# Um Novo Sistema de Rastreamento de Pacotes IP contra Ataques de Negação de Serviço*

**Rafael P. Laufer[1], Pedro B. Velloso[2] e Otto Carlos M. B. Duarte[1]**

[1]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro - COPPE/POLI
Rio de Janeiro, Brasil

[2]Laboratoire d´Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie - Paris VI
Paris, França

rlaufer@gta.ufrj.br, pedro.velloso@lip6.fr, otto@gta.ufrj.br

***Abstract.*** *On most denial-of-service attacks, packets with spoofed source addresses are employed in order to disguise the attack's true origin. A defense strategy is to trace back the source of every attack packet for the sake of penalizing the attacker or isolating him from the network. To date, the proposed traceback systems require either large amounts of storage space on routers or a sufficient number of received attack packets. In this paper, a new IP traceback system is proposed to determine the source of every packet received by the victim without storing state in the network infrastructure. For practical purposes, a generalization of the Bloom-filter theory is deployed and a corresponding mathematical evaluation is performed. Analytical results are presented to show efficacy of the proposed system.*

***Resumo.*** *Em grande parte dos ataques de negação de serviço, pacotes IP com endereços de origem forjados são usados para ocultar a verdadeira origem do atacante. Uma possível estratégia de defesa é rastrear a origem de cada pacote de ataque, de forma a penalizar o atacante ou isolá-lo da rede. Até o momento, os sistemas propostos para o rastreamento requerem o armazenamento de um grande volume de informação nos roteadores ou um número mínimo de pacotes de ataque recebidos. Neste artigo, é proposto um novo sistema de rastreamento de pacotes IP para determinar a origem de cada pacote recebido pela vítima sem armazenar estado na infra-estrutura de rede. Para seu funcionamento, é desenvolvida uma generalização da teoria de filtro de Bloom e realizada sua respectiva análise matemática. Resultados analíticos são apresentados de forma a mostrar a eficácia do sistema proposto.*

## 1. Introdução

A atual infra-estrutura de roteamento ainda é extremamente vulnerável a ataques de negação de serviço (*Denial of Service* - DoS) anônimos [CERT, 2003]. Tais ataques são caracterizados pelo completo desconhecimento da sua verdadeira origem e visam tornar inacessíveis os serviços providos pela vítima. Este objetivo geralmente é alcançado através do envio de pacotes a uma taxa maior do que podem ser servidos, fazendo com que legítimas requisições ao serviço não sejam atendidas. Em sua versão distribuída (*Distributed* DoS), os pacotes são enviados de diferentes origens e o tráfego agregado

*Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ, FINEP, RNP e FUNTTEL.

gerado é responsável pela total inutilização dos serviços da vítima. Ultimamente, o número de ataques distribuídos a grandes sítios é cada vez mais alarmante, sendo inclusive desenvolvidas pragas digitais específicas para tal finalidade [CERT, 2000]. Embora menos comum, também existem ataques de negação de serviço constituídos pelo envio de um único pacote [CERT, 1996]. Em ambos os casos, os resultados são financeiramente desastrosos [Garber, 2000] e a necessidade de uma solução que identifique a verdadeira origem dos pacotes se torna evidente.

Devido à técnica datagrama usada no protocolo IP, o anonimato do atacante é facilmente mantido, pois é possível injetar pacotes na rede com endereço de origem forjado. Em outras palavras, não existe uma entidade ou um mecanismo responsável pela verificação da autenticidade da fonte. Como toda infra-estrutura de roteamento é baseada exclusivamente no endereço de destino, pacotes com endereço de origem forjado geralmente alcançam a vítima sem dificuldades. Outra característica que permite a execução de ataques anônimos é a ausência de estado nos roteadores. Nenhuma informação relativa aos pacotes roteados é armazenada para consultas futuras. Em conseqüência, o encaminhamento de pacotes não deixa "rastros", tornando impossível deduzir a rota percorrida por um pacote. A identificação da fonte também é dificultada caso ataques indiretos sejam empregados. Tais ataques são caracterizados pelo uso de estações intermediárias entre o atacante e a vítima, de forma a ocultar a sua verdadeira origem.

Uma solução completa para qualquer tipo de ataque de negação de serviço é muito complexa. Assim, este artigo aborda o problema do rastreamento de pacotes IP [Savage et al., 2001] que objetiva identificar as estações que geram diretamente o tráfego de ataque e a sua respectiva rota. Mesmo sem uma identificação completa da fonte, informações parciais sobre a rota que permitam identificar um roteador mais próximo da origem do ataque são úteis para controlar o tráfego desnecessário na rede.

Neste artigo, é introduzida uma nova abordagem para o rastreamento de pacotes IP através de uma generalização aqui proposta para a teoria de filtro de Bloom [Bloom, 1970] e da sua respectiva análise matemática. Tal generalização surge como uma solução natural contra a facilidade de evasão do sistema de rastreamento caso um filtro de Bloom convencional fosse usado. A proposta consiste em usar um filtro de Bloom generalizado embutido no pacote de forma a armazenar de maneira compacta todos os roteadores atravessados. Desta forma, probabilisticamente, é possível rastrear a rota completa percorrida por cada pacote individualmente. Além disso, todo o processo de rastreamento pode ser efetuado após a finalização do ataque e sem nenhuma ajuda de operadores de rede. Até o momento, as propostas existentes que possuem objetivos e resultados equivalentes requerem uma enorme capacidade de armazenamento em dispositivos acoplados aos roteadores [Snoeren et al., 2002], [Li et al., 2004].

O artigo está organizado da seguinte forma. Na Seção 2, são apresentados os trabalhos relacionados ao rastreamento de pacotes IP. Na Seção 3, a nova proposta para o rastreamento é explicada, incluindo a generalização do filtro de Bloom e os algoritmos empregados. Resultados analíticos são apresentados na Seção 4 mostrando a eficácia do método proposto. Por fim, na Seção 5 são relatadas as conclusões retiradas deste trabalho.

## 2. Trabalhos Relacionados

Um método simples para rastrear a verdadeira origem de um pacote é incluir o endereço IP de cada roteador no próprio pacote à medida que ele atravessa a rede. A idéia é bastante semelhante com a opção IP Record Route [Postel, 1981]. Desta forma, cada pacote recebido apresenta toda a rota de ataque embutida em si. Este algoritmo consegue rastrear a origem do ataque a partir de um único pacote, apresentando altíssima escalabilidade e atendendo ao caso de ataques distribuídos. Entretanto, algumas desvantagens

o desqualificam como um algoritmo de rastreamento ideal [Savage et al., 2001]. Em primeiro lugar, o ato de adicionar dados ao pacote durante o processo de roteamento implica processamento adicional. Além disso, o tamanho do pacote varia ao longo da rota, uma vez que cada roteador acrescenta seu endereço IP ao reencaminhá-lo. Entretanto, não há como garantir espaço suficiente no pacote para todos os endereços da rota de ataque. Conseqüentemente, pacotes sendo rastreados podem sofrer fragmentações desnecessárias e causar um maior processamento aos roteadores.

Uma estratégia intuitiva para rastrear a fonte de um tráfego suspeito é chamada depuração de entrada (*input debugging*) [Stone, 2000]. A partir de uma assinatura de ataque, é possível determinar a interface de entrada utilizada pelos pacotes de ataque no roteador mais próximo da vítima e, portanto, o roteador adjacente que os enviou. Esta consulta, feita recursivamente em cada roteador, revela a rota do ataque até o próprio autor ou até a borda de outro provedor de serviços. Neste caso, este segundo provedor necessita ser contatado para dar continuidade ao processo. Embora simples, essa aplicação depende da comunicação entre provedores de serviço para alcançar um sucesso no rastreamento. Além disso, o ataque precisa ser suficientemente longo para permitir o rastreamento completo, não sendo possível realizá-lo após o seu término.

Sob uma outra abordagem, uma técnica de rastreamento baseada em testes de enlaces foi desenvolvida [Burch e Cheswick, 2000]. Seu funcionamento é baseado na observação do fluxo de ataque recebido pela vítima à medida que cada enlace é inundado por grandes rajadas de tráfego. Apesar de não necessitar de nenhuma intervenção por parte dos operadores da rede, esta técnica exige um mapa topológico apurado assim como possíveis nós dispostos a sofrerem curtas inundações. Como na técnica anterior, é possível que as verdadeiras fontes de um ataque nem cheguem a ser reveladas se o ataque for subitamente interrompido no meio do processo de rastreamento. Além disso, a variação observada na vítima para o caso de ataques distribuídos ou ainda para pequenos fluxos pode ser imperceptível.

Em [Savage et al., 2001], é introduzido um sistema baseado em auditoria, onde a informação necessária para o rastreamento é encontrada na vítima. Os roteadores a informam sobre sua presença na rota, usando para isso campos pouco usados do cabeçalho IP. Basicamente, os campos usados para fragmentação são sobrecarregados com a informação sobre a rota. Visando reduzir o processamento no roteador e o espaço necessário em cada pacote, técnicas de codificação e amostragem são empregadas. Posteriormente, a vítima consegue reconstruir a rota tomada por um fluxo de ataque depois de recebido um número suficiente de pacotes deste fluxo. Embora inovadora, a proposta necessita de um alto poder computacional durante a reconstrução da rota de ataque pela vítima e gera diversos falsos positivos mesmo em ataques distribuídos de pequeno porte [Song e Perrig, 2001]. Análises realizadas por [Park e Lee, 2001] mostram ainda a vulnerabilidade do sistema para o caso do atacante forjar as marcações do cabeçalho IP do pacote.

Outro método baseado em auditoria foi proposto por [Bellovin et al., 2003]. Ao rotear um pacote, cada roteador probabilisticamente envia para a vítima um pacote ICMP com informações sobre si mesmo e sobre seus roteadores adjacentes. Para um fluxo suficientemente longo, a vítima usa estes dados recebidos para reconstruir a rota de ataque. Entretanto, como as informações de auditoria são enviadas em pacotes separados, a autenticação das mensagens é necessária de forma a evitar mensagens forjadas pelo atacante. Logo, a adoção de uma infra-estrutura de chave pública se torna inevitável. Em uma extensão desse trabalho [Mankin et al., 2001], novos conceitos como “utilidade” e “valor” das mensagens de rastreamento são introduzidos, assim como uma proposta para melhorá-los.

Sob a perspectiva de armazenamento na própria infra-estrutura de rede, a maneira mais simples de recolher rastros de auditoria é cada roteador registrar todos os pacotes que o atravessam [Stone, 2000]. Porém, recursos excessivos são requisitados tanto para armazenagem quanto para a mineração dos dados. Além disso, a invasão de um roteador acarretaria ainda em problemas de privacidade, uma vez que ele contém informações sobre todos os pacotes roteados.

Uma alternativa para reduzir a armazenagem de um grande volume de informações é utilizar um filtro de Bloom [Bloom, 1970] (Apêndice A). Ultimamente, estes filtros têm sido amplamente usados em redes de computadores [Broder e Mitzenmacher, 2002]. Em [Snoeren et al., 2002], os autores propõem um mecanismo que possui a vantagem de rastrear um único pacote IP que tenha passado na rede sem a necessidade de se armazenar todo o tráfego roteado. Para isso, são usados filtros de Bloom em dispositivos acoplados aos roteadores que armazenam os pacotes roteados de forma compacta. Periodicamente, os filtros saturados são armazenados para futuras requisições e trocados por novos filtros vazios. Para mais tarde determinar se um pacote passou pelo roteador, o seu filtro simplesmente é verificado. Um processo repetitivo pode ser feito por cada roteador para reconstruir o caminho do pacote até a sua verdadeira origem. Porém, mesmo com o uso de filtros de Bloom, tal sistema exige uma alta capacidade de armazenamento. Melhorias propostas em [Li et al., 2004] diminuem drasticamente o espaço necessário para o armazenamento embora a capacidade de se rastrear um único pacote seja comprometida.

## 3. Sistema Proposto para o Rastreamento de Pacotes IP

Esta seção apresenta uma nova técnica para rastreamento de pacotes IP que objetiva rastrear a origem de cada pacote individualmente. Ela se baseia na abordagem de inserção de informações nos pacotes para evitar o armazenamento de estados nos roteadores. Ao invés de usar uma marcação tal qual a proposta por [Savage et al., 2001], cada roteador insere no pacote uma “assinatura” que indica sua presença na rota. A técnica de filtro de Bloom (Apêndice A) é usada para que a quantidade de informação inserida seja reduzida e permaneça constante a fim de evitar a fragmentação dos pacotes. Além disso, propõe-se o uso de uma generalização do filtro de Bloom (Seção 3.1) para evitar que o atacante possa forjar as “assinaturas” e prejudicar o rastreamento.

De forma a diminuir tanto o espaço necessário em cada pacote como o custo de processamento, a rota de ataque é armazenada em um filtro de Bloom embutido em cada pacote. Para isso, um campo fixo é reservado no cabeçalho do pacote para o filtro. O algoritmo de marcação para este caso é bem simples. Pouco antes de reencaminhar um pacote, todo roteador insere no filtro daquele pacote o endereço IP da sua interface de saída. Desta forma, ao receber um pacote de ataque, a vítima dispõe de um filtro cujos elementos são todos os roteadores componentes da rota de ataque.

Para reconstruir esta rota, o seguinte algoritmo é utilizado. Inicialmente, a vítima verifica a presença de todos os seus roteadores vizinhos no filtro do pacote recebido. Aquele que for reconhecido como elemento do filtro é identificado como o roteador pelo qual o pacote chegou e é, portanto, integrado à rota de ataque. Em seguida, este roteador verifica qual dos seus roteadores vizinhos também é reconhecido como elemento do filtro, identificando assim o próximo componente da rota de ataque. Um processo recursivo é então realizado sucessivamente por cada roteador visando reconstruir o caminho do pacote até a sua verdadeira origem.

Algumas vantagens surgem como resultado da adoção dessa abordagem. Primeiramente, a rota completa de cada pacote pode ser determinada individualmente. Tal comportamento é idealizado por todo sistema de rastreamento de pacotes uma vez que possibilita a identificação de qualquer fonte em um ataque distribuído. Logo, uma de suas ca-

racterísticas é a alta escalabilidade. Além disso, nenhum estado necessita ser armazenado na infra-estrutura de rede. Todos os dados relativos ao rastreamento estão localizados na própria vítima, que opta por guardá-los ou não de acordo com a política de segurança local. Outra vantagem é a capacidade de se realizar o rastreamento após o término do ataque e sem ajuda de operadores de rede. Mais especificamente, o tempo restante para se rastrear as fontes depois de um ataque depende exclusivamente dos recursos alocados para o rastreamento na vítima.

Por outro lado, a adoção do filtro de Bloom para representar a rota de ataque introduz uma probabilidade de falso positivo. Durante o algoritmo de reconstrução, um falso positivo implicaria a incorreta integração de um roteador à rota de ataque. Porém, se esta probabilidade for pequena, a ocorrência de falsos positivos não tem nenhum impacto significativo na reconstrução. Algumas rotas para um mesmo pacote existiriam em paralelo, mas ainda assim o escopo de possíveis atacantes seria restringido. No entanto, como o atacante tem controle sobre o conteúdo inicial do pacote, ele pode preencher todos os bits do filtro com 1. Ao saturar o filtro, todo roteador é integrado à rota de ataque durante a reconstrução, tornando impraticável a descoberta da verdadeira rota.

Para minimizar a possibilidade do atacante burlar o sistema e torná-lo menos dependente da condição inicial do vetor, uma generalização do filtro de Bloom é aqui proposta. A idéia básica do filtro generalizado é utilizar tanto funções *hash* que preenchem como funções que zeram bits. Desta forma, é mostrado que a probabilidade de falso positivo diminui e que ela depende pouco da condição inicial. Por outro lado, falsos negativos que eram inexistentes no filtro de Bloom convencional são introduzidos nesta proposta de filtro de Bloom generalizado. A seguir, a idéia da generalização do filtro é formalizada e uma análise das probabilidades de falso negativo e falso positivo é desenvolvida a fim de mostrar a eficácia desta nova abordagem.

### 3.1. O Filtro de Bloom Generalizado

Assim como o filtro convencional, o filtro de Bloom generalizado é uma estrutura de dados usada para representar de forma compacta um conjunto $S = \{s_1, s_2, \ldots, s_n\}$ de $n$ elementos. Ele é constituído por um vetor de $m$ bits e por $k_0 + k_1$ funções *hash* independentes $g_1, g_2, \ldots, g_{k_0}, h_1, h_2, \ldots, h_{k_1}$ cujas saídas variam uniformemente no espaço discreto $\{0, 1, \ldots, m-1\}$. O vetor de bits é calculado de maneira semelhante ao do caso convencional. Entretanto, não há mais a restrição de que seus bits são inicialmente zerados. No filtro generalizado, os bits do vetor podem assumir qualquer valor inicial. Para cada elemento $s_i \in S$, os bits do vetor correspondentes às posições $g_1(s_i), g_2(s_i), \ldots, g_{k_0}(s_i)$ são zerados e os bits correspondentes às posições $h_1(s_i), h_2(s_i), \ldots, h_{k_1}(s_i)$ são preenchidos com 1. No caso de uma colisão entre uma função $g_i$ com uma função $h_j$ em um mesmo elemento, arbitra-se que o bit é zerado $\forall i, j$. O mesmo bit pode ser zerado ou preenchido diversas vezes sem restrições. A Figura 1 ilustra de maneira sucinta a inserção de um elemento no filtro generalizado. Posteriormente, testes de pertinência podem ser realizados visando determinar a afiliação de um elemento ao conjunto. Para verificar se um elemento $x$ pertence a $S$, é preciso checar se os bits $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ estão zerados e se os bits $h_1(x), h_2(x), \ldots, h_{k_1}(x)$ estão preenchidos com 1. Se pelo menos um bit está invertido, então $x \notin S$ com alta probabilidade. Diferentemente do filtro convencional, agora há uma possibilidade de um elemento $x \in S$ não ser reconhecido pelo filtro, criando um falso negativo. Tal anomalia ocorre quando pelo menos um dos bits $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ é preenchido ou quando pelo menos um dos bits $h_1(x), h_2(x), \ldots, h_{k_1}(x)$ é zerado por algum outro elemento inserido posteriormente. Por outro lado, se nenhum bit está invertido, então $x \in S$ também com alta probabilidade. A incerteza se deve à possibilidade do filtro reconhecer como afiliado um elemento externo $x \notin S$, criando um falso positivo. Um falso positivo ocorre quando os bits $g_1(x), g_2(x), \ldots, g_{k_0}(x)$ estão zerados e os

bits $h_1(x), h_2(x), \ldots, h_{k_1}(x)$ estão preenchidos com 1 em virtude de um subconjunto dos elementos de $S$ ou da própria condição inicial do vetor.

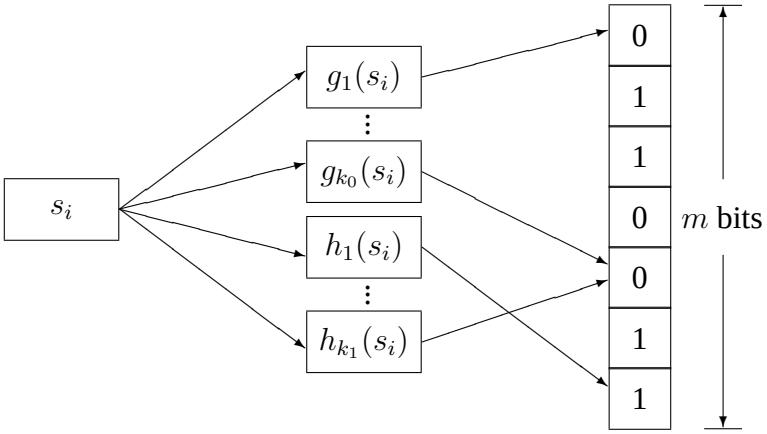



**Figura 1: Inserção de um elemento em um filtro de Bloom generalizado.**

A probabilidade de falso positivo do filtro de Bloom generalizado é calculada de maneira similar à do caso convencional. Dado que no caso de uma colisão as funções $g_i$ sempre têm prioridade sobre as funções $h_j$, a probabilidade $q_0$ de um determinado bit ser zerado por um elemento qualquer é expressa pela probabilidade de pelo menos uma das $k_0$ funções zerar o bit; ou seja,

$$q_0 = \left[1 - \left(1 - \frac{1}{m}\right)^{k_0}\right] \approx \left(1 - e^{-\frac{k_0}{m}}\right). \tag{1}$$

Por outro lado, a probabilidade $q_1$ de um determinado bit ser preenchido por um elemento é a probabilidade de pelo menos uma das $k_1$ funções preencher o bit e nenhuma das $k_0$ funções o zerar; ou seja,

$$q_1 = \left[1 - \left(1 - \frac{1}{m}\right)^{k_1}\right] \left(1 - \frac{1}{m}\right)^{k_0} \approx \left(1 - e^{-\frac{k_1}{m}}\right) e^{-\frac{k_0}{m}}. \tag{2}$$

Por fim, a probabilidade de um bit específico permanecer intocado, isto é, não ser nem preenchido e nem zerado por um elemento, é simplesmente

$$(1 - q_0 - q_1) = \left(1 - \frac{1}{m}\right)^{k_0+k_1} \approx e^{-\frac{k_0+k_1}{m}}. \tag{3}$$

Como o mesmo cálculo se aplica para todos os bits do vetor, na média, uma fração $q_0$ de bits é zerada, uma fração $q_1$ de bits é preenchida e uma fração $(1 - q_0 - q_1)$ de bits permanece intocada a cada inserção de elemento. Seguindo o mesmo raciocínio, tem-se na média $b_0 = m.q_0$ bits zerados, $b_1 = m.q_1$ bits preenchidos e $(m - b_0 - b_1)$ bits intocados a cada inserção.

A partir destes valores, é possível determinar a fração de bits zerados e preenchidos do vetor. A probabilidade $p_0(n)$ de um bit estar em 0 após $n$ inserções é calculada através das probabilidades de $n + 1$ eventos mutuamente exclusivos. O primeiro evento é aquele onde o bit está inicialmente em 0 e permanece intocado pelos $n$ elementos. Notando que $p_0(0)$ representa a probabilidade do bit estar inicialmente em 0, a probabilidade de tal evento é $p_0(0)\,(1 - q_0 - q_1)^n$. Os outros $n$ eventos são aqueles onde o bit é zerado pelo $(n - i)$-ésimo elemento e não é mais tocado pelos $i$ elementos seguintes, para

$0 \leq i \leq n-1$. A probabilidade de ocorrência de cada um destes eventos é expressa por $q_0 \left(1 - q_0 - q_1\right)^i$. Desta forma,

$$\begin{aligned} p_0(n) &= p_0(0)\left(1 - q_0 - q_1\right)^n + \sum_{i=0}^{n-1} q_0 \left(1 - q_0 - q_1\right)^i \\ &= p_0(0) e^{-\frac{(k_0+k_1)n}{m}} + \sum_{i=0}^{n-1} \left(1 - e^{-\frac{k_0}{m}}\right) e^{-\frac{(k_0+k_1)i}{m}} \\ &= p_0(0) e^{-\frac{(k_0+k_1)n}{m}} + \left(1 - e^{-\frac{k_0}{m}}\right) \left(\frac{1 - e^{-\frac{(k_0+k_1)n}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}}\right). \end{aligned} \tag{4}$$

Analogamente, a probabilidade $p_1(n)$ de um bit estar em 1 depois das $n$ inserções é

$$p_1(n) = p_1(0) e^{-\frac{(k_0+k_1)n}{m}} + \left(1 - e^{-\frac{k_1}{m}}\right) e^{-\frac{k_0}{m}} \left(\frac{1 - e^{-\frac{(k_0+k_1)n}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}}\right) \tag{5}$$

e obviamente $p_0(n) + p_1(n) = 1$. Dado que o mesmo cálculo pode ser aplicado para todos os bits do vetor, na média, uma fração $p_0(n)$ dos bits fica em 0 e uma fração $p_1(n)$ dos bits fica em 1 depois de $n$ inserções.

A partir das proporções de bits no vetor, a probabilidade de falso positivo é calculada de maneira trivial. Dado que na média $b_0$ bits são zerados e $b_1$ bits são preenchidos por elemento, a probabilidade de falso positivo do filtro de Bloom generalizado é calculada da seguinte forma

$$f_p = p_0(n)^{b_0} p_1(n)^{b_1} = p_0(n)^{b_0} \left[1 - p_0(n)\right]^{b_1} = \left[1 - p_1(n)\right]^{b_0} p_1(n)^{b_1}. \tag{6}$$

Como esperado, a Equação 6 se reduz à probabilidade do filtro convencional quando seus parâmetros são usados, isto é, $k_0 = 0$, $p_0(0) = 1$ e $p_1(0) = 0$. Neste caso, tem-se $p_0(n) = e^{-k_1 n/m}$, $p_1(n) = 1 - e^{-k_1 n/m}$, $b_0 = 0$ e $b_1 = m(1 - e^{-k_1/m})$. Expandindo a expressão de $b_1$ em série e notando que $m \gg k_1$, pode-se chegar à simplificação realizada para o caso convencional

$$b_1 = m(1 - e^{-k_1/m}) = m\left[1 - \left(1 - \frac{k_1}{m} + \frac{k_1^2}{2m^2} - \cdots\right)\right] \approx k_1. \tag{7}$$

Logo, a probabilidade de falso positivo $f_p$ se reduz ao mesmo resultado da Equação 18, ou seja,

$$f_p = \left(1 - e^{-\frac{k_1 n}{m}}\right)^{k_1}. \tag{8}$$

A probabilidade de falso negativo pode ser calculada se antes for determinada a probabilidade de um bit do $(n-i)$-ésimo elemento inserido não ser invertido pelos $i$ elementos seguintes, para $0 \leq i \leq n-1$. Como nos falsos positivos, a probabilidade $p_{00}(i)$ de um bit zerado pelo $(n-i)$-ésimo elemento permanecer zerado ao fim das $i$ inserções é calculada a partir das probabilidades de $i+1$ eventos mutuamente exclusivos. Desta maneira, ou o bit permanece intocado pelos $i$ elementos seguintes ou é zerado por algum deles e não é mais tocado. Equivalentemente,

$$\begin{aligned} p_{00}(i) &= e^{-\frac{(k_0+k_1)i}{m}} + \sum_{j=0}^{i-1} \left(1 - e^{-\frac{k_0}{m}}\right) e^{-\frac{(k_0+k_1)j}{m}} \\ &= e^{-\frac{(k_0+k_1)i}{m}} + \left(1 - e^{-\frac{k_0}{m}}\right) \left(\frac{1 - e^{-\frac{(k_0+k_1)i}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}}\right). \end{aligned} \tag{9}$$

Analogamente, a probabilidade $p_{11}(i)$ de um bit preenchido pelo $(n-i)$-ésimo elemento permanecer preenchido após as $i$ inserções seguintes é

$$p_{11}(i) = e^{-\frac{(k_0+k_1)i}{m}} + \left(1 - e^{-\frac{k_1}{m}}\right) e^{-\frac{k_0}{m}} \left(\frac{1 - e^{-\frac{(k_0+k_1)i}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}}\right). \quad (10)$$

A partir das Equações 9 e 10, a probabilidade de falso negativo de qualquer elemento inserido no filtro pode ser determinada. A probabilidade de falso negativo $f_n(i)$ do $(n-i)$-ésimo elemento não ser reconhecido pelo filtro é calculada tomando-se o complemento da probabilidade de nenhum de seus bits serem invertidos. Logo,

$$f_n(i) = 1 - p_{00}(i)^{b_0} p_{11}(i)^{b_1}. \quad (11)$$

Como esperado, a Equação 11 se reduz a zero para o caso do filtro de Bloom convencional. Neste caso, $b_0 = 0$ e $p_{11} = 1$ e, independente de outros parâmetros, a probabilidade de falso negativo é zero. Para o último elemento inserido, ou seja, para o $n$-ésimo elemento, esta probabilidade também é zero dado que nenhum outro elemento pode inverter os seus bits. Neste caso, $i = 0$ e $p_{00} = p_{11} = 1$; logo, a probabilidade de falso negativo também se reduz a zero.

No sistema de rastreamento proposto, os parâmetros do filtro de Bloom generalizado são definidos da seguinte forma:

- os $n$ elementos do filtro de Bloom generalizado são os roteadores por onde o pacote passa;
- $m$ é o tamanho (em bits) do vetor alocado no cabeçalho do pacote;
- $k_0$ e $k_1$ são o número de funções *hash* que zeram e preenchem os bits, respectivamente. Estas funções são as mesmas em todos os roteadores;
- $p_0(0)$ e $p_1(0)$ são a fração inicial de bits em 0 e a fração de bits em 1, respectivamente. Este parâmetro é determinado pelo atacante.

## 4. Resultados

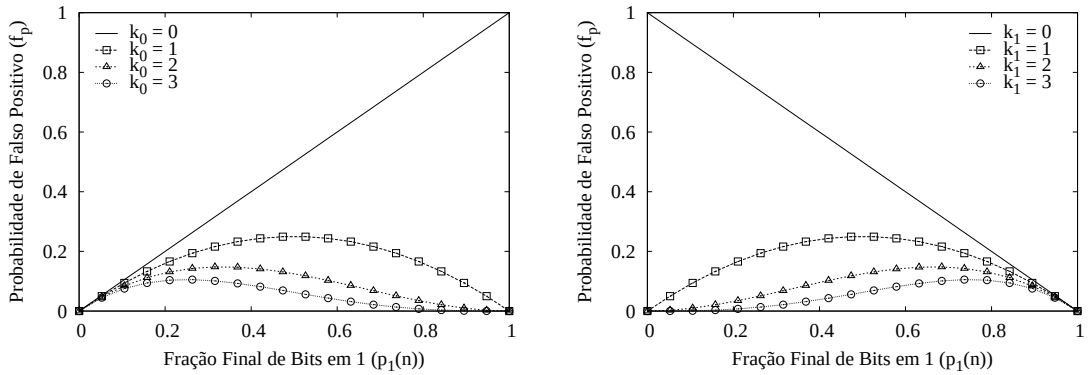
Nesta seção, são comparados analiticamente os dois sistemas de rastreamento propostos neste artigo: a versão simplificada que emprega o filtro de Bloom convencional e a versão estendida que utiliza o novo conceito de filtro de Bloom generalizado. O objetivo é demonstrar a necessidade e as vantagens da utilização do filtro generalizado no lugar do filtro convencional. Para isso, três análises distintas são realizadas. Primeiramente, os falsos positivos de cada sistema são detalhados. Em seguida, os falsos negativos são abordados. Por fim, a interferência do atacante em cada sistema é explicada.

### 4.1. Falsos Positivos

Durante o algoritmo de reconstrução de rota, um falso positivo implica na incorreta integração de um roteador à rota do ataque. À medida que a probabilidade de falso positivo aumenta, mais possíveis rotas para um mesmo pacote existem, dificultando a identificação do atacante. Logo, busca-se diminuir a probabilidade de falso positivo.

A Figura 2 mostra como varia a probabilidade de falso positivo de um filtro de Bloom generalizado $f_p$ em função de $p_1(n)$, de acordo com a Equação 6. A probabilidade $p_1(n)$ também pode ser encarada como a fração de bits do vetor em 1 depois de inseridos os $n$ elementos. Na Figura 2(a), a curva onde $k_0 = 0$ representa o filtro de Bloom convencional. Para este caso, pode-se constatar que a probabilidade de falso positivo aumenta à medida que $p_1(n)$ aumenta. Este resultado está de acordo com a Equação 18. Os outros valores de $k_0$ representam o filtro de Bloom generalizado. Pode-se observar que

tanto para $p_1(n) = 0$ quanto para $p_1(n) = 1$, a probabilidade de falso positivo vale zero. Isso ocorre porque, quando pelo menos uma função de cada tipo é usada, é necessário que existam bits em 0 e bits em 1 para que um falso positivo ocorra. Na Figura 2(b), um comportamento similar é observado. A curva para $k_1 = 0$ representa o dual do filtro de Bloom convencional, com somente funções que zeram bits. Para este filtro, quanto maior a fração de bits em 0, maior é a probabilidade de falso positivo. As outras curvas representam o filtro de Bloom generalizado.




(a) $f_p$ em função de $p_1(n)$, para $k_1 = 1$.

(b) $f_p$ em função de $p_1(n)$, para $k_0 = 1$.

**Figura 2: Probabilidade de falso positivo de um filtro de Bloom generalizado em função da fração final de bits em 1.**

O ponto máximo das Figuras 2(a) e 2(b) pode ser calculado tomando-se a derivada de $f_p$ em relação a $p_1(n)$ e igualando-a a zero. Assumindo $m \gg k_0$ e $m \gg k_1$, os números de bits zerados e preenchidos por elemento podem ser expressos por $b_0 \approx k_0$ e $b_1 \approx k_1$, respectivamente. Assim, a Equação 6 pode ser simplificada por

$$f_p = [1 - p_1(n)]^{k_0} \, p_1(n)^{k_1}. \tag{12}$$

Pode ser mostrado que o ponto máximo da Equação 12 ocorre para

$$p_1(n) = \frac{k_1}{k_0 + k_1}, \tag{13}$$

assumindo $p_1(n) \neq 0$ e $p_1(n) \neq 1$.

Ao substituir a Equação 13 na Equação 12, a probabilidade máxima de falso positivo do filtro de Bloom generalizado $f_p^{max}$ pode ser encontrada e seu valor é

$$f_p^{max} = \left(\frac{k_0}{k_0 + k_1}\right)^{k_0} \left(\frac{k_1}{k_0 + k_1}\right)^{k_1}. \tag{14}$$

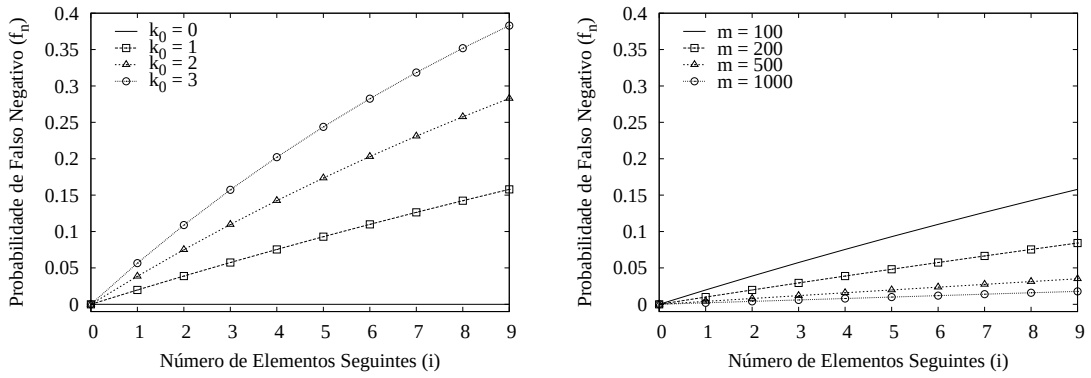
Ao contrário do filtro de Bloom convencional, a probabilidade de falso positivo do filtro generalizado é sempre limitada por $f_p^{max}$. Esta probabilidade máxima é exclusivamente determinada pelos parâmetros $k_0$ e $k_1$ escolhidos para o sistema. Em virtude desta limitação, a interferência da ação do atacante no processo de rastreamento também é restringida, como tratado na Seção 4.3.

### 4.2. Falsos Negativos

Com a adoção do filtro de Bloom generalizado, falsos negativos podem ocorrer durante o processo de reconstrução da rota de ataque. Um falso negativo significa não

detectar um roteador pelo qual o pacote rastreado realmente passou. Somente um falso negativo já é suficiente para interromper precocemente a reconstrução da verdadeira rota de ataque. Logo, a probabilidade de falsos negativos deve necessariamente ser mantida baixa. Vale ressaltar que o atacante não interfere na probabilidade de falso negativo, de acordo com as Equações 9, 10 e 11.

A Figura 3 mostra a probabilidade de falso negativo para cada elemento $(n - i)$, $0 \leq i \leq n - 1$, de acordo com a Equação 11. No sistema de rastreamento, o $n$-ésimo elemento ($i = 0$) corresponde ao roteador mais próximo da vítima e o primeiro elemento ($i = n - 1$) corresponde ao roteador mais perto do atacante. Na Figura 3(a), a curva onde $k_0 = 0$ representa o filtro de Bloom convencional. Conforme esperado, os falsos negativos para este caso são sempre zero independentemente do roteador. As outras curvas destas figuras representam um filtro de Bloom generalizado e verifica-se que à medida que o roteador está mais afastado da vítima, maior é a probabilidade daquele roteador ser um falso negativo. Isso acontece porque quanto mais roteadores são inseridos depois de um determinado roteador $(n-i)$, maior é a probabilidade de algum dos $i$ roteadores seguintes inverter algum bit do roteador $(n - i)$.




(a) $n = 10, m = 100, k_1 = 1$.

(b) $n = 10, k_0 = k_1 = 1$.

**Figura 3: Probabilidade de falso negativo do (n-i)-ésimo elemento do filtro de Bloom generalizado.**

Ao se incremetar $k_0$, a probabilidade de falso negativo aumenta rápido como mostra a Figura 3(a). Um comportamento semelhante é observado ao fixar $k_0 = 1$ e alterar $k_1$ (não mostrado). Por outro lado, um aumento em $m$ diminui a probabilidade de falso negativo dos roteadores. Intuitivamente, variações em $k_0$ e $k_1$ têm efeito oposto a variações em $m$. À medida que mais funções são usadas, maior é a chance dos bits de cada elemento serem invertidos pelos elementos seguintes. Entretanto, quanto maior é $m$, maior é o espaço de saída das funções e, portanto, a probabilidade de um bit ser invertido diminui.

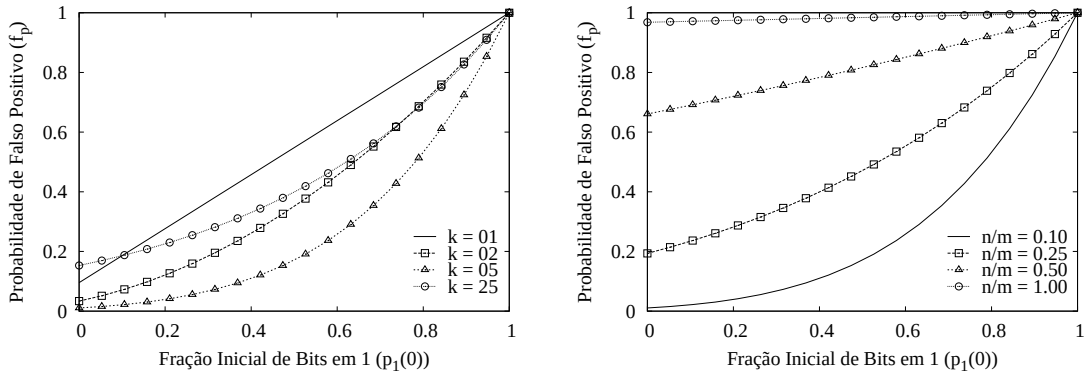
Os falsos negativos são uma desvantagem para o uso de um filtro generalizado. Para o caso onde $k_0 = 1$, $k_1 = 1$, $n = 10$ e $m = 100$, a probabilidade de falso negativo do primeiro elemento ($i = 9$) chega a 15,8%.

### 4.3. Interferência do Atacante

Tanto no sistema que utiliza o filtro de Bloom convencional quanto no que usa o filtro generalizado, o atacante pode interferir no processo de rastreamento, pois o filtro está embutido no pacote. Assim, o atacante pode determinar a proporção inicial de bits em 0 e bits em 1 do filtro de forma a interferir na probabilidade de falso positivo.

Primeiramente, o sistema que emprega um filtro convencional é abordado. A Figura 4 mostra a dependência da probabilidade de falso positivo de um filtro convencional

em relação à sua condição inicial. Como visto na Figura 4(a), para qualquer número $k$ de funções *hash* escolhido, a probabilidade de falso positivo $f_p$ sempre tende a 1 à medida que mais bits são inicialmente preenchidos. Devido ao compromisso entre $f_p$ e $k$ já citado anteriormente, pode-se perceber que a probabilidade de falso positivo é menor para alguns valores de $k$ e maior para outros, mas em todos os casos o comportamento é o mesmo. A Figura 4(b) mostra um comportamento semelhante, porém variando-se a relação $n/m$. Enfim, a probabilidade de falso positivo pode atingir 100% quando o atacante simplesmente preenche todos os bits do filtro com 1.




(a) $n/m = 0.1$ (b) $k = 5$

**Figura 4: Probabilidade de falso positivo de um filtro de Bloom convencional em função da fração inicial de bits em 1.**

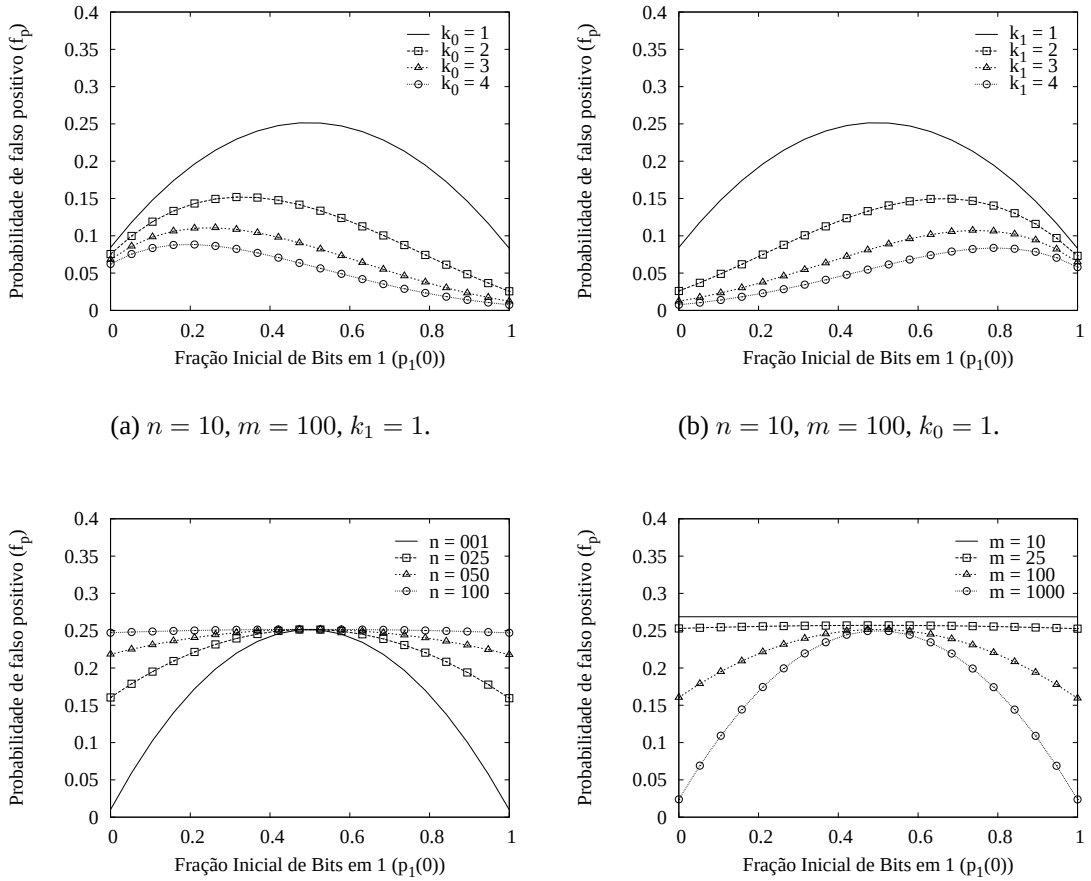
Sendo utilizado um filtro de Bloom generalizado no pacote ao invés de um filtro convencional, a interferência do atacante diminui consideravelmente. A Figura 5 mostra a probabilidade de falso positivo do filtro de Bloom generalizado em função da fração inicial de bits em 1, para variações dos seus parâmetros. Pode ser mostrado que o valor de $p_1(0)$ que maximiza $f_p$ é $k_1/(k_0+k_1)$, ou seja, o mesmo valor apresentado na Equação 13. As Figuras 5(a) e 5(b) comprovam este resultado. A Figura 5(c) mostra que à medida que mais elementos são inseridos no filtro, a probabilidade de falso positivo depende menos da condição inicial. Este resultado pode ser obtido ao perceber que as parcelas dependentes da condição inicial das Equações 4 e 5 tendem a zero à medida que mais elementos são inseridos. Além disso, a probabilidade de falso positivo tende ao valor máximo devido à proporção de bits em 0 e bits em 1 tenderem a $k_0/(k_0+k_1)$ e $k_1/(k_0+k_1)$, respectivamente, à medida que $n$ aumenta, como mostram as Equações 15 e 16:

$$\lim_{n\to\infty} p_0(n) = \frac{1 - e^{-\frac{k_0}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}} \approx \frac{k_0}{k_0 + k_1}, \tag{15}$$

$$\lim_{n\to\infty} p_1(n) = \frac{\left(1 - e^{-\frac{k_1}{m}}\right) e^{-\frac{k_0}{m}}}{1 - e^{-\frac{k_0+k_1}{m}}} \approx \frac{k_1}{k_0 + k_1}. \tag{16}$$

A Figura 5(d) comprova o resultado da Equação 14. Dado que $m \gg k_0$ e $m \gg k_1$, aumentos no tamanho do vetor não influenciam na probabilidade máxima de falso positivo. Somente para resultados onde $m$ não é muito maior que $k_0$ e $k_1$ é que a probabilidade de falso positivo aumenta ligeiramente. Entretanto, os falsos negativos diminuem consideravelmente com aumentos em $m$, conforme mostrado na Figura 3(b).

Assim, com a adoção de um filtro de Bloom generalizado ao invés de um filtro de Bloom convencional, a probabilidade máxima de falso positivo cai no mínimo 75%, caso onde $k_0 = k_1 = 1$.




(a) $n = 10$, $m = 100$, $k_1 = 1$.

(b) $n = 10$, $m = 100$, $k_0 = 1$.

(c) $k_0 = k_1 = 1$, $m = 100$.

(d) $k_0 = k_1 = 1$, $n = 25$.

**Figura 5: Probabilidade de falso positivo de um filtro de Bloom generalizado em função da fração inicial de bits em 1.**

## 5. Conclusões

Neste artigo, é introduzida uma nova abordagem para o rastreamento de pacotes IP que insere dados no cabeçalho dos próprios pacotes. Probabilisticamente, o sistema proposto é capaz de descobrir a origem de um ataque através do rastreamento de um único pacote. Ao atravessar a rede, o pacote é marcado por cada roteador pertecente a sua rota com uma "assinatura". Dessa forma, a vítima consegue identificar todos os roteadores componentes da rota de ataque. O sistema proposto usa um filtro de Bloom no cabeçalho de cada pacote para que a informação inserida seja compacta e apresente um tamanho fixo, evitando assim possíveis fragmentações do pacote. Entretanto, o filtro de Bloom é altamente dependente da sua condição inicial. É mostrado que, ao se usar um filtro de Bloom no pacote para armazenar toda a rota de ataque, o atacante facilmente consegue dificultar o processo de rastreamento. Por isso, uma generalização do filtro de Bloom foi proposta e empregada no sistema. O sistema possui a característica única de ser resistente a burla do atacante.

Os dois sistemas de rastreamento propostos foram comparados analiticamente a fim de comprovar a eficácia do filtro de Bloom generalizado. Um importante resultado mostra que a capacidade do atacante de dificultar o processo de rastreamento é drasticamente diminuída. Enquanto para o filtro de Bloom convencional o atacante consegue atingir uma probabilidade máxima de falso positivo de 100%, para o filtro de Bloom generalizado, a probabilidade máxima de falso positivo depende exclusivamente de parâmetros escolhidos no projeto do sistema e é no máximo 25%. Ou seja, a probabilidade de

falso positivo máxima é diminuída de pelo menos 75%. Em contrapartida, a possibilidade da ocorrência de falsos negativos é introduzida no sistema. Contudo, novos algoritmos podem ser introduzidos no processo de reconstrução de rotas de maneira a minimizar a probabilidade de falso negativo.

## Referências


Bellovin, S. M., Leech, M. D. e Taylor, T. (2003). ICMP Traceback Messages. *Internet Draft: draft-ietf-itrace-04.txt*.

Bloom, B. H. (1970). Space-time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 7(13):442–426.

Broder, A. e Mitzenmacher, M. (2002). Network Applications of Bloom Filters: A Survey. Relatório técnico, Harvard University, Div. of Engineering and Applied Sciences.

Burch, H. e Cheswick, B. (2000). Tracing Anonymous Packets to their Approximate Source. Em *USENIX LISA'00*, páginas 319–327, New Orleans, Lousiana, EUA.

CERT (1996). *CERT Advisory CA-1996-26 Denial-of-Service Attack via ping*.

CERT (2000). *CERT Advisory CA-2000-01 Denial-of-Service Developments*.

CERT (2003). *CERT Advisory CA-2003-20 W32/Blaster worm*.

Fan, L., Cao, P., Almeida, J. e Broder, A. Z. (2000). Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293.

Garber, L. (2000). Denial-of-Service Attacks Rip the Internet. *IEEE Computer*, 4(33):12–17.

Li, J., Sung, M., Xu, J. e Li, L. (2004). Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation. Em *Proceedings of the 25th IEEE Symposium on Security and Privacy*, Oakland, Califórnia, EUA.

Mankin, A., Massey, D., Wu, C.-L., Wu, S. F. e Zhang, L. (2001). On Design and Evaluation of "Intention-Driven" ICMP Traceback. Em *Proceedings of the IEEE ICCCN 2001 Conference*, Scottsdale, Arizona, EUA.

Mitzenmacher, M. (2002). Compressed Bloom Filters. *IEEE/ACM Transactions on Networking*, 10(5):604–612.

Park, K. e Lee, H. (2001). On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack. Em *Proceedings of the IEEE INFOCOM 2001 Conference*, Anchorage, Alaska, EUA.

Postel, J. (1981). Internet Protocol. *RFC 791*.

Savage, S., Wetherall, D., Karlin, A. e Anderson, T. (2001). Network Support for IP Traceback. *IEEE/ACM Transactions on Networking*, 9(3):226–237.

Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Schwartz, B., Kent, S. T. e Strayer, W. T. (2002). Single-Packet IP Traceback. *IEEE/ACM Transactions on Networking*, 10(6):721–734.

Song, D. X. e Perrig, A. (2001). Advanced and Authenticated Marking Schemes for IP Traceback. Em *Proceedings of the IEEE INFOCOM 2001 Conference*, Anchorage, Alaska, EUA.

Stone, R. (2000). CenterTrack: An IP Overlay Network for Tracking DoS Floods. Em *9th USENIX Security Symposium*, páginas 199–212, Denver, Colorado, EUA.

## A. O Filtro de Bloom

Neste apêndice, o filtro de Bloom é explicado seguindo a análise matemática realizada por [Fan et al., 2000] e [Mitzenmacher, 2002]. Sua explicação é fundamental para o completo entendimento do algoritmo de rastreamento proposto na Seção 3.

O filtro de Bloom [Bloom, 1970] é uma estrutura de dados usada para representar de forma compacta um conjunto $S = \{s_1, s_2, \ldots, s_n\}$ de $n$ elementos. Ele é constituído por um vetor de $m$ bits e por $k$ funções *hash* independentes $h_1, h_2, \ldots, h_k$ cujas saídas variam uniformemente no espaço discreto $\{0, 1, \ldots, m-1\}$. O vetor de bits é obtido da seguinte forma. Inicialmente, todos os seus bits encontram-se zerados. Para cada elemento $s_i \in S$, os bits do vetor correspondentes às posições $h_1(s_i), h_2(s_i), \ldots, h_k(s_i)$ são preenchidos com 1. O mesmo bit pode ser preenchido com 1 diversas vezes sem restrições. A Figura 6 ilustra de maneira sucinta a inserção de um elemento no filtro. Uma vez que o filtro de Bloom é uma forma compacta de representar um conjunto de elementos, testes de pertinência podem ser realizados visando determinar se um elemento $x$ pertence ou não ao conjunto $S$. Para isso, verifica-se se os bits do vetor correspondentes às posições $h_1(x), h_2(x), \ldots, h_k(x)$ estão preenchidos com 1. Se pelo menos um bit estiver zerado, então com certeza $x \notin S$. Por outro lado, se todos os bits estão preenchidos, então $x \in S$ com alta probabilidade. A incerteza se deve à possibilidade do filtro reconhecer como afiliado um elemento externo $x \notin S$, criando um falso positivo. Tal anomalia ocorre quando todos os bits $h_1(x), h_2(x), \ldots, h_k(x)$ são preenchidos por um subconjunto dos elementos de $S$.

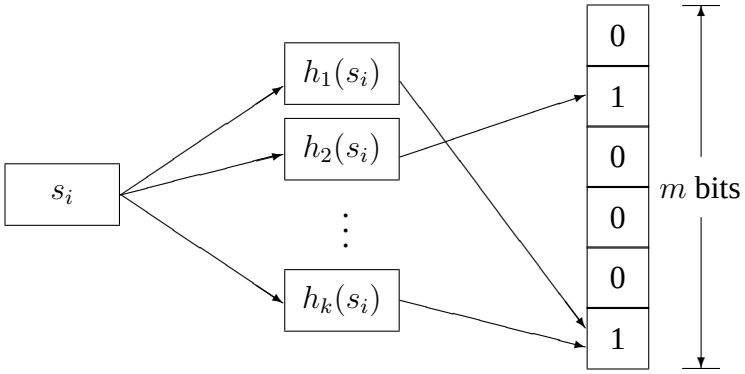



**Figura 6: Inserção de um elemento em um filtro de Bloom.**

A probabilidade de se encontrar um falso positivo para um elemento fora do conjunto é calculada de maneira trivial. Dado que as funções *hash* usadas são uniformes e independentes, a probabilidade de um determinado bit permanecer em zero mesmo depois de inseridos os $n$ elementos é

$$\left(1 - \frac{1}{m}\right)^{kn} \approx e^{-\frac{kn}{m}}. \tag{17}$$

Como o mesmo cálculo se aplica para todos os bits do vetor, na média, uma fração $e^{-kn/m}$ dos bits permanece zerada após as inserções [Mitzenmacher, 2002]. A probabilidade de falso positivo $f_p$ é então a probabilidade de se encontrar um bit em 1 para cada uma das $k$ posições indicadas, ou seja

$$f_p = \left(1 - e^{-\frac{kn}{m}}\right)^k. \tag{18}$$

É importante mencionar que na Equação 18 é assumido que o número médio de colisões por elemento é próximo de zero, comportamento válido quando $m \gg k$. Caso esta condição não seja satisfeita, também é preciso levar em consideração as probabilidades de falso positivo para os casos de $1, 2, \ldots, k-1$ colisões.