Segurança Verde: Usando Desafios Adaptativos com Espera Variável para Conter *Sybils* em Redes Par-a-Par

Weverton Luis da Costa Cordeiro, Paolo Cemim, Flávio Roberto Santos, Marinho Pilla Barcellos, Luciano Paschoal Gaspary

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{weverton.cordeiro, pcemim, flavio.santos,
marinho, paschoal}@inf.ufrgs.br

Abstract. *Requiring the resolution of computacional puzzles prior to granting identities has shown to be a promising approach to tackle* Sybil *attacks. Although effective, mechanisms based on such an approach lead to an increased power consumption of the entire system (due to puzzle resolution). To address this issue, in this paper we propose combining puzzles with "varying wait time". The goal is to reduce the average complexity of assigned puzzles, without compromising the ability of slowing down the assignment of identities to attackers. The results achieved evidence the effectiveness of the proposed approach in mitigating* sybils*, which comes at significantly lower power costs to participants.*

Resumo. *Conceder identidades mediante a resolução de desafios computacionais tem se mostrado uma abordagem promissora para frear ataques* Sybil*. Apesar de efetivos, os mecanismos baseados nessa abordagem causam um elevado consumo energético no sistema como um todo (devido à resolução dos desafios). Para lidar com essa questão, neste artigo propõe-se combinar o uso de desafios com "espera variável". O objetivo é reduzir a complexidade média dos desafios atribuídos, sem abdicar da capacidade de frear a concessão de identidades aos atacantes. Os resultados obtidos evidenciam a efetividade da abordagem proposta na contenção de* sybils*, a qual ocorre a um custo energético significativamente menor aos participantes.*

1. Introdução

A criação indiscriminada de identidades falsas (ou ataque *Sybil* [Douceur 2002]), embora rudimentar, persiste como um dos ataques mais nocivos à segurança de sistemas distribuídos [Yang et al. 2011]. A motivação para um usuário lançar mão desse ataque é obter a maioria, ou pelo menos uma parte significativa, das identidades no sistema. Desse modo, um atacante poderá obter vantagens indevidas, bem como aumentar seu poder e influência sobre o sistema. Em redes par-a-par (P2P), o poder corruptivo do uso de identidades falsas (*sybils*) é bastante conhecido, tendo sido objeto de diversos estudos [Jetter et al. 2010]. A subversão dos algoritmos de reputação baseados em votação, a manipulação de mensagens de controle trocadas entre os pares e o lançamento de outros ataques como *Eclipse* [Singh et al. 2006] e *Free-riding* [Feldman et al. 2006] são exemplos de ações que um atacante – de posse dessas identidades – pode realizar na rede P2P.

A atribuição (ou renovação) de identidades aos usuários mediante a resolução de desafios computacionais tem se mostrado uma abordagem promissora para mitigar a ocorrência de ataques *Sybil* [Borisov 2006]. Essa abordagem baseia-se na premissa de que os recursos computacionais à disposição do usuário (ciclos de processador, memória, armazenamento, etc.) são finitos. Portanto, ao exigir que os usuários provem possuir uma

determinada fração desses recursos para cada identidade solicitada, limita-se a quantidade de *sybils* que um atacante poderá criar. As principais propostas nessa área focam na capacidade computacional dos usuários e utilizam desafios criptográficos de complexidade fixa para limitar a disseminação de *sybils* na rede [Borisov 2006, Rowaihy et al. 2007].

Apesar das potencialidades, as propostas baseadas em desafios não fazem distinção entre solicitações de identidades oriundas de usuários legítimos e de eventuais atacantes. Logo, ambos estão sujeitos ao pagamento do mesmo preço (computacional) por cada identidade solicitada. Idealmente, as solicitações de atacantes receberiam desafios muito mais caros que as de usuários legítimos. Assim, para uma capacidade computacional arbitrária de posse do atacante, o número de identidades que o mesmo poderia obter seria muito menor. O desafio para essa abordagem é, portanto, "separar o joio do trigo". Essa limitação foi abordada em trabalhos anteriores [Mauch et al. 2010, Cordeiro et al. 2011], com a proposta de "desafios adaptativos" – um mecanismo que define a complexidade dos desafios a serem resolvidos em função da freqüência com que os usuários (fontes de solicitação) solicitam novas identidades.

Embora efetivos, os mecanismos dessa classe demandam uma larga fração de recursos computacionais, o que em última instância resulta em um elevado consumo energético do sistema. Esse problema é agravado com o aumento de popularidade da rede P2P (e o interesse de atacantes em criar *sybils* na mesma). No contexto atual de crescente preocupação com o uso racional dos recursos naturais disponíveis, a busca por "desafios verdes" (que consumam menos recursos, mas sem abrir mão da capacidade de limitar *sybils*) torna-se de fundamental importância. Para lidar com esse problema, neste artigo propõe-se agregar uma "espera variável" ao uso de desafios. O objetivo é reduzir a complexidade média dos desafios atribuídos, sem abrir mão da capacidade de retardar a concessão de identidades a atacantes. Quanto maior a frequência de solicitação de uma determinada fonte, maior a complexidade do desafio computacional a ser resolvido – e o tempo de espera a ser obedecido – pelos usuários daquela fonte. Para aferir a eficácia do mecanismo proposto, foram realizadas avaliações por meio de simulações e de experimentos no PlanetLab. Os resultados obtidos evidenciam a potencial economia de energia proporcionada pelo uso de espera variável, cujo ganho ocorre sem prejuízo da efetividade na contenção de *sybils* (em relação aos desafios adaptativos originais).

O restante do artigo está organizado como segue. A Seção 2 discute os principais trabalhos relacionados. O mecanismo para gerenciamento de identidades baseado em desafios adaptativos com espera variável é apresentado na Seção 3, enquanto que os resultados alcançados são descritos na Seção 4. Por fim, a Seção 5 conclui o artigo com as considerações finais e possíveis desdobramentos para pesquisas futuras.

2. Trabalhos Relacionados

O interesse em estabelecer identidades em sistemas *on-line* e combater a existência de identidades falsas não é novo [Fiege et al. 1987, Ellison 1996]. Apesar dos esforços feitos, não há solução na literatura que permita a uma entidade local verificar, em um sistema distribuído e sem conhecimento físico e/ou direto das entidades remotas com as quais se relaciona, se diferentes identidades de fato pertencem a entidades distintas.

Em 2002, Douceur [Douceur 2002] sugeriu o termo *sybil* para se referir às identidades falsas criadas por uma entidade maliciosa. O autor mostrou ainda que, na ausência de uma entidade certificadora central, uma entidade desconhecida sempre pode apresentar mais de uma identidade, exceto sob condições e premissas inviáveis para sistemas distribuídos de larga escala. Desde então, as pesquisas na área tem focado no combate à disseminação de *sybils* e na mitigação do efeito nocivo da existência dos mesmos.

De acordo com o mecanismo empregado para garantir a autenticidade dos pares, essas pesquisas podem ser categorizadas entre *descentralizadas* (ou baseadas em *identidades fracas*) e *centralizadas* (ou baseadas em *identidades fortes*) [Danezis and Mittal 2009].

A primeira categoria compreende soluções nas quais o usuário tem autonomia para criar suas próprias identidades na rede P2P. O principal objetivo das investigações que se encaixam nessa categoria é limitar a quantidade de *sybils* existentes a um "nível aceitável". Exemplos de soluções nessa categoria incluem SybilInfer [Danezis and Mittal 2009] e GateKeeper [Tran et al. 2011]. A principal característica em comum às mesmas é o uso de redes sociais para limitar o número de *sybils* na rede. Além do problema de violação à anonimidade, Yang *et al.* [Yang et al. 2011] mostraram recentemente que *sybils* não tendem a formar uma malha densa de relações entre si, invalidando uma das premissas fundamentais sobre a qual essas soluções se apoiam.

Na segunda categoria de soluções, os usuários obtêm identidades via *entidades certificadoras*. Embora virtualmente elimine a ocorrência de *sybils*, tais mecanismos forçam usuários a confiar em entidades certificadoras muitas vezes desconhecidas e dificultam o acesso de potenciais usuários (por exemplo, quando os mesmos precisam fornecer dados pessoais e/ou pagar taxas para obter identidades). As soluções que se encaixam nessa categoria [Aberer et al. 2005, Morselli et al. 2006] tentam minimizar esses problemas empregando, por exemplo, infra-estruturas de chaves públicas (*public key infra-structures*, PKI) descentralizadas. No entanto, as mesmas requerem a troca de uma grande quantidade de mensagens entre os pares, além de dependerem da pré-existência de uma certa fração de usuários legítimos, para funcionarem como esperado.

Segundo Rowaihy [Rowaihy et al. 2007], qualquer solução baseada em identidades fracas está fadada ao fracasso, em virtude dos problemas inerentes a esse método de autenticação. Da mesma forma, usar identidades fortes "universais" persiste como um problema intratável. Por conta disso, a corrente de soluções em que a concessão de identidades é feita mediante a resolução de desafios computacionais ganhou força [Borisov 2006, Rowaihy et al. 2007]. O objetivo é associar um custo (computacional) à criação (e à renovação) de identidades, minando a capacidade de um atacante controlar uma grande quantidade de *sybils* na rede.

Apesar do sucesso das propostas baseadas em desafios, as mesmas não abordam a questão do dimensionamento da complexidade dos mesmos. Mais especificamente, ao atribuir desafios de igual complexidade para quaisquer usuários, torna-se difícil escolher um nível de complexidade que efetivamente limite a disseminação de *sybils* e ao mesmo tempo penalize minimamente os usuários legítimos. Para abordar essa limitação, propusemos em um trabalho anterior [Mauch et al. 2010, Cordeiro et al. 2011] um mecanismo em que a complexidade do desafio a ser resolvido é adaptada dinamicamente, baseando-se na frequência relativa de solicitações do(s) usuário(s) por trás de cada fonte na rede. Usuários associados a fontes cuja recorrência é menor ou similar a das demais são beneficiados com desafios menos complexos. Ao contrário, usuários associados a fontes cuja recorrência é relativamente maior são forçados a resolver desafios mais complexos.

Embora promissor, o uso de desafios aumenta consideravelmente o consumo total de energia do sistema. Medições usando o software JouleMeter [MSR 2011], em um *notebook* com Intel Core i3-350M e Windows 7 (mais detalhes na Seção 4), estimaram em 364,5 *joules* o consumo energético de um desafio que leva 5 minutos em média para ser resolvido. Acumulado ao longo do tempo, em uma comunidade P2P bastante popular, esse consumo torna-se extremamente relevante. As seções a seguir apresentam um novo mecanismo para lidar com essa limitação.

3. “Desafios Verdes” para o Gerenciamento de Identidades em Redes P2P

Dado o problema do elevado consumo energético nos mecanismos baseados em desafios computacionais, neste artigo propõe-se o estabelecimento de desafios adaptativos com espera variável para o gerenciamento de identidades em redes P2P. As subseções a seguir concentram-se em três questões relacionadas a essa proposta: (*i*) o protocolo e as entidades envolvidas no processo de solicitação de identidades, (*ii*) a identificação da origem das solicitações e (*iii*) a redução da complexidade média dos desafios atribuídos.

É importante destacar que o presente artigo considera uma rede P2P usando um esquema de identidades fracas, no qual o usuário obtêm uma nova identidade ao juntar-se à rede (e cujo tempo de vida é a duração da sessão do agente P2P do usuário). No entanto, nosso mecanismo pode ser utilizado, de maneira quase direta, em outros cenários – por exemplo um considerando identidades de longo prazo. Nesse caso, uma revalidação contínua dessas identidades durante o ciclo de vida das mesmas teria um efeito similar ao esquema de identidades fracas abordado na avaliação do mecanismo.

3.1. Definindo o Protocolo para Solicitação de Identidades

A Figura 1 apresenta uma visão geral do mecanismo proposto, destacando o protocolo para solicitação de identidades e as entidades envolvidas no processo.

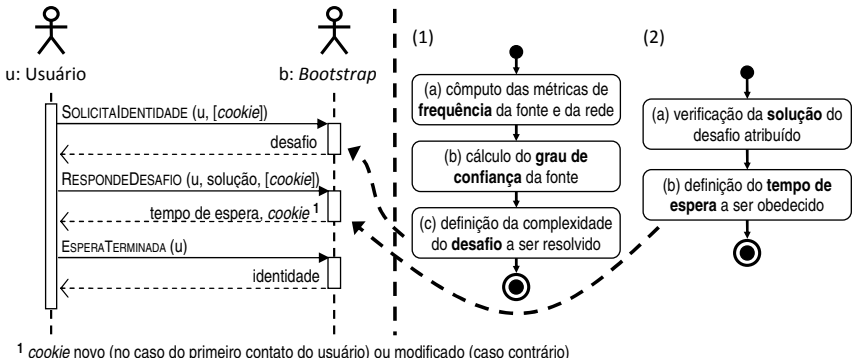


Figura 1. Protocolo para solicitação de identidades (esquerda), e sequência de passos para determinar a complexidade do desafio e o tempo de espera (direita)

Quando o usuário torna-se interessado em juntar-se a uma comunidade P2P, ele emite uma mensagem do tipo SolicitaIdentidade ao serviço de *bootstrap* da rede P2P (primeiro fluxo na Figura 1 (esquerda)). O *bootstrap* responde a essa solicitação com um desafio a ser resolvido (segundo fluxo), cuja complexidade é definida com base na frequência de solicitações da fonte de onde a solicitação foi originada. Após resolver o desafio, o usuário contacta novamente o *bootstrap* e emite a mensagem RespondeDesafio (terceiro fluxo). Uma vez verificada a validade da solução, o *bootstrap* incrementa o número de identidades obtidas pela fonte e calcula (e informa) o tempo de espera a ser obedecido (quarto fluxo). Após expirar o tempo de espera, o usuário envia a mensagem EsperaTerminada (quinto fluxo), e em resposta obtêm a identidade solicitada (sexto fluxo). Cabe destacar que as mensagens podem ser assinadas digitalmente, ou registradas pelo *bootstrap*, de modo a impedir, por exemplo, que um atacante informe a solução de um desafio diferente do atribuído.

3.2. Identificando as Fontes de Solicitação de Identidades com *Cookies*

O conceito de fontes de solicitação de identidades [Mauch et al. 2010] representa um dos pilares fundamentais sobre o qual o mecanismo proposto neste artigo é apoiado. Uma

fonte é definida como uma agregação de um ou mais usuários – que compartilham similaridades de localização – de onde partem as solicitações de identidades.

Duas estratégias tem sido tradicionalmente usadas para identificar a fonte de onde partem as solicitações por serviços (no contexto desse trabalho, a obtenção de identidades): a diferenciação por endereço IP e o uso de sistemas de coordenadas de rede [Sherr et al. 2009]. Embora efetivas, elas possuem importantes limitações. Por um lado, a diferenciação por IP pode penalizar usuários localizados em redes que usam NAT. Por outro lado, sistemas de coordenadas de rede podem não distinguir precisamente usuários localizados, por exemplo, em uma mesma região, e são vulneráveis a ataques que visam forjar a real localização geográfica do usuário. Para superar essas limitações, é introduzido no contexto deste trabalho o conceito de *cookies* de solicitações. Criado pelo *bootstrap* e concedido a cada usuário na primeira solicitação, um *cookie* possui (*i*) um identificador universal, (*ii*) o *timestamp* da última validação, (*iii*) o grau de confiança (discutido com mais detalhes a seguir), (*iv*) um histórico de identidades obtidas e (*v*) uma assinatura digital (de modo a proteger o *cookie* contra modificações não autorizadas).

Para obter um *cookie*, o usuário deverá pagar o preço computacional definido em função da reputação da fonte a qual está associado. Esse preço, parametrizado com base em um valor diferenciado (maior) de complexidade máxima, visa impedir a obtenção indiscriminada de *cookies* por um atacante. A partir da segunda solicitação, o usuário informa o *cookie* que possui (como ilustrado no primeiro fluxo na Figura 1 (esquerda)). O *boostrap* – ao computar o "preço" que deverá ser pago (complexidade do desafio e tempo de espera) – considera apenas o histórico de solicitações registradas no *cookie* (ao invés de todas as solicitações da fonte a qual o usuário está associado). Quando o desafio é resolvido, o *boostrap* adiciona ao *cookie* o *timestamp* de quando ocorreu esse evento.

O identificador universal e o *timestamp* da última validação servem para impedir que um usuário mal intencionado informe arbitrariamente um *cookie* "desatualizado". Ao receber o *cookie*, o *bootstrap* consulta o identificador em uma base de dados interna, e compara o *timestamp* registrado com o do *cookie*. Caso sejam iguais, o processamento da solicitação segue normalmente. Caso contrário, o atendimento da solicitação pode ser abortado. Toda modificação no *cookie* é acompanhada da atualização desse *timestamp*. Observe que embora incentivado, o uso de cookies é opcional para um usuário em particular; caso opte por não salvar cookies, ele arcará com custo diferenciado (maior) de complexidade máxima de desafios.

3.3. Empregando Desafios Adaptativos e Espera Variável para Conter *Sybils*

A seguir, é discutido (seguindo o fluxo do algoritmo apresentado na Figura 1) como a espera variável é empregada para reduzir o consumo energético no mecanismo proposto.

Caracterizando o Comportamento das Fontes de Solicitações de Identidades

A caracterização do comportamento das fontes representa a primeira etapa do processo de gerenciamento de identidades (atividade 1(a) na Figura 1). Para tal, é necessário manter um registro das identidades concedidas aos usuários associados a uma determinada fonte (ou *cookie*). Essa informação, definida como $\phi_i(t)$ para a *i*-ésima fonte (ou *cookie*, caso seja apresentado pelo solicitante) no instante t (com $\phi_i(t) \in \mathbb{N}$), serve como base para o cálculo das taxas de recorrência da fonte (dado por $\Delta\phi_i = \phi_i(t) - \phi_i(t - \Delta t)$, em função de uma janela de tempo Δt) e da rede ($\Phi(t)$, definido pela Equação 1). Nesse contexto, n representa o número de fontes *online*, isto é, que obtiveram pelo menos uma identidade durante a janela de tempo Δt.

$$\Phi(t) = \begin{cases} 1 & \text{, se } n = 0 \\ \frac{1}{n} \times \sum_{i=1}^{n} \Delta\phi_i & \text{, se } n \geq 1 \end{cases} \quad (1)$$

Os cálculos de $\Delta\phi_i$ e $\Phi(t)$ seguem uma metodologia diferente da proposta por Mauch *et al.* [Mauch et al. 2010]. Na proposta original, $\phi_i(t)$ considera o número de identidades *solicitadas*, enquanto $\Phi(t)$ é dado pela *média harmônica* da recorrência das fontes na rede. No contexto deste artigo, $\phi_i(t)$ considera o número de identidades *obtidas*, e $\Phi(t)$ representa a *média simples* das recorrências. Essa nova metodologia torna o mecanismo mais flexível a mudanças no comportamento da rede (ao empregar a média simples para computar $\Phi(t)$), mas sem perder a resistência a ataques que visam alterar a percepção de normalidade (ao definir $\phi_i(t)$ como indicador de identidades obtidas).

Determinando o Grau de Confiança a partir dos Comportamentos Observados

A segunda etapa do processo de gerenciamento de identidades (atividade 1(b) na Figura 1) inicia-se com o cálculo da relação entre as recorrências da fonte ($\Delta\phi_i$) e da rede ($\Phi(t)$). Essa relação, denotada por $\rho_i(t)$ e definida de acordo com a Equação 2, assume valores negativos para indicar quantas vezes a recorrência da i-ésima fonte é menor que a da rede, e positivos para indicar o contrário.

$$\rho_i(t) = \begin{cases} \frac{1}{\Phi(t)} - 1 & \text{, se } \Delta\phi_i = 0 \\ 1 - \frac{\Phi(t)}{\Delta\phi_i} & \text{, se } 1 \leq \Delta\phi_i \leq \Phi(t) \\ \frac{\Delta\phi_i}{\Phi(t)} - 1 & \text{, se } \Delta\phi_i > \Phi(t) \end{cases} \quad (2)$$

A métrica $\rho_i(t)$ serve como entrada para calcular o grau de confiança parcial da fonte ($\theta_i(t)$). Definido conforme a Equação 3, $\theta_i(t)$ assume valores no intervalo $[0, 1]$, com o extremo 0 representando total desconfiança e o 1, total confiança na legitimidade do(s) usuário(s) associado(s) à i-ésima fonte.

$$\theta_i(t) = 0.5 - \frac{\arctan(\Phi(t) \times \rho_i(t)^3)}{\pi} \quad (3)$$

A partir de $\theta_i(t)$ é obtido um *grau de confiança amortizado*, que leva em consideração o comportamento histórico do(s) usuário(s) associado(s) à fonte. Denotado por $\theta'_i(t_k)$ no instante t_k, ele atribui peso β (com $\beta \in (0, 1]$) para o grau de confiança parcial $\theta_i(t)$, e peso $1 - \beta$ para o último grau de confiança amortizado calculado.

$$\theta'_i(t_k) = \begin{cases} \theta_i(t) & \text{, se } k = 0 \\ \beta \times \theta_i(t) + (1 - \beta) \times \theta'_i(t_{k-1}) & \text{, se } k \geq 1 \end{cases} \quad (4)$$

Calculando a Complexidade do Desafio a ser Atribuído

Essa etapa (atividade 1(c) na Figura 1) utiliza como entrada o grau de confiança amortizado $\theta'_i(t_k)$ para estabelecer a complexidade do desafio a ser resolvido. A função de mapeamento entre confiança e complexidade, definida abstratamente como $f : \Theta \rightarrow \mathbb{N}$, depende essencialmente da natureza do desafio adotado.

No contexto deste artigo, o seguinte desafio é utilizado [Douceur 2002]: dado um número aleatório y, encontrar dois números x e z de modo que a concatenação $x|y|z$, após processada por uma função *hash*, gere como resultado um número cujos γ *bits* menos significantes sejam 0. O tempo para encontrar uma solução para esse desafio é proporcional a $2^{\gamma-1}$, enquanto que o tempo para verificar a validade da mesma é constante. Nesse caso, uma função de mapeamento para obter a complexidade γ é dada pela Equação 5. Nessa equação, Γ define a maior complexidade que um desafio pode assumir (para o caso em que o grau de confiança da fonte for zero).

$$\gamma_i(t_k) = \lfloor \Gamma \times (1 - \theta'_i(t_k)) + 1 \rfloor \tag{5}$$

Conforme mencionado na Subseção 3.2, Γ deve ser diferenciado para aquelas solicitações sem *cookie*. Para isso, define-se como $\Gamma = \Gamma_{orig}$ a complexidade máxima a ser usada para solicitações sem *cookie*, e $\Gamma = \Gamma_{cookie}$, caso contrário (com $\Gamma_{orig} > \Gamma_{cookie}$). Essa penalidade maior não somente incentiva o uso de *cookies*, como também dificulta ainda mais a obtenção indiscriminada de *cookies* (e de identidades).

Validando a Solução do Desafio e Determinando o Tempo de Espera

Após resolver o desafio, o usuário envia a mensagem RESPONDEDESAFIO ao *bootstrap* (terceiro fluxo Figura 1 (esquerda)), informando a solução encontrada. Ao verificar que a mesma é válida (atividade 2(a)), o *bootstrap* incrementa $\phi_i(t)$ e define (e informa) o tempo ω a ser esperado pelo usuário (atividade 2(b)). Após expirado o tempo de espera, o usuário envia a mensagem ESPERATERMINADA. O *bootstrap* concede então a identidade solicitada, caso o tempo de espera tenha sido integralmente obedecido.

O tempo de espera a ser obedecido pelo usuário é definido em função do grau de confiança atual da fonte a qual o mesmo está associado, de acordo com a Equação 6. Nessa equação, Ω representa o maior tempo de espera passível de ser atribuído.

$$\omega_i(t_k) = 2^{\Omega \times (1 - \theta'_i(t_k))} \tag{6}$$

Um possível ataque a esse protocolo é a "paralelização" da espera. Isto é, um atacante pode solicitar n identidades num mesmo instante, resolver os desafios atribuídos e usar o mesmo intervalo de tempo para cumprir simultaneamente as n esperas estipuladas. Para mitigar esse ataque, a mensagem informando o tempo de espera a ser obedecido pode incluir o grau de confiança usado para determiná-lo. Quando o usuário enviar a mensagem ESPERATERMINADA, o *bootstrap* verifica se a diferença entre a confiança usada para definir o tempo de espera e a confiança atual excede um limiar $\Delta\theta$ (isto é, $\theta'_i(t_{k-1}) - \theta'_i(t_k) > \Delta\theta$). Caso positivo, o *bootstrap* poderá abortar unilateralmente o processo de solicitação da identidade, e uma nova requisição deverá ser feita pelo usuário. Essa proteção baseia-se no fato de que o grau de confiança da fonte diminui cada vez que um desafio é solucionado – isso porque $\phi_i(t)$ é incrementado durante a verificação da validade da solução, ao invés do momento em que a identidade é concedida.

4. Avaliação Experimental

Para avaliar o desempenho e a eficiência energética do mecanismo proposto, foram realizados diversos experimentos por meio de simulação e posterior instanciação no PlanetLab. Os experimentos foram norteados por três questões fundamentais. Primeiro, qual o desempenho do mecanismo na contenção de *sybils*? Segundo, qual a sua eficiência energética? E terceiro, qual o benefício dos *cookies* aos usuários legítimos? Para responder

Tabela 1. Características do traço de solicitações de identidades empregado

Primeira solicitação	Dom Jan 1 00:00:47 2006	Última solicitação	Sab Jan 7 20:53:09 2006
Número total de fontes de solicitação	44.066	Número total de solicitações	203.060
Duração do traço (horas)	164,87	Média de solicitações por minuto	20,52
Média de solicitações por fonte	4,60808	Desvio padrão das frequências	4,57688
Menor e maior frequência de solicitação	1 e 273	Moda e mediana das frequências	1 e 3

essas questões, foram avaliados cenários com e sem ataque, para cada um dos mecanismos estudados. A Subseção 4.1 concentra-se nos resultados obtidos via simulação, enquanto que a Subseção 4.2 decreve os experimentos realizados no PlanetLab.

4.1. Simulação Baseada em Traços Reais de Solicitações de Identidades

Nessa avaliação são considerados os seguintes mecanismos: sem controle, baseado em desafios fixos [Borisov 2006], adaptativo [Mauch et al. 2010] e o proposto neste artigo. Para avaliá-los via simulação, os seguintes aspectos precisam ser observados: (*i*) o comportamento dos usuários legítimos, (*ii*) o poder computacional dos usuários, (*iii*) o custo computacional da resolução dos desafios e (*iv*) o objetivo do atacante (suas estratégias de ataque e seu poder computacional). Cada um desses aspectos é discutido a seguir.

Configuração do Ambiente de Simulação

Em relação ao comportamento dos usuários legítimos, foi utilizado um traço real de sessões de participação de pares em *torrents*, em uma comunidade BitTorrent. As principais características do mesmo são apresentadas na Tabela 1. O poder computacional dos usuários legítimos é modelado seguindo uma distribuição exponencial ($\lambda_{pc} \approx 0.003$), variando de 0,1 a 2,5 vezes o poder de um computador pessoal típico (tido como referência). Essa é uma modelagem arbitrária para fins de simulação somente, uma vez que os autores desconhecem um modelo que seja representativo de usuários de redes P2P.

Para modelar o atraso causado pela resolução dos desafios, foi considerado o desafio apresentado por Douceur [Douceur 2002] e descrito na Subseção 3.3. Além disso, por simplicidade, considera-se que um desafio com complexidade $\gamma_i(t_k)$ leva $2^6+2^{\gamma_i(t_k)-1}$ segundos para ser resolvido em um computador poderoso (com poder computacional normalizado em 1, para referência); um computador 2 vezes mais rápido, por sua vez, demora metade desse tempo para resolver o mesmo desafio.

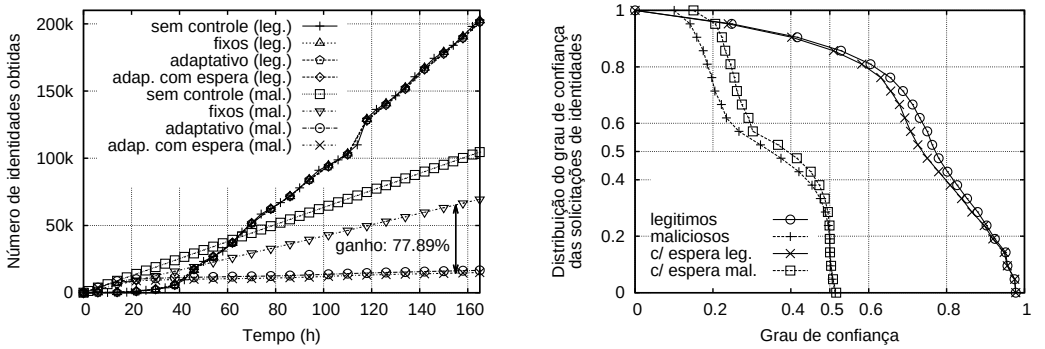
O objetivo do atacante é controlar 1/3 das identidades na rede, ou seja, 104.606 identidades (proporção que excede a quantidade tolerável de *sybils* na rede [Yu et al. 2008]). Considerando o traço empregado, foram definidos três cenários para a avaliação: (*i*) sem ataque, (*ii*) com 1% de fontes maliciosas (em relação ao número de fontes legítimas do traço) e (*iii*) 10% de fontes. No segundo cenário, o atacante controla 440 fontes distintas, e possui um *cluster* de 100 computadores do tipo mais potente (2,5 vezes mais rápido que o computador usado como referência) para resolver os desafios. No terceiro cenário, assume-se que o atacante controla uma *botnet* com 4.406 computadores zumbis, todos do tipo mais potente.

Os parâmetros dos mecanismos são definidos como segue. Para o mecanismo adaptativo, a janela de tempo Δt é igual a 48 horas, o fator de amortização $\beta = 0.125$, e a complexidade máxima possível de um desafio é $\Gamma = 18$. Para o presente mecanismo, além dos valores de Δt e β já mencionados, utiliza-se $\Gamma_{cookie} = 13$ e $\Gamma_{orig} = 15$, e o fator de tempo de espera máximo $\Omega = 17$ (o que resulta em 131.072 segundos, ou aproximadamente 36 horas, de espera máxima possível). O mecanismo baseado em desafios

fixos considera um valor de complexidade $\gamma = 12$, o qual leva entre 14 minutos e 6 horas (aproximadamente) para ser resolvido, dependendo da capacidade do computador.

Eficácia do Mecanismo Proposto na Contenção de Sybils

Os resultados obtidos evidenciam a eficácia do mecanismo proposto na contenção de *sybils*. Observe na Figura 2(a) que o mesmo obteve ganho equivalente ao do adaptativo (redução de aproximadamente 78% no número de *sybils*, em relação ao uso de desafios fixos). Ao mesmo tempo, causou uma sobrecarga negligível aos usuários legítimos. A Figura 2(b), por sua vez, mostra que a distribuição do grau de confiança das solicitações foi minimamente afetada. Enquanto a maioria das solicitações legítimas recebeu valores de confiança mais altos (maior ou igual a 0,5 para 87%), as maliciosas foram bastante penalizadas (sendo atribuídos valores menores que 0,5 para mais que 74%).



(a) Identidades concedidas (b) CCDF do grau de confiança das solitiações

Figura 2. Identidades concedidas e grau de confiança das solicitações, para o segundo cenário de avaliação (1% de fontes maliciosas)

A Tabela 2 apresenta, para cada cenário avaliado, indicadores sobre ambas as *solicitações legítimas* e *maliciosas*. Observe que a contenção de *sybils* com desafios adaptativos (com ou sem espera variável) é superior em relação à conseguida com desafios fixos em todos os cenários. Para as solicitações legítimas, a proporção de identidades concedidas (em relação a um cenário "sem controle") foi no mínimo de 98,75%, similar à proporção alcançada com desafios fixos (99,74%). Para as solicitações maliciosas, apenas 14,25% das mesmas são atendidas (em contraste com 66,44% de *sybils* criados com desafios fixos), no cenário em que o atacante tem recursos mais limitados. Em uma situação extrema, na qual o atacante é bastante provido de recursos para resolver os desafios (uma *botnet* com 4.406 computadores poderosos), essa proporção foi de 96,4% – melhor que o desempenho dos desafios fixos, que não teve efetividade nessa situação.

Observe ainda na Tabela 2 que a quantidade de desafios menos complexos atribuídos (em relação aos de complexidade fixa, coluna "$<$ *fixo*") é significativamente maior. Por exemplo, no cenário com 1% de fontes maliciosas, enquanto que 91,79% das solicitações legítimas receberam desafios adaptativos de complexidade comparativamente menor, no mecanismo proposto esse número aumentou para 97,12%. Para as solicitações maliciosas, esse índice saltou de 48,65% para 70,93%. É importante lembrar que, embora seja desejável continuar sobrecarregando o atacante com desafios cada vez mais complexos, isso acarreta em um maior consumo de energia – conflitando assim com o objetivo do mecanismo proposto. É nesse contexto que a espera de tempo é empregada como penalidade adicional, de forma inversamente proporcional à confiança na solicitação.

Tabela 2. Resultados obtidos com cada mecanismo, nos cenários avaliados

Cenário	Solicitações legítimas				Solicitações maliciosas			
	atendidas	% "sem controle"	< fixo	% solicit.	atendidas	% "sem controle"	< fixo	% solicit.
Fixo, sem mal.	202.539	99,74	-	-	-	-	-	-
Adaptativo, sem mal.	201.495	99,23	176.257	86,80	-	-	-	-
Adap. c/ espera, sem mal.	200.524	98,75	195.372	96,21	-	-	-	-
Fixo, 1% mal.	202.539	99,74	-	-	69.500	66,44	-	-
Adaptativo, 1% mal.	201.744	99,35	186.385	91,79	16.631	15,90	8.139	48,65
Adap. c/ espera, 1% mal.	200.931	98,95	197.210	97,12	14.902	14,25	10.734	70,93
Fixo, 10% mal.	202.539	99,74	-	-	104.457	99,86	-	-
Adaptativo, 10% mal.	201.971	99,46	191.050	94,09	99.981	95,58	30.842	29,55
Adap. c/ espera, 10% mal.	201.387	99,18	198.562	97,79	100.837	96,40	99.505	95,12

Análise da Eficiência Energética

A estimativa do consumo energético considerado nessa análise foi obtida com base na resolução dos desafios usando um programa implementado em *python*, em um *notebook* Intel Core i3-350M, 3 MB de memória cache, 2.26 GHz de velocidade de CPU, executando Windows 7. O *software* JouleMeter [MSR 2011] foi utilizado para as medições, e somente o consumo do processador foi considerado. Em uma média de 10 execuções do programa para cada valor de complexidade considerado neste estudo, observou-se que o consumo (com a utilização do processador em 100%) é constante e igual a 1,215 *joules*. É importante ressaltar que, embora não considere os mais variados tipos de *hardware* e processadores existentes, essa estimativa não deixa de ser um indicativo importante – e até então desconsiderado por trabalhos anteriores – do consumo médio esperado para um mecanismo baseado em desafios computacionais.

A Tabela 3 fornece um panorama geral do consumo energético durante uma semana (período de duração do traço), comparando o mecanismo proposto com o adaptativo, para o segundo cenário avaliado. Nela são apresentadas as complexidades consideradas na avaliação, estimativas do tempo necessário para resolver um desafio de tal complexidade (coluna *resolução*) e o consumo estimado em *joules*. Observe que o consumo energético total (somando o dos legítimos com o dos atacantes) causado pelo uso de desafios adaptativos com espera variável (67,67 *megajoules*) é significativamente menor que no uso do mecanismo adaptativo (903,02 MJ). Mais importante, é menor que o consumo estimado para o mecanismo baseado em desafios fixos (699,66 MJ; resultado de 203.058 desafios de complexidade 12 atribuídos aos legítimos e 69.600 atribuídos aos maliciosos). De fato, o consumo do mecanismo proposto representa apenas 7,49% do consumo do mecanismo adaptativo e 9,67% do consumo do mecanismo baseado em desafios fixos. Esse contraste representa uma diferença de 835,34 MJ (232,04 KWh) e 631,98 MJ (175,55 KWh), respectivamente; segundo o IBGE [IBGE 2001], essas diferenças são maiores que o consumo médio mensal domiciliar nas cidades de Brasília (221 KWh) e Salvador (171 KWh). Esses resultados não apenas enfatizam a necessidade de desafios mais "verdes", como também destacam as potencialidades do uso de espera variável para este fim.

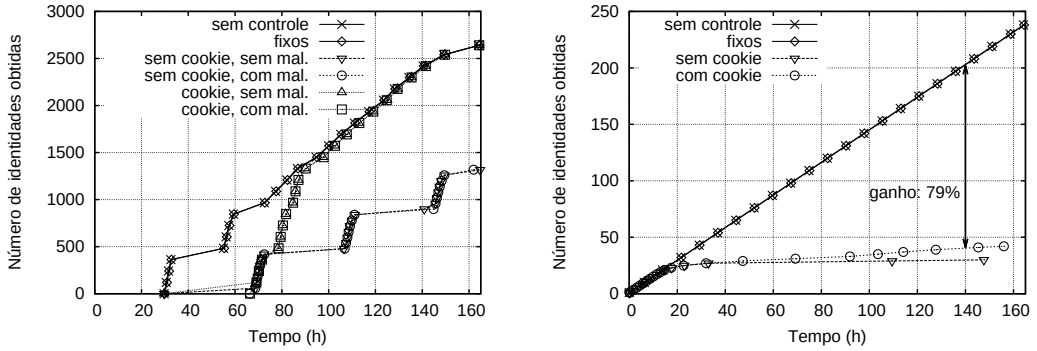
Benefícios dos Cookies *de Solicitação de Identidades*

Para a análise sobre o uso de *cookies*, foi considerado o caso de uma fonte contendo 1% (440) de usuários legítimos (em relação ao total descrito na Tabela 1). Os usuários associados a essas fontes solicitam seis identidades cada, durante uma semana, perfazendo um total de 2.640 identidades. A primeira chegada de cada usuário segue uma distribuição exponencial, com $\lambda = 1.195 \times 10^{-5}$; o intervalo entre solicitações é uniformemente distribuído entre 12 e 30 horas. O poder computacional desses usuários

Tabela 3. Complexidade dos desafios e consumo energético (estimativas)

Complexidade do desafio	Resolução (segundos)	Consumo (*joules*)	Desafios adaptativos				Desafios adaptativos com espera variável			
			# leg.	KJ leg.	# mal.	KJ mal.	# leg.	KJ leg.	# mal.	KJ mal.
0	65	78,97	40.471	3.196,20	0	0	42.695	3.371,84	0	0
1	66	80,19	21.564	1.729,22	0	0	27.833	2.231,93	0	0
2	68	82,62	19.214	1.587,46	0	0	23.407	1.933,89	0	0
3	72	87,48	21.294	1.862,80	0	0	30.384	2.657,99	0	0
4	80	97,2	30.530	2.967,52	0	0	34.026	3.307,33	0	0
5	96	116,64	21.427	2.499,25	0	0	13.888	1.619,90	0	0
6	128	155,52	10.658	1.657,53	0	0	7.628	1.186,31	84	13,06
7	192	233,28	7.110	1.658,62	0	0	5.976	1.394,08	6.346	1.480,39
8	320	388,8	5.604	2.178,84	3.075	1.195,56	4.657	1.810,64	1.375	534,60
10	1.088	1.321,92	4.634	3.243,06	3.992	2.793,76	3.674	2.571,21	750	524,88
9	576	699,84	3.879	5.127,73	1.072	1.417,10	3.042	4.021,28	2.179	2.880,46
11	2.112	2.566,08	3.306	8.483,46	826	2.119,58	2.517	6.458,82	3.777	9.692,08
12	4.160	5.054,4	2.826	14.283,73	761	3.846,40	3.331	16.836,21	623	3.148,89
13	8.256	10.031,04	2.384	23.914,00	1.639	16.440,87	0	0	0	0
14	16.448	19.984,32	2.133	42.626,55	3.533	70.604,60	0	0	0	0
15	32.832	39.890,88	1.850	73.798,13	1.801	71.843,47	0	0	0	0
16	65.600	79.704	1.578	125.772,91	32	2.550,53	-	-	-	-
17	131.136	159.330,24	2.596	413.621,30	0	0	-	-	-	-
Total			203.058	730.208,30	16.731	172.811,88	203.058	49.401,42	15.134	18.274,38

segue o mesmo modelo descrito anteriormente. Para fins de comparação, foram avaliados cenários sem controle, com desafios fixos, e com controle; nos casos com controle, um sub-cenário sem e outro com uso de *cookies* foram analisados. Para todos esses cenários, foram avaliados casos sem e com ataque. Nos casos com ataque, um atacante sozinho (com um *hardware* 2,5 vezes mais rápido que o de referência) solicita 238 novas identidades (10% das solicitações maliciosas), no mesmo período de uma semana.



(a) Legítimos (b) Maliciosos

Figura 3. Identidades concedidas a usuários associados a uma mesma fonte

A Figura 3 apresenta os resultados obtidos. As curvas "com mal." indicam os cenários em que há um atacante situado na mesma fonte, e "sem mal." quando não há. Observe que o uso de *cookies* melhora significativamente a dinâmica de concessão de identidades aos usuários legítimos, ao passo que proporciona uma vantagem negligível aos atacantes. Além disso, é possível observar que a influência dos atacantes na concessão de identidades aos legítimos é praticamente nula. É importante mencionar que a sobrecarga aos usuários legítimos no período entre 30 e 64 horas (Figura 3(a)) deve-se ao fato de que a primeira solicitação dos legítimos ocorre considerando a reputação da fonte,

a qual foi bastante prejudicada devido às inúmeras solicitações feitas pelos maliciosos, no período entre 0 e 20 horas (Figura 3(b)). No entanto, uma vez obtido o *cookie*, os usuários legitimos são minimamente penalizados, conforme esperado.

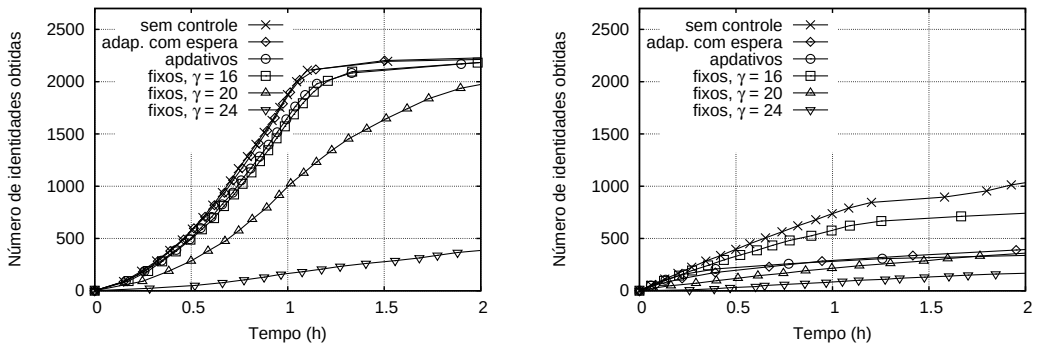
4.2. Avaliação Experimental no PlanetLab

O objetivo dessa avaliação – realizada utilizando o *framework BitTornado*, e considerando os mesmos mecanismos estudados na subseção anterior – é verificar a viabilidade técnica do uso de desafios adaptativos com espera variável na contenção de *sybils*. Além disso, o estudo visa comparar o desempenho do mecanismo proposto ao dos demais existentes.

Para os fins dessa avaliação, foi gerado um traço de solicitações de identidades com 240 usuários legítimos e 20 maliciosos. Os legítimos solicitam 2.400 identidades no período de uma hora. A primeira solicitação de cada usuário é uniformemente distribuída no período; a recorrência dos mesmos segue uma distribuição exponencial ($\lambda_r \approx 0,5069$), variando de 1 a 15. O tempo entre chegadas também é exponencialmente distribuído ($\lambda_{tc} \approx 0,0084$), entre 1 e 10 minutos. Em relação ao atacante, este solicita 1.200 identidades (1/3 das identidades do sistema), perfazendo uma média de 60 identidades por fonte maliciosa; a recorrência dos mesmos segue uma taxa fixa de uma solicitação de identidade por minuto. A avaliação (incluindo o comportamento dos usuários legítimos e maliciosos) foi definida atendendo às restrições impostas pelo ambiente PlanetLab.

Nesse experimento, um usuário contacta o *tracker* para solicitar uma identidade e ingressar num enxame BitTorrent. Após obter a identidade (observando o protocolo definido anteriormente), o usuário entra em contato com o *tracker* para obter uma lista de pares e, finalmente, baixar um arquivo de 16 MB. Terminado o *download*, o usuário encerra o processo, desconecta-se da rede e, após esperar o tempo entre chegadas definido no traço de solicitações de identidades, entra novamente em contato com o *tracker*, repetindo o ciclo. Os usuários maliciosos obedecem esse mesmo comportamento.

Os parâmetros dos mecanismos foram definidos como segue. Para o mecanismo adaptativo, $\Delta t = 48$ horas, $\beta = 0.125$ e $\Gamma_{pl} = 25$ (o que equivale a $\Gamma = 7$ no modelo matemático usado na simulação). Para o mecanismo proposto, além dos mesmos valores para Δt e β, são usados $\Gamma_{pl,cookie} = 20$ ($\Gamma_{cookie} = 2$), $\Gamma_{pl,orig} = 22$ ($\Gamma_{orig} = 4$) e $\Omega = 10$. Para o mecanismo baseado em desafios fixos, três cenários são considerados: $\gamma_{pl} = 16$ (1), $\gamma_{pl} = 20$ (2) e $\gamma_{pl} = 24$ (6). É importante mencionar que a diferença entre o modelo matemático de desafio e as configurações reais adotadas são ajustes necessários para adaptar os mecanismos às restrições de uso de CPU vigentes no ambiente PlanetLab.



(a) Legítimos (b) Maliciosos

Figura 4. Resultados alcançados com a avaliação dos mecanismos no PlanetLab

Os resultados, apresentados na Figura 4, mostram que a dinâmica de concessão de identidades aos legítimos no mecanismo proposto ("adap. com espera") é similar à observada no caso "sem controle". Além disso, ela mostrou-se ligeiramente melhor que no caso do mecanismo adaptativo. Ao mesmo tempo, observa-se a sobrecarga/ineficácia decorrente do uso de desafios fixos. Focando nos usuários maliciosos, observa-se que o mecanismo proposto reduziu significativamente a quantidade de *sybils* criados (em comparação ao caso "sem controle").

As estimativas de consumo energético obtidas nesses experimentos também indicam a superioridade do mecanismo proposto. Enquanto que o consumo energético para o uso de desafios fixos com $\gamma_{pl} = 16$, $\gamma_{pl} = 20$ e $\gamma_{pl} = 24$ ficou em 58,70 KJ, 553,85 KJ e 803,92 KJ, respectivamente, o mecanismo proposto gerou um consumo de 13,39 KJ. Esse consumo representa 22,81%, 2,41% e 1,66% do consumo observado com o uso de desafios fixos, respectivamente. Além disso, o consumo do mecanismo proposto foi 88,59% menor que o observado com o mecanismo adaptativo (117,35 KJ). Em resumo, os experimentos com o ambiente PlanetLab não somente confirmaram os resultados alcançados via simulação, como também evidenciam a viabilidade técnica do uso de desafios adaptativos com espera variável na contenção de *sybils* com eficiência energética.

5. Considerações Finais

O uso de desafios computacionais há longo tempo despertou o interesse da comunidade científica como uma potencial solução para contenção de *sybils*. No entanto, a falta de mecanismos justos com usuários legítimos e severos com atacantes, e que ao mesmo tempo promovessem o uso eficiente dos recursos, impediu o emprego mais amplo dessa abordagem. Para superar essa lacuna, propusemos neste artigo o uso de desafios adaptativos com espera variável como um mecanismo para conter a disseminação de *sybils*. O uso de tempo de espera, técnica tradicionalmente utilizada em *websites* para limitar o acesso a serviços, apoiada pelo conceito de *cookies* de solicitação de identidades (também proposto no escopo deste trabalho), proporcionou uma significativa economia de energia (de no mínimo 90,33% em comparação com as propostas baseadas em desafios fixos [Borisov 2006, Rowaihy et al. 2007]). Aliado ao consumo reduzido de energia, comparativamente, o mecanismo proposto proporciona um ganho de 8,89% na eficácia de contenção de *sybils* em relação à melhor proposta até então.

Os experimentos realizados via simulação e PlanetLab evidenciaram duas questões associadas a desafios computacionais. Primeiro, confirmou-se a inviabilidade do uso de desafios fixos, dada a dificuldade em encontrar um valor de complexidade ao mesmo tempo eficaz contra atacantes e pouco danoso aos usuários legítimos. E segundo, os desafios criptográficos não se mostraram totalmente confiáveis em garantir que um usuário será realmente penalizado conforme esperado. Por exemplo, o tempo de resolução de desafios de uma mesma complexidade, em um mesmo *hardware multi-core*, variou de alguns segundos a centenas de minutos. O tratamento dessa questão é proposto como trabalho futuro na área. Outras propostras incluem o aprimoramento do mecanismo para aliviar ainda mais o impacto sobre os usuários legítimos, uma análise mais aprofundada sobre o dimensionamento do tempo para a resolução de um desafio, e a análise do mecanismo em face a casos mais extremos e adversos, com o apoio do PlanetLab.

Agradecimentos

O trabalho apresentado neste artigo foi parcialmente conduzido com recursos provenientes do Projeto no. 560226/2010-1, financiado pelo CNPq (Edital MCT/CNPq no. 09/2010 PDI - Pequeno Porte).

Referências

Aberer, K., Datta, A., and Hauswirth, M. (2005). A decentralized public key infrastructure for customer-to customer e-commerce. In *International Journal of Business Process Integration and Management*, pages 26–33.

Borisov, N. (2006). Computational puzzles as sybil defenses. In *6th IEEE International Conference on Peer-to-Peer Computing (P2P 2006)*, pages 171–176.

Cordeiro, W., Santos, F. R., Mauch, G. H., Gaspary, L. P., and Barcellos, M. P. (2011). Securing p2p systems from sybil attacks through adaptive identity management. In *CNSM 2011, Mini-conference Proceedings*, pages 1–6.

Danezis, G. and Mittal, P. (2009). Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*. The Internet Society.

Douceur, J. R. (2002). The sybil attack. In *1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pages 251–260.

Ellison, C. (1996). Establishing identity without certification authorities. In *6th USENIX Security Symposium*, pages 67–76.

Feldman, M., Papadimitriou, C., Chuang, J., and Stoica, I. (2006). Free-riding and whitewashing in peer-to-peer systems. *IEEE JSAC*, 24(5):1010–1019.

Fiege, U., Fiat, A., and Shamir, A. (1987). Zero knowledge proofs of identity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 210–217, New York, NY, USA. ACM.

IBGE (2001). Metodologia de Cálculo da Estimativa da Variação de Preços do Subitem Energia Elétrica.

Jetter, O., Dinger, J., and Hartenstein, H. (2010). Quantitative analysis of the sybil attack and effective sybil resistance in peer-to-peer systems. In *ICC 2010*, pages 1 –6.

Mauch, G. H., Santos, F. R., da Costa Cordeiro, W. L., Gaspary, L. P., and Barcellos, M. P. (2010). Dois pesos, duas medidas: Gerenciamento de identidades orientado a desafios adaptativos para contenção de sybils. In *SBRC 2010*, pages 17–30.

Morselli, R., Bhattacharjee, B., Katz, J., and Marsh, M. A. (2006). Keychains: A decentralized public-key infrastructure. http://hdl.handle.net/1903/3332.

MSR (2011). Joulemeter: Computational Energy Measurement and Optimization.

Rowaihy, H., Enck, W., McDaniel, P., and La Porta, T. (2007). Limiting sybil attacks in structured p2p networks. In *INFOCOM 2007*, pages 2596–2600.

Sherr, M., Blaze, M., and Loo, B. T. (2009). Veracity: Practical Secure Network Coordinates via Vote-based Agreements. In *USENIX Annual Conference (USENIX '09)*.

Singh, A., Ngan, T.-W., Druschel, P., and Wallach, D. S. (2006). Eclipse attacks on overlay networks: Threats and defenses. In *INFOCOM 2006*, pages 1–12.

Tran, N., Li, J., Subramanian, L., and Chow, S. (2011). Optimal sybil-resilient node admission control. In *INFOCOM 2011*, pages 3218 –3226.

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. (2011). Uncovering social network sybils in the wild. *CoRR*, abs/1106.5321.

Yu, H., Gibbons, P. B., Kaminsky, M., and Xiao, F. (2008). Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17. IEEE Computer Society.