Serviço de nomes e roteamento para redes de anonimização de tráfego

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Diego de Freitas Aranha e aprovada pela Banca Examinadora.

Campinas, 19 de Março de 2007.

Prof. Dr. Julio César López Hernández
Instituto de Computação (IC) - UNICAMP
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Aranha, Diego de Freitas

Ar14s Serviço de nomes e roteamento para redes de anonimização de tráfego / Diego de Freitas Aranha -- Campinas, [S.P. :s.n.], 2007.

Orientador : Julio César López Hernández

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Criptografia. 2. Sistemas distribuídos. 3. Redes de computadores. I. López Hernández, Julio César. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

(mjmr/imecc)

Título em inglês: Name service and routing for traffic anonymizing networks.

Palavras-chave em inglês (Keywords): 1. Cryptography. 2. Distributed systems. 3. Computer networks.

Área de concentração: Criptografia e Segurança Computacional

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Julio César López Hernández (IC-UNICAMP)
Prof. Dr. Paulo Sérgio L. M. Barreto (Poli-USP)
Prof. Dr. Nelson Luis Saldanha da Fonseca (IC-UNICAMP)
Prof. Dr. Ricardo Dahab (IC-UNICAMP)

Data da defesa: 21/03/2007

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 20 de março de 2007, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Paulo Sérgio Licciardi Messeder Barreto
Escola Politécnica – USP.

Prof. Dr. Nelson Luis Saldanha da Fonseca
IC – UNICAMP.

Prof. Dr. Julio César Lopez Hernández
IC – UNICAMP.

Instituto de Computação
Universidade Estadual de Campinas
Instituto de Computação
Universidade Estadual de Campinas

Serviço de nomes e roteamento para redes de anonimização de tráfego

Diego de Freitas Aranha[1]

Fevereiro de 2007

Banca Examinadora:

- Prof. Dr. Julio César López Hernández
  Instituto de Computação (IC) - UNICAMP (Orientador)
- Prof. Dr. Paulo Sérgio L. M. Barreto
  Escola Politécnica - USP
- Prof. Dr. Nelson Luis Saldanha da Fonseca
  Instituto de Computação (IC) - UNICAMP
- Prof. Dr. Ricardo Dahab
  Instituto de Computação (IC) - UNICAMP

[1]**Financiado pelo CNPq, processo número 131820/2005-2.**

Prefácio

Em diversos cenários, é desejável que não apenas o conteúdo de uma comunicação seja preservado, mas também a identidade dos seus participantes. Satisfazer esta propriedade requer mecanismos diferentes dos comumente utilizados para fornecer sigilo e autenticidade. Neste trabalho, a problemática da comunicação anônima na *Internet* é abordada a partir do projeto e implementação de componentes específicos para este fim. Em particular, são apresentados um componente para roteamento anônimo eficiente em sistemas *peer-to-peer* estruturados e um serviço de nomes para facilitar a publicação de serviços anonimizados.

As principais contribuições deste trabalho são: *(i)* estudo de definições, métricas e técnicas relacionadas a anonimato computacional; *(ii)* estudo do paradigma de Criptografia de Chave Pública Sem Certificados; *(iii)* projeto de uma rede de anonimização completa, adequada tanto para comunicação genérica como para funcionalidade específica; *(iv)* estudo e projeto de esquemas de roteamento em ambientes anônimos; *(v)* projeto de um serviço de nomes que aplica técnicas criptográficas avançadas para fornecer suporte a serviços anonimizados; *(vi)* implementação em *software* dos conceitos apresentados.

Abstract

In several scenarios, it's desirable to protect not only the content of a communication, but the identities of its participants. To satisfy this property, different techniques from those used to support confidentiality and authentication are commonly required. In this work, the problem of anonymous communication on the Internet is explored through the design and implementation of specific components with this function. In particular, a name service and a routing component for anonymous environments are presented.

The main contributions of this work are: *(i)* the study of definitions, metrics and techniques related to computational anonymity; *(ii)* the study of Certificateless Public Key Cryptography, a new model of public key cryptography; *(iii)* the design of a complete anonymization network, suitable for both generic communication and dedicated functionality; *(iv)* the study and design of routing schemes for anonymous communication; *(v)* the design of a name service to support location-hidden services in the anonymous network; *(vi)* the implementation of the concepts presented.

Agradecimentos

À Fernanda, minha namorada, pelo carinho concedido, pelo amor dedicado, pela paciência ao me escutar e pelo sempre presente incentivo.

Aos meus pais, José e Maria José, e aos meu irmãos Pablo e Rodrigo, pelo suporte.

Ao professor Julio López, cujo conhecimento e experiência possibilitaram a conclusão deste trabalho.

Ao professor Ricardo Dahab, pelo apoio e oportunidades importantes.

Ao professor Pedro Rezende, pelas lições valiosas de cidadania.

Aos colegas Leonardo Oliveira, Augusto Devegili, Roberto Gallo, Rafael Castro e Eduardo Morais, pelas discussões pertinentes.

Aos demais que, de alguma maneira, me ajudaram na conclusão deste trabalho: docentes, funcionários e colegas do Instituto de Computação da UNICAMP.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro.

À Criptografia, por servir como um ótimo refúgio.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Algoritmos

Capítulo 1

Introdução

A *Internet* nasceu como ferramenta para a cooperação científica e logo se tornou um ambiente de importância fundamental na sociedade contemporânea. A distribuição frenética de informação propiciada pela rede fornece novas oportunidades de todas as naturezas. Alguns fenômenos humanos, embora milenares, ganharam amplitude nunca antes vista. Entre eles: a propaganda, o comércio e o debate de idéias. Entretanto, ao mesmo tempo que a tecnologia fomenta o progresso, cria desequilíbrios.

Ao se tornar um fator estratégico na batalha por mercados consumidores e hegemonia política, o avanço da tecnologia de telecomunicações tende a prejudicar dois aspectos fundamentais da convivência social: os direitos à *privacidade* e à *liberdade de expressão*.

1.1 Privacidade

A aglomeração de pessoas nas grandes cidades trouxe uma característica inexistente nas pequenas comunidades da Antigüidade. Pela primeira vez na História humana, o indivíduo ganhou o direito de se misturar à multidão, agindo como um desconhecido e interagindo com desconhecidos. A capacidade de proteger informação sobre si mesmo provocou uma revolução na imprensa e estabeleceu novos limites para a atuação governamental. O conceito de privacidade nasceu justamente para conferir ao indivíduo formas legítimas de controlar a exposição de sua própria identidade, e foi concretizado com o seu reconhecimento por parte da legislação e dos direitos humanos.

Infelizmente, as conquistas do passado vêm se esvaindo em nome da segurança nacional e da obtenção de vantagem competitiva. É tendência comum de governos atuais, através da polícia e agências de inteligência, construir bancos de dados de comportamento para traçar o perfil de cidadãos e identificar ameaças antecipadamente. O FBI, por exemplo, tem um histórico

de vigilância eletrônica utilizando detecção de padrão para identificar comunicação suspeita[1]. O governo da Suécia, país onde foi desenvolvida a primeira legislação de privacidade, planeja conferir à sua principal agência de inteligência poderes de vigilância absoluta (sem mandado de segurança) sobre as ligações telefônicas e tráfego internacional que atravessam o país[2]. Um exemplo mais próximo da presença do Estado na invasão de privacidade dos cidadãos é a recente aprovação, pelo Conselho Nacional de Trânsito, da obrigatoriedade da instalação de *chips* de identificação na frota automotiva e de antenas leitoras nas cidades brasileiras[3]. As grandes corporações seguem o exemplo e bombardeiam diariamente seus clientes com formulários de cadastro e pesquisas de mercado; e fornecemos nossos dados pessoais em troca de descontos promocionais. É certo que nossos rostos e vozes são também capturados e armazenados em forma eletrônica durante uma simples visita ao supermercado.

Toda essa informação foge ao controle do indivíduo para residir em sistemas computacionais que poderão mantê-la intacta por décadas. Técnicas de indexação e recuperação permitem a extração de informação de bancos de dados gigantescos quase que instantaneamente, enquanto técnicas de mineração de dados encontram padrões implícitos que podem ser muito reveladores. A *Internet* interliga estes sistemas computacionais e oferece formas extremamente simples de acessar toda esta informação distribuída. O problema, que já era grave, toma nova magnitude.

Muitas instituições detém e fazem uso de informações privadas da população, mas poucas delas estão preparadas para protegê-las[4]. Não há qualquer garantia de que dados pessoais não serão roubados por um invasor ou comercializados por um funcionário inescrupuloso, tanto que os dados pessoais das declarações de Imposto de Renda de 2006 foram encontrados à venda em camelôs do centro de São Paulo[5]. A necessidade de mecanismos que devolvam o equilíbrio, permitindo que as pessoas exerçam maior controle da informação que revelam de si mesmos, é evidente.

Como o usuário não pode controlar o vazamento dos seus dados armazenados em governos e empresas, ele pode no mínimo tentar limitar a informação que fornece diretamente à *Internet*. Ao menos temporariamente, já que uma medida recente, cuja votação foi adiada no Senado, obriga a identificação de todos os usuários brasileiros da rede por meio de certificados digitais, sob pena de reclusão[6]. Correio eletrônico, mensagens em fóruns e informações de cadastro são

[1] *Governmentality and the war on terror: FBI Project Carnivore and the diffusion of disciplinary power* : `http://www.cas.sc.edu/socy/faculty/deflem/zgovernterror.html`

[2] *Protests over plans to let military spy on Swedes:* `http://www.sr.se/cgi-bin/International/nyhetssidor/artikel.asp?ProgramID=2054&Nyheter=&format=1&artikel=113865`

[3] Contran cria novo sistema de identificação de veículos: `http://www.denatran.gov.br/ultimas/20061122_sis_identificacao.htm`

[4] O próprio autor já teve sua impressão digital solicitada para cadastro em uma locadora de filmes, sem que fosse apresentado um documento com regulamentações e procedimentos para utilização dessa informação.

[5] Apreendidos CDs com informações sigilosas da Receita Federal: `http://www.ssp.sp.gov.br/home/noticia.aspx?cod_noticia=8856`

[6] Senado adia votação de projeto que obriga identificação de usuários na *Internet*: `http://www1.folha.uol.`

sujeitos à captura por terceiros e criptografia deve ser utilizada para o transporte e armazenamento deste conteúdo explícito. Mas o conteúdo implícito continua desprotegido: o próprio padrão de visitação de páginas *Web* de um usuário já pode revelar muito a seu respeito (banco, padrão econômico, posicionamento político). Por mais que o conteúdo explícito seja protegido por criptografia, a ligação entre este conteúdo e os endereços de origem e destino associados continua clara. Uma opção imediata para reduzir a quantidade de informação que fornecemos implicitamente na rede é acessar os recursos da *Internet* de forma *anônima*.

1.2 Liberdade de expressão

A liberdade de expressão resguarda o direito de divulgação de informação de qualquer natureza. O valor desta liberdade, na sociedade contemporânea, é extremo. É confortante contar com a sensação de que se é livre para formar e manter opiniões, quaisquer que estas sejam.

Quando uma entidade tem o poder de controlar deliberadamente a informação que chega à sociedade, a habilidade de manipular opiniões ou esconder fatos estabelece sem demora um regime de mentiras ou meias verdades, onde qualquer informação contraditória é banida do domínio público pela censura. Apesar da situação parecer um tanto distante, já é prática notável de alguns governos divulgar informação manipulada e, em um volume tal, que as pessoas passam a tomá-la por autêntica. Agindo assim, o governo viola os princípios democráticos que justificam a sua própria existência. A única forma de se garantir uma democracia eficaz é permitir que a população possa compartilhar informação livremente. Enquanto tudo o que for ouvido e visto pela sociedade for filtrado, não há verdadeira liberdade.

A instalação de mecanismos de censura começa a se tornar prática perniciosa à liberdade de expressão na *Internet*. A legitimação desta prática vem por meio da aprovação de legislação sob influência dos grandes consórcios das indústria fonográfica, cinematográfica e de *software*. A DMCA (*Digital Millenium Copyright Act*), legislação americana para proteção de direitos autorais aprovada no ano 2000, exige que os provedores de acesso retirem do ar imediatamente qualquer conteúdo que possa violar direitos de cópia, antes mesmo que seja determinada a validade da acusação. Mais gravemente, impede indivíduos de publicar resultados de pesquisa autônoma em programas de computador e dispositivos eletrônicos, sem a autorização dos fabricantes. O *INDUCE Act*, outra legislação americana introduzida recentemente, transporta a responsabilidade da utilização de um programa de computador para o desenvolvedor: se o programa for utilizado de forma criminosa ou para violar direitos de cópia, o desenvolvedor pode ser acusado como cúmplice. Os recentes esforços na direção do patenteamento de *software* e na adoção de dispositivos de computação confiável (*Trusted Computing – TC*) para gerência de direitos digitais (*Digital Right Management – DRM*) são outro exemplo claro de censura: as

com.br/folha/informatica/ult124u20934.shtml

empresas de *software* proprietário e multimídia objetivam estabelecer formalmente o controle das plataformas computacionais e ampliar ainda mais o monopólio que detém, eliminando toda forma de concorrência, como a demonstrada pelas iniciativas de *software* livre.

Além da legislação que cerceia os direitos dos usuários da rede, ainda contamos com governos que declaradamente censuram o tráfego que entra os seus territórios. Recentemente, uma lista com os treze países que mais censuram a *Internet* foi divulgada[7]. São eles, em ordem decrescente de censura: Myanmar, China, Bielorrússia, Irã, Tunísia, Cuba, Egito, Arábia Saudita, Turcomenistão, Vietnã, Coréia do Norte, Síria e Uzbequistão. As técnicas comumente utilizadas nestes países abrangem a instalação de filtros de conteúdo que interrompem qualquer comunicação que envolva idéias controversas. O governo chinês, por exemplo, utiliza um *firewall* avançado apelidado de *Grande Firewall da China* [CMW06], para filtrar a parcela de tráfego indesejável ao regime.

O assunto é polêmico e traz posições diversas. É tentador imaginar que a censura pode exercer papel favorável à sociedade em algumas circunstâncias, como no controle de propaganda racista ou de divulgação de material moralmente questionável. Seria então dever do governo impedir que as pessoas defendam idéias danosas à sociedade? Primeiramente, não é possível conferir o poder de censura de boa vontade sem fornecer o poder nocivo análogo. Para se impor qualquer forma de censura, a entidade responsável ganha automaticamente a habilidade de monitorar e restringir a circulação de informação. Existe ou não existe censura, não há meio termo. Além disso, é importante observar que se a sociedade for exposta aos seus próprios males, mecanismos de defesa sadios podem ser desenvolvidos e aprimorados por esta, com conseqüentes melhorias na formação de cidadãos conscientes.

A proteção da liberdade de expressão na *Internet* em face de governos autoritários, com o poder de manipular a mídia, registros históricos e ideologia[8] depende essencialmente da publicação *anônima* de conteúdo. Mais amplamente, conjectura-se a verdadeira liberdade de expressão, em qualquer veículo de comunicação, depende do anonimato.

1.3 Aplicações reais

A aplicação fundamental de uma ferramenta de anonimização na Internet é permitir que informação, mesmo que indigesta ao governo ou certas elites, possa ser publicada e distribuída livremente na rede. Mais do que isso e provavelmente ainda mais importante, objetiva garantir o direito de qualquer cidadão acessar e contribuir com a informação que circula pela rede, independentemente da posição que o seu governo sustenta. É fundamental fornecer verdadeira

[7] *Nations that censor the Internet*: `http://www.businessweek.com/technology/content/nov2006/tc20061109_790623.htm?campaign_id=bier_tcv.g3a.rss1112d`

[8] George Orwell, em sua ficção *1984*, caracteriza de forma realista a sobrevivência sob um regime dessa natureza.

liberdade de expressão no mundo digital, já que a análoga do mundo real está sujeita à forte repressão pela polícia e à ação alienante da propaganda. Para não restringir as aplicações apenas à publicação de conteúdo, comunicação anônima pode ser empregada em diversos outros cenários de risco [Dan04a] [Mar99]:

- Proteger a identidade de denunciantes em sistemas de denúncia anônimos disponibilizados por autoridades policiais;
- Proteger a identidade e localização de colaboradores de grupos de direitos humanos;
- Obter dados anônimos da população para fins de pesquisa médica e estudos sociais;
- Fornecer ajuda especializada a alcoólatras, viciados em drogas, vítimas de abuso físico ou sexual, entre outros;
- Conferir liberdade e imparcialidade à imprensa, com veículos de mídia independente de pressão ou influência externa;
- Orientar o público geral, especialmente em casos de portadores de doenças passíveis de discriminação;
- Prevenir fraudes eleitorais por coerção de eleitores e proteger a infra-estrutura de comunicação de sistemas de votação digital;
- Proteger testemunhas envolvidas em investigações críticas;
- Minimizar a coleta de informação por parte de governos e empresas para traçar perfis de cidadãos e clientes;

A comunidade ainda discute se mecanismos de anonimização na *Internet* não são uma manobra arriscada demais e se as conseqüências de sua utilização não seriam muito perigosas em relação às liberdades que podem trazer [Mar99]. Aumento no número de ações terroristas e volume de distribuição de material criminoso são especialmente apontadas como possíveis repercussões importantes. Ora, estas atividades continuarão a existir na rede independentemente da existência de um mecanismo de anonimização. Sempre haverá uma forma mais obscura de comunicação e que ofereça menor risco para a identidade dos criminosos e esta sempre será utilizada. Condenar o uso de anonimização, seja para proteger privacidade, seja para conferir liberdade, porque esta tecnologia pode ser utilizada para fins ilegais e imorais é equivalente a condenar o uso de criptografia pela mesma razão. Prevenir ou amenizar tais práticas de abuso em um cenário anonimizado depende das mesmas medidas utilizadas no mundo tangível: o abuso do sistema não cresce com o aumento do anonimato, mas com o decréscimo da capacidade de se estabelecer reputação. Se é possível estabelecer reputação de uma entidade, mesmo

que sua identidade real não seja conhecida, todos os demais componentes do sistema podem tomar atitudes defensivas e particulares, que protejam seus interesses da melhor forma possível.

1.4 Objetivo

Este trabalho abrange uma classe específica de ferramentas que visam prover *anonimato* em um contexto de rede aberta largamente adotada e padronizada, na qual as condições de anonimato devem ser construídas sobre a arquitetura e infra-estrutura existentes. Garantir anonimato em um cenário com estas características representa um desafiante tema de pesquisa.

O objetivo é projetar, analisar e implementar componentes que colaborem com a anonimato dos usuários na *Internet*: um componente de rede e um serviço de nomes. A função do componente de rede é assegurar que os padrões de comunicação dos usuários não revelem informação relevante a seu respeito. O serviço de nomes, por sua vez, é utilizado para minimizar ou até mesmo solucionar o importante problema da publicação de serviços anonimizados. São contribuições deste trabalho:

- O estudo da teoria de anonimato computacional, incluindo definições, métricas, técnicas e alternativas relacionadas ao fornecimento de anonimato em recursos computacionais;
- O estudo do paradigma de Criptografia de Chave Pública Sem Certificados;
- O projeto de uma nova rede de anonimização estruturada e moderna, que fornece boa qualidade de anonimato sob condições razoáveis;
- O projeto de uma política de roteamento eficiente e com boa distribuição de tráfego, validada empiricamente por meio de simulações;
- O projeto de um serviço de nomes que colabora com o ganho de usabilidade na rede de anonimização, pela substituição dos tradicionais pseudônimos criptográficos por nomes amigáveis e de fácil publicação;
- A implementação destes componentes em caráter de protótipo, em uma coleção de *software* dedicado para comunicação anônima.

Como substrato para este projeto, foi utilizada a rede de anonimização *Libfreedom* [AS05], projeto desenvolvido pelo autor como Trabalho de Graduação no curso de Bacharelado em Ciência da Computação da Universidade de Brasília. A agregação dos novos componentes à concepção original resultou em um novo projeto, nomeado *Kurupira*, que tem como principal vantagem a validação mais rigorosa do anonimato que fornece. Acredita-se que a nova rede de anonimização seja capaz de satisfazer as necessidades atuais de comunicação anônima.

Anonimização representa uma nova área de pesquisa em Sistemas Distribuídos e Segurança Computacional. Diversos problemas relacionados à publicação de serviços, organização, qualidade, eficiência e confiabilidade dos mecanismos de anonimização estão ainda sem solução ótima e definitiva. Buscar soluções ou alternativas para estas questões traz novas abordagens para a construção de mecanismos de anonimização e eleva a qualidade do *software* disponível para este fim.

1.5 Organização do trabalho

Este documento é organizado como se segue. O Capítulo 2 formaliza a definição de anonimato e o adversário considerado e apresenta os trabalhos relacionados de maior respaldo na área. O Capítulo 3 detalha o projeto de uma nova rede de anonimização, enfatizando a construção de uma política de roteamento escalável e eficiente para comunicação anônima. A nova rede combina ténicas utilizadas em diversas outras redes, em busca de um compromisso ótimo entre desempenho e anonimato e suporte à validação experimental de suas características. O Capítulo 4 apresenta a primitiva criptográfica utilizada para o projeto do serviço de nomes. O Capítulo 5 explora as características de projeto do serviço de nomes propriamente dito. O Capítulo 6 descreve os aspectos de implementação do *software* resultante deste trabalho e o Capítulo 7 apresenta conclusões e direções para trabalhos futuros.

Capítulo 2

Anonimato

Os serviços de segurança da informação mais comumente agregados a protocolos são o sigilo, a integridade e a autenticação. Entretanto, uma propriedade desejável em diversos cenários é o obscurecimento das identidades das partes comunicantes, tanto entre si como em relação a terceiros. Alcançar este objetivo normalmente requer estratégias peculiares, às vezes exigindo a combinação entre heurísticas e técnicas formais.

2.1 Definições

A palavra *anonimato* é derivada do grego $\alpha\nu\omega\nu\upsilon\mu\iota\alpha$, e significa a qualidade daquilo que não tem nome e mais originalmente, daquilo que não tem lei[1]. Coloquialmente, o termo se refere a uma pessoa cuja identidade ou qualquer informação relacionada não é conhecida. Quando se refere a uma entidade arbitrária (humano, objeto, computador), dentro de um conjunto bem-definido, o anonimato é a propriedade de não ser identificável dentro deste conjunto [PH06].

O anonimato não é absoluto, ou seja, a *qualidade de anonimato* que cada entidade possui pode variar. Freqüentemente, é diretamente proporcional ao tamanho do conjunto de entidades associado, chamado comumente de *conjunto de anonimato* [PH06]. O objetivo da entidade que deseja manter-se anônima é maximizar o tamanho deste conjunto, para agir dentro de uma multidão cada vez maior de entidades similares. Esta definição qualitativa implica ainda que, dado um evento particular, pode-se construir um conjunto de possíveis origens, mas a exatidão na determinação de uma única origem deve ser *difícil*. Claramente, a dificuldade cresce com o aumento da qualidade do anonimato, e vice-versa. Pode existir ainda a figura do *adversário*, que tem como objetivo exclusivo impedir que as entidades tornem-se anônimas.

Esta definição tradicional falha em capturar um aspecto fundamental: também não é desejável que seja fácil apontar a origem única de um evento com *grande probabilidade*. De nada

[1] Extraído de `http://dictionary.reference.com/`.

adianta contar com um conjunto de anonimato com 100 entidades se uma delas pode ser apontada como origem de um evento com 90% de probabilidade. Esta observação agrega um novo requisito ao conceito de anonimato: a *distribuição* dos eventos dentro do conjunto de anonimato deve ser o mais uniforme possível. A qualidade do anonimato passa a depender não só do conjunto de entidades, mas da distribuição dos eventos ocorridos entre os componentes do conjunto.

Esta definição mais acurada pode ser instanciada a partir do conceito de *quantidade de informação* na Teoria da Informação de Shannon [Sha48, Sha49], e baseia-se nos comportamentos contrastantes do adversário e sua vítima: o primeiro deseja obter informação que identifique unicamente o segundo, enquanto o segundo procura simultaneamente aumentar o trabalho do primeiro em obtê-la. Com esta nova definição, a qualidade do anonimato é diretamente proporcional à quantidade de informação que um adversário necessita angariar para indicar unicamente uma ligação entre origem e evento [DSCP02].

Recentemente, o anonimato também foi modelado como um problema criptográfico complexo [Kon05], incluindo ataques análogos aos normalmente confrontados por sistemas criptográficos, realizados por um adversário com poder computacional limitado polinomialmente.

2.2 Classificação de anonimato

A definição de anonimato pode ser contextualizada em um ambiente de comunicação: as entidades são usuários que trocam mensagens, os eventos são o envio e recebimento destas mensagens e a comunicação ocorre sob a observação de um adversário, que objetiva relacionar eventos de envio e recebimento aos seus *emissores* e *receptores*, respectivamente.

Os papéis dos usuários são diferenciados em *emissor* e *receptor*, quando a mensagem é o referencial. Se considerarmos um determinado *serviço* como referência, os usuários se especializam em *consumidor* e *provedor* do serviço. Entende-se por serviço uma entidade controlada por um usuário, que disponibiliza alguma funcionalidade para os demais usuários por meio de troca de mensagens. O consumidor é tipicamente o emissor da primeira mensagem que estabelece comunicação para disponibilização do serviço. Uma entidade pode atuar como emissor e receptor de mensagens distintas, bem como consumidor e provedor de serviços distintos simultaneamente. Dada esta taxonomia, costuma-se classificar o anonimato em três tipos [PH06]:

- *Anonimato de envio*: é difícil determinar o emissor de uma mensagem particular e, dado um usuário, é difícil atribuir uma mensagem particular como enviada por ele;

- *Anonimato de resposta*: é difícil determinar o receptor de uma mensagem particular e, dado um usuário, é difícil atribuir uma mensagem particular como recebida por ele;

- *Anonimato de par comunicante*: é difícil determinar um par de usuários comunicantes, ou seja, relacionar o emissor ao receptor de uma mensagem. Em comparação às anteriores, representa uma noção mais fraca de anonimato, já que é possível relacionar o emissor à mensagem enviada e o receptor à mensagem recebida, atuando-se apenas na associação entre estas duas mensagens.

Idealmente, o anonimato deve abranger os três aspectos acima definidos. A ausência de anonimato de envio provoca intimidação e conseqüente escassez de usuários (diminuição do *conjunto de anonimato*). A ausência de anonimato de resposta expõe os usuários que disponibilizam serviços anônimos. A ausência de desligamento entre emissor e receptor permite o rastreamento das mensagens trocadas no sistema, expondo potencialmente tanto os emissores quanto os receptores.

2.3 Métricas de anonimato

De posse de uma certa quantidade de informação a respeito de um evento de envio ou recebimento, obtida a partir de observação ou manipulação direta do ambiente, o adversário pode inferir a probabilidade de cada entidade ter participado do evento como emissor ou receptor.

Seja $\mathcal{A}$ um conjunto finito de usuários e seja $r \in \mathcal{R}$ o papel de um usuário $\mathcal{R} = \{\mathsf{emissor}, \mathsf{receptor}\}$, em relação a uma certa mensagem $m \in \mathcal{M}$. A *probabilidade de envolvimento* é a distribuição de probabilidade p dos usuários $a_i \in \mathcal{A}$ terem o papel r em relação a m, tomada pelo adversário.

O anonimato é descrito pela distribuição de probabilidade $p : \mathcal{A} \times \mathcal{R} \rightarrow [0,1]$. Dependendo das circunstâncias, a função p pode atribuir valores extremos no intervalo $[0,1]$. Se por exemplo, a_j for observado como o receptor direto da mensagem m, $p(a_j, \mathsf{receptor}) = 1$ e $\forall a_i \in \mathcal{A}, i \neq j$, $p(a_i, \mathsf{receptor}) = 0$. Claramente, tem-se que:

$$\sum_{a_i \in \mathcal{A}} p(a_i, r) = 1. \tag{2.1}$$

2.3.1 Entropia

A qualidade do anonimato pode ser quantificada por uma métrica de entropia [SD02]. A *entropia H* da distribuição de probabilidade p das probabilidades de envolvimento p_{a_i} associadas a cada usuário é dada por:

$$H = -\sum_{a_i \in \mathcal{A}} p_{a_i} \cdot \log_2 (p_{a_i}). \tag{2.2}$$

Esta métrica mede o grau de incerteza do adversário em identificar um usuário. Representa ainda a quantidade de informação que precisa ser obtida para que o usuário a_i seja identificado corretamente com papel r para a mensagem m. É fácil mostrar que se um usuário a_j possui probabilidade de envolvimento 1, a entropia é 0, ou seja, o adversário já detém informação suficiente para identificar a_j. São propriedades adicionais desta métrica:

- Para qualquer conjunto não-vazio de usuários $\mathcal{A}$, a entropia é tal que $0 \leq H \leq \log_2 |\mathcal{A}|$, e o valor $\log_2 |\mathcal{A}|$ é obtido quando p é uma distribuição uniforme;
- Se $H = 0$, o canal de comunicação não fornece anonimato;
- Se $H = \log_2 |\mathcal{A}|$, o canal de comunicação fornece *anonimato perfeito*;
- Se $H = h$, o canal de comunicação fornece anonimato equivalente a um canal de comunicação perfeito com 2^h usuários. A grandeza 2^H é chamada de *tamanho efetivo do conjunto de anonimato*.

Entropia mínima

A qualidade do anonimato também pode ser quantificada por uma métrica de *entropia mínima*, que representa o grau de exposição do usuário mais exposto. Idealmente, esta grandeza deve ser maximizada. Caso contrário, o adversário pode identificar um usuário e seu papel com grande probabilidade.

A entropia mínima H_{min} da distribuição de probabilidade p é dada por:

$$H_{min} = -\log_2 (\max_{a_i \in \mathcal{A}} p_{a_i}). \tag{2.3}$$

Entropia condicional

As grandezas de entropia e entropia mínima são parametrizadas a partir das observações de um adversário. Entretanto, existem situações nas quais nem sempre o adversário pode observar a transmissão de uma mensagem. A *entropia condicional* é a entropia média, calculada a partir da probabilidade p' de um adversário não observar uma mensagem e da entropia H' da distribuição de probabilidade das mensagens não-observadas pelo adversário.

A entropia condicional H_c da distribuição de probabilidade p, dadas as grandezas p' e H' é dada por:

$$H_c = (1 - p')H + p'H'. \tag{2.4}$$

A entropia condicional também pode ser formulada diretamente a partir da Teoria da Informação. Sejam A e Y duas variáveis aleatórias que modelam os usuários e as observações do

adversário, respectivamente. Percebe-se que as métricas de entropia apresentadas nas seções anteriores dizem respeito à distribuição da variável A dada uma observação particular y, ou seja, calculam $H(A|Y = y)$. A entropia condicional $H(A|Y)$ pode ser calculada como a média ponderada das entropias individuais [DSCP02]:

$$H(A|Y) = \sum_{y} Pr[Y = y]H(A|Y = y) = \mathcal{E}_y H(A|Y = y). \tag{2.5}$$

A entropia condicional é uma métrica de esperança, mais apropriada para a avaliação do anonimato em um ambiente persistente, em que um grande número de mensagens são trocadas durante um longo período de tempo. Entretanto, esta métrica não captura totalmente o potencial de exposição de um usuário. Mesmo que uma mensagem m seja observada com uma probabilidade $p' < 1$, é importante estimar o risco de exposição de um usuário se esta mensagem for observada.

A *entropia condicional mínima* calcula o grau potencial de exposição que um usuário pode ter que lidar. A entropia condicional mínima H_w da distribuição de probabilidade p é dada por:

$$H_w = \min_{y} H(A|Y = y). \tag{2.6}$$

2.3.2 Grau de anonimato

Cada um dos tipos de anonimato pode ainda ser avaliado de acordo com o *grau de anonimato* conferido, de acordo com as probabilidades de envolvimento tomadas pelo adversário [RR98]:

- *Privacidade Absoluta:* um usuário possui privacidade absoluta contra um adversário se o adversário não pode distinguir as situações em que um usuário participa de uma comunicação daquelas em que o usuário não participa;

- *Fora de suspeita:* um usuário está fora de suspeita se, do ponto de vista do adversário, mesmo existindo evidência da participação do usuário em um comunicação, a probabilidade do usuário ter participado não é significativamente maior do que a probabilidade de qualquer outro usuário ter participado;

- *Inocência provável:* um usuário é provavelmente inocente se, do ponto de vista do adversário, a probabilidade do usuário ter participado de uma comunicação não é maior do que a probabilidade do usuário não ter participado. Esta noção é mais fraca do que a anterior, no sentido de que o adversário tem informação suficiente para destacar um usuário entre os demais como provável participante na comunicação, mas a probabilidade do usuário ter participado da comunicação ainda é menor do que a probabilidade do usuário não ter participado;

- *Inocência possível:* um usuário é possivelmente inocente se, do ponto de vista do adversário, há uma probabilidade significativa do participante real em uma comunicação ser outro usuário;

- *Exposto:* um usuário está exposto se o adversário pode identificar o usuário como participante de uma comunicação com absoluta certeza;

- *Comprovadamente exposto:* o adversário não só pode identificar o usuário como participante de uma comunicação como pode provar este fato para terceiros.

Pode-se observar que, para alcançar grau de privacidade absoluta, o adversário sequer pode detectar que o usuário participou de alguma comunicação: o envio de uma mensagem, por exemplo, não pode resultar em qualquer efeito perceptível para o adversário. Ou seja, é necessário combinar uma primitiva de comunicação anônima perfeita a um mecanismo de anonimização perfeito da própria primitiva, impedindo efetivamente que o adversário detecte quando o usuário comunica-se com qualquer outro usuário. É possível implementar grau de privacidade absoluta combinando a primitiva de comunicação anônima com *esteganografia*[2].

Em grande parte das aplicações reais, o grau de inocência provável já é considerado suficiente. A diferença entre a probabilidade do usuário não ter participado e a probabilidade do usuário ter participado de uma comunicação impede que o adversário defenda posições conclusivas.

2.4 Técnicas de anonimização

Quaisquer que sejam as técnicas utilizadas, duas grandezas que parametrizam o anonimato são o *desligamento* e o *obscurecimento* [PH06]:

- *Desligamento:* desconexão entre dois ou mais objetos (endereços, mensagens, eventos de envio e recebimento) presentes no sistema, sob a perspectiva de um observador. A relação existente entre eles deve se manter inalterada, ou seja, não deve evoluir e nem regredir com o passar do tempo;

- *Obscurecimento:* controle da exposição dos objetos, ou indistinguibilidade de um evento dentre os demais, de um mesmo tipo. A transmissão de uma mensagem particular deve ser indistinguível da transmissão de qualquer outra mensagem, por exemplo.

Desligamento e obscurecimento são princípios análogos à *confusão* e *difusão* em Teoria da Informação. Técnicas sofisticadas de desligamento e obscurecimento colaboram para ganho de

[2]Esteganografia é o estudo e uso de técnicas para ocultar a existência de uma mensagem dentro de outra.

anonimato. Um *mecanismo de anonimização* é o conjunto de técnicas que um usuário utiliza para alcançar o máximo anonimato possível.

Se estes requisitos de anonimato são mapeados para um contexto de rede aberta, pode-se dizer que o mecanismo de anonimização deve utilizar técnicas para eliminar os padrões de comunicação (envio, recebimento, rota) que ligam *endereços de rede* a *pacotes* que trafegam na rede. O endereço de rede é aqui interpretado como identidade, visto que a partir dele é possível determinar com exatidão a identidade real de um usuário. O mecanismo de anonimização é então implementado como um *protocolo* de comunicação entre os usuários da rede. A união entre protocolo, usuários e *conexões* entre usuários define uma *rede de anonimização* dentro da rede aberta.

Tipicamente, não é necessário alcançar o anonimato do mecanismo de anonimização propriamente dito, ou seja, não há a necessidade de um usuário esconder que utiliza técnicas de anonimização ou impedir o conhecimento das técnicas de anonimização utilizadas. Este conhecimento por parte do adversário deve também ser previsto durante o projeto do mecanismo, visto que o adversário normalmente pode simular usuários legítimos do sistema e utilizar o mesmo mecanismo de anonimização que os demais usuários utilizam. Esta observação é similar à recomendação de que algoritmos criptográficos devem ser publicados em padrões abertos, pois assume-se que o adversário sempre pode obter uma descrição correta do algoritmo, mesmo que recorra à engenharia reversa, por exemplo.

2.4.1 Mecanismos convencionais de anonimização

Os serviços convencionais de anonimização confiam na centralização do mecanismo. Uma estratégia comum é prover acesso por meio de *proxy*, para que os endereços reais de origem dos pacotes sejam substituídos pelo endereço do *proxy* no momento em que o atravessam. Um serviço que ilustra esta técnica é o *Anonymizer*[3].

Existem inúmeras desvantagens nesta abordagem. Em primeiro lugar, este método só provê anonimato aos emissores. A inserção de um dispositivo totalmente centralizado traz um ponto único de falha, expondo-o a ataques remotos e repercussões legais. O anonimato dos clientes pode ser quebrado se forem distribuídos pontos de escuta nas interfaces de entrada e de saída do serviço. Após realizar capturas massivas nos dois pontos, é trivial determinar os endereços de origem: basta projetar um isomorfismo entre os conteúdos dos pacotes que entram e saem do *proxy*. Dada esta falha estrutural, é sensato dizer que os clientes de um mecanismo centralizado de anonimização apenas aglomeram tráfego de cobertura para os operadores do serviço: somente os operadores do serviço centralizado contam com anonimização considerável do tráfego que produzem. É importante examinar a facilidade com a qual o próprio adversário pode instalar um serviço centralizado falso de anonimização ou tomar o controle de algum já existente.

[3] Sediado em `http://www.anonymizer.com`.

2.4.2 Técnicas modernas de anonimização

São várias as técnicas de anonimização que promovem desligamento e obscurecimento.

Descentralização

A descentralização do mecanismo de anonimização permite a construção de uma rede de nós interligados e controlados por usuários que implementam procedimentos distribuídos e colaborativos de anonimização de tráfego, o que traz um cenário significativamente mais robusto do que o original. Elimina-se o ponto único de falha e torna-se pouco viável a suspensão do serviço por força legal. Assim, arquitetura e protocolo descentralizados são requisitos de projeto essenciais para a construção de um mecanismo de anonimização útil e resistente.

Indireção [BG03]

A forma mais simples de se desfigurar a ligação entre um pacote e seu endereço de origem é permitir que cada nó atue como *roteador*. Em cada ponto de comutação desta rede, o pacote recebe o endereço de origem do roteador que o retransmite, e, dadas as informações locais de roteamento, uma nova direção é calculada.

É necessário o estabelecimento de um *sistema de identificação* externo ao endereçamento na rede aberta para que os pacotes possam ser encaminhados corretamente aos seus respectivos destinos. A *política de roteamento* utilizada deve atuar sobre o esquema de identificação adicional. O anonimato do roteador é alcançado quando este envia os seus próprios pacotes em meio aos pacotes roteados: o tráfego roteado provê cobertura para o tráfego produzido e consumido no roteador, dificultando, para um adversário, diferenciar as mensagens que se originam ou são consumidas daquelas que apenas o atravessam. É requisito adicional que a ligação entre a nova identidade e o endereço na rede aberta seja mantida em segredo. Quando a identidade adicional é duradoura e utilizada para estabelecer reputação, passa a ser chamada de *pseudônimo*.

Igualdade entre mensagens

Se os pacotes trocados são visíveis para um adversário em seu formato original, é possível separar o tráfego de um roteador por tamanho e conteúdo: os pacotes de saída que não tiverem tamanho ou conteúdo correspondente a nenhum dos pacotes que entraram em uma determinada janela de tempo devem ter sido ali originados. Analogamente, os pacotes de entrada que não possuírem pacotes correspondentes de saída devem ter sido ali consumidos. É necessário, portanto, que os pacotes roteados e produzidos pelo roteador tenham tamanho e conteúdo com características similares. A análise de tráfego utilizada para se tentar estabelecer relações entre os pacotes enviados e recebidos, com a finalidade de separar o tráfego roteado do tráfego originado e consumido localmente, caracteriza um ataque de *correlação temporal*.

Injeção de ruído [DP04]

Quando existem alguns roteadores que, por algum motivo, roteiam um volume de tráfego muito baixo e produzem ou consomem um volume de tráfego muito alto, esta assimetria torna o roteador vulnerável a ataques de correlação temporal. O equilíbrio entre os fluxos de entrada e saída do roteador pode ser obtido pela adição de tráfego na direção desfavorecida, para se dificultar os ataques de correlação temporal, desde que o *tráfego de ruído* seja indistinguível do tráfego legítimo. A utilização desta técnica gera um compromisso entre qualidade de anonimato e percentual útil de banda utilizada.

Atraso no roteamento [KEB98]

É possível atrasar a retransmissão de cada pacote recebido. Uma parcela do desempenho de roteamento é comprometida, mas obtém-se um aspecto difuso nos pacotes de entrada e saída, o que dificulta ainda mais ataques de correlação temporal. A conjunção entre atraso de envio e injeção de ruído pode ser demasiadamente danosa ao desempenho, sendo necessário um exame minucioso da necessidade de se implementar os dois recursos, e dos parâmetros de configuração que os definem (tempo máximo de atraso, percentual de tráfego que representa ruído).

Sigilo

A atuação dos roteadores deve ser confidencial. Se é possível para um adversário examinar absolutamente todas as mensagens enviadas e recebidas por um roteador, mesmo que várias das medidas anteriormente discutidas sejam aplicadas, é trivial empregar análise de tráfego para quebra de anonimato: é possível separar por conteúdo os volumes de tráfego consumido, produzido, roteado e que representa mero ruído. Para que as técnicas discutidas anteriormente funcionem a contento, é necessário que uma camada de cifração seja adicionada às comunicações entre os roteadores e que cada nó estabeleça ao menos uma conexão livre de monitoramento (ou seja, uma conexão cifrada com um nó que não é controlado por um adversário). Uma camada adicional de sigilo também pode ser inserida entre as partes comunicantes propriamente ditas (emissor e receptor da mensagem), trazendo a vantagem de que os nós intermediários desconhecem o conteúdo dos pacotes sendo roteados, impedindo a filtragem dos pacotes que assemelham-se a um padrão característico. Caso contrário, um roteador poderia empregar o descarte de pacotes tanto por convicção própria como por imposição da lei.

Descentralização, indireção e atraso no envio são técnicas que implementam o princípio de desligamento, enquanto que igualdade entre mensagens, injeção de ruído e sigilo colaboram com o obscurecimento do comportamento dos rotadores.

2.5 Adversário

Levando-se em consideração a preservação da identidade de um nó, é importante definir com precisão o poder com o qual o adversário conta para violar esta proteção. O adversário é aquele que tenta derrotar o serviço de segurança da informação sendo utilizado, e, para o caso particular, é aquele que tenta quebrar o anonimato e estabelecer ligações entre pacotes, rotas e endereços de origem ou destino.

A qualidade do anonimato depende das probabilidades de envolvimento dos usuários em um conjunto de eventos de comunicação. Estas probabilidades são atribuídas pelo adversário, a partir da coleta de informação do ambiente. Esta coleta de informação pode ocorrer de diversas formas e pode contar com a distorção do ambiente por parte do adversário. Assim, faz sentido definir a qualidade do anonimato em termos do *ataque* realizado.

2.5.1 Características

O adversário e a classe de ataques que realiza podem ser classificados por diferentes critérios [DSCP02]:

- *Localização:* um adversário interno controla um ou mais componentes do sistema. Ou seja, o adversário pode prevenir um nó de enviar mensagens ou pode acessar informação exclusiva ao nó. Um adversário externo pode apenas comprometer canais de comunicação (monitorar ou manipular mensagens);

- *Atuação:* um adversário passivo apenas monitora ou captura informação interna. Um adversário ativo é capaz de adicionar, remover e modificar mensagens ou alterar informação exclusiva ao nó;

- *Amplitude:* um adversário global tem acesso a todo o sistema de comunicação, enquanto um adversário local pode apenas controlar parte dos recursos envolvidos.

Diferentes combinações dessas propriedades são possíveis: por exemplo, um adversário global, passivo e externo é capaz de monitorar todos os canais de comunicação, enquanto um adversário local, ativo e interno pode apenas controlar um subconjunto dos nós participantes.

Assume-se que a única forma possível para um adversário quebrar o anonimato de um nó é a análise ou participação nas interações do protocolo que conecta os participantes da rede. É importante notar também que é dever do adversário provar que um fragmento de comunicação foi iniciado em um nó específico, ou seja, o *ônus da prova* é de responsabilidade total do adversário. Se o cenário jurídico fosse distinto e o sistema legal impusesse ao acusado a tarefa de provar que a comunicação não foi originada por ele, anonimato poderia ser considerado essencialmente ilegal [BG03].

Com exceção da condição de que o adversário só pode conjecturar relações entre uma ação particular e sua origem a partir de evidências colhidas do protocolo, o modelo de adversário permite que este possa fazer praticamente qualquer coisa, exceto quebrar primitivas criptográficas. O adversário pode ter o poder de observar todo o tráfego entre todos os nós a todo instante, mas não pode decifrar comunicações entre dois nós que não são controlados por ele. O adversário pode também controlar um número arbitrário de nós na rede, e estes nós podem tanto violar o protocolo quanto se comportar como um participante qualquer. Entretanto, deve-se impor um limite superior à fração de nós maliciosos na rede, de forma que qualquer nó conectado consiga estabelecer uma conexão segura com pelo menos um nó íntegro.

2.5.2 Ataques

Assim como os mecanismos de anonimização podem combinar um conjunto de técnicas básicas para alcançar o máximo anonimato possível, o adversário pode combinar uma série de ataques distintos para maximizar sua eficiência. Os ataques em redes de anonimização podem ser divididos em diversas classes [CC05b, Ray00], que são detalhadas nos tópicos subseqüentes.

Ataques de rastreamento

Em um ataque de rastreamento, o adversário manipula as mensagens trocadas para facilitar seu rastreamento. A manipulação deve alterar características particulares das mensagens que permitam a fácil identificação do tráfego manipulado em outros pontos da rede controlados pelo adversário. Um exemplo de ataque de rastreamento é a manipulação dos contadores de tempo de vida (*time-to-live* - TTL) das mensagens. Contadores de tempo de vida são normalmente utilizados para garantir o descarte de mensagens antigas. Um exemplo simples de manipulação de contadores de tempo de vida para obtenção de informação é o envio consecutivo de diversas mensagens com contadores de tempo de vida decrescentes. Pode-se inferir o número de nós intermediários entre emissor e receptor pela observação das respostas: a mensagem original com menor contador de tempo de vida que obtiver resposta revela o comprimento da rota entre o adversário e o nó atacado. A utilização de contadores de tempo de vida não-determinísticos dificulta a análise, mas ainda assim pode fornecer informação estatística relevante.

Ataques de identidade múltipla

O adversário provoca o estabelecimento de conexões entre nós que controla e o nó atacado, utilizando diferentes identidades simultaneamente. O adversário pode então monitorar uma parcela do tráfego que atravessa o nó, proporcional à razão de conexões monitoradas. O ataque torna-se definitivo quando o nó conecta-se apenas a nós maliciosos, ou seja, é completamente confinado pelo adversário.

Ataques clássicos são os *ataques de maioria* (*Sybil attacks*) [Dou02] e de *eliminação* [GT96]. Em um ataque de maioria, o adversário tenta controlar o maior número possível de recursos (nós, notas, canais de comunicação), objetivando confinar o nó atacado e monitorar todo o tráfego que o atravessa. Um ataque de eliminação é um ataque ativo em que o adversário envia mensagens consecutivas ao nó atacado com o objetivo de diminuir a vazão de tráfego legítimo que o atravessa. O tráfego legítimo que restar pode então ser marcado e rastreado ao longo da rede, nos pontos que também são controlados pelo adversário. Este ataque exige o controle por parte do adversário de uma porção da vizinhança do nó atacado, já que os vizinhos podem enviar mensagens para o nó atacado com maior rapidez e podem marcar o tráfego legítimo assim que ele sai do nó.

Ataques de identidade múltipla estão entre os ataques mais perigosos em redes de anonimização e várias medidas podem ser utilizadas para contorná-los [Dou02]:

- A rede de anonimização deve conectar o maior número possível de entidades, para aumentar o custo de se controlar uma porção significativa da rede. Isto depende exclusivamente da fidelidade dos usuários e da popularidade da rede;

- Um problema criptográfico deve ser solucionado para se assumir uma identidade na rede. Este mesmo problema deve ter um caráter aleatório e solução verificável pelos outros nós. Um exemplo de problema criptográfico com estas características é a geração de um par de chaves com propriedades peculiares, por exemplo, com o *hash* da chave pública terminando em *n bits* iguais a zero [CDG+02]. A solução do problema criptográfico diminui a razão com a qual o adversário controla novos recursos na rede;

- As conexões na rede podem obedecer a uma relação matemática bem-definida e as identidades dos nós podem ser geradas a partir de uma função com saída aleatória. Desta forma, as identidades distribuem-se de maneira mais uniforme no espaço de identidades, prevenindo ataques de eliminação ou o confinamento de um nó. A regularidade na organização da rede dificulta o controle da vizinhança de um nó [CDG+02, Bor05].

- Os usuários podem assumir identidades temporárias. Desta forma, o adversário se vê forçado a resolver o mesmo problema criptográfico seguidas vezes sempre que o nó atacado altera sua identidade.

Outra forma de se evitar ataques de identidade múltipla é permitir que cada nó estabeleça conexões para todos os outros nós da rede [RR98]. Obviamente, esta abordagem não tem boa escalabilidade com o crescimento da rede.

Ataques estatísticos

O adversário coleta informação estatística sobre o sistema. Se o adversário tem o poder de forçar duas partes comunicantes a trocar pacotes com freqüência, a observação dos efeitos colaterais pode revelar informação adicional. A observação de padrões de comunicação simples, como os nós que enviam e recebem mensagens ao longo do tempo, já pode revelar informação relevante. Entidades envolvidas em um fluxo de comunicação duradouro também não costumam enviar e receber mensagens simultaneamente, as etapas de envio e recepção são alternadas.

Ataques de predecessor [WALS02, WALS04] destacam-se como ataques estatísticos importantes. Um ataque de predecessor consiste em contar o número de vezes em que cada nó roteia uma mensagem que faz parte de um fluxo de comunicação para qualquer um dos nós controlados pelo adversário. Dentro de cada fluxo de comunicação, o verdadeiro emissor das mensagens deve aparecer com maior freqüência.

Ataques de correlação

O adversário utiliza a informação coletada do protocolo para isolar padrões de comunicação [Ray00]. Ataques de correlação são naturalmente combinados a ataques estatísticos e especializam-se em *ataques de correlação temporal* e *ataques de correlação por conteúdo*. Injeção de ruído e atrasos na retransmissão de pacotes dificultam a montagem de ataques de correlação temporal, enquanto cifração dificulta a montagem de ataques de correlação por conteúdo. A simples medição do tempo de resposta de uma mensagem pode fornecer informações a respeito da localização ou comprimento da rota utilizada. A observação dos efeitos colaterais de um atraso na retransmissão de uma mensagem particular também pode fornecer informação relevante [Dan04b, MD05].

Ataques na entrada e saída de nós

O adversário monitora eventos de entrada e saída de nós para determinar pares comunicantes. *Ataques de intersecção* [BPS00] são representantes desta classe. O adversário realiza um ataque de intersecção quando divide os nós por disponibilidade ao longo do tempo. A associação entre o subconjunto de nós conectados e o subconjunto de mensagens trocadas durante o mesmo período de tempo pode apontar os pares de nós comunicantes, especialmente quando os nós costumam comunicar-se com um pequeno número de nós da rede. A alteração freqüente das identidades dos usuários dificulta ataques desta natureza.

Ataques de negação de serviço

O adversário tenta colocar um subconjunto de nós ou o sistema completo em situação crítica. Uma rede de anonimização é inútil se não pode ser utilizada pelos usuários. *Ataques de sobrecarga*, onde o adversário satura toda a capacidade de transmissão da rede com pacotes fabricados, são exemplos clássicos de ataques de negação de serviço. A eliminação forçada de um nó e observação dos efeitos colaterais, especialmente as rotas descritas pelas mensagens subseqüentes, também pode fornecer informação reveladora.

Ataques de negação de serviço podem ser minimizados por projeto e implementação cuidadosos dos protocolos, políticas de roteamento que privilegiem balanceamento de carga e regularidade na organização da rede. *Ataques jurídicos* realizados por adversários que contam com influência política considerável também podem impedir a utilização de uma rede de anonimização.

Ataques de personificação

O adversário personifica o consumidor ou fornecedor de um serviço polêmico com o objetivo de capturar informação a respeito do real fornecedor ou de possíveis clientes.

2.6 Trabalhos relacionados

Os esforços iniciais na área de anonimização datam de 1981, com a criação das *redes de mistura* por Chaum [Cha81]. As primeiras aplicações de anonimização concentraram-se em sistemas de credenciais e de micro-pagamentos [Cha85] e em serviços de publicação de documentos [And96]. Entre eles, a concepção do serviço *Eternity* [And96], um dos primeiros trabalhos a reconhecer a real importância do projeto e análise de mecanismos de anonimização. Diversas abordagens radicalmente distintas foram propostas posteriormente como soluções para o problema da anonimização. Tanto técnicas criptográficas formais [Cha88] quanto probabilísticas [GRS96] foram apresentadas, dando origem a várias redes de anonimização. Algumas destas redes apresentam aplicação reduzida, sendo dedicadas especificamente às funcionalidades de distribuição de documentos ou compartilhamento de arquivos; outras funcionam como camada genérica de comunicação, permitindo anonimização de tráfego de qualquer natureza. As implementações funcionais com melhor aceitação dos usuários são: *Freenet* [CSWH00], *GNUnet* [BG03] e *Tor* [DMS04].

Pode-se dividir as redes de anonimização existentes em subclasses distintas que implementam técnicas similares. Uma comparação entre as famílias de redes de anonimização e o grau aproximado de anonimato que fornecem, considerando adversários com diversas características, é apresentada na Tabela 2.1. Representantes modernos de cada uma das famílias também são apresentados, acompanhados dos seus respectivos graus de anonimato aproximados. Os graus

de anonimato inferidos são os que constam na formulação original das redes, desconsiderando ataques elaborados. Ou seja, o grau de anonimato listado é compatível com as técnicas que a rede utiliza para garantir um certo grau de anonimato contra um adversário fixo, sempre que as medidas para derrotá-las são não-triviais. O grau de anonimato apresentado também não prevê a combinação de adversários com características distintas e não considera o sucesso do melhor ataque já publicado. Ainda na Tabela 2.1, são considerados como adversários o receptor de uma mensagem, o emissor de uma mensagem, um nó intermediário interno que intercepta uma mensagem e um observador externo global. O grau de anonimato é medido para anonimatos de envio, de resposta e do par comunicante, na presença de cada um dos adversários. Considerando o atacante e tipo do anonimato, o símbolo "★" indica que a rede alcança grau *fora de suspeita*, o símbolo "⋆" indica que a rede alcança grau de *inocência provável* e o símbolo "⊥" indica que a rede não toma medidas para fornecer anonimato naquele cenário. Adicionalmente, graus alternativos de anonimato e as respectivas condições para alcançá-los são apontados entre parêntesis.

	ANONIMATO							
	ENVIO			RESPOSTA			PAR	
	ADVERSÁRIO			ADVERSÁRIO			ADVERSÁRIO	
REDE	**Nó**	**Receptor**	**Global**	**Nó**	**Emissor**	**Global**	**Nó**	**Global**
MIXes	⊥	★	⊥	⊥	⊥	⊥	★	★
GNUnet	★	★	★	★	★	★	★	★
Onion Routing	⊥(★†)	★	⊥(★†)	⊥	⊥	⊥	★	★
Tor	⊥(★†)	★	⊥(★†)	⊥(★‡)	⊥(★‡)	⊥(★‡)	★	★
Crowds	⋆	★	⊥	⊥	⊥	⊥	⋆	⊥
AP3	⋆	★	⊥	⋆	★	⊥	⋆	⊥
DC-nets	★	★	★	★	★	★	★	★
Herbivore	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆
Multicast	⊥	⊥	⊥	★	★	★	★	★
$\mathcal{P}^5$	⋆	★	★	⋆	⋆	★	★	★
Ants	⋆	⋆	⊥	⋆	⋆	⊥	⋆	⊥
Mantis	⋆	⋆(★≀)	⊥	⋆	⋆	⊥	⋆	⊥
Freenet	⋆	⋆	⊥	⊥	⊥	⊥	⋆	⊥

(⊥) A rede não fornece anonimato; (⋆) O usuário tem grau de inocência provável;
(★) O usuário tem grau de fora de suspeita; (†) O usuário executa um roteador local;
(‡) O serviço encontra-se sediado na rede anônima e é contactado por meio de *pontos de encontro*;
(≀) As mensagens são de transferência de arquivos.

Tabela 2.1: Graus de anonimato aproximados fornecidos pelas redes existentes.

2.6.1 Redes de mistura

As redes de mistura [Cha81] constituem as idéias precursoras das redes de anonimização atuais e assumem um conjunto de usuários de um sistema criptográfico assimétrico composto por uma função de cifração E e uma função de cifração D, cada um dotado de um par de chaves pública e privada (e,d). O sistema ainda possui um dispositivo chamado *MIX* que processa cada mensagem cifrada trocada pelos usuários.

Para o envio de uma mensagem m a um participante com endereço A, um usuário cifra a mensagem com a chave pública e_A, acrescenta o endereço de destino A, e cifra o resultado com a chave pública e_M do *MIX*. O *MIX* recebe então o resultado de $E_{e_M}(E_{e_A}(m),A)$, decifra a mensagem recebida e envia $E_{e_A}(m)$ para o endereço A. O propósito do *MIX* é esconder a correspondência entre os dados que recebe de entrada e os que emite como saída: a ordem de chegada é escondida pela emissão de resultados em unidades de tamanho fixo, em lotes ordenados lexicograficamente. A utilização de uma cascata ou série de dispositivos *MIX* oferece a vantagem de que qualquer *MIX* envolvido é capaz de prover sigilo na correspondência entre as entradas e as saídas ao longo da cascata. Um item é preparado para ser transmitido por uma cascata de múltiplos *MIX* agregando-se uma camada de cifração para cada *MIX* que receberá a mensagem. O processamento de uma mensagem por uma seqüência de dispositivos *MIX* emprega o desligamento gradativo entre emissor e mensagem.

Diversas modificações da formulação original foram propostas [DP04], utilizadas principalmente para anonimização de correio eletrônico [GT96, DSD04, DDM03]. O projeto *Free Haven* [DFM00] utiliza redes de mistura para construir um serviço de publicação e distribuição de documentos anonimizado. O projeto *Tarzan* [FM02] emprega redes de mistura para fornecer primitivas de comunicação anônima transparentes para aplicação. A rede *MorphMix* [RP02] constrói um mecanismo genérico para comunicação anônima na *Internet* utilizando redes de mistura. O diferencial de *MorphMix* é a possibilidade de detecção de adversários por nós legítimos e posterior aplicação de medidas restritivas. Recentemente, foram apresentadas técnicas para se contornar a detecção de adversários [TB06].

GNUnet [BG03] é uma camada de comunicação anônima para troca livre de informação. Trata-se de uma rede baseada no conceito de redes de mistura, onde a única imposição a um nó conectado é colaborar para que os recursos que consome não afetem os demais. É objetivo suportar qualquer operação tipicamente realizada em uma rede, apesar da versão atual apenas fornecer compartilhamento anonimizado de arquivos. A rede é construída ao redor da idéia de que usuários podem ser anônimos se conseguirem esconder suas ações dentro do tráfego produzido por outros usuários. Existe, portanto, a preocupação de tornar o tráfego produzido no nó indistinguível do tráfego por ele roteado. A rede é constituída por nós interligados por conexões cifradas. As mensagens transmitidas trafegam por estas conexões e cifração adicional é empregada nas pontas, visando garantir que os nós intermediários não possam descartar mensagens. As conexões entre os nós são estabelecidas utilizando uma abstração que esconde os detalhes

das camadas superiores. Logo, o protocolo da *GNUnet* pode ser transportado por TCP, UDP, SMTP e HTTP, entre outros, o que dificulta sensivelmente sua filtragem por *firewalls* draconianos. Para controle de abusos, cada nó monitora as atividades de todos os nós que mantêm contato e um modelo econômico [Gro03] garante que os nós com comportamento abusivo não sejam servidos com a mesma prioridade que os demais. As principais desvantagens da abordagem utilizada pela *GNUnet* são a interface de programação incomum, que dificulta a migração de aplicações existentes para um contexto anônimo, e o enfoque em compartilhamento de arquivos, que pode comprometer o desempenho da rede para aplicações diversas. A *GNUnet* fornece uma otimização que permite aos nós intermediários diminuir o comprimento das rotas utilizadas. Este mecanismo, criado para controlar o compromisso entre desempenho e qualidade de anonimato, pode ser utilizado para a montagem de ataques de intersecção especiais na rede [Küg03].

2.6.2 *DC-nets*

As *DC-nets* baseiam-se no *Problema dos Criptólogos Glutões*, proposto originalmente por Chaum [Cha88]. Esta construção fornece anonimato teórico *perfeito*. Para ilustrar o funcionamento destas redes, sejam dois participantes A e B, que desejam publicar mensagens m_A e m_B, respectivamente. Os dois participantes desejam fazê-lo de forma anônima, para que um observador externo não consiga diferenciar qual participante publicou qual mensagem. Os participantes A e B ainda compartilham os segredos k_{AB_0} e k_{AB_1} e um *bit* aleatório b. As mensagens e os segredos têm comprimento k, em *bits*. Os participantes publicam os seguintes pares de mensagens [GJ04]:

b	A	B
0	$M_{A,0} = k_{AB_0} \oplus m_A, M_{A,1} = k_{AB_1}$	$M_{B,0} = k_{AB_0}, M_{B,1} = k_{AB_1} \oplus m_B$
1	$M_{A,0} = k_{AB_0}, M_{A,1} = k_{AB_1} \oplus m_A$	$M_{B,0} = k_{AB_0} \oplus m_B, M_{B,1} = k_{AB_1}$

Tabela 2.2: Rodada de comunicação em uma *DC-net*.

Um observador externo pode recuperar as mensagens m_A e m_B a partir do cálculo de $M_{A,0} \oplus M_{B,0}$ e $M_{A,1} \oplus M_{B,1}$. A origem das mensagens, entretanto, permanece incondicionalmente secreta. Sem conhecer previamente as chaves compartilhadas entre A e B, o observador não consegue determinar que participante publicou cada mensagem. Em contraste direto com as redes de mistura, uma propriedade importante deste protocolo é a *não-interatividade*: uma vez estabelecidos os segredos, os participantes A e B não precisam trocar mensagens diretamente.

Este protocolo básico pode ser estendido para múltiplos participantes $P_1, P_2, \ldots, P_n$. Cada par de participantes (P_i, P_j) compartilha um conjunto de segredos k_{ij_w} para $i, j, w \in \{1, 2, \ldots, n\}$,

onde $k_{ij_w} = k_{ji_w}$. Cada participante P_i calcula:

$$W_i = \{W_{i_1} = \oplus_{j=1}^{n} k_{ij_1}, \ldots, W_{i_n} = \oplus_{j=1}^{n} k_{ij_n}\}.$$

Para publicar mensagens neste esquema, cada participante escolhe uma posição aleatória b_i e calcula o XOR de sua mensagem m_i com a posição b_i de W_i. Isto produz um novo vetor $V_i = \{V_{i_1}, V_{i_2}, \ldots, V_{i_n}\}$ que difere de W_i apenas na posição b_i. Se todos os participantes escolherem posições b_i diferentes, o vetor $V = \oplus_{i=1}^{n} V_i$ resulta no conjunto de mensagens m_i publicadas. O estabelecimento prévio dos segredos compartilhados pode ser realizado através de iterações sucessivas de um protocolo de acordo de chaves *Diffie-Hellman* [DH76]. Se os segredos compartilhados e posições b_i selecionadas forem secretos, a anonimização da origem das mensagens é *perfeita* [Cha88].

Duas das limitações desse protocolo são a necessidade de um nó estabelecer conexões com todos os outros e a necessidade de um mecanismo para tratamento de colisões (quando dois participantes selecionam posições $b_i = b_j$ idênticas). As colisões podem ser tratadas por meio de procedimentos de reserva antecipada, apesar do próprio sistema de reservas também exigir tratamento de colisões [GJ04]. Aplicações reais que utilizem esse protocolo devem tolerar perda de mensagens por ineficiência no tratamento de colisões.

A maior desvantagem do esquema é que um único participante desonesto pode impedir a comunicação de todos os outros participantes: nenhuma das mensagens originais é publicada caso um dos participantes publique um conjunto de valores corrompidos. Pela própria natureza do esquema, não é possível detectar a origem da fraude. A solução deste problema reside na modificação do esquema original para permitir a detecção do participante desonesto. Uma nova variante das *DC-nets* foi proposta para resolver este problema utilizando emparelhamentos bilineares, obtendo um esquema de detecção probabilístico eficiente em comunicação [GJ04].

Tentativas de implementação de *DC-nets* agrupam os nós em pequenos cliques[4]. Os cliques comunicam-se entre si como uma *DC-net* e cada clique comunica-se internamente como uma *DC-net*. Com isso, o problema da escalabilidade é amenizado, pois não existe mais a necessidade de todos os nós possuírem conexão entre si. A rede *Herbivore* [GRPS03] demonstra que *DC-nets* são viáveis para implementação real se todas as limitações forem tratadas cuidadosamente durante o projeto do protocolo. Entretanto, o tratamento de colisões e demais mecanismos de suporte ainda elevam a complexidade de comunicação, tornando as alternativas que fornecem anonimato probabilístico mais adequadas para utilização na *Internet*.

[4]Um clique em um grafo G é um subgrafo de G que é completo.

2.6.3 *Onion Routing*

O protocolo *Onion Routing* [GRS96] permite comunicação anônima em rede aberta e tem funcionamento similar a uma cascata de dispositivos *MIX*. Ao enviar uma mensagem, o emissor seleciona um conjunto de roteadores e cria uma estrutura em camadas que codifica a rota. Cada camada é cifrada com a chave pública do roteador escolhido. Ao receber a mensagem codificada, cada roteador decifra a camada mais externa para obter o endereço do roteador seguinte e encaminha a mensagem. Ao atingir o último roteador escolhido, a mensagem é entregue ao destino. Como cada camada é cifrada com uma chave diferente, os roteadores intermediários não obtém informação a respeito do caminho além do endereço do roteador seguinte. Existem duas configurações possíveis para um usuário final: o primeiro roteador pode ser executado localmente ou selecionado a partir de um conjunto de roteadores públicos. A configuração local requer mais recursos, mas fornece melhor qualidade de anonimato. Implementações iniciais deste conceito foram os sistemas de anonimização *Freedom* [BSG00] e *Cebolla* [Bro02].

A rede *Tor* [DMS04] é um sistema genérico de comunicação anônima de baixa latência para a *Internet* que implementa uma nova versão do protocolo *Onion Routing*. Esta nova versão fornece sigilo futuro e corrige a vulnerabilidade a ataques de repetição presente na versão anterior. *Tor* pode ser usada como *proxy* para se acessar recursos na *Internet* convencional de forma anonimizada ou para sediar serviços dentro da rede anônima, através de *pontos de encontro* (*rendezvous servers*) [OS06b]. A técnica fundamental utilizada é a construção de túneis, que transportam as mensagens de uma aplicação de origem a uma aplicação de destino, empregando cifração em cada conexão do túnel. O comprimento e a composição do túnel são definidos durante o procedimento de construção, e a renovação freqüente de túneis é recomendada para obscurecer o tráfego que atravessa a rede. Por possuir uma característica de baixa latência, a rede é especialmente vulnerável a ataques de correlação temporal. Um adversário observando padrões de tráfego no emissor e no receptor é capaz de confirmar a correspondência com grande probabilidade [MD05, OS06a].

A rede *I2P*[5] é outra implementação de uma camada de comunicação variante do protocolo de *Onion Routing*. Nesta variante, chamada *Garlic Routing*, emissor e receptor constróem túneis para envio e recebimento de mensagens. É também possível controlar o compromisso entre latência e anonimato, a partir da definição do tamanho dos túneis e de outros parâmetros do protocolo de roteamento. A aplicação *APFS*, baseada em *Onion Routing*, é dedicada para o compartilhamento anonimizado de arquivos [SLS01]. A aplicação *SSMP* [HLX+05] combina *Onion Routing* com o esquema de compartilhamento de segredo de Shamir [Sha79] para fornecer compartilhamento de arquivos mutuamente anônimo.

[5] `http://www.i2p.net`

2.6.4 *Crowds*

Crowds [RR98] é um protocolo anonimizado que permite a um cliente comunicar-se com um serviço sem revelar sua identidade. O protocolo envolve um grupo de usuários e roteia cada mensagem dentro deste grupo até que um dos membros do grupo decida encaminhar a mensagem para o serviço propriamente dito. Esta decisão é condicionada a uma probabilidade de encaminhamento fixa p_f. Isto garante que o destino e os nós do sistema não possam determinar a origem da mensagem. O sistema exige que cada nó conecte-se a todos os outros, o que prejudica sua escalabilidade em aplicações reais. A qualidade de anonimato fornecida pelo protocolo degrada com o crescimento do sistema e com comunicação prolongada [WALS02, Shm02].

O protocolo *AP3* [MOP+04] é uma adaptação do protocolo *Crowds* para redes com organização regular e acrescenta suporte ao estabelecimento de pseudônimos, agregando anonimato de resposta ao esquema original. Diversas falhas de projeto na formulação original da rede *AP3* foram publicadas, com suas devidas correções [LM].

2.6.5 *Multicast*

A transmissão de uma mensagem em modo *multicast* pode ser usada para fornecer anonimato de resposta se o número de nós que recebem mensagens é suficiente para esconder o receptor real da mensagem. Comunicação em modo *multicast* é de fácil implementação em redes de anonimização, já que cada nó pode encaminhar as mensagens recebidas para vários dos vizinhos que mantém contato. São necessários mecanismos de descarte para eliminar mensagens recebidas múltiplas vezes e contadores de tempo de vida para limitar a permanência das mensagens na rede.

O protocolo $\mathcal{P}^5$ [SBS02] agrupa os nós conectados em uma árvore de grupos de *multicast*. O anonimato de envio é implementado com a injeção constante de ruído para tornar indistinguíveis fases de atividade e períodos de inatividade. O anonimato de resposta é implementado pela substituição dos endereços de destino reais dos receptores pelos endereços dos grupos de *multicast*. Cifração das mensagens e manutenção de volume de tráfego constante em cada nó completam o sistema, fornecendo anonimato da relação entre emissor e receptor. Os requisitos elevados de banda passante e a necessidade de um mecanismo eficiente e confiável de *multicast* limitam a utilização da rede.

2.6.6 *Ants*

O protocolo *Ants* [CD97, GSB02] baseia-se no comportamento de colônias de formigas e foi projetado para redes dinâmicas, onde os nós não possuem posição fixa. Cada nó mantém um pseudônimo utilizado para enviar mensagens e este não revela informações adicionais a respeito de sua identidade real. Para buscar recursos na rede, um nó envia uma mensagem com

seu pseudônimo e um contador de tempo de vida, em modo *broadcast*. Cada nó que recebe a mensagem a retransmite para todos os seus vizinhos, até que o contador de tempo de vida expire ou a mensagem alcance o seu destino. Quando um nó recebe a mensagem, ele armazena a conexão na qual a mensagem foi recebida e o pseudônimo do emissor. Cada nó mantém uma tabela de roteamento dinâmica para cada um dos pseudônimos que observa. As tabelas locais de roteamento encaminham as mensagens pela conexão na qual mais mensagens do pseudônimo foram recebidas, otimizando as rotas utilizadas.

As implementações *MUTE*[6], *ANTS*[7] e *Mantis*[BSM04] implementam o protocolo *Ants* com a finalidade específica de compartilhamento de arquivos. *MUTE* utiliza contadores de tempo de vida probabilísticos para proteção contra ataques de rastreamento. Como *ANTS* não utiliza contadores, o descarte de mensagens em qualquer ponto está condicionado a uma probabilidade fixa. *Mantis* utiliza endereços de retorno falsos para envio de arquivos e controle de acesso para minimizar o efeito de ataques de identidade múltipla. A *falsificação de endereços de retorno* é utilizada para esconder a identidade do emissor de um pacote. *Mantis* utiliza UDP como camada de transporte para evitar o descarte de pacotes com endereços falsos pelos roteadores da *Internet*. O maior problema desta técnica é que normalmente os *firewalls* de provedores de acesso são configurados para descartar pacotes com endereço de rede falsificado.

2.6.7 *Freenet*

Freenet [CSWH00] é uma rede descentralizada para distribuição de documentos que tem como princípio disseminar informação de acordo com a freqüência em que ela é acessada. O anonimato concentra-se em dificultar a determinação da identidade do autor de um documento e em tornar impossível a localização de todas as cópias de um documento. Cada usuário que deseja participar da rede *Freenet* deve fornecer à rede espaço em disco para armazenamento de dados e a união destes espaços locais de armazenamento funciona como um imenso *cache* distribuído do conteúdo armazenado. É mandatório que os dados armazenados neste *cache* sejam cifrados, para que nenhum nó possa exercer censura do conteúdo que armazena. Uma vantagem que a *Freenet* oferece é o fornecimento de *repudiação*, já que qualquer nó pode negar convictamente o conhecimento dos dados que armazena.

Os documentos são identificados pelo *hash* de seu título e de palavras-chave que o identificam. Cada nó armazena uma lista dos documentos armazenados e dos disponibilizados em nós vizinhos. Para se realizar uma busca, calcula-se o *hash* do título do documento procurado e encaminha-se a requisição para o vizinho que possui o documento com o *hash* mais próximo. Este procedimento de busca termina por especializar os nós em porções mutuamente exclusivas do espaço de *hashes*, balanceando a carga e otimizando a busca de documentos. Recentemente,

[6] `http://mute-net.sourceforge.net`
[7] `http://antsp2p.sourceforge.net`

foi descoberto que a qualidade do anonimato de envio fornecido pela *Freenet* não se distribui bem entre os nós conectados [Bor05].

A implementação *Nodezilla*[8] é baseada no protocolo *Everlink*, que generaliza o conceito da *Freenet* como uma camada de comunicação anônima de finalidade geral.

2.7 Resumo

Neste capítulo, foram apresentados os conceitos básicos de anonimato computacional e as métricas fundamentais para sua avaliação. A qualidade do anonimato foi quantificada em termos de entropia e do grau de anonimato fornecido. Técnicas de anonimização convencionais e distribuídas foram apresentadas, e alguns princípios para construção de redes de anonimização foram delineados.

Adicionalmente, o modelo de adversário foi caracterizado e os principais ataques montados contra redes de anonimização foram descritos. Finalmente, várias redes de anonimização foram detalhadas e comparadas de acordo com sua viabilidade de implementação e grau de anonimato oferecido na presença de adversários distintos.

[8] `http://www.nodezilla.net`

Capítulo 3

Projeto de rede de anonimização

No capítulo anterior, um conjunto de técnicas foram descritas para se construir redes de anonimização colaborativas. Neste capítulo, são construídas técnicas eficientes para anonimatos de envio, resposta e de par comunicante viáveis para implementação em uma rede de anonimização real.

3.1 Considerações de arquitetura

Mecanismos de anonimização devem ter arquitetura descentralizada. A descentralização da estrutura do mecanismo elimina pontos únicos de falha e dificulta a monitoração por adversários poderosos. A descentralização do protocolo distribui a implementação das técnicas de anonimização entre os participantes e minimiza a influência de participantes maliciosos.

Uma arquitetura do tipo *peer-to-peer* satisfaz ambos os requisitos. Sistemas *peer-to-peer* são sistemas distribuídos compostos por nós interconectados com o propósito de compartilhar recursos – como conteúdo, processamento, armazenamento e banda – e capazes de se adaptar a falhas e acomodar populações transientes de nós, enquanto mantém conectividade e desempenho aceitáveis sem requerer a intermediação ou suporte de um servidor central ou autoridade global [ATS04]. A utilização de sistemas *peer-to-peer* na solução de problemas computacionais distribuídos é prática tradicional. Aplicações *peer-to-peer* famosas são os projetos *SETI@Home*[1] e *Folding@Home* [2].

3.1.1 Organizações

Sistemas *peer-to-peer* podem diferir quanto à organização. O tipo de organização normalmente define uma topologia e determina a alocação de identificadores para os nós conectados

[1] `http://setiathome.berkeley.edu`
[2] `http://folding.stanford.edu/`

e para a localização dos recursos. Estes identificadores são necessários para um nó conectado ao sistema poder ser encontrado e para se controlar a divisão dos recursos compartilhados entre os nós. A topologia pode governar o número de conexões que cada nó deve manter, bem como os pares de nós que podem se conectar. Quanto à organização, os sistemas dividem-se em estruturados e não-estruturados [ATS04].

Sistemas não-estruturados

Em um sistema não-estruturado, os identificadores utilizados pelos nós são alocados arbitrariamente e a localização dos recursos independe da topologia. A busca de recursos pode ser realizada por comunicação em *multicast* e caminhos aleatórios. Sistemas não-estruturados possuem topologia irregular, que apresenta diferenças de densidade e outros desequilíbrios de organização, sendo mais apropriados para acomodar populações de nós transientes [ATS04]. Exemplos de sistemas não-estruturados são Gnutella[3] e KaZaA[4]. Exemplos de redes anônimas não-estruturadas são as redes *Tor* [DMS04] e *Freenet* [CSWH00].

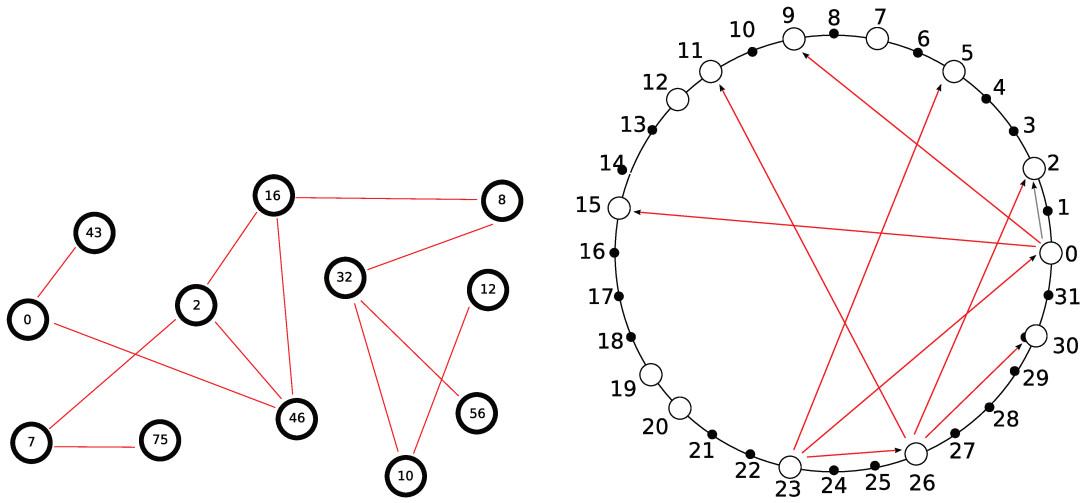
Sistemas estruturados

Em um sistema estruturado, os identificadores e as conexões obedecem a um algoritmo e a localização dos recursos compartilhados depende fortemente da topologia. Geralmente, as estruturas compartilham um esquema básico: todas utilizam um espaço de endereçamento grande, como anéis de inteiros $\mathbb{Z}_{2^{128}}$ e $\mathbb{Z}_{2^{160}}$, de onde são alocados os identificadores. As conexões dependem de uma relação matemática entre os identificadores, para aproximar a topologia de um grafo regular, como um hipercubo. A estrutura garante que cada nó mantenha um número de conexões constante ou logarítmico no tamanho total da rede e que, similarmente, todo caminho entre dois pontos quaisquer do espaço de endereçamento tenha comprimento logarítmico no tamanho da rede. A busca de recursos é realizada por algoritmos e políticas de roteamento que utilizam esta regularidade da topologia para obter eficiência. Existem vários tipos de estruturas com propriedades distintas, entre elas Chord [SMK+01], Pastry [RD01] e Tapestry [ZKJ01]. Para anonimização, temos como exemplo a rede estruturada *AP3* [MOP+04].

Os sistemas estruturados foram concebidos para minimizar os problemas típicos decorrentes da ausência de estrutura, especialmente relacionados à disponibilidade e escalabilidade. Apesar de exigirem procedimentos adicionais para a entrada e saída de nós e para manutenção da estrutura, costumam apresentar vantagens que compensam esta sobrecarga.

A Figura 3.1 apresenta dois esboços de sistema não-estruturado e estruturado. As setas em vermelho representam conexões entre pares de nós.

[3] `http://www.the-gdf.org/`
[4] `http://www.kazaa.com/`



(a) Exemplo de sistema não-estruturado.

(b) Exemplo de sistema estruturado.

Figura 3.1: Comparação entre sistema não-estruturado e sistema estruturado.

3.1.2 Escolha da organização

Como apontado anteriormente, a qualidade do anonimato não só depende do número de participantes, mas também da distribuição dos eventos entre os participantes. A uniformidade na distribuição dos eventos depende intimamente da organização do sistema e da política de roteamento utilizada.

Sistemas não-estruturados são construídos arbitrariamente e, por isso, suportam não-determinismo intrínseco que a princípio pode parecer útil para comunicação anônima. Entretanto, este mesmo não-determinismo tende a provocar desequilíbrios que prejudicam a distribuição da qualidade do anonimato entre os participantes, como comprovado por [Bor05] a partir de simulação exaustiva da *Freenet*. Outras desvantagens de sistemas não-estruturados incluem a falta de confiabilidade no roteamento (pode não ser possível encontrar um recurso mesmo ele estando presente) e a dificuldade em se estimar ou derivar limites para aspectos de desempenho e balanceamento de carga. A grande vantagem de sistemas não-estruturados é a baixa sobrecarga de manutenção da rede.

Sistemas estruturados, por sua vez, fornecem roteamento mais eficiente e confiável. Como as conexões são governadas por propriedades matemáticas, cada nó conectado à rede possui um conhecimento limitado dos demais nós, e este limite pode ser controlado com rigor. Esta propriedade de visão limitada controla a quantidade de informação que um adversário pode obter da rede. A topologia regular também distribui a responsabilidade uniformemente entre os nós, o que inibe a existência de pontos mais vulneráveis para ataque. As propriedades de

regularidade e simetria têm o potencial de fornecer qualidade de anonimato mais uniforme que abordagens não-estruturadas [Bor05]. As características determinísticas de sistemas estruturados ainda permitem simulações mais fiéis ao comportamento real das redes, o que facilita sua análise. Limites rigorosos podem ser particularmente obtidos para métricas de desempenho e balanceamento de carga e até para a eficiência de ataques efetuados por um adversário.

Sistemas estruturados fornecem maior potencial para comunicação anônima e são utilizados no projeto da rede de anonimização desenvolvido neste trabalho. As discussões subseqüentes irão se restringir, portanto, a sistemas estruturados.

3.2 Comunicação anônima em sistemas estruturados

A funcionalidade básica de sistemas estruturados é fornecer uma primitiva de *tabela de hash distribuída* (*Distributed Hash Table - DHT*) [SMK+01]. Em uma tabela de *hash* distribuída, as operações básicas são de *armazenamento* e *recuperação*, implementadas a partir de troca de mensagens entre os nós. Os recursos compartilhados são mapeados para o espaço de endereçamento utilizado pelos nós a partir da aplicação de uma função de *hash* ao conteúdo ou descrição do recurso. A localização de cada recurso é baseada na similaridade entre os identificadores do recurso e dos nós conectados, calculada por uma função de distância. Para se garantir o balanceamento da carga de armazenamento, o controle do espaço de endereçamento é particionado eqüitativamente entre todos os nós conectados. Um esquema de particionamento comumente utilizado é conferir para cada nó o controle da porção de endereços maiores que o identificador do seu predecessor e menores ou iguais ao seu próprio identificador.

O primeiro passo para se suportar comunicação anônima em sistemas estruturados é anonimizar o roteamento que transporta as operações de armazenamento e recuperação de recursos. Esta modificação agrega anonimato de envio à estrutura. A anonimização destas operações básicas transforma a tabela de *hash* distribuída em sua versão anonimizada. Isto permite a anonimização direta de diversas aplicações que utilizam tabelas de *hash* distribuídas, especialmente para publicação de documentos [CDKR02]. No Capítulo 5, a funcionalidade de tabela de *hash* anonimizada será aproveitada para o projeto de um serviço de nomes para redes de anonimização.

Como o mesmo roteamento que transporta uma requisição de operação através da rede pode também ser utilizado para o envio de uma mensagem qualquer, a anonimização do roteamento possibilita a utilização de um sistema estruturado para comunicação anônima genérica. Na seção seguinte, um sistema estruturado é adaptado para suportar anonimato de envio. Modificações adicionais são necessárias para suporte a anonimato de resposta e são discutidas posteriormente.

3.2.1 Anonimato de envio

O roteamento em sistemas estruturados é *recursivo*. O nó que efetua uma operação de armazenamento ou recuperação utiliza a chave do recurso compartilhado para avaliar cada um dos seus vizinhos quanto à distância e selecionar o próximo ponto na rota. Os nós intermediários repetem o procedimento recursivamente até que o nó detentor da porção de endereçamento compatível com o recurso seja encontrado. A característica de regularidade da estrutura do sistema permite que um destino único sempre seja encontrado e que o roteamento seja *convergente*. Similarmente, um nó que envia uma mensagem qualquer para um nó de destino utiliza o identificador do destino como chave no roteamento. As respostas são encaminhadas percorrendo a mesma rota utilizada para envio, em sentido contrário.

A Figura 3.2 apresenta em setas pretas a rota percorrida por uma mensagem enviada pelo nó com identificador 23 para o nó de destino 12. Na figura, é possível observar o espaço de endereçamento $\mathbb{Z}_{2^5}$, com os nós conectados representados na cor branca. É possível também observar em vermelho as conexões mantidas pelo nó 23, o círculo que delimita a porção do endereçamento sob controle do nó 23 e a atuação de uma função de distância nas decisões de roteamento.

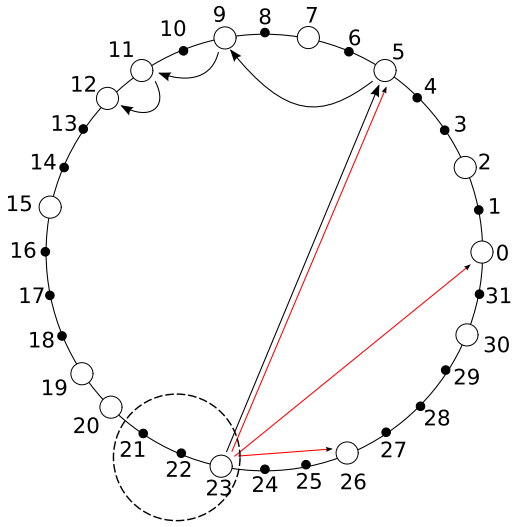


Figura 3.2: Exemplo de roteamento determinístico em um sistema estruturado.

É fácil perceber que o roteamento determinístico revela informações úteis para um adversário. Um nó intermediário pode calcular, por exemplo, a distância entre o seu identificador e o destino para inferir o identificador do nó emissor. O primeiro nó intermediário da rota também pode identificar a origem da mensagem com grande probabilidade. Os identificadores dos nós não podem ser mantidos em segredo, já que precisam ser conhecidos pelos seus vizi-

nhos para roteamento e manutenção da estrutura. Logo, descobrir o identificador do emissor de uma mensagem é praticamente equivalente a quebrar o seu anonimato de envio.

Uma solução para o problema do roteamento determinístico [MOP+04] utiliza uma idéia da rede *Crowds* [RR98]: um caminho aleatório é percorrido antes da entrega da mensagem ao destino real. O nó emissor primeiramente seleciona um nó arbitrário na rede e encaminha a mensagem. O nó selecionado realiza um sorteio p_r condicionado por uma *probabilidade de encaminhamento* $0 \leq p_f < 1$ e decide se a mensagem deve ser encaminhada novamente para outro nó arbitrário ou deve ser entregue ao destino final. Cada nó intermediário repete o procedimento até que o sorteio falhe e a mensagem seja devidamente entregue. A Figura 3.3 apresenta o esquema: as setas tracejadas correspondem a sorteios que decidiram pelo encaminhamento para outro nó e a seta restante corresponde ao sorteio que decidiu pela entrega da mensagem.

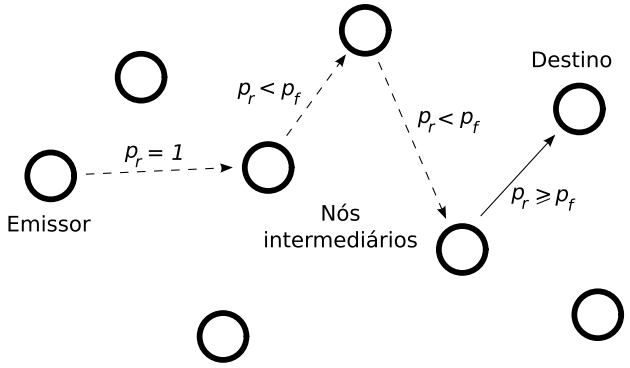


Figura 3.3: Exemplo de anonimização de envio na rede *Crowds*.

Aplicando a idéia em um sistema estruturado, o roteamento divide-se em duas partes: a primeira parte percorre um caminho aleatório entre os nós da estrutura e a segunda parte corresponde ao roteamento recursivo determinístico. O nó emissor da mensagem seleciona um identificador aleatório no espaço de endereçamento e executa o algoritmo de roteamento determinístico para entregar a mensagem. Cada nó intermediário repete o procedimento até que um deles decida por entregar a mensagem para o destino e execute a segunda fase do roteamento. Desta forma, as mensagens roteadas atingem um ponto aleatório na rede antes de serem encaminhadas para o destino final.

A Figura 3.4 apresenta o roteamento em duas fases utilizado pela rede *AP3* [MOP+04]. Na figura, o nó 23 envia uma mensagem para o nó 12, e sorteios sucessivos da probabilidade p_r são realizados, alternados por execuções do algoritmo de roteamento determinístico. Cada sorteio $p_r < p_f$ provoca uma execução do roteamento determinístico no caminho aleatório e o sorteio $p_r \geq p_f$, realizado pelo nó 20, provoca a entrega da mensagem.

O caminho aleatório composto por uma seqüência de roteamentos determinísticos apresenta

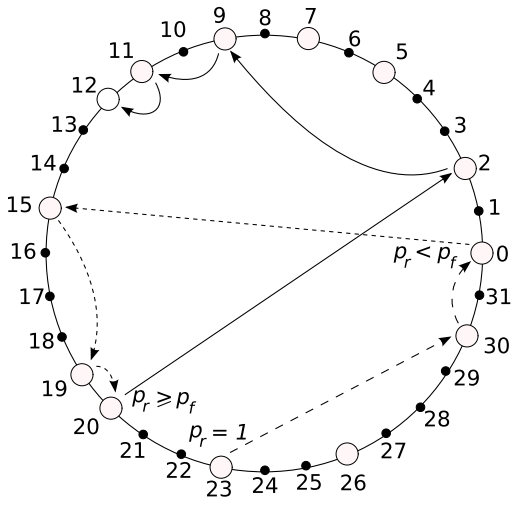


Figura 3.4: Exemplo de anonimização de envio na rede *AP3*.

uma desvantagem. Como o comprimento da rota entre dois pontos quaisquer na rede não tem tamanho fixo, apesar de ser limitado logaritmicamente pelo tamanho da rede, a previsão do tamanho do caminho aleatório percorrido é inacurada: as rotas determinísticas utilizadas têm tamanhos variados e podem resultar em caminhos aleatórios muito longos e de baixo desempenho. Um aperfeiçoamento desta idéia foi proposto por Borisov [Bor05] e utiliza as conexões do sistema estruturado nas decisões do caminho aleatório. Ao invés de encaminhar a mensagem para um identificador aleatório, cada nó encaminha a mensagem para um dos vizinhos que conhece, sorteado aleatoriamente, até que a segunda fase do roteamento seja iniciada. A mesma probabilidade de encaminhamento p_f é utilizada para decidir se a mensagem é encaminhada novamente ou entregue ao destino. Um caminho aleatório desta natureza e suficientemente longo irá atingir um ponto aleatório na rede, independente da origem, e a partir do qual o roteamento pode ser completado. Este novo algoritmo de roteamento é referenciado posteriormente como *roteamento randomizado* e é utilizado como algoritmo de roteamento para o envio de qualquer mensagem na rede.

A Figura 3.5 ilustra o algoritmo de roteamento randomizado. Novamente o nó 23 envia uma mensagem para o nó 12, mas os sorteios de probabilidade são agora alternados com encaminhamentos diretos da mensagem para um nó vizinho. Cada sorteio $p_r < p_f$ provoca a retransmissão da mensagem para um vizinho selecionado aleatoriamente e o sorteio $p_r \geq p_f$, realizado pelo nó com identificador 5, inicia a entrega da mensagem com roteamento determinístico.

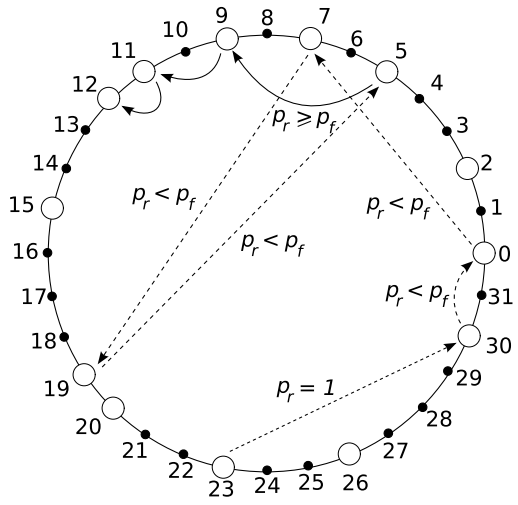


Figura 3.5: Exemplo de anonimização de envio com roteamento randomizado.

3.2.2 Anonimato de resposta

Apesar do anonimato de envio proteger a identidade do emissor das mensagens, é insuficiente para suportar os fluxos de requisição e resposta presentes na maioria dos protocolos. Para permitir resposta, cada mensagem enviada deve transportar um endereço de resposta, a partir do qual o emissor pode ser encontrado. O endereço de resposta deve ser independente do identificador do emissor para não comprometer sua identidade.

Uma solução simples para o problema é proposta em [MOP+04] e utiliza endereços de resposta aleatórios. Antes de enviar uma mensagem, o emissor cria um *canal de resposta* identificado por um endereço aleatório no espaço de endereçamento. Para criar o canal, o emissor envia uma mensagem especial utilizando roteamento randomizado até o nó que detém a porção de endereçamento que contém com o endereço do canal. Cada nó intermediário do caminho aleatório que recebe esta mensagem especial grava a direção pela qual a mensagem foi recebida em uma *tabela de resposta* local. A mensagem especial eventualmente atinge o seu destino e o canal é completado quando o nó de destino concorda em atuar como *ponto de entrada*, utilizando o canal no sentido contrário para encaminhar todas as mensagens destinadas ao seu criador. A criação do canal deve estar condicionada a um limite de tempo que, quando ultrapassado, força a expiração do canal antigo e obriga o estabelecimento de um novo canal.

A Figura 3.6 ilustra a criação de um canal de resposta pelo nó 23 com ponto de entrada 9. Na figura, as setas de maior espessura indicam o canal de resposta, e os nós com identificadores 23 e 12 trocam mensagens. A requisição R transporta a mensagem m para o nó 12 utilizando roteamento randomizado e especifica o endereço do nó 9 como endereço de resposta.

A requisição R apenas revela o endereço do ponto de entrada do canal de resposta, protegendo a identidade do emissor. A resposta R' é encaminhada para o nó 9, por meio do qual atinge o emissor percorrendo o canal na ordem inversa de criação. O nó 12 não utiliza canal de resposta para receber mensagens.

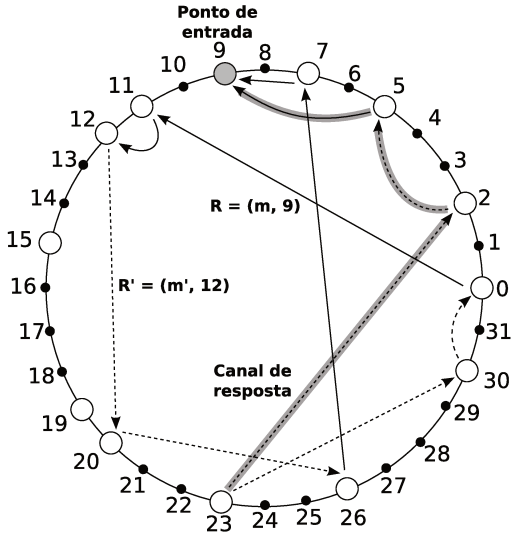


Figura 3.6: Exemplo de anonimização de resposta na rede *AP3*.

A qualidade do anonimato de resposta é intuitivamente dependente da qualidade do anonimato de envio, já que utiliza as mesmas primitivas de comunicação. A modificação desta técnica para anonimato de resposta permite ainda o estabelecimento de pseudônimos.

Pseudônimos

As técnicas de roteamento randomizado e de canais de resposta são ideais para troca eventual de mensagens. Entretanto, não há qualquer impedimento para um adversário personificar qualquer emissor para forjar mensagens ou modificar o endereço de resposta de mensagens interceptadas para que atravessem um dos nós que controla. É necessário, portanto, a utilização de um mecanismo que agregue reputação e permita aos nós confirmar a autenticidade da parte com a qual se comunicam.

Sendo desconhecida a identidade real dos nós conectados, a reputação é construída em torno de um identificador persistente e cuja posse pode ser provada criptograficamente. Um identificador com estas características qualifica-se como um *pseudônimo*.

Para um nó a_i estabelecer um pseudônimo confiável, deve gerar um par de chaves assimétrico (e_{a_i}, d_{a_i}) e calcular um pseudônimo dependente do seu par de chaves. O pseudônimo

α_i é derivado a partir da aplicação de uma função de *hash* criptográfica h à chave pública. Um canal de resposta autenticado utiliza o *hash* do pseudônimo $h(\alpha_i) = h(h(e_{a_i}))$ como ponto de entrada. Qualquer nó que conheça o pseudônimo previamente pode confirmar a autenticidade da chave pública, pois detém um *hash* da mesma, e verificar a prova da chave privada correspondente utilizando assinatura digital [MOP+04]. Ataques de *espelhamento* (do inglês, *man-in-the-middle attack*) são completamente evitados [BG03].

A Figura 3.7 ilustra o estabelecimento de pseudônimo por parte do nó com identificador 12 e de um canal de resposta autenticado pelo pseudônimo $h(e_{12})$ e com ponto de entrada $h(h(e_{12})) = 19$. O nó 23 envia uma mensagem para o nó 12, utilizando o ponto de entrada do canal autenticado. O nó 23 deriva o ponto de entrada do canal autenticado a partir do conhecimento prévio do pseudônimo. Como na figura anterior, o nó 23 mantém um canal de resposta não-autenticado com ponto de entrada 9.

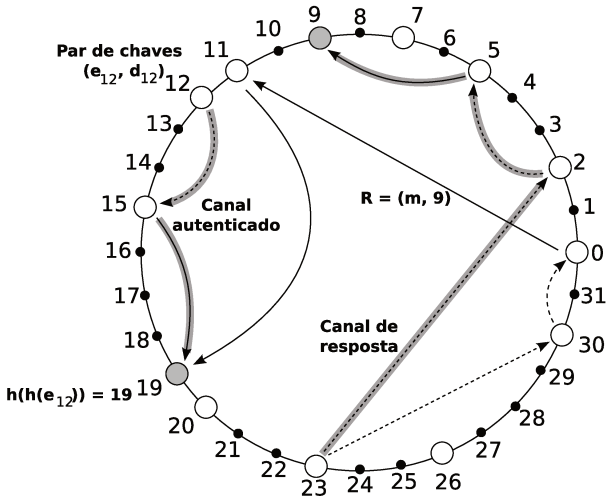


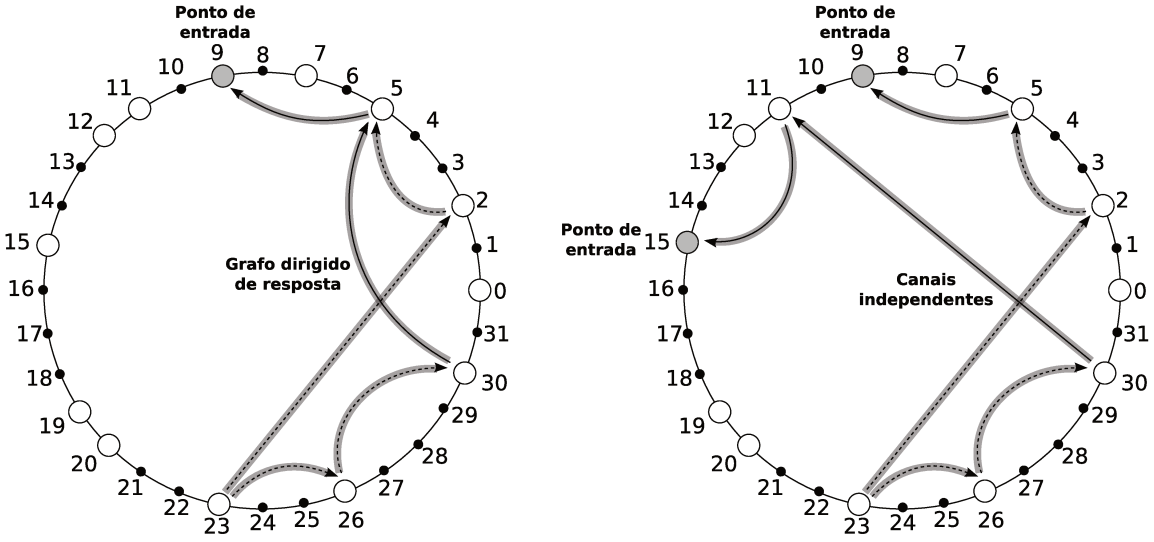
Figura 3.7: Exemplo de estabelecimento de pseudônimo na rede *AP3*.

Aprimoramento

A proposição original apresenta limitações práticas relacionadas a desempenho e resistência a ataques. A utilização de um único canal de resposta que concentra todas as respostas a variadas requisições pode prejudicar muito a latência de transmissão. Além disso, nós controlados pelo adversário presentes no canal de resposta podem descartar as mensagens, realizado um ataque efetivo de negação de serviço.

As soluções para ambas as limitações partem do estabelecimento de múltiplos canais de resposta com endereços distintos. Uma proposta na literatura sugere a utilização de *grafos diri-*

gidos de resposta [LM]. Para formar o grafo dirigido, o criador seleciona um ponto de entrada e solicita o estabelecimento do canal de resposta para vários nós. Cada um dos nós que recebe a solicitação realiza um sorteio ponderado e replica a solicitação para outro nó em caso de sucesso. Quando um nó falha no sorteio, conecta-se diretamente ao ponto de entrada. O grafo dirigido funciona como um canal único, tendo um ponto único de entrada e um ponto único de saída e diversas bifurcações internas. As desvantagens desta abordagem são a centralização do ponto de entrada e o alto custo de estabelecimento e renovação do grafo. Além disso, a carga e a responsabilidade dos nós próximos às pontas do canal é maior, pois a indisponibilidade repentina dos nós próximos às pontas acarreta conseqüências graves em desempenho e confiabilidade.



(a) Exemplo de grafo dirigido de resposta.

(b) Exemplo de canais de resposta múltiplos e independentes.

Figura 3.8: Diferentes estratégias para estabelecimento de canais múltiplos de resposta.

Uma alternativa é proposta e utilizada neste trabalho e envolve o estabelecimento independente de múltiplos canais de resposta com pontos de entrada distintos. Canais de resposta não-autenticados podem ser utilizados se os vários endereços de resposta de um emissor acompanharem suas mensagens enviadas. O suporte a pseudônimos é mantido se os endereços dos canais corresponderem a aplicações sucessivas da função de *hash* h ao pseudônimo: $h(\alpha_i)$, $h(h(\alpha_i))$, $h(h(h(\alpha_i)))$. A vantagem desta nova abordagem é a possibilidade de se estabelecer canais múltiplos paralelamente, distribuindo o custo de renovação no decorrer do tempo. O ponto de entrada centralizado também é eliminado. Qualquer nó que conheça o pseudônimo previamente continua podendo verificar a posse do par de chaves associado ao pseudônimo

e pode contatar o criador do canal utilizando vários pontos de entrada simultaneamente. A descentralização adicional ainda distribui a carga e a responsabilidade igualmente entre os canais e os nós que os compõem.

A Figura 3.8 apresenta duas estratégias de estabelecimento de canais múltiplos de resposta, ilustradas por um grafo de resposta simplificado (Figura 3.8(a)) e um par de canais de resposta independentes (Figura 3.8(b)). As vantagens da alternativa proposta, como descentralização do ponto de entrada e distribuição de responsabilidade, são evidenciadas na figura: a indisponibilidade repentina do nó 5 no grafo dirigido de resposta invalida todo o grafo, enquanto a indisponibilidade do nó 5 no canal independente, invalida apenas o canal em que participa.

Para o projeto da rede de anonimização, canais múltiplos e independentes são utilizados para fornecimento de anonimato de resposta.

3.2.3 Anonimato de par comunicante

A randomização do roteamento, utilizada para fornecer anonimatos de envio e de resposta, espalha as mensagens trocadas por vários nós intermediários, implementando a técnica de indireção. Entretanto, o conteúdo das mensagens permanece às claras e suscetível à análise por parte de um adversário externo:

- Um adversário local externo pode identificar pares comunicantes com algum esforço, correlacionando a informação às claras capturada nas conexões da rede que monitora; e
- Um adversário global externo que monitora todo o sistema de comunicação pode identificar trivialmente rotas e pares comunicantes.

A utilização de cifração impede a eficácia destes ataques. Para o projeto da rede de anonimização, a cifração é adicionada em dois níveis: inicialmente no nó emissor, utilizando a chave pública do receptor, cujo pseudônimo conhece; e posteriormente nas conexões diretas utilizadas para roteamento. A cifração nas partes comunicantes propriamente ditas impede o descarte sistemático de mensagens por nós intermediários e a cifração nas conexões intermediárias limita a observação de adversários externos. Técnicas já discutidas de igualdade entre mensagens são ainda utilizadas para obscurecer o tráfego que atravessa a rede.

Adversários internos, entretanto, ainda podem obter informação a respeito de rotas percorridas examinando as mensagens cifradas que atravessam os nós que controla. Esta limitação não oferece perigo significativo, já que cada mensagem enviada na rede utiliza um caminho aleatório diferente.

A utilização de cifração de dois níveis fornece ainda *repudiação*: todos os nós podem negar convictamente o conhecimento do tráfego que roteiam. As técnicas descritas para anonimato de envio e resposta também colaboram para a qualidade de anonimato do par comunicante.

3.3 Avaliação de topologias

Utilizando técnicas propostas em diversos trabalhos [RR98, BG03, MOP+04, Bor05] e aprimoramentos como canais múltiplos de resposta e cifração em dois níveis, um sistema estruturado genérico pôde ser adaptado para comunicação anônima. Deve-se, por último, escolher qual das topologias estruturadas propostas na literatura tem maior potencial para anonimato.

3.3.1 Critérios de seleção

A *capacidade de mistura* de um sistema estruturado é a capacidade do sistema em atingir um ponto independente da origem após o percorrimento de um caminho aleatório em sua topologia. Sistemas têm melhor capacidade de mistura quando são capazes de atingir um ponto aleatório na rede com um caminho aleatório de menor comprimento. Esta grandeza está intimamente relacionada à qualidade de anonimato e, mais especificamente, ao compromisso entre desempenho e anonimato. Sistemas estruturados com melhor capacidade de mistura devem fornecer melhor desempenho e comunicação anônima de melhor qualidade [Bor05].

A capacidade de mistura é medida a partir da *distância de variação* entre a distribuição uniforme e a distribuição dos nós finais em caminhos aleatórios. Recentemente, várias topologias estruturadas foram avaliadas de acordo com a capacidade de mistura [Bor05]. O objetivo deste estudo foi apontar, a partir de simulação, as topologias com melhor capacidade de mistura para utilização em comunicação anônima.

Para selecionar uma topologia estruturada útil para comunicação anônima, neste trabalho, optou-se pela realização de novos experimentos análogos aos descritos em [Bor05] e com o mesmo objetivo, mas com critérios de avaliação distintos. A seleção de uma topologia adequada foi condicionada à observação do comportamento de cada uma das topologias avaliadas quando utilizadas para comunicação anônima. O critério de seleção não foi a capacidade de mistura, mas a métrica de entropia. O compromisso entre a métrica de entropia e o desempenho, ou entre a qualidade do anonimato e desempenho, também foi considerado. A capacidade de resistência das estruturas a ataques de negação de serviço também foi rapidamente examinada. Esta abordagem aponta com maior clareza as vantagens das topologias selecionadas e deve confirmar a relação íntima entre capacidade de mistura e entropia.

A métrica de entropia é calculada a partir da simulação de redes comunicando-se anonimamente. As redes são construídas dinamicamente e de forma não-determinística. A entropia poderia ser calculada formalmente a partir das fórmulas apresentadas no Capítulo 2, mas a complexidade do sistema e de sua população impossibilita esta abordagem. A utilização de simulações é motivada diretamente por esta complexidade.

3.3.2 Candidatos

Foram consideradas algumas das topologias estruturadas mais populares na área de pesquisa em sistemas estruturados: *Chord* [SMK+01], *Chord* randomizado [GGG+03], hipercubo [RD01, ZKJ01], hipercubo randomizado [CDG+02], *SkipGraph* [AS03], *SkipNet* [HJS+03] e *Koorde* [KK03]. As diferenças em relação ao experimento original são:

- Inclusão da versão randomizada da topologia *Chord*;
- Inclusão das topologias de hipercubo e hipercubo randomizado, para suplantar as topologias *Pastry* [RD01], *Tapestry* [ZKJ01] e variantes;
- Inclusão das topologias *SkipGraph* e *SkipNet*;
- Remoção das topologias *CAN* [RFH+01] e *Viceroy* [MNR02], por apresentarem resultados muito desfavoráveis no experimento original [Bor05];
- Simulação da segunda fase do roteamento randomizado, resultando em uma simulação mais realista do ambiente;
- Avaliação de diversas variantes da topologia *Koorde*; e
- Análise dos percentuais de chegada de pacotes comprimentos de rota descritos.

A razão para a inclusão das topologias *SkipGraph* e *SkipNet* é a presença de um grau extra de randomização no procedimento de construção da rede, que pode ser favorável para comunicação anônima e merece ser observado. Esta mesma observação também motivou a inclusão das topologias *Chord* randomizado e hipercubo randomizado.

Nas descrições subseqüentes, n é o número de nós conectados à rede e o espaço de endereçamento é o anel de inteiros $\mathbb{Z}_{2^b}$. O sucessor e o predecessor de um nó são os nós que o precede e o sucede, respectivamente, com a ordenação no sentido anti-horário do anel. Cada nó é responsável pela porção de endereçamento compreendida entre o primeiro identificador após o seu predecessor e o seu próprio identificador. A atribuição de identificadores é aleatória, para que o espaço de endereçamento seja dividido eqüitativamente entre os nós. As conexões são denotadas por setas em vermelho, podendo ser unidirecionais ou bidirecionais. Em todas as topologias, além das conexões determinadas por relação matemática, cada nó deve manter pelo menos uma conexão com o seu sucessor na estrutura. As conexões de cada nó para o seu sucessor são necessárias para que uma coesão mínima da rede seja mantida na presença de falhas consecutivas de uma porção significativa dos nós conectados.

A Figura 3.9 ilustra estes conceitos. São apresentados os nós sucessor e predecessor do nó 0, o círculo que delimita o espaço de endereçamento sob controle do nó 0 e as conexões que o nó 0 mantém com os nós 2, 5, 9 e 19, governadas por uma topologia hipotética.

Nos tópicos subseqüentes, as topologias avaliadas são resumidamente apresentadas.

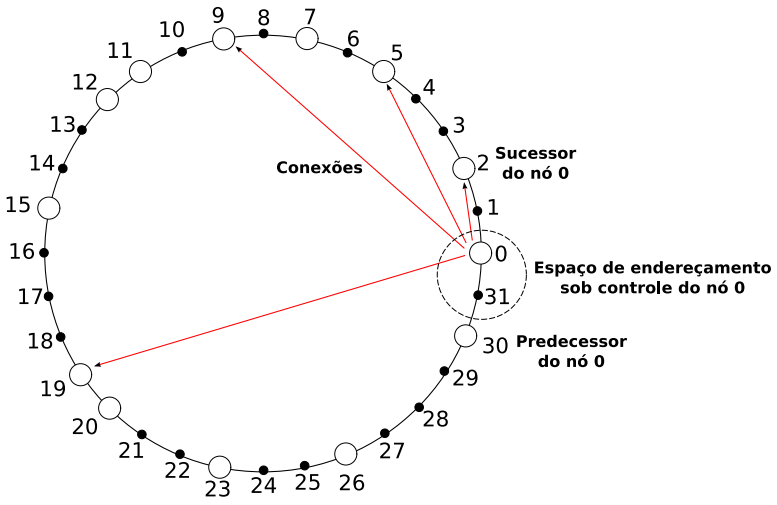


Figura 3.9: Exemplo de topologia estruturada, ilustrando a participação de um nó particular.

Chord [SMK+01]

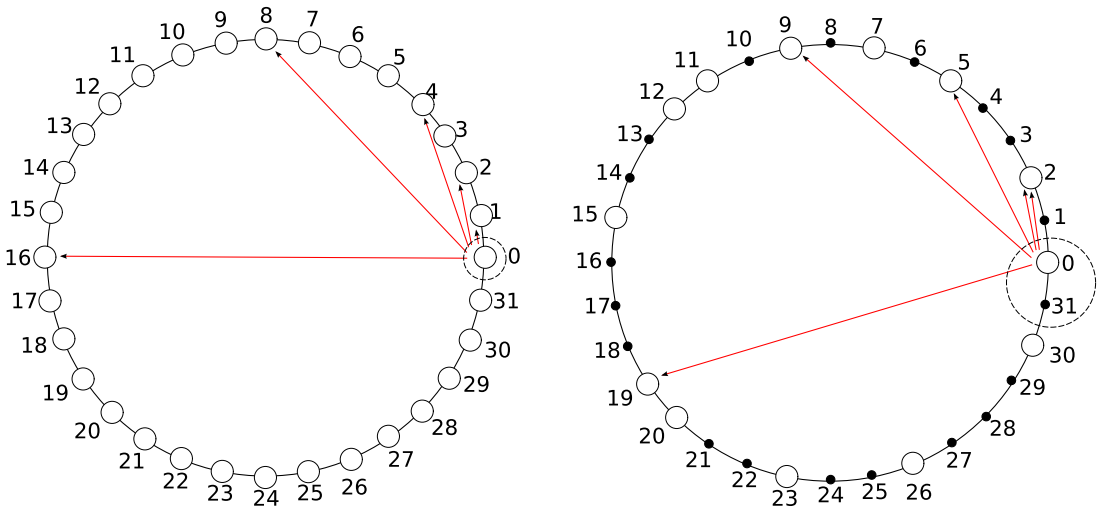
A topologia *Chord* é uma das mais populares. Nesta topologia, um nó com identificador x conecta-se ao seu sucessor e a b outros nós, com identificadores $x+2^i \pmod{2^b}$ para $0 \leq i < b$.

Como normalmente a rede é esparsa, tendo muito menos que 2^b nós, um nó com identificador x conecta-se aos nós que possuem os identificadores $x+2^i \pmod{2^b}$ para $i = 0, 1, \ldots, b-1$ nas porções do espaço de endereçamento que controlam. Conseqüentemente, algumas das conexões mantidas por um nó terminam em um mesmo vizinho.

A Figura 3.10 ilustra a topologia *Chord*, em uma configuração ideal (rede completa) e em uma configuração prática (rede esparsa). Em ambas as configurações, as conexões estabelecidas pelo nó com identificador 0 apresentam-se em vermelho, bem como a porção de endereçamento de responsabilidade do nó 0. Na rede esparsa, pode-se observar que a compensação da topologia determina os vizinhos do nó 0 a partir do particionamento do espaço de endereçamento.

Chord randomizado [GGG+03]

Chord randomizado é uma variante da topologia *Chord*. A diferença reside no não-determinismo da topologia: um nó com identificador x conecta-se ao seu sucessor e a b outros nós, com identificadores $x + 2^i + r(i) \pmod{2^b}$, com $0 \leq i < b$ e $r(i)$ um inteiro uniformemente aleatório no intervalo $[0, 2^i)$.



(a) Exemplo de topologia *Chord* completa.

(b) Exemplo de topologia *Chord* esparsa.

Figura 3.10: Exemplos de configurações ideal e prática de uma topologia *Chord*.

Hipercubo [RD01, ZKJ01]

Em uma topologia de hipercubo, cada nó conecta-se a seu sucessor e a b outros nós. Para $i \leq 1 \leq b$, um nó com identificador x conecta-se com um nó y, se os *bits* de x e y forem idênticos, com exceção do i-ésimo *bit*. As observações a respeito do caráter esparso da rede também se aplicam neste caso.

Hipercubo randomizado [CDG+02]

Em um hipercubo randomizado, para $1 \leq i \leq b$, um nó com identificador x conecta-se ao seu sucessor e aos nós que compartilham os mesmos i *bits* mais significativos e diferem no i-ésimo *bit*. Os demais *bits* são gerados aleatoriamente.

SkipGraph [AS03]

Em um *SkipGraph*, um nó possui um identificador x e um conjunto de conexões. As conexões são definidas por um *vetor de pertinência* m_x, formado por uma cadeia infinita de *bits* aleatórios. Vetores de pertinência são gerados independentemente por cada nó.

Como nas outras topologias, os nós escolhem identificadores no espaço $\{0,1,\ldots,n-1\}$ e organizam-se em um círculo ordenado. O nó com identificador x conecta-se necessariamente ao seu predecessor e ao seu sucessor. As demais conexões são determinadas pelos vetores de pertinência. Seja $m_{x,i}$ os i primeiros *bits* de m_x e seja (x,y) o intervalo de identificadores entre x

e y, em sentido anti-horário de x para y. Os nós x e y são conectados se para algum j, $m_{x,j} = m_{y,j}$, e não há qualquer nó $z \in (x,y)$ tal que $m_{z,j} = m_{x,j}$. Ou seja, dois nós estão conectados se os seus vetores de pertinência compartilham algum prefixo que não é compartilhado por nenhum dos nós entre eles. Com alta probabilidade, cada nó mantém um número de conexões logarítmico em n. Por esta razão, o vetor de pertinência pode ser instanciado sob demanda e, normalmente, apenas b *bits* do vetor de pertinência precisam ser instanciados.

Uma propriedade útil do *SkipGraph* é que as conexões não dependem de propriedades matemáticas dos identificadores, mas apenas de sua ordenação e vetores de pertinência. Por isso, a topologia *SkipGraph* oferece funcionalidade de árvore e suporta buscas complexas, como a busca de recursos que se localizam dentro de um intervalo desejado [AS03].

SkipNet [HJS+03]

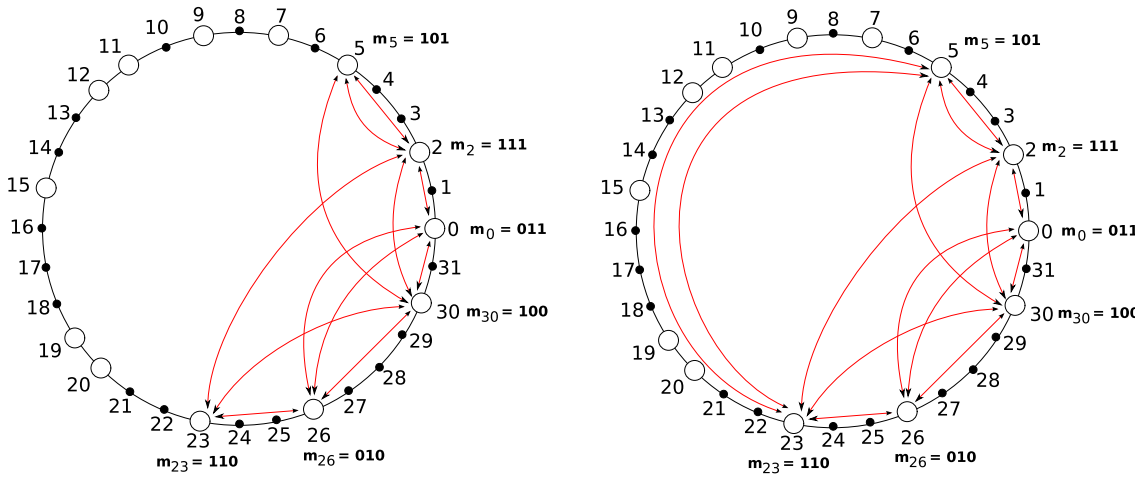
SkipNet é uma topologia bastante similar a *SkipGraph*, proposta independentemente. Uma *SkipNet* é uma superposição de múltiplos anéis, construídos probabilisticamente. Cada nó armazena um identificador numérico especial, que funciona como um vetor de pertinência. Nós que compartilham um prefixo de j *bits* do identificador especial participam de um dos anéis de nível j.

A Figura 3.11 ilustra tanto um *SkipGraph* como uma *SkipNet*. Alguns nós da estrutura são desconsiderados na figura por simplificação. As conexões bidirecionais são apresentados e pode-se verificar a relação entre as conexões e os vetores de pertinência. A diferença essencial entre as duas topologias reside na superposição de anéis da topologia *SkipNet*, em contraste à superposição de listas da topologia *SkipGraph*.

Koorde [KK03]

Koorde é uma estrutura baseada em grafos *de Bruijn* [dB46]. Um grafo *de Bruijn* com t dimensões é um grafo dirigido que representa sobreposições entre seqüências de símbolos. O grafo completo tem t^n vértices, consistindo em todas as seqüências de símbolos de comprimento n. Se um vértice v pode ser representado pelo deslocamento de todos os símbolos de um vértice u e adição de um novo símbolo à direita, então existe uma aresta dirigida de u para v. Grafos *de Bruijn* são favoráveis para aplicações *peer-to-peer* porque possuem grau constante.

Na topologia *Koorde*, um nó com identificador x conecta-se a seu sucessor e a d outros nós com identificadores $d \cdot x + j \pmod{2^b}$, com $0 \leq j < d$. Para um espaço de endereçamento binário, d é escolhido como uma potência de 2. Assim, os identificadores podem ser vistos como seqüências de $\frac{b}{\log_2 d}$ dígitos na base d, com as conexões definidas por um deslocamento à esquerda e inserção de um novo dígito à direita. Uma propriedade relevante de grafos *de Bruijn* é que um caminho aleatório iniciado no nó x com comprimento $\frac{b}{\log_2 d}$, termina em um nó com identificador totalmente diferente de x. A quantidade d é denominada *grau* da topologia.



(a) Exemplo de topologia *SkipGraph*.

(b) Exemplo de topologia *SkipNet*.

Figura 3.11: Comparação entre as topologias *SkipGraph* e *SkipNet*.

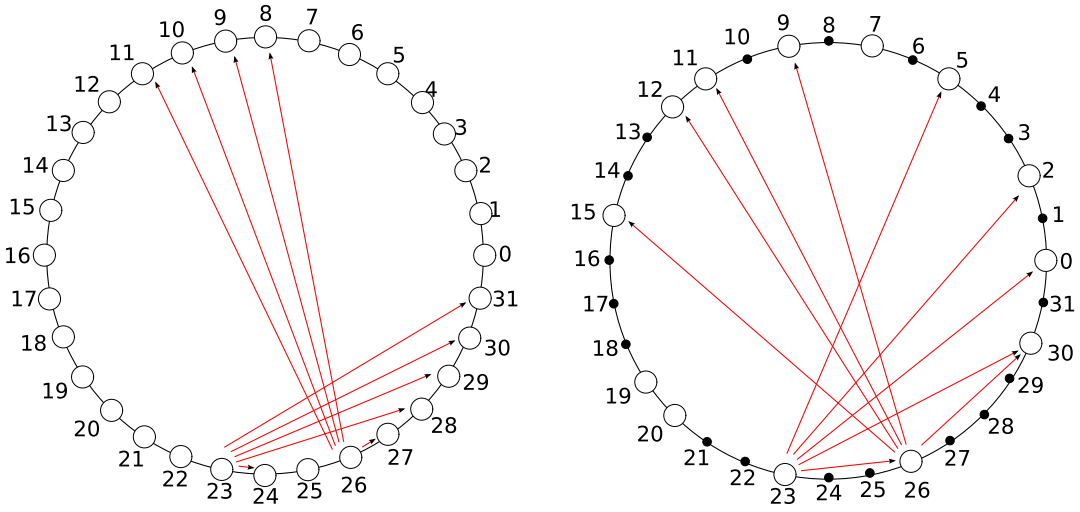
A estrutura *Koorde* exige adaptação para redes esparsas: um nó com identificador x conecta-se ao nó $y = d \cdot x \pmod{2^b}$ e os d sucessores de y. Isto é necessário para que as propriedades de roteamento na rede sejam mantidas, especialmente o comprimento logarítmico de qualquer rota.

A Figura 3.12 ilustra configurações ideal e prática para uma topologia *Koorde*. Na figura, as conexões dos nós 23 e 26 são destacadas e o grau utilizado é $d = 4$. A notação *Koorde-d* é utilizada para denotar uma topologia *Koorde* de grau d.

3.3.3 Adversário

O adversário é modelado como um conjunto de nós comprometidos e atuando em conluio, compartilhando conhecimento entre si. Considerando o contexto de sistemas *peer-to-peer* estruturados, é um adversário local, interno e passivo. As observações do adversário são realizadas por meio da captura de mensagens nos nós que controla. O ataque executado pelo adversário é um ataque de predecessor [WALS04]: analisando os vizinhos imediatos que encaminharam as mensagens para os nós comprometidos, o adversário tenta inferir as verdadeiras origens das mensagens.

Como ataques de predecessor são particularmente efetivos em redes que utilizam roteamento por caminhos aleatórios [WALS04], avaliar as topologias utilizando a efetividade de um ataque de predecessor é um procedimento válido de comparação. Entretanto, as conclusões obtidas são estritamente válidas para cenários que reproduzem com fidelidade as características do adversário e ataque considerados.



(a) Exemplo de topologia *Koorde-4* completa.

(b) Exemplo de topologia *Koorde-4* esparsa.

Figura 3.12: Exemplos de configurações ideal e prática de uma topologia *Koorde* de grau 4.

3.3.4 Simulação

Para cada uma das topologias estruturadas, um experimento é realizado. O número total de nós na rede é n, dos quais c nós são controlados por adversários. Em cada experimento, 22 comprimentos distintos de caminho aleatório são avaliados. Para cada comprimento, são executadas 50 simulações. Uma única simulação é composta pelas seguintes etapas:

1. Seleção de um nó não-controlado por adversário $a_d \in \mathcal{A}$, o destino único para todas as mensagens da simulação;

2. Execução de eventos de comunicação consecutivos, que simulam a transmissão de k mensagens de cada um dos participantes $a_i \in \mathcal{A}$ para o destino a_d. Os nós controlados pelo adversário e o destino a_d não participam da simulação como emissores de mensagem. Assim, o número total de eventos de comunicação de uma simulação é $k(n-c-1)$;

3. Gravação do predecessor observado para cada mensagem interceptada por nó malicioso. Os nós controlados pelo adversário descartam as mensagens capturadas imediatamente após a gravação da observação correspondente; e

4. Cálculo da métrica de entropia condicional do sistema, considerando as observações do adversário.

Para cada comprimento de caminho aleatório l, a média aritmética é tomada sobre as 50 medidas de entropia condicional calculadas. O resultado é a *entropia condicional do roteamento*

randomizado para um caminho aleatório de comprimento l.

A simulação objetiva facilitar o cálculo da quantidade de informação que o roteamento randomizado revela a respeito da origem das mensagens. Ou seja, a simulação avalia as topologias apenas pela qualidade de anonimato de envio que fornecem. Mas, geralmente, os anonimatos de resposta e de par comunicante são também favorecidos por uma boa qualidade de anonimato de envio.

Para o cálculo da entropia, é utilizado o procedimento descrito por Borisov [Bor05]. Primeiramente, é necessário obter a distribuição conjunta de probabilidade das variáveis A, Y, com A tomando valores no conjunto de participantes e Y tomando valores no domínio de observações do adversário. O domínio da variável A é o conjunto dos identificadores dos participantes $\mathcal{A} = \{1, 2, \ldots, n-c\}$. O domínio da variável Y é a união entre o conjunto dos identificadores dos nós que podem ser observados como predecessor e um elemento especial para indicar que nenhum predecessor foi observado (a mensagem não foi interceptada), ou seja, $\mathcal{Y} = \mathcal{A} \cup \{\varnothing\}$. Um contador $c_{a_i,y}$ é utilizado para cada par (a_i, y) e armazena o número de vezes que um predecessor $y \in \mathcal{Y}$ foi observado pelo adversário em eventos de comunicação iniciados pelo participante $a_i \in \mathcal{A}$.

Este procedimento permite estimar a distribuição conjunta de probabilidade A, Y empiricamente. A probabilidade estimada no ponto (a_i, y), é dada por $q_{a_i,y} = c_{a_i,y}/k$. Com a distribuição estimada de probabilidade, calcula-se a entropia estimada $\widetilde{H}(A|Y = y)$ de cada um dos predecessores observados. Este cálculo de entropia é realizado utilizando uma adaptação da expressão 2.2 para levar em conta as probabilidade estimadas:

$$\widetilde{H} = -\sum_{a_i \in \mathcal{A}} q_{a_i,y} \cdot \log_2 (q_{a_i,y}). \tag{3.1}$$

Para calcular a entropia condicional, é preciso estimar a probabilidade $Pr[Y = y]$ de cada observação $Y = y$ ocorrer. Esta probabilidade é estimada a partir da razão q_y entre o número total de observações de y como predecessor e o número de mensagens totais:

$$q_y = \frac{\sum_{a_i \in \mathcal{A}} c_{a_i,y}}{(n-c-1)k}. \tag{3.2}$$

A partir das entropias individuais, pode-se calcular a entropia condicional estimada $\widetilde{H_c}$ pela modificação da expressão 2.5:

$$\widetilde{H_c} = \sum_y q_y \cdot \widetilde{H}(A|Y = y). \tag{3.3}$$

Acurácia das estimativas

A acurácia das estimativas [Bor05] depende da diferença entre as distribuições de probabilidade estimada empiricamente q e da probabilidade real p.

A estimativa de entropia é chamada de *estimador de máxima verossimilhança*. Este estimador tem distribuição normal, para uma média μ e uma variância σ^2 [Pan03]. A variância depende do número de amostras $k(n-c-1)$ e tem limite superior:

$$\sigma^2 \leq \frac{\log_2 m}{k(n-c-1)}, \tag{3.4}$$

tendendo a 0 com o crescimento de k, onde m é o número de posições não-nulas da distribuição q. A média apresenta uma polarização negativa b limitada:

$$-\log_2\left(1+\frac{m-1}{k(n-c-1)}\right) \leq b \leq 0, \tag{3.5}$$

que também tende a 0 com o crescimento de k [Pan03].

Para o cálculo da variância e da polarização da estimativa de entropia condicional, a seguinte identidade de entropia é necessária:

$$H(A|Y) = H(A,Y) - H(Y), \tag{3.6}$$

onde $H(A,Y)$ é a entropia da distribuição conjunta de probabilidade A,Y e pode ser estimada por

$$\widetilde{H}(A,Y) = \sum_{a_i \in \mathcal{A}, y \in \mathcal{Y}} q_{a_i,y} \log_2 (q_{a_i,y}). \tag{3.7}$$

Similarmente, pode-se estimar a entropia $H(Y)$ por:

$$\widetilde{H}(Y) = \sum_{y \in \mathcal{Y}} q_y \log_2 (q_y). \tag{3.8}$$

Sejam b_0 e b_1 os limites da polarização negativa para $\widetilde{H}(A,Y)$ e $\widetilde{H}(Y)$, respectivamente, e sejam σ_0^2 e σ_1^2 suas respectivas variâncias, todas calculadas a partir dos limites fornecidos anteriormente. Tanto b_0 e b_1, quanto σ_0^2 e σ_1^2, tendem a 0 quando $k \to \infty$. Para a entropia condicional estimada

$$\widetilde{H_c} = \widetilde{H}(A|Y) = \widetilde{H}(A,Y) - \widetilde{H}(Y), \tag{3.9}$$

o erro gerado pela polarização negativa encontra-se no intervalo aberto $(-b_0, b_1)$ e a variância é dada por [Pan03]:

$$\sigma_c \leq \sigma_0^2 + \sigma_1^2 + 2\sigma_0\sigma_1. \tag{3.10}$$

Como as fontes de erro da estimativa diminuem com o crescimento do número de amostras $k(n-c-1)$, conclui-se que as estimativas de entropia e entropia condicional são acuradas e podem ser confiadas como resultados da experimentação.

A simulação não reproduz eventos de entrada e saída de nós na rede nem latências de comunicação. Apesar disso, é suficiente para comparar as topologias estruturadas para comunicação anônima, considerando o adversário e o tipo de ataque já descritos.

Validação

A rede *Crowds* foi utilizada para validar o ambiente de simulação construído [Bor05]. A metodologia de validação consistiu em calcular analiticamente a entropia da rede *Crowds* (equação 2.2) e comparar o resultado com a entropia medida a partir de simulação.

Seja uma rede *Crowds* com n participantes e probabilidade de encaminhamento $0 < p_f < 1$. Quando um nó intermediário tenta determinar a origem de uma mensagem que encaminha, ele pode considerar um conjunto de anonimato de tamanho máximo $n-1$ (excluindo a si mesmo). Entretanto, a probabilidade de que o predecessor observado da mensagem seja a origem é igual a probabilidade de que nenhum outro nó tenha encaminhado a mensagem para o predecessor:

$$p_p = 1 - \frac{p_f(n-2)}{n}. \tag{3.11}$$

A probabilidade de qualquer outro nó ter originado a mensagem é p_f/n [RR98].

Este cálculo pode ser estendido para o caso onde c dos n participantes são nós controlados pelo adversário atuando em conluio, ou seja, podem excluir-se entre si de suas conclusões. Neste caso, a probabilidade do predecessor observado ser a origem é:

$$p_p = 1 - \frac{p_f(n-c-1)}{n}, \tag{3.12}$$

enquanto a probabilidade de qualquer outro nó ter originado a mensagem continua a mesma. Esta diferença entre a probabilidade do predecessor ser a origem e a probabilidade de qualquer outro nó ser a origem motiva o ataque de predecessor: ao observar vários predecessores para um volume considerável de mensagens interceptadas, o adversário deve detectar uma distorção na distribuição de probabilidade, que fornece informação para identificação da verdadeira origem. A entropia da rede *Crowds* é calculada por [DSCP02]:

$$H = -\left(1 - \frac{p_f(n-c-1)}{n}\right)\log_2\left(1 - \frac{p_f(n-c-1)}{n}\right) - \left(\frac{n-c-1}{n}\frac{p_f}{n}\right)\log_2\left(\frac{p_f}{n}\right) \tag{3.13}$$

Por exemplo, uma rede *Crowds* com 100 nós, dos quais 10 são controlados por adversário, e com probabilidade de encaminhamento $p_f = 0.75$, tem entropia $H \approx 5.24$ *bits*. A entropia teórica ótima para um sistema assim corresponde à distribuição uniforme dos eventos sobre os

$(n-c)$ nós legítimos: $\log_2(n-c) = \log_2 90 \approx 6.49$ *bits* de informação. O sistema revela pouco mais de 1 *bit* de informação para o adversário e o tamanho efetivo do conjunto de anonimato passa de 90 para $2^{5.24} \approx 38$ nós.

Deve-se ainda considerar o volume de mensagens que não são interceptadas por nós controlados pelo adversário. A probabilidade de uma mensagem atingir o destino passando apenas por nós honestos (não-controlados pelo adversário) é:

$$p' = \frac{n-c}{n}(1-p_f) \cdot \sum_{i=0}^{\infty} \frac{n-c}{n}(p_f)^i = 1 - \frac{c}{n - p_f(n-c)}. \tag{3.14}$$

Neste caso, a entropia dos eventos é $H' = \log_2(n-c)$, já que o adversário não captura nenhuma mensagem e não obtém informação. A entropia condicional, considerando os dois cenários é:

$$H_c = (1-p')H + p'H' = \frac{c}{n - p_f(n-c)}H + \left(1 - \frac{c}{n - p_f(n-c)}\right)\log_2(n-c). \tag{3.15}$$

Usando esta expressão, o sistema com as mesmas características anteriores ($n = 100$, $c = 10$ e $p_f = 0.75$) tem entropia condicional $H_c = 6.11$ *bits*. O tamanho efetivo do conjunto de anonimato passa a ser $2^{6.11} \approx 69$ nós.

A validação do experimento consiste na comparação entre a entropia condicional estimada por simulação e a entropia condicional calculada analiticamente. A Figura 3.13 apresenta a entropia teórica e a estimada por simulação. Os intervalos de confiança de 95% da estimativa de entropia, considerando a polarização negativa, são também apresentados. A partir deste gráfico, percebe-se que a entropia estimada converge para a entropia teórica e o erro diminui com o crescimento de amostras. A interpretação do gráfico ainda sugere que um número de amostras adequado para as simulações posteriores é da ordem de 1 milhão.

3.3.5 Resultados experimentais

As topologias descritas anteriormente foram submetidas ao experimento de simulação, que permitiu a coleta das observações realizadas pelo adversário. Os experimentos simularam redes com 256 nós comunicando-se anonimamente, sendo 10% destes nós controlados pelo adversário. Nos gráficos subseqüentes, cada ponto corresponde a uma média aritmética de 50 simulações sucessivas e cada simulação reproduz 1 milhão de eventos de comunicação. O espaço de endereçamento utilizado foi o anel de inteiros $\mathbb{Z}_{2^{160}}$.

Segundo os critérios de seleção adotados, as topologias foram avaliadas de acordo com a métrica de entropia, a resistência da topologia a ataques de negação de serviço e o compromisso entre desempenho e qualidade de anonimato.

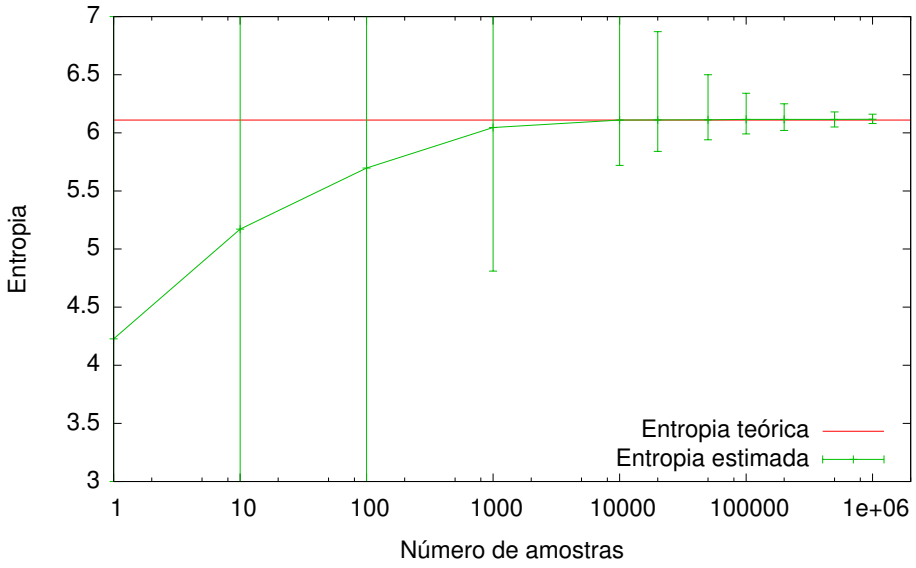


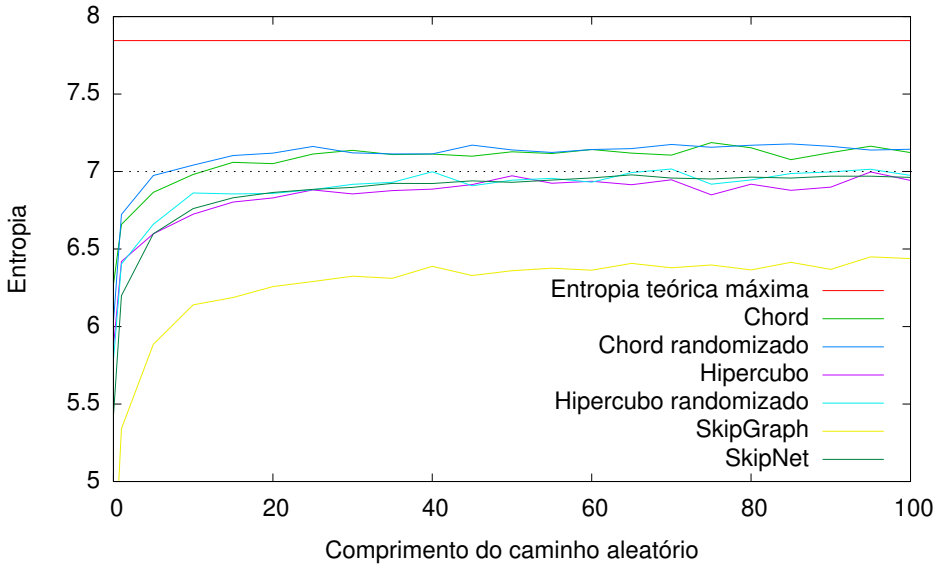
Figura 3.13: Estimativa de entropia condicional na rede *Crowds*.

Entropia

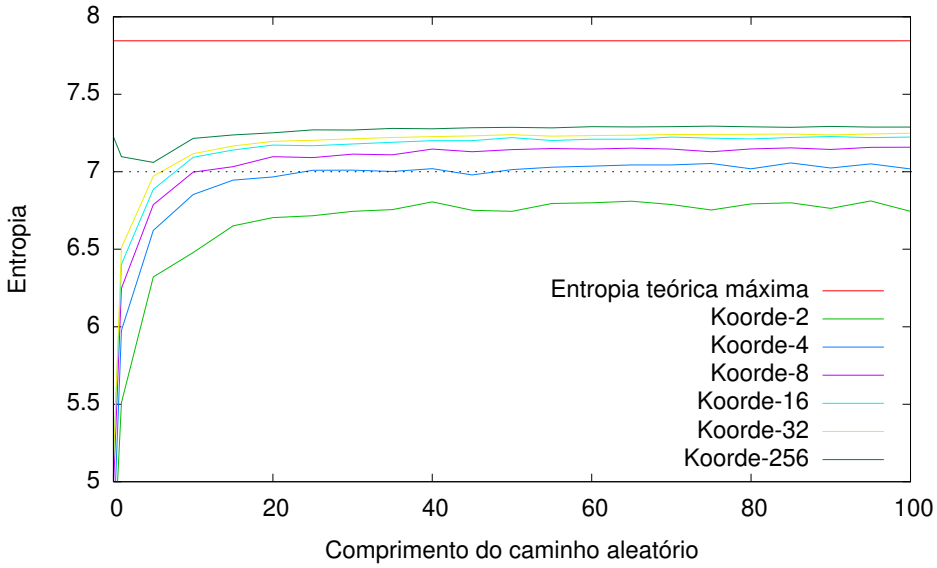
Os resultados experimentais da métrica de entropia condicional encontram-se na Figura 3.14. As topologias estruturadas foram divididas em dois conjuntos para facilitar a visualização dos resultados. O valor rotulado como entropia teórica máxima é o nível $\log_2(n-c)$. O nível 7 de entropia também foi acrescentado para facilitar a comparação entre as curvas dos dois gráficos.

A partir dos gráficos, pode-se concluir que:

- O aumento do grau nas topologias *Koorde* colabora para o aumento de entropia;
- Os resultados da rede *Koorde-256* representam uma espécie de limite prático para a topologia *Koorde*. Isto se deve ao fato de que em uma rede *Koorde* com 256 nós e grau 256, cada um dos nós conecta-se a todos os outros. Este é um cenário ideal, do ponto de vista prático;
- A entropia nas topologias *Koorde-32* e *Koorde-16* é superior à entropia de qualquer topologia presente no primeiro gráfico;
- A topologia *Koorde-16* é superior às topologias *Koorde-32* e *Koorde-256*, considerando uma razão entre custo e benefício, já que as três possuem entropias bastante próximas e a primeira tem uma sobrecarga de manutenção muito inferior às demais (menor número de conexões para manter);



(a) Primeiro conjunto de topologias.



(b) Segundo conjunto de topologias.

Figura 3.14: Resultados experimentais de entropia condicional.

- As versões randomizadas das topologias fornecem ganho em relação às topologias originais. Isto mostra que um certo grau de aleatoriedade presente na topologia pode colaborar com a entropia;

- As topologias com maior aleatoriedade e menor regularidade, *SkipGraph* e *SkipNet*, obtiveram alguns dos piores resultados. Pode-se inferir que o aumento de aleatoriedade, apesar de contribuir com o ganho de entropia, deve vir combinado a um certo grau de regularidade para ser útil. A topologia *SkipNet* obteve resultado superior à topologia *SkipGraph*, por causa da organização orientada a anéis, que garante aos nós um maior número de possibilidades de roteamento durante o caminho aleatório; e

- A seqüencia *Koorde*, *Chord* e Hipercubo, em ordem decrescente de entropia, é idêntica à seqüência obtida por [Bor05], em ordem decrescente de capacidade de mistura. Isto confirma experimentalmente a hipótese intuitiva de que a capacidade de mistura é uma grandeza intimamente relacionada à entropia.

Resistência à negação de serviço

Considerando que os nós controlados pelo adversário descartam todas as mensagens que deveriam rotear, as mesmas simulações são utilizadas para examinar a eficácia deste ataque distribuído de negação de serviço no funcionamento da rede. A resistência à negação de serviço foi medida a partir do percentual de mensagens que chegaram com sucesso em seus destinos. Os resultados experimentais para o percentual de chegada encontram-se na Figura 3.15.

As conclusões do experimento são:

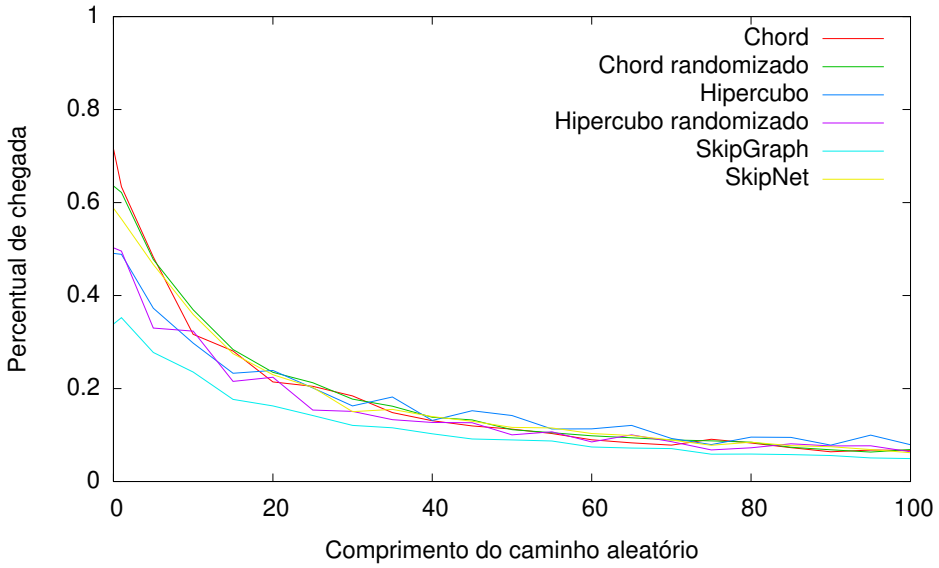
- A topologia *Koorde-256* apresentou resultado extremamente superior às demais topologias, como esperado. Percebe-se que a taxa de chegadas das mensagens, para um caminho aleatório de comprimento mínimo, chega a 90% – justamente a proporção de nós íntegros em relação aos nós totais. A razão para este fenômeno é que o percorrimento do caminho aleatório, apesar de aumentar a entropia, eleva as chances de captura e descarte por um nó malicioso;

- O aumento de grau nas topologias *Koorde* não só colabora com o aumento de entropia, como verificado anteriormente, mas também aumenta a resistência a ataques de negação de serviço executados por um adversário interno;

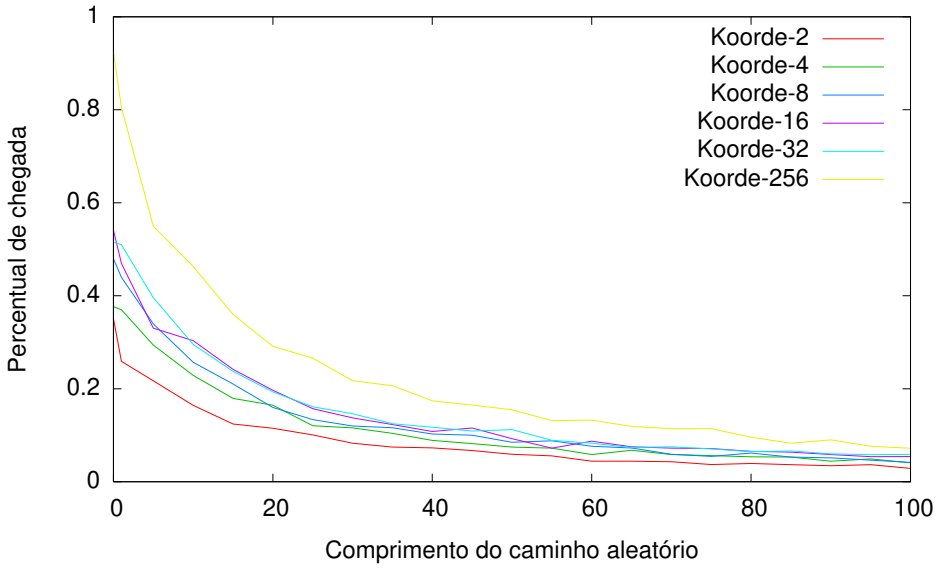
- A randomização das topologias não provoca ganhos significativos na taxa de chegada de mensagens;

- Apesar das topologias *Koorde* apresentarem resultado inferior às demais topologias nesta categoria, a diferença não é muito significativa; e

- Existe uma relação entre entropia e taxa de chegada: topologias que distribuem melhor as mensagens têm chance maior de entregá-las com sucesso. Várias das topologias que obtiveram as maiores entropias, também apresentaram as melhores taxas de chegada.



(a) Primeiro conjunto de estruturas.



(b) Segundo conjunto de estruturas.

Figura 3.15: Resultados experimentais de resistência à negação de serviço.

Desempenho

O critério final de avaliação foi a medida do comprimento das rotas percorridas durante a segunda fase de roteamento. A primeira fase de roteamento é ignorada, porque está condicionada à uma probabilidade de encaminhamento idêntica para todas as topologias. A seleção de uma topologia com rotas curtas e que privilegia a métrica de entropia é decisiva, por representar um compromisso ótimo entre desempenho e anonimato. Os resultados experimentais de comprimento de rotas encontram-se na Figura 3.16.

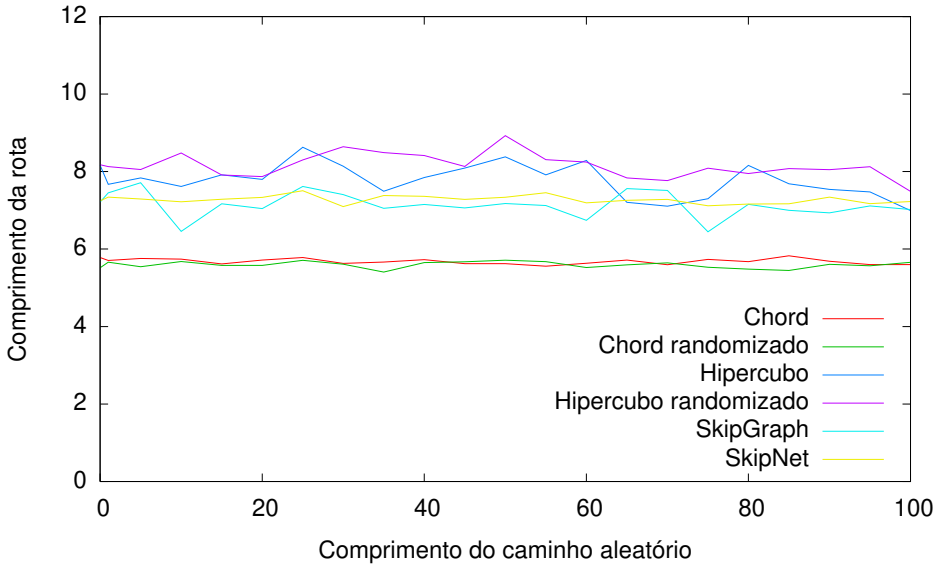
A partir da interpretação dos gráficos, conclui-se que:

- O aumento do grau na topologia *Koorde* também diminui o comprimento das rotas, incrementando diretamente o desempenho da topologia;
- A randomização das topologias não provoca ganhos significativos de desempenho;
- As topologias com melhor entropia também apresentaram as rotas mais curtas; e
- A topologia *Koorde-32* apresenta desempenho favorável em relação à topologia *Koorde-16*, mas a diferença no tamanho médio das rotas não chega a alcançar 1 nó.

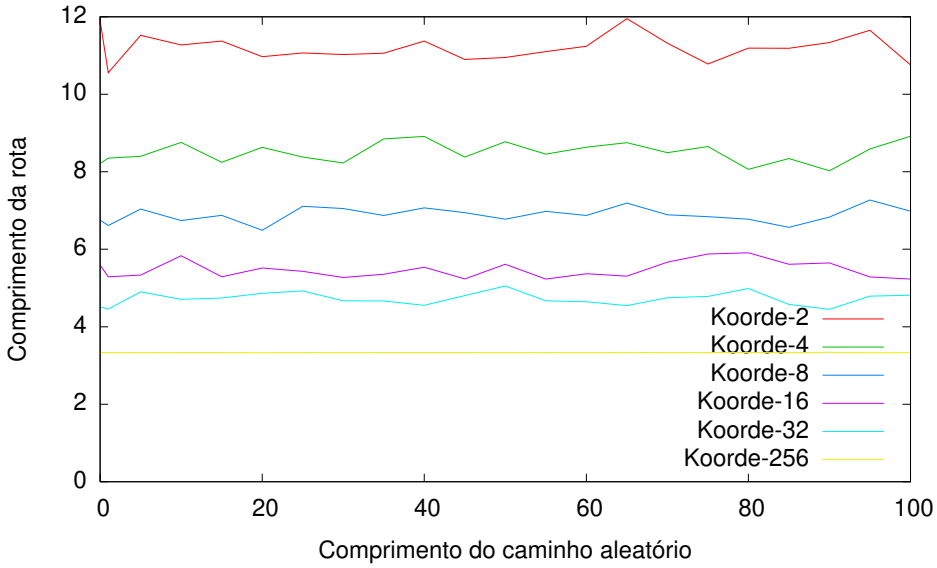
3.3.6 Seleção da topologia

Apesar da topologia *Koorde-256* apresentar os melhores resultados em todas as categorias, a sobrecarga de manutenção da rede é muito elevada. O número de conexões simultâneas que esta estrutura exige para funcionamento correto é bem superior a 256 conexões por nó, somando-se as conexões que o nó inicia com as conexões iniciadas pelos demais que terminam no nó. Avaliar a sobrecarga de manutenção é importante, visto que o percentual de banda útil para as aplicações anonimizadas depende diretamente da complexidade de manutenção da rede.

Considerando todas as conclusões apresentadas na seção anterior, para os critérios de entropia, desempenho e resistência a ataques de negação de serviço, a topologia que apresentou os melhores resultados foi a topologia *Koorde* com grau 16. Tendo entropia muito próxima das variantes de grau 32 e 256, bom desempenho e resistência razoável a ataques de negação de serviço, a topologia *Koorde-16* permite a construção de uma rede anônima eficiente e com boa qualidade de anonimato que exige baixa sobrecarga de manutenção. Apesar da topologia *Koorde-32* apresentar rotas um pouco menores, a baixa sobrecarga de manutenção da topologia *Koorde-16* privilegia a latência de transmissão em uma rede funcional, e deve compensar o maior comprimento das rotas com transmissões mais rápidas. Além disso, o baixo número de conexões que cada nó deve manter fornece maior flexibilidade para a modificação de algumas das características da estrutura. Isto é decisivo para o aprimoramento da topologia, realizado no restante deste capítulo.



(a) Primeiro conjunto de estruturas



(b) Segundo conjunto de estruturas

Figura 3.16: Resultados experimentais de desempenho.

3.4 Aprimoramento da topologia

Apesar de já oferecer entropia elevada, bom desempenho e resistência razoável à negação de serviço, a topologia *Koorde-16* ainda fornece amplo espaço para aperfeiçoamento. Nesta seção, diversas estratégias são sugeridas e avaliadas experimentalmente para aprimorar ainda mais as vantagens desta topologia. O objetivo é aproximá-la dos ótimos resultados da topologia *Koorde-256*, mantendo uma relação aceitável entre custo e benefício. O interesse por trás dos aprimoramentos é obter o máximo de entropia possível com o menor comprimento esperado de caminho aleatório.

3.4.1 Probabilidade de encaminhamento

A atribuição de uma probabilidade de encaminhamento adequada é vital para se construir uma rede de anonimização que apresente bom desempenho. Quanto maior a probabilidade de encaminhamento, maior o comprimento dos caminhos aleatórios e, conseqüentemente, maior a latência de transmissão. Em uma topologia *Koorde* com n nós de grau d, os caminhos aleatórios devem ter comprimento esperado de $\log_d n$ idealmente, para que alcancem um identificador aleatório [KK03].

Assim como na rede *Crowds*, a probabilidade de uma mensagem percorrer um caminho aleatório de comprimento l com probabilidade de encaminhamento p_f é dada por

$$p_l = {p_f}^l \cdot (1 - p_f). \tag{3.16}$$

O comprimento esperado l_e de um caminho aleatório associado a uma probabilidade de encaminhamento p_f é

$$l_e = \sum_{i=0}^{\infty} (i+1){p_f}^i(1 - p_f) = \frac{1}{1 - p_f}. \tag{3.17}$$

Inversamente, para se percorrer um caminho aleatório de comprimento aproximado l_e, cada nó deve encaminhar a mensagem para um de seus vizinhos com probabilidade $p_f = \frac{l_e - 1}{l_e}$ e iniciar a segunda fase do roteamento com probabilidade $\frac{1}{l_e}$ [Bor05]. A Figura 3.17 ilustra a distribuição de probabilidade do comprimento de um caminho aleatório para probabilidades de encaminhamento 0,6, 0,75 e 0,9.

É possível escolher a probabilidade de encaminhamento a partir da magnitude da rede. Para a escolha particular da topologia *Koorde-16* com suporte a uma rede anônima composta por 2^{20} nós, um nó com identificador x atinge um identificador aleatório após um caminho aleatório de comprimento esperado igual a $l_e = 5$. A probabilidade de encaminhamento condicionada ao comprimento do caminho aleatório é $p_f = \frac{5-1}{5} = 0,8$. Entretanto, considerando-se que o não-determinismo proveniente do caráter esparso da rede, pode inserir pequenos desequilíbrios

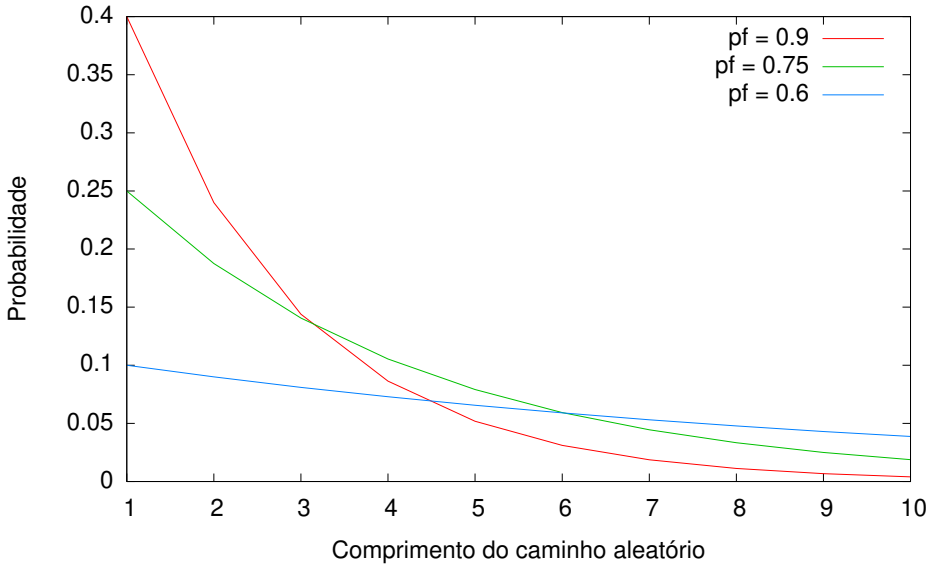


Figura 3.17: Distribuição de probabilidade do comprimento de um caminho aleatório.

em sua estrutura, utiliza-se um comprimento esperado do caminho aleatório maior do que o comprimento estritamente necessário. Em compatibilidade com o valor sugerido por [Bor05], utiliza-se uma probabilidade de encaminhamento correspondente ao dobro do comprimento aleatório necessário para se atingir um ponto independente. Logo: $p_f = \frac{(2l_e - 1)}{2l_e} = 0,9$.

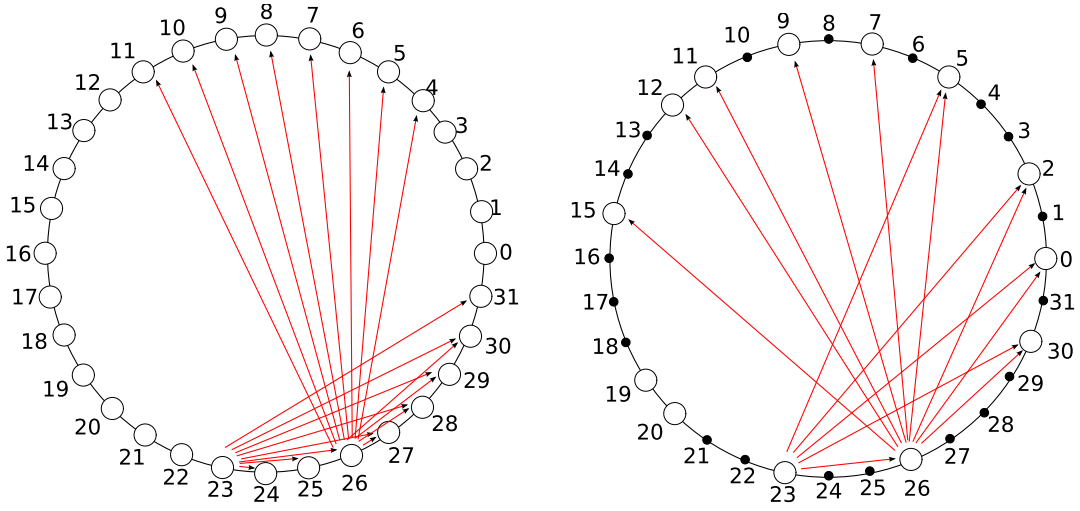
3.4.2 Tolerância a falhas

Tolerância a falhas refere-se à capacidade do sistema em continuar sua operação após a falha de uma porção significativa do sistema.

A formulação original da topologia *Koorde* sugere que, independente do grau, deve-se ter conexões adicionais para que a estrutura forneça tolerância a falhas. Um nó da estrutura *Koorde* com identificador x conecta-se a dois conjuntos de nós em locais diferentes do espaço de endereçamento: o seu sucessor e d nós com identificadores próximos a $d \cdot x \pmod{2^b}$. A conexão com o sucessor é extremamente importante, já que a coesão da estrutura em face da falha simultânea de uma porção significativa dos nós na rede depende diretamente da integridade das conexões entre os nós e seus sucessores. Assim, ao invés de manter uma única conexão com o seu sucessor, cada nó deve manter conexões com os seus d sucessores. Adicionalmente, cada nó deve ainda manter conexões com os d predecessores do nó $d \cdot x \pmod{2^b}$ [KK03]. A topologia modificada e tolerante a falhas é denominada *Koorde-t*.

A Figura 3.18 ilustra a modificação quando efetuada nas configurações apresentadas na

Figura 3.12. As conexões dos nós 23 e 26, representadas por setas em vermelho, incluem as conexões necessárias para a topologia fornecer tolerância a falhas.



(a) Exemplo de topologia *Koorde-t-4* completa. (b) Exemplo de topologia *Koorde-t-4* esparsa.

Figura 3.18: Configurações ideal e prática de uma topologia *Koorde-t* de grau 4.

Entretanto, é importante verificar se as novas conexões, quando utilizadas na seleção aleatória de vizinhos da primeira fase do roteamento randomizado, exercem alguma influência na entropia da estrutura modificada. Para examinar esta hipótese, um novo experimento de simulação foi realizado para a estrutura *Koorde-t*. As mesmas características de simulação dos experimentos anteriores foram conservadas, alterando-se apenas o tamanho da rede para $n = 1024$ e o número de nós comprometidos para $c = 102$. A entropia teórica máxima deste sistema é $\log_2(n-c) \approx 9.85$. Os experimentos subseqüentes simularão redes com estas exatas características, visto que o aumento do tamanho da rede eleva a precisão e o rigor do experimento.

Foram testadas duas formas distintas de combinar as conexões para tolerância a falhas com as conexões convencionais da topologia *Koorde*. A primeira consiste na seleção aleatória de vizinhos dentre as 48 possibilidades totais (32 conexões para tolerância a falhas e 16 convencionais). A segunda consiste na seleção aleatória entre uma conexão para um nó próximo ao sucessor e 16 conexões convencionais, totalizando 17 possibilidades. Quando a conexão para um nó próximo ao sucessor é selecionada, um sorteio uniforme adicional entre as 32 conexões para tolerância a falhas escolhe o vizinho. Foi verificado experimentalmente que a primeira estratégia diminui a entropia da rede, porque as conexões para nós próximos ao sucessor tem menor entropia e são tomadas com probabilidade muito alta na seleção aleatória de vizinho. A Figura 3.19 mostra que a segunda estratégia aumenta a entropia da rede, especialmente para a

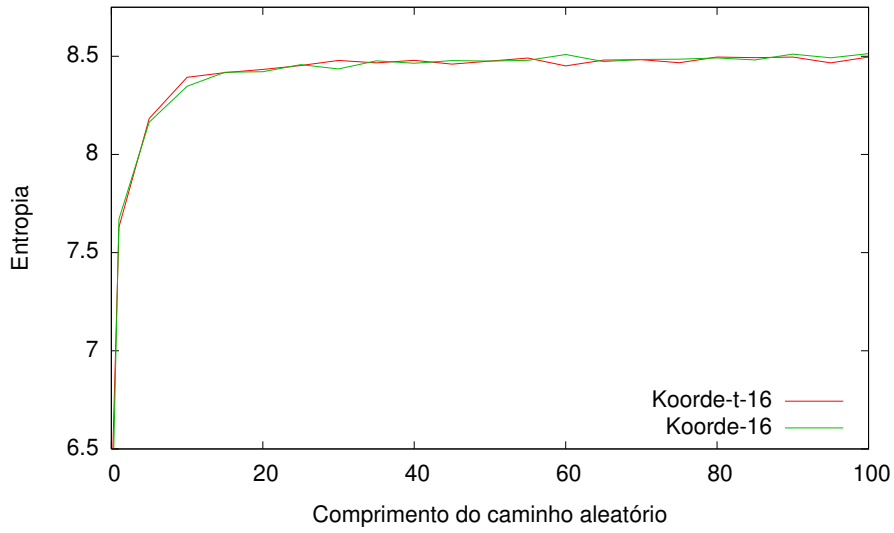


Figura 3.19: Resultados experimentais de entropia condicional da topologia *Koorde-t-16*.

probabilidade de encaminhamento $p_f = 0,9$, apesar do ganho ser pouco significativo.

3.4.3 Conexões adiantadas

Uma técnica recentemente proposta na área de pesquisa em sistemas *peer-to-peer*, para otimizar decisões de roteamento e a eficiência de sistemas estruturados [NW04], consiste em fornecer, para cada nó da rede, conhecimento privilegiado que antes era de exclusividade dos seus vizinhos. O algoritmo de roteamento determinístico, executado em cada nó que participa do roteamento, é adaptado para considerar informação a respeito da vizinhança dos vizinhos do nó. A adaptação altera a função de distância para que ela escolha nós cujos vizinhos são mais próximos do nó de destino, possibilitando a escolha antecipada de rotas mais curtas e o ganho conseqüente em desempenho [NW04].

Considerando esta técnica, o fornecimento de conhecimento privilegiado a respeito da topologia também foi verificado experimentalmente, em busca de ganhos na métrica de entropia. Cada nó da rede recebe informação a respeito de um vizinho de cada um dos seus vizinhos imediatos na topologia. A seleção deste vizinho é realizada aleatoriamente. Assim, cada nó recebe d conexões privilegiadas adicionais, chamadas *conexões adiantadas*, que passam a participar da seleção de vizinhos do caminho aleatório. A motivação desta modificação é que, encaminhando uma mensagem para um vizinho de um vizinho na topologia *Koorde*, o roteamento insere dois novos dígitos à direita do identificador em um único passo, acelerando a taxa em que se atinge um identificador aleatório.

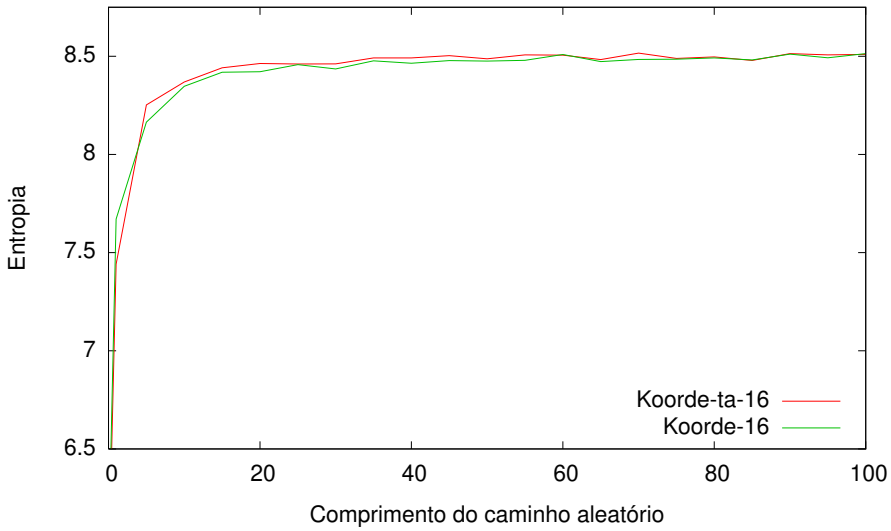


Figura 3.20: Resultados experimentais de entropia condicional da topologia *Koorde-ta-16*.

A nova topologia, que acumula as modificações de tolerância a falhas e adiantamento de conexões, é denominada *Koorde-ta*. Nesta topologia, cada nó estabelece 64 conexões – 32 para tolerância a falhas, 16 convencionais e 16 adiantadas – e, durante a seleção de vizinho para um caminho aleatório, são consideradas 33 possibilidades – uma para nós próximos ao sucessor, 16 para conexões convencionais e 16 para conexões adiantadas. A vantagem das conexões adiantadas é que elas não são utilizadas durante a segunda fase (fase determinística) do roteamento randomizado, pois a topologia *Koorde* não permite flexibilidade nas decisões de roteamento [KK03]. Logo, as conexões adiantadas não precisam ser atualizadas com freqüência e não trazem sobrecarga significativa de manutenção.

A Figura 3.20 apresenta os resultados experimentais da métrica de entropia para a topologia *Koorde-ta-16*. Comparando-se o gráfico com a Figura 3.19, pode-se observar um ganho de entropia em relação à topologia *Koorde-t-16*.

3.4.4 Canais múltiplos

A utilização de canais múltiplos de resposta foi sugerida na Seção 3.2.2 para descentralizar o ponto de entrada e obter confiabilidade. Entretanto, esta possibilidade não foi explorada na experimentação, que considerou apenas um único destino por simulação. A hipótese de que o estabelecimento de canais múltiplos colabora com a métrica de entropia foi então testada experimentalmente. Em cada evento de comunicação, o destino é selecionado aleatoriamente entre os pontos de entrada e o emissor tem múltiplas opções de destino para o envio de cada

mensagem. O número de pontos de entrada foi fixado em d, para equivalência com a natureza da topologia. A nova topologia, com as modificações prévias acumuladas, é chamada *Koorde-tac*.

Nova simulação

O novo procedimento de simulação é composto pelas seguintes etapas:

1. Seleção de d nós não-controlados por adversário para servir como pontos de entrada para um nó a_d. Por simplificação, os canais de resposta são assumidos como íntegros e livres da presença de nós controlados pelo adversário. Esta simplificação não tem impacto na métrica de entropia, que, no caso, avalia estritamente a qualidade do anonimato de envio;

2. Execução de eventos de comunicação consecutivos que simulam a transmissão de k mensagens de cada um dos participantes $a_i \in \mathcal{A}$ para o nó a_d. Os nós controlados pelo adversário e o destino a_d não participam da simulação como emissores de mensagem. Cada uma das mensagens enviadas tem como destino real um dos pontos de entrada do nó a_d, selecionado aleatoriamente entre os d pontos de entrada disponíveis;

3. Gravação do predecessor observado para cada mensagem interceptada por nó malicioso. Os nós controlados pelo adversário descartam as mensagens capturadas imediatamente após a gravação da observação correspondente; e

4. Cálculo da métrica de entropia condicional do sistema, considerando-se as observações do adversário.

Novos resultados

A medida de entropia para a variante *Koorde-tac* é apresentada na Figura 3.21. Pode-se observar um ganho mais consistente de entropia da topologia *Koorde-tac-16* em relação à topologia original *Koorde-16*, com a descentralização do ponto de entrada. A topologia *Koorde-tac-16* também supera em entropia a topologia *Koorde-32* para caminhos aleatórios de comprimento superior a 10. O comprimento 10 é justamente o comprimento esperado dos caminhos aleatórios para a probabilidade de encaminhamento adotada $p_f = 0,9$. O projeto da topologia *Koorde-tac-16* mostra que a escolha cuidadosa dos recursos e da política de roteamento permite a construção de topologias com propriedades úteis e compromisso ótimo entre qualidade de anonimato e desempenho.

A análise detalhada do gráfico indica ainda que a topologia *Koorde-tac-16*, para caminhos aleatórios de comprimento 10, fornece a mesma entropia da topologia *Koorde-16* para caminhos aleatórios de comprimento 20. Ou seja, a entropia *Koorde-tac-16* apresenta uma relação entre desempenho e qualidade de anonimato duas vezes melhor que a topologia original, para caminhos aleatórios de comprimento esperado 10. Considerando apenas os caminhos aleatórios de

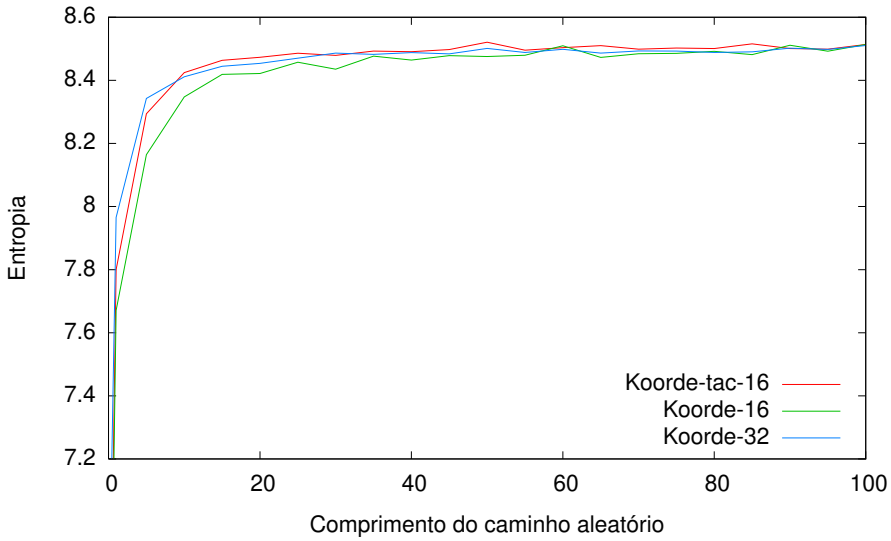


Figura 3.21: Resultados experimentais de entropia condicional da topologia *Koorde-tac-16*.

comprimento 10, temos um conjunto efetivo de anonimato de tamanho $2^{8.34} \approx 325$ para a topologia *Koorde-16* e um conjunto efetivo de anonimato de tamanho $2^{8.42} \approx 344$ para a topologia *Koorde-tac*. Como a entropia é uma grandeza logarítmica, um ganho de quase 1% na entropia condicional provocou um aumento de 6% no tamanho do conjunto efetivo de anonimato. Espera-se que estes ganhos sejam bastante amplificados em redes mais populosas.

3.5 Resumo

Neste capítulo, foi apresentada uma política de roteamento eficiente para comunicação anônima em sistema estruturados. Esta nova política fornece anonimatos de envio, resposta e de par comunicante. O anonimato de envio é resultante da utilização de roteamento randomizado, o anonimato de resposta é obtido a partir de canais de resposta independentes e o anonimato de par comunicante é proveniente da camada dupla de cifração.

Por meio de verificação empírica, a topologia *Koorde* de grau 16 foi selecionada. Aprimoramentos significativos foram ainda propostos e inseridos na topologia original, incluindo tolerância a falhas, conexões adiantadas e pontos de entrada múltiplos. Estes aprimoramentos permitiram a produção da topologia *Koorde-tac-16*, tolerante a falhas e favorável na métrica de entropia, o que indica ganho na qualidade de anonimato em relação à topologia original.

Capítulo 4

Criptografia de Chave Pública Sem Certificados

O advento da criptografia de chave pública [DH76] revolucionou a forma de se construir sistemas criptográficos e possibilitou a integração de forma definitiva entre teoria criptográfica e implementação em aplicações reais. Particularmente, trouxe a possibilidade de se estabelecer serviços criptográficos como sigilo e assinatura não-repudiável em ambientes em que não existe qualquer relação de confiança entre os envolvidos ou canal seguro para distribuição de chaves. O antigo problema da distribuição de chaves converteu-se na dificuldade de obtenção de uma chave pública ausência de uma entidade. Como solução para este novo problema, um repositório público foi inicialmente proposto como ponto de distribuição de chaves [DH76], mas é fácil perceber que não há como um repositório deste tipo fornecer autenticidade e que, sem garantias de autenticidade, é possível para um atacante substituir uma chave pública armazenada e posteriormente participar em protocolos personificando entidades legítimas. A autenticação mútua das chaves é crucial para que tais intervenções por parte de agentes não-autorizados possam ser detectadas e evitadas.

Convencionalmente, a verificação da autenticidade de chaves públicas é reduzida à validação de *certificados de titularidade* [Koh78]. O papel de um certificado é mapear uma chave pública a uma entidade, utilizando-se uma assinatura por uma terceira parte confiada, a *autoridade certificadora*, como atestado de integridade. As autoridades certificadoras costumam se organizar em uma estrutura hierárquica para descentralização dos serviços comumente disponibilizados como: requisição, emissão, validação e revogação de certificados. A união entre padrões de certificado, autoridades certificadoras e procedimentos para gerência de certificados formam uma *Infra-estrutura de Chaves Públicas* (*Public Key Infrastructure* – PKI) [Gut02]. O funcionamento básico de uma infra-estrutura de chaves públicas está representado na Figura 4.1.

Uma infra-estrutura de chaves públicas representa relações que existem entre entidades do mundo real, como filiação a organizações e delegação de serviços de uma organização para

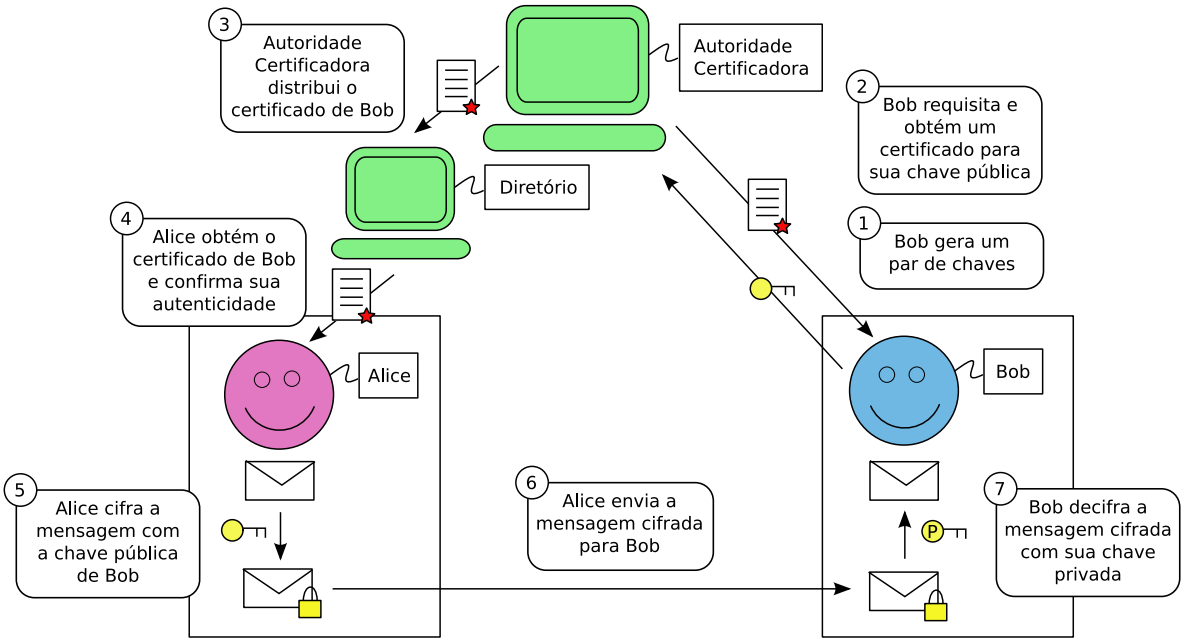


Figura 4.1: Funcionamento básico de uma PKI tradicional.

outra. Os tipos de relações presentes são diversos, o que traz sérios problemas para o projeto de infra-estruturas que sejam tanto genéricas o suficiente para representar qualquer tipo de relação, quanto simples o suficiente para terem ampla aceitação, possibilidade de padronização e eficiência. Operações de validação de certificados e revogação de chaves públicas tendem a representar situações extremas [Gut02]. A validação exige a determinação de uma cadeia de assinaturas que garante a autenticidade de uma chave pública, bem como a verificação computacionalmente custosa de cada uma destas assinaturas. Técnicas como certificação cruzada criam caminhos múltiplos nas árvores de validação de certificados e dificultam ainda mais a determinação dos conjuntos de assinaturas para verificação, conferindo características exponenciais para um problema que deveria ter complexidade linear. A revogação, por sua vez, traz problemas de distribuição da informação de revogação e demanda agilidade para que o tempo entre a solicitação e a revogação efetiva de um certificado seja mínimo. Isto é claramente importante para que seja minimizado o tempo em que uma chave revogada continue em uso. Pode-se observar que os maiores problemas que afetam o projeto e o emprego de uma infra-estrutura de chaves públicas são relacionados à escalabilidade.

Para atenuar alguns dos problemas conhecidos de infra-estruturas tradicionais de chaves públicas – o armazenamento, distribuição e verificação de certificados – alternativas que implementam certificação implícita vêm sendo propostas. Destacam-se a Criptografia Baseada em Identidades (*Identity-based Public Key Cryptography* – ID-PKC) [Sha84] e suas derivadas

[Gir91, CCV04], entre elas a Criptografia de Chave Pública Sem Certificados (*Certificateless Public Key Cryptography* – CL-PKC) [ARP03].

4.1 ID-PKC

Sistemas baseados em identidades foram concebidos por Shamir [Sha84] em 1984, mas suas primeiras realizações funcionais para cifração só foram apresentadas em 2001, por Cocks [Coc01] utilizando resíduos quadráticos; e por Boneh e Franklin[BF01] a partir de emparelhamentos bilineares sobre curvas elípticas. A motivação original para sistemas baseados em identidade era aproveitar a autenticidade de informação publicamente conhecida para simplificar a autenticação de chaves públicas. O objetivo era desenvolver primitivas criptográficas que pudessem utilizar identificadores arbitrários como chave pública, quando a verificação da autenticidade da ligação entre o identificador e a entidade correspondente é trivial.

Uma aplicação imediata para primitivas baseadas em identidade são sistemas de correio eletrônico, como apresentado na Figura 4.2. Quando uma entidade *A* deseja enviar uma mensagem para uma entidade *B* que possui o endereço *bob@mail.com*, basta que a entidade *A* cifre sua mensagem utilizando a cadeia de caracteres *bob@mail.com* como chave. Não há necessidade da entidade *A* obter ou verificar a autenticidade da chave pública de *B*. Ao receber a mensagem cifrada, a entidade *B* recorre a uma terceira parte confiada chamada *Gerador de Chaves Privadas* (*Private Key Generator* – PKG). A entidade *B* autentica-se com o PKG e obtém sua chave privada, podendo então decifrar a mensagem recebida. Note que, ao contrário da infra-estrutura atual, a entidade *A* pode enviar mensagens cifradas para a entidade *B* mesmo que *B* ainda não possua um certificado de chave pública. A certificação é implícita: a entidade *B* só poderá decifrar a mensagem caso possua uma chave privada correta emitida pelo PKG (com o efeito de certificá-la).

Generalizando as técnicas apresentadas para o sistema de correio eletrônico, define-se um modelo ID-PKC a partir de seis algoritmos [BF01]:

Inicializar. Recebe um parâmetro de segurança k e retorna os parâmetros de sistema params e a chave mestre s. Os parâmetros do sistema incluem uma descrições do espaço de mensagens $\mathcal{M}$, do espaço de criptogramas $\mathcal{C}$ e do espaço de assinaturas $\mathcal{S}$. Seguindo a prática comum, os parâmetros de sistema são conhecidos publicamente, enquanto o segredo s é mantido em sigilo pelo PKG.

Extrair. Recebe como entrada params, s e um identificador arbitrário $\mathsf{ID} \in \{0,1\}^*$, e retorna uma chave privada d. Como descrito anteriormente, o identificador ID é utilizado como chave pública e d é a chave privada correspondente. Resumidamente, o algoritmo extrai a chave privada de uma chave pública.

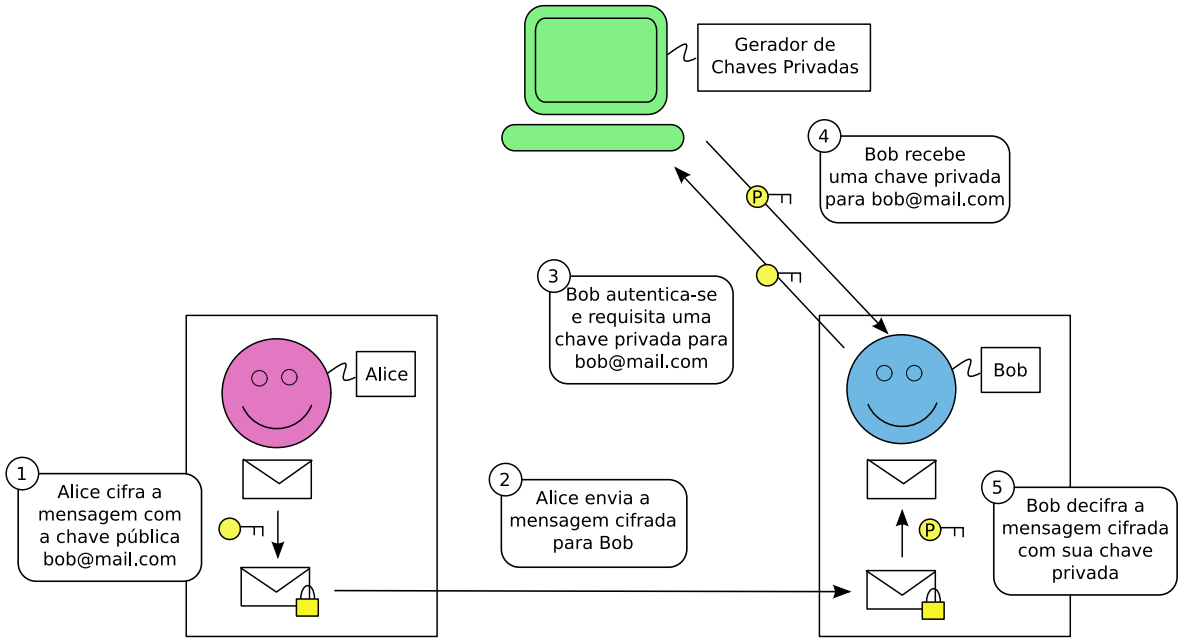


Figura 4.2: Sistema de Criptografia Baseada em Identidades.

Cifrar. recebe como entrada params, ID, $M \in \mathcal{M}$ e retorna um criptograma $C \in \mathcal{C}$.

Decifrar. recebe como entrada params, $C \in \mathcal{C}$, uma chave privada d e retorna $M \in \mathcal{M}$.

Assinar. recebe como entrada params, uma mensagem $M \in \mathcal{M}$, a chave privada d e produz uma assinatura $S \in \mathcal{S}$ para M.

Verificar. recebe como entrada params, a assinatura $S \in \mathcal{S}$ da mensagem $M \in \mathcal{M}$ para uma identidade ID e retorna verdadeiro ou falso, dependendo se a assinatura é aceitável ou não.

Os algoritmos devem satisfazer condições básicas de consistência:

$$\forall M \in \mathcal{M} : \mathsf{Decifrar}(\mathsf{params}, C = \mathsf{Cifrar}(\mathsf{params}, \mathsf{ID}, M), d) = M$$

$$\forall M \in \mathcal{M} : \mathsf{Verificar}(\mathsf{params}, S = \mathsf{Assinar}(\mathsf{params}, d, M), \mathsf{ID}, M) = \mathsf{verdadeiro}$$

A possibilidade de se produzir chaves públicas a partir de aspectos de identidade elimina a necessidade de certificados e resolve alguns dos problemas associados, mas duas conseqüências negativas são inseridas: a dependência na geração de chaves privadas introduz custódia inerente

em sistemas baseados em identidade (o PKG conhece a chave privada das entidades participantes); e o transporte de chaves privadas entre PKG e clientes exige um canal seguro de comunicação, o que dificulta a distribuição de chaves. É importante observar também que ainda há a necessidade de verificação da autenticidade dos parâmetros de sistema, para verificar que os parâmetros do PKG legítimo estejam sendo utilizados. Vantagens que compensam em parte estas limitações aparecem quando consideram-se a simplicidade trazida para o controle de expiração de chaves (a data de validade da chave pode ser concatenada à chave pública) e para a delegação de privilégios (o papel que cada par de chaves exerce pode também ser codificado diretamente no identificador fornecido ao PKG). Entretanto, a revogação de chaves é difícil, visto que revogar a identidade do detentor do par de chaves pode trazer complicações.

Um volume considerável de pesquisa tem sido realizado no que tange à derivação de primitivas e protocolos que façam uso dessas características. Parte deste esforço está focado em produzir sistemas criptográficos semelhantes aos baseados em identidade, mas com a eliminação da necessidade de custódia das chaves privadas e minimização dos perigos relacionados – comprometimento da chave mestre, ausência de irretratabilidade e efeitos-colaterais da atuação de um PKG malicioso.

4.2 CL-PKC

As limitações impostas pelo modelo ID-PKC restringem sua aplicabilidade a grupos fechados, com relações de confiança bem-definidas. Assim, a Criptografia de Chave Pública Sem Certificados, proposta por Paterson e Riyami [ARP03], surge como um novo paradigma para criptografia de chave pública que não requer o uso de certificados e não possui a custódia de chave inerente a ID-PKC. Dadas estas características, pode-se afirmar que o modelo situa-se entre a PKI convencional e ID-PKC.

O paradigma CL-PKC mantém a terceira parte confiada presente em ID-PKC, mas altera radicalmente seu papel. A terceira parte agora é denominada *Centro de Geração de Chaves* (*Key Generation Center* – KGC). Ao invés de gerar a chave privada a partir da identidade ID_A de uma entidade A requisitante, o KGC fornece apenas uma chave privada parcial D_A. Esta modificação possibilita que a entidade A combine a chave privada parcial D_A com um segredo d_A de seu conhecimento exclusivo, produzindo a chave privada completa S_A. Desta forma, a chave privada S_A não é conhecida pelo KGC. Intuitivamente, para que as chaves privadas e públicas tenham funcionalidades complementares, é necessário que a chave pública P_A de A seja produzida pela combinação entre o segredo d_A e os parâmetros de sistema distribuídos pelo KGC. O procedimento de geração de chaves privadas parciais, a princípio, continua exigindo confidencialidade e autenticação, já que o KGC deve ter garantias de que a chave privada parcial foi gerada e transmitida para a entidade correta. Observa-se que o sistema deixa de ser baseado em identidades, já que a chave pública não pode ser produzida utilizando-se apenas informação

a respeito da identidade de A. A propriedade de que a chave pública pode ser gerada antes da chave privada é mantida, possibilitando efetivamente que a chave pública armazene informação para certificação implícita. A chave pública da entidade A pode então ser publicada em um repositório, não existindo mais a necessidade de se vincular a chave pública à entidade A por meio de um certificado – a chave pública isolada permite a verificação de titularidade. As alterações em relação ao paradigma ID-PKC podem ser verificadas na Figura 4.3.

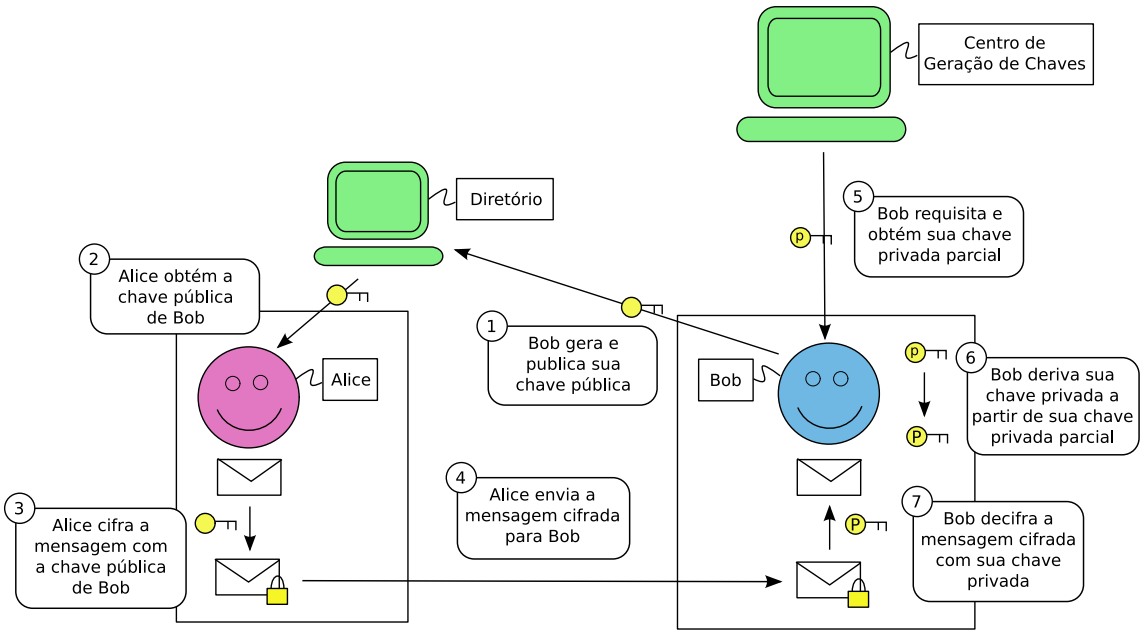


Figura 4.3: Sistema de Criptografia de Chave Pública Sem Certificados.

4.2.1 Propriedades

São várias as propriedades do paradigma CL-PKC.

Revogação

Assim como em ID-PKC, a identidade de A é mantida como um identificador ID_A arbitrário, podendo carregar datas de validade ou informação para delegação de privilégios. Instâncias desta técnica podem construir as chaves com curto tempo de vida ou mensagens cifradas para serem lidas no futuro. A revogação de chaves expiradas é automática e eficiente [BF01].

O problema de revogação imediata, quando ocorre comprometimento da chave privada, por exemplo, não é tão simples. A informação de revogação ainda precisa ser distribuída para os

usuários do sistema e pode ser realizada por meios tradicionais (listas de revogação ou protocolos de verificação em tempo real). O requisito de agilidade na distribuição de informação de revogação ainda é mantido. A melhor solução encontrada para este problema utiliza revogação de identificadores produzidos a partir do identificador original e mantém os mesmos custos de comunicação presentes na PKI tradicional [HHSI04]. A construção de um sistema baseado em identidades ou sem certificados que suporte revogação imediata mais eficiente que a PKI tradicional ainda é um problema em aberto e de difícil tratamento [Cal05].

Estas observações aplicam-se também à revogação dos parâmetros da autoridade central, ao se forçar a reinicialização de todos os pares de chaves e a revogação efetiva das chaves antigas [AR05].

Ausência de certificados

O modelo elimina a necessidade de implementar certificados de titularidade. A chave privada parcial cumpre o papel de certificado, mas altera os mecanismos de distribuição e de informações de certificação. Efeitos colaterais são a simplificação da gerência de chaves e a diminuição dos requisitos de banda, já que é desnecessário contatar a autoridade central para obter certificados adicionais que permitam determinar a validade de uma chave pública. A privacidade das chaves também é elevada, já que os identificadores utilizados carregam apenas a exata informação para funcionamento do sistema, enquanto que, dependendo da aplicação, certificados comuns costumam carregar informação adicional (conta em banco, cadastro de pessoa física, entre outros).

Flexibilidade

A possibilidade de se gerar uma chave pública antes da chave privada correspondente permite que uma entidade B cifre uma mensagem para A utilizando um identificador específico. O identificador deve conter a identidade de A, mas pode conter também informação ligada a uma condição que A precisa demonstrar para o KGC antes que sua chave privada parcial seja emitida. Adicionalmente, o modelo CL-PKC co-existe com ID-PKC: uma entidade A pode solicitar a geração de uma chave privada para Criptografia Baseada em Identidades e imediatamente convertê-la em uma chave adequada para criptografia sem certificados (a mesma infra-estrutura pode ser utilizada para dar suporte aos dois cenários). Vários trabalhos [YL04a, YL04b] já propõem e analisam a segurança de construções genéricas de CL-PKC utilizando sistemas baseados em identidades, para as primitivas de cifração e assinatura.

Confiança e irretratabilidade

Em infra-estruturas de chave pública, o nível de confiança na terceira parte confiada pode assumir diferentes patamares. Uma formulação simples para confiança na autoridade central é proposta por Girault [Gir91]:

- **Nível de confiança 1:** a autoridade confiada conhece ou pode calcular as chaves privadas das entidades registradas. Assim, pode personificar qualquer das entidades a qualquer hora sem ser detectada;
- **Nível de confiança 2:** a autoridade confiada não conhece e não pode calcular as chaves privadas das entidades, mas ainda pode personificar uma entidade sem ser detectada, pela geração de informação falsa de autenticação; ou
- **Nível de confiança 3:** a autoridade confiada não conhece e não pode calcular as chaves privadas das entidades. A autoridade pode personificar uma entidade, mas a fraude sempre pode ser detectada.

O modelo de PKI alcança nível de confiança 3: a geração desonesta de pares de chaves para uma entidade específica pode ser detectada pela existência de múltiplos certificados válidos para uma mesma chave. ID-PKC e demais sistemas onde há custódia de chaves privadas restringem-se ao nível de confiança 1. O modelo CL-PKC, em sua formulação original, alcança nível de confiança 2: a existência de múltiplos pares de chaves para uma mesma identidade é evidência da ação desonesta por parte da entidade cliente ou do KGC, e não é possível decidir entre ambos.

A propriedade de que a chave pública pode ser gerada antes da chave privada pode ser utilizada para elevar o nível de confiança da autoridade central no modelo CL-PKC, com a inserção da chave pública de A na informação ID_A utilizada para a geração da chave parcial. Esta simples modificação confina a chave privada parcial gerada à uma chave pública específica e permite a transmissão de chaves privadas parciais em claro [AR05].

A existência de duas chaves públicas funcionais para uma entidade implica a existência de duas chaves privadas parciais para a mesma entidade, o que indica ação desonesta do KGC. Agora, o nível de confiança é superior a 2, mas não é suficiente para atingir o nível 3. Para examinar esta possibilidade, tem-se o seguinte cenário: dada uma entidade A dotada de uma chave pública P_A, a autoridade central gera uma nova chave privada parcial ligando uma chave pública P'_A à identidade de A. Se uma entidade B cifrar uma mensagem para A utilizando a chave pública P'_A, o KGC pode interceptar e decifrar a mensagem para depois recifrá-la sob a chave P_A. A evidência de que existem duas chaves públicas funcionais para uma mesma entidade depende da existência de uma mesma mensagem M cifrada sob chaves públicas distintas. As entidades A e B só podem detectar o ataque se puderem comparar posteriormente as chaves públicas P_A e P'_A utilizadas para cifração, ou seja, é necessário que as entidades ao menos suspeitem que um

ataque aconteceu. Infelizmente, não é possível implicar o KGC em atuação maliciosa, pois B poderia ter cifrado M sob chaves distintas P_A e P'_A propositadamente. O ataque do KGC pode ser detectado, entretanto, se B for considerado honesto e garantir para a entidade A que cifrou M com apenas uma chave pública – o KGC é a única entidade que poderia converter a mensagem M cifrada com a chave pública $P'A$ em um criptograma de M cifrado com a chave pública P_A, já que detém a chave privada correspondente à chave pública P'_A. A solução definitiva deste problema vem de um efeito colateral importante do confinamento de chaves públicas: a possibilidade de se transmitir chaves privadas parciais em claro – a chave privada parcial agora está restrita à escolha prévia de uma chave pública, que depende de um segredo conhecido apenas por A. Se as chaves privadas parciais são públicas, a chave privada parcial falsa é evidência da ação desonesta do PKG e o sistema alcança nível de confiança 3, compatível com a PKI tradicional. Portanto, o nível de confiança da formulação original é ligeiramente inferior ao nível 3, e o fator de diferença depende da disponibilidade de chaves privadas parciais ou da honestidade dos participantes [AR05].

O confinamento de chaves privadas parciais permite que o sistema atinja nível 3 para primitivas que exigem prova de posse de uma chave privada, como autenticação e assinatura. A existência de duas assinaturas verificáveis sob chaves públicas distintas ou de fragmentos de comunicação que exibem provas de possessão para chaves privadas distintas sempre permitem a detecção de um PKG desonesto. O nível de confiança 3 agrega irretratabilidade à primitiva de assinatura, já que não há a possibilidade de qualquer entidade personificar outra sem ser detectada. Desta forma, esquemas de assinatura baseados no modelo CL-PKC suportam irretratabilidade.

4.2.2 Trabalhos relacionados

O conceito de Criptografia de Chave Pública Sem Certificados foi inspirado nas chaves *auto-certificadas* propostas por Girault [Gir91]. Uma chave pública é auto-certificada quando é construída em conjunto pelo usuário e pela autoridade central e carrega informação de certificação embutida. O esquema funciona da seguinte forma: ao criar um par de chaves (e_A, d_A), o usuário A entrega a chave pública e_A para a autoridade central. A autoridade central constrói um *testemunho* w a partir da combinação entre a chave pública e_A e a identidade real do usuário. Este testemunho tem valor compatível ao de uma assinatura de autoridade certificadora, legitimando a ligação entre chave e identidade. Após este procedimento, qualquer usuário pode cifrar mensagens para o usuário A utilizando o testemunho w, a identidade de A e a chave pública da autoridade central. O testemunho funciona como um certificado simplificado, que diminui requisitos de comunicação, armazenamento e processamento. Entretanto, uma falha grave foi encontrada no modelo original [Sae03] e permite à autoridade central construir parâmetros públicos vulneráveis à fatoração ou cálculo do logaritmo discreto. Uma autoridade central ma-

liciosa pode assim recuperar mensagens cifradas para os seus usuários. Esta vulnerabilidade abaixa o nível de confiança do esquema para o nível 2 e corrigi-la implica em aumentar significativamente o tamanho dos parâmetros envolvidos, comprometendo as vantagens obtidas com a simplificação dos certificados.

O trabalho de Gentry [Gen03] compartilha uma estrutura similar e utiliza emparelhamentos bilineares para simplificar o procedimento de revogação na PKI tradicional. Neste modelo, a chave privada de uma entidade A consiste em dois componentes: um permanente, gerado por A; e outro emitido temporariamente pela autoridade certificadora. Duas chaves públicas correspondentes completam o conjunto de chaves de A: uma escolhida por A e outra calculada a partir da chave pública da autoridade certificadora, da chave pública escolhida por A e de informação temporal. Por construção, uma mensagem cifrada por qualquer entidade só pode ser decifrada por A se ela detiver as duas chaves privadas. A segunda chave privada fornece certificação implícita: um usuário B ao cifrar uma mensagem para A tem a garantia de que a entidade A só poderá decifrá-la caso tenha realizado o procedimento de certificação no intervalo de tempo corrente. Também não há necessidade de B verificar a chave pública de A previamente: a validade vem da confiança depositada na autoridade certificadora para a distribuição do componente secundário da chave privada de A.

Outros sistemas de certificação implícita foram propostos tentando apresentar propriedades avançadas sem depender de emparelhamentos bilineares. Um sistema CL-PKC eficiente que não utiliza emparelhamentos baseia-se na intratabilidade de problemas convencionais [BSNS05]. Entretanto, não fornece a possibilidade de confinar chaves privadas parciais a chaves públicas geradas previamente. Mais radicalmente, um sistema híbrido entre ID-PKC e a PKI tradicional foi proposto, para argumentar que o paradigma ID-PKC é desnecessário [Cal05]. Neste novo sistema, a autoridade central utiliza sua chave-mestre e a identidade do usuário para alimentar o gerador pseudo-aleatório que produzirá o par de chaves. Os demais procedimentos, incluindo a revogação e verificação, são idênticos à PKI tradicional. Este esquema poderia ser transformado em um híbrido entre CL-PKC e a PKI tradicional mas, como as chaves geradas são convencionais, também não haveria suporte para a geração de chaves públicas antes das chaves privadas parciais correspondentes. Esta desvantagem limita o nível de confiança máximo alcançado pelo sistema, sendo um fator decisivo para sua avaliação.

A proposta original de criptografia sem certificados utiliza emparelhamentos bilineares e permite o confinamento de chaves privadas parciais. A técnica de confinamento agrega irretratabilidade às assinaturas e alcança nível de confiança compatível com a PKI tradicional, sem a necessidade de certificados. Estas são características essenciais para a utilização de um paradigma de criptografia em um ambiente de rede aberta.

4.2.3 Fundamentos matemáticos

Como visto anteriormente, as formulações de CL-PKC que exibem as propriedades mais atraentes são instanciadas a partir de emparelhamentos bilineares sobre curvas elípticas.

Curvas Elípticas

Seja p um número primo, m um número positivo e $\mathbb{F}_q$ um corpo finito com $q = p^m$ elementos; p é chamado de *característica* do corpo e m de *grau de extensão*. Denota-se por $\mathbb{F}_q^*$ o corpo $\mathbb{F}_q - \{0\}$.

Uma *curva elíptica* E sobre o corpo $\mathbb{F}_q$ é o conjunto de soluções $(x,y) \in \mathbb{F}_q \times \mathbb{F}_q$ que satisfazem a equação de Weierstrass na forma:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{4.1}$$

onde $a_i \in \mathbb{F}_q$ para $i = 1,2,3,4,6$, e um *ponto no infinito* denotado por ∞. Se $\mathbb{K}$ é uma extensão $\mathbb{K} \equiv \mathbb{F}_{q^k}$ do corpo $\mathbb{F}_q$, o conjunto de pontos $\mathbb{K}$-*racionais* de E, denotado por $E(\mathbb{K})$, é o conjunto de pontos $(x,y) \in E$ tais que $x,y \in \mathbb{K}$. O número de pontos da curva $E(\mathbb{K})$, denotado por $\#E(\mathbb{K})$, é chamado de *ordem* da curva sobre o corpo $\mathbb{K}$. A *condição de Hasse* afirma que $\#E(\mathbb{F}_q) = q + 1 - t$, onde $|t| \leq 2\sqrt{q}$ é chamado de *traço de Frobenius*. Curvas em que a característica p divide t são chamadas de *curvas supersingulares*.

Sem perda de generalidade, se a característica do corpo não é 2 ou 3, a equação da curva elíptica pode ser simplificada na forma:

$$y^2 = x^3 + ax + b, \tag{4.2}$$

onde a, $b \in \mathbb{F}_q$ e o discriminante $\delta = -16(4a^3 + 27b^2) \neq 0$. O *twist quadrático* $E^t(\mathbb{F}_q)$ de uma curva $E(\mathbb{F}_q)$ é dada por $y^2 = x^3 + v^2ax + v^3b$ para todo não-resíduo quadrático $v \in \mathbb{F}_q$.

O conjunto $\{(x,y) \in \mathbb{K} \times \mathbb{K} : E(\mathbb{K})\} \cup \{\infty\}$ sob a operação de grupo $+$, forma um grupo aditivo denotado por $(E(\mathbb{K}),+)$. A operação de adição de pontos no grupo é definida da seguinte forma:

- Seja $P \in E(\mathbb{K})$. Então, $P + \infty = P$ e $\infty + P = P$. O elemento ∞ serve como identidade aditiva para o grupo. Se $P = \infty$, então $-P = \infty$. A notação $E(\mathbb{K})^*$ denota o grupo $E(\mathbb{K})$ excluindo o elemento de identidade ∞;
- Seja $P = (x,y) \in E(\mathbb{K})^*$. Então, $-P = (-x,y) = (x,-y)$ e $P + (-P) = \infty$. O inverso de P é $-P$;
- Seja $P = (x,y) \in E(\mathbb{K})^*$ e $Q = (x',y') \in E(\mathbb{K})^*$. Se $x \neq x'$, então $P + Q = -R$, onde $-R$ é a reflexão do ponto R no eixo x e R é o ponto de intersecção entre a curva elíptica e a linha que passa por P e Q;

- Seja $P = (x,y) \in E(\mathbb{K})^*$. Então, $P+P = -R$, onde $-R$ é a reflexão do ponto R no eixo x. O ponto R é a intersecção entre a curva elíptica e a tangente à curva E que passa por P.

A operação do grupo é comutativa e associativa. Logo, $(E(\mathbb{K}),+)$ é um grupo abeliano com elemento de identidade ∞. A notação mP denota a multiplicação de $P \in E(\mathbb{K})$ por $m \in \mathbb{Z}_q$. O valor de mP é dado pela relação de recorrência:

$$mP = \begin{cases} \infty, & \text{se } m = 0; \\ (-m)(-P) & \text{se } m \leq -1; \\ (m-1)P + P & \text{se } m \geq 1. \end{cases} \tag{4.3}$$

Seja $n = \#(\mathbb{F}_{q^k})$. A *ordem* de um ponto $P \in E$ é o menor inteiro $r > 0$ tal que $rP = \infty$ e sempre divide a ordem da curva. O conjunto de pontos de torção r de E, denotado por $E(\mathbb{K})[r]$, é o conjunto $\{P \in E(\mathbb{K}) | rP = \infty\}$. Destas definições, segue que $\langle P \rangle$, o grupo de pontos gerado por P, é um subgrupo de $E(\mathbb{K})[r]$, que por sua vez é um subgrupo de $E(\mathbb{K})[n]$. Dizemos que o subgrupo $\langle P \rangle$ tem *grau de mergulho* k se $r | q^k - 1$ e $r \nmid q^s - 1$ para todo $0 < s < k$.

Emparelhamentos Bilineares

Seja $\mathbb{G}_1$ um grupo cíclico aditivo de ordem prima q e $\mathbb{G}_2$ um grupo cíclico multiplicativo tal que $|\mathbb{G}_1| = |\mathbb{G}_2|$. Seja P o gerador de $\mathbb{G}_1$. Um mapeamento $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ é dito um *emparelhamento bilinear admissível* [BF01] se satisfaz as seguintes propriedades:

1. *Bilinearidade:* dados $Q, W, Z \in \mathbb{G}_1$, temos

   $$e(Q, W+Z) = e(Q,W) \cdot e(Q,Z) \text{ e } e(Q+W,Z) = e(Q,Z) \cdot e(W,Z).$$

   Conseqüentemente, para quaisquer $a, b \in \mathbb{Z}_q$, temos:

   $$e(aQ,bW) = e(Q,W)^{ab} = e(abQ,W) = e(Q,abW) = e(bQ,aW) = e(bQ,W)^a.$$

2. *Não-degeneração:* $e(P,P) \neq 1_{\mathbb{G}_2}$, onde $1_{\mathbb{G}_2}$ é o elemento de identidade do grupo $\mathbb{G}_2$.

3. *Eficiência:* O mapeamento e pode ser calculado eficientemente, ou seja, tem complexidade polinomial.

Tipicamente, $\mathbb{G}_1$ é um subgrupo do grupo de pontos de uma curva elíptica sobre um corpo finito $E(\mathbb{F}_q)$ e $\mathbb{G}_2$ é um subgrupo do grupo multiplicativo de um corpo finito relacionado a $\mathbb{F}_q$ (uma de suas extensões, por exemplo). O mapeamento e é obtido pela modificação do emparelhamento de Weil ou de Tate sobre uma curva elíptica supersingular em $\mathbb{F}_q$ [BKLS02].

A definição de emparelhamentos bilineares pode ser generalizada para a construção de *emparelhamentos assimétricos*, ou seja: o mapeamento e é da forma $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, com $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = q$ e $\mathbb{G}_1 \neq \mathbb{G}_2$. Há ainda dois geradores $P \in \mathbb{G}_1^*$ e $Q \in \mathbb{G}_2^*$, tais que $P = \psi(Q)$, onde ψ é um homomorfismo de $\mathbb{G}_2$ em $\mathbb{G}_1$ eficientemente computável.

Problemas Relacionados

Um conjunto de problemas clássicos dos quais se tem evidência de intratabilidade são utilizados explicitamente ou implicitamente por sistemas criptográficos de chave pública. São estes:

Problema do Logaritmo Discreto (Discrete Logarithm Problem – DLP): Seja $\mathbb{G}$ um grupo cíclico finito e g um gerador de $\mathbb{G}$. Dados $\langle g, g^a \rangle$, com escolha uniformemente aleatória de $a \in \mathbb{Z}_{|\mathbb{G}|}$, encontrar $a \in \mathbb{G}$.

O algoritmo mais eficiente para cálculo de logaritmos discretos [Gor93] é uma variação do algoritmo de cálculo de índices [Adl77] e apresenta complexidade sub-exponencial. Para um grupo de pontos em curva elíptica, o Problema do Logaritmo Discreto consiste em obter m a partir do resultado da operação de multiplicação mP. Existem evidências de que a técnica de cálculo de índices não pode ser estendida para grupos de pontos em curvas elípticas [SS98].

Problema Diffie-Hellman Computacional (Computational Diffie Hellman Problem – CDHP): Seja $\mathbb{G}$ um grupo cíclico finito e g um gerador de $\mathbb{G}$. Dados $\langle g, g^a, g^b \rangle$ com escolha uniformemente aleatória de $a, b \in \mathbb{Z}_{|\mathbb{G}|}$, encontrar $g^{ab} \in \mathbb{G}$.

Problema de Decisão Diffie-Hellman (Decisional Diffie Hellman Problem – DDHP): Seja $\mathbb{G}$ um grupo cíclico finito e g um gerador de $\mathbb{G}$. Dados $\langle g, g^a, g^b, g^c \rangle$ com escolha uniformemente aleatória de $a, b, c \in \mathbb{Z}_{|\mathbb{G}|}$, determinar se $g^{ab} = g^c \in \mathbb{G}$.

A derivação de problemas análogos aos problemas ditos convencionais no contexto de emparelhamentos bilineares é direta. Sistemas criptográficos construídos a partir de emparelhamentos bilineares também utilizam a intratabilidade potencial destes problemas para fornecer segurança e suporte para prova de propriedades específicas. A instanciação de um sistema particular depende da existência de um gerador de parâmetros compatíveis com o sistema criptográfico considerado.

Gerador de Parâmetros Diffie-Hellman Bilinear: Um algoritmo probabilístico IG é um *gerador de parâmetros Diffie-Hellman bilinear*, se:

1. IG recebe um parâmetro de segurança k como entrada, para $k \geq 1$;
2. IG tem complexidade polinomial em k; e
3. IG produz um primo q, a descrição dos grupos $\mathbb{G}_1$, $\mathbb{G}_2$ de ordem prima q e um emparelhamento $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.

Problema Diffie-Hellman Bilinear (Bilinear Diffie-Hellman Problem – BDHP): Seja $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ a saída de um algoritmo $IG(k)$ e seja P um gerador de $\mathbb{G}_1$. Dados $\langle P, aP, bP, cP \rangle$, com

escolhas uniformemente aleatórias de $a,b,c \in \mathbb{Z}_q$, calcular $e(P,P)^{abc} \in \mathbb{G}_2$.

Problema Diffie-Hellman Bilinear Generalizado (Generalized Bilinear Diffie-Hellman Problem – GBDHP): Seja $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ a saída de um algoritmo $IG(k)$ e seja P um gerador de $\mathbb{G}_1$. Dados $\langle P, aP, bP, cP \rangle$, com escolhas uniformemente aleatórias de $a,b,c \in \mathbb{Z}_q$, encontrar um par $\langle Q \in \mathbb{G}_1^*, e(P,Q)^{abc} \in \mathbb{G}_2 \rangle$.

Problema de Decisão Diffie-Hellman Bilinear (Decisional Bilinear Diffie-Hellman Problem – DBDHP): Seja $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$ a saída de um algoritmo $IG(k)$ e seja P um gerador de $\mathbb{G}_1$. Dados $\langle P, aP, bP, cP \rangle$ e $Q = e(P,P)^{abc} \in \mathbb{G}_2^*$, com escolhas uniformemente aleatórias de $a,b,c \in \mathbb{Z}_q$, determinar se $Q = e(P,P)^{abc}$.

Uma conseqüência da bilinearidade é que para instâncias de emparelhamento $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$, o DDHP no grupo $\mathbb{G}_1$ pode ser solucionado em tempo polinomial: determinar se $cP = abP$ para a tupla $\langle P, aP, bP, cP \rangle$, com $a,b,c \in \mathbb{Z}_q^*$ pode ser eficientemente computado verificando se $e(aP,bP) = e(P,cP)$.

A partir dos problemas relacionados, pode-se perceber que a intratabilidade do Problema do Logaritmo Discreto em $\mathbb{G}_1$ depende da intratabilidade do mesmo problema em $\mathbb{G}_2$, visto que o cálculo de logaritmos discretos em $\mathbb{G}_2$ fornece um método para o cálculo de logaritmos discretos em $\mathbb{G}_1$. Assim, o parâmetro de segurança k utilizado deve ser de magnitude suficiente para que o cálculo de logaritmos discretos em $\mathbb{G}_2$ seja difícil.

4.2.4 Formalização

Formalmente, o modelo CL-PKC compreende sete algoritmos [ARP03]:

Inicializar. Recebe um parâmetro de segurança k e retorna os parâmetros de sistema params e a chave mestre s. Os parâmetros de sistema incluem uma descrições do espaço de mensagens $\mathcal{M}$, do espaço de criptogramas $\mathcal{C}$ e do espaço de assinaturas $\mathcal{S}$. Seguindo a prática comum, os parâmetros de sistema são conhecidos publicamente, enquanto s é mantida em sigilo pelo KGC.

Extrair. Recebe como entrada params, s e um identificador arbitrário $\mathsf{ID}_A \in \{0,1\}^*$ para a entidade A e retorna uma chave privada parcial D_A. Tipicamente, este algoritmo é executado pelo KGC e a chave é transportada para A por meio de um canal opcionalmente autenticado e confidencial (depende do confinamento da chave privada parcial para o identificador, como discutido anteriormente).

Gerar-chaves. Recebe como entrada params e constrói a chave privada S_A e a chave pública P_A para a entidade A. A construção da chave privada completa S_A está condicionada à

geração de um valor secreto d_A e sua combinação com a chave privada parcial D_A. A chave pública geralmente pode ser gerada antes da chave privada.

Cifrar. Recebe como entrada params, $M \in \mathcal{M}$, a chave pública P_A e o identificador ID_A da entidade A. Retorna um criptograma $C \in \mathcal{C}$ ou um símbolo nulo $\perp$, indicando falha durante a cifração.

Decifrar. Recebe como entrada params, $C \in \mathcal{C}$, uma chave privada S_A e retorna a mensagem em claro $M \in \mathcal{M}$ ou o símbolo nulo $\perp$ indicando falha durante a decifração (ou seja, a certificação implícita não pôde ser verificada).

Assinar. Recebe como entrada params, uma mensagem $M \in \mathcal{M}$ e uma chave privada S_A e produz uma assinatura $S \in \mathcal{S}$ para M.

Verificar. Recebe como entrada params, a chave pública P_A, o identificador ID_A da entidade A e a assinatura $S \in \mathcal{S}$ para uma mensagem $M \in \mathcal{M}$. Retorna verdadeiro ou falso, dependendo de quando a assinatura é aceitável, ou $\perp$, caso haja falha durante a verificação.

Normalmente, os algoritmos para geração das chaves privada e pública são executados pela entidade A, após gerar seu segredo d_A. Fica claro perceber que não existe uma ordenação temporal na geração das chaves pública e privada. Supõe-se ainda que d_A é selecionado aleatoriamente em um conjunto de tamanho adequado, e que A é a única entidade que conhece S_A e d_A. Os algoritmos devem conservar propriedades de consistência:

$$\forall M \in \mathcal{M} : \mathsf{Decifrar}(\mathsf{params}, C = \mathsf{Cifrar}(\mathsf{params}, P_A, \mathsf{ID}_A, M), S_A) = M$$

$$\forall M \in \mathcal{M} : \mathsf{Verificar}(\mathsf{params}, S = \mathsf{Assinar}(\mathsf{params}, S_A, M), P_A, \mathsf{ID}_A, M) = \mathsf{verdadeiro}$$

4.2.5 Modelo de adversário

O adversário para CL-PKC tem poder idêntico ao de um adversário para PKI no que diz respeito à substituição de chaves públicas em repositórios de chaves. Mesmo com a ausência de informação de autenticação para as chaves públicas, pode-se ver que um ataque desta natureza não tem resultados úteis: sem a chave privada correta, cuja produção depende da cooperação do KGC, um adversário não é capaz de decifrar mensagens cifradas com a chave pública falsa nem produzir assinaturas verificáveis com a chave pública falsa.

De forma similar à PKI tradicional, deve-se assumir que o KGC não participa de ataques de substituição de chaves já que, com o poder de gerar um par de chaves para qualquer entidade e distribuir a chave pública correspondente, o KGC poderia personificar qualquer entidade com sucesso. Assim, é premissa de segurança do sistema que o KGC não distribua chaves públicas

falsas. A capacidade de participar em outras atividades maliciosas, como escutas passiva ou ativa, é conservada.

Desta forma, são considerados dois adversários distintos no modelo CL-PKC:

- Um adversário que não tem acesso à chave mestre do KGC, mas pode extrair chaves privadas parciais ou gerar chaves privadas completas, tem acesso às chaves públicas das entidades e pode substituí-las em um repositório público de chaves; e
- Um adversário que detém a chave mestre do KGC e possui acesso às chaves públicas das entidades, mas não pode substituir nenhuma chave pública. A posse da chave mestre permite que o adversário compute chaves privadas parciais e completas para si.

O poder do adversário inclui a habilidade de substituir indefinidamente chaves públicas presentes em um repositório público de chaves. Como resultado, o adversário pode impedir que uma determinada entidade comunique-se com as demais por meio da distribuição de chaves públicas falsas para as entidades comunicantes. Este ataque chama-se *negação de decifração*, por efetivar uma espécie de negação de serviço para a confidencialidade da vítima, e é inerente aos sistemas de cifração baseados no paradigma CL-PKC. A solução deste problema depende da combinação de sistemas criptográficos convencionais de criptografia sem certificados [LAS06].

Outros modelos de adversário foram propostos, em buscas de características mais próximas a ataques reais [Den06]. O modelo de adversário aqui apresentado é utilizado neste trabalho, porque a maioria dos sistemas criptográficos CL-PKC propostos teve sua segurança foi provada considerando adversários com estas características [Den06].

4.3 Sistemas criptográficos

Nesta seção, esquemas criptográficos construídos a partir do conceito de CL-PKC são descritos. Em todos os esquemas, a primitiva é instanciada utilizando emparelhamentos bilineares sobre curvas elípticas. É importante observar que os protocolos a seguir não são os mais eficientes já propostos e têm o procedimento de geração de chaves vulnerável à ação de um KGC malicioso [ACL$^+$06]. A apresentação destes protocolos, portanto, tem a finalidade única de ilustrar os conceitos apresentados.

4.3.1 Cifração

São apresentados dois sistemas criptográficos para cifração. O primeiro sistema é um instanciamento simplificado para destacar a construção. O segundo é uma modificação do primeiro pela aplicação de uma transformação [FO99], que confere resistência contra ataques adaptativos de criptograma escolhido sob o modelo do oráculo aleatório [BR93]. Nota-se que o sistema

de cifração é uma modificação do sistema ID-PKC de Boneh e Franklin [BF01], agregando as propriedades que caracterizam um sistema sem certificados.

Sistema Básico

Seja k o parâmetro de segurança do sistema e IG um gerador de parâmetros BDH com entrada k. São definidos os cinco algoritmos para cifração:

Inicializar. Consiste em:

1. Executar IG com entrada k e obter como saída a tupla $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$, onde $\mathbb{G}_1$ e $\mathbb{G}_2$ são grupos de ordem q e $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ é um emparelhamento bilinear;
2. Escolher um gerador arbitrário $P \in \mathbb{G}_1$;
3. Selecionar s uniformemente aleatório em $\mathbb{Z}_q^*$ e atribuir $P_0 = sP$;
4. Escolher funções de *hash* criptográficas $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ e $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$. O parâmetro n é o comprimento em *bits* das mensagens em texto claro.
5. Retornar os parâmetros de sistema $\mathsf{params} = \langle \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2 \rangle$ e a chave mestre $s \in \mathbb{Z}_q^*$. O espaço de mensagens é $\mathcal{M} = \{0,1\}^n$ e o espaço de criptogramas é $\mathcal{C} = \mathbb{G}_1 \times \{0,1\}^n$.

Extrair. Recebe como entrada um identificador $\mathsf{ID}_A \in \{0,1\}^*$ e produz a chave privada parcial para a entidade A. Consiste em:

1. Calcular $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1^*$;
2. Retornar a chave privada parcial $D_A = sQ_A \in \mathbb{G}_1^*$.

Por construção, a entidade A pode verificar a correção do algoritmo de extração testando a condição $e(D_A, P) = e(Q_A, P_0)$.

Gerar-chaves. Recebe como entrada params e seleciona aleatoriamente $d_A \in \mathbb{Z}_q^*$ como o valor secreto da entidade A. Utiliza d_A para transformar D_A na chave privada completa S_A, calculando $S_A = d_A D_A = d_A s Q_A \in \mathbb{G}_1^*$. Constrói a chave pública P_A como $P_A = \langle X_A, Y_A \rangle$, onde $X_A = d_A P$ e $Y_A = d_A P_0 = d_A s P$.

Cifrar. Para cifrar uma mensagem $M \in \mathcal{M}$ para a entidade A com identificador $\mathsf{ID}_A \in \{0,1\}^*$ e chave pública $P_A = \langle X_A, Y_A \rangle$, o algoritmo executa os seguintes passos:

1. Verificar que $X_A, Y_A \in \mathbb{G}_1^*$ e que $e(X_A, P_0) = e(Y_A, P)$. Se qualquer das verificações falhar, retornar $\perp$ e abortar a cifração;
2. Calcular $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1^*$;

3. Escolher um valor aleatório $r \in \mathbb{Z}_q^*$;
4. Calcular e retornar o criptograma:

$$C = \langle rP, M \oplus H_2(e(Y_A, Q_A)^r) \rangle.$$

Decifrar. Seja $C = \langle U, V \rangle \in \mathcal{C}$. Para decifrar o criptograma utilizando a chave privada S_A, calcular e retornar:

$$V \oplus H_2(e(U, S_A)).$$

Se $\langle U = rP, V \rangle$ é o criptograma de M para a entidade A com chave pública $P_A = \langle X_A, Y_A \rangle$, o princípio de consistência é satisfeito:

$$\begin{aligned} V \oplus H_2(e(S_A, U)) &= V \oplus H_2(e(rP, x_A s Q_A)) \\ &= V \oplus H_2(e(x_A s P, Q_A)^r) \\ &= V \oplus H_2(e(Y_A, Q_A)^r) \\ &= M. \end{aligned}$$

Sistema Completo

Tendo a versão básica descrita, pode-se especificar o esquema completo de cifração, aplicando a técnica de transformação Fujisaki-Okamoto [FO99]:

Inicializar. Idêntico à versão básica com a exceção de que são necessárias duas funções de *hash* criptográficas adicionais $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_n^*$ e $H_4 : \{0,1\}^n \to \{0,1\}^n$. Retorna os parâmetros de sistema $\mathsf{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2, H_3, H_4 \rangle$ e a chave mestre $s \in \mathbb{Z}_q^*$. O espaço de mensagens é $\mathcal{M} = \{0,1\}^n$ e o espaço de criptogramas é $\mathcal{C} = \mathbb{G}_1 \times \{0,1\}^{2n}$.

Extrair. Idêntico à versão básica.

Gerar-chaves. Idêntico à versão básica.

Cifrar. Para cifrar uma mensagem $M \in \mathcal{M}$ para a entidade A com identificador $\mathsf{ID}_A \in \{0,1\}^*$ e chave pública $P_A = \langle X_A, Y_A \rangle$, o algoritmo executa os seguintes passos:

1. Verificar que $X_A, Y_A \in \mathbb{G}_1^*$ e que $e(X_A, P_0) = e(Y_A, P)$. Se qualquer das verificações falhar, retornar $\perp$ e abortar a cifração;
2. Computar $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1^*$;
3. Escolher um valor aleatório $\sigma \in \{0,1\}^n$;

4. Fazer $r = H_3(\sigma, M)$;
5. Calcular e retornar o criptograma:

$$C = \langle rP, \sigma \oplus H_2(e(Y_A, Q_A)^r), M \oplus H_4(\sigma) \rangle.$$

Decifrar. Seja $C = \langle U, V, W \rangle \in \mathcal{C}$. Decifrar o criptograma utilizando a chave privada S_A requer:

1. Calcular $\sigma' = V \oplus H_2(e(U, S_A))$;
2. Calcular $M' = W \oplus H_4(\sigma')$;
3. Fazer $r' = H_3(\sigma', M')$ e testar se $U = r'P$. Se não for o caso, retornar $\perp$ e rejeitar o criptograma.
4. Retornar M' como a decifração de C.

Quando C é uma cifração legítima de M sob a chave pública P_A e o identificador (ID_A), a decifração de C irá produzir M:

$$\begin{aligned} V \oplus H_2(e(S_A, U)) &= V \oplus H_2(e(rP, x_A s Q_A)) \\ &= V \oplus H_2(e(x_A s P, Q_A)^r) \\ &= V \oplus H_2(e(Y_A, Q_A)^r) \\ &= \sigma; \\ W \oplus H_4(\sigma) &= M. \end{aligned}$$

4.3.2 Assinatura

Inicializar. Idêntico ao algoritmo de inicialização para o esquema de cifração, com a exceção de que é usada apenas uma função de *hash* $H_1 : \{0,1\}^* \times \mathbb{G}_2 \to \mathbb{Z}_q^*$. Retorna os parâmetros de sistema $\mathsf{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, P, P_0, H_1 \rangle$ e a chave mestre $s \in \mathbb{Z}_q^*$. O espaço de assinaturas é $S = \mathbb{G}_1 \times \mathbb{Z}_q^*$.

Extrair. Idêntico ao algoritmo presente no esquema de cifração.

Gerar-chaves. Idêntico ao algoritmo presente no esquema de cifração.

Assinar. Para assinar uma mensagem $M \in \mathcal{M}$ usando a chave privada S_A, executar os passos:

1. Escolher um valor aleatório $k \in \mathbb{Z}_q^*$;
2. Calcular $r = e(kP, P) \in \mathbb{G}_2$;
3. Fazer $h = H_1(M, r) \in \mathbb{Z}_q^*$;

4. Calcular $S = hS_A + kP \in \mathbb{G}_1$;
5. Retornar a assinatura $\langle S, h \rangle$.

Verificar. Para verificar uma suposta assinatura $\langle S, h \rangle$ de uma mensagem $M \in \mathcal{M}$ para a identidade ID_A e a chave pública $P_A = (X_A, Y_A)$, executar os passos:

1. Verificar que a igualdade $e(X_A, P_0) = e(Y_A, P)$ é verdadeira. Se não for o caso, retornar $\perp$ e abortar a verificação;
2. Calcular $r' = e(P,S) \cdot e(-Y_A, Q_A)^h$;
3. Verificar se $h' = H_1(M, r')$. Se a igualdade for verificada, retornar verdadeiro, caso contrário, retornar falso.

Claramente, a consistência é mantida:

$$\begin{aligned} e(P,S) \cdot e(-Y_A, Q_A) &= e(P, hS_A + kP) \cdot e(-d_A sP, Q_A) \\ &= e(P, hS_A) \cdot e(P, kP) \cdot e(P, Q_A)^{-d_A s} \\ &= e(P, d_A sQ_A) \cdot e(kP, P) \cdot e(P, Q_A)^{-d_A s} \\ &= e(P, Q_A)^{d_A s} \cdot e(P, Q_A)^{-d_A s} \cdot e(kP, P) \\ &= r; \\ H(M, r) &= h. \end{aligned}$$

4.4 Implementação

A concretização da primitiva e dos esquemas de cifração e assinatura consistiu na implementação do emparelhamento de Tate [Sco05]. O emparelhamento de Tate foi escolhido por apresentar melhor desempenho que o emparelhamento de Weil [Maa04].

4.4.1 Emparelhamento de Tate

A formulação do emparelhamento de Tate parte da teoria de divisores [Maa04]. Para uma curva elíptica com característica $p > 3$, utilizamos a notação $E(x,y) = 0$ para indicar o conjunto de valores (x,y), tais que $E(x,y) = y^2 - (x^3 + ax + b) = 0$, com $x, y \in \mathbb{F}_{q^k}$. A mesma notação é utilizada para uma função $f(x,y) = 0$.

Divisores

Um *divisor* é uma soma formal de pontos na curva $E(\mathbb{F}_{q^k})$:

$$\mathcal{D} = \sum_{P \in E} d_P(P), \tag{4.4}$$

onde d_P é um inteiro e (P) é um símbolo formal. A adição tem as mesmas propriedades aritméticas da adição de inteiros, com a exceção de que é realizada sobre símbolos formais. Divisores podem envolver símbolos formais para vários pontos na curva elíptica, mas limita-se o enfoque a divisores de funções, por possuírem poucos símbolos. O divisor de uma função $f(x,y) = 0$ contém apenas os termos $d_P(P)$ tais que $P = (x,y)$ está na curva $E(x,y) = 0$ e na função $f(x,y) = 0$, ou seja, os pontos em que E e f se encontram. Para qualquer ponto P na curva elíptica tal que E e f não têm intersecção, $d_p = 0$.

O *grau de um divisor* $\mathcal{D}$ é definido como a soma de seus coeficientes:

$$deg(\mathcal{D}) = \sum_{P \in E} d_P \tag{4.5}$$

O *suporte de um divisor* $\mathcal{D}$ é o conjunto:

$$sup(\mathcal{D}) = \{P \in E | n_P \neq 0\}. \tag{4.6}$$

Uma estrutura de grupo abeliano aditivo é definida no conjunto dos divisores pela soma dos coeficientes correspondentes em suas somas formais. Assim:

$$\sum_{P \in E} m_P(P) + \sum_{P \in E} n_P(P) = \sum_{P \in E} (m_P + n_P)(P) \quad e \quad n\mathcal{D} = \sum_{P \in E} (nd_P)(P). \tag{4.7}$$

Funções racionais sobre uma curva elíptica

Uma função $f(x,y)$ em um corpo finito $\mathbb{F}_q$ é dita racional se

$$f(x,y) = \frac{P(x,y)}{Q(x,y)}, \tag{4.8}$$

e $P(x,y)$ e $Q(x,y)$ são polinômios em $\mathbb{F}_q$. Um função racional está sobre uma curva $E(\mathbb{F}_q)$ se $f(x,y) = 0$ e $E(x,y) = 0$ têm ao menos uma solução em comum. Utiliza-se a notação $P \in f \cap E$ para um ponto $P = (x,y)$ que satisfaz esta condição.

Seja $f(x,y)$ uma função racional sobre $E(x,y) = 0$. Para um ponto $P \in f \cap E$, P é chamado *zero* se $f(P) = 0$ e *pólo* se $f(P) = \infty$.

Para todo ponto $P \in E(\mathbb{F}_q)$, existe uma função racional u com $u(P) = 0$ que satisfaz a seguinte propriedade: toda função racional não-nula f pode ser escrita como $f(P) = u^d s(P)$ para algum inteiro d e uma função racional s, tal que $s(P) \neq 0, \infty$ [Maa04]. Assim, a *ordem*

de P, denotada por ord_P é d. Se P é um zero da função f, então a ordem de P é positiva e P tem *multiplicidade* ord_P. Se P é uma pólo de f, então a ordem de P é negativa e P tem *multiplicidade* $-ord_P$.

Divisor de uma função racional

Seja f uma função racional sobre E. O divisor de F é

$$div(f) = \sum_{P \in E} ord_P(P). \tag{4.9}$$

O divisor de uma função é chamado *divisor principal*. Um divisor $\mathcal{D}$ é principal se e somente se:

$$deg(\mathcal{D}) = 0 \qquad \text{e} \qquad \sum_{P \in E} d_P P = \infty.$$

Dois divisores $\mathcal{C}$ e $\mathcal{D}$ são *equivalentes* ($\mathcal{C} \sim \mathcal{D}$) se a diferença $\mathcal{C} - \mathcal{D}$ é um divisor principal. Para se avaliar uma função racional f em um divisor $\mathcal{D}$ que satisfaz $sup(div(f)) \cap sup(\mathcal{D}) = \varnothing$, calcula-se [Maa04]:

$$f(\mathcal{D}) = \prod_{P \in sup(\mathcal{D})} f(p)^{n_P}. \tag{4.10}$$

Definição do emparelhamento

Seja $P \in E(\mathbb{E}_{q^k})[r]$ com r e q co-primos, e seja $\mathcal{D}_P$ um divisor equivalente a $(P) - (\infty)$. Sob estas circunstâncias, o divisor $r\mathcal{D}_P$ é principal e existe uma função f_P tal que $div(f_P) = r\mathcal{D}_P = n(P) - n(\infty)$. O *emparelhamento de Tate* de ordem r é a função

$$e_r : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})[r] \rightarrow \mathbb{F}^*_{q^k}, \tag{4.11}$$

dada por:

$$e_r(P,Q) = f_P(\mathcal{D})^{(q^k-1)/r}. \tag{4.12}$$

para algum divisor $\mathcal{D} \sim (Q) - (\infty)$.

Cálculo do emparelhamento

O emparelhamento de Tate é um *emparelhamento bilinear* e a computação de $f_P(\mathcal{D})$ é obtida pela aplicação do *Algoritmo de Miller* [Mil86], que substitui o cálculo de $f_P(\mathcal{D})$ pelo cálculo de $f_P(Q)$ e cuja saída define uma potência r-ésima em $\mathbb{F}^*_{q^k}$. A exponenciação final em

$(q^k-1)/r$ é necessária para produzir um resultado único, já que $a^{(q^k-1)}=1$, para qualquer $a \in \mathbb{F}_{q^k}^*$. Conseqüentemente, o valor após esta exponenciação é uma raiz r-ésima da unidade.

4.4.2 Parâmetros

A escolha de parâmetros foi dominada pelos requisitos principais de segurança e de viabilidade, considerando o suporte disponível em bibliotecas criptográficas livres. Como os protocolos aqui descritos são destinados a uma aplicação real, o requisito de segurança exigiu uma decisão conservadora na escolha do tamanho dos parâmetros e a viabilidade exigiu código portável, de fácil implementação e desempenho razoável. Utilizou-se, portanto, a formulação do emparelhamento de Tate em curvas não-supersingulares sobre corpos primos. Os mesmos requisitos direcionaram a escolha de um grau de mergulho $k=2$, pelo bom desempenho, possibilidade de otimizações [Sco05], ampla disponibilidade de implementações de curvas elípticas sobre corpos $\mathbb{F}_p$, facilidade na implementação de aritmética no corpo de extensão $\mathbb{F}_{p^2}$ e por ser improvável o aparecimento de novos ataques que sejam poderosos o suficiente para colocar em risco a segurança da implementação, se feita uma escolha cuidadosa no tamanhos dos parâmetros [Sco05]. Nesta instanciação particular, ataques que utilizam o cálculo de índices estão disponíveis, sendo improvável o surgimento de ataques mais poderosos.

Corpo finito

O corpo finito escolhido é o corpo primo $\mathbb{F}_p$, com $p = 3 \bmod 4$. Para esta escolha particular de p, temos que $v=-1$ é sempre não-resíduo quadrático, o que permite a implementação eficiente de operações de quadrado e multiplicação na aritmética da extensão quadrática de $\mathbb{F}_p$ [Sco05]. O tamanho de p foi fixado em 512 *bits*, considerando que o melhor ataque conhecido (subexponencial) no contexto considerado corresponde ao cálculo do logaritmo discreto no corpo $\mathbb{F}_{p^2}$, onde cada elemento tem 1024 *bits*. Um ataque desta natureza tem poder semelhante a um ataque no sistema criptográfico RSA de 1024 *bits*, que é considerado seguro para o poder computacional vigente.

Curva

A curva escolhida é da forma

$$E : y^2 = x^3 - 3x + b, \tag{4.13}$$

com $b \in \mathbb{F}_p$. Se $x, y \in \mathbb{F}_p$, a curva tem $\#E(\mathbb{F}_p) = p+1-t$ pontos, onde t é o traço de Frobenius. Se $x, y \in \mathbb{F}_{p^2}$, a curva tem $\#E(\mathbb{F}_{p^2}) = (p+1-t)(p+1+t)$ pontos. Como $p = 3 \bmod 4$, o *twist* $E^t(\mathbb{F}_p)$ da curva é da forma

$$E : y^2 = x^3 - 3x - b. \tag{4.14}$$

e tem $p+1+t$ pontos, se $x, y \in \mathbb{F}_p$. Considerando o nível de segurança de sistemas criptográficos de curvas elípticas não-supersingulares, é desejável que $E(\mathbb{F}_p)$ tenha um subgrupo de ordem r com pelo menos 160 *bits*. Por questões de eficiência, r deve ter baixo peso de Hamming, de preferência um primo de Solinas como $r = 2^{159} + 2^{17} + 1$. Além disso, o grupo de pontos de ordem r em $E(\mathbb{F}_{p^2})$ deve ter grau de mergulho $k = 2$ e a condição $r|p+1$ deve ser mantida para que os parâmetros possibilitem o cálculo do emparelhamento de Tate.

Particularmente, foi escolhida a curva (em base hexadecimal) [Sco05]:

```
p = 0xDF9BD3ED0034174E54597AA4E2AB033D21C7F6F1AFDD080D4708BC67CAC2AED5
      54FE43F3DA7CD547ED458502C46356BB2A76688DDF064094EBE7785EDE2E413F
a = 0xDF9BD3ED0034174E54597AA4E2AB033D21C7F6F1AFDD080D4708BC67CAC2AED5
      54FE43F3DA7CD547ED458502C46356BB2A76688DDF064094EBE7785EDE2E413C
b = 0xCFEC8DDB4E226F34828D4F9B30571BB52E14D1611FA34031423862B3ACB17910
      2A1C152E860FC993A87999CB6A8539516C04950344270037ABC0905175FD47E
#E = 0xDF9BD3ED0034174E54597AA4E2AB033D21C7F6F1AFDD080D4708BC67CAC2AED3
      767AC584178BA7D62E6F13DDC46356BB2A6EEE7C284037F0B03E22219BED5EF6
t = 0x1DE837E6FC2F12D71BED6712500000000000077A11B6C608A43BA9563D4240E24A
r = 0x8000000000000000000000000000000000020001
```

4.4.3 Algoritmo

Considerando o grau de mergulho $k = 2$, o emparelhamento de Tate toma dois pontos P e Q na curva elíptica $E(\mathbb{F}_{p^2})$, com P de ordem r, e retorna um elemento em $\mathbb{F}_{p^2}$. O algoritmo consiste em duas etapas: a aplicação do Algoritmo de Miller e uma exponenciação final.

O Algoritmo de Miller inicialmente efetua uma multiplicação implícita de P por r, utilizando o algoritmo padrão de multiplicação. O resultado desta multiplicação é ∞, visto que P tem ordem r. Em cada etapa do algoritmo, um valor $m \in \mathbb{F}_{p^2}$ é calculado a partir de uma relação de distância entre a linha ou tangente atual e o ponto Q. Este valor é acumulado e sua avaliação final é a saída do algoritmo. Para garantir que esta saída seja única, o resultado final é elevado a $(p^2-1)/r = (p-1)(p+1)/r$. Se $\bar{m}$ é o *conjugado* de m, a potência $m^{(p-1)}$ pode ser calculada como $\bar{m}/m$, já que para $m \in \mathbb{F}_{p^2}$, temos que $m^{(p-1)} = m^p/m = \bar{m}/m$ [BS04]. A ponto $Q \in E(\mathbb{F}_{p^2})$ pode ser tratado como um ponto no *twist* $E^t(\mathbb{F}_p)$ da curva $E(\mathbb{F}_p)$, porque existe um projeção eficiente $\psi : E(\mathbb{F}_{p^2}) \rightarrow E^t(\mathbb{F}_p)$ dado por $\psi(([-x,0],[0,y])) = (x,y)$ [BLS02]. Isto facilita a implementação de operações sobre Q, como requisitado em protocolos que exigem que Q tenha ordem r [BF01].

O Algoritmo 4.1 detalha o cálculo do emparelhamento. A função $f(A,B,Q)$ avalia o divisor da linha que passa por A e B no ponto Q, a contribuição da adição ou duplicação mais recente

para a variável m. Esta função encontra-se detalhada no Algoritmo 4.2. A exponenciação final pode ser calculada a partir de seqüências de Lucas [BS04], já que o valor de m após a décima linha é unitário.

Algoritmo 4.1 Emparelhamento de Tate [Sco05].

Entrada: Pontos P e Q linearmente independentes, fator r de $\#E(\mathbb{F}_p)$, ordem p do corpo finito subjacente.
Saída: $e(P,Q)$.

1: $m = 1$
2: $A = P$
3: $n = r - 1$
4: **for** $i = \lfloor lg(n) - 2 \rfloor$ **downto** 0 **do**
5: $\quad m = m^2 \cdot f(A,A,Q)$
6: $\quad$ **if** $n_i = 1$ **then**
7: $\quad\quad m = m \cdot f(A,P,Q)$
8: $\quad$ **end if**
9: **end for**
10: $m = \bar{m}/m$
11: $m = m^{(p+1)/r}$
12: **return** m

Algoritmo 4.2 Função f [Sco05].

Entrada: Pontos A, B e Q.
Saída: Avaliação do divisor da linha que passa por A e B no ponto Q.
Nota: λ é o coeficiente angular da linha que passa por A e B. O operador $\leftarrow$ indica a extração das coordenadas do ponto. O valor i denota um não-resíduo quadrático tal que $i^2 = -1$.

1: $\lambda = A.add(B)$
2: $(x,y) \leftarrow A$
3: $(a,d) \leftarrow Q$
4: **return** $y - \lambda(a+x) - di$

4.4.4 Protocolos

Foram implementados protocolos de cifração e assinatura. Ambos os protocolos corrigem a vulnerabilidade ao ataque de um KGC malicioso e têm computação mais eficiente do que os apresentados anteriormente.

Cifração

Este protocolo é seguro contra ataques adaptativos de criptograma escolhido sob o modelo do oráculo aleatório e apresenta uma forma eficiente de cifração, onde apenas um emparelhamento por operação de cifração ou decifração é necessário [CC05a].

Inicializar. Consiste em:

1. Executar IG com entrada k e obter como saída a tupla $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$, onde $\mathbb{G}_1$ e $\mathbb{G}_2$ são grupos de ordem q e $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ é um emparelhamento bilinear;
2. Escolher um gerador arbitrário $P \in \mathbb{G}_1^*$;
3. Selecionar s uniformemente aleatório em $\mathbb{Z}_q^*$ e atribuir $P_0 = sP$;
4. Escolher funções de *hash* criptográficas $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1 \to \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$ e $H_4 : \{0,1\}^n \to \{0,1\}^n$. A parâmetro n é o comprimento em *bits* das mensagens em texto claro;
5. Retornar os parâmetros de sistema $\mathsf{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, n, P, P_0, H_1, H_2, H_3, H_4 \rangle$ e a chave mestre $s \in \mathbb{Z}_q^*$. O espaço de mensagens é $\mathcal{M} = \{0,1\}^n$ e o espaço de criptogramas é $\mathcal{C} = \mathbb{G}_1 \times \{0,1\}^{2n}$.

Extrair. Recebe como entrada um identificador $\mathsf{ID}_A \in \{0,1\}^*$ e executa os seguintes passos para produzir a chave privada parcial para a entidade A:

1. Calcular $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1^*$;
2. Retornar a chave privada parcial $D_A = sQ_A \in \mathbb{G}_1^*$.

Gerar-chaves. Recebe como entrada params e seleciona aleatoriamente $d_A \in \mathbb{Z}_q^*$ como o valor secreto da entidade A. Retorna a chave privada completa $S_A = (d_A, D_A)$ e constrói a chave pública $P_A = d_A P$.

Cifrar. Cifrar uma mensagem $M \in \mathcal{M}$ para a entidade A com identificador $\mathsf{ID}_A \in \{0,1\}^*$ e chave pública P_A requer os seguintes passos:

1. Calcular $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1^*$;
2. Escolher um valor aleatório $\sigma \in \{0,1\}^n$;
3. Fazer $r = H_3(\sigma, M)$;
4. Calcular e retornar o criptograma:

$$C = \langle rP, \sigma \oplus H_2(rP, e(P_0, Q_A)^r, rP_A), M \oplus H_4(\sigma) \rangle.$$

Decifrar. Seja $C = \langle U, V, W \rangle \in \mathcal{C}$. Para decifrar o criptograma utilizando a chave privada S_A:

1. Calcular $\sigma' = V \oplus H_2(U, e(U, D_A), d_A U)$;
2. Calcular $M' = W \oplus H_4(\sigma')$;
3. Fazer $r' = H_3(\sigma', M')$ e testar se $U = r'P$. Se não for o caso, retornar $\perp$ e rejeitar o criptograma;
4. Retornar M' como a decifração de C.

Se $\langle U = rP, V, W \rangle$ é o criptograma de M para a entidade A com chave pública $P_A = d_A P$, o princípio de consistência é satisfeito:

$$\begin{aligned} V \oplus H_2(U, e(U, D_A), d_A U) &= V \oplus H_2(rP, e(rP, sQ_A), d_A U) \\ &= V \oplus H_2(rP, e(P, sQ_A)^r, d_A U) \\ &= V \oplus H_2(rP, e(P_0, Q_A)^r, d_A U) \\ &= \sigma; \\ W \oplus H_4(\sigma) &= M. \end{aligned}$$

Assinatura

O protocolo de assinatura [Goy06] tem segurança provada sob o modelo do oráculo aleatório. Consiste na adaptação de um sistema eficiente de assinaturas baseado em identidades também provado seguro sob o mesmo modelo [BLMQ05].

Inicializar. Consiste em:

1. Executar IG com entrada k e obter como saída a tupla $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$, onde $\mathbb{G}_1$ e $\mathbb{G}_2$ são grupos de ordem q e $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ é um emparelhamento bilinear;
2. Escolher dois geradores arbitrários linearmente independentes $P, Q \in \mathbb{G}_1^*$;
3. Calcular $g = e(P, Q) \in \mathbb{G}_2$;
4. Selecionar s uniformemente aleatório em $\mathbb{Z}_q^*$ e atribuir $Q_0 = sQ$;
5. Escolher funções de *hash* criptográficas $H_1 : \{0,1\}^* \to Z_q^*$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_2 \times \mathbb{G}_2 \to \mathbb{Z}_q^*$;
6. Retornar os parâmetros de sistema $\mathsf{params} = \langle q, \mathbb{G}_1, \mathbb{G}_2, e, g, P, Q, Q_0, H_1, H_2 \rangle$ e a chave mestre $s \in \mathbb{Z}_q^*$. O espaço de mensagens é $\mathcal{M} = \{0,1\}^*$ e o espaço de assinaturas é $\mathcal{S} = \mathbb{G}_1 \times \mathbb{Z}_q^*$.

Extrair. Recebe como entrada um identificador $\mathsf{ID}_A \in \{0,1\}^*$, calcula e retorna a chave privada parcial $D_A = (H_1(\mathsf{ID}_A) + s)^{-1} P \in \mathbb{G}_1^*$;

Gerar-chaves. Recebe como entrada params e seleciona aleatoriamente $d_A \in \mathbb{Z}_q^*$ como o valor secreto da entidade A. Retorna a chave privada completa $S_A = (d_A, D_A)$ e constrói a chave pública $P_A = g^{d_A} \in \mathbb{G}_2$.

Assinar. Assinar uma mensagem $M \in \mathcal{M}$ usando a chave privada S_A da identidade requer os seguintes passos:

1. Escolher um valor aleatório $k \in \mathbb{Z}_q^*$;
2. Calcular $r = g^k \in \mathbb{G}_2$;
3. Fazer $h = H_2(M, \mathsf{ID}_A, P_A, r) \in \mathbb{Z}_q^*$;
4. Calcular $S = (k + hd_A)D_A \in \mathbb{G}_1$;
5. Retornar a assinatura $\langle S, h \rangle$.

Verificar. Verificar uma suposta assinatura $\langle S, h \rangle$ de uma mensagem $M \in \mathcal{M}$ para a identidade ID_A com chave pública P_A requer os seguintes passos:

1. Calcular $r' = e(S, H_1(\mathsf{ID}_A)Q + Q_0)(P_A)^{-h}$;
2. Verificar se $h = H_2(M, \mathsf{ID}_A, P_A, r')$. Se a igualdade for verificada, retornar verdadeiro, caso contrário, retornar falso.

Claramente, o algoritmo de verificação retorna verdadeiro para uma assinatura legítima:

$$\begin{aligned} e(S, H_1(\mathsf{ID}_A)Q + Q_0)(P_A)^{-h} &= e((k+hd_A)D_A, H_1(\mathsf{ID}_A)Q + sQ) \cdot (g^{d_A})^{-h} \\ &= e((H_1(\mathsf{ID}_A)+s)^{-1}P, (H_1(\mathsf{ID}_A)+s)Q)^{k+hda} \cdot (g^{d_A})^{-h} \\ &= e(P,Q)^{(H_1(\mathsf{ID}_A)+s)^{-1}(H_1(\mathsf{ID}_A)+s)(k+hda)} \cdot g^{-hd_A} \\ &= e(P,Q)^{(k+hda)} \cdot e(P,Q)^{-hd_A} \\ &= e(P,Q)^k \\ &= r; \\ H_2(M, \mathsf{ID}_A, P_A, r) &= h. \end{aligned}$$

4.4.5 Resultados

Na Tabela 4.1, encontra-se o custo de computação dos protocolos implementados de cifração e assinatura, em termos de operações de emparelhamento bilinear (e), exponenciação no grupo multiplicativo $\mathbb{G}_2$ (a^x), multiplicação de ponto por escalar no grupo aditivo $\mathbb{G}_1$ (mP), multiplicação no grupo multiplicativo $\mathbb{G}_2$ ($\times$) e aplicações de funções de *hash* (h). As operações

	Operações				
Operação	e	a^x	mP	$\times$	h
Cifração CL-PKC	1	1	2	0	4
Decifração CL-PKC	1	0	2	0	3
Assinatura CL-PKC	0	1	1	0	2
Verificação CL-PKC	1	1	1	1	1

Tabela 4.1: Custo computacional em operações executadas dos protocolos implementados.

encontram-se em ordem decrescente de complexidade. As funções de *hash* foram implementadas a partir da função *SHA-1* e dos Algoritmos 4.3 e 4.4.

A implementação propriamente dita foi realizada utilizando a linguagem C, o compilador GCC 3.4.6 e a biblioteca *OpenSSL*[1] 0.9.8d para aritmética em corpos finitos e curvas elípticas. A aritmética na extensão quadrática $\mathbb{F}_{p^2}$ foi implementada a partir das operações no corpo finito $\mathbb{F}_p$ disponibilizadas pela biblioteca. As otimizações implementadas foram:

- Utilização de coordenadas projetivas;
- Aritmética eficiente na extensão quadrática, pela escolha de um grau de mergulho $k = 2$. Cada operação de multiplicação foi implementada com 3 multiplicações modulares e cada operação de quadrado foi implementada com 2 multiplicações modulares. Foi utilizado o método de Karatsuba para a multiplicação e a relação $(a+bi)^2 = (a+b)(a-b)+2abi$ para o quadrado [Sco05];
- Fator r da ordem da curva com baixo peso de Hamming [Sco05];
- Cálculo de potências do emparelhamento como avaliações de seqüência de Lucas [BS04];
- Compressão do emparelhamento[BS04];
- Eliminação do denonimador do acumulador [BKLS02] no Algoritmo de Miller;
- Cálculo encapsulado do acumulador [CSB05] dentro das operações de duplicação e soma de pontos no Algoritmo de Miller; e
- Isomorfismo entre subgrupo da curva na extensão quadrática $E(\mathbb{F}_{q^2})$ e o *twist* $E^t(\mathbb{F}_q)$ [Sco05].

[1] *The OpenSSL Project:* `http://www.openssl.org`

Na Tabela 4.2, é apresentado o tempo médio de cifrações, decifrações, assinaturas e verificações. O tempo destas operações inclui a aplicação de funções de *hash SHA-1* e chamada da cifras simétrica *AES-128*. O tempo de execução de diversas operações aritméticas implementadas na biblioteca *OpenSSL* é apresentado para referência, em conjunção com o custo de assinaturas e cifras de chave pública comumente utilizados. Todos os tempos apresentados são tomados em uma plataforma Intel Core 2 Duo 2.0 GHz, 1 GB de memória RAM, utilizando apenas um processador pra computação.

Algoritmo 4.3 *Hash* para um inteiro modulo p [BM06].

Entrada: Cadeia de caracteres s de comprimento $|s|$ e inteiro p.
Saída: Inteiro no intervalo $[0, p-1]$.
Nota: A função executa l aplicações da função de *hash SHA-1*, com l escolhido tal que, para entrada aleatória, a saída é quase uniforme no intervalo $[0, p-1]$ com uma não-uniformidade estatística não superior a $\frac{1}{\sqrt{p}}$. O operador $||$ representa a concatenação de duas cadeias de caracteres e o operador $\leftarrow$ representa a decodificação de uma cadeia de caracteres para um inteiro positivo em convenção *big-endian*.

1: $v_0 = 0$
2: $h_0 = \texttt{0x0000000000000000000000000000000000000000}$
3: $l = \lceil \frac{3}{5} \lceil \log_2 p \rceil \rceil$
4: **for** $i = 0$ **to** l **do**
5: $\quad t_i = h_{i-1} || s$
6: $\quad h_i =$*SHA-1*(t_i)
7: $\quad a_i \leftarrow h_i$
8: $\quad v_i = 256^{20} \cdot v_{i-1} + a_i \bmod p$
9: **end for**
10: **return** v_l

A diferença entre os tempos de execução de operações CL-PKC e RSA pode ser justificada pela complexidade dos protocolos de criptografia sem certificados, que exigem maior número de aplicações de funções de *hash* e cifras simétricas e pelo tempo de execução do emparelhamento. O tempo de execução de operações RSA é claramente dominado pela exponenciação modular. Assim, se compararmos diretamente o tempo de decifração e assinatura RSA com o tempo de execução de um emparelhamento bilinear, um emparelhamento corresponde a menos que o quádruplo do tempo de execução de uma exponenciação modular dos protocolos do sistema RSA, sendo viável para aplicações reais. Aprimoramentos nestes resultados experimentais podem ser obtidos a partir da utilização de emparelhamentos mais eficientes [BGhS04].

Os resultados obtidos são compatíveis com os resultados das implementações do emparelhamento de Tate com a mesma escolha de parâmetros publicados [Sco05].

Algoritmo 4.4 *Hash* para subgrupo de ordem r de pontos na curva $E(\mathbb{F}_p)$ [BM06].

Entrada: cadeia de caracteres s, cofator $h = \frac{\#E(\mathbb{F}_p)}{r}$.
Saída: Ponto P de ordem r na curva $E(\mathbb{F}_p)$.
Nota: O operador $||$ representa a concatenação de duas cadeiras de caracteres. A função HASH implementa o Algoritmo 4.3. O operador $\leftarrow$ descomprime o ponto da curva elíptica armazenado na forma comprimida $(x, \{0,1\})$.

1: $i = 0$
2: **repeat**
3: $\quad t = s||i$
4: $\quad x =$HASH(t_i, r)
5: $\quad b = i \bmod 2$ $\{bit$ menos significativo de $i\}$
6: $\quad P \leftarrow (x, b)$
7: $\quad i = i + 1$
8: **until** P seja um ponto válido na curva $E(\mathbb{F}_p)$
9: **return** hP

4.5 Resumo

Neste capítulo, foi apresentado um novo paradigma para definição de sistemas criptográficos de chave pública, chamado Criptografia de Chave Pública sem Certificados (CL-PKC). O objetivo primordial deste novo modelo é utilizar as facilidades advindas de sistemas criptográficos baseados em identidade, mas eliminando as principais desvantagens inerentes a tais sistemas, como a custódia de chaves privadas e os riscos associados. Esquemas eficientes de cifração e assinatura que se baseiam no paradigma foram apresentados com suas respectivas implementações. Ao compararmos suas características, propriedades e desempenho com a PKI tradicional, o modelo de Criptografia Sem Certificados destaca-se como uma alternativa viável para a construção de infra-estruturas modernas de criptografia de chave pública.

OPERAÇÃO	TEMPO DE EXECUÇÃO (μS)
Quadrado na extensão quadrática	2
Multiplicação na extensão quadrática	4
Inversão na extensão quadrática	134
Duplicação encapsulada de ponto na curva elíptica	28
Adição encapsulada de pontos da curva elíptica	38
Exponenciação final com seqüência de Lucas	720
Hash para um inteiro (Algoritmo 4.3)	169
Hash para um ponto na curva (Algoritmo 4.4)	1078
Multiplicação de gerador de subgrupo da curva elíptica†	346
Multiplicação de ponto na curva elíptica†	1690
Emparelhamento	6353
Exponenciação de um emparelhamento comprimido	486
Multiplicação de emparelhamentos comprimidos	25
Geração de Parâmetros (Cifração)	15615
Geração de Parâmetros (Assinatura)	24796
Extração (Cifração)	2727
Extração (Assinatura)	534
Geração de chaves pública e privada	507
Cifração CL-PKC	9654
Decifração CL-PKC	8722
Assinatura CL-PKC	2476
Verificação CL-PKC	8049
Cifração RSA-1024†	45
Decifração RSA-1024†	1099
Assinatura RSA-1024†	1542
Verificação RSA-1024†	87

As rotinas marcadas com (†) são rotinas da biblioteca *OpenSSL*;

As demais rotinas foram implementadas por completo, utilizando aritmética da biblioteca *OpenSSL*.

Tabela 4.2: Custo computacional em tempo de execução dos protocolos implementados. Os tempos são tomados em um processador da plataforma Intel Core 2 Duo 2.0 GHz e representam a média aritmética dos tempos de execução de 1000 cópias da rotina apresentada, com entradas aleatórias.

Capítulo 5

Projeto de serviço de nomes

As diversas redes de anonimização existentes fornecem uma boa base para comunicação anônima, incluindo o suporte à reputação quase sempre exigida para que serviços disponibilizados na rede funcionem a contento. Este requisito é facilmente satisfeito com a utilização de pseudônimos duradouros, que fornecem garantias criptográficas de que a entidade com a qual se comunica é sempre a mesma – desde que não haja comprometimento de pares de chaves associados a pseudônimos. Entretanto, um problema ainda persiste: não existem formas triviais seguras para se *divulgar* um serviço sediado dentro de uma rede anônima.

5.1 Publicação de serviços

A divulgação do endereço em que um serviço reside depende da publicação de um pseudônimo. A partir do pseudônimo, os clientes podem conectar-se ao serviço e verificar a sua autenticidade. Para se publicar um pseudônimo, geralmente existem duas escolhas: distribuir a informação na *Internet* convencional ou no ambiente anonimizado. A primeira opção é perigosa, já que a simples publicação pode deixar rastros que possibilitam a identificação do provedor do serviço. A segunda alternativa exige a presença de um serviço de cadastro que armazena informações a respeito dos serviços anonimizados existentes. Entretanto, é fácil verificar que um serviço de cadastro centralizado é inadequado para o cenário aqui apresentado e a divulgação do próprio serviço é uma instância do problema inicial. Outra característica agravante é o fato de que pseudônimos são formados por cadeias numerosas de *bytes*, de difícil manuseio. O formato fixo da cadeia de *bytes* facilita ainda a filtragem de mensagens de publicação por parte de um adversário. Este problema, de ordem prática, é chamado *problema da publicação de serviços*.

Nenhuma das redes mais populares trata do problema de publicação de serviços satisfatoriamente. Dada sua orientação mais dedicada, *Freenet* e *Gnunet* apenas oferecem recursos de busca baseados em identificadores de arquivo ou palavras-chave [CSWH00, BGHP02]. *Tor* for-

nece a possibilidade de geração de um descritor que possibilita a um usuário comum contatar um serviço sediado dentro da rede anônima. Entretanto, nenhuma solução para a distribuição segura e eficiente deste tipo de informação é sugerida. O descritor ainda sofre dos mesmos problemas de formato que um pseudônimo comum [DMS04]. A rede *I2P* utiliza um arquivo centralizado que mapeia nomes amigáveis a pseudônimos na rede. Entretanto, esta abordagem tem inúmeras desvantagens, como baixa escalabilidade, ponto único de falha e pouca resistência contra ataques poderosos: basta substituir um registro do arquivo central para que um mapeamento falso seja rapidamente propagado para toda a rede.

O objetivo deste capítulo é apresentar o projeto de um serviço de nomes que pretende minimizar o problema da publicação de serviços. O serviço de nomes projetado é genérico o suficiente para ser utilizado por qualquer rede de anonimização estruturada, especialmente a rede projetada no Capítulo 3.

5.2 Requisitos de projeto

O problema da publicação de serviços é análogo ao problema da publicação de endereços IP na *Internet*. Assim, uma solução satisfatória para o problema deve ser análoga ao *Domain Name System* (DNS) [Moc87] presente na *Internet* e envolver o projeto de uma extensão para a rede de anonimização, que atue como um serviço de nomes descentralizado. O serviço de nomes é o componente responsável por armazenar dados a respeito de endereços e domínios em uma base distribuída e disponível para qualquer cliente. Também é responsável por realizar também a tradução de nomes de domínio em endereços que podem ser repassados diretamente para a camada de rede e permite a transmissão de informação técnica de forma amigável [Moc87].

A presença de um serviço de nomes em uma rede anônima, com a função de armazenar e converter domínios em pseudônimos válidos, traz vantagem idêntica. Divulgar um serviço sediado no endereço `no-censorship.service`, por exemplo, é infinitamente mais fácil que o seu pseudônimo correspondente `0x182f084ebbcf5f3dc425d1b147409fba`. Publicar na *Internet* convencional um endereço na primeira forma é mais eficaz, porque é difícil para um adversário diferenciar domínios que confundem-se a domínios comuns da *Internet*. Ao mesmo tempo, pseudônimos na segunda forma são necessários para se encontrar o serviço propriamente dito na rede. Uma entrada adequada na base de dados do serviço de nomes pode unificar estas duas referências a um mesmo recurso de forma simples e transparente.

Outro fator favorável é que a ligação entre o serviço de nomes e os serviços sediados é pouco acoplada. Os riscos para os provedores de serviço são menores caso haja necessidade de se divulgar na *Internet* o endereço de um índice em que seu serviço encontra-se registrado: o operador do serviço de nomes não tem ligação direta com os serviços registrados e a quebra do seu anonimato não traz implicações diretas para os provedores de serviço. A construção de sistemas indexadores passa também a ser mais simples, já que pode-se adicionar níveis à

hierarquia, permitindo aos servidores de nomes fornecer domínios para outros servidores de nomes. A hierarquização reduz a quantidade de informação sob responsabilidade de cada servidor, suavizando requisitos de processamento, armazenamento e transmissão. A adição de níveis adicionais à hierarquia traz ainda a vantagem de se enfraquecer cada vez mais a ligação entre índices e serviços, permitindo a divulgação segura de endereços dos componentes mais altos da cadeia de forma menos cuidadosa. A característica hierárquica e a delegação de serviços exibe uma analogia próxima às infra-estruturas de chave-pública e às cadeias de validação de certificados digitais, expondo questões em comum como cadastro, autenticação, renovação e revogação de registros. Procedimentos seguros para realizar estas operações devem ser fornecidos pelo servidor de nomes, com autenticação adequada.

A gama de qualidades trazidas por um serviço de nomes estende-se além do ganho de usabilidade para usuários comuns e da facilidade de publicação de serviços. Um efeito colateral planejado é o suporte a identidades temporárias, bastando-se variar o pseudônimo ligado ao nome permanente e efetuar atualização do registro quando necessário. Como requisitos de projeto para este serviço, devemos citar:

- Independência relativa entre serviço de nomes e rede de anonimização, para que alterações em um dos componentes não force alterações no outro;
- Simplicidade e escalabilidade dos protocolos;
- Aplicação de primitivas criptográficas aprovadas pela comunidade;
- Verificação rigorosa das técnicas utilizadas para que conservem as propriedades de anonimização fornecidas pela rede; e
- Adequação ótima das premissas de segurança do serviço de nomes ao contexto anônimo em que é aplicado.

As contribuições que um serviço de nomes com estas características traz para o cenário de anonimização são:

- Ganho em usabilidade, sem qualquer comprometimento das premissas de anonimato;
- Agrupamento seletivo de serviços similares em um mesmo servidor de nomes, facilitando a busca por um determinado tipo de serviço;
- Utilização de técnicas criptográficas em um contexto inédito, já que nenhuma das redes de anonimização existentes hoje conta com o suporte de um serviço de nomes; e
- Tratamento do problema da publicação de serviços, questão fundamental para a usabilidade de redes de anonimização de tráfego.

5.2.1 Arquitetura

A arquitetura do serviço de nomes deve ser obviamente descentralizada. É desejável também que apenas uma fração dos nós da rede executem servidores de nomes. O cenário que se deseja atingir é o esboçado anteriormente, onde uma estrutura hierárquica de registro, iniciada a partir de múltiplas raízes, gerencia o cadastro de diversos servidores de nomes adicionais ou de serviços propriamente ditos. Isso permite às raízes simples divulgação na *Internet*, sem a necessidade de anonimização do publicante.

O protocolo de resolução de nomes deve funcionar corretamente independentemente da disponibilidade dos nós que compõem a hierarquia do serviço, exigindo-se apenas a disponibilidade do servidor de nomes onde o endereço sendo buscado encontra-se registrado. Este aspecto trata da possibilidade de substituição ou fechamento de parte da estrutura do serviço e exige que a funcionalidade de um servidor de nomes não seja prejudicada em face da indisponibilidade de servidores de nível mais alto na hierarquia.

5.2.2 Criptografia

O serviço de nomes deve oferecer meios para que um provedor de serviço cadastrado possa autenticar-se seguramente com o serviço de nomes para atualizar o pseudônimo de sua entrada correspondente ou realizar outras operações. A necessidade de autenticação é ainda maior quando o par de chaves utilizado como pseudônimo tem curto tempo de vida, já que os clientes necessitam de uma garantia estável de que o provedor de serviço mantém-se o mesmo. É desejável também que os clientes possam verificar a autenticidade dos servidores de nomes com os quais se comunicam. Para isso, um mecanismo de autenticação foi projetado e agregou um par de chaves adicional para autenticação dos provedores de serviço. Isto permite isolar as premissas de segurança relacionadas ao anonimato das premissas ligadas ao serviço de nomes.

A natureza do serviço é bastante similar ao esquema de certificação digital, o que aponta para um modelo de autenticação e verificação análogo. Entretanto, as desvantagens que as infra-estruturas de chave pública apresentam hoje (dificuldade de gerência e validação de certificados [LQ04]) e a mínima acoplação entre entidades anônimas e suas respectivas identidades reais responsáveis, sugerem a aplicação de um modelo mais relaxado, que traga simplicidade de gerência, agilidade nas transações e premissas de segurança plausíveis e compatíveis com o contexto considerado. Assim, a utilização de sistemas criptográficos baseados em identidades como primitiva criptográfica fundamental parece promissora. A viabilidade da utilização de Criptografia Baseada em Identidades para segurança do próprio DNS também tem sido estudada [Cha03].

Entretanto, a custódia de chaves inerente a sistemas baseados em identidades é incompatível com a funcionalidade do serviço de nomes. O paradigma de Criptografia de Chave Pública Sem Certificados, que elimina essa característica peculiar, tem aplicação ideal neste contexto.

5.3 Serviço de nomes

As seções a seguir detalham um serviço de nomes que respeita os requisitos de projeto anteriormente delineados e tenta fornecer funcionalidade compatível com as contribuições esperadas. Por simplificação, apenas hierarquias de nível único serão consideradas. Generalizar o serviço de nomes para múltiplos níveis é trivial, mas por enquanto desnecessário para o cenário anonimizado, onde não existem países, corporações ou entidades que necessitem de hierarquias complexas.

O funcionamento do serviço de nomes é a seguir examinado sob o ponto de vista de cada uma das entidades envolvidas: servidores de nomes, provedores de serviço e clientes.

5.3.1 Considerações iniciais

O serviço de nomes utiliza a funcionalidade de tabela de *hash* distribuída de um sistema estruturado como base de dados. Outros serviços de nomes têm sido propostos recentemente para aproveitar as propriedades de eficiência e balanceamento de carga de tabelas de *hash* distribuídas [TJ02, RS04b, BLR$^+$04].

Nas discussões posteriores, o conjunto de chaves $h_{1-16}(i)$ indica os 16 resultados da aplicação sucessiva de uma função de *hash* criptográfico ao identificador i. Este conjunto de chaves é utilizado para se efetuar operações de armazenamento e recuperação na tabela de *hash* distribuída. Múltiplas chaves são úteis para se dificultar a substituição de informação legítima por informação forjada [MM02]. Como o armazenamento de conteúdo é permitido a qualquer entidade, a integridade da informação que persiste também depende de monitoramento e atualização freqüentes. No serviço de nomes, uma operação de armazenamento preenche todas as 16 chaves com conteúdo e uma operação de recuperação obtém um subconjunto aleatório de entradas no intervalo $[1,16]$.

A notação $Sig(i,I)$ indica a assinatura da informação i sob a chave da entidade I sob o paradigma de Criptografia de Chave Pública Sem Certificados (CL-PKC).

5.3.2 Servidor

O papel de um serviço de nomes restringe-se a mapear um identificador de serviço ao pseudônimo do provedor do serviço correspondente, imerso no ambiente anônimo. O papel específico de cada servidor de nomes é executar esta funcionalidade para um subconjunto dos identificadores possíveis. Ou seja, cada servidor responsabiliza-se pela fração dos nomes inclusos no seu subconjunto [Moc87]. Idealmente, cada subconjunto tem ligação direta com o tipo de serviço fornecido. A presença de servidores de nomes distintos é necessária para que se controle a alocação de nomes; caso contrário, teríamos um sistema caótico, onde a permissão de uso de cada nome seria obtida apenas por uso da força. Para diferenciar os nomes amigáveis aqui

considerados dos nomes comuns da *Internet*, os nomes da rede anônima tem sufixo `.service`. Observe que este requisito não necessariamente facilita a detecção de sua publicação por um adversário, já que o nome na rede anônima pode ser publicado sem o sufixo, se o contexto indicar que se trata de um serviço anonimizado.

Para exercer o seu papel, um servidor de nomes deve ter a capacidade de delegar nomes para entidades. Esta delegação deve poder ser provada para terceiros, para que os usuários de um serviço particular possam confirmar que aquele nome foi realmente delegado para o provedor cadastrado. Delegação criptográfica é naturalmente aplicada em casos assim e, portanto, cada servidor de nomes atua como um *Centro de Geração de Chaves*, contando com um conjunto de parâmetros de sistema criptográfico para assinatura e as capacidades de registro e revogação associadas [Cha03].

Um índice é utilizado para reunir informação a respeito de todos os serviços de nomes cadastrados. Este índice não é um requisito, mas uma facilidade, por permitir aos usuários a pesquisa dos servidores de nomes cadastrados. Um índice desta natureza pode ser disponibilizando tanto na *Internet* convencional quanto no ambiente anônimo, como discutido anteriormente. Um servidor de nomes especial, chamado servidor-raiz, é responsável por reconhecer o cadastro dos servidores de nomes e assinar os parâmetros de sistema dos servidores de nomes com um par de chaves confiado por todos os usuários do sistema. O endereço do servidor-raiz dentro da rede anônima é `root.service`. A chave pública do servidor-raiz, que permite a verificação de assinaturas do par de chaves confiado, é armazenada nas chaves $h_{1-16}(\texttt{key.root.service})$ da rede estruturada. Recomenda-se que o servidor-raiz forneça formas alternativas seguras de distribuição da chave, como a publicação direta na *Internet* convencional.

Estabelecimento de um novo servidor

O procedimento para estabelecimento de um novo servidor de nomes é:

1. Escolher um domínio D e pseudônimo S;
2. Gerar parâmetros de sistema para um sistema criptográfico CL-PKC (params);
3. Gerar um par de chaves CL-PKC relativo à identidade `root.D.service`. Este par de chaves é utilizado para revogação de serviços cadastrados;
4. Contatar o servidor-raiz a partir do pseudônimo S. O mapeamento estático entre o servidor-raiz e o pseudônimo S deve ser disponibilizado pelo servidor-raiz na *Internet* comum;
5. Cadastrar o servidor de nomes no índice do servidor-raiz e solicitar a assinatura $Sig((S, D, \mathsf{params}), \texttt{root.service})$ do servidor-raiz.

A assinatura do servidor-raiz funciona como uma delegação de privilégios ao servidor de nomes, conferindo o controle sob o domínio D [Cha03]. O resultado desta assinatura deve ser armazenado e distribuído pelo servidor de nomes para que os clientes, ao entrarem em contato pela primeira vez com o servidor, possam verificar a delegação e considerar o servidor de nomes como um servidor legítimo do domínio D. As informações de delegação são distribuídas pelo servidor de nomes a partir das chaves $h_{1-16}(\texttt{key.D.service})$. Além disso, o índice do servidor-raiz distribui pacotes de assinaturas para armazenamento local por parte dos clientes. Para contatar um servidor de nomes, um provedor de serviço precisa obter a assinatura do servidor-raiz, verificar sua validade e utilizar o pseudônimo S para conexão a partir da rede anônima subjacente. Cada servidor de nomes fornece uma interface adequada para disponibilização das operações de registro, consulta e revogação.

Assim, sempre que for inicializado, um servidor de nomes deve:

1. Armazenar a assinatura do servidor-raiz nas chaves $h_{1-16}(\texttt{key.D.service})$;
2. Disponibilizar interface para operações de registro, consulta e revogação na interface;
3. Disponibilizar a delegação recebida do servidor-raiz na interface;

O estabelecimento e inicialização de um novo servidor encontra-se esboçado na Figura 5.1. As setas tracejadas indicam instâncias de comunicação anônima.

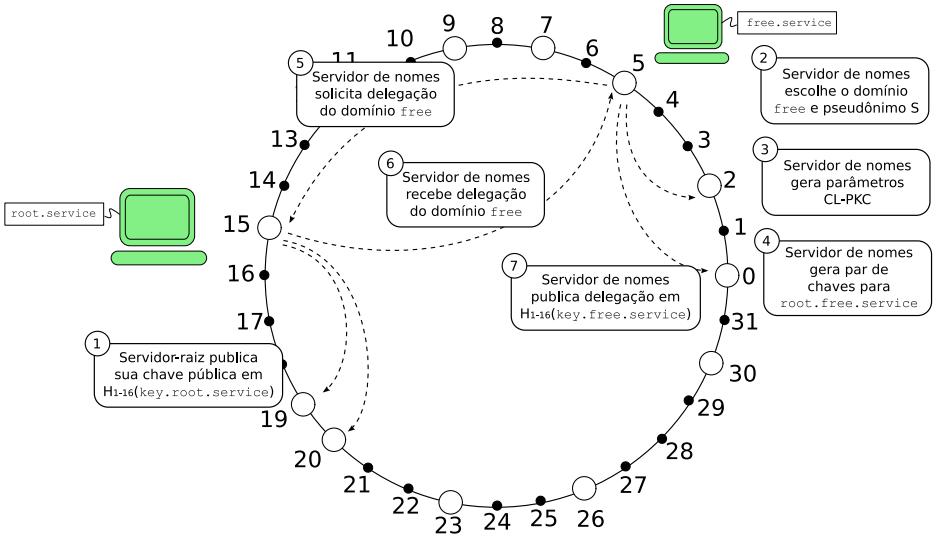


Figura 5.1: Estabelecimento de um novo servidor de nomes.

A utilização paralela de múltiplos servidores-raiz também é possível. Isto garante a completa descentralização do serviço de nomes e total independência de autoridades centrais.

Revogação de domínio

A revogação imediata R do domínio D no instante de tempo t por parte do serviço de nomes com pseudônimo S implica na distribuição da assinatura $Sig((S,D,R,t\mathsf{params}),\texttt{root.service})$ pelo servidor-raiz nas chaves $h_{1-16}(\texttt{key.D.service})$. O servidor de nomes recebe uma cópia desta assinatura como prova de revogação e fornece a assinatura como confirmação de revogação a assinatura $Sig((S,D,R,t,\mathsf{params}),\texttt{root.D.service})$. Estas assinaturas podem ser utilizadas para resolução de disputas, pois as provas de revogação recebidas pelo servidor-raiz são distribuídas publicamente em seu índice dedicado.

Assim que o domínio for reutilizado por um outro servidor de nomes com pseudônimo S', a assinatura com o novo pseudônimo $Sig((S',D,\mathsf{params}'),\texttt{root.service})$ pode voltar a ser distribuída nas chaves $h_{1-16}(\texttt{key.D.service})$ pelo servidor substituto. A distribuição da nova delegação impede a utilização da chave antiga como chave legítima, já que os clientes verificam o estado da delegação sempre que precisarem utilizar o domínio D.

A alternância de distribuição de delegação entre o servidor-raiz e o servidor de nomes substituto garante que a informação nova sempre está disponível. Ou seja, o servidor-raiz toma a responsabilidade de distribuir a revogação assim que ela acontece e cede a responsabilidade da distribuição da nova delegação para o servidor de nomes substituto após o cadastro.

Validação da assinatura do servidor-raiz

Observe que a obtenção de assinaturas corretas é fundamental para que os clientes contatem o servidor de nomes que recebeu a delegação legítima. Sempre que for necessário validar uma delegação de um servidor de nomes, a assinatura é recuperada da base de dados distribuída e validada uma única vez. Após ser validada é considerada como *confiada*. O pseudônimo presente nesta assinatura passa a ser utilizado estaticamente para economizar recuperações e validações da assinatura do servidor-raiz em todos as traduções subseqüentes ao servidor de nomes. Entretanto, a assinatura deve ser recuperada e verificada com alguma freqüência e em segundo plano para que revogações de domínio e atualizações de pseudônimo sejam detectadas. A detecção de qualquer revogação provoca a eliminação imediata da cópia local confiada, para que a assinatura contendo o novo pseudônimo possa ser incorporada sem problemas. A detecção de mudança de pseudônimo sem a observação de revogação provoca uma disputa, que deve ser solucionada manualmente pelo usuário, utilizando as informações disponíveis no índice do servidor-raiz, que incluem as provas de uma possível revogação. Isto limita a eficácia de ataques de substituição dos parâmetros de sistema de um servidor de nomes nas chaves correspondentes.

5.3.3 Provedor

Ao escolher um domínio D compatível com as suas atividades, o provedor recupera os parâmetros de sistema do servidor escolhido nas chaves $h_{1-16}(\texttt{key.D.service})$ e verifica a validade da delegação do domínio D, conforme discussão anterior. Para registrar um serviço que fornece, o provedor deve:

1. Contatar o servidor de nomes a partir da interface;
2. Escolher um identificador n e pseudônimo N;
3. Gerar uma chave pública P_N;
4. Extrair do servidor de nomes S uma chave privada parcial confinada à chave pública P_N;
5. Gerar um par de chaves (P_N, S_N) relativo à identidade `n.D.service`. Qualquer assinatura deste par de chaves pode ser validada apenas com a chave pública P_N e os parâmetros de sistema de S, sem a necessidade de certificados;
6. Calcular a tupla $\{P_N, Sig(N, \texttt{n.D.service})\}$ e transmitir o resultado para o servidor de nomes.

O servidor de nomes responsável distribui a informação de mapeamento $\{P_N, Sig(N, \texttt{n.D.service})\}$ entre n e N a partir das chaves $h_{1-16}(\texttt{n.D.service})$. O procedimento para o registro de um serviço encontra-se detalhado na Figura 5.2.

Revogação de serviço

A revogação R de um identificador de serviço no instante de tempo t pode ser solicitada a partir da interface do servidor de nomes. Ao invés de revogar o identificador associado ao provedor de serviço, o servidor de nomes revoga apenas o privilégio do provedor de serviço em utilizar o identificador [HHSI04].

Ao revogar um identificador n no instante de tempo t, duas provas da revogação são produzidas: $\{P_N, Sig((N,R,t), \texttt{root.D.service})\}$ de posse do provedor do serviço; e $\{P_N, Sig((N,R,t), \texttt{n.D.service})\}$, armazenada pelo servidor de nomes. O servidor de nomes inicia a distribuição da informação de revogação assim que possível, e as buscas posteriores pela chave $h_{1-16}(\texttt{n.D.service})$ devem retornar $\{P_N, Sig((N,R,t), \texttt{root.D.service})$, efetivamente revogando a chave antiga. O registro do identificador por um novo provedor de serviço N' provoca a distribuição da nova informação $\{P'_N, Sig(N', \texttt{n.D.service})\}$, gerada por N'.

Este mecanismo impede que o servidor de nomes revogue maliciosamente o serviço sem ser detectado, já que cada uma das partes recebe uma prova de revogação. Também impede que o servidor de nomes adultere o mapeamento para desviar as requisições ao serviço N para

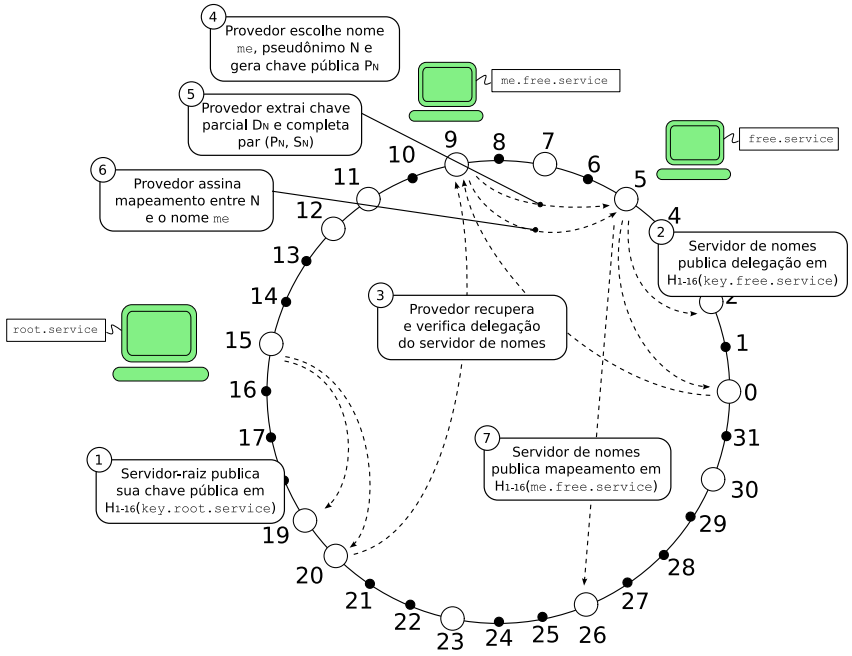


Figura 5.2: Registro de um serviço utilizando o serviço de nomes.

um pseudônimo N'' com mesmo endereço e chave pública distinta. Como as chaves privadas parciais são confinadas às chaves públicas, a presença de assinaturas válidas para pseudônimos diferentes com mesmo identificador n implica em ação maliciosa do servidor de nomes [AR05].

Para se desviar as requisições para um pseudônimo N'' com mesmo endereço e mesma chave pública, é necessário gerar uma assinatura com a chave privada exclusiva do provedor de serviço. Neste caso, pode-se concluir que houve comprometimento deste par de chaves e recomenda-se a revogação imediata pelo titular legítimo.

É importante salientar que um provedor de serviço só deve fornecer sua assinatura da revogação após receber a prova de revogação do servidor de nomes. Isto dá a capacidade ao provedor de serviço de provar que solicitou a revogação para terceiros, em caso de disputa. O servidor de nomes, por sua vez, só deve registrar um novo pseudônimo para um nome n após receber a prova de revogação do pseudônimo antigo. Isto exime de culpa o servidor de nomes que fornecer uma prova de revogação para um provedor de serviço e não receber a confirmação de revogação correspondente do mesmo. Em situações assim, o provedor de serviço poderia acusar maliciosamente o servidor de nomes de registrar outro provedor de serviço para o nome n, ou seja, de extrair uma nova chaves privadas parcial para um mesmo nome n, sem revogar a chave anterior.

5.3.4 Cliente

O cliente utiliza o serviço de nomes para resolver os endereços que recebe em pseudônimos confiáveis. O procedimento de resolução de um endereço `n.D.service` consiste em:

1. Obter os parâmetros de sistema e a delegação do servidor de nomes armazenados nas chaves $h_{1-16}(\texttt{key.D.service})$;
2. Verificar se os parâmetros de sistema são confiados. Caso sejam, comparar a assinatura obtida com a assinatura local e atestar a validade. Caso não sejam, verificar a assinatura recuperada utilizando os parâmetros do servidor-raiz armazenados localmente e depositar confiança se a assinatura for válida. Em caso de disputa, requisitar intervenção manual;
3. Obter o mapeamento armazenado na chave $h_{1-16}(\texttt{n.D.service})$;
4. Verificar o mapeamento: caso válida, determinar o pseudônimo N ligado ao endereço; caso inválida, reportar erro na resolução do endereço;
5. Proceder com a conexão para o pseudônimo N.

A interação de um cliente com o serviço de nomes é ilustrada na Figura 5.3. O cliente com identificador 23 utiliza o serviço de nomes para resolver o endereço `me.free.service` no pseudônimo do nó com identificador 9.

O serviço de nomes foi construído de forma que todas as operações sensíveis para o seu funcionamento são de responsabilidade dos servidores de nomes. Isso diminui os requisitos de comunicação dos servidores-raiz e dos serviços propriamente ditos. Além disso, o custo de resolução de nomes é de um acesso à base de dados, quando os parâmetros de sistema de um servidor de nomes são considerados como confiados. Assim, ao se resolver um nome ligado a um servidor de nomes nunca antes visto, o custo é de duas recuperações na base de dados distribuída.

É importante observar que essa arquitetura descentraliza e minimiza os custos de comunicação e armazenamento. Em particular, não existe a necessidade de se utilizar certificados: de posse de uma cópia legítima da chave pública de um servidor-raiz, e de parâmetros de sistema legítimos de um servidor de nomes (validados a partir da chave pública do servidor-raiz), todos os serviços cadastrados no servidor de nomes particular podem ser utilizados facilmente e de maneira transparente. Os custos de comunicação e armazenamento são ainda diminuídos pela característica de balanceamento de carga da base de dados subjacente. O número de assinaturas também é controlado: para o caso comum, onde o servidor de nomes é conhecido e os seus parâmetros de sistema já foram validados, apenas uma assinatura precisa ser verificada para traduzir um identificador de serviço ao pseudônimo associado.

Os requisitos de comunicação do serviço de nomes podem ainda ser reduzidos com a utilização de replicação e *caching* [CK01, RS04a] na base de dados distribuída.

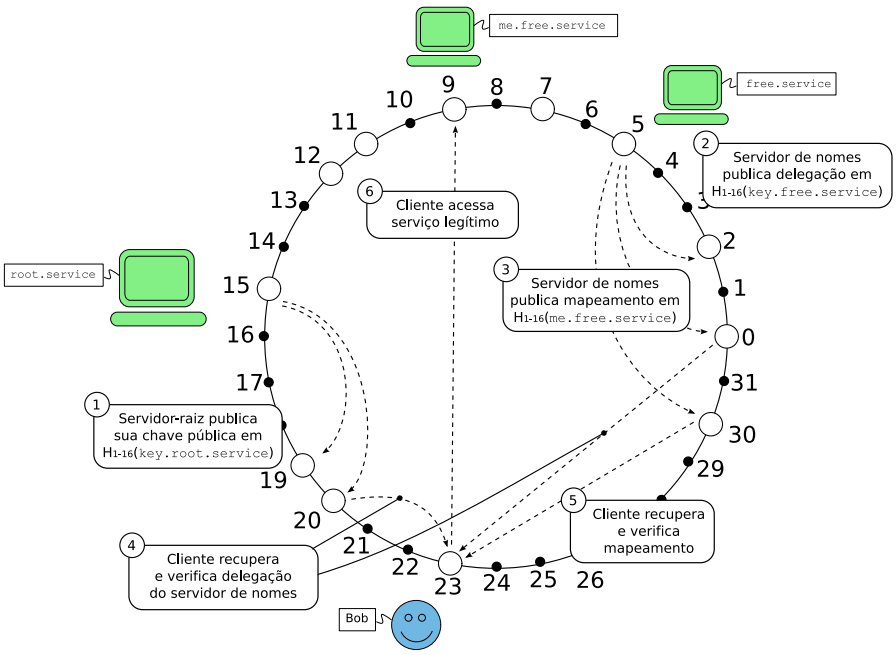


Figura 5.3: Utilização do serviço de nomes para acesso a um serviço legítimo.

5.4 Resumo

Neste capítulo, foi apresentado um serviço de nomes eficiente que faz uso da estrutura subjacente de uma rede estruturada para armazenar e recuperar informações de delegação e revogação. Em particular, a revogação suportada pelo serviço de nomes é *imediata*. O funcionamento do serviço de nomes sob o ponto de vista de todas as entidades participantes foi apresentado e garantias de funcionamento foram discutidas. Em todos os cenários de ataque considerados, uma única entidade sempre pode ser apontada como culpada com comprovação criptográfica do ato malicioso. O serviço de nomes é capaz de alcançar estabilidade, requerendo intervenção manual dos usuários e provedores de serviço apenas em casos de disputa. Entretanto, é evidente que uma limitação deste serviço de nomes é a ciência por parte dos participantes dos detalhes de funcionamento. Esta abordagem foi escolhida conscientemente, pelo fato de que tornar detalhes importantes transparentes ao usuário quase sempre facilita a montagem de ataques de difícil detecção.

Capítulo 6

Implementação

A implementação da nova rede de anonimização projetada neste trabalho foi realizada por meio da modificação direta de sua rede antecessora, a *Libfreedom* [AS05]. A *Libfreedom* foi projetada como uma rede de anonimização não-estruturada, construída especificamente para facilitar a manipulação dos seus mecanismos internos. A política de roteamento da *Libfreedom* baseia-se no comportamento de formigas, pertencendo à família de protocolos *Ants*.

A nova implementação foi batizada de *Kurupira*[1] e representa uma versão totalmente modernizada e reformulada da *Libfreedom*. *Kurupira* é uma plataforma de comunicação anônima genérica com organização estruturada, que tem como requisitos de projeto a divisão em componentes das técnicas de anonimização, protocolos e serviços, o fornecimento de uma interface de programação simples (*sockets* POSIX [SW95]) e uma busca por relação ótima entre anonimato e desempenho. A implementação fornece comunicação anônima genérica e suporta a utilização da estrutura subjacente para funcionalidades específicas, como distribuição de arquivos e publicação de conteúdo.

A rede utiliza a política de roteamento para a topologia *Koorde-tac-16*, projetada no Capítulo 3 e conta com o suporte do *Kurupira Name Service* (*KNS*), uma implementação do serviço de nomes projetado no Capítulo 5.

6.1 Rede de anonimização *Kurupira*

A rede *Kurupira* orienta-se a partir do modelo de referência OASIS (*Open and Anonymous Systems Interconnection Specification*) [AS05] para redes de anonimização, baseado no padrão OSI [dJ83]. Este modelo possui cinco camadas (Figura 6.1):

[1] Curupira é uma entidade do folclore brasileiro que protege as florestas e tem os pés virados para trás, para que suas pegadas não revelem o sentido em que caminha.

Figura 6.1: Modelo em camadas OASIS para redes de anonimização.

- *Camada de infra-estrutura:* é representada tipicamente por uma pilha de protocolos que realiza algumas das camadas do modelo OSI [dJ83], permitindo a dois equipamentos quaisquer que desejam conectar-se à rede anônima trocar mensagens;

- *Camada de enlace:* mantém a conectividade entre os nós da rede, estabelecendo as ligações por onde as mensagens produzidas e consumidas por aplicações possam trafegar. Os princípios de igualdade de mensagens e injeção de ruído são aspectos importantes que guiam o projeto de camadas de enlace;

- *Camada de rede:* estabelece um esquema de endereçamento que atribui unicamente um identificador a cada nó conectado. Um algoritmo agregado de roteamento utiliza as informações locais de roteamento presentes em cada nó para transportar a mensagem de sua origem a seu destino. A técnica de indireção é utilizada neste nível, sendo responsabilidade vital desta camada evitar ao máximo a possibilidade de ataques de correlação temporal;

- *Camada de transporte:* não exerce qualquer função de anonimização, tendo características similares a camadas de transporte comuns (controle de fluxo, seqüenciamento, retransmissão);

- *Camada de aplicação:* utiliza os serviços disponibilizados pelas camadas inferiores para atuar em um cenário anônimo. Destaca-se a importância de se ter um protocolo de aplicação que não forneça informações que comprometam o anonimato do nó.

6.1.1 Pilha de protocolos

Definida a arquitetura, cada camada é concretizada em um protocolo específico. São eles KLP (*Kurupira Link Protocol*), KNP (*Kurupira Network Protocol*), KUP (*Kurupira Unreliable Transport Protocol*) e KRP (*Kurupira Reliable Transport Protocol*).

Protocolo de enlace

O protocolo de enlace é transportado por túneis DTLS (*Datagram-TLS*) [MR04], uma versão do protocolo TLS *Transport Layer Security* [DR06] especialmente projetada para utilizar UDP como camada de transporte. A utilização de UDP é motivada pela baixa latência de transmissão e baixa carga de manutenção das conexões. O protocolo DTLS fornece suporte a cifração e autenticação, *padding* para igualdade entre pacotes e proteção contra influências externas, como ataques de repetição.

O principal objetivo do protocolo de enlace é prover serviços primitivos de transferência de mensagens para as camadas superiores através de conexões seguras que inviabilizam a escuta passiva. As conexões da camada de enlace são cifradas com chaves estabelecidas por um acordo de chaves *Diffie-Hellman* [DH76] sobre curvas elípticas, com tempo de vida configurável. O estabelecimento e manutenção da topologia estruturada *Koorde-tac-16* é responsabilidade direta da camada de enlace. Os identificadores na topologia são gerados a partir da aplicação de uma função de *hash* a um par de chaves opcionalmente temporário.

A implementação fornece a possibilidade de se configurar endereços IP de máquinas confiadas que sempre serão contactados para a entrada na rede. Para promover obscurecimento, o protocolo ainda injeta ruído na forma de mensagens de controle, que são indistinguíveis de pacotes comuns de conteúdo.

Protocolo de rede

O protocolo de rede é encapsulado dentro de *frames* do protocolo de enlace. A função do protocolo de rede é utilizar os serviços fornecidos pela camada de enlace para rotear os pacotes produzidos pelas camadas de transporte superiores. Oferece como propriedades:

- Esquema de identificação que previne ataques de *man-in-the-middle* (cada nó conta com um par de chaves de curvas elípticas e sua identificação é obtida pela aplicação da função de *hash* SHA-1 à chave pública);
- Algoritmo de roteamento que implementa roteamento randomizado e canais de resposta múltiplos e independentes. O comprimento esperado do caminho aleatório é configurável a partir da parametrização da probabilidade de encaminhamento p_f. O valor inicial da probabilidade de encaminhamento é padronizado em $p_f = 0.9$, para suportar tranqüilamente uma rede de anonimização com pelo menos 2^{20} participantes;

- *Handshake* utilizando criptografia assimétrica para derivação de chaves;
- Determinação dinâmica dos algoritmos criptográficos utilizados;
- Sigilo na comunicação e igualdade entre mensagens, impossibilitando que roteadores intermediários empreguem filtros de conteúdo e garantindo possibilidade de repudiação do tráfego roteado;
- Autenticação de origem e verificação de integridade dos dados; e
- Delegação de fragmentação para a camada de transporte, para que a mesma utilize procedimentos ótimos dependentes do contexto em que atua.

A identificação no protocolo de rede anônima implementa um *pseudônimo*, a ser utilizado para referenciar um nó particular conectado à rede. A ligação entre este pseudônimo e a identificação da camada de infra-estrutura subjacente representa o cerne da anonimização fornecida pela pilha.

Protocolos de transporte

A pilha conta com um protocolo de transporte não-confiável e um protocolo de transporte confiável. O protocolo não-confiável pode ser visto como análogo ao UDP, já que ambos possuem características similares em seus respectivos domínios. Entretanto, algumas das responsabilidades descritas no UDP não são necessárias no protocolo, e vice-versa, pois as camadas inferiores de ambos são bastante diferentes. As duas principais diferenças entre as camadas KUP e UDP são:

- *Controle de integridade:* ao contrário do IP e do UDP, o protocolo não-confiável não necessita exercer nenhum controle de integridade, pois o protocolo de rede garante tanto a integridade como a autenticidade dos pacotes; e
- *Fragmentação:* como a camada de rede não suporta fragmentação, os datagramas de transporte não-confiável devem possuir campos que controlem o tamanho total do datagrama e que representem o deslocamento dos seus fragmentos em relação ao pacote original.

O protocolo de transporte confiável é similar ao TCP, porém possui simplificações em alguns aspectos que não se aplicam ao cenário em que atua. As diferenças entre KRP e TCP são:

- *Controle de integridade:* o protocolo de transporte confiável isenta-se dessa responsabilidade, já que controle de integridade e autenticidade são fornecidos pelo protocolo de rede;

- *Fragmentação:* o controle de divisão e remontagem do *payload* a ser transmitido é função da camada de transporte. No protocolo de transporte não-confiável, uma fragmentação similar à clássica, aplicada pelo IP, é realizada. Como este protocolo de transporte confiável possui seqüenciamento embutido (existe continuidade entre pacotes), pode-se evitar a necessidade de fragmentação, enviando sempre um *payload* de comprimento adequado ao tamanho máximo de segmento fornecido pela camada de rede; e

- *Seqüenciamento*: como o tráfego é cifrado na camada de rede, nenhum cuidado precisa ser tomado para a escolha dos valores iniciais de seqüenciamento da conexão, já que o protocolo só permite a injeção de pacotes por detentores da chave negociada em *handshake*. Capturar esta chave durante o *handshake* da camada de rede envolve quebra de sistemas criptográficos robustos. O perigo de que pacotes atrasados ou ataques de repetição prejudiquem o funcionamento de uma conexão também é inexistente, já que aplica-se renovação regular das chaves utilizadas.

6.1.2 Garantias de anonimato

Para situar a rede *Kurupira* dentre as várias implementações de redes de anonimização, devemos analisar o grau de anonimato aproximado que a rede fornece, na presença de adversários com diversas características.

Anonimato de envio

O grau de anonimato de envio contra adversários internos está intimamente relacionado à probabilidade de encaminhamento p_f utilizada na parte randomizada do roteamento. Particularmente, quando a probabilidade de encaminhamento é $p_f > 0.5$, uma propriedade emerge: para qualquer nó que receba uma mensagem, a probabilidade do vizinho que a encaminhou não ter originado a mensagem é maior que a probabilidade do vizinho tê-la originado. Desta forma, a utilização de caminhos aleatórios é ideal para construção de sistemas com grau de anonimato de inocência provável [RR98].

Um limite inferior mais restrito pode ser calculado para se examinar as condições em que a rede fornece grau de anonimato de inocência provável. Seja uma rede com n nós, dos quais c são controlados por adversário atuando em conluio. Por simplificação, considere que os nós controlados por adversário estejam distribuídos igualmente na rede. Define-se o *fator de comprometimento* $f = \frac{c}{n}$ como a razão entre nós comprometidos e nós totais. Na presença de nós controlados por adversário que descartam mensagens, a probabilidade p_l de um pacote percorrer um caminho aleatório de comprimento l é igual à probabilidade do mesmo pacote percorrer um caminho aleatório com $l-1$ nós íntegros e posteriormente ser capturado por um nó controlado por adversário ou ter a sua segunda fase do roteamento iniciada [MOP+04]:

$$p_l = [f + (1-f)(1-p_f)](p_f)^{(l-1)}(1-f)^{(l-1)}. \tag{6.1}$$

O pior caso é quando um adversário intercepta um pacote cujo verdadeiro emissor é o predecessor observado na captura. Isto ocorre sempre que o adversário captura o pacote após um caminho aleatório de comprimento 1, ou seja, quando o primeiro nó do caminho aleatório é controlado por adversário ou quando o pacote é capturado após atravessar um primeiro nó íntegro que decide por iniciar a segunda fase do roteamento randômico. Para conservar o grau de anonimato de inocência provável, a probabilidade desta situação ocorrer deve ser inferior a 50% [MOP+04]. Assim:

$$f + (1-f)(1-p_f) < 0.5 \Rightarrow f < 1 - \frac{1}{2p_f}. \tag{6.2}$$

A probabilidade de encaminhamento p_f, necessária para se manter o grau de anonimato de inocência provável em uma rede com fator de comprometimento f, é dada por:

$$p_f > \frac{1}{2(1-f)}. \tag{6.3}$$

Se o adversário controla 10% dos nós conectados à rede, a probabilidade de encaminhamento utilizada para conservar o aspecto de inocência provável é $p_f > 0.5556$. Inversamente, uma probabilidade de roteamento $p_f = 0.9$ tolera um fator de comprometimento de 44% sem prejuízo do grau de anonimato [MOP+04].

Se o adversário tenta identificar o emissor personificando o receptor da mensagem (ou fornecedor de serviço), não há como reduzir o número de emissores possíveis do conjunto total, desde que o comprimento esperado do caminho aleatório seja suficiente para a mensagem atingir um ponto na rede independente da origem, ao final do caminho aleatório. Em um cenário com estas características, o grau de anonimato é fora de suspeita. Caso contrário, o grau de anonimato é de inocência provável: o adversário sabe que existe um subconjunto de nós com maior probabilidade de conter a origem da mensagem.

A utilização de cifração nas conexões de enlace na rede *Kurupira* impede um adversário global externo de identificar emissores trivialmente. Como um adversário global externo não consegue obter nenhuma informação a partir da monitoração das conexões de enlace, o grau de anonimato é fora de suspeita.

Anonimato de resposta

O anonimato de resposta está condicionado ao estabelecimento de pelo menos um canal de resposta. O estabelecimento de um canal de resposta requer o envio anônimo de uma única

mensagem especial. Logo, o anonimato de resposta depende diretamente do anonimato de envio. Desta forma, o anonimato de resposta atinge inocência provável contra adversários internos quando o anonimato de envio atingir.

Se o adversário personifica o emissor de uma mensagem com o objetivo de identificar o receptor da mesma, não há como reduzir o número de receptores possíveis sempre que o canal de resposta for bem-estabelecido, ou seja, sempre que o comprimento esperado do caminho aleatório utilizado para estabelecimento seja suficiente para a mensagem de criação do canal atingir um ponto independente na rede ao final da primeira fase do roteamento randomizado. Em um cenário com estas características, o grau de anonimato é fora de suspeita. Caso contrário, o grau de anonimato é de inocência provável: o adversário sabe que existe um subconjunto de nós com maior probabilidade de conter o destino da mensagem.

Pelos motivos citados, a utilização de cifração nas conexões de enlace na rede impede um adversário global externo de identificar receptores trivialmente. Como um adversário global externo não consegue obter nenhuma informação a partir da monitoração das conexões de enlace, o grau de anonimato é fora de suspeita.

Anonimato de par comunicante

A rede *Kurupira* garante grau de inocência provável para envio e resposta contra adversários internos sempre que a probabilidade de encaminhamento utilizada tolera a fração de comprometimento e permite, às mensagens trocadas, descrever caminhos aleatórios que atingem pontos na rede independentes da origem. Como não é possível para um adversário obter vantagem na identificação do emissor ou receptor de uma mensagem, a rede fornece grau de inocência provável para o par comunicante.

Como a utilização de cifração nas conexões de rede impede a identificação de origem ou destino de uma mensagem por parte de um adversário externo, o grau de anonimato do par comunicante é idêntico aos graus de anonimato de envio e resposta contra um adversário externo. A utilização de cifração nas conexões de rede impede ainda o descarte sistemático de pacotes e fornece repudiação aos usuários.

A Tabela 6.1 apresenta as conclusões obtidas e as compara com outras redes de destaque que utilizam caminhos aleatórios para comunicação anônima.

6.1.3 Módulos

O projeto *Kurupira* realizou a implementação dos protocolos descritos em qualidade de protótipo, como prova de conceito das idéias produzidas, e para estabelecer um *framework* completo para projeto, análise e testes com redes de anonimização. Apenas *software* livre foi utilizado durante a fase de desenvolvimento.

	Anonimato							
	Envio			Resposta			Par	
	Adversário			Adversário			Adversário	
Rede	**Nó**	**Receptor**	**Global**	**Nó**	**Emissor**	**Global**	**Nó**	**Global**
Crowds	$\star$	★	$\perp$	$\perp$	$\perp$	$\perp$	$\star$	$\perp$
AP3	$\star$	★	$\perp$	$\star$	★	$\perp$	$\star$	$\perp$
Kurupira	$\star^{\dagger}$	$\star^{\dagger}$(★‡)	★	$\star^{\dagger}$	$\star^{\dagger}$(★‡)	★	$\star^{\dagger}$	★

($\perp$) Neste cenário, a rede não fornece anonimato;
($\star$) Neste cenário, o usuário tem grau de anonimato de inocência provável;
(★) Neste cenário, o usuário tem grau de anonimato fora de suspeita;
(†) A probabilidade de encaminhamento p_f e o fator de comprometimento f são tais que $p_f > \frac{1}{2(1-f)}$;
(‡) A probabilidade de encaminhamento p_f garante um caminho aleatório de comprimento adequado;

Tabela 6.1: Grau de anonimato aproximado fornecido pela rede *Kurupira*.

O *framework* possui três módulos principais: um *daemon*, que instancia e controla os serviços disponibilizados pela rede; uma biblioteca de acesso, através da qual as aplicações podem acessar os serviços providos pelo *daemon*; e um *console*, que permite ao usuário observar e alterar o funcionamento do *daemon*.

- *Daemon*: é o principal componente da implementação. É responsável pela carga e controle dos serviços implementados pelas camadas da pilha. Cada camada é implementada como uma biblioteca compartilhada com uma interface bem-definida, de forma que a troca de um protocolo particular seja fácil, sem exigir recompilação de todo o *software*. O *daemon* também provê mecanismos de comunicação para a biblioteca de acesso e para o *console* e disponibiliza *logging* unificado para todas as camadas.

- *Biblioteca de acesso:* é responsável por fornecer uma interface de *sockets* POSIX, permitindo operações básicas (enviar, receber, conectar, encerrar, etc.) sobre a rede anônima. Um componente para acesso direto à tabela de *hash* distribuída também é fornecido.

- *Console:* permite que os usuários coletem estatísticas, alterem parâmetros de configuração e monitorem o funcionamento interno da rede.

6.1.4 Ataques e defesas

Além de fornecer graus de anonimato adequados, uma implementação real também deve proteger seu ambiente de diversos ataques. Vários dos ataques em redes de anonimização são tratados pela rede *Kurupira*, entre eles:

- *Ataques de rastreamento:* todos os pacotes são cifrados, e tem o mesmo tamanho e aparência. Para um adversário externo, o mesmo pacote capturado em conexões distintas será visto como dois pacotes diferentes. Para um adversário interno, ainda é possível reunir informações a respeito de um mesmo pacote coletadas em diferentes pontos. A topologia regular da rede, que distribui bem o tráfego entre os participantes, e a utilização de rotas não-determinísticas deve minimizar a quantidade de informação revelada para um adversário interno. A modificação de pacotes pode também ser sempre identificada pelas partes comunicantes, limitando a eficácia de estratégias de marcação de pacotes;

- *Ataques de identidade múltipla:* os endereços da camada de enlace (sistema estruturado propriamente dito) são diferentes dos endereços da camada de rede (pseudônimos) e ambos são gerados pela aplicação de uma função de *hash* criptográfico a uma chave pública. Isto dificulta o controle, por parte do adversário, de uma porção consecutiva do espaço de endereçamento. A topologia regular também limita a eficácia de ataques de maioria. A proteção contra ataques de eliminação advém da boa capacidade da rede em distribuir tráfego entre os participantes;

- *Ataques estatísticos:* O potencial da rede em distribuir o tráfego entre os nós limita a quantidade de informação estatística que pode ser obtida por um adversário. Ataques de predecessor foram especialmente considerados durante a fase de projeto da rede;

- *Ataques de correlação:* Não existem formas simples de se proteger uma rede contra ataques de correlação. A utilização de técnicas de modelagem de tráfego pode vir a ser utilizada no futuro. Por enquanto, apenas o potencial da rede na distribuição de tráfego e transmissão de mensagens de controle na forma de ruído devem dificultar a montagem de ataques simples de correlação temporal. A utilização de criptografia impede adversários externos de montar ataques de correlação por conteúdo;

- *Ataques na entrada e saída de nós:* O suporte a identidades temporárias impede o monitoramento do conjunto de nós conectados em busca da determinação de pares comunicantes;

- *Ataques de negação de serviço:* Assim como em ataques de correlação, qualquer rede está sujeita a ataques poderosos de negação de serviço. A escolha de uma topologia regular privilegia o balanceamento de carga e deve limitar a eficácia de ataques de sobrecarga dentro da rede; e

- *Ataques de personificação:* A utilização de pseudônimos com propriedades criptográficas impede a personificação de nós que controlem bem a exposição do seu par de chaves.

6.2 Serviço de nomes *KNS*

O serviço de nomes da rede *Kurupira* organiza-se hierarquicamente, como descrito no Capítulo 5. Um servidor-raiz padrão `root.kurupira` é mantido pelos desenvolvedores do projeto *Kurupira* e seu mapeamento para um pseudônimo vem atribuído estaticamente na distribuição de *software*. A chave pública deste servidor-raiz, além de publicada na base de dados da rede estruturada, é distribuída na página do projeto e acompanha a distribuição de *software*. É aconselhável para todos os participantes examinar com alguma freqüência os três veículos de distribuição da chave, em busca de atualizações ou detecção de alterações no par de chaves do servidor-raiz armazenado localmente.

Inicialmente, apenas o servidor-raiz padrão será disponibilizado. Caso haja necessidade, a criação de servidores-raiz alternativos será incentivada.

6.2.1 Módulos

O *software* do serviço de nomes *KNS* da rede *Kurupira* foi implementado em quatro componentes principais:

- *Módulo criptográfico:* executa as operações de assinatura e verificação no paradigma de Criptografia de Chave Pública Sem Certificados;

- *Módulo raiz:* responsável por armazenar os cadastros de servidores de nomes e distribuir informação de revogação de domínios. Os requisitos de armazenamento de um servidor-raiz englobam apenas as assinaturas sobre parâmetros de sistema de servidores de nomes cadastrados e as provas de revogação de domínio. A única responsabilidade adicional do servidor-raiz é distribuir a sua própria chave pública;

- *Módulo servidor:* responsável por atualizar as entradas do serviço de nomes com assinaturas válidas dos provedores de serviço sobre os seus respectivos pseudônimos. Cada servidor de nomes deve armazenar as assinaturas dos provedores e as provas de revogação e distribuir os mapeamentos correspondentes na base de dados;

- *Módulo provedor:* responsável por realizar o cadastro no serviço de nomes, disponibilizar o serviço propriamente dito e armazenar provas de revogação existentes;

- *Módulo cliente:* responsável por acessar a base de dados distribuída em busca de chaves públicas e mapeamentos entre nomes e pseudônimos. O módulo cliente deve ainda recuperar freqüentemente a chave pública do servidor-raiz, as delegações dos servidores de nomes. Estas informações podem ser copiadas localmente e alterações na base distribuída devem ser refletidas diretamente nas cópias. Em caso de informações díspares,

como alteração aparente na chave pública do servidor-raiz, mudança nas assinaturas de serviço e de domínios sem revogação, o usuário deve ser informado, para que tome ações corretivas manualmente.

6.2.2 Ataques e defesas

O servidor de nomes ainda está sujeito a ataques, vários dos quais tratados explicitamente pela arquitetura e protocolos:

- *Ataques de negação de serviço:* o adversário corrompe sistematicamente a informação do serviço de nomes distribuída na base de dados. Não há proteção específica para este ataque, a não ser a atualização freqüente de informação. A comparação entre as chaves armazenadas localmente e as distribuídas na rede auxilia a detecção de informação contraditória, e medidas corretivas manuais são solicitadas;

- *Ataques de seqüestro de endereço:* para seqüestrar um endereço, um adversário precisa forjar assinaturas sobre um pseudônimo diferente do registrado. Para forjar a assinatura de um nome n em um domínio D que inicialmente aponta para o pseudônimo N, um adversário precisa calcular $\{P'_N, Sig(N', \texttt{n.D.kurupira})\}$, que mapeia o nome n para o pseudônimo falso N'. A forja desta assinatura requer a geração de um par de chaves compatível com o nome n, mas confinado à chave pública P'_N. Isto só pode ser obtido com colaboração do servidor de nomes. A existência de duas assinaturas simultâneas para um mesmo nome n e chaves públicas distintas configura imediatamente a atuação maliciosa por parte do servidor de nomes. Outra forma de se seqüestrar um endereço é comprometer a chave privada S_N de posse do pseudônimo legítimo. A existência de duas assinaturas simultâneas para a mesma chave pública e pseudônimos distintos sem evidência de revogação por qualquer das partes envolvidas configura culpa do provedor de serviço em não manter sua chave privada em sigilo;

- *Ataques do servidor de nomes:* o servidor de nomes pode facilitar a personificação de um serviço legítimo de duas formas: emitindo uma chave privada parcial para um adversário, quando uma chave privada parcial para o mesmo nome já foi emitida para um usuário legítimo; e revogar maliciosamente um nome para que um adversário possa tomar controle do mesmo. A primeira infração pode ser detectada pela presença de duas assinaturas válidas para um mesmo nome e chaves públicas diferentes e implica a atuação maliciosa do servidor de nomes. A segunda infração, em caso de disputa, configura atuação maliciosa do servidor de nomes se o mesmo não detiver prova de revogação assinada pelo cliente;

- *Ataques do provedor de serviço:* o provedor de serviço pode tentar difamar um servidor de nomes apresentando assinaturas válidas para chaves públicas diferentes sob o mesmo nome. A posse da confirmação de revogação do par de chaves original pelo servidor de nomes o exime de qualquer culpa ou atuação maliciosa.

É difícil implantar punição para servidores de nomes que atuem de forma maliciosa, com o perigo de justamente efetivar a prática de censura. Espera-se que, sempre que uma disputa surgir, o verdadeiro atacante possa ser apontado e os participantes da rede tomem conhecimento da infração. Isto permite uma forma de auto-moderação da rede: servidores de nomes com histórico de fraudes podem ser totalmente ignorados pelos clientes individualmente. Entretanto, caso os ataques prejudiquem o bom funcionamento da rede, procedimentos adicionais automáticos para resolução de conflitos podem ser adotados para auxiliar os clientes a encontrar informação legítima. Entre eles, a assinatura de clientes sobre chaves públicas de serviços, atestando sua autenticidade. Isto define o serviço de nomes como uma *meritocracia*: o bom comportamento é recompensado com popularidade e um equilíbrio baseado em reputação é obtido. Ambientes de edição colaborativa de conteúdo, como a *Wikipédia* [2], demonstram que este princípio pode funcionar bem na prática.

6.3 Resumo

Neste capítulo, foram tratados os aspectos de implementação dos componentes projetados nos Capítulos 3 e 5. A rede *Kurupira*, resultante da implementação da rede de anonimização projetada, apresenta grau de anonimato não-inferior a inocência provável e implementa defesas para diversos ataques publicados na literatura de anonimato computacional. O serviço de nomes *KNS* integra-se à rede *Kurupira* e objetiva suportar o mapeamento de nomes amigáveis e de fácil publicação a pseudônimos confiáveis dentro da rede. A distribuição do *software* implementado encontra-se em `http://www.kurupira.go.dyndns.org`.

[2] `http://www.wikipedia.org`

Capítulo 7

Conclusões e trabalhos futuros

Neste trabalho, foram explorados os ambientes de comunicação anônima, sob o ponto de vista teórico e prático. Para a área de pesquisa em mecanismos de comunicação anônima, podem-se destacar as seguintes contribuições:

- Estudo detalhado de anonimato computacional e do paradigma de Criptografia de Chave Pública Sem Certificados;

- Projeto de uma rede de anonimização estruturada que oferece boa qualidade de anonimato. A rede implementa técnicas modernas de anonimização de tráfego para garantir anonimatos de envio, resposta e de par comunicante quando a fração de nós comprometidos pelo adversário tem magnitude controlada. Além disso, a implementação da rede trata e oferece resistência a vários dos ataques mais comuns a mecanismos de anonimização propostos na literatura;

- Projeto de uma política de roteamento específica para comunicação anônima em sistemas estruturados a partir da seleção de uma topologia regular e posterior aprimoramento com adaptações viáveis para implementação e conseqüente ganho em qualidade de anonimato. As características de qualidade de anonimato de envio, desempenho e resistência contra negação de serviço da política de roteamento foram demonstradas empiricamente através de simulações acuradas;

- Projeto de um serviço de nomes para promover o ganho de usabilidade nas redes de anonimização correntes. O serviço de nomes utiliza delegação criptográfica sob o paradigma de Criptografia de Chave Pública Sem Certificados para fornecer as operações de cadastro, consulta e revogação imediata. As operações de armazenamento e recuperação inerentes à rede estruturada subjacente são aproveitadas para distribuição de informação de revogação e validação de assinaturas e parâmetros do sistema criptográfico utilizado;

- Implementação dos componentes projetados utilizando uma arquitetura bem-definida e requisitos de projeto que facilitam a integração da rede anônima ao *software* disponível atualmente. A implementação dos componentes envolveu a criação de um módulo criptográfico para cifração e assinatura sob o paradigma de Criptografia de Chave Pública Sem Certificados, adequado para utilização em aplicações reais. A rede de anonimização projetada foi implementada na forma da rede *Kurupira* e o serviço de nomes foi implementado sob o codinome *KNS*. A qualidade de anonimato fornecida pela rede *Kurupira* foi analisada e comparada a projetos semelhantes. As defesas para os ataques mais comuns contra a rede de anonimização ou o serviço de nomes foram também apresentadas.

A rede *Kurupira*, em conjunção com o serviço de nomes *KNS*, satisfazem os objetivos iniciais deste trabalho, por fornecerem um ambiente ideal para manifestação de privacidade e liberdade de expressão, com suporte à publicação de serviços anonimizados. Ainda assim, diversas questões importantes e diretamente relacionadas são apontadas como sugestões de trabalhos futuros:

- Proposta de metodologia rigorosa para avaliação da qualidade dos anonimatos de envio, resposta e par comunicante, considerando adversários e ataques distintos dos considerados neste trabalho;

- Aprimoramento das técnicas de simulação, para suportar ambientes de maior população. Simulação de ambientes sob condições mais realistas, com entrada e saída de nós e latência de transmissão também é necessária. A observação de variações na qualidade de anonimato em casos extremos é importante;

- Aprimoramento da política de roteamento projetada ou da topologia selecionada, objetivando ganho adicional em qualidade de anonimato;

- Aumento de usabilidade do serviço de nomes, por agregação de transparência a algumas das situações que requerem intervenção manual. Isto está diretamente relacionado à concepção de mecanismos de revogação imediata mais simples, desde que as premissas gerais para a qualidade de anonimato sejam mantidas e o funcionamento geral do sistema seja preservado.

- Divulgação do *software* na comunidade, para formar um núcleo estável da rede *Kurupira* na *Internet*;

- Extensão da rede *Kurupira* para acessar recursos na *Internet* convencional de forma anônima; e

- Implementação mais eficiente de emparelhamentos bilineares e das primitivas de Criptografia de Chave Pública Sem Certificados.

Referências Bibliográficas

[ACL+06] M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wong, and G. Yang. Malicious KGC attack in Certificateless Cryptography. Cryptology ePrint Archive, Report 2006/255, 2006.

[Adl77] L. M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *Proc. 18th IEEE Symp. on Foundations of Comp. Science*, pages 55–60, Providence, 1977. IEEE.

[And96] R. Anderson. The Eternity service. In *Proceedings of Pragocrypt '96*, 1996.

[AR05] S. S. Al-Riyami. *Cryptographic schemes based on elliptic curve pairings*. PhD Thesis, Department of Mathematics, Royal Holloway, University of London, 2005.

[ARP03] S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. Cryptology ePrint Archive, Report 2003/126, 2003.

[AS03] J. Aspnes and G. Shah. Skip graphs. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 384–393, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[AS05] D. F. Aranha and E. F. O. Sandes. Libfreedom: camada de transporte anônima para construção de serviços resistentes à censura, 2005. Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação, Universidade de Brasília.

[ATS04] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[BF01] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229, London, UK, 2001. Springer-Verlag.

[BG03] K. Bennett and C. Grothoff. GAP – practical anonymous networking. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop - PET '03*, 2003.

[BGHP02] K. Bennett, C. Grothoff, T. Horozov, and I. Patrascu. Efficient sharing of encrypted data. In *Proceedings of ASCIP 2002*, pages 107–120. Springer-Verlag, July 2002.

[BGhS04] P. S. L. M. Barreto, S. Galbraith, C. Ó hÉigeartaigh, and M. Scott. Pairing computation on supersingular abelian varieties. Cryptology ePrint Archive, Report 2004/375, 2004.

[BKLS02] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, pages 354–368, London, UK, 2002. Springer-Verlag.

[BLMQ05] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology (ASYACRYPT '05)*, Lecture Notes in Computer Science, pages 515–532. Springer Berlin / Heidelberg, 2005.

[BLR+04] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish. A layered naming architecture for the internet. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 343–352, New York, NY, USA, 2004. ACM Press.

[BLS02] P. S. L. M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. Cryptology ePrint Archive, Report 2002/088, 2002. `http://eprint.iacr.org/2002/088`.

[BM06] X. Boyen and L. Martin. Identity-Based Cryptography Standard (IBCS) #1: Supersingular curve implementations of the BF and BB1 cryptosystems. S/MIME Working Group - Internet Draft, June 2006. `http://www.ietf.org/internet-drafts/draft-ietf-smime-ibcs-00.txt`.

[Bor05] N. Borisov. *Anonymous routing in structured peer-to-peer overlays*. PhD thesis, University of California, Berkeley, 2005.

[BPS00] O. Berthold, A. Pfitzmann, and R. Standtke. The disadvantages of free MIX routes and how to overcome them. In H. Federrath, editor, *Proceedings of Designing*

Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, pages 30–45. Springer-Verlag, LNCS 2009, July 2000.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[Bro02] Z. Brown. Cebolla: pragmatic IP anonymity. In *Proceedings of the 2002 Ottawa Linux Symposium*, June 2002.

[BS04] P. S. L. M. Barreto and M. Scott. Compressed pairings. In *Advances of Cryptology (CRYPTO '04)*, volume 3152 of *Lecture Notes in Computer Science*, pages 140–156. Springer-Verlag, 2004. `http://eprint.iacr.org/2004/032/`.

[BSG00] P. Boucher, A. Shostack, and I. Goldberg. Freedom Systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.

[BSM04] S. Bono, C. A. Soghoian, and F. Monrose. Mantis: a high-performance, anonymity preserving P2P network. Technical report, John's Hopkins University Information Security Institute, 2004. Technical Report TR-2004-01-B-ISI-JHU.

[BSNS05] J. Baek, R. Safavi-Naini, and W. Susilo. Certificateless public key encryption without pairing. In *ISC*, pages 134–148, 2005.

[Cal05] J. Callas. Identity-based encryption with conventional public-key infrastructure, April 2005. PKI' 05.

[CC05a] Z. Cheng and R. Comley. Efficient Certificateless Public Key Encryption. Cryptology ePrint Archive, Report 2005/12, 2005.

[CC05b] T. Chothia and K. Chatzikokolakis. A survey of anonymous peer-to-peer file-sharing. In *Proceedings of the IFIP International Symposium on Network-Centric Ubiquitous Systems (NCUS '05)*, Lecture Notes in Computer Science. Springer-Verlag, 2005.

[CCV04] Z. Cheng, R. Comley, and L. Vasiu. Remove key escrow from the Identity-Based Encryption System. In *IFIP TCS*, pages 37–50, 2004.

[CD97] G. Di Caro and M. Dorigo. AntNet: a mobile agents approach to adaptive routing. Technical Report IRIDIA/97-12, Université Libre de Bruxelles, Belgium, 1997.

[CDG+02] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the 5th USENIX Symposium on Operating Systems Design and Implementation - OSDI '02*, 2002.

[CDKR02] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications (JSAC)*, 20(8):1489–1499, 2002.

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[Cha85] D. Chaum. Security without identification: Transaction systems to make Big Brother obsolete. *CACM*, 28(10), October 1985.

[Cha88] D. Chaum. The Dining Cryptographers Problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[Cha03] B. Chan. Identity-Based PKI for DNSSEC. Master's thesis, Royal Holloway University of London, 2003.

[CK01] E. Cohen and H. Kaplan. Proactive caching of DNS records: addressing a performance bottleneck. In *SAINT '01: Proceedings of the 2001 Symposium on Applications and the Internet (SAINT 2001)*, page 85, Washington, DC, USA, 2001. IEEE Computer Society.

[CMW06] R. Clayton, S. J. Murdoch, and R. N. M. Watson. Ignoring the Great Firewall of China. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, UK, June 2006. Springer.

[Coc01] C. Cocks. An Identity Based Encryption Scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, London, UK, 2001. Springer-Verlag.

[CSB05] S. Chatterjee, P. Sarkar, and R. Barua. Efficient computation of Tate pairing in projective coordinates over general characteristic fields. In *Information Security and Cryptology (ICISC '04)*, Lecture Notes in Computer Science, pages 168–181. Springer Berlin / Heidelberg, 2005.

[CSWH00] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: a distributed anonymous information storage and retrieval system. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, July 2000.

[Dan04a] G. Danezis. *Better anonymous communications*. PhD thesis, University of Cambridge, July 2004.

[Dan04b] G. Danezis. The traffic analysis of continuous-time MIXes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *LNCS*, May 2004.

[dB46] N. G. de Bruijn. A combinatorial problem. *Nederl. Akad. Wetensch. Proc.*, 49:758–764, 1946.

[DDM03] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: design of a Type III anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[Den06] A. W. Dent. A survey of Certificateless Encryption schemes and security models. Cryptology ePrint Archive, Report 2006/211, 2006. `http://eprint.iacr.org/2006/211/`.

[DFM00] R. Dingledine, M. J. Freedman, and D. Molnar. The Free Haven Project: distributed anonymous storage service. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*. Springer-Verlag, LNCS 2009, July 2000.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, November 1976.

[dJ83] R. des Jardins. ISO open systems interconnection standardization status report. *SIGCOMM Comput. Commun. Rev.*, 13(2):4–5, 1983.

[DMS04] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[Dou02] J. Douceur. The Sybil attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS '02)*, 2002.

[DP04] C. Díaz and B. Preneel. 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, August 2004.

[DR06] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346 (Proposed Standard), April 2006. Updated by RFCs 4366, 4680, 4681.

[DSCP02] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET '02)*, 2002.

[DSD04] C. Díaz, L. Sassaman, and E. Dewitte. Comparison between two practical mix designs. In *Proceedings of ESORICS 2004*, LNCS, France, September 2004.

[FM02] M. J. Freedman and R. Morris. Tarzan: a peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 537–554, London, UK, 1999. Springer-Verlag.

[Gen03] C. Gentry. Certificate-Based Encryption and the certificate revocation problem. Cryptology ePrint Archive, Report 2003/183, 2003.

[GGG+03] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 381–394, New York, NY, USA, 2003. ACM Press.

[Gir91] M. Girault. Self-certified public keys. In *EUROCRYPT '91*, pages 490–497. Springer, 1991. LCNS vol.547.

[GJ04] P. Golle and A. Juels. Dining cryptographers revisited. In *Proceedings of Eurocrypt 2004*, May 2004.

[Gor93] D. M. Gordon. Discrete logarithms in GF(p) using the number field sieve. In *Siam Discrete Math*, volume 6, pages 124–138, 1993.

[Goy06] D. H. Goya. Proposta de esquemas de criptografia e assinatura sob modelo de criptografia de chave pública sem certificado. Master's thesis, Instituto de Matemática e Estatística / Universidade de São Paulo, 2006.

[Gro03] C. Grothoff. An excess-based economic model for resource allocation in peer-to-peer networks. *Wirtschaftsinformatik*, June 2003.

[GRPS03] S. Goel, M. Robson, M. Polte, and E. G. Sirer. Herbivore: a scalable and efficient protocol for anonymous communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

[GRS96] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[GSB02] M. Gunes, U. Sorges, and I. Bouazzi. Ara - the ant-colony based routing algorithm for MANETs. In *Proceedings of the Internation Workshop on AD Hoc Networking (IWAHN '02)*, 2002.

[GT96] C. Gülcü and G. Tsudik. Mixing e-mail with Babel. In *Proceedings of the Network and Distributed Security Symposium (NDSS '96)*, 1996.

[Gut02] P. Gutman. PKI: it's not dead, just resting. *IEEE Computer*, 35(8):41–49, 2002.

[HHSI04] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai. Identity-Based Hierarchical Strongly Key-Insulated Encryption and its application. Cryptology ePrint Archive, Report 2004/338, 2004. `http://eprint.iacr.org/2004/338`.

[HJS+03] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. SkipNet: a scalable overlay network with practical locality properties. In *USENIX Symposium on Internet Technologies and Systems*, 2003.

[HLX+05] J. Han, Y. Liu, L. Xiao, R. Xiao, and L. M. Ni. A mutually anonymous peer-to-peer protocol design. In *Proceedings of IPDPS '05*, volume 1, page 68. IEEE Computer Society, 2005.

[KEB98] D. Kesdogan, J. Egner, and R. Büschkes. Stop-and-go MIXes: providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525, 1998.

[KK03] M. F. Kaashoek and D. R. Karger. Koorde: A simple degree-optimal distributed hash table. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

[Koh78] L. M. Kohnfelder. Towards a practical public-key cryptosystem. B.S. Thesis, supervised by L. Adleman, May 1978.

[Kon05] J. Kong. Formal notions of anonymity for peer-to-peer networks. Cryptology ePrint Archive, Report 2005/132, 2005. `http://eprint.iacr.org/`.

[Küg03] D. Kügler. An analysis of GNUnet and the implications for anonymous, censorship-resistant networks. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760, March 2003.

[LAS06] J. K. Liu, M. H. Au, and W. Susilo. Self-Generated-Certificate Public Key Cryptography and certificateless signature/encryption scheme in the Standard Model. Cryptology ePrint Archive, Report 2006/373, 2006.

[LM] B. Lipinski and P. MacAlpine. A security review of an anonymous peer-to-peer file transfer protocol. `http://www.lix.polytechnique.fr/~tomc/P2P/Papers/Systems/AP3.pdf`.

[LQ04] B. Libert and J.-J. Quisquater. What is possible with identity based cryptography for PKIs and what still must be improved. *Lecture Notes in Computer Science*, 3093:57–70, 2004.

[Maa04] Martijn Maas. Pairing-based cryptography. Master's thesis, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, 2004.

[Mar99] G. T. Marx. What's in a name? some reflections on the sociology of anonymity. `http://web.mit.edu/gtmarx/www/anon.html`, 1999.

[MD05] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[Mil86] V. Miller. Short programs for functions on curves. Manuscrito não-publicado, 1986.

[MM02] P. Maymounkov and D. Mazières. Kademlia: a peer-to-peer information system based on the XOR metric. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 53–65, London, UK, 2002. Springer-Verlag.

[MNR02] D. Malkhi, M. Naor, and D. Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing*, 2002.

[Moc87] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), 1987.

[MOP+04] A. Mislove, G. Oberoi, A. Post, C. Reis, and P. Druschel. AP3: cooperative, decentralized anonymous communication. In *Proceedings of the 11th ACM SIGOPS European Workshop*, 2004.

[MR04] N. Modadugu and E. Rescorla. The design and implementation of Datagram TLS. In *NDSS*. The Internet Society, 2004.

[NW04] M. Naor and U. Wieder. Know thy neighbor's neighbor: better routing for Skip-Graphs and small worlds. In *IPTPS*, volume 3279, pages 269–277. Springer, 2004.

[OS06a] L. Overlier and P. F. Syverson. Locating hidden servers. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 2006.

[OS06b] L. Overlier and P. F. Syverson. Valet Services: improving hidden servers with a personal touch. In *Proceedings of Privacy Enhancing Technologies Workshop (PET '06)*, 2006.

[Pan03] L. Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, 2003.

[PH06] A. Pfitzmann and M. Hansen. Anonymity, unobservability, and pseudonymity: a consolidated proposal for terminology. Draft, version 0.28, 2006.

[Ray00] J.-F. Raymond. Traffic Analysis: protocols, attacks, design issues, and open problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 10–29. Springer-Verlag, 2000.

[RD01] A. Rowstron and P. Druschel. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218, 2001.

[RFH+01] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, volume 31, pages 161–172. ACM Press, October 2001.

[RP02] M. Rennhard and B. Plattner. 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, November 2002.

[RR98] M. Reiter and A. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information Systems Security*, 1, 1998.

[RS04a] V. Ramasubramanian and E. G. Sirer. Beehive: exploiting power law query distributions for O(1) lookup performance in peer-to-peer overlays. In *Proceedings of the 1st USENIX Symposium on Networked Systems Design and Implementation (NSDI '04)*, pages 331–342, San Francisco, CA, USA, March 2004.

[RS04b] V. Ramasubramanian and E. G. Sirer. The design and implementation of a next generation name service for the internet. *SIGCOMM Comput. Commun. Rev.*, 34(4):331–342, 2004.

[Sae03] S. Saeednia. A note on Girault's self-certified model. *Inf. Process. Lett.*, 86(6):323–327, 2003.

[SBS02] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5: a protocol for scalable anonymous communication. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, May 2002.

[Sco05] M. Scott. Computing the Tate pairing. In *Topics in Cryptology/CT-RSA '05*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer Berlin/Heidelberg, 2005.

[SD02] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET '02)*. Springer-Verlag, LNCS 2482, 2002.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell Systems Technology Journal*, 28:657–715, 1949.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology (CRYPTO '84)*, pages 47–53, New York, NY, USA, 1984. Springer-Verlag New York, Inc.

[Shm02] V. Shmatikov. Probabilistic analysis of anonymity. In *Proceedings of 15th IEEE Computer Security Foundations Workshop*, pages 198–218, 2002.

[SLS01] V. Scarlata, B. N. Levine, and C. Shields. Responder anonymity and anonymous peer-to-peer file sharing. In *Proceedings of IEEE International Conference on Network Protocols (ICNP '01)*. IEEE Computer Society Press, 2001.

[SMK+01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[SS98] J. H. Silverman and J. Suzuki. Elliptic curve discrete logarithms and the index calculus. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security (ASIACRYPT '98)*, pages 110–125, London, UK, 1998. Springer-Verlag.

[SW95] W. R. Stevens and G. R. Wright. *TCP/IP illustrated (vol. 2): the implementation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[TB06] P. Tabriz and N. Borisov. Breaking the collusion detection mechanism of MorphMix. In *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, UK, June 2006. Springer.

[TJ02] M. Theimer and M. B. Jones. Overlook: scalable name service on an overlay network. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 52, Washington, DC, USA, 2002. IEEE Computer Society.

[WALS02] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium (NDSS '02)*. IEEE Computer Society Press, 2002.

[WALS04] M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: an analysis of a threat to anonymous communication systems. *ACM Transactions on Information Systems Security*, 7:489–522, 2004.

[YL04a] D. H. Yum and P. J. Lee. Generic construction of Certificateless Encryption. In *EUROPKI '04*, volume 3043 of *Lecture Notes in Computer Science*, pages 802–811. Springer-Verlag, 2004.

[YL04b] D. H. Yum and P. J. Lee. Generic construction of Certificateless Signature. In *ACISP '04*, volume 3108 of *Lecture Notes in Computer Science*, pages 200–211. Springer-Verlag, 2004.

[ZKJ01] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: an infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, April 2001.