as Autoridades de Escrutínio que aquele votante ainda não realizou seu voto (10). Após a confirmação das Autoridades de Escrutínio, o Cesto 1 retira uma cédula aleatória (que sairá do último servidor

integrante da rede de mistura) e a envia ao votante (11). O votante então deposita a cédula recebida no Cesto 2 (12), finalizando o processo de emissão do voto.

Uma questão importante deve ser levada em consideração para a implementação do protocolo Farnel, em especial o processo de emissão do voto. Trata-se do cuidado com as comunicações realizadas entre as diversas entidades existentes, que estão sujeitas à ocorrência de erros na rede de dados. Assim, o processo deve ocorrer de forma transacional, possibilitando o reinício do processo caso ocorram erros de comunicação.

5.4.3 Fases de encerramento e apuração

Na fase de encerramento as Autoridades de Escrutínio solicitam o esvaziamento das cédulas que ainda restarem no Cesto 1. Elas então assinam este conjunto e o depositam no Cesto 2, que passa a conter todas as cédulas assinadas pelas Autoridades de Escrutínio desde o início do processo de votação. Este conjunto de cédulas é então publicado no diretório público da votação.

Na fase de apuração os votos inicialmente criados para preencher o Cesto 1, que abrangiam todas as possibilidades de voto, são retirados do conjunto obtido no encerramento. Este conjunto final, que consiste dos votos realizados pelos votantes, é então colocado no diretório público da votação, bem como o resultado da votação em formato apropriado.

Pelo fato de o conjunto inicial de votos ter sido publicado na fase de alistamento, de o conjunto presente no Cesto 2 ter sido publicado na fase de encerramento, e de o conjunto final resultante da apuração também ter sido publicado, qualquer entidade pode fazer a verificação da exatidão da apuração dos votos.

5.5 Sistema Ostracon

É preciso que a implementação de um sistema de votação digital trate de todos os aspectos abordados por um protocolo criptográfico para votação digital, como o protocolo Farnel, suas definições, entidades envolvidas e fases de operação. Além disso

o sistema deve abordar aspectos de imunidade aos problemas que podem ocorrer quando se trata do uso de redes de computadores, como os ataques a sistemas computacionais, bem como falhas na comunicação e na utilização da rede. Um ponto agravante no aspecto da imunidade a ataques é o fato de um sistema deste tipo fazer uso da Internet, que por ser um meio de comunicação muito utilizado, muitos dos usuários conhecem as deficiências da rede, assim como as formas de se promover ataques.

O Sistema Ostracon, parte do esforço do Projeto Ostracon, tem o objetivo de ser completo em todas as necessidades de uma votação digital, tanto no ponto da implementação de protocolos quanto no que se refere à imunidade a ataques.

5.5.1 Versão inicial

A primeira versão do Sistema Ostracon buscou implementar de forma parcial o protocolo Farnel. Foi feita uma análise da especificação do protocolo visando identificar as necessidades de aplicações para torná-lo funcional. Foram implementadas seis aplicações, sendo que três refletem as entidades do protocolo (Alistamento, Votação e Escrutínio); uma fica responsável por gerenciar o diretório público, outra por atuar como rede de mistura, e a última atua como interface do sistema para com um votante.

Quanto às entidades implementadas, foram feitas duas simplificações com o objetivo de tornar a implementação viável em tempo reduzido. A primeira foi com relação à Autoridade de Escrutínio, sendo implementada apenas uma autoridade, e não um conjunto delas como prevê o protocolo Farnel. A segunda simplificação foi com relação à rede de mistura, que na implementação foi concentrada em apenas uma aplicação. Mesmo com essas simplificações o sistema continuou apresentando as características de segurança proporcionadas por estas entidades, apenas em um nível menor do que se teria com a implementação completa do protocolo.

As fases do protocolo foram implementadas conforme a especificação, diferindo apenas nos locais onde as aplicações simplificadas descritas acima entram em ação. Isto ocorre nas assinaturas realizadas pelas autoridades de escrutínio ao longo do protocolo (que nesta implementação é feita por apenas uma entidade), e no envio de men-

sagens através da rede de mistura (que aqui é constituída por apenas um elemento).

Esta primeira implementação do Sistema Ostracon foi feita sob a plataforma Microsoft Windows e fazendo uso da biblioteca de criptografia deste sistema. Assim, é preciso instalar cada aplicação no seu respectivo local, o que difere da segunda versão do sistema, descrita na próxima seção, que foi feita para o ambiente Web, tornando assim mais simples o uso do sistema.

5.5.2 Versão Web

A segunda implementação do Sistema Ostracon teve como objetivo maior tornar possível o teste de outros protocolos para votação digital além do protocolo Farnel, e de forma a facilitar o uso de um sistema dessa categoria, daí a opção por adotar a interface Web. Esta é uma vantagem em relação à primeira versão do sistema, que necessitava da instalação dos softwares nos computadores dos usuários. Fazendo uso da Web, basta que o votante e os administradores de uma votação utilizem um navegador Web para ter acesso ao sistema.

O sistema foi concebido tendo em vista as fases de um processo de votação digital. Estas fases em geral são aquelas descritas anteriormente: **configuração**, **alistamento**, **votação** e **apuração** (com a devida divulgação dos resultados), sendo que a fase de alistamento pode estar incluída na de configuração (caso seja uma votação onde o administrador mesmo cadastre os votantes aptos), e as demais fases sempre são necessárias para a realização de uma votação digital. Desta forma a segunda versão do Sistema Ostracon dispõe de pelo menos três fases (**configuração**, **votação** e **apuração**) para serem utilizadas no protocolo que estiver sendo implementado.

Nesta nova versão também foram implementadas funcionalidades que vão além daquelas necessárias para um protocolo de votação digital. Pode-se definir características extras de uma votação, tais como os horários de início e término, e regras para aceitação de votantes que desejem alistar-se. Também criou-se mecanismos que permitem a realização de auditorias nos procedimentos da votação, através da publicação dos dados da votação, periodicamente, à medida que a votação vai avançando ao longo das

Figura 5.2: Interface Web do Sistema Ostracon

fases. Contudo, apesar destas possibilidades de auditoria, é de fundamental importância a escolha de um protocolo que tenha implementado o requisito da verificabilidade dos resultados.

A implementação do sistema foi feita utilizando uma linguagem de script no servidor (PHP GROUP, 2004), sendo que para as funcionalidades que necessitam de criptografia foi utilizada a biblioteca OpenSSL (OPENSSL GROUP, 2004), que permite o uso de certificados digitais, criptografia simétrica e assimétrica, funções resumo, e demais ferramentas criptográficas. Para a operação dos protocolos de votação digital é necessário que alguns processamentos sejam realizados pelo cliente, no caso o navegador Web. Para tanto utilizou-se *applets* Java, e scripts em JavaScript para se ter acesso à biblioteca de criptografia no cliente. Os scripts da parte servidora são executados no sistema opera-

cional Linux, enquanto que o acesso pela parte cliente precisa ser feito utilizando-se o navegador Internet Explorer, devido à biblioteca de criptografia utilizada no cliente, que está disponível apenas no sistema operacional Windows. A figura 5.2 exibe a interface do cliente ao acessar o Sistema Ostracon.

Por buscar implementar diferentes protocolos para votação digital o sistema permite que se personalize cada votação, pois se em uma votação em especial não for necessário o requisito do anonimato, então pode-se escolher um protocolo mais simples, que cumpra apenas os requisitos desejados àquela votação em especial.

5.6 Integração da JMixNet

A integração da JMixNet ao Sistema Ostracon pode ser feita de duas maneiras. O restante desta seção descreve estas duas possibilidades. A primeira integração, que foi realizada com o desenvolver do presente trabalho, consiste em criar um novo tipo de protocolo de votação no Sistema Ostracon, fazendo uso da JMixNet para entrega dos votos. Com esta integração o votante não entrega seu voto diretamente para o sistema, mas faz uso da rede de mistura para tanto. A segunda possibilidade de integração da JMixNet ao Sistema Ostracon consiste em aplicá-la ao protocolo Farnel, e acrescentar este protocolo aos protocolos de votação disponíveis no sistema.

5.6.1 Integração realizada ao Sistema Ostracon

A integração da JMixNet ao Sistema Ostracon que foi obtida como resultado do presente trabalho permite o envio dos votos de forma anônima. A JMixNet funciona para o Sistema Ostracon como o elemento responsável por entregar os votos de forma misturada, evitando relacionar o voto ao respectivo votante. E para a JMixNet, o Sistema Ostracon funciona como o sistema hospedeiro, que faz uso dos serviços de entrega de mensagens disponíveis.

A primeira alteração no Sistema Ostracon para integração da JMixNet foi a criação de um novo tipo de protocolo de votação, que foi denominado “Rede de

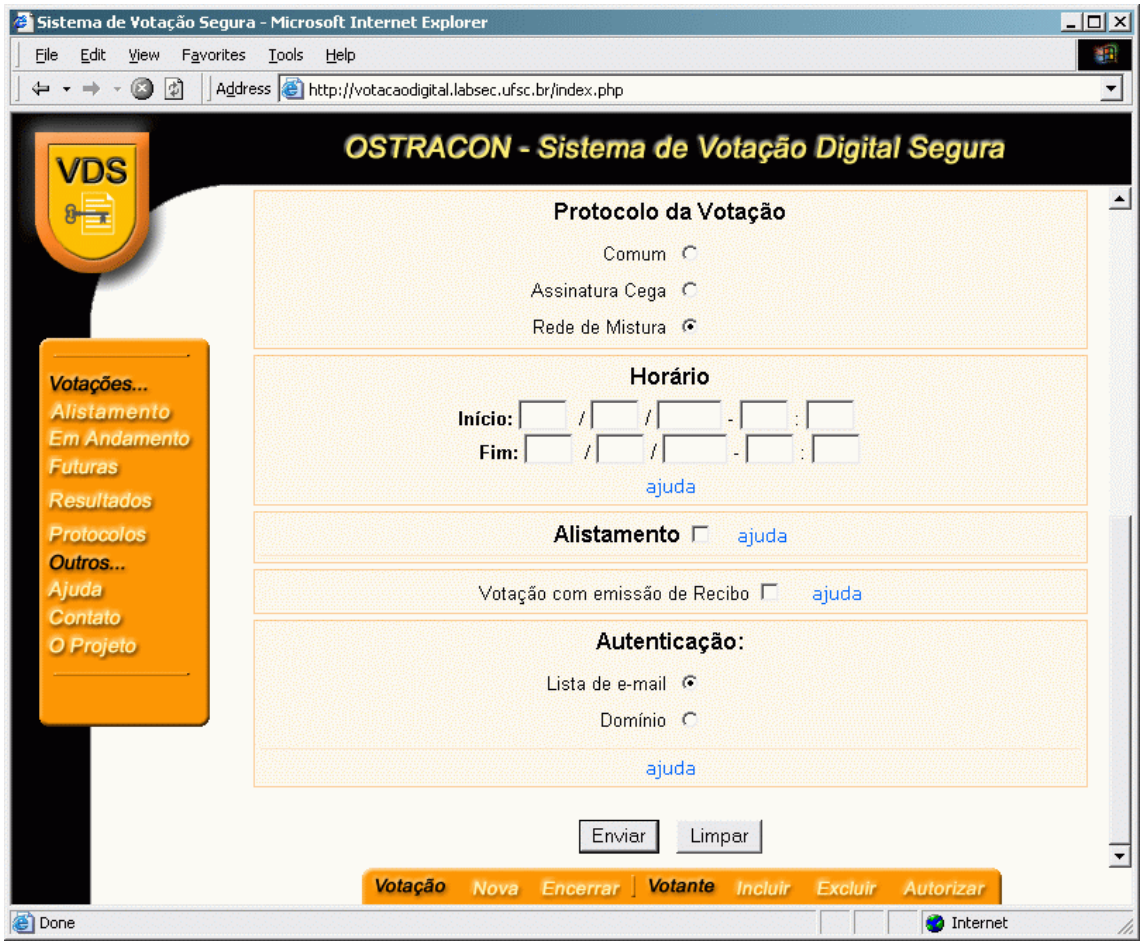

Figura 5.3: Interface para escolha de protocolo para uma nova votação

Mistura" e acrescentado aos dois já existentes: "Simples" e "Assinatura Cega". A figura 5.3 mostra a interface para escolha de protocolo, que é exibida durante a criação de uma nova votação, contendo os dois protocolos já existentes, e o novo protocolo criado.

Uma votação que utilize o protocolo "Rede de Mistura" criado, possui fases de configuração e alistamento idênticas aos demais protocolos do sistema. Na fase de votação é que entram em ação as integrações realizadas. Quando o voto é emitido pelo votante, um *applet* Java é iniciado no navegador, para realizar a entrega do voto. Este *applet* foi implementado através da classe `JMixNetApplet`, contida no pacote `br.edu.ufsc.labsec.farnel`. Ao ser iniciado, o *applet* cria um cliente JMixNet (objeto da classe `JMixNetClient`), e no momento do envio do voto, o *applet* solicita ao cliente o envio de uma mensagem contendo o voto.

Conforme descrito na seção 4.5 (página 94), a JMixNet permite a utilização de um procedimento específico para a entrega das mensagens, que depende dos requisitos do sistema hospedeiro. Para a integração realizada, este procedimento consiste em fazer com que a JMixNet estabeleça uma conexão com o Sistema Ostracon (no caso, o acesso à uma página Web), e entregue as informações do voto.

Para implementar este procedimento de entrega foi criada uma classe `VoteDeliverer`, pertencente ao pacote `br.edu.ufsc.labsec.farnel.jmixnet`. Conforme previsto pela JMixNet, esta classe implementa a interface `MessageDeliverer`, para que seja reconhecida pelo último servidor da rede. Esta classe é informada no arquivo de configuração do último servidor da rede, que quando é iniciado cria um objeto desta classe e delega ao mesmo a entrega das mensagens. Quando uma entrega de mensagem é solicitada, é feita uma conexão com o Sistema Ostracon (via protocolo HTTP), e o voto contido na mensagem é entregue. O Fragmento de Código 11 mostra como foi implementado este procedimento: o voto do votante é obtido (linhas 2 e 3), uma conexão com o Sistema Ostracon é estabelecida (linhas 5 a 7), e o voto é entregue pela JMixNet (linha 9).

```
1 //obter o voto
2 voteSize = msgBuffer.getShort();
3 vote = new String(msgBuffer.array(), 2, voteSize);
4 //conectar ao site do Ostracon
5 connection = (HttpURLConnection)url.openConnection();
6 connection.setRequestMethod("POST");
7 connection.setDoOutput(true);
8 //enviar o voto
9 connection.getOutputStream().write(("voto=" + vote).getBytes());
```

Fragmento de Código 11: Entrega de voto ao Sistema Ostracon

5.6.2 Integração ao Protocolo Farnel

Para integração ao Protocolo Farnel, a JMixNet deve ser utilizada pela entidade **Cesto 1**, entrando em ação na fase de votação (veja seção 5.4.2, na página 112). Conforme o funcionamento da fase de votação do protocolo Farnel, outra responsabili-

dade da entidade Cesto 1, além de garantir o anonimato, é a verificação de que o votante está votando pela primeira vez. Para tanto, o Cesto 1 comunica-se com as Autoridades de Escrutínio, obtendo informações sobre a situação do votante. Desta forma, para a garantia do anonimato basta a utilização da JMixNet, restando a implementação do procedimento de verificação do votante para a implementação completa da entidade Cesto 1.

Para a implementação completa do Protocolo Farnel e utilização no Sistema Ostracon, também é preciso a implementação e integração das demais entidades que o protocolo especifica: Autoridade de Votação, Autoridades de Escrutínio, e Cesto 2.

A melhoria do anonimato advinda da utilização da JMixNet pelo Cesto 1 se dá principalmente pela forma como os votos são entregues à Autoridade de Votação (AV). Sem a utilização de uma rede de mistura, prevista no protocolo Farnel, o votante entregaria seu voto diretamente para a AV. Com o uso de canais seguros de comunicação seria possível garantir que terceiros não tivessem acesso à relação do voto ao respectivo votante, mas ainda restaria a confiança necessária na AV, pois se esta agisse de forma desonesta, esta relação poderia ser divulgada pela AV. Com o uso da JMixNet, o voto é entregue à AV de forma indireta, através da rede de mistura. Com a mistura realizada dos votos, a AV não tem mais meios de estabelecer a relação do voto ao respectivo votante.

5.7 Conclusão

O Projeto Ostracon tem trazido diversas contribuições para a pesquisa em sistemas de votação digital, conforme discutido ao longo deste capítulo. As contribuições abrangem a proposta de um novo protocolo criptográfico, análise de outros protocolos existentes, implementação de sistemas reais para votação digital, e propostas de melhorias para o sistema eleitoral brasileiro.

Uma questão fundamental da pesquisa em votação digital é a garantia do sigilo do voto, de forma a não ocorrer a associação de um voto ao respectivo votante. Conforme discutido em capítulos anteriores, a obtenção de anonimato em comunicações em rede é um problema complexo, que exige sistemas dedicados a este propósito.

A utilização de uma rede de mistura para garantir o anonimato do votante, que é prevista na especificação do Protocolo Farnel, era uma questão que ainda não havia sido estudada com maior profundidade no Projeto Ostracon. Com os resultados de pesquisa do presente trabalho, e com a implementação da JMixNet (também resultado do presente trabalho) foi possível solucionar esta questão do Projeto Ostracon. Estes resultados permitiram a utilização de uma rede de mistura real integrada ao Sistema Ostracon. Este capítulo também contemplou a descrição da integração realizada entre estes dois sistemas.

Capítulo 6

Considerações Finais

Com o grande aumento do uso de redes de computadores, em especial a Internet, diversos perigos surgiram para os seus usuários. Um dos maiores problemas decorrentes disto é a questão do anonimato da comunicação. Em especial, pode-se destacar as ameaças à privacidade dos usuários, que podem ser prejudicados ao terem dados pessoais roubados, tais como números de cartão de crédito, senhas de acesso a bancos *on-line*, e etc.

Os resultados deste trabalho permitem perceber a dificuldade existente em se obter o anonimato, sendo necessários sistemas específicos para este fim. Estes sistemas, que são geralmente denominados Redes de Comunicação Anônima (RCAs), utilizam combinações das técnicas para comunicação anônima pesquisadas. O objetivo principal deste trabalho, de descrever as técnicas empregadas em RCAs, e implementá-las em um sistema real, foi alcançado em sua totalidade. A descrição realizada das técnicas reuniu uma grande quantidade de publicações existentes nesta área de pesquisa em anonimato, abrangendo as diversas características e procedimentos de segurança presentes em RCAs. Isto faz do presente trabalho uma referência para projetistas de sistemas de comunicação que desejem utilizar uma RCA para o anonimato da comunicação. Com as informações aqui presentes é possível escolher a arquitetura, modo de operação, e demais características que devem ser levadas em consideração para a utilização de uma RCA, e que melhor se adaptam aos requisitos estabelecidos.

A definição das técnicas a serem utilizadas pode ser feita comparando-se as características de cada rede, com os requisitos de comunicação e especificamente do anonimato desejado para o sistema em questão. Um exemplo é a característica de latência que cada rede apresenta, um sistema que exija comunicação rápida deve preferir uma rede que possua baixa latência.

A implementação realizada de uma rede de mistura (a JMixNet) também consiste em uma contribuição concreta do presente trabalho. Outras implementações encontradas atualmente dedicam-se especificamente à garantir anonimato no uso de correio eletrônico ou navegação na Web. Com a arquitetura empregada na JMixNet é possível utilizá-la em qualquer sistema que tenha requisitos de anonimato, não se limitando ao correio eletrônico.

Outras contribuições advindas deste trabalho foram os dois artigos científicos publicados. O primeiro (PEREIRA; CUSTÓDIO, 2003), fez uma descrição do Sistema Ostracon, descrevendo também a utilização de uma rede de comunicação anônima neste sistema para a garantia do anonimato do votante. O segunda artigo (DIAS et al., 2004), propôs a utilização de uma rede de mistura em conjunto com um sistema de votação por lista de discussão, mostrando a melhoria no processo democrático que a utilização de um sistema deste tipo pode trazer.

Em especial, destaca-se a contribuição deste trabalho para o projeto maior no qual ele se encontra inserido, o Projeto Ostracon. Com a análise feita da técnica da rede de mistura, foi possível confirmar a viabilidade do uso desta técnica em um sistema de votação digital, como prevê a proposta inicial do Protocolo Farnel (DEVEGILI, 2001). A viabilidade deste uso também pôde ser confirmada através da implementação desenvolvida, que permitiu uma avaliação prática, em um sistema real de votação.

O levantamento feito das possibilidades de ataque ao anonimato foi importante para a definição dos requisitos necessários para a implementação da JMixNet. As características de implementação e decisões de projeto presentes na JMixNet tiveram como base estas características dos ataques, e também as informações obtidas dos outros trabalhos pesquisados.

Mesmo estando em estado de protótipo esta implementação já possui bom desempenho de acordo com a avaliação realizada, cujos dados foram submetidos ao método estatístico da análise de variância, conforme mostrado no presente trabalho. Com pesquisas futuras, melhorias no protótipo implementado podem trazer novas funcionalidades e características para a JMixNet.

Com a arquitetura utilizada para a implementação da JMixNet, e com os resultados de desempenho obtidos tanto no cliente da rede quanto nos servidores, dispõe-se de fatores suficientes para a aplicação da JMixNet em outros sistemas de comunicação em rede. Este uso alternativo trará novos resultados e identificará novas necessidades para a implementação, podendo trazer aprimoramentos e evoluções com o objetivo de se ter uma implementação de um rede de comunicação anônima cada vez mais robusta e funcional.

Também os objetivos específicos inicialmente definidos para este trabalho foram alcançados em sua totalidade:

- A compreensão dos conceitos envolvidos no uso de anonimato (primeiro objetivo específico), e o conhecimento das formas de ataque ao anonimato (segundo objetivo específico), descritos no capítulo 2, foram fundamentais para a pesquisa e o estudo de redes de comunicação anônima;

- Os resultados da pesquisa e estudo de técnicas para comunicação anônima encontradas na literatura (terceiro objetivo específico) estão presentes no capítulo 3, o qual também faz uma análise da aplicação destas técnicas em sistemas existentes (quarto objetivo específico);

- Estes resultados contribuíram principalmente para o projeto e desenvolvimento de uma rede de comunicação anônima (quinto objetivo específico), a JMixNet, descrita no capítulo 4. Esta implementação não teria sido possível sem a identificação das formas de defesa encontradas, que foi fruto da análise realizada das técnicas de anonimato. Para a implementação da JMixNet foram utilizados conceitos complexos para sistemas computacionais, tais como programação paralela, sistemas dis-

tribuídos, e reflexão computacional. A JMixNet é um ponto de partida para muitas pesquisas futuras;

- A aplicação prática da JMixNet (sexto objetivo específico) foi realizada no Sistema Ostracon, e foi descrita no capítulo 5. Este exemplo de integração a um sistema já existente mostra a capacidade presente na JMixNet para se integrar a outros sistemas que tenham o anonimato como requisito. Isto não teria sido possível sem as decisões de projeto tomadas durante a implementação da JMixNet.

É com base nos resultados advindos dos objetivos específicos alcançados que pode-se vislumbrar oportunidades futuras de pesquisa e desenvolvimento, que tenham como base as contribuições feitas neste trabalho.

6.1 Trabalhos Futuros

A realização de pesquisas futuras pode ser concentrada na implementação da arquitetura em clique, que apresenta uma complexidade sensivelmente maior que a arquitetura em cadeia atualmente utilizada pela JMixNet. Esta implementação permitiria um fluxo maior de dados, pois cada servidor integrante da rede poderia atuar como ponto de entrada, intermediário, ou de saída da rede.

Também pode-se realizar pesquisas para tornar possível a replicação de servidores, fazendo com que computadores diferentes possam agir como sendo o mesmo servidor da rede. Desta forma, se um servidor estivesse indisponível, as mensagens destinadas a ele poderiam ser processadas por uma de suas réplicas. Esta funcionalidade aumentaria consideravelmente a tolerância à falhas da JMixNet.

Outra possibilidade de pesquisa seria desenvolver uma forma de implementar o endereço de retorno não-rastreável (veja seção 3.4.2, na página 62) na JMixNet. Esta implementação permitiria a comunicação bidirecional na rede, o que não ocorre atualmente. Aliada a esta implementação, também seria importante fazer com que o tamanho das mensagens enviadas não fosse limitado ao tamanho configurado na rede. Para tanto a JMixNet deveria tratar de dividir as mensagens com tamanho excedente em mensagens de tamanho menor, fazendo a união dos pedaços no momento da entrega.

Pela sua característica de alta latência, a JMixNet não é indicada para sistemas hospedeiros que necessitem de comunicação bidirecional com respostas rápidas, como por exemplo a navegação na Web. Um trabalho futuro seria avaliar a possibilidade de transformar a JMixNet em uma rede de baixa latência.

Outra questão importante a ser pesquisada é a estratégia de agrupamento de mensagens que uma rede de mistura utiliza. Conforme mostrado neste trabalho, existem diversas procedimentos que podem ser utilizados para a implementação de estratégias de agrupamento. Trabalhos recentes têm proposto novas estratégias e têm realizado análises das estratégias existentes. Pesquisas nesta questão trarão benefícios para a rede de mistura, possibilitando um aumento do nível de anonimato provido pelas implementações desta RCA.

A reputação dos misturadores de uma rede de mistura é outra importante questão a ser pesquisada, principalmente com o surgimento de redes de comunicação anônima que utilizam uma configuração com servidores dinâmicos, formando conexões P2P, onde qualquer usuário torna-se também um servidor. A pesquisa por mecanismos de reputação destes servidores é recente, e concentra-se em obter procedimentos criptográficos que permitam o fornecimento de uma prova de correto funcionamento.

Com as diversas técnicas de comunicação anônima estudadas, seria interessante pesquisar a aplicação de uma ou mais destas técnicas aos outros procedimentos presentes no Protocolo Farnel, além da aplicação ao Cesto 1, conforme descrito na seção 5.6.2 (página 120). A pesquisa com a aplicação destas técnicas pode trazer melhorias aos processos já definidos no Farnel.

Outra possibilidade de pesquisa está na definição de métricas para se avaliar a segurança da comunicação proporcionada por determinado sistema, principalmente no que diz respeito ao anonimato e à privacidade. Conforme discutido na seção 3.2.6 (página 41), já existem algumas propostas de quantificação do anonimato; entretanto, melhorias ainda são necessárias para se obter medidas mais abrangentes, que possam ser aplicadas a um número maior de situações e sistemas, incluindo a quantificação da privacidade.

Apêndice A

Glossário

Assinatura cega É uma assinatura digital feita sem se conhecer o conteúdo assinado.

Assinatura digital É uma assinatura realizada em um documento digital utilizando uma chave privada, que permite ao verificador da assinatura saber quem realizou a assinatura, utilizando um certificado digital.

Autoridade de Alistamento É a autoridade responsável por alistar os votantes registrados que desejam participar da uma votação.

Autoridade de Escrutínio É a autoridade responsável por fiscalizar a votação.

Autoridade de Registro É a autoridade responsável por realizar o registro dos votantes.

Autoridade de Votação É a autoridade responsável por realizar a votação.

Certificado Digital Documento digital cuja finalidade é identificar o proprietário de uma chave pública correspondente a uma chave privada.

Envelope Digital Técnica que faz uso da criptografia simétrica e da criptografia assimétrica para o envio sigiloso de uma grande quantidade de dados.

Função Resumo (Hash) É uma função que, dado uma mensagem de entrada de tamanho variável, produz uma saída de tamanho fixo, que identifica a entrada de forma única.

Protocolo Conjunto de passos envolvendo duas ou mais partes, com um objetivo definido.

Protocolo Criptográfico É um protocolo que utiliza criptografia.

Rede de Comunicação Anônima É uma rede formada por servidores que recebem mensagens de usuários remetentes e as entregam para os respectivos destinatários de forma anônima.

Rede de Mistura É uma rede de comunicação anônima que realiza a mistura das mensagens recebidas, entregando as mesmas em uma ordem diferente da que foi recebida.

Roteamento de Cebolas É uma rede de comunicação anônima que envia as mensagens anônimas pelos seus servidores fazendo uso de uma estrutura de dados que se parece com uma cebola, contendo diversas camadas de cifração.

Sistema de Votação Conjunto de programas e computadores que oferecem a estrutura necessária para a realização de uma votação.

Votação Digital É uma aplicação distribuída constituída por um conjunto de protocolos e mecanismos criptográficos que juntos permitem que uma votação aconteça inteiramente sobre redes de computadores, de maneira segura.

Votação Tradicional Votação que utiliza urna e cédulas de papel.

Referências Bibliográficas

AGRAWAL, D.; KESDOGAN, D.; PENZ, S. Probabilistic Treatment of MIXes to Hamper Traffic Analysis. In: *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. [S.l.: s.n.], 2003.

ARAÚJO, R. S. D. S. *Protocolos Criptográficos para Votação Digital*. Dissertação (Mestrado) — Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, 2002.

ARAÚJO, R. S. D. S.; DEVEGILI, A. J.; CUSTÓDIO, R. F. Farnel: Um protocolo criptográfico para votação digital. *WSeg 2002 - Workshop em Segurança de Sistemas Computacionais*, p. 113–120, 2002.

BACK, A.; MÖLLER, U.; STIGLIC, A. Traffic analysis attacks and trade-offs in anonymity providing systems. *Lecture Notes in Computer Science*, v. 2137, p. 245–256, 2001. Disponível em: <http://citeseer.nj.nec.com/back01traffic.html>. Acesso em: 04 de Outubro de 2004.

BERTHOLD, O.; FEDERRATH, H.; KÖHNTOPP, M. Project "Anonymity and Unobservability in the Internet". In: *Workshop on Freedom and Privacy by Design / CFP2000*. [s.n.], 2000. Disponível em: <http://citeseer.nj.nec.com/berthold00project.html>. Acesso em: 04 de Outubro de 2004.

BERTHOLD, O.; FEDERRATH, H.; KÖPSELL, S. Web MIXes: A system for anonymous and unobservable Internet access. *Workshop on Design Issues in Anonymity and Unobservability*, v. 2009, p. 115–129, 2001. Disponível em: <http://citeseer.nj.nec.com/berthold01web.html>. Acesso em: 04 de Outubro de 2004.

BERTHOLD, O.; LANGOS, H. Dummy traffic against long term intersection attacks. In: DINGLEDINE, R.; SYVERSON, P. (Ed.). *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*. Springer-Verlag, LNCS 2482, 2002. Disponível em: <http://www.inf.fu-berlin.de/~berthold/publ/BeLa_02.pdf>. Acesso em: 04 de Outubro de 2004.

BERTHOLD, O.; PFITZMANN, A.; STANDTKE, R. The disadvantages of free MIX routes and how to overcome them. In: FEDERRATH, H. (Ed.). *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*. [S.l.]: Springer-Verlag, LNCS 2009, 2000. p. 30–45.

BONEH, D.; GOLLE, P. Almost entirely correct mixing with application to voting. In: ATLURI, V. (Ed.). *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*. Washington, DC: [s.n.], 2002. p. 68–77. Disponível em: <http://crypto.stanford.edu/~pgolle/papers/psp.ps>. Acesso em: 04 de Outubro de 2004.

BRASIL. Decreto no 3.587, de 5 de setembro de 2000. estabelece normas para a infra-estrutura de chaves públicas do poder executivo federal - icp-gov, e dá outras providências. *Diário Oficial [da] República Federativa do Brasil*, Poder Executivo, Brasília, DF, 5 sep. 2000.

CARVALHO, A. L. ICP-Brasil viabiliza voto via Web. *tema - A REVISTA DO SERPRO*, ago. 2000.

CHAUM, D. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, v. 28, n. 10, October 1985.

CHAUM, D. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, ACM Press, v. 1, p. 65–75, 1988.

CHAUM, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. In: *Communications of the ACM*. [S.l.]: ACM Press, 1981. v. 24, n. 2, p. 84–90. ISSN 0001-0782.

CLEMENTI, A. E. F.; IANNI, M. D. On the hardness of approximating optimum schedule problems in store and forward networks. *IEEE/ACM Trans. Netw.*, ACM Press, v. 4, n. 2, p. 272–280, 1996. ISSN 1063-6692.

CRANOR, L. F.; CYTRON, R. K. Sensus: A security-conscious electronic polling system for the internet. In: *Proceedings of the Hawaii International Conference on System Sciences*. Wailea, Hawaii: [s.n.], 1997.

DANEZIS, G. Mix-networks with restricted routes. In: DINGLEDINE, R. (Ed.). *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760, 2003. Disponível em: <http://www.cl.cam.ac.uk/~gd216/ExpMix.pdf>. Acesso em: 04 de Outubro de 2004.

DANEZIS, G.; DINGLEDINE, R.; MATHEWSON, N. Mixminion: Design of a Type III Anonymous Remailer Protocol. In: *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. [s.n.], 2003. Disponível em: <http://mixminion.net/minion-design.pdf>. Acesso em: 04 de Outubro de 2004.

DEVEGILI, A. J. *Farnel: Uma Proposta de Protocolo Criptográfico para Votação Digital*. Dissertação (Mestrado) — Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, 2001.

DIAS, J. S. et al. Votação anônima segura utilizando lista de discussão. *WSeg 2004 - Workshop em Segurança de Sistemas Computacionais*, p. 177–186, 2004. Disponível em: <http://www.inf.ufsc.br/~castro/artigos/wseg2004.pdf>. Acesso em: 04 de Outubro de 2004.

DÍAZ, C.; PRENEEL, B. Taxonomy of mixes and dummy traffic. In: *Proceedings of I-NetSec04: 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems*. Toulouse, France: [s.n.], 2004. Disponível em: <http://www.esat.kuleuven.ac.be/~cdiaz/papers/cdiaz_inetsec.pdf.gz>. Acesso em: 04 de Outubro de 2004.

DÍAZ, C.; SERJANTOV, A. Generalising mixes. In: DINGLEDINE, R. (Ed.). *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760, 2003. Disponível em: <http://www.esat.kuleuven.ac.be/˜cdiaz/papers/DS03.ps.gz>. Acesso em: 04 de Outubro de 2004.

DINGLEDINE, R.; MATHEWSON, N.; SYVERSON, P. Reputation in P2P Anonymity Systems. In: *Proceedings of Workshop on Economics of Peer-to-Peer Systems*. [s.n.], 2003. Disponível em: <http://freehaven.net/doc/econp2p03/econp2p03.pdf>. Acesso em: 04 de Outubro de 2004.

DINGLEDINE, R.; MATHEWSON, N.; SYVERSON, P. Tor: The second-generation onion router. In: *Proceedings of the 13th USENIX Security Symposium*. [s.n.], 2004. Disponível em: <http://freehaven.net/tor/tor-design.pdf>. Acesso em: 04 de Outubro de 2004.

DINGLEDINE, R.; SHMATIKOV, V.; SYVERSON, P. Synchronous batching: From cascades to free routes. In: *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*. [s.n.], 2004. (LNCS). Disponível em: <http://freehaven.net/doc/sync-batching/sync-batching.pdf>. Acesso em: 04 de Outubro de 2004.

ELGAMAL, T. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, July 1985.

FREEDMAN, M. J.; MORRIS, R. Tarzan: A peer-to-peer anonymizing network layer. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*. Washington, DC: [s.n.], 2002. Disponível em: <http://pdos.lcs.mit.edu/tarzan/docs/tarzan-ccs02.pdf>. Acesso em: 04 de Outubro de 2004.

FRIEDMAN, E.; RESNICK, P. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, v. 10, n. 2, p. 173–199, 2001. Disponível em:

<http://www.si.umich.edu/~presnick/papers/identifiers/081199.pdf>. Acesso em: 04 de Outubro de 2004.

GARFINKEL, S. *Database Nation: The Death of Privacy in the 21st Century*. 101 Morris Street, Sebastopol, CA 95472: O'Reilly & Associates, Inc., 2000. 388 p.

GARFINKEL, S.; SPAFFORD, E. H. *Web Security & Commerce*. 101 Morris Street, Sebastopol, CA 95472: O'Reilly & Associates, Inc., 1997. 506 p.

GOLDBERG, I. *A Pseudonymous Communications Infrastructure for the Internet*. Tese (Doutorado) — UC Berkeley, December 2000. Disponível em: <http://www.isaac.cs.berkeley.edu/~iang/thesis-final.pdf>. Acesso em: 04 de Outubro de 2004.

GOLDBERG, I. Privacy-enhancing technologies for the Internet, II: Five years later. In: DINGLEDINE, R.; SYVERSON, P. (Ed.). *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*. Springer-Verlag, LNCS 2482, 2002. Disponível em: <http://freehaven.net/anonbib/papers/petfive.pdf>. Acesso em: 04 de Outubro de 2004.

GOLDBERG, I.; WAGNER, D.; BREWER, E. Privacy-enhancing technologies for the internet. In: *Proceedings of the 42nd IEEE Spring COMPCON*. IEEE Computer Society Press, 1997. ISBN 0-8186-7804-6. Disponível em: <http://www.cs.berkeley.edu/~daw/papers/privacy-compcon97.ps>. Acesso em: 04 de Outubro de 2004.

GOLDSCHLAG, D.; REED, M.; SYVERSON, P. Onion routing. In: *Communications of the ACM*. ACM Press, 1999. v. 42, n. 2, p. 39–41. ISSN 0001-0782. Disponível em: <http://citeseer.nj.nec.com/goldschlag99onion.html>. Acesso em: 04 de Outubro de 2004.

GOLDSCHLAG, D. M.; REED, M. G.; SYVERSON, P. F. Hiding routing information. In: *Workshop on Information Hiding - Cambridge, UK*. [s.n.], 1996. p. 137–150. Disponível em: <http://citeseer.nj.nec.com/goldschlag96hiding.html>. Acesso em: 04 de Outubro de 2004.

GUAN, Y. et al. An Optimal Strategy for Anonymous Communication Protocols. In: *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems*. [S.l.: s.n.], 2002. p. 257–266.

GÜLCÜ, C.; TSUDIK, G. Mixing E-mail with Babel. In: *Proceedings of the Network and Distributed Security Symposium - NDSS '96*. IEEE, 1996. p. 2–16. Disponível em: <http://citeseer.nj.nec.com/2254.html>. Acesso em: 04 de Outubro de 2004.

HITCHENS, R. *Java NIO*. 101 Morris Street, Sebastopol, CA 95472: O'Reilly & Associates, Inc., 2002. 302 p. ISBN 0-596-00288-2.

HUGHES, D.; SHMATIKOV, V. Information hiding, anonymity and privacy: A modular approach. *Journal of Computer Security*, v. 12, n. 1, p. 3–36, 2004. Disponível em: <http://www.csl.sri.com/users/shmat/shmat_anon.ps>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *Internet Relay Chat Protocol*. [S.l.], 1993. Disponível em: <http://www.ietf.org/rfc/rfc1459.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *The IP Network Address Translator (NAT)*. [S.l.], 1994. Disponível em: <http://www.ietf.org/rfc/rfc1631.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *Internet Protocol, Version 6 (IPv6) Specification*. [S.l.], 1998. Disponível em: <http://www.ietf.org/rfc/rfc2460.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *Security Architecture for the Internet Protocol*. [S.l.], 1998. Disponível em: <http://www.ietf.org/rfc/rfc2401.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *The TLS Protocol Version 1.0*. [S.l.], 1999. Disponível em: <http://www.ietf.org/rfc/rfc2246.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *HTTP State Management Mechanism*. [S.l.], 2000. Disponível em: <http://www.ietf.org/rfc/rfc2965.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *Cryptographic Message Syntax (CMS)*. [S.l.], 2002. Disponível em: <http://www.ietf.org/rfc/rfc3369.txt>. Acesso em: 04 de Outubro de 2004.

IETF The Internet Engineering Task Force. *IPsec-NAT Compatibility Requirements*. [S.l.], 2004. Disponível em: <http://www.ietf.org/rfc/rfc3715.txt>. Acesso em: 04 de Outubro de 2004.

INTERNET POLICY INSTITUTE. *Report of the National Workshop on Internet Voting: Issues and Research Agenda*. [S.l.], 2001. Disponível em: <http://www.digitalgovernment.org/library/library/doc/ipi_onlinevoting.jsp>. Acesso em: 04 de Outubro de 2004.

JAKOBSSON, M.; JUELS, A.; RIVEST, R. Making mix nets robust for electronic voting by randomized partial checking. *USENIX Annual Technical Conference*, 2002. Disponível em: <http://citeseer.nj.nec.com/jakobsson02making.html>. Acesso em: 04 de Outubro de 2004.

JERICHOW, A. et al. Real-Time MIXes: A Bandwidth-Efficient Anonymity Protocol. *IEEE Journal on Selected Areas in Communications*, v. 16, n. 4, 1998.

KESDOGAN, D.; EGNER, J.; BÜSCHKES, R. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In: *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525, 1998. Disponível em: <http://www.uow.edu.au/~ldn01/infohide98.pdf>. Acesso em: 04 de Outubro de 2004.

LEVINE, B. N. et al. Timing attacks in low-latency mix-based systems. In: JUELS, A. (Ed.). *Proceedings of Financial Cryptography (FC '04)*. Springer-Verlag, LNCS 3110, 2004. Disponível em: <http://www.cs.umass.edu/~mwright/papers/levine-timing.pdf>. Acesso em: 04 de Outubro de 2004.

LUOTONEN, A.; ALTIS, K. World-Wide Web proxies. *Journal of Computer Networks and ISDN Systems*, v. 27, n. 2, p. 147–154, 1994. Disponível em: <http://citeseer.nj.nec.com/luotonen94worldwide.html>. Acesso em: 04 de Outubro de 2004.

MASELLA, E. G. *Sistema de votação digital seguro*. 2002. Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Catarina.

MAZIÈRES, D.; KAASHOEK, M. F. The Design, Implementation and Operation of an Email Pseudonym Server. In: *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS 1998)*. ACM Press, 1998. Disponível em: <ftp://cag.lcs.mit.edu/pub/dm/papers/mazieres:pnym.pdf>. Acesso em: 04 de Outubro de 2004.

MONTGOMERY, D. C. *Design and Analysis of Experiments*. [S.l.]: John Wiley and Sons, 1996.

NAZÁRIO, D. C. *Uma Análise da Segurança da Urna Eletrônica Brasileira*. Dissertação (Mestrado) — Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, 2003.

NEFF, C. A. A verifiable secret shuffle and its application to e-voting. In: SAMARATI, P. (Ed.). *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS 2001)*. ACM Press, 2001. p. 116–125. Disponível em: <http://www.votehere.net/ada_compliant/ourtechnology/technicaldocs/shuffle.pdf>. Acesso em: 04 de Outubro de 2004.

NIST National Institute of Standards and Technology. *Secure Hash Standard*. [S.l.], May 1993. Disponível em: <http://www.itl.nist.gov/fipspubs/fip180-1.htm>. Acesso em: 04 de Outubro de 2004.

NIST National Institute of Standards and Technology. *Digital Signature Standard*. [S.l.], May 1994. Disponível em: <http://www.itl.nist.gov/fipspubs/fip186.htm>. Acesso em: 04 de Outubro de 2004.

NIST National Institute of Standards and Technology. *Advanced encryption standard (AES)*. [S.l.], 2001. Disponível em: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. Acesso em: 04 de Outubro de 2004.

OPENSSL GROUP. *OpenSSL*. [S.l.], 2004. Disponível em: <http://www.openssl.org>. Acesso em: 04 de Outubro de 2004.

PARK, C.; ITOH, K.; KUROSAWA, K. Efficient anonymous channel and all/nothing election scheme. In: *Proceedings of EUROCRYPT 1993*. [S.l.]: Springer-Verlag, LNCS 765, 1993. p. 248–259.

PEREIRA, F. C.; CUSTÓDIO, R. F. Ostracon: Um sistema de votação digital segura pela internet. *WSeg 2003 - Workshop em Segurança de Sistemas Computacionais*, 2003. Disponível em: <http://www.inf.ufsc.br/~castro/artigos/wseg2003.pdf>. Acesso em: 04 de Outubro de 2004.

PEREIRA, F. C.; MAZZI, C. E. *Ostracon: Sistema de Votação Digital na Internet*. 2001. Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Catarina.

PFITZMANN, A.; WAIDNER, M. Networks without user observability – design options. In: *Proceedings of EUROCRYPT 1985*. Springer-Verlag, LNCS 219, 1985. Disponível em: <http://www.semper.org/sirene/publ/PfWa_86anonyNetze.html>. Acesso em: 04 de Outubro de 2004.

PHP GROUP. *PHP: Hypertext Preprocessor*. [S.l.], 2004. Disponível em: <http://www.php.net>. Acesso em: 04 de Outubro de 2004.

POLÍCIA CIVIL DE SANTA CATARINA. *Denúncia anônima*. [S.l.], 2004. Disponível em: <http://www.pc.sc.gov.br/bo_cidadao/bo_denuncia_anonima.htm>. Acesso em: 04 de Outubro de 2004.

RACKOFF, C.; SIMON, D. R. Cryptographic defense against traffic analysis. In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. [S.l.]: ACM Press, 1993. p. 672–681. ISBN 0-89791-591-7.

RAO, J. R.; ROHATGI, P. Can pseudonymity really guarantee privacy? In: *Proceedings of the 9th USENIX Security Symposium*. USENIX, 2000. p. 85–96. Disponível em: <http://www.usenix.org/publications/library/proceedings/sec2000/full_papers/rao/rao.pdf>. Acesso em: 04 de Outubro de 2004.

RAYMOND, J.-F. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In: FEDERRATH, H. (Ed.). *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*. Springer-Verlag, LNCS 2009, 2000. p. 10–29. Disponível em: <http://www.geocities.com/j_f_raymond/mesarticles/berkeley_ws_lncs.pdf>. Acesso em: 04 de Outubro de 2004.

REED, M.; SYVERSON, P.; GOLDSCHLAG, D. Proxies for anonymous routing. In: *Proceedings of the 12th Annual IEEE Computer Security Applications Conference*. [S.l.]: IEEE CS Press, 1996. p. 95–104.

REED, M.; SYVERSON, P.; GOLDSCHLAG, D. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection*, 1998.

REITER, M. K.; RUBIN, A. D. Crowds: anonymity for web transactions. In: *ACM Trans. Inf. Syst. Secur.* [S.l.]: ACM Press, 1998. v. 1, n. 1, p. 66–92. ISSN 1094-9224.

RENNHARD, M.; PLATTNER, B. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In: *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*. Washington, DC, USA: [s.n.], 2002. Disponível em: <http://www.tik.ee.ethz.ch/~rennhard/publications/morphmix.pdf>. Acesso em: 04 de Outubro de 2004.

RENNHARD, M.; PLATTNER, B. Practical Anonymity for the Masses with Mix-Networks. In: *Proceedings of the IEEE 8th Intl. Workshop on Enterprise Security (WET ICE 2003)*. Linz, Austria: [s.n.], 2003. Disponível em: <http://www.tik.ee.ethz.ch/~rennhard/publications/WetIce2003.pdf>. Acesso em: 04 de Outubro de 2004.

RENNHARD, M.; PLATTNER, B. Practical anonymity for the masses with morphmix. In: JUELS, A. (Ed.). *Proceedings of Financial Cryptography (FC '04)*. Springer-Verlag, LNCS 3110, 2004. Disponível em: <http://www.cs.umass.edu/~mwright/papers/levine-timing.pdf>. Acesso em: 04 de Outubro de 2004.

RENNHARD, M. et al. Analysis of an Anonymity Network for Web Browsing. In: *Proceedings of the IEEE 7th Intl. Workshop on Enterprise Security (WET ICE 2002)*. Pittsburgh, USA: [s.n.], 2002. p. 49–54. Disponível em: <http://www.tik.ee.ethz.ch/~rennhard/publications/WetIce2002.pdf>. Acesso em: 04 de Outubro de 2004.

RIERA, A. *Design of Implementable Solutions for Large Scale Electronic Voting Schemes*. Tese (Doutorado) — Autonomous University of Barcelona, dec 1999.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. In: *Communications of the ACM*. [S.l.]: ACM Press, 1983. v. 26, n. 1, p. 96–99. ISSN 0001-0782.

SAKO, K. *A Network Voting System Using a Mix-net in a Japanese Private Organization*. 2004. DIMACS Workshop on Electronic Voting – Theory and Practice. Disponível em: <http://dimacs.rutgers.edu/Workshops/Voting/slides/sako.pdf>. Acesso em: 04 de Outubro de 2004.

SERJANTOV, A.; DINGLEDINE, R.; SYVERSON, P. From a trickle to a flood: Active attacks on several mix types. In: PETITCOLAS, F. (Ed.). *Proceedings of Information Hiding Workshop (IH 2002)*. Springer-Verlag, LNCS 2578, 2002. Disponível em:

<http://freehaven.net/doc/batching-taxonomy/taxonomy.pdf>. Acesso em: 04 de Outubro de 2004.

SHAMIR, A. How to share a secret. In: *Communications of the ACM*. [S.l.]: ACM Press, 1979. v. 22, n. 11, p. 612–613. ISSN 0001-0782.

SHERWOOD, R.; BHATTACHARJEE, B.; SRINIVASAN, A. P5: A protocol for scalable anonymous communication. In: *Proceedings of the 2002 IEEE Symposium on Security and Privacy*. [s.n.], 2002. Disponível em: <http://www.cs.umd.edu/projects/p5/p5.pdf>. Acesso em: 04 de Outubro de 2004.

SHIELDS, C.; LEVINE, B. N. A protocol for anonymous communication over the internet. In: *Proceedings of the 7th ACM conference on Computer and communications security*. ACM Press, 2000. p. 33–42. ISBN 1-58113-203-4. Disponível em: <http://citeseer.nj.nec.com/shields00protocol.html>. Acesso em: 04 de Outubro de 2004.

SIMMONS, J. R. *Hardcore Java*. 101 Morris Street, Sebastopol, CA 95472: O'Reilly & Associates, Inc., 2004. 344 p.

SONG, R.; KORBA, L. Anonymous Internet Infrastructure based on PISA Agents. 2001. Disponível em: <http://citeseer.nj.nec.com/song01anonymous.html>. Acesso em: 04 de Outubro de 2004.

SONG, R.; KORBA, L. Review of network-based approaches for privacy. In: *Proceedings of the 14th Annual Canadian Information Technology Security Symposium*. [s.n.], 2002. Disponível em: <http://citeseer.nj.nec.com/song02review.html>. Acesso em: 04 de Outubro de 2004.

SUN, Q. et al. Statistical identification of encrypted web browsing traffic. In: *Proceedings of the 2002 IEEE Symposium on Security and Privacy*. Berkeley, California: [s.n.], 2002. p. 19.

SYVERSON, P.; GOLDSCHLAG, D.; REED, M. Anonymous connections and onion routing. In: *Proceedings of the 18th Annual IEEE Symposium on Security and Privacy*. [S.l.]: IEEE CS Press, 1997. p. 44–54.

SYVERSON, P.; REED, M.; GOLDSCHLAG, D. Private web browsing. *Journal of Computer Security Special Issue on Web Security*, v. 5, n. 3, p. 237–248, 1997.

SYVERSON, P. et al. Towards an analysis of onion routing security. In: *Workshop on Design Issues in Anonymity and Unobservability*. [S.l.: s.n.], 2000.

TANENBAUM, A. S. *Modern Operating Systems*. [S.l.]: Prentice Hall, 2001. 976 p. ISBN 0-13-031358-0.

WAIDNER, M.; PFITZMANN, B. The dining cryptographers in the disco: unconditional sender and recipient untraceability with computationally secure servicability. In: *Proceedings of EUROCRYPT 1989*. [S.l.]: Springer-Verlag, LNCS 434, 1990.

WRIGHT, M. et al. An analysis of the degradation of anonymous protocols. In: *Proceedings of the Network and Distributed Security Symposium - NDSS '02*. IEEE, 2002. Disponível em: <http://www.cs.umass.edu/~mwright/papers/wright-degrade.pdf>. Acesso em: 04 de Outubro de 2004.

WRIGHT, M. et al. Defending anonymous communication against passive logging attacks. In: *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. [s.n.], 2003. Disponível em: <http://www.cs.umass.edu/~mwright/papers/wright-passive.pdf>. Acesso em: 04 de Outubro de 2004.

ZHU, Y. et al. On flow correlation attacks and countermeasures in mix networks. In: *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*. [s.n.], 2004. (LNCS). Disponível em: <http://students.cs.tamu.edu/xinwenfu/paper/PET04.pdf>. Acesso em: 04 de Outubro de 2004.