UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUSTAVO HUFF MAUCH

# Dois Pesos, Duas Medidas: Gerenciamento de Identidades Orientado a Desafios Adaptativos para Contenção de *Sybils*

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Luciano Paschoal Gaspary
Orientador

Porto Alegre, outubro de 2010

**CIP – CATALOGAÇÃO NA PUBLICAÇÃO**

Mauch, Gustavo Huff

Dois Pesos, Duas Medidas: Gerenciamento de Identidades Orientado a Desafios Adaptativos para Contenção de *Sybils* / Gustavo Huff Mauch. – Porto Alegre: PPGC da UFRGS, 2010.

59 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2010. Orientador: Luciano Paschoal Gaspary.

1. Redes par-a-par. 2. Autenticação. 3. Gerenciamento de identidades. I. Gaspary, Luciano Paschoal. II. Título.

*“Wer? Wie? Was? Warum? Wer nicht fragt, bleibt Dumm!*
*Mit Gedüld und Spücke fängt mann Mücke.”*
— DEUTSCHEN SPRICHWÖRTERN

# AGRADECIMENTOS

Agradeço ao grupo de pesquisa de segurança em P2P e a todos seus integrantes, em especial meu orientador Prof. Luciano Gaspary, Prof Marinho Barcellos e colegas Wéverton Cordeiro e Flávio Santos pela inestimável ajuda na elaboração dessa dissertação. Agradeco também ao Prof. Nazareno Andrade (UFCG) pela concessão dos traços históricos do Bitsoup.org utilizados na avaliação experimental apresentada nesta dissertação.

# SUMÁRIO

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| CAN | Content Adressable Network |
| CCDF | Complementary Cumulative Distribution Function |
| DHT | Distributed Hash Table |
| DNS | Domain Name Service |
| ID | Identificador |
| IP | Internet Protocol |
| IRC | Internet Relay Chat |
| ISP | Internet Service Provider |
| MAC | Message Authentication Code |
| MSN | Microsoft Network |
| NAT | Network Address Translation |
| P2P | Peer-to-Peer |
| PDA | Personal Digital Assistant |
| PGP | Pretty Good Privacy |
| PKI | Public Key Infrastructure |
| SHA | Secure Hash Algorithm |
| SPKI | Simple Public Key Infrastructure |
| VoIP | Voice over Internet Protocol |

# LISTA DE FIGURAS

# LISTA DE TABELAS

# RESUMO

O ataque *Sybil* consiste na criação indiscriminada de identidades forjadas por um usuário malicioso (atacante). Uma abordagem promissora para mitigar esse ataque consiste em conceder novas identidades mediante a resolução de desafios computacionais. Apesar de suas potencialidades, as soluções baseadas em tal abordagem não distinguem solicitações de usuários corretos das de atacantes, fazendo com que ambos paguem o mesmo preço por identidade solicitada. Por conta disso, essas soluções podem não ser efetivas quando os recursos computacionais dos atacantes são muito superiores aos que os usuários legítimos dispõem. Assumindo desafios de uma determinada dificuldade, atacantes com hardware de maior capacidade conseguiriam resolver um conjunto muito superior de desafios e, com isso, obter um número elevado de identidades. Aumentar uniformemente a dificuldade dos desafios poderia, no outro extremo, tornar proibitivo o ingresso de pares à rede. Para lidar com esse problema, nesta dissertação propõe-se o uso de desafios adaptativos como limitante à disseminação de *Sybils*. Estima-se um grau de confiança da fonte de onde partem as solicitações de identidade em relação às demais. Quanto maior a freqüência de solicitação de identidades, menor o grau de confiança e, conseqüentemente, maior a complexidade do desafio a ser resolvido pelo(s) usuário(s) associado(s) àquela fonte. Resultados obtidos por meio de experimentação mostram a capacidade da solução de atribuir desafios mais complexos a potenciais atacantes, penalizando minimamente usuários legítimos.

**Palavras-chave:** Redes par-a-par, autenticação, gerenciamento de identidades.

**Two Weights and Two Measures: Using Adaptive Puzzles in Identity Management for *Sybil* Contention**

# ABSTRACT

The *Sybil* attack consists on the indiscriminate creation of counterfeit identities by a malicious user (attacker). An effective approach to tackle such attack consists of establishing computational puzzles to be solved prior to granting new identities. Despite its potentialities, solutions based on such approach do not distinguish between identity requests from correct users and attackers, and thus require both to afford the same cost per identity requested. Therefore, those approaches may not be effective when the attacker's computational resources are superior than those used by correct users. Assuming any choice of puzzle hardness, attackers that have access to high-performance computing resources will be able to solve puzzles several order of magnitude faster than legitimate users and thus obtain a large amount of identities. On the other way, raising the cost to solve the puzzles could restrict legitimate users too much. To tackle this problem, in this paper we propose the use of adaptive computational puzzles to limit the spread of *Sybils*. We estimate a trust score of the source of identity requests in regard to the behavior of others. The higher the frequency a source requests identities, the lower its trust score and, consequently, the higher the complexity of the puzzle to be solved by the user(s) associated to that source. Results achieved by means of an experimental evaluation evidence our solution's ability to establish more complex puzzles to potential attackers, while minimally penalizing legitimate users.

**Keywords:** Peer-to-peer networks, authentication, identity management.

# 1 INTRODUÇÃO

Aplicações Par-a-Par (do inglês *Peer-to-Peer*, P2P) tornam-se a cada dia mais comuns. Estudos recentes (conduzidos entre 2007 e 2009 (SCHULZE; MOCHALSKI, 2007) e (SCHULZE; MOCHALSKI, 2009)) mostraram que aplicações de compartilhamento de arquivos em redes par-a-par são responsáveis por cerca de 70% do tráfego na Internet. Lawrence Roberts (ROBERTS, 2009) argumenta que, mesmo representando apenas 5% dos usuários da Internet, eles consomem cerca de 75% da largura de banda mundial. Entre os principais motivos pelos quais o modelo par-a-par é mais atrativo que o tradicional cliente-servidor para aplicações de troca de arquivos, destacam-se os seguintes. Primeiro, redes par-a-par são *escaláveis*, isto é, lidam eficientemente tanto com grupos pequenos quanto grandes de participantes. Segundo, é possível *depender* (de *dependability*) mais do funcionamento dessas redes, já que não possuem ponto central de falhas e resistem melhor a ataques intencionais de negação de serviço, por exemplo. Por fim, essas redes oferecem *autonomia* aos seus participantes, possibilitando que eles *entrem* e *saiam* da rede de acordo com seus interesses e disponibilidades, bem como tomem suas decisões independentemente de entidades externas (BARCELLOS; GASPARY, 2006).

Apesar das redes par-a-par contribuírem para aplicações de compartilhamento de recursos e para colaboração em larga escala, nenhuma instituição está disposta a adotar soluções, por mais promissoras que sejam, sem um bom grau de confiança em relação à manutenção da segurança de sua infraestrutura de *hardware* e *software*, bem como de suas informações. Nesse sentido, um dos principais entraves para o emprego mais amplo da tecnologia par-a-par em contextos mais sensíveis reside na carência de soluções eficazes e eficientes em áreas-chave como autenticação (LAWTON, 2004) e gerenciamento das identidades dos participantes. A ineficácia desses mecanismos permite a criação de múltiplas identidades falsas por parte de um único usuário malicioso, constituindo um ataque denominado pela literatura de *Sybil* (DOUCEUR, 2002). Atacantes que consigam realizar tal ataque podem, por exemplo, violar o princípio da cooperação e obter benefícios sem compartilhar com a rede (*Free-riding* (LOCHER et al., 2006)) ou, então, segmentar a rede e manipular a troca de mensagens entre segmentos (*Eclipse* (SINGH et al., 2006)).

Segundo Douceur, que primeiro descreveu o ataque *Sybil*, não é possível banir completamente a criação de identidades falsas sem utilizar algum ponto de centralização no sistema (por exemplo, com o uso de entidades certificadoras). No entanto, essa solução mais extrema não é desejada, pois entidades centrais tornam-se pontos únicos de falhas e potenciais gargalos. Técnicas alternativas têm sido propostas visando impedir a criação de identidades falsas, sem abrir mão de características essenciais das redes par-a-par. Uma das mais promissoras consiste em atribuir ou renovar a concessão de identidades aos usuários solicitantes mediante a resolução de desafios computacionais (BORISOV, 2006). A idéia por trás da exigência da resolução de desafios é que pares legítimos pro-

vem suas boas intenções com a rede, comprometendo uma parte de seus recursos. Ao mesmo tempo, pares maliciosos interessados em criar múltiplas identidades serão obrigados a passar grande parte de seu tempo processando desafios e, portanto, consumindo recursos, o que reduz seu poder de assumir um número elevado de identidades.

Diversos trabalhos foram publicados propondo o emprego de desafios computacionais para o gerenciamento de identidades em redes par-a-par (BORISOV, 2006; CASTRO et al., 2002; ROWAIHY et al., 2007). Apesar de suas potencialidades, as propostas que adotam tal abordagem não fazem distinção entre solicitações de identidades oriundas de usuários corretos e de atacantes. À medida que ambos estão sujeitos ao pagamento do mesmo preço (computacional) por cada identidade solicitada, essas propostas podem não ser efetivas quando os recursos computacionais dos atacantes são muito superiores aos que os usuários legítimos dispõem. Assumindo desafios de uma determinada dificuldade, atacantes com *hardware* de maior capacidade conseguiriam resolver um conjunto muito superior de desafios e, com isso, obter um número elevado de identidades. Aumentar uniformemente a dificuldade dos desafios poderia, no outro extremo, tornar proibitivo o ingresso de pares legítimos à rede.

Para lidar com essa limitação, a presente dissertação propõe o uso de desafios adaptativos como estratégia de contenção contra a disseminação de *Sybils*. Em contraste com as propostas existentes na literatura, nossa solução estima um grau de confiança da *fonte* de onde parte a solicitação de identidade em relação ao comportamento das demais fontes. No contexto deste trabalho, *fonte* pode referir-se à estação de um usuário (identificada pelo seu endereço IP), à rede local a qual a estação pertença, a um sistema autônomo (*Autonomous System*, AS), etc. Essa decisão depende essencialmente da granularidade que se desejar ou seja possível adotar para a fonte (por exemplo, no caso de usuários posicionados atrás de redes usando NAT, a granularidade a ser considerada é associar todos os usuários daquela rede a uma única fonte). À medida que aumenta a freqüência com que novas solicitações por identidades partem de uma dada fonte, diminui a confiabilidade da mesma. Conseqüentemente, maior será a complexidade do desafio computacional a ser resolvido antes que a identidade solicitada seja obtida pelo(s) usuário(s) associado(s) àquela fonte. Para avaliar a eficácia da solução proposta na contenção de ataques *Sybil*, foi realizada avaliação experimental considerando traços históricos de solicitações de identidades em uma comunidade par-a-par. Os resultados obtidos mostram a capacidade da solução em atribuir desafios computacionais mais complexos a potenciais atacantes, ao passo que usuários legítimos são minimamente penalizados.

As contribuições da dissertação desdobram-se portanto em quatro:

1. Proposta de uma medida para apoiar a identificação de pares que exibem comportamentos suspeitos;

2. Proposta de um modelo que, dado um grau de suspeita, determina um valor de confiança, este usado, na parametrização do desafio a ser resolvido para obtenção ou renovação de uma identidade;

3. Avaliação do modelo amparada por informações obtidas de sistemas reais, permitindo observar como a solução desempenharia caso adotada;

4. Reflexão sobre aspectos operacionais da solução, em especial sua instanciação em arcabouços par-a-par reais, possibilidades de mapeamento de valores de confiança em desafios e alternativas para materializar a noção de fonte (origem) das solicitações de identidade.

A dissertação esá organizada da seguinte forma: inicialmente são revisados, no Capítulo 2, os fundamentos sobre redes par-a-par e segurança. O Capítulo 3 aprofunda questões relacionadas à segurança dessas redes, incluindo uma descrição do ataque *Sybil* e outros decorrentes. O Capítulo 3 aborda ainda mecanismos de gerenciamento de identidades empregados em redes par-a-par e associa-os com os principais trabalhos existentes na literatura que buscam evitar a ocorrência de *Sybils*. O Capítulo 4 apresenta o mecanismo proposto para o uso de desafios adaptativos como uma proteção ao ataque *Sybil*, enquanto o Capítulo 5 descreve a avaliação do mesmo. O Capítulo 6 discute questões relacionadas ao emprego da solução proposta em arcabouços par-a-par. Por fim, o Capítulo 7 conclui essa dissertação, apresentando as considerações finais e conclusões obtidas durante essa pesquisa.

# 2 FUNDAMENTOS DE REDES PAR-A-PAR

Aplicações par-a-par tornam-se a cada dia mais comuns. Seu uso é crescente não só entre usuários domésticos como também em ambientes acadêmicos e corporativos (BARCELLOS; GASPARY, 2006). Isso se deve principalmente ao seu potencial em oferecer o substrato necessário à criação de compartilhamento de dados em larga escala, distribuição de conteúdo e *multicast* em nível de aplicação(LUA et al., 2005). Esse modelo de computação é empregado em aplicações com finalidades bastante diversas que abrangem desde as mais tradicionais como compartilhamento de arquivos e troca de mensagens instantâneas, até computação distribuída, transmissão de dados e armazenamento de arquivos em rede.

Neste capítulo é apresentada uma revisão sobre as principais características de sistemas par-a-par. Essa revisão é focada em aspectos mais elementares, tais como a organização dessas redes e seus principais usos. Os aspectos de segurança relevantes a este trabalho, incluindo mecanismos de gerenciamento de identidades e ataques, serão aprofundados posteriormente no Capítulo 3. Na Seção 2.1 são inicialmente apresentadas as características-chave e definição de sistemas par-a-par. Na seqüência, na Seção 2.2 são listadas e caracterizadas algumas das principais aplicações que utilizam essa arquitetura. O capítulo segue com a Seção 2.3 descrevendo as possíveis organizações da rede de sobreposição (*overlays*) utilizadas em redes par-a-par. A Seção 2.4 lista e caracteriza os principais sistemas que possuem uma rede de sobreposição não-estruturada, enquanto a Seção 2.5 apresenta e descreve os principais sistemas que possuem uma rede de sobreposição estruturada.

## 2.1 Características-chave e Definição

Não há consenso na literatura para uma definição precisa de o que são redes par-a-par, nem quais suas características fundamentais. A definição inicial refere-se a sistemas totalmente distribuídos nos quais os pares apresentam as mesmas características e funcionalidades. Essa definição, entretanto, exclui vários sistemas atualmente aceitos como par-a-par. Algumas características, no entanto, são comuns a diversas definições ((THEOTOKIS; SPINELLIS, 2004), (BARCELLOS; GASPARY, 2006), (LUA et al., 2005) p.ex.) tais como a descentralização e a autonomia concedida aos participantes.

Segundo Theotokis e Spinellis (THEOTOKIS; SPINELLIS, 2004) as duas características primordiais de sistemas par-a-par são:

1. Compartilhamento direto de recursos entre os participantes, sem a necessidade de servidores centrais, apesar de ser aceito um certo nível de centralização em tarefas pontuais como atribuição de identidades, ou manutenção da reputação dos pares,

mas que não retirem o caráter distribuído desse tipo de aplicação.

2. Capacidade de auto-organização possibilitando que a rede lide eficientemente com alterações freqüentes na sua população de pares.

Além das características já citadas, outra importante é o fato das redes par-a-par serem criadas com base na colaboração voluntária entre seus participantes. Com base nessas características, Barcellos e Gaspary (BARCELLOS; GASPARY, 2006) formularam a seguinte definição para redes par-a-par:

> "*Redes Peer-to-Peer (P2P) são sistemas distribuídos consistindo de nodos interconectados capazes de se auto-organizar em "redes de sobreposição" (overlays) com o objetivo de compartilhar recursos tais como conteúdo (música, vídeos, documentos, etc.), ciclos de CPU, armazenamento e largura de banda, capazes de se adaptar a populações transientes de nodos enquanto mantendo conectividade aceitável e desempenho, sem necessitar da intermediação ou apoio de uma entidade central.*"

## 2.2 Aplicações Par-a-Par

Sistemas par-a-par são empregados em diversas categorias de aplicações. A lista abaixo apresenta as principais, no entanto, não pretende ser uma lista exaustiva, pois acredita-se que outras aplicações poderiam beneficiar-se dessa tecnologia.

- **compartilhamento de arquivos** (*file sharing*). Também conhecida como *distribuição de conteúdo*, seu objetivo é permitir que usuários transfiram arquivos diretamente entre si. Tipicamente qualquer usuário participante da rede par-a-par pode ”publicar” arquivos cujo conteúdo permanecerá imutável e poderá ser recuperado por quaisquer outros participantes. Essa categoria de aplicações é uma das mais populares e provavelmente a responsável pela disseminação do uso de de redes par-a-par em ambiente doméstico. São exemplos desse tipo de aplicação o BitTorrent (Bittorrent, 2010), o KaZaa (LIANG; KUMAR; ROSS, 2004), o Gnutella (Gnutella, 2010) e o Napster (OpenNap, 2010);

- **sistema de armazenamento de arquivos em rede** (*network storage*). Essa categoria é semelhante à anterior, porém com uma diferença importante: o conteúdo armazenado pode ser modificado pelos usuários. Dessa forma, as alterações devem considerar uma possível replicação dos dados e propagá-las para todas as cópias existentes. Restrições de acesso (escrita e leitura) costumam ser consideradas. Como exemplos podem ser citados o PAST (DRUSCHEL; ROWSTRON, 2001), o OceanStore (OceanStore, 2010), o Ivy (MUTHITACHAROEN et al., 2002) e o JetFile (GRöNVALL; MARSH; PINK, 1996);

- **colaboração e comunicação entre usuários**. São aplicações que permitem a comunicação direta e em tempo real entre usuários.As aplicações existentes possibilitam comunicação através de voz (VoIP), mensagens de texto, vídeo ou transmissão direta de arquivos. São exemplos desse tipo de aplicações o Google Talk (Google, 2010), o Skype (Skype, 2010), o MSN Messenger (MSN, 2010), o Jabber (Jabber, 2010), o Yahoo Messenger (Yahoo, 2010) e o ICQ (ICQ, 2010);

- **computação distribuída**. Aplicações dessa categoria visam a execução de processamento intensivo normalmente criando infra-estruturas de grade e explorando a capacidade ociosa dos computadores que dela fazem parte. Em determinados sistemas pode, inclusive, haver controle centralizado, no formato mestre-escravo. Alguns exemplos são o Seti@Home (SETI, 2010), o Genome@Home (Genome, 2010) e o OurGrid (ANDRADE et al., 2004).

- **transmissão de dados ou overlay multicast**. Nestas aplicações, o *overlay* forma uma infraestrutura de comunicação baseada em *multicast* em nível de aplicação. O objetivo é possibilitar que um mesmo conteúdo seja transmitido por um par e entregue a um número potencialmente grande de usuários dispersos geograficamente. Normalmente essa tecnologia é empregada para a transmissão de eventos ao vivo (*Live Streaming*). Pode-se citar como exemplos o Justin.tv (Justin.tv, 2010), o Joost (Joost, 2010) e o End System Multicast (ESM, 2010).

## 2.3 Organização da Rede de Sobreposição

Sistemas par-a-par podem ser organizados em duas principais modalidades, no que diz respeito a sua rede de sobreposição (*overlay*). Essa organização define basicamente como funciona a alocação de objetos (e suas chaves), os algoritmos de busca desses objetos e o posicionamento lógico dos pares interconectados. Essas características influenciam também diversos aspectos de segurança como robustez e desempenho. Abaixo são apresentadas as principais características de cada categoria.

- *Overlays* não-estruturados: em sistemas com *overlay* não estruturado, a topologia é determinada de maneira *ad hoc*. Os pares entram e saem aleatoriamente do *overlay* e estabelecem ligações com outros pares arbitrariamente. Assim, não existe qualquer regra para o posicionamento dos pares, objetos e serviços.

- *Overlays* estruturados: em sistemas com *overlay* estruturado a topologia é determinada por um esquema de alocação de chaves aos pares, de forma a associar um determinado objeto ou serviço a um par (ou conjunto de pares). Essa associação é determinística e conhecida globalmente no *overlay*, o que facilita a busca de objetos e/ou serviços. *Overlays* estruturados costumam empregar tabelas *hash* distribuídas (*Distributed Hash Tables*, *DHTs*) que funcionam como tabelas de roteamento, permitindo que a localização dos objetos e/ou serviços seja realizada em poucos passos.

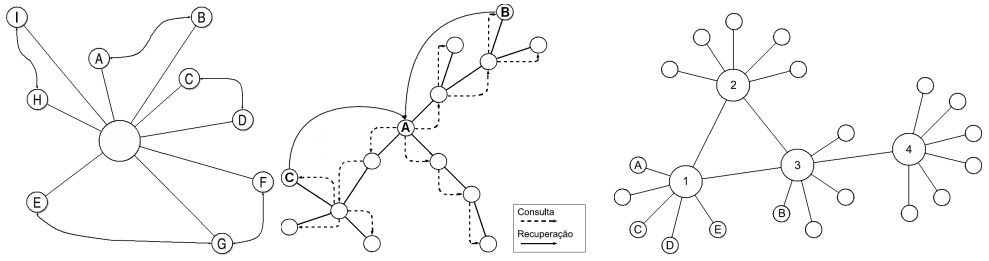
## 2.4 Principais *Overlays* Não-estruturados

*Overlays* não-estruturados apresentam uma dificuldade na localização dos objetos, visto não existirem regras que associem seu conteúdo com o posicionamento. Para resolver essa questão, utilizou-se inicialmente métodos que realizavam a inundação da rede com mensagens. Esses métodos mostraram-se ineficientes em termos de número de mensagens, tráfego e tempo para a localização de objetos. Essa ineficiência fomentou a pesquisa de novas formas de busca de objetos que empregam caminhada aleatória (GKANTSIDIS; MIHAIL; SABERI, 2004) ou índices de roteamento (TSOUMAKOS; ROUSSOPOULOS, 2003). Em contrapartida, sistemas não-estruturados adaptam-se diretamente ao modelo de operação par-a-par, sem necessitar qualquer reorganização em função de

entrada e saída de pares, que é potencialmente alta nessas redes. Os principais exemplos de infraestruturas par-a-par que possuem uma organização não estruturada são descritos a seguir.

**Napster.** O Napster foi o a aplicação precursora em termos de compartilhamento de arquivos em redes par-a-par e muito provavelmente o responsável pela disseminação dessa cultura. O Napster, no entanto, vai contra o importante princípio de descentralização dessas redes, pois depende de um servidor central para seu funcionamento. Os usuários que ingressam na rede Napster oferecem seu conteúdo enviando informações sobre arquivos locais a um servidor central. Usuários interessados em obter conteúdo fazem uma consulta ao servidor que retorna ao requisitor uma lista de endereços dos pares que disponibilizam o conteúdo buscado. A recuperação do arquivo então é feita diretamente entre os pares envolvidos, evitando assim a sobrecarga no servidor. A arquitetura do Napster está exemplificada na Figura 2.1(a): computadores de usuários buscam informações sobre arquivos no diretório central e então se comunicam com outros pares diretamente para obter os arquivos desejados. Maiores informações sobre o Napster podem ser obtidas em (OpenNap, 2010).

**Gnutella.** É um sistema de compartilhamento de arquivos com topologia *ad hoc*. Nele, todos os pares são servidores e clientes ao mesmo tempo. Um par (requisitor) que deseja buscar um arquivo faz uma inundação de mensagens na rede. Pares que possuem o arquivo especificado na busca retornam pelo caminho inverso a resposta positiva até o requisitor. Caso o solicitante obtenha mais de uma resposta positiva ele escolhe uma delas e efetua a recuperação do arquivo diretamente do par escolhido. A Figura 2.1(b) ilustra um exemplo de busca e recuperação de conteúdo. Nela, o par A efetua uma busca por inundação e encontra o recurso procurado nos pares B e C. Após isso ele escolhe um deles e recupera o conteúdo diretamente. Maiores informações sobre o Gnutella podem ser obtidas em (Gnutella, 2010).




(a) Exemplo de busca e recuperação de conteúdo em redes Napster & BitTorrent

(b) Exemplo de busca e recuperação de conteúdo em rede Gnutella.

(c) Exemplo de busca e recuperação de conteúdo em rede Overnet/eDonkey2000

Figura 2.1: Exemplos de par-a-par não-estruturado

**FastTrack/KaZaa.** É um sistema de compartilhamento de arquivos que utiliza uma arquitetura hierárquica em dois níveis, composta por *pares simples* e *super-pares*. *Pares simples* são os próprios pares da rede, enquanto os *super-pares* são as entidades responsáveis por executar as buscas por arquivos. Quando um par ingressa na

rede, ele se conecta a um *super-par* e envia uma lista com a descrição dos arquivos que está disponibilizando. Para efetuar uma busca um *par simples* deve encaminhar uma requisição a seu *super-par* que pode responder diretamente quando um dos *pares simples* conectados a ele estiver disponibilizando o arquivo desejado, ou então executa uma busca enviando mensagens a outros *super-pares* (um *super-par* pode manter uma lista com milhares de outros *super-pares*). Apesar do gerenciamento ser caracterizado como um *overlay* não-estruturado, o papel desempenhado pelos *super-pares* recomendaria que ele fosse caracterizado como híbrido. Maiores informações sobre o FastTrack/KaZaa podem ser obtidas em (LIANG; KUMAR; ROSS, 2004).

**Overnet/eDonkey2000.** Também é um sistema com arquitetura de duas camadas, porém híbrida, que é composto por pares "clientes" e pares "servidores". Os servidores são responsáveis por indexar informações sobre arquivos e participar das operações de busca. Ambos, cliente e servidor são executados por usuários quaisquer. A Figura 2.1(c) apresenta um exemplo de busca e recuperação de conteúdo. Nela o par A consulta o super-par 1 sobre um dado recurso, que através de inundação é encontrado no par B conectado ao super-par 3, e então solicita recurso diretamente ao par A. Maiores informações sobre o Overnet/eDonkey2000 podem ser obtidas em (KULBAK et al., 2005).

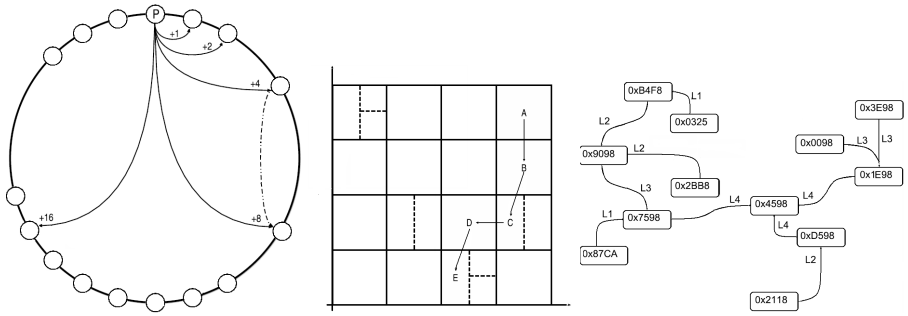
**BitTorrent.** É um sistema de compartilhamento de arquivos baseado em "enxames" (do inglês *swarms*) de pares, que trocam diretamente entre si blocos de arquivos mas que são coordenados por um par central, denominado *tracker*. Um usuário interessado em disponibilizar um conteúdo para ser compartilhado precisa preparar um arquivo (denominado *torrent*) que contém as propriedades do arquivo a ser disponibilizado e o endereço do *tracker* responsável pelo enxame. A busca e recuperação do *torrent* é feita externamente à rede BitTorrent em si (utilizando normalmente motores de busca na web). De posse do *torrent*, o usuário fornece-o ao software cliente, que se conecta ao *tracker* e este responde com uma lista aleatória de pares presentes no enxame e que portanto estão interessados no mesmo conteúdo além de tambem estarem distribuindo-o. O par requisitante então contacta múltiplos pares dessa lista, solicitando blocos do conteúdo desejado. A arquitetura de um enxame BitTorrent está ilustrada na Figura 2.1(a): o elemento central representa o *tracker* do enxame, e a ele se conectam até várias centenas de pares usuários que trocam dados entre si. Note-se que não há apenas um único ponto central, mas sim um número arbitrário de *trackers* espalhados pela Internet e que dividem a responsabilidade por milhares de *torrents*. Inclusive, um arquivo *torrent* pode especificar mais de um *tracker* responsável pela distribuição de um conteúdo. Apesar do gerenciamento ser caracterizado como um *overlay* não-estruturado, o papel desempenhado pelo *tracker* recomendaria que ele fosse caracterizado como híbrido. Maiores informações sobre o BitTorrent podem ser obtidas em (Bittorrent, 2010) e (JUN; AHAMAD, 2005).

## 2.5 Principais *Overlays* Estruturados

Sistemas par-a-par estruturados, tipicamente baseados em DHT, ao contrário dos não-estruturados, possibilitam que os objetos sejam encontrados em um pequeno número de passos. Entretanto, essa vantagem traz a necessidade da perfeita correspondência entre

o termo solicitado na busca e a chave do objeto. Com isso, o par requisitante necessita conhecer antecipadamente a chave do objeto procurado. Alguns autores argumentam que a organização da rede com a saída e entrada de pares é difícil de ser mantida. Os principais exemplos de infraestruturas par-a-par que possuem uma organização estruturada são descritos a seguir.

**Chord.** Utiliza *hashing* SHA-1 para associar chaves de objetos aos pares. Identificadores dos pares são obtidos fazendo-se um *hash* do endereço IP, enquanto chaves de objetos são obtidas fazendo-se um *hash* da descrição do objeto. Uma chave $k$ é associada ao par de identificador igual a $k$, ou se o mesmo não existe, ao próximo no anel (dito "sucessor"). Cada par mantém um ponteiro para os $N$ pares imediatamente sucessores e uma tabela (denominada *finger table*) com até $m$ apontadores para outros pares espalhados no anel de forma que a busca por um específico possa ser realizada na menor quantidade possível de passos. O roteamento de buscas é unidirecional ao longo do anel e pode ser *recursivo* ou *iterativo*. No modo recursivo, a mensagem vai sendo encaminhada de par em par e se aproximando do predecessor do objeto; quando chega no par com o objeto, a busca volta recursivamente ao requisitante. No iterativo, o par requisitante vai perguntando àqules que ficam cada vez mais próximos do par com o objeto; quando a busca chega ao par que possui o objeto, ele responde com os dados. A falha de pares não causa uma falha global, podendo haver replicação de objetos em pares consecutivos. A Figura 2.2(a) ilustra uma rede Chord contendo um par P os apontadores que ele possui para outros pares espalhados no anel, o que permite uma busca eficiente e em tempo logarítmico. Maiores informações sobre o Chord podem ser obtidas em (DABEK et al., 2001; STOICA et al., 2003; Chord, 2010).




(a) Exemplo de rede Chord.

(b) Exemplo de rede CAN.

(c) Exemplo de rede Tapestry.

Figura 2.2: Exemplos de par-a-par estruturado

**CAN.** *Content-Addressable Network* é uma infraestrutura descentralizada que se baseia na utilização de um espaço cartesiano $n$-dimensional. Esse espaço é dividido em "zonas" e cada par é responsável por armazenar uma delas, além de uma tabela *hash* contendo informações sobre o endereço de seus vizinhos e as zonas pelas quais são responsáveis. A busca de objetos emprega pares $< chave, objeto >$ para mapear um ponto $P$ no espaço de coordenadas, utilizando para isso uma função *hash*. A busca é roteada em direção ao destino usando um encaminhamento simples ao par

que está mais próximo das coordenadas. Para fazer parte da rede, um novo participante precisa conhecer algum outro que já faça parte da CAN. Assim ele escolhe aleatoriamente um ponto $P$ e contata o responsável por essa zona. O responsável *X* por tal zona então divide a sua em duas e torna o novo par responsável por uma metade. Por fim, o novo par constrói uma tabela de rotas com os endereços dos vizinhos de *X* e também notifica os mesmos para que atualizem sua tabela de rotas. Ao sair da CAN, as zonas ocupadas e a tabela *hash* de um par são repassadas a algum de seus vizinhos. A Figura 2.2(b) ilustra o espaço de coordenadas de uma rede CAN e o roteamento de uma mensagem em direção ao par responsável pela chave procurada. Nela, o par A busca um recurso que se encontra abrigado na região sob responsabilidade do par E. Maiores informações sobre a CAN podem ser obtidas em (RATNASAMY et al., 2001).

**Tapestry.** Tapestry é uma infraestrutura par-a-par que permite o roteamento de mensagens a objetos ou a uma cópia mais próxima a eles, caso exista replicação. As estratégias de roteamento e localização do Tapestry se baseiam na estrutura distribuída de *Plaxton mesh* (PLAXTON; RAJARAMAN; RICHA, 1997), na qual os pares podem assumir papel de "servidores" de objetos, "roteadores" de mensagens e "clientes" que solicitam dados. Cada par mantém uma tabela de vizinhos onde cada entrada possui múltiplos níveis de endereçamento. Dado um total de *L* níveis, cada nível *l* contém uma seqüência de tamanho *L*, onde apenas os *l* primeiros são conhecidos e indicam qual o vizinho pode continuar o roteamento. Mensagens são incrementalmente roteadas através dos pares dígito por dígito, da direita para a esquerda. Por exemplo, uma mensagem do par com identificador 0x87CA para aquele identificado por 0x3E98 poderia passar pelos seguintes pares: $xxxx8- > xxx98- > xxE98- > x3E98$. A malha Plaxton utiliza um *par raiz* para cada objeto, o que serve como garantia de que ele será localizado em fututras operações de busca. Quando um objeto $o$ é inserido na rede no par $ps$, um par raiz $pr$ ($ps \neq pr$) é associado ao objeto. Uma mensagem é então roteada de $ps$ para $pr$, armazenando dados na forma de um mapeamento $< o, ps >$ em todos os pares ao longo do caminho. Durante uma busca, mensagens destinadas a $o$ são inicialmente roteadas com destino $pr$, até que um par seja encontrado contendo o mapeamento $< o, ps >$. A Figura 2.2(c) demonstra um exemplo de uma fração de topologia Plaxton com os pares e seus identificadores e de roteamento de mensagem, extraído de (ZHAO; KUBIATOWICZ; JOSEPH, 2001). Maiores informações sobre o Tapestry podem ser obtidas em (ChimeraTapestry, 2010).

**Pastry.** Malhas do estilo *Plaxton mesh* são a base para outras redes, tais como OceanStore (OceanStore, 2010) e Pastry. As diferenças principais residem na utilização de outras estratégias para replicação, localidade e performance. Pastry, especificamente, é empregado pelo sistema de armazenamento persistente de larga escala PAST (DRUSCHEL; ROWSTRON, 2001; ROWSTRON; DRUSCHEL, 2001a) e no Scribe (CASTRO et al., 2002), um sistema de comunicação em grupo e de comunicação de eventos de larga escala. Maiores informações sobre o Pastry podem ser obtidas em (ROWSTRON; DRUSCHEL, 2001b).

**Kademlia.** Infraestrutura de roteamento que usa um mecanismo inovador para roteamento de mensagens e busca de objetos segundo uma métrica de distância entre identificadores de pares (não de proximidade de rede) baseada em xor. A topolo-

gia tem a propriedade que toda a mensagem trocada carrega ou reforça informações úteis de contato. O sistema explora essa informação para enviar mensagens de busca assíncronas e paralelas que toleram falhas de pares sem impor atrasos e *timeouts* a usuários. Diversas aplicações par-a-par estão empregando o algoritmo Kademlia: Overnet, eDonkey e eMule, além de BitTorrent, que emprega Kademlia para permitir o uso de *torrents* sem um *tracker*. Maiores informações sobre Kademlia podem ser obtidas em (MAYMOUNKOV; MAZIERES, 2002).

Tendo apresentado neste capítulo a organização básica das redes par-a-par, além das principais aplicações e sistemas existentes, é possível verificar que esse modelo é bastante versátil, apresentando vantagens no uso em diversos tipos de aplicação quando comparado ao tradicional modelo cliente-servidor. No entanto, ainda existem questões em aberto no que diz respeito à segurança de sistemas par-a-par. De especial interesse para essa dissertação são as questões que envolvem autenticação e gerenciamento das identidades dos participantes que, devido a fragilidades, podem ter conseqüências danosas para os usuários dessas redes. Essas questões serão abordadas em maior profundidade a seguir no Capítulo 3.

# 3 SEGURANÇA EM REDES PAR-A-PAR

Este capítulo aborda aspectos de segurança em redes par-a-par. Na Seção 3.1 (e suas subseções) são apresentados ataques a essas redes que só são possíveis graças a características delas e, principalmente, aos fracos ou inexistentes mecanismos de autenticação e gerenciamento de identidades. A Seção 3.2 apresenta e categoriza as principais abordagens utilizadas em redes par-a-par para a autenticação e gerenciamento das identidades dos participantes.

## 3.1 Ataques

Esta seção descreve em maiores detalhes o ataque *Sybil*, suas características e seu potencial danoso. Relaciona, também, outros ataques que se beneficiam do *Sybil*, ou que dele sejam derivados, o que acaba multiplicando o potencial danoso para a rede. A seção aborda, ainda, ataques não-relacionados ao *Sybil*, nomeadamente Traição e Conluio. O ataque da Traição é abordado porque a solução proposta nesta dissertação, por considerar o comportamento dos pares como dado de entrada, está sujeita a ele e, portanto, precisa entender seu impacto e mitigá-lo. Já Conluio é descrito porque a avaliação da solução proposta (realizada no Capítulo 5) considera diferentes cenários, entre eles um ataque em que diversos atacantes atuam em conjunto, em conluio.

### 3.1.1 *Sybil*

Douceur (DOUCEUR, 2002) descreve o ataque *Sybil* como sendo a criação de diversas identidades (falsas), todas elas controladas por uma mesma entidade. A idéia motivadora desse ataque é que uma única entidade possa controlar a maior, ou pelo menos grande parte, das identidades presentes na rede. Cabe ressaltar que o controle de diversas identidades não é danoso em si, não trazendo prejuízos à rede, porém o controle sobre diversas identidades permite e potencializa os efeitos danosos de outros ataques tais como *Eclipse*, *Free-riding* e poluição de conteúdo, que serão detalhados nas próximas subseções. Além desses ataques, o *Sybil* faz com que toda comunicação de pares para troca de dados tenha grande chance de recair sobre uma das identidades controladas e possa ser alterada na forma que mais aprouver ao seu controlador. Um atacante com várias identidades falsas pode também subverter algoritmos baseados em votação, pois pode utilizá-las para criar pares-fantasma que votarão de acordo com seus interesses, desviando-se assim do funcionamento correto esperado.

Para que um participante possa criar diversas identidades de forma a lançar um ataque *Sybil*, é preciso que o mecanismo de gerência de identidades utilizado na rede permita a criação de identidades sem uma verificação de sua autenticidade. Algumas aplicações

par-a-par utilizam um mecanismo por meio do qual o próprio participante é livre para criar sua identidade sem espécie alguma de verificação de autenticidade. Assim, a criação de pares virtuais é bastante simples e acaba facilitando um ataque *Sybil*. O gerenciamento de identidades será abordado em maiores detalhes na Seção 3.2.

O ataque *Sybil* é particularmente importante não apenas devido ao seu potencial danoso, mas também à dificuldade de impedi-lo. A prevenção ao *Sybil* é muito difícil de ser feita sem introduzir um certo grau de centralização no sistema. Segundo Douceur, sem uma entidade central de autenticação é impossível impedir completamente um ataque *Sybil* e essa centralização pode causar a diminuição ou mesmo violação de características-chave para as rede par-a-par, tais como descentralização, tolerância a falhas e escalabilidade.

### 3.1.2 Eclipse

O ataque *Eclipse* consiste no controle de diversas identidades por uma mesma entidade em que elas estejam dispostas no *overlay* formado pela rede par-a-par de forma a segmentá-la. Segundo Marling Engle e Javed Khan (ENGLE, 2006), o objetivo desse ataque é particionar a rede em dois ou mais segmentos. Quando bem sucedido, toda comunicação entre essas partições será roteada por, pelo menos, um *Sybil*. O nome "Eclipse"decorre do fato que um ataque bem sucedido é capaz de "eclipsar", isto é, esconder um segmento da rede do(s) outro(s). A Figura 3.1 ilustra uma rede sob esse tipo de ataque. À esquerda, pode-se observar a rede como é percebida pelos pares corretos, que desconhecem o fato de várias identidades serem controladas por uma mesma entidade. À direita, observa-se a topologia real da rede, em que ela aparece particionada em dois segmentos. O caminho natural para um atacante obter identidades para criar pares-fantasma é justamente lançando um prévio ataque *Sybil* à rede.

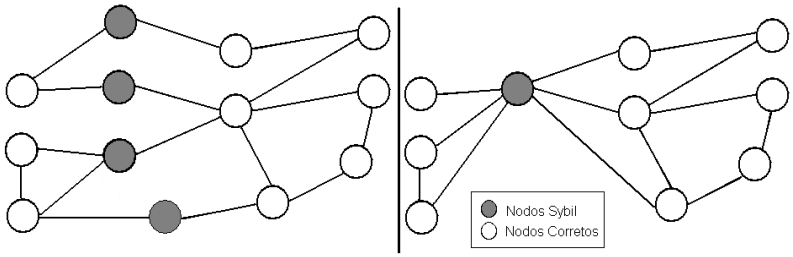



Figura 3.1: Exemplo de ataque Eclipse

Esse tipo de ataque é particularmente danoso em redes par-a-par estruturadas como Pastry (ROWSTRON; DRUSCHEL, 2001c), Can (RATNASAMY et al., 2001) ou Chord (STOICA et al., 2001), pois nesse tipo de redes o identificador de cada par é atribuído de acordo com seu posicionamento no *overlay* formado. Em redes estruturadas, um ataque *Eclipse* pode ser planejado de forma a separar um segmento específico, impedindo que seus pares tenham acesso aos recursos da rede, ou que outros pares acessem os recursos dos eclipsados. Outro possível efeito de um ataque *Eclipse* é permitir que o atacante controle a visão que os pares de um segmento tenham do(s) outro(s), por exemplo forjando mensagens, ou até mesmo a própria existência de pares.

### 3.1.3 Free-Riding

*Free-riding* é uma violação ao princípio-chave da colaboração voluntária entre os participantes de redes par-a-par, em especial, as que têm por finalidade o compartilhamento

de arquivos. Esse ataque consiste em aproveitar os recursos da rede sem oferecer a devida contrapartida em troca. Os usuários que atuam dessa maneira apenas recuperando arquivos dos outros participantes, sem oferecer os seus, são chamados de caroneiros (*free-riders*). Diferentemente do *Eclipse*, ataque cuja motivação é, em geral, bastante clara, no *Free-riding* podem ser elencadas algumas possíveis razões para que um usuário o execute (ENGLE, 2006):

- diminuição no uso da banda de *upload*, pois grande parte dos provedores de acesso à Internet restringe seu uso. Assim, o usuário quer obter arquivos da rede, mas não compartilha seus arquivos pois não quer extrapolar o limite de uso de sua banda de *upload*.

- A distribuição de conteúdo protegido por direitos autorais é, na maior parte das vezes, ilegal e pode sujeitar o usuário a sanções legais por parte das autoridades competentes, visto que na maioria das aplicações par-a-par de compartilhamento de arquivos é fácil associar o par da rede com o usuário real. Esse temor é justificado, pois no ano de 1998 diversos usuários da rede Napster (Napster, 2010) sofreram restrições legais pelo compartilhamento de arquivos sem respeito a seus direitos autorais. Assim, muitos usuários que não desejam abrir mão da facilidade de obter arquivos (usando redes par-a-par) preferem não contribuir com seus arquivos para não correrem o risco de sofrer sanções legais.

- Algumas pessoas tendem a abusar do uso de certos recursos se não tiverem que pagar por eles de alguma forma. Esse comportamento, cuja explicação pode estar relacionada à "Tragédia dos Comuns" (HARDING, 1968) , é bastante nocivo, pois se adotado por todos participantes inviabilizaria as redes par-a-par de troca de arquivos como as conhecemos hoje.

Em geral, as redes de compartilhamento de arquivos incentivam a colaboração. Por exemplo, as redes BitTorrent (COHEN, 2003) empregam um mecanismo chamado *tit-for-tat*. Simplificadamente, pode ser afirmado que quanto maior for a taxa de *upload* de um usuário para outros, maior será sua taxa de *download* destes mesmos usuários, gerando assim um esquema justo de distribuição, em que aquele que mais contribui mais recebe em troca. Esse esquema, além de premiar os pares que mais contribuem, também penaliza aqueles que pouco o fazem, desestimulando assim comportamentos como o de usuários egoístas. À primeira vista, o *tit-for-tat* é capaz de impedir a existência de caroneiros, visto que eles não conseguirão obter os recursos da rede por não contribuírem com a mesma. No entanto, caso o atacante possua uma quantidade significativa de identidades, ele pode obter melhor taxa de *download* que os usuários legítimos, mesmo não compartilhando seus recursos. Isso ocorre, pois redes de compartilhamento de arquivos costumam empregar uma escolha aleatória de parceiros, isto é, a escolha inicial de quais participantes terão suas requisições atendidas é feita de forma randômica. Dessa forma, quanto mais identidades um usuário controlar, mais provável será que um deles seja escolhido aleatoriamente. Brasileiro e Andrade (PONTES; BRASILEIRO; ANDRADE, 2007) demonstram que em redes BitTorrent - que aplicam essa escolha aleatória de novos parceiros para troca de arquivos - um atacante que controle apenas 0,2% do total de identidades (o estudo cita atacantes controlando cerca de 100 identidades em um sistema com um total de 50.000 pares) já obtém vantagens na sua taxa de recuperação de arquivos quando comparado com usuários legítimos que compartilham seus arquivos.

Redes de compartilhamento de arquivos, tais como BitTorrent, que não possuem mecanismos robustos de autenticidade de identidades são particularmente vulneráveis ao ataque *Sybil*. Dessa forma, fica mais evidente a importância desse ataque como base para o lançamento de outros, pois constitui-se no caminho natural para que um usuário malicioso obtenha a quantidade necessária de identidades para lançar um ataque *Free-riding* bem sucedido.

### 3.1.4 Lavagem de Identidade

A lavagem de identidade (do inglês *White-washing*) é mais um ataque que não causa diretamente danos à rede, porém pode ser usado como ferramenta para potencializar ou prolongar outros ataques, em especial o de *Free-riding*. Para entendê-lo, é importante saber que diversas redes par-a-par fazem uso de sistemas de reputação (como *Eigentrust* (KAMVAR; SCHLOSSER; GARCIA-MOLINA, 2003), *TrustGuard* (SRIVATSA; XIONG; LIU, 2005), *PeerTrust* (XIONG; LIU, 2004), *FuzzyTrust* (SONG et al., 2005) e *XRep* (DAMIANI et al., 2002)). Segundo Audun Josang *et al* (JOSANG; ISMAIL; BOYD, 2007), a idéia básica desses sistemas é permitir que os pares se avaliem, por exemplo, após completar uma transação e usar a agregação dessas avaliações para inferir um valor para a confiança ou reputação de cada participante. Esse valor é considerado por cada par na sua decisão de efetuar novas transações com quaisquer outros nós no futuro. Pares com baixa reputação sofrem penalidades por isso, tipicamente não tendo suas requisições atendidas. Usuários que praticam o ataque *Free-riding* têm uma reputação baixa, em virtude da sua recusa em disseminar seu próprio conteúdo. Após um certo tempo, esse ataque não será mais possível, pois todos os outros pares se recusarão a fornecer arquivos para um usuário tão pouco confiável. Esse é o momento em que um usuário malicioso lançaria um ataque de lavagem de identidade, que consiste em entrar e sair repetidas vezes da rede, sempre com uma identidade nova. Como a reputação é associada ao identificador, um usuário caroneiro que repetidamente entre na rede com uma nova identidade, livra-se das penalidades causadas pela má reputação obtida por sua não-cooperação (FELDMAN et al., 2006) e pode, então, continuar recuperando arquivos.

A lavagem de identidade só traz vantagens a um atacante caso o custo computacional para a obtenção de novas identidades seja baixo, comparado com o necessário para readquirir uma boa reputação. Caso o custo de criação de identidades falsas seja alto, o ataque perde sua motivação de ser um modo rápido de obter melhores níveis de confiança perante os outros participantes da rede. Cabe ressaltar que se um usuário malicioso não pudesse lançar um ataque de lavagem de identidade, então dificilmente um *Free-riding* teria sucesso, pois o atacante teria problemas em livrar-se de sua má reputação e, assim, poder continuar recuperando arquivos.

### 3.1.5 Traição

O ataque da traição é mais um relacionado a sistemas de reputação. Em tal ataque, um par comporta-se adequadamente por um tempo de forma a construir uma boa reputação e, então, explora o sistema valendo-se da mesma. Segundo Marti e Garcia-Molina (MARTI; GARCIA-MOLINA, 2006), esse ataque é especialmente efetivo quando os pares ganham privilégios à medida em que conquistam boa reputação. Um exemplo do ataque de traidor é quando um usuário no eBay (eBay, 2010) constrói uma reputação com muitas transações de pequeno valor, e então lesa alguém em uma transação de grande valor. Cabe ressaltar que um par traidor pode surgir não de uma mudança intencional no comportamento de um usuário, mas de uma mudança no ambiente, por exemplo, uma máquina cliente perfeita-

mente correta pode ser infectada com um vírus estilo Cavalo de Tróia, que, então poderia aleatoriamente abusar de um par que já obtivesse uma boa reputação.

O ataque da traição está diretamente ligado ao da lavagem de identidade. Logo que um traidor passa a comportar-se maliciosamente, aproveitando-se para isso de sua boa reputação, ele começará a ser punido. Em breve sua reputação atingirá baixos patamares e o traidor terá dificuldade em obter cooperação para suas transações. Neste ponto, é necessário empregar o ataque de lavagem de identidade para obter uma nova identidade, que terá uma reputação inicial maior que a sua atual. Sua má-reputação é anulada e o atacante pode reiniciar o ciclo de obter um alto valor de reputação para, então, voltar a lesar o sistema.

Esse ataque não possui uma ligação direta com o *Sybil*, pois não é beneficiado pelo controle de diversas identidades (falsas) por uma mesma entidade. Ele está sendo aqui descrito, pois a proposta deste trabalho, de certa forma, assemelha-se a um sistema de reputação. Sistemas de reputação, ou que de alguma maneira consideram o comportamento dos pares como um dado de entrada, devem preocupar-se com *traidores* e as influências que eles podem ter no sistema proposto. A questão dos *traidores* será abordada mais adiante, na Seção 4.3, quando procura-se demonstrar que a solução proposta é robusta a ataques dessa natureza.

### 3.1.6 Poluição de Conteúdo

A poluição de conteúdo é um ataque que difere dos apresentados anteriormente, pois ele não traz benefício direto para o atacante. Seu único propósito é impedir que outros usuários tenham acesso a algum conteúdo específico disponível em redes par-a-par de compartilhamento de arquivos. De acordo com Flavio Santos *et al* (SANTOS; GASPARY; BARCELLOS, 2009) esse ataque pode se manifestar de três formas diferentes. A primeira consiste em corromper o conteúdo de um arquivo, mantendo seus metadados originais (LIANG; KUMAR, 2005). A segunda forma na qual a poluição de conteúdo pode se manifestar consiste em anunciar um arquivo corrompido com o mesmo identificador de um conteúdo original, através da manipulação do gerador de identificadores de objetos. Essas 2 primeiras formas de poluição de conteúdo se baseiam no corrompimento de arquivos que podem ocorrer de diversas formas, entre elas:

- arquivos inválidos, incapazes de serem executados;
- arquivos que são válidos, porém apresentam defeitos. Por exemplo, em caso de arquivos de áudio podem ser inseridos intervalos de silêncio ou cortados intervalos do áudio original;
- arquivos que são válidos, mas que simplesmente não correspondam ao que foi anunciado;

Por fim, a terceira forma de poluição de conteúdo consiste na inserção de identificadores inválidos para conteúdos, induzindo o sistema de busca de arquivos a falhas (LIANG; NAOUMOV; ROSS, 2006).

Em qualquer dos casos acima, o usuário legítimo que deseja obter o arquivo anunciado acaba sendo prejudicado e não tendo acesso ao arquivo original. Após freqüentes *downloads* sem sucesso na obtenção do arquivo desejado, o usuário pode sentir-se frustrado e deixar de utilizar o compartilhamento de arquivos via redes par-a-par (LIANG; KUMAR, 2005).

Nesse ponto, a motivação para um ataque de poluição de conteúdo parece um pouco obscura, pois não fica evidente quais os benefícios que um atacante pode obter ao impedir que outros pares tenham acesso a algum arquivo íntegro. Além disso, para que consiga seu objetivo, será necessário que invista uma parcela de seu tempo e recursos. A resposta para essa pergunta é que realmente existem entidades interessadas na não disseminação de conteúdo de forma gratuita como é feita nas redes par-a-par, a chamada *copyright industry*, ou a "indústria do direito autoral". São principalmente grandes gravadoras, estúdios cinematográficos e editoras que tentam impedir a distribuição não autorizada de conteúdo protegido por direitos autorais nas redes par-a-par (LIANG; NAOUMOV; ROSS, 2006). É interessante ressaltar que essa é uma técnica bastante difundida, existindo inclusive empresas especializadas tais como MediaDefender (MEDIADEFENDER, 2010) e PeerMedia (PEERMEDIA, 2010).

Para que um conteúdo poluído seja distribuído no lugar do correspondente legítimo, sua disponibilidade deve ser mais ampla, deve estar disponível no maior número possível de pares da rede. Uma forma de aumentar essa disponibilidade é através do uso de uma grande quantidade de identidades, todas elas disseminando o conteúdo poluído ao invés do legítimo. Um caminho eficiente para a obtenção dessas identidades por parte do usuário malicioso é lançar previamente um ataque *Sybil* à rede. Tal evidencia mais uma vez a importância do *Sybil* no lançamento de ataques mais elaborados como a própria poluição de conteúdo.

### 3.1.7 Conluio

Julian Grizzard *et al* (GRIZZARD; JOHNS, 2007) definem conluio como uma rede de máquinas comprometidas sob o controle de um único atacante. É importante notar que a participação na rede de ataque não é necessariamente voluntária, pois um atacante pode assumir o controle de máquinas que executem códigos maliciosos inadvertidamente (*worms*, *trojans*, entre outros). O objetivo do atacante é expandir sua capacidade de ataque através do uso do poder computacional de um conjunto de máquinas.

Inicialmente, essas redes de atacantes possuíam uma estrutra de comando e controle centralizada, em geral utilizando canais de IRC (*Internet Relay Chat*) para sua comunicação (BARFORD; YEGNESWARAN, 2007; COOKE; JAHANIAN; MCPHERSON, 2005). Mais recentemente, esse tipo de ataque evoluiu e surgiram redes de atacantes que utilizam a arquitetura par-a-par para sua comunicação. Enquanto uma estrutura centralizada de comando e controle pode ser derrubada com relativa facilidade ao ter seu centro de comando desconectado, uma estrutura descentralizada é muito mais difícil de ser desmontada. É difícil de medir a extensão dos danos que podem ser causados por redes de atacantes em conluio, mas é amplamente aceito que o dano é significativo (GRIZZARD; JOHNS, 2007).

Por representar um avanço no constante embate entre atacantes e defensores, espera-se que, no futuro, redes de atacantes que utilizem a arquitetura par-a-par tornem-se mais comuns (RUITENBEEK; SANDERS, 2008). Considerar a resistência a esse tipo de ataque passa a ser, portanto, uma medida não apenas interessante, mas mesmo necessária para quaisquer sistemas cujos objetivos estejam relacionados em prover segurança. Esse cuidado torna-se mais importante ainda em sistemas de redes par-a-par, como o apresentado nessa Dissertação. Para demonstrar a segurança da solução proposta, a avaliação é conduzida considerando diferentes cenários, entre eles um ataque em que diversos atacantes atuam em conjunto, em conluio.

## 3.2 Gerenciamento de Identidades em Redes Par-a-Par

Essa seção busca apresentar e caracterizar as principais abordagens utilizadas em redes par-a-par para a autenticação e gerenciamento de identidades dos participantes. Em especial, busca-se ressaltar as fragilidades dessas abordagens que acabam por permitir que usuários maliciosos obtenham vantagens indevidas ou prejudiquem o correto funcionamento das redes através de ataques. Além disso serão apresentados os principais trabalhos relacionados a cada uma das abordagens e que se propõem a empregar estratégias para solucionar, ou ao menos diminuir o impacto de ataques *Sybil*

Como mencionado anteriormente, redes par-a-par não apresentam um mecanismo efetivo de autenticação dos participantes ou de gerenciamento de suas identidades. Para resolver essa limitação, diferentes abordagens foram propostas, cada qual com características que oferecem segurança em maior ou menor escala. Invariavelmente, a oferta de maior segurança afeta outros aspectos importantes das redes par-a-par, tais como escalabilidade e descentralização. Barcellos e Gaspary (BARCELLOS; GASPARY, 2006) apontam as três abordagens existentes, listadas abaixo em ordem crescente de segurança e decrescente de escalabilidade:

- identificação gerada pelo próprio par;
- identificação obtida a partir de um outro par (mediante desafio);
- identificação obtida a partir de uma entidade certificadora;

No decorrer da presente seção, cada uma dessas abordagens será apresentada com maiores detalhes.

### 3.2.1 Identificação Atribuída pelo Próprio Par

Identificação atribuída pelo próprio par é a abordagem mais simples de todas, pois implica a não existência de um mecanismo global de atribuição ou verificação da identidade gerada. Cada par que tenha a intenção de participar da rede é livre para criar uma identificação própria, contanto que ela não esteja sendo usada atualmente. Essa identificação é assumida pelos outros pares como sendo verdadeira e adotada por eles na comunicação com esse novo participante, sem nenhuma outra espécie de verificação ou confirmação.

Essa abordagem apresenta algumas vantagens, e a principal delas é justamente sua simplicidade, quando comparada com os mecanismos apresentados nas próximas subseções. Esse tipo de gerenciamento de identidades apresenta a menor sobrecarga para que um novo participante ingresse na rede, visto não exigir verificação alguma da identidade gerada. Outra vantagem é o fato de ser o único que mantém mais estritamente as características-chave das redes par-a-par, tais como a descentralização da rede e conseqüente tolerância a falhas, pois, ao contrário dos outros mecanismos estudados, não introduz novas entidades na rede.

Ironicamente, as mesmas características que são consideradas vantagens nessa abordagem de gerenciamento de identidades, podem ser vistas como sérias desvantagens quando analisadas sob a ótica da segurança. A facilidade com que novas identidades podem ser criadas é uma delas. O baixo custo computacional, aliado à ausência de verificação da identidade gerada, permite a criação indiscriminada de identidades fantasmas. Como foi analisado no início deste capítulo, o controle de diversas identidades por uma mesma entidade constitui o ataque chamado *Sybil*.

As propostas que buscam resolver o problema da criação de identidades falsas em sistemas que utilizam a abordagem em que o próprio par cria sua identidade têm em comum o fato de estimarem limites para a quantidade de identidades *Sybil* na rede. Essa garantia, apesar de restrita, ainda pode ser útil em alguns cenários, pois é possível planejar aplicações que lidem adequadamente com uma fração previsível de pares *Sybil*. As soluções apresentadas por Yu Haifeng *et al.* (YU et al., 2006) e George Danezis *et al.* (DANEZIS et al., 2005) são exemplos. Ambas utilizam relações de confiança estabelecidas fora da rede par-a-par para estimar o limite máximo de identidades *Sybil* presentes na rede em um dado momento. No entanto, ambas as propostas apresentam problemas de violação de anonimidade, não restrição à existência de pares *Sybil* e não garantia de autenticidade dos pares.

### 3.2.2 Identificação Baseada em Certificados

A autenticação baseada em certificados difere das apresentadas anteriormente pelo fato de usar entidades externas à rede para assegurar a identificação dos pares. Essas entidades não participam nas interações do sistema, mas agem como facilitadoras (DATTA; HAUSWIRTH; ABERER, 2003). Cada participante da rede deve possuir um certificado expedido por uma entidade certificadora que será a responsável por assegurar a autenticidade da identidade. Os certificados tipicamente usam chaves assimétricas e são assinados pela entidade certificadora a qual garante, dessa forma, sua autenticidade. Assim, sempre que um par precisar conferir a identidade de um outro, ele deve primeiramente conferir a assinatura da entidade certificadora, constatando que ela provém de uma entidade acreditada, e, então, verificar a assinatura do par. Caso ambas as assinaturas confiram, então a autenticação do par estará correta.

Esse mecanismo apresenta algumas vantagens bastante interessantes para redes par-a-par. A principal delas é a incapacidade de que um usuário falsifique certificados, obviamente assumindo que as entidades certificadoras não estejam comprometidas. Sem poder criar novas identidades, ficam impedidos ataques como *Sybil*, Eclipse e lavagem de identidade, abordados no início desse capítulo. Por conseqüência, os ataques que deles dependem, ou se beneficiam, tais como *Free-riding* e poluição de conteúdo (ambos abordados nesse capítulo) também são evitados. Outra vantagem é a identificação segura de um par por outro, bastando para que isso aconteça que ambos acreditem na entidade certificadora.

Por outro lado, as desvantagens apresentadas por esse mecanismo são significativas, sendo a principal delas a necessidade de que a autoridade certificadora seja aceita por todos os participantes do sistema (KAUFMAN; PERLMAN; SPECINER, 2002). Essa necessidade pode tornar inviável o acesso de potenciais usuários (por exemplo, quando for necessário informar dados pessoais ou pagar taxas para obter uma identidade). De toda forma, um usuário pode escolher em quais entidades certificadoras ele acreditará e em quais não, porém assim surge o problema de como comunicar dois ou mais usuários que não compartilhem entidades certificadoras nas quais confiem (MORSELLI et al., 2006). Outra significativa desvantagem desse mecanismo é a centralização, que vai diretamente contra um dos princípios básicos das redes par-a-par e que traz consigo alguns problemas clássicos como a existência de ponto central de falha, problemas de balanceamento de carga no servidor, entre outros. Além disso, uma entidade certificadora que seja comprometida traz grande impacto negativo ao sistema, pois ela pode, entre outras ações, negar-se a fornecer certificados a alguns usuários, como também a revogar certificados comprometidos.

À primeira vista, essa parece ser a solução mais adequada a ser adotada em uma rede par-a-par, pois sob o aspecto da segurança é a única que resolve definitivamente o problema da criação de falsas identidades. Algumas soluções na literatura, como o *Pretty Good Privacy* (PGP) (ZIMMERMANN, 1995) e *Simple Public Key Infrastructure* (SPKI) (ELLISON, 1999; ELLISON et al., 1999), se propõem ainda a atacar alguns dos problemas surgidos com essa abordagem. Ainda, assim, no caso de PGP, certificados precisam ser trocados entre pares que, em princípio, não se conhecem *a priori*. A questão chave não resolvida nesse caso é: como fazê-lo de forma segura e escalável? Já SPKI, apesar de abordar questões como centralização e ponto único de falhas, não trata do problema de escolha do grupo de entidades certificadoras que serão aceitas na rede, nem da comunicação de pares que não confiem em entidades certificadoras em comum. Tal impede que sua adoção seja utilizada como padrão.

No outro extremo, as soluções para redes que utilizam a autenticação baseada em entidades certificadoras buscam minimizar as conseqüências danosas da introdução de um elemento central. Os trabalhos propostos por Ruggero Morselli *et al.* (MORSELLI et al., 2006) e Karl Aberer *et al.* (ABERER; DATTA; HAUSWIRTH, 2005) buscam descentralizar a infra-estrutura de distribuição de chaves públicas (do inglês *Public-Key Infrastructure* ou *PKI*). No entanto, essas propostas apresentam problemas pois exigem para seu correto funcionamento a troca de uma grande quantidade de mensagens entre os pares, além de apresentarem restrições quanto ao número mínimo de pares que devem colaborar para que o sistema funcione como esperado.

### 3.2.3 Identificação Obtida Mediante Desafio

O mecanismo de identificação mediante desafio é uma tentativa de solucionar o problema da facilidade para criação de múltiplas identidades existente na autenticação atribuída pelo próprio par. Essa abordagem se caracteriza pela existência de um serviço de *bootstrap*. Esse serviço é composto por um ou, possivelmente, mais servidores de autenticação, pontos esses responsáveis pela atribuição de identificadores a todos os usuários que desejam tomar parte na rede. Para que um par obtenha sua identificação, o serviço de *bootstrap* exigirá, em contrapartida, que seja resolvido um desafio computacional que demandará tempo e recursos por parte do par interessado em obter sua identidade. É um requisito que a verificação da solução seja uma tarefa trivial, ao contrário da efetiva resolução da mesma.

A idéia por trás da exigência da resolução do desafio é que usuários legítimos provem suas boas intenções com a rede, comprometendo uma parte de seus recursos. Ao mesmo tempo, os maliciosos interessados em criar múltiplas identidades são obrigados a passar grande parte de seu tempo consumindo recursos apenas na sua obtenção e não utilizando os recursos da rede de forma indevida. O ideal é que em qualquer ponto antes da autenticação, o custo do protocolo executado pelo cliente seja maior do que no servidor (AURA; NIKANDER; LEIWO, 2001), objetivando assim que o servidor de autenticação comprometa seus recursos o minimamente possível.

A proposta da utilização de desafios não é recente. Serviços tradicionais como e-mail e *web*, também estão sujeitos a abusos (*spam* e *denial of service*, respectivamente) e para eles já foram criadas abordagens com a intenção de impor um custo computacional como forma de impedir, ou ao menos diminuir, o impacto desses abusos (BORISOV, 2006). A identificação obtida mediante desafio em redes par-a-par busca adaptar o mesmo princípio, porém levando em consideração a característica descentralizada dessa arquitetura, em contraste com a centralização do modelo cliente-servidor e das soluções até então

existentes.

A maior vantagem desse mecanismo é justamente sua capacidade de diminuir a quantidade de identidades que podem ser criadas por um usuário malicioso, o que possivelmente impede, ou ao menos diminui, os danos causados por uma entidade com diversas identidades. Esse efeito é obtido, pois a cada nova identificação que um participante desejar obter, terá que resolver um novo desafio, diferente do anterior, o que lhe custará igualmente tempo e recursos. Além da resolução desse enigma inicial, algumas técnicas prevêem a resolução periódica de desafios pelos participantes de forma que mantenham seus identificadores (ROWAIHY et al., 2007), e, portanto, sua característica de membro da rede. Essa resolução periódica tem o potencial de restringir ainda mais a capacidade de criação de identidades falsas de usuários maliciosos ao comprometer seus recursos mais freqüentemente.

O mecanismo em questão apresenta, no entanto, algumas desvantagens. Segundo Nikita Borisov (BORISOV, 2006), a principal desvantagem é que a identificação obtida mediante desafio não é efetiva quando há uma grande diferença computacional entre os atacantes e usuários honestos. É plausível imaginar cenários em que os usuários legítimos tenham acesso a computadores de baixa capacidade de processamento quando comparados com usuários maliciosos dispostos a atacar a rede e que para isso utilizem equipamentos de última geração. Nesse cenário, os atacantes serão capazes de resolver desafios mais rapidamente. Apesar da dificuldade dos desafios empregados ser facilmente parametrizável, qualquer escolha fixa de dificuldade fatalmente restringirá usuários legítimos ou será incapaz de bloquear adequadamente os atacantes. Outra desvantagem desse mecanismo é que ao criar entidades centrais a que todos os participantes devem ter acesso, se está diminuindo a descentralização das redes par-a-par, reduzindo desta forma alguns de seus benefícios, como a tolerância a ataques de negação de serviço. Também se está reduzindo a escalabilidade, pois uma maior quantidade de participantes acarreta a necessidade de um maior número de entidades atribuidoras de identidades.

As propostas para conter a criação de *Sybils* em ambientes que utilizam a obtenção de identidades mediante desafios vêm obtendo bons resultados ao utilizarem desafios que são criados ou verificados de forma distribuída. O trabalho de Nikita Borisov (BORISOV, 2006) mostra a viabilidade da utilização de desafios gerados de forma distribuída e periódica e que contém dados gerados por todos os pares da rede. Dessa forma, um par pode verificar que o desafio resolvido por um outro par qualquer continha dados que ele próprio gerou. Segundo o autor, essa possibilidade garante que o sistema de geração e verificação de desafios não foi comprometido e o desafio resolvido é legítimo. Rowaihy *et al.* (ROWAIHY et al., 2007) propõem um esquema com múltiplas entidades geradoras de desafios. Para um usuário ingressar na rede é necessário contatar uma dessas entidades e resolver uma seqüência de desafios propostos. Após resolver todos eles, o próprio par torna-se capaz de gerar desafios para outros pares. Ambos os trabalhos mostram que a capacidade de criação de *Sybils* é bastante reduzida ao ponto de os ataques não serem efetivos.

Os trabalhos que abordam a utilização de desafios para contenção de *Sybils* apresentam um problema e uma crítica recorrente a esse modelo de identificação de usuários. Eles utilizam desafios atribuídos estaticamente, isto é, com uma mesma dificuldade para todos participantes. Qualquer escolha fixa de dificuldade pode não bloquear adequadamente os atacantes ou causar uma sobrecarga inaceitável aos usuários legítimos. Essa questão torna-se ainda mais importante ao considerar que atacantes podem ter acesso a equipamentos relativamente poderosos, com capacidade de processamento bastante supe-

riores a de usuários comuns. Nesse contexto, a utilização de um peso e uma medida na atribuição dos desafios tenderá a favorecer os atacantes, em detrimento dos usuários legítimos do sistema. O diferencial da proposta a ser apresentada é justamente parametrizar a dificuldade dos desafios para que esteja de acordo com o comportamento que cada usuário apresenta na rede. Pares legítimos, que se comportem de maneira colaborativa, serão beneficiados com desafios cuja resolução demande menos tempo e recursos. Por outro lado, os maliciosos, ou que se comportem de maneira não colaborativa, deverão resolver desafios mais custosos.

# 4 PROPOSTA DE SOLUÇÃO PARA COMBATER ATAQUES *SYBILS*

A solução proposta neste trabalho visa estabelecer o uso de desafios computacionais adaptativos para o gerenciamento de identidades em redes par-a-par. De uma forma geral, há três questões chave associadas à adoção de desafios adaptativos: (*i*) como caracterizar o comportamento das fontes (ou dos usuários associados às mesmas), (*ii*) como calcular o custo de um desafio a partir dos comportamentos observados, e (*iii*) como adaptar os desafios considerando a dinâmica do comportamento dos usuários da rede, observados pelo sistema como fontes de requisições de identidades. Cada uma dessas questões é abordada nas Seções a seguir.

## 4.1 Empregando *Taxas de Recorrências* para Caracterizar Comportamentos

Para permitir a caracterização do comportamento das diversas fontes de solicitação de identidades, duas métricas são introduzidas no contexto deste trabalho: *taxa de recorrência da fonte* ($\phi$) e *taxa de recorrência da rede* ($\Phi$). A primeira reflete a freqüência com que os usuários associados a uma determinada fonte solicitam novas identidades ao serviço de *bootstrap* da rede par-a-par em um intervalo de tempo $t_w$ (com $t_w > 0$). A segunda, por sua vez, reflete a freqüência média com que as fontes recorrem ao serviço de *bootstrap* para solicitar novas identidades.

O valor da taxa de recorrência da rede é calculado segundo a Equação 4.1, a qual utiliza a média harmônica das taxas de recorrências das fontes. Nessa equação, $n$ representa a quantidade de fontes de onde partiram solicitações de identidade e $\phi_i$ representa a taxa de recorrência da *i-ésima* fonte da rede par-a-par. Note que quando $\phi_i = 0$ equivale à situação em que nenhum usuário da fonte $i$ solicitou alguma identidade; nesse caso, tal fonte não é conhecida e, portanto, não é considerada para o cálculo da taxa de recorrência da rede.

$$\Phi = \frac{n}{\sum_{i=0}^{n} \frac{1}{\phi_i}} \tag{4.1}$$

É importante frisar que a média harmônica foi escolhida em detrimento de outras medidas estatísticas (média simples, média geométrica e a mediana) por ser especialmente resistente a alterações causadas por recorrências muito discrepantes do padrão observado (*outliers*). Essa característica é desejável e extremamente importante, uma vez que torna mais difícil para atacantes manipularem o comportamento da rede (por exemplo através

de um ataque em conluio) para tornarem-se menos suspeitos. A resistência da taxa de recorrência da rede a ataques em conluio é avaliada em maior profundidade no Capítulo 5.

## 4.2 Calculando o Grau de Confiança a partir dos Comportamentos Observados

Considerando que o objetivo de um ataque *Sybil* é controlar uma fração significativa de identidades na rede par-a-par, para executá-lo o atacante deverá solicitar um grande número de identidades ao serviço de *bootstrap*. A consequência direta desse comportamento é um aumento da taxa de recorrência da fonte associada ao atacante. Por outro lado, é esperado que as fontes com usuários legítimos recorram minimamente para solicitar identidades (por exemplo, no momento que se registrarem na rede par-a-par). Logo, a idéia principal para conter ataques *Sybil* é atribuir desafios mais complexos ao(s) usuário(s) associado(s) às fontes cujas taxas de recorrência se tornarem superiores à taxa de recorrência da rede.

A partir da comparação entre o comportamento de cada fonte (inferido a partir de $\phi$) e do comportamento considerado padrão para a rede (inferido a partir de $\Phi$), é calculada a *relação entre as taxas de recorrências da fonte e da rede* ($\rho$). Obtida de acordo com a Equação 4.2, ela assume valores menores que zero para indicar quantas vezes a taxa de recorrência da fonte *i* é menor que a da rede, e maiores que zero para indicar quantas vezes a taxa de recorrência da fonte *i* é maior.

$$\rho = \begin{cases} -\frac{\Phi(t)}{\phi_i(t)} & \text{se } \phi_i(t) < \Phi(t) \\ \frac{\phi_i(t)}{\Phi(t)} & \text{se } \phi_i(t) \geq \Phi(t) \end{cases} \tag{4.2}$$

A relação entre as taxas de recorrências da fonte e da rede ($\rho$) serve como base para o cômputo do *grau de confiança da fonte* origem das solicitações de identidade ($C$). Esse grau, estimado para cada instante $t$ de acordo com a Equação 4.3, assume valores no intervalo $[0, 1]$: em um extremo, valores mais próximos de 1 indicam uma maior confiança sobre a legitimidade do(s) usuário(s) associado(s) à fonte em questão; no outro extremo, valores mais próximos de 0 indicam maior desconfiança, isto é, uma maior probabilidade de que o(s) usuário(s) associados à fonte em questão está(ão) lançando um ataque *Sybil*. A complexidade do desafio computacional aplicado ao(s) usuário(s) será determinada pelo grau de confiança da fonte ao(s) qual(is) ele(s) está(ão) associado(s), no momento da solicitação de uma nova identidade. A Equação 4.3 é normalizada de modo que os extremos 0 e 1 representem total desconfiança e total confiança sobre uma determinada fonte, respectivamente.

$$C(t) = 0.5 - \frac{\arctan(a \times (\rho - c)^{(1+2\times b)})}{\pi} \tag{4.3}$$

A Figura 4.1 mostra quatro diferentes configurações que ilustram como varia o grau de confiança obtido para uma determinada fonte em função de $\rho$. Nessas configurações, os termos $a$, $b$ e $c$ da Equação 4.3 assumem valores arbitrários e desempenham um importante papel no controle da agressividade com que as configurações decrescem, da amplitude e da translação das mesmas, respectivamente.

A partir da Figura 4.1 é possível observar duas propriedades importantes que a Equação 4.3 apresenta. A primeira reside no fato do valor de confiança ter variações mínimas para valores de $\rho$ mais próximos de 0 (situação em que a fonte se comporta de forma semelhante ou igual à média da rede), proporcionando assim uma certa tolerância na avaliação

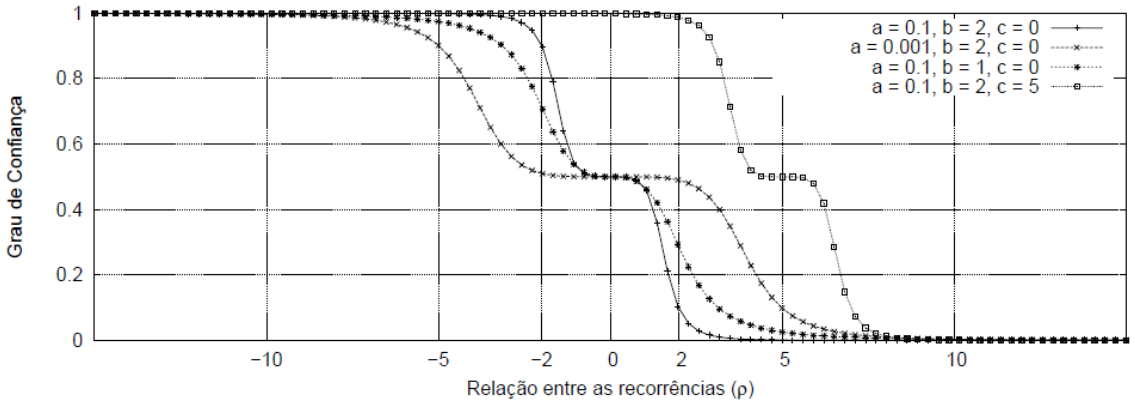



Figura 4.1: Exemplos de valores para os parâmetros $a$, $b$, e $c$ da Equação 4.3 para cálculo do grau de confiança da fonte

dos comportamentos das fontes. Considerando por exemplo a configuração ($a = 0,1$ , $b = 2$ , $c = 0$) na Figura 4.1, dentro do intervalo $-2 \leq \rho \leq 2$, variações são pouco consideradas, por serem ligeiramente semelhantes ao padrão observado na rede. Os comportamentos que desviam significativamente desse intervalo, no entanto, terão atribuídos menores (ou maiores) valores de confiança, como pode ser observado pelas súbitas variações da configuração ($a = 0,1$ , $b = 2$ , $c = 0$) nos intervalos $-5 \leq \rho \leq -2$ e $2 \leq \rho \leq 5$. A segunda propriedade reside no fato de ser assintótica em 0 e 1. Desse modo, para $\rho \to -\infty$ ou $\rho \to +\infty$, sempre haverá um valor de confiança associado.

## 4.3 Lidando com a Dinâmica do Comportamento dos Usuários da Rede

Uma característica importante de redes par-a-par é a ampla autonomia concedida aos pares. Desse modo, os pares podem entrar e sair da rede de acordo com seus interesses e disponibilidade, sem depender de entidades externas. Um dos possíveis desdobramentos dessa dinamicidade é a ocorrência de variações constantes (e eventualmente significativas) do padrão de comportamento tanto das fontes quanto da rede par-a-par como um todo. A seguir, é discutido como o mecanismo proposto lida com a dinâmica dos comportamentos observados.

A Equação 4.3, embora seja capaz de determinar a confiança de uma determinada fonte no instante $t$, não considera o histórico de comportamento da mesma. Com o objetivo de representar de modo apropriado o grau de confiança de uma determinada fonte, ao mesmo tempo considerando o histórico do comportamento da mesma, é inserido na solução um parâmetro $\beta$, o qual permite o cálculo da confiança suavizada, $C_s$, conforme apresentado na Equação 4.4. O parâmetro $\beta$ é um fator de suavização que determina o peso do passado no cálculo do valor de confiança para o instante atual ($t$), e assume valores no intervalo (0,1]. Em um extremo, valores de $\beta$ mais próximos de 0 conferem um peso maior ao comportamento histórico da fonte em questão. Em outro extremo, valores de $\beta$ mais próximos de 1 dão um peso maior ao comportamento atual da fonte. No caso especial em que $\beta = 1$, o valor de confiança atual (tal como calculado pela Equação 4.3) é considerado integralmente, sendo o passado totalmente desconsiderado.

$$C_s(t) = \beta \times C(t) + (1 - \beta) \times C(t - 1) \tag{4.4}$$

A adição do parâmetro $\beta$ ao cálculo da confiança é importante para tratar adequadamente as alterações no comportamento de cada fonte de solicitação de identidades. Em particular, as alterações intencionais e repentinas de comportamento, de usuários interessados em obter benefícios, como os *traidores*, são capturadas e refletidas no grau de confiança da fonte à qual o mesmo está associado. Um traidor é um atacante que busca angariar altos valores de confiança em sistemas de reputação e passa, então, a se aproveitar dela para prejudicar outros pares, ou obter vantagens indevidas. O correto dimensionamento do valor de $\beta$, nesse contexto, pode impedir que um traidor manipule a solução proposta de modo que a fonte em que se situa consiga (ou recupere) rapidamente uma alta confiança do sistema. Na medida em que o passado é considerado para determinar o presente, somente aquelas fontes cujos usuários apresentem bom comportamento histórico serão considerados confiáveis.

Outra questão importante, ainda em relação à dinâmica do comportamento da rede, reside no fato de que as taxas de recorrência podem variar em épocas diferentes. É razoável esperar que o comportamento dos usuários se altere com o passar do tempo. Por exemplo, a distribuição de um novo arquivo de grande sucesso em uma rede par-a-par de compartilhamento de arquivos tende a aumentar a taxa de recorrência da rede devido à grande quantidade de usuários interessados em obter esse arquivo. É preciso evitar que o cálculo do grau de confiança seja feito com base em comportamentos que não mais reflitam o padrão observado mais recentemente. Outra motivação para que sejam limitadas temporalmente as solicitações de identidade consideradas no cálculo do grau de confiança é que caso fossem consideradas todas desde o início ($t = 0$), seria mais fácil para um atacante lançar mão de ataques *Sybil*. Isso seria possível visto que a quantidade de requisições cresceria indefinidamente, conseqüentemente ofuscando as altas taxas de recorrência de fontes suspeitas.

Para acomodar essas questões temporais no comportamento dos pedidos de identidades, optou-se por utilizar uma *janela deslizante* – um intervalo de tempo $t_w$, que se inicia no passado e termina no momento presente – para restringir a quantidade de requisições a serem consideradas no cálculo das taxas de recorrência de cada fonte e da rede. Note que $t_w$ corresponde ao tempo considerado para calcular a taxa de recorrência $\phi$ de cada fonte no sistema e, conseqüentemente, a taxa de recorrência da rede, $\Phi$ (tal como discutido na Seção 4.1). À medida em que o tempo passa, a janela avança em passos com duração $t_d$ (com $t_d \leq t_w$); com isso, as solicitações de identidade mais antigas vão sendo desconsideradas, dando lugar a solicitações mais recentes, as quais são mais representativas do estado atual da rede par-a-par.

# 5 AVALIAÇÃO DA SOLUÇÃO PROPOSTA

Para avaliar a viabilidade técnica, a eficácia e a eficiência do uso de desafios adaptativos para combater *Sybil* em redes par-a-par, foi realizada a implementação prototípica de um serviço de *bootstrap*. Por meio desse protótipo, foram executados diversos experimentos, considerando solicitações sintéticas de identidades baseadas em traços históricos de uma comunidade par-a-par real. Como resultados da avaliação conduzida, procurou-se observar que (*i*) os desafios computacionais propostos para usuários legítimos penalizam minimamente os mesmos, (*ii*) desafios atribuídos a potenciais atacantes possuem maiores complexidades computacionais, e (*iii*) a solução proposta é robusta e resiliente mesmo na presença de uma fração significativa de atacantes, bem como sob a ocorrência de ataques em conluio.

O restante deste capítulo está organizado como segue. A Seção 5.1 apresenta a metodologia de avaliação utilizada. Na Seção 5.2 é apresentada uma análise dos parâmetros utilizados no cálculo do grau de confiança com o intuito de verificar a influência que diferentes valores trazem para esse cálculo e auxiliar na escolha de valores adequados visando a eficácia e eficiência da solução proposta. A Seção 5.3 apresenta a configuração utilizada na avaliação experimental, que terá seus resultados acerca da contenção de ataques *Sybil* apresentados na Seção 5.4.

## 5.1 Metodologia de Avaliação

O protótipo desenvolvido para a avaliação experimental pode ser visto como um serviço que recebe como entrada uma série de requisições de identidade. Cada requisição é composta por um identificador da fonte que originou a requisição e um indicador do momento em que ela foi gerada. Esse serviço possui configurações relativas aos parâmetros (a, b, c, duração e passo da janela, e $\beta$) utilizados nas equações que influenciam o cálculo do grau de confiança. Esse serviço efetua o processamento das requisições de forma semelhante ao que um servidor de *boostrap* de uma rede par-a-par faria caso recebesse um pedido de identidade de um usuário. Como saída o serviço gera uma lista associando cada requisição com o grau de confiança calculado de acordo com o explicado no Capítulo 4.

A Tabela 5.1 apresenta um resumo das características do traço utilizado como base para a montagem da lista de requisições que é fornecida como entrada para o serviço. Essa lista é utilizada tanto na análise de sensibilidade dos parâmetros para o cálculo do grau de confiança (Seção 5.2), quanto nos experimentos efetuados para análise da eficácia e eficiência da proposta (Seção 5.4).

Os experimentos e a análise de sensibilidade foram feitos com base em traços históricos de solicitações de identidades na comunidade par-a-par fechada Bitsoup (BITSOUP.ORG, 2009). Uma vez que a admissão na comunidade é feita mediante autentica-

Tabela 5.1: Informações sobre o traço utilizado na análise de sensibilidade e avaliação experimental.

| Características do Traço Empregado | |
|---|---|
| Duração | 15 dias |
| Quantidade de identidades solicitadas | 625.079 identidades |
| Número de fontes distintas | 44.315 fontes |
| Intervalo médio entre requisições | 2,08 segundos |
| Quantidade média de requisições por fonte | 14,21 identidades |

ção por usuário e senha, e a criação de novas contas é moderada, assumiu-se para os fins da avaliação que o traço não contém registros de atacantes *Sybil*. Essa premissa baseia-se na idéia de que o custo necessário para criar e manter diversas identidades falsas em uma comunidade fechada com moderação é de várias ordens de grandeza maior do que em uma comunidade aberta sem moderação.

Os traços considerados registram atividades de solicitação de identidades durante 15 dias consecutivos. Durante esse período, foram obtidas $625.079$ identidades, solicitadas por $44.315$ fontes distintas, perfazendo uma média de $14,21$ solicitações de identidades por fonte, e uma taxa global de 1 solicitação a cada $2,08$ segundos.

Um dos objetivos do modelo proposto é penalizar potenciais atacantes (ou usuários que tenham comportamento semelhante). Para que que a avaliação experimental do modelo proposto pudesse ser avaliada nessas condições foi preciso considerar cenários em que a rede se encontrava sob ataque *Sybil*. Com esse intuito, foram gerados ataques sintéticos sob duas formas. Primeiramente, foram considerados ataques em que uma única fonte produzia requisições de identidade com uma taxa variável e crescente, simulando o caso de um atacante estar angariando identidades na iminência de lançar um ataque *Sybil*. A outra forma de ataque consistiu em diversas fontes requisitando identidades a uma taxa constante, simulando o caso em que um atacante estaria orquestrando um conluio contra a rede.

## 5.2 Análise de Sensibilidade

Essa seção apresenta uma série de experimentações efetuadas com o intuito de compreender melhor a influência de cada um dos parâmetros utilizados no cálculo do grau de confiança de uma requisição. Essa análise é importante para determinar valores que sejam considerados adequados tendo em vista seu uso em redes com diferentes perfis. A Tabela 5.2 apresenta os valores considerados para cada um dos parâmetros envolvidos.

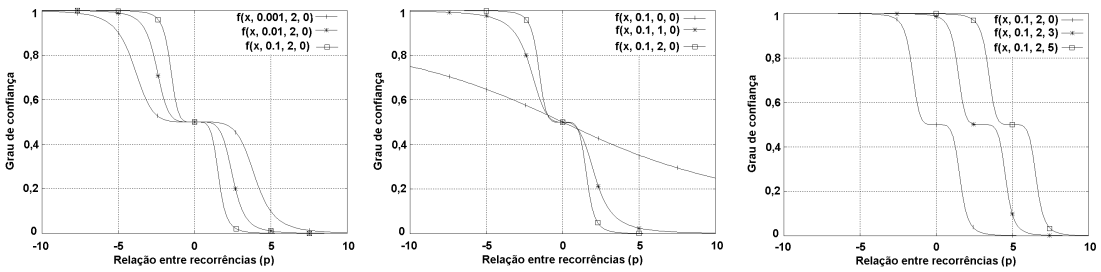
Tabela 5.2: Valores considerados na análise de sensibilidade.

| Parâmetros considerados | |
|---|---|
| $a$ | 0,001; 0,01; 0,1 |
| $b$ | 0; 1; 2 |
| $c$ | 0; 3; 5 |
| $\beta$ | 0,125; 0,5; 0,875 |
| Duração da Janela ($t_w$) | 4; 8; 16 horas |
| Passo da Janela ($t_d$) | 1; 2; 4 horas |

### 5.2.1 Parâmetros *a*, *b* e *c*

A análise dos valores para os parâmetros *a*, *b* e *c* da Equação 4.3 foi feita de forma analítica. Isto ocorreu porque esses três parâmetros apenas determinam o formato da curva que associa a relação entre as recorrências ($\rho$) e o grau de confiança C(t) associado.

A Figura 5.1(a) ilustra a influência do parâmetro *a* no cálculo do grau de confiança. Pode-se observar que conforme aumenta o valor de *a*, menor fica o segmento da reta paralelo ao eixo x situado na altura de 0,5 do eixo y. Em termos do modelo proposto, esse segmento de reta pode ser considerado como o intervalo de tolerância para o valor de recorrência classificado como comportamento padrão, pois variações no valor de $\rho$ (a relação entre a recorrência do par e a da rede) representado no eixo x, não implicarão alterações no grau de confiança. Assim, quanto menor o valor de *a*, mais pares, potencialmente, terão suas requisições com grau de confiança valendo 0,5. Isso equivale a considerá-los como pares "normais", pois este é o ponto médio do intervalo de variação do grau de confiança.




(a) Influência de *a* no cálculo de C(t)

(b) Influência de *b* no cálculo de C(t)

(c) Influência de *c* no cálculo de C(t)

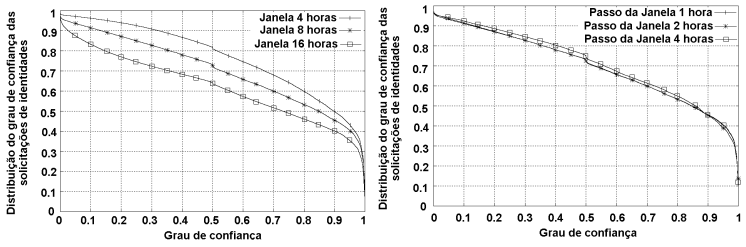
Figura 5.1: Influência dos parâmetros *a*, *b* e *c* no cálculo de C(t)

A Figura 5.1(b) mostra a influência do parâmetro *b*. Nela pode-se notar que à medida em que *b* aumenta, mais agressiva é a subida da curva em relação ao eixo y. Quando *b* vale 1, a curva percorre o intervalo entre 0 e 1 do eixo y, aproximadamente, entre os pontos -10 e 10 do eixo x. Já quando *b* vale 2, ela sobe rapidamente, praticamente percorrendo esse mesmo trecho apenas entre os pontos -4 e 4 do eixo x. Em termos do modelo proposto pode-se dizer que o parâmetro *b* da Equação 4.3 serve para estabelecer o quanto os pares serão punidos (ou beneficiados) caso seu comportamento destoe do padrão calculado para o restante da rede.

Por fim, a Figura 5.1(c) mostra a influência do parâmetro *c*. Nela pode ser observado claramente a influência desse parâmetro na translação da curva. Em termos do modelo proposto, o parâmetro *c* pode ser utilizado para aumentar ou diminuir globalmente o grau de confiança para determinados valores de ($\rho$). Esse parâmetro pode ser considerado como o quão benevolente (ou malevolente) o modelo se comporta. por exemplo ao atribuir o valor 5 para *c*, aqueles pares que apresentarem sua taxa de recorrência $\phi$ igual a taxa de recorrência ($\Phi$) da rede (e portanto ($\rho$) igual a 0) terão um grau de confiança próximo de 1. Já no caso em que *c* vale 0, pares com esse mesmo comportamento terão seu grau de confiança calculado em 0,5.

### 5.2.2 Duração e Passo da Janela

A janela ($t_w$) é um parâmetro importante para o cálculo do grau de confiança, ainda que não esteja explícito na Equação 4.3 para o cálculo de C(t). Sua função é delimitar no tempo quais requisições serão considerada para o cálculo da *taxa de recorrência da fonte* ($\phi$) e da *taxa de recorrência da rede* ($\Phi$). Portanto influencia a partir dessas variáveis o valor de C(t).




(a) Influência da duração da janela no cálculo de C(t)

(b) Influência do passo da janela no cálculo de C(t)

Figura 5.2: Influência da duração e passo da janela no cálculo de C(t)

A duração de $t_w$ não pode ser muito longa, pois isso pode acabar tornando-se um problema. É razoável assumir que o comportamento dos participantes de uma rede par-a-par tem influência direta do horário do dia, com mais participantes usando a rede no período noturno. Se a duração de $t_w$ fosse próxima de um dia ela acabaria por juntar dois comportamentos distintos, o diurno e o noturno, e fatalmente o comportamento resultante não seria fiel ao comportamento mais atual da rede. Por outro lado, com a duração da janela muito curta corre-se o risco de não ser possível captar o comportamento padrão da rede, sofrendo maior influência de pares com comportamento distinto. Por esses motivos foi decidido fazer experimentos considerando a janela com duração de 4, 8 e 16 horas. A Figura 5.2(a) ilustra a relação entre a duração da janela e a distribuição do grau de confiança entre os pares. Como esperado, o grau de confiança calculado para a rede como um todo (isto é, a média das confianças calculadas para todos os pares da rede) é diretamente proporcional à duração da janela.

Além da duração, a janela possui outra característica chamada de *passo*, que determina o quanto ela avança no tempo por vez. É razoável supor que o comportamento geral da rede par-a-par (indicado por $\Phi$) seja aproximadamente constante. Dessa maneira, o fato da janela avançar no tempo em saltos de 1, 2 ou 4 horas não deve trazer diferenças significativas na distribuição do grau de confiança das solicitações de identidades. Realmente, como pode ser observado na Figura 5.2(b), o passo da janela não apresenta influência significativa no cálculo do grau de confiança.

### 5.2.3 Ponderação $\beta$ para Consideração de Comportamentos Passados e Recentes

O parâmetro $\beta$ foi criado com a intenção de ponderar a importância relativa de valores de confiança passados e atual no cálculo do grau de confiança. Com isso busca-se evitar traidores, pares que repentinamente alteram seu comportamento visando a obtenção de vantagens indevidas. Acredita-se que quanto mais próximo de 1 (priorização do valor atual em detrimento do passado) for o valor de $\beta$, mais vai variar o grau de confiança. Isso ocorre, pois mesmos em traços supostamente contendo apenas usuários legítimos,

algumas fontes permanecem longos períodos sem requisitar identidades, mesmo que no passado tenham requisitado diversas. Assim, seu grau de confiança instantânea seria bastante alto quando comparado com seu passado. Quando o valor de $\beta$ for mais próximo de 0 (valorização de mediçõs passadas no cálculo do grau de confiança) , espera-se que a distribuição dos graus de confiança dos pares varie menos abruptamente, pois essa ponderação tende a suavizar as mudanças que podem ocorrer no comportamento de pares com o decorrer do tempo. Como pode ser observado na Figura 5.3 tal é exatamente o que ocorre, confirmando o que intuitivamente era esperado.

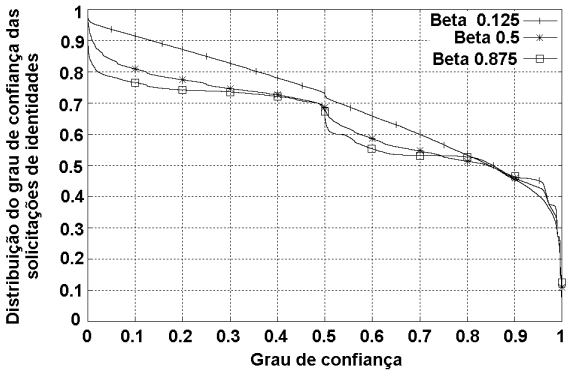



Figura 5.3: Influência do $\beta$ no cálculo de C(t)

## 5.3 Configuração do Ambiente de Experimentação

Com a análise de sensibilidade apresentada na seção anterior, foi possível selecionar um conjunto de adequado de parâmetros para conduzir a avaliação expeimental do modelo. Esses parâmetros estão sintetizados na Tabela 5.3 e explicados a seguir

Tabela 5.3: Informações sobre o ambiente considerado na avaliação experimental.

| Parâmetros do Modelo | |
|---|---|
| $a$ | 0,1 |
| $b$ | 2 |
| $c$ | 5 |
| $\beta$ | 0,125 |
| Duração da Janela ($t_w$) | 8 horas |
| Passo da Janela ($t_d$) | 1 hora |
| **Estratégias de Ataque** | |
| Taxa de requisição por fonte atacante | 1; 1,25; 1,5; 2; e 2,5 requisições/hora |
| Quantidade de fontes atacantes | 1; 100; 500; 1.000; e 2.000 |

O valor de $\beta$ foi definido como $0,125$, isto é, o comportamento histórico do grau de confiança possui um peso de 87,5% sobre o valor atual. Esses valores mostraram-se adequados, após sucessivas experimentações, para impedir que fontes com histórico de *mau comportamento* alcançassem valores elevados de confiança ao tornarem-se repentinamente *bem comportadas*. Os valores de $a$, $b$ e $c$, por sua vez, foram definidos como $0, 1$, $2$ e $5$, respectivamente, visando controlar a forma como a relação entre as recorrências se reflete no grau de confiança obtido pela fonte. Com esses valores, uma taxa de

recorrência da fonte similar ou igual a da rede ($\phi \simeq \Phi$) fará com que a fonte alcance um grau de confiança de aproximadamente 1 (por exemplo, vide configuração $a = 0,1$ , $b = 2$ e $c = 5$ na Figura 4.1). A janela deslizante tem duração de 8 horas ($t_w = 8 \times 60$ *min*) e desliza de hora em hora ($t_d = 1 \times 60$ *min*). A duração de 8 horas mostrou-se adequada considerando as características da comunidade Bitsoup.org (materalizadas nos traços históricos estudados), sendo capaz de capturar adequadamente o comportamento passado de cada usuário, e ao mesmo tempo desconsiderar solicitações que não refletem mais o estado atual da rede. O deslizamento de hora em hora, por sua vez, mostrou-se adequado para capturar a evolução nos comportamentos dos participantes da rede, sem impor uma sobrecarga maior ao processo de cálculo do grau de confiança.

Para avaliar cenários em que a rede se encontra sob ataque *Sybil*, foram injetadas artificialmente solicitações maliciosas de identidades, considerando duas estratégias diferentes. Na primeira, o atacante lança um ataque *Sybil* a partir de uma única fonte. A segunda estratégia, por sua vez, considerou que o atacante possui a sua disposição um determinado número de fontes. Nesse caso, o ataque *Sybil* realizado é distribuído, com solicitações partindo de cada uma das fontes sob o controle do atacante – cada fonte solicita uma quantidade pequena de identidades, de modo que não sejam classificadas como suspeitas. Em ambos os casos, busca-se avaliar a quantidade de identidades que um atacante consegue solicitar por meio do ataque *versus* a dificuldade do desafio que o sistema atribui para cada nova solicitação vinda de uma das fontes envolvidas no ataque.

## 5.4 Resultados Obtidos e Análise

Para organizar a discussão dos resultados obtidos, primeiramente é discutida a sobrecarga causada a usuários legítimos, em situações em que não há ocorrência de ataques *Sybil* na rede par-a-par. Em seguida, é avaliada a efetividade da solução na contenção de ataques *Sybil*, e o impacto que estes causam nos desafios com os quais usuários legítimos terão de arcar. Por fim, é analisada a sua resiliência em situações em que diversos atacantes agem em conluio, com o propósito específico de atacar a própria solução.

### 5.4.1 Sobrecarga Causada a Usuários Legítimos na Ausência de Ataques *Sybil*

A Figura 5.4 exibe a função cumulativa complementar de distribuição (*Complementary Cumulative Distribution Function*, *CCDF*) das confianças calculadas para as solitações de identidades partindo das fontes (consideradas legítimas) do traço estudado. É importante ressaltar que esse resultado refere-se somente às solicitações de identidade presentes no traço original, não tendo sido perturbado pela ocorrência de ataques *Sybil*.

É possível notar no gráfico da Figura 5.4 que a maioria das solicitações de identidade geradas pelas fontes é de alta confiança. Por exemplo, aproximadamente 45% das solicitações de identidades foram realizadas por usuários oriundos de fontes com confiança maior ou igual a $0,9$. Em outras palavras, existe um grau de confiança igual ou maior do que $0,9$, para aproximadamente 45% das solicitações, de que as mesmas não estejam relacionadas a um ataque *Sybil*. Esse percentual aumenta para 60% se considerarmos as solicitações com confiança maior ou igual a $0,7$, e para aproximadamente 75% se considerarmos aquelas com confiança maior ou igual a $0,5$. Com esses valores de confiança obtidos, uma significativa fração das fontes obterá desafios computacionais de menor complexidade, logo causando mínimo impacto para os respectivos usuários.

Um aspecto importante a ser discutido sobre os resultados da Figura 5.4 diz respeito aos 25% das fontes com confiança menor que $0,5$. Embora as fontes contidas no traço

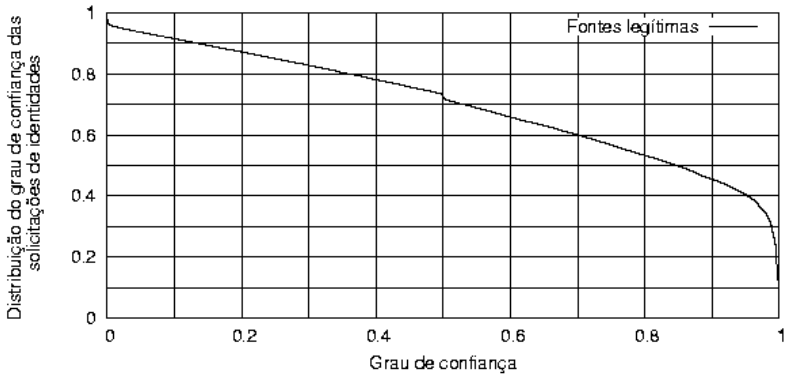



Figura 5.4: *CCDF* do grau de confiança de solicitações de identidades originadas por fontes legítimas

sejam presumidamente legítimas (isto é, não lançaram algum ataque *Sybil* contra a rede par-a-par), existem casos em que as fontes podem recorrer mais vezes que a média da rede para solicitar identidades. Esse é o caso, por exemplo, em que vários usuários acessam a Internet através de redes utilizando o mecanismo de NAT, o qual faz com que os mesmos sejam associados à uma única fonte. De qualquer forma, o número de usuários afetados no experimento executado foi mínimo. Pouco mais de 10% das fontes alcançou valores de confiança menores ou iguais a $0,2$.

### 5.4.2 Impacto Causado a Potenciais Atacantes

A Figura 5.5 apresenta os resultados obtidos a partir do desdobramento do cenário ilustrado na Figura 5.4 em cinco novos cenários, cada um sob os efeitos de um ataque *Sybil* gerado artificialmente. Os ataques, em cada um dos cenários, são orquestrados por uma única fonte (maliciosa). A principal diferença entre os ataques de cada cenário reside nas taxas de solicitação de identidades adotadas: $1$; $1,25$; $1,5$; $2$; e $2,5$ solicitações por hora, respectivamente.

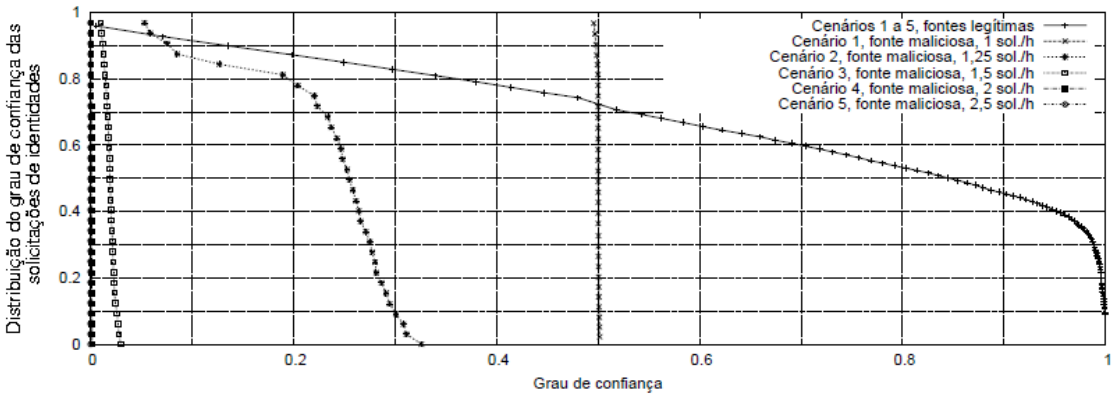



Figura 5.5: Resistência da solução proposta a ataques *Sybil* partindo de uma única fonte maliciosa, considerando diferentes taxas de recorrência da mesma

Uma observação importante em relação aos resultados apresentados na Figura 5.5 corresponde à influência do ataque *Sybil* sobre o grau de confiança obtido pelas fontes legítimas. Independente da taxa de solicitação de identidades adotada pelo atacante, as curvas que mostram a distribuição do grau de confiança dos pares legítimos mantêm-se

inalteradas e idênticas. Tal se deve à resistência da média harmônica – medida empregada para calcular a taxa de recorrência da rede, conforme discutido na Seção 4.1 – à presença de taxas de recorrência com desvio significativo em relação as das demais fontes. Por questões de legibilidade, apenas uma curva é apresentada na Figura 5.5 para ilustrar a distribuição do grau de confiança das fontes legítimas.

Analisando os resultados da Figura 5.5 por uma perspectiva diferente, é possível observar que um aumento gradual na taxa de recorrência da fonte maliciosa é suficiente para que a mesma sofra quedas significativas no seu grau de confiança. Por exemplo, quando a taxa de recorrência da fonte maliciosa corresponde a 1 solicitação por hora (cenário 1), aproximadamente 100% das solicitações de identidades partindo daquela fonte obteve grau de confiança igual a $0,5$ (isto é, um grau de confiança de $0,5$ de que a solicitação não está relacionada a um ataque *Sybil*). Para a taxa de $1,25$, por sua vez, apenas 10% das solicitações de identidades obteve grau de confiança maior ou igual a $0,3$. No cenário 5, o mais extremo ilustrado, a taxa de $2,5$ faz com que todas as solicitações de identidades partindo da fonte maliciosa sejam consideradas como parte de um ataque *Sybil* (uma vez que 100% das solicitações de identidade obteve grau de confiança 0). A consequência direta das quedas observadas é a imposição, aos usuários associados à fonte maliciosa, de desafios computacionais de complexidade computacional extrema.

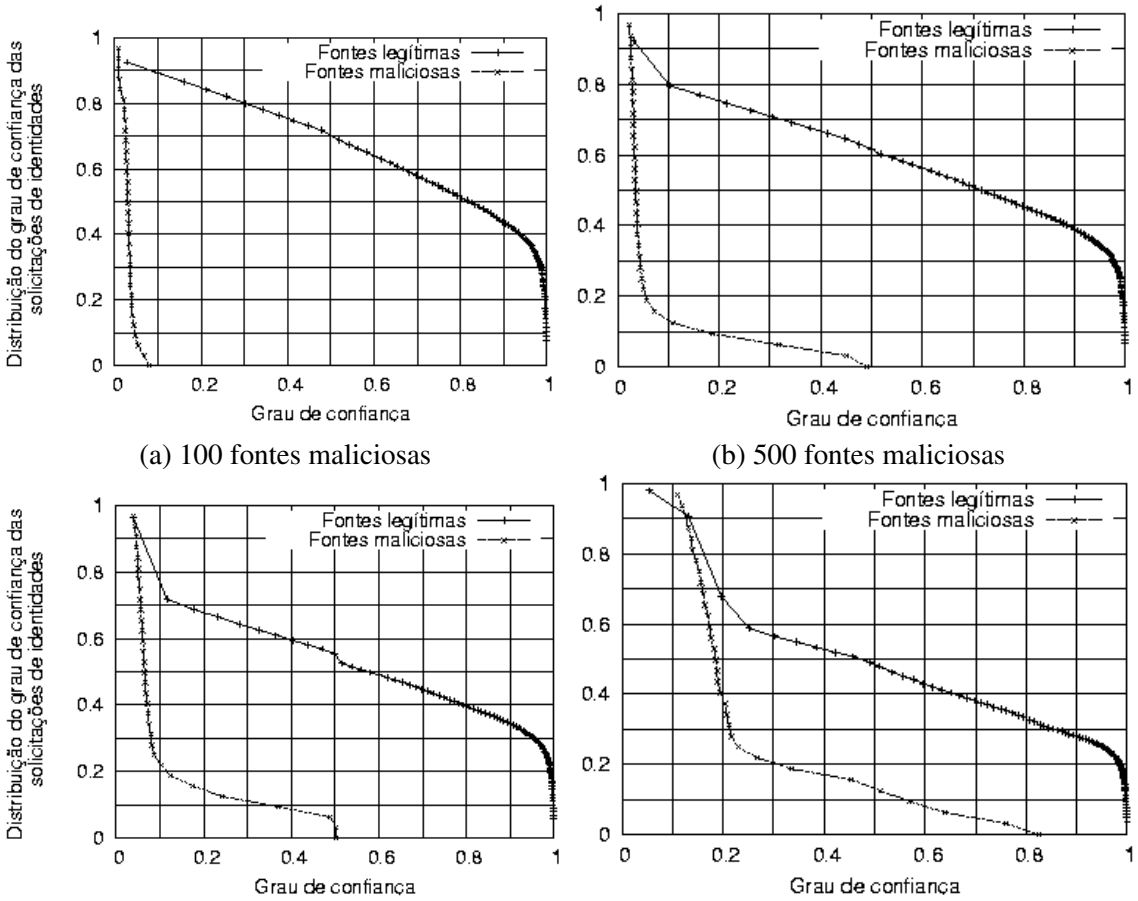
Os resultados apresentados levam a duas conclusões distintas, dependendo da perspectiva pela qual são analisados. Por um lado, evidenciam que a solução proposta reage adequadamente ao aumento na taxa de recorrência das fontes, penalizando severamente aquelas que recorrem a uma taxa muito maior que a média observada na rede. Por outro lado, mostra que a solução compele as fontes a se "comportarem adequadamente" – isto é, recorrendo harmonicamente em relação às demais fontes – caso não desejem ser penalizadas com desafios computacionais mais complexos.

### 5.4.3 Resiliência da Solução Proposta a Ataques em Conluio

Após ter-se analisado o efeito de um ataque *Sybil* considerando uma única fonte, avaliou-se o efeito do ataque realizado de forma distribuída, isto é, considerando múltiplas fontes como origem das solicitações de identidade. Nesse caso, ao invés de aumentar a taxa de recorrência para obter mais identidades falsas, o atacante age em conluio com outros atacantes (ou lança mão de uma *botnet* formada por várias estações zumbi conectadas à Internet). São dois os objetivos do lançamento de um ataque *Sybil* em conluio. Primeiro, busca-se aumentar a velocidade com que o atacante obtém identidades falsas na rede par-a-par, sem ter de arcar com desafios mais complexos. Segundo, procura-se alterar a percepção de normalidade da rede. Em outras palavras, parte-se da idéia de que mais fontes maliciosas atuando com o mesmo comportamento tende a mudar a percepção sobre qual é, efetivamente, o comportamento da maioria das fontes na rede.

A Figura 5.6 apresenta os resultados obtidos considerando a nova estratégia de ataque. Quatro cenários distintos são considerados, cada um com um número distinto de fontes maliciosas à disposição do atacante: 100; 500; 1000; e 2000 fontes. Em todos os cenários, cada fonte maliciosa atua a uma taxa de $1,5$ solicitações de identidades por hora. Essa taxa foi escolhida porque permite ao atacante obter um número significativo de identidades e, ao mesmo tempo, passar mais despercebido como um atacante na rede (conforme evidenciado na análise anterior).

Observe na Figura 5.5 que, mesmo utilizando uma quantidade extremamente alta de fontes para lançar o ataque *Sybil*, o efeito que o atacante consegue exercer sobre o padrão de normalidade da rede é relativamente limitado. Por exemplo, na Figura 5.5 (a), 70%




(a) 100 fontes maliciosas

(b) 500 fontes maliciosas

(c) 1000 fontes maliciosas

(d) 2000 fontes maliciosas

Figura 5.6: Resiliência da solução proposta a ataques *Sybil* partindo de várias fontes maliciosas, considerando uma mesma taxa de recorrência

das solicitações de identidades foram realizadas por fontes com confiança maior ou igual a $0,5$. Esse percentual decresce para 61% na Figura 5.5 (b), 56% na Figura 5.5 (c) e aproximadamente 50% na Figura 5.5 (d).

Em contrapartida, as fontes associadas aos atacantes continuam a apresentar um comportamento discrepante em relação às demais fontes. Apesar de os atacantes conseguirem algum sucesso no ataque em conluio, estes continuam a obter baixíssimos valores de confiança (logo, desafios computacionais mais complexos). Com 100 fontes, nenhuma solicitação obtém grau de confiança maior ou igual a $0,05$. Embora haja um ganho considerável no ataque para o caso em que 500 fontes são empregadas, apenas 13% das solicitações obtiveram confiança maior ou igual a $0,1$. Para o caso com 1000 fontes, 15% das solicitações obtiveram confiança maior ou igual a $0,2$, e para o caso com 2000 fontes, 35% das solicitações. Esses resultados evidenciam, ao mesmo tempo, a robustez e a eficácia da solução proposta frente a ataques *Sybil*, mesmo quando estes ocorrem em conluio. Mais importante, mostra que o atacante precisa dedicar uma gigantesca quantidade de recursos para obter sucesso no ataques, tanto em termos de *fontes* distribuídas (para despistar o esquema de diferenciação por fontes de solicitação), como em termos de capacidade computacional (para resolver os desafios propostos).

# 6 DISCUSSÕES SOBRE A SOLUÇÃO PROPOSTA

Este capítulo busca fazer uma discussão sobre a aplicação do modelo proposto em redes par-a-par reais. Primeiramente são abordadas as possibilidades de aplicação do modelo em arcabouços par-a-par e as alterações necessárias para instanciação em *overlays* estruturados e não-estruturados. A seguir é abordada a questão de como efetuar o mapeamento do grau de confiança na complexidade de diferentes tipos de desafio computacional, considerando desafios já propostos na literatura. Por fim, é realizada uma discussão sobre a materialização do conceito de *fonte*, quais as granularidades possíveis e as implicações de sua adoção.

## 6.1 Instanciação em Arcabouços Par-a-Par Reais

Conforme apresentado no Capítulo 4, a solução proposta baseia-se na exigência da resolução de desafios computacionais adaptativos antes que potenciais usuários obtenham identidades que os permitam ingressar em uma rede par-a-par. É importante frisar que a instanciação do modelo em redes existentes requereria, no geral, pequenas alterações nas entidades que compõem a rede. Naturalmente, essas alterações dependeriam diretamente do sistema par-a-par considerado e de suas características.

### 6.1.1 *Overlay* Não-estruturado

Como exemplo de um arcabouço par-a-par para troca de arquivos com *overlay* não-estruturado adotou-se o BitTorrent (COHEN, 2003) devido à sua popularidade e extensa bibliografia descrevendo seu funcionamento. Nesse sistema, cada par é identificado por uma tupla composta por (IP, Porta, Peer ID). Peer ID é um string pseudo-randômico de 20 bytes composto por duas partes. A primeira, de 6 a 8 bytes de comprimento, indica o cliente BitTorrent e a versão utilizada. Por exemplo, "M4-0-4-"indica a versão 4.0.4 do cliente BitTorrent originalmente desenvolvido por Bram Cohen, criador desse protocolo. A segunda parte é formada por um valor aleatório. Cada novo usuário que deseja participar da rede par-a-par gera um Peer ID e informa ao *tracker* que deseja ingressar na rede. O *tracker* envia como resposta uma lista contendo tuplas de pares, usualmente limitada em cinqüenta. A partir dessas tuplas podem ser recuperados os dados de IP e porta para contatar diretamente o par desejado a fim de obter um arquivo. Simplificadamente, a adesão de um novo par $P_i$ a uma rede BitTorrent pode ser descrita nos seguintes passos:

1. $P_i$ gera um identificador $ID_i$ para si próprio.

2. $P_i$ envia uma mensagem para o *tracker* $T_j$ contendo $ID_i$.

3. $T_i$ responde com uma lista de pares os quais $P_i$ pode contatar diretamente e obter o arquivo desejado.

Para que o modelo proposto possa funcionar no BitTorrent seria preciso criar duas novas mensagens a serem trocadas entre os pares e o *tracker* antes que $P_i$ pudesse obter sua identidade.

1. $P_i$ envia uma mensagem para o *tracker* $T_j$ requisitando um identificador $ID_i$.
2. $T_j$ responde com a definição de um desafio que deve ser resolvido.
3. $P_i$ resolve o desafio e submete a solução para $T_j$.
4. $T_j$ confere a correta resolução do desafio e responde com uma mensagem contendo $ID_i$ e uma lista de pares, os quais $P_i$ pode contatar diretamente e obter o arquivo desejado. Caso $P_i$ não retorne uma solução para o desafio ou retorne uma solução incorreta, $T_j$ envia uma mensagem indicando que a requisição de identidade não será atendida e não adiciona $P_i$ nem $ID_i$ em sua lista de participantes da rede.

Outros *overlays* não-estruturados poderiam adotar estratégias semelhantes para gerenciar identidades. Por exemplo as redes *FastTrack/KaZaa* utilizam o conceito de *super-pares*, que também são contatados por usuários ingressantes. Nessas redes, e em todas outras que utilizam sistemas de admissão semelhantes, poderia ser aplicado um modelo como o proposto nessa dissertação. As alterações necessárias não seriam significativas do ponto de vista da infraestrutura da rede, nem do usuário final. Na rede, essas modificações se resumiriam a criar passos intermediários para que a entidade (*tracker* ou *super-par*), que já é contatada no ingresso de um novo participante, tivesse a capacidade de gerar desafios e conferir sua correta resolução. Já no lado dos clientes seria necessário basicamente o comprometimento de tempo e recursos na resolução dos desafios propostos.

### 6.1.2 *Overlay* Estruturado

Como exemplo de *overlay* estruturado adotou-se a Content Adressable Network (CAN) (RATNASAMY et al., 2001) por ser uma das mais antigas e ainda muito utilizada, além de bastante versátil em sua finalidade. Segundo os autores, CAN pode ser utilizada não apenas como substrato para compartilhamento de arquivos, mas também para sistemas de armazenamento em rede ou serviços distribuídos de resolução de nomes.

O ingresso de um novo par $P_i$ em uma rede CAN se processa da seguinte maneira:

- $P_i$ envia uma mensagem *M* endereçada a um ponto $p_j$ qualquer no espaço lógico de coordenadas.
- O par $R_j$, responsável pela zona a qual o ponto $p_j$ faz parte recebe a mensagem de ingresso de $P_i$ e, então, divide a zona sob sua responsabilidade e repassa metade da mesma a $P_i$.

Para que a mensagem de $P_i$ chegue até $R_j$ é preciso que $P_i$ conheça pelo menos um participante da rede CAN. Ao enviar sua mensagem de ingresso a esse participante, ela será encaminhada utilizando o mecanismo de roteamento da rede até chegar a $R_j$. Os criadores da CAN assumem que a descoberta do endereço dos chamados "pares de *boostrap*" é realizada por um sistema de nomes de domínios (*Domain Name System*, DNS) como o utilizado no YOID (FRANCIS, 1999).

Para que o modelo proposto nesta dissertação possa ser utilizado em uma rede CAN, seria necessário que o serviço de descoberta de pares de *boostrap* tivesse a capacidade de propor e verificar desafios, de maneira semelhante à explicada no final da Subseção 6.1.1. Esse serviço deixaria de responder automaticamente consultas sobre o endereço de um par da rede CAN, pois exigiria a correta resolução de um desafio por parte do par requisitante. Essa solução ainda não seria completa, pois o par $P_i$ poderia já conhecer de antemão o endereço de um participante da rede. Para evitar esse problema, a mensagem *M* poderia passar a conter uma prova (por exemplo uma assinatura) de que $P_i$ resolveu o desafio exigido para ingressar na rede. Qualquer par no caminho até $R_j$ que verificasse a ausência da assinatura simplesmente descartaria a mensagem.

## 6.2 Mapeando Confiança em Desafios

Em relação ao mapeamento do grau de confiança na complexidade do desafio proposto, tal depende essencialmente da natureza do mesmo. Essa seção aborda dois desafios que demandam, respectivamente, o uso de poder de processamento e uso de memória, e exemplifica como associar o grau de confiança calculado com o custo de resolução.

### 6.2.1 Desafio *CPU-bound*

Para que um problema possa ser utilizado como um desafio computacional é necessário que ele atenda três características:

1. a verificação de sua resolução deve ser trivial;

2. não deve existir uma maneira de resolver o problema diretamente, sendo a melhor alternativa a força-bruta;

3. a principal característica exigida para sua resolução é a capacidade de processamento, em detrimento de outras características como capacidade de armazenamento ou comunicação.

Adicionalmente, como uma quarta característica é interessante (e uma exigência dos desafios que podem ser utilizados no contexto desse trabalho) que exista algum parâmetro do problema que represente sua dificuldade. Esse parâmetro deve ser tal que quando alterado torne mais extensiva a busca por uma solução satisfatória. Problemas np-completo são bons candidatos, pois costumam satisfazer as duas primeiras exigências e, não raro, obedecem também às duas últimas. Uma lista com diversos desses problemas pode ser encontrada em (GAREY; JOHNSON, 1979)).

Como exemplo concreto de desafio que demanda capacidade de processamento, destaca-se aquele apresentado no trabalho seminal de Douceur sobre identidades *Sybil* (DOUCEUR, 2002): dado um número aleatório suficientemente grande $y$, encontrar dois números $x$ e $z$ em um período de tempo limitado tal que a concatenação $x|y|z$, após processada por uma função *hash* segura, leve a um número cujos $n$ bits menos significantes sejam todos 0.

Nesse desafio a dificuldade pode ser parametrizada de duas maneiras: *(i)* aumentando a quantidade $n$ de bits que devem estar zerados ou *(ii)* aumentando o tamanho de $x$ e $z$. Uma vez que o tempo para resolver esse desafio é proporcional a $2^{n-1}$, e o tempo para verificar a resolução é constante, uma estratégia de mapeamento seria adotar uma função $f(x)$ que receberia como parâmetro o grau de confiança e retornaria um número inteiro $n$

(ou o tamanho de $x$ e $z$) que define a complexidade do desafio. Quanto maior o grau de confiança, menor seria o tamanho de $n$ (ou de $x$ e $z$) e vice-versa.

### 6.2.2 Desafio *Memory-bound*

Desafios que demandam poder de computação são os mais populares na literatura, mas seu maior problema reside no fato de que computadores com grande poder de processamento são diversas (até mesmo dezenas) de vezes mais poderosos do que computadores com baixa capacidade de processamento. Além disso, existem técnicas sofisticadas (como *pipelines*) empregadas em equipamentos mais avançados que aceleram ainda mais sua capacidade de processamento. Como resultado dessa disparidade, se a computação de uma função leva poucos segundos em um computador de alta tecnologia, pode levar um minuto em um ultrapassado tecnologicamente e diversos minutos em um PDA. Esse tempo pode representar um contratempo para usuários de equipamentos antigos e ser intolerável para usuários de PDAs. Para contornar esse problema, cientistas têm se dedicado a descobrir funções cuja resolução seja mais igualitária na maior parte dos computadores. Martin Abadi *et al* (ABADI et al., 2005) propuseram o uso de funções *memory-bound*, isto é, funções cujo tempo de resolução é dominado pelo *delay* no acesso a áreas de memória. Segundo os autores, a latência no acesso a memória entre computadores varia tipicamente em até duas vezes e raramente ultrapassa quatro vezes.

Cynthia Dwork *et al* (DWORK; GOLDBERG; NAOR, 2002) propuseram uma função *memory-bound* chamada de *MBound*. A função consiste basicamente em acessos seqüenciais e não-previsíveis a posições de uma tabela *T* suficientemente grande. Essa tabela é preenchida a partir da computação de uma função *F* recursiva. É importante que a computação da função $F^{-1}$, inversa de *F*, seja mais custosa do que um acesso a memória. Assim, é mais rápido computar $F$ para todos os valores possíveis e então acessar as posições de memória do que simplesmente computar $F^{-1}$ repetidas vezes. A tabela *T* representa uma árvore e a seqüência da formulação, resolução e conferência do resultado é apresentada abaixo:

- Seja *R* o responsável pela elaboração e verificação do desafio e *S* quem irá resolvê-lo.
- Sejam *K* e *N* dois valores inteiros e *F* uma função cujo domínio e intervalo sejam inteiros em $0..2^{N-1}$.
- *R* comunica a *S* os valores *K*, *N* e a definição de *F*, a partir da qual será montada toda a árvore.
- *R* escolhe um nó $X_k$ entre os $(2^{N-1})$ nós da árvore e computa seu caminho até a raiz da mesma. *R* envia para *S* um *checksum* aplicado sobre o caminho desde $X_k$ até $X_0$.
- *S* deve responder qual a seqüência de nós $X_1$, $X_2$ ... $X_{k-1}$ percorridos desde a raiz $X_0$ da árvore até chegar em $X_k$.
- A verificação por parte de *R* é trivial, bastando apenas comparar se a seqüência de nós percorridos por *S* realmente leva de $X_0$ até $X_k$.

No desafio acima, a dificuldade pode ser parametrizada por meio do dimensionamento dos valores *K* e *N*, respectivamente a profundidade da árvore e quantos nós ela possui.

Quanto maior for *K*, mais acessos a memória serão necessários para que *S* compute a solução, pois mais passos existirão entre $X_0$ e $X_k$. De forma análoga, o aumento de *N* torna a árvore maior, pois existem mais nós na mesma e isso também aumenta a quantidade de acessos necessários para resolver o problema.

O mapeamento do grau de confiança, calculado de acordo com a Equação 4.3, na dificuldade de um desafio de memória poderia ser feito estabelecendo valores mínimos e máximos para a quantidade de pares na árvore. Quanto maior for o grau de confiança, menos nós e, conseqüentemente, mais rápida tenderia a ser a resolução do desafio. Por outro lado, quando a confiança se aproximasse de 0, a árvore teria tamanho e, portanto, custo máximo para resolução de um desafio.

## 6.3 Materializando a Noção de Fonte

Sobre a materialização da noção de fonte, uma estratégia é considerar um endereço IP, uma sub-rede ou um sistema autônomo como uma fonte distinta. Outra estratégia seria o uso de sistemas de coordenadas de rede, por exemplo o Vivaldi (DABEK et al., 2004), ou Veracity (SHERR; BLAZE; LOO, 2009) para distinguir solicitações vindas de determinadas regiões, cidades, estados, ou mesmo países.

Independemente da noção utilizada, a granularidade da fonte considerada tem algumas implicações no funcionamento do modelo proposto. A primeira possibilidade a considerar é o caso onde cada fonte identifica unicamente um par. Essa abordagem parece interessante do ponto de vista do modelo proposto, pois todas as medições seriam feitas em função do comportamento de um único par. Cada um seria avaliado exclusivamente a partir de sua recorrência, sem que fosse possível que o comportamento de outros pares interferisse no cálculo de seu $\phi$ e, em última análise, na dificuldade dos desafios que deveriam ser resolvidos por si. No entanto, existem alguns obstáculos para que possa ser considerada uma fonte com uma granularidade tão fina. O primeiro problema está em definir qual informação pode ser utilizada para identificar cada fonte. A opção mais natural seria considerar o endereço IP, porém com tal opção surge o problema de como tratar pares posicionados atrás de um mesmo NAT. Externamente é visível apenas um mesmo IP, porém são pares distintos. Outro problema que surge quando se considera uma granularidade tão fina é a relativa facilidade com que atacantes podem subverter o sistema. Usuários maliciosos poderiam obter IPs próximos ao seu e que estivessem sem uso através de técnicas de *IP-harvesting* (RAMACHANDRAN; FEAMSTER, 2006) e, assim, simular diversas fontes, o que lhes permitiria alterar de maneira mais efetiva o comportamento padrão da rede.

Aumentando a granularidade da fonte, uma possibilidade seria considerar todos endereços IP de uma sub-rede ou região (talvez até mesmo algumas regiões) de um sistema de coordenadas. Essa abordagem apresenta a desvantagem de o comportamento de um par ser influenciado por pares próximos de si. Caso uma parcela significativa dos pares associados à mesma fonte passem a se comportar como atacantes, então os honestos serão prejudicados injustamente. Como principal vantagem está a maior aplicabilidade prática, pois não existindo a restrição de identificar unicamente cada par, NAT deixa de ser um problema. Se a granularidade for pequena, as desvantagens de se medir o comportamento de alguns par e não de cada um individualmente diminuem, o que torna essa abordagem a que potencialmente apresenta melhor custo-benefício para aplicação prática.

No extremo de maior granularidade, pode ser considerada como fonte uma larga faixa de IPs, ou então estados e até mesmo países em um sistema de coordenadas. Essa aborda-

gem não parece adequada pois apresenta uma desvantagem considerável. Quanto maior for a quantidade de pares associados a cada fonte, mais o comportamento de cada um se dilui no todo. A fonte representa a unidade mínima tratada no modelo proposto, assim não é possível identificar potenciais atacantes dentro de uma fonte. Um usuário desonesto que esteja associado a uma fonte composta majoritariamente por bem comportados acabaria sendo tratado como honesto, visto ser esse o comportamento comum dos pares dessa fonte e observado externamente.

# 7 CONSIDERAÇÕES FINAIS

Redes par-a-par são responsáveis por grande parte do tráfego de dados da Internet na atualidade. São numerosas as aplicações e finalidades dessa arquitetura de rede que beneficiam tanto usuários domésticos quanto dos setores acadêmico e empresarial. Essas redes, no entanto, estão sujeitas a uma ampla gama de ataques que impedem seu correto funcionamento e aproveitamento por parte dos usuários interessados. Um dos principais ataques, que ocorre devido a fragilidades nos mecanismos de autenticação e gerenciamento de identidades, é conhecido por *Sybil*. Ele consiste na criação indiscriminada de identidades fantasma por uma mesma entidade e pode servir como base para o lançamento de ataques mais elaborados.

Diversas soluções têm sido propostas na literatura para impedir, ou ao menos diminuir, o problema da criação de identidades falsas que pode resultar em um ataque *Sybil*. O emprego de desafios computacionais para obtenção de identidades é uma dessas alternativas que tem se mostrado promissoras para combater a ocorrência desse ataque em redes par-a-par. Essa abordagem consiste em exigir a correta resolução de um desafio computacional antes da obtenção de uma identidade por parte de um usuário interessado em participar da rede. Dessa forma, os usuários honestos demonstram, através do comprometimento de parte de seus recursos, suas boas intenções com a rede. Por outro lado, os atacantes têm sua capacidade de requisição de identidades limitada.

Estudos foram conduzidos ((BORISOV, 2006; CASTRO et al., 2002; ROWAIHY et al., 2007)) para comprovar a eficácia do uso de desafios para a contenção de *Sybils* em redes par-a-par. Apesar dos bons resultados obtidos, o custo dos desafios empregados são determinados de maneira estática, em outras palavras, a dificuldade dos desafios propostos são definidos sem levar em consideração o comportamento da rede, dos pares, ou sua capacidade para resolução dos desafios. Ainda que a dificuldade dos desafios possa ser variada, a falta de mecanismos que permitam lidar adequadamente com situações em que existe significativa disparidade de poder computacional entre usuários legítimos e atacantes, consiste na principal crítica a essa abordagem e impede o seu uso mais efetivo e disseminado.

A presente dissertação tratou essa limitação ao definir um modelo de desafios adaptativo como limitante à disseminação de *Sybils*, que considera o comportamento dos pares de uma rede para determinação do custo de um desafio. Propôs-se o uso da recorrência como indicativo de comportamento suspeito, pois um usuário malicioso que esteja prestes a efetuar um ataque *Sybil* à rede, deve criar diversas identidades e esse fato eleva sua taxa de recorrência. O modelo proposto calcula a *taxa de recorrência de cada fonte* ($\phi$) e a compara com a *taxa de recorrência da rede* ($\Phi$), ambas considerando a dinâmica dos usuários. Dessa maneira, busca-se identificar comportamentos que se distanciam do padrão observado na rede e se assemelham ao de atacantes. A partir desses valores estabelece-se

um grau de confiança no comportamento da fonte e atribui-se um desafio com um custo computacional de acordo com o quão suspeito é o comportamento observado de cada par.

Os experimentos realizados, embora não exaustivos, evidenciaram a capacidade da solução proposta em diminuir a capacidade dos atacantes de criarem identidades falsas de forma indiscriminada, ao mesmo tempo sendo favorável a usuários legítimos, os quais foram, em geral, penalizados minimamente. Ao calcular valores de confiança menores a fontes com taxas de solicitação de identidade mais altas, os usuários (maliciosos) atrelados a essas fontes tiveram de arcar com desafios computacionais de maior complexidade. Por outro lado, os usuários associados a fontes presumidamente legítimas (e que recorreram menos vezes para solicitar identidades) receberam desafios computacionais menos complexos (dados os maiores valores de grau de confiança que as fontes em questão possuiam perante a rede par-a-par).

Uma preocupação constante na elaboração deste trabalho foi a de tornar viável sua utilização em arcabouços par-a-par reais. Para isso foi feita uma reflexão sobre a instanciação do modelo proposto em redes par-a-par. Essa reflexão buscou associar as entidades e os procedimentos necessários para o funcionamento do modelo em entidades e procedimentos já presentes tanto em *overlays* estruturados quanto em não-estruturados. Outra evidência da preocupação com a viabilidade prática desse trabalho foi a utilização de traços obtidos em comunidades par-a-par atualmente em funcionamento, o que confere maior legitimidade aos bons resultados obtidos.

Como trabalhos futuros, visualiza-se hoje três possíveis continuações. Primeiramente, pretende-se estender a solução proposta para capturar o comportamento das fontes face ao atraso associado à resolução dos desafios, posto que hoje tal não é considerado por questões de simplicidade. Isso permitiria a análise do modelo proposto de maneira mais completa, visto que durante a resolução do desafio um atacante terá sua capacidade de requisição de identidades diminuída, pois seus recursos estarão sendo empregados na resolução do mesmo.

Outro desdobramento possível é a investigação de um mecanismo que apóie a valoração mais adequada dos parâmetros da solução proposta considerando comunidades com características distintas. Busca-se criar heurísticas que permitiriam estabelecer os valores para os parâmetros utilizados no cálculo do grau de confiança (parâmetros a, b, c, a duração e o passo da janela, além do $\beta$), de acordo com a comunidade par-a-par em que o modelo estiver sendo aplicado.

Por fim ainda pretende-se estudar a aplicabilidade de desafios de natureza diferentes, de maneira concomitante. A idéia é criar relações de equivalência entre desafios que demandem capacidades diferentes (processamento, uso de memória, ou mesmo CAPTCHAs (Completely Automated Turing-test to tell Computer and Humans Apart, (AHN et al., 2003)) de forma a tornar o modelo adaptativo não apenas na dificuldade, mas também no tipo de desafios empregados.

# REFERÊNCIAS

ABADI, M.; BURROWS, M.; MANASSE, M.; WOBBER, T. Moderately hard, memory-bound functions. **ACM Transactions Inter. Tech.**, New York, NY, USA, v.5, n.2, p.299–327, May 2005.

ABERER, K.; DATTA, A.; HAUSWIRTH, M. A decentralized public key infrastructure for customer-to customer e-commerce. In: INTERNATIONAL JOURNAL OF BUSINESS PROCESS INTEGRATION AND MANAGEMENT, 2005. **Anais...** [S.l.: s.n.], 2005. p.26–33.

AHN, L. von; BLUM, M.; HOPPER, N.; LANGFORD, J. CAPTCHA: using hard ai problems for security. In: . [S.l.: s.n.], 2003. p.646.

ANDRADE, N.; BRASILEIRO, F.; CIRNE, W.; MOWBRAY, M. Discouraging Free Riding in a Peer-to-Peer CPU-sharing Grid. In: IEEE SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING (HPDC'04), 13., 2004. **Anais...** [S.l.: s.n.], 2004.

AURA, T.; NIKANDER, P.; LEIWO, J. DOS-Resistant Authentication with Client Puzzles. In: REVISED PAPERS FROM THE 8TH INTERNATIONAL WORKSHOP ON SECURITY PROTOCOLS, 2001, London, UK. **Anais...** Springer-Verlag, 2001. p.170–177.

BARCELLOS, M. P.; GASPARY, L. P. Fundamentos, Tecnologias e Tendências rumo a Redes P2P. **SBC Jornadas de Atualização em Informática**, [S.l.], 2006.

BARFORD, P.; YEGNESWARAN, V. **An Inside Look at Botnets**. [S.l.]: Springer-Verlag, 2007. 171–191p.

BITSOUP.ORG. **Bitsoup.org – The Number One Site for your Torrent Appetite**. Disponível em: http://www.bitsoup.org/. Acesso em jan. 2010.

Bittorrent. **BitTorrent website**. Disponível em: http://www.bittorrent.com/. Acesso em jan. 2010.

BORISOV, N. Computational Puzzles as Sybil Defenses. In: IEEE INTERNATIONAL CONFERENCE ON PEER-TO-PEER COMPUTING (P2P 2006), 6., 2006. **Anais...** [S.l.: s.n.], 2006. p.171–176.

CASTRO, M.; DRUSCHEL, P.; KERMARREC, A.; ROWSTRON, A. SCRIBE: a large-scale and decentralized application-level multicast infrastructure. **IEEE Journal on Selected Areas in communications (JSAC)**, [S.l.], v.20, n.8, p.1489–1499, 2002.

CASTRO, M.; DRUSHEL, P.; GANESH, A.; ROWSTRON, A.; WALLACH, D. S. Secure Routing for Structured Peer-to-Peer Overlay Networks. In: USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI 2002), 5., 2002. **Anais...** [S.l.: s.n.], 2002. p.299–314.

ChimeraTapestry. **Chimera and Tapestry website**. Disponível em: http://p2p.cs.ucsb.edu/chimera/. Acesso em jan. 2010.

Chord. **The Chord Project**. Disponível em: http://pdos.csail.mit.edu/chord/. Acesso em jan. 2010.

COHEN, B. **Incentives Build Robustness in BitTorrent**. 2003.

COOKE, E.; JAHANIAN, F.; MCPHERSON, D. The Zombie Roundup: understanding, detecting, and disrupting botnets. In: 2005. **Anais...** [S.l.: s.n.], 2005. p.39–44.

DABEK, F.; BRUNSKILL, E.; KAASHOEK, F. M.; KARGER, D.; MORRIS, R.; STOICA, I.; BALAKRISHNAN, H. Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service. In: WORKSHOP ON HOT TOPICS IN OPERATING SYSTEMS (HOTOS), 8., 2001. **Anais...** [S.l.: s.n.], 2001. p.81–86.

DABEK, F.; COX, R.; KAASHOEK, F.; MORRIS, R. Vivaldi: a decentralized network coordinate system. **SIGCOMM Comput. Commun. Rev.**, New York, NY, USA, v.34, n.4, p.15–26, October 2004.

DAMIANI, E.; VIMERCATI, D. C. di; PARABOSCHI, S.; SAMARATI, P.; VIOLANTE, F. A Reputation-based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In: CCS '02: PROCEEDINGS OF THE 9TH ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 2002, New York, NY, USA. **Anais...** ACM Press, 2002. p.207–216.

DANEZIS, G.; LESNIEWSKI-LAAS, C.; KAASHOEK, F. M.; ANDERSON, R. Sybil-Resistant DHT Routing. In: . [S.l.: s.n.], 2005. p.305–318.

DATTA, A.; HAUSWIRTH, M.; ABERER, K. Beyond "web of trust": Enabling P2P E-commerce. In: IEEE INTERNATIONAL CONFERENCE ON E-COMMERCE TECHNOLOGY (CEC'03), 2003. **Anais...** [S.l.: s.n.], 2003.

DOUCEUR, J. R. The Sybil Attack. In: INTERNATIONAL WORKSHOP ON PEER-TO-PEER SYSTEMS (IPTPS 2002), 1., 2002. **Anais...** [S.l.: s.n.], 2002. p.251–260.

DRUSCHEL, P.; ROWSTRON, A. PAST: a large-scale, persistent peer-to-peer storage utility. **8th IEEE Workshop on Hot Topics in Operating Systems (HotOS 2001)**, [S.l.], p.75, 2001.

DWORK, C.; GOLDBERG, A.; NAOR, M. On Memory-Bound Functions for Fighting Spam. In: IN CRYPTO, 2002. **Anais...** Springer-Verlag, 2002. p.426–444.

eBay. **eBay website**. Disponível em: http://www.ebay.com/. Acesso em jan. 2010.

ELLISON, C. **SPKI Requirements**. United States: RFC Editor, 1999.

ELLISON, C.; FRANTZ, B.; LAMPSON, B.; RIVEST, R.; THOMAS, B.; YLONEN, T. **SPKI Certificate Theory**. United States: RFC Editor, 1999.

ENGLE, M. **Vulnerabilities of P2P Systems and a Critical look at Their Solutions**. [S.l.]: Kent State University, 2006.

ESM. **End System Multicast website**. Disponível em: http://esm.cs.cmu.edu/. Acesso em jan. 2010.

FELDMAN, M.; PAPADIMITRIOU, C.; CHUANG, J.; STOICA, I. Free-riding and whitewashing in peer-to-peer systems. **Selected Areas in Communications, IEEE Journal on**, [S.l.], v.24, n.5, p.1010–1019, May 2006.

FRANCIS, P. Yoid : extending the multicast internet architecture. In: SPRINGER-VERLAG, 1999. **Anais...** [S.l.: s.n.], 1999. White paper, http://www.aciri.org/yoid.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability**: a guide to the theory of np-completeness. [S.l.]: W. H. Freeman, 1979.

Genome. **Genome@home website**. Disponível em: http://genomeathome.stanford.edu/. Acesso em jan. 2010.

GKANTSIDIS, C.; MIHAIL, M.; SABERI, A. Random walks in peer-to-peer networks. In: ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES (INFOCOM 2004), 23., 2004. **Anais...** [S.l.: s.n.], 2004. v.1, p.120–130.

Gnutella. **Gnutella website**. Disponível em: http://rfc-gnutella.sourceforge.net/. Acesso em jan. 2010.

Google. **Google Talk**. Disponível em: http://www.google.com/talk/. Acesso em jan. 2010.

GRIZZARD, J. B.; JOHNS, T. Peer-to-Peer Botnets: overview and case study. In: IN USENIX WORKSHOP ON HOT TOPICS IN UNDERSTANDING BOTNETS (HOTBOTS07, 2007. **Anais...** [S.l.: s.n.], 2007.

GRöNVALL, B.; MARSH, I.; PINK, S. A Multicast-based Distributed File System for the Internet. In: IN OPERATING SYSTEMS DESIGN AND IMPLEMENTATION, 1996. **Anais...** [S.l.: s.n.], 1996. p.251–264.

HARDING, G. The tragedy of the commons. **Science**, [S.l.], v.162, p.1243–1248, 1968.

ICQ. **ICQ.com website**. Disponível em: http://www.icq.com/. Acesso em jan. 2010.

Jabber. **Jabber**: open Instant Messaging. Disponível em: http://www.jabber.org/. Acesso em jan. 2010.

Joost. **Joost.com website**. Disponível em: http://www.joost.com/. Acesso em jan. 2010.

JOSANG, A.; ISMAIL, R.; BOYD, C. A survey of trust and reputation systems for online service provision. **Decision Support Systems**, Amsterdam, The Netherlands, The Netherlands, v.43, n.2, p.618–644, March 2007.

JUN, S.; AHAMAD, M. Incentives in BitTorrent Induce Free Riding. In: ACM SIGCOMM WORKSHOP ON ECONOMICS OF PEER-TO-PEER SYSTEMS (P2PECON), 2005. **Anais...** [S.l.: s.n.], 2005. p.116–121.

Justin.tv. **Justin.tv website**. Disponível em: http://www.justin.tv/. Acesso em jan. 2010.

KAMVAR, S. D.; SCHLOSSER, M. T.; GARCIA-MOLINA, H. The Eigentrust algorithm for reputation management in P2P networks. In: WORLD WIDE WEB (WWW '03), 12., 2003, New York, NY, USA. **Proceedings...** ACM Press, 2003. p.640–651.

KAUFMAN, C.; PERLMAN, R.; SPECINER, M. **Network Security**: private communication in a public world. 2nd.ed. [S.l.]: Prentice Hall, 2002.

KULBAK, Y.; BICKSON, D.; KULBAK, Y.; PROF, A.; KIRKPATRICK, S.; YORAM KULBAK, C. (c; BICKSON, D.; BICKSON, D.; LAB, D. **The eMule Protocol Specification**. [S.l.: s.n.], 2005.

LAWTON, G. Is Peer-to-Peer Secure Enough for Corporate Use? **IEEE Computer**, [S.l.], v.37, n.1, p.22–25, 2004.

LIANG, J.; KUMAR, R. Pollution in P2P File Sharing Systems. In: INFOCOM 2005. 24TH IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS. PROCEEDINGS, 2005. **Anais...** [S.l.: s.n.], 2005. p.1174–1185.

LIANG, J.; KUMAR, R.; ROSS, K. W. The KaZaA Overlay: a measurement study. In: IEEE ANNUAL COMPUTER COMMUNICATIONS WORKSHOP (CCW 2004), 19., 2004. **Anais...** [S.l.: s.n.], 2004.

LIANG, J.; NAOUMOV, N.; ROSS, K. W. The Index Poisoning Attack in P2P File Sharing Systems. In: INFOCOM 2006. 25TH IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS. PROCEEDINGS, 2006. **Anais...** [S.l.: s.n.], 2006. p.1–12.

LOCHER, T.; MOOR, P.; SCHMID, S.; WATTENHOFER, R. Free Riding in BitTorrent is Cheap. In: WORKSHOP ON HOT TOPICS IN NETWORKS (HOTNETS-V), 5., 2006. **Anais...** [S.l.: s.n.], 2006.

LUA, E. K.; CROWCROFT, J.; PIAS, M.; SHARMA, R.; LIM, S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. **IEEE Communications Surveys & Tutorials**, [S.l.], v.7, n.2, p.72–93, 2005.

MARTI, S.; GARCIA-MOLINA, H. Taxonomy of trust: categorizing p2p reputation systems. **Computer Networks**, [S.l.], v.50, n.4, p.472–484, March 2006.

MAYMOUNKOV, P.; MAZIERES, D. Kademlia: a peer-to-peer information system based on the xor metric. In: INTERNATIONAL PEER-TO-PEER SYMPOSIUM (IPTPS02), 2002. **Anais...** [S.l.: s.n.], 2002.

MEDIADEFENDER. **Media Defender Inc.** Disponível em: http://www.mediadefender.com. Acesso em jan. 2010.

MORSELLI, R.; BHATTACHARJEE, B.; KATZ, J.; MARSH, M. A. **KeyChains**: a decentralized public-key infrastructure. [S.l.]: Digital Repository at the University of Maryland [http://drum.umd.edu/oai] (United States), 2006.

MSN. **MSN Messenger**. Disponível em: http://www.msn.com/. Acesso em jan. 2010.

MUTHITACHAROEN, A.; MORRIS, R.; GIL, T. M.; CHEN, B. Ivy: a read/write peer-to-peer file system. In: 2002. **Anais...** [S.l.: s.n.], 2002. p.31–44.

Napster. **Napster**. Disponível em: http://free.napster.com/. Acesso em jan. 2010.

OceanStore. **The OceanStore Project website**. Disponível em: http://oceanstore.cs.berkeley.edu/. Acesso em jan. 2010.

OpenNap. **OpenNap**: Open Source Napster Server website. Disponível em: http://opennap.sourceforge.net/. Acesso em jan. 2010.

PEERMEDIA. **Peer Media Technologies**. Disponível em: http://www.peermediatech.com. Acesso em jan. 2010.

PLAXTON, G. C.; RAJARAMAN, R.; RICHA, A. W. Accessing nearby copies of replicated objects in a distributed environment. In: SPAA '97: PROCEEDINGS OF THE NINTH ANNUAL ACM SYMPOSIUM ON PARALLEL ALGORITHMS AND ARCHITECTURES, 1997, New York, NY, USA. **Anais...** ACM Press, 1997. p.311–320.

PONTES, F.; BRASILEIRO, F.; ANDRADE, N. Sobre Calotes e Múltiplas Personalidades no BitTorrent. **25o Simpósio Brasileiro de Redes de Computadores**, Campina Grande, PB, Brasil, 2007.

RAMACHANDRAN, A.; FEAMSTER, N. Understanding the network-level behavior of spammers. In: APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS (SIGCOMM 2006), 2006., 2006, New York, NY, USA. **Proceedings...** ACM Press, 2006. p.291–302.

RATNASAMY, S.; FRANCIS, P.; HANDLEY, M.; KARP, R.; SCHENKER, S. A scalable content-addressable network. In: ACM SIGCOMM CONFERENCE ON APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS (SIGCOMM '01), 2001., 2001, New York, NY, USA. **Proceedings...** ACM, 2001. v.31, n.4, p.161–172.

ROBERTS, L. G. A Radical New Router. **IEEE Spectrum**, [S.l.], v.46, n.7, p.30–35, 2009.

ROWAIHY, H.; ENCK, W.; MCDANIEL, P.; LA PORTA, T. Limiting Sybil Attacks in Structured P2P Networks. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS (INFOCOM 2007), 26., 2007, Anchorage, Alaska , USA. **Anais...** [S.l.: s.n.], 2007. p.2596–2600.

ROWSTRON, A.; DRUSCHEL, P. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In: ACM SYMPOSIUM ON OPERATING SYSTEM PRINCIPLES (SOSP'01), 18., 2001. **Anais...** [S.l.: s.n.], 2001. p.188–201.

ROWSTRON, A.; DRUSCHEL, P. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. **Lecture Notes in Computer Science**, [S.l.], v.2218, p.329–350, 2001.

ROWSTRON, A. I. T.; DRUSCHEL, P. Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: MIDDLEWARE '01: PROCEEDINGS OF THE IFIP/ACM INTERNATIONAL CONFERENCE ON DISTRIBUTED SYSTEMS PLATFORMS HEIDELBERG, 2001, London, UK. **Anais...** Springer-Verlag, 2001. p.329–350.

RUITENBEEK, E. V.; SANDERS, W. H. Modeling Peer-to-Peer Botnets. In: QEST '08: PROCEEDINGS OF THE 2008 FIFTH INTERNATIONAL CONFERENCE ON QUANTITATIVE EVALUATION OF SYSTEMS, 2008, Washington, DC, USA. **Anais...** IEEE Computer Society, 2008. p.307–316.

SANTOS, F. R.; GASPARY, L. P.; BARCELLOS, M. P. Separando Joio de Trigo com Funnel: combate à poluição de conteúdo em comunidades bittorrent. In: SIMPóSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUíDOS (SBRC 2009), 27., 2009, Recife, PE, Brazil. **Anais...** [S.l.: s.n.], 2009. p.393–406.

SCHULZE, H.; MOCHALSKI, K. **Internet Study 2007**. Disponível em: http://www.ipoque.com/userfiles/file/ internet_study_2007.pdf. Acesso em jan. 2010.

SCHULZE, H.; MOCHALSKI, K. **Internet Study 2008-2009**. Disponível em: https://portal.ipoque.com/downloads/index/get/id/265/. Acesso em jan. 2010.

SETI. **SETI@home website**. Disponível em: http://setiathome.ssl.berkeley.edu/. Acesso em jan. 2010.

SHERR, M.; BLAZE, M.; LOO, B. T. Veracity: practical secure network coordinates via vote-based agreements. In: USENIX Annual Technical Conference (USENIX '09), 2009. **Anais...** [S.l.: s.n.], 2009.

SINGH, A.; NGAN, T.-W.; DRUSCHEL, P.; WALLACH, D. S. Eclipse Attacks on Overlay Networks: threats and defenses. In: CONFERENCE ON COMPUTER COMMUNICATIONS (INFOCOM 2006), 25., 2006, Barcelona, Catalunya, Spain. **Anais...** [S.l.: s.n.], 2006. p.1–12.

Skype. **Skype**. Disponível em: http://www.skype.com/. Acesso em jan. 2010.

SONG, S.; HWANG, K.; ZHOU, R.; KWOK, Y. K. Trusted P2P Transactions with Fuzzy Reputation Aggregation. **Internet Computing, IEEE**, [S.l.], v.9, n.6, p.24–34, 2005.

SRIVATSA, M.; XIONG, L.; LIU, L. TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. In: WWW '05: PROCEEDINGS OF THE 14TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2005, New York, NY, USA. **Anais...** ACM Press, 2005. p.422–431.

STOICA, I.; MORRIS, R.; KARGER, D.; KAASHOEK, M. F.; BALAKRISHNAN, H. Chord: a scalable peer-to-peer lookup service for internet applications. In: SIGCOMM '01: PROCEEDINGS OF THE 2001 CONFERENCE ON APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS, 2001, New York, NY, USA. **Anais...** ACM, 2001. v.31, n.4, p.149–160.

STOICA, I.; MORRIS, R.; LIBEN-NOWELL, D.; KARGER, D. R.; KAASHOEK, F. M.; DABEK, F.; BALAKRISHNAN, H. Chord: a scalable peer-to-peer lookup protocol for internet applications. **IEEE/ACM Trans. Netw.**, [S.l.], v.11, n.1, p.17–32, February 2003.

THEOTOKIS, S. A.; SPINELLIS, D. A Survey of Peer-to-Peer Content Distribution Technologies. **ACM Computing Surveys**, [S.l.], v.36, n.4, p.335–371, 2004.

TSOUMAKOS, D.; ROUSSOPOULOS, N. A Comparison of Peer-to-Peer Search Methods. In: INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES (WEBDB 2003), 6., 2003. **Anais...** [S.l.: s.n.], 2003.

XIONG, L.; LIU, L. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. **IEEE Transactions on Knowledge and Data Engineering**, [S.l.], v.16, n.7, p.843–857, 2004.

Yahoo. **Yahoo! Messenger**. Disponível em: http://messenger.yahoo.com/. Acesso em jan. 2010.

YU, H.; KAMINSKY, M.; GIBBONS, P. B.; FLAXMAN, A. SybilGuard: defending against sybil attacks via social networks. In: SIGCOMM '06: PROCEEDINGS OF THE 2006 CONFERENCE ON APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS, 2006, New York, NY, USA. **Anais...** ACM Press, 2006. p.267–278.

ZHAO, B. Y.; KUBIATOWICZ, J. D.; JOSEPH, A. D. **Tapestry**: an infrastructure for fault-tolerant wide-area location and routing. [S.l.]: UC Berkeley, 2001. (UCB/CSD-01-1141).

ZIMMERMANN, P. **PGP source code and internals**. Cambridge, MA, USA: MIT Press, 1995.