Um Mecanismo de Proteção Contra a Previsibilidade de Informações em Pacotes

Bruno G. D'Ambrosio, Eduardo F. de Souza, Paulo André da S. Gonçalves

Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE)
50.740-540 – Recife – PE – Brasil

{bgda, efs, pasg}@cin.ufpe.br

Abstract. *The information predictability from some types of encrypted packets that are captured by malicious entities may present a significant risk to the security of the networks. Recently, many attacks that exploit the knowledge of the size and the plaintext of encrypted packets were developed. These attacks are mainly exploited against IEEE 802.11 networks. This paper proposes a mechanism, namely EPP, which eliminates these vulnerabilities by randomly inserting bytes into the packets before encryption. The proposed solution uses a HMAC (Hash-based Message Authentication), which combines a cryptographic hash function with a secret key. The EPP has manageable overhead, which makes it capable of giving more protection to smaller and more vulnerable packets than to bigger ones, minimizing its impact on the network traffic.*

Resumo. *A previsibilidade de informações em alguns tipos de pacotes criptografados que são capturados por entidades maliciosas pode oferecer riscos significativos à segurança das redes. Recentemente, diversos ataques que se utilizam de pacotes criptografados com tamanho e texto-plano conhecidos foram desenvolvidos. Esses ataques são explorados principalmente em redes IEEE 802.11. Este artigo propõe um mecanismo, denominado EPP, que elimina essas vulnerabilidades inserindo, de forma aleatória, bytes nos pacotes antes dos mesmos serem criptografados pelo protocolo de segurança. A solução proposta utiliza um algoritmo HMAC (Hash-based Message Authentication) que combina uma função hash criptográfica com uma chave secreta. O EPP possui overhead controlável, podendo inserir mais proteção em pacotes pequenos e mais vulneráveis do que em pacotes grandes, minimizando seu impacto no tráfego da rede.*

1. Introdução

Um crescimento expressivo na utilização de recursos interligados em redes vem sendo observado nos últimos anos. Juntamente com esse crescimento, também vem sendo observado um aumento no número de ataques e de vulnerabilidades reportadas em relação aos mecanismos que visam garantir segurança às redes, incluindo segurança ao tráfego transportado por elas. Assim sendo, se faz necessário um constante aprimoramento dos mecanismos de segurança, evitando que os dados que trafegam na rede possam ser indevidamente decifrados ou adulterados.

A previsibilidade do texto-plano de informações criptografadas sendo transportadas em pacotes pela rede pode fornecer a uma entidade maliciosa informações

necessárias para a criação de diversos ataques quando tal entidade possui acesso a esses pacotes. O reconhecimento de pacotes criptografados que podem ser usados em ataques é, em geral, baseado no tamanho do pacote e na associação desse tamanho ao conteúdo em texto-plano que o mesmo potencialmente carregava antes de ser criptografado.

Recentemente, diversos ataques baseados em pacotes criptografados com tamanho e texto-plano conhecidos foram desenvolvidos [Tews et al. 2007], [Beck and Tews 2009], [Ohigashi and Morii 2009] e [Paterson et al. 2009]. Analisando-se esses trabalhos, observa-se que tanto protocolos de segurança que atuam em meio guiado quanto em meio sem fio são afetados. As vulnerabilidades são exploradas em decorrência de problemas inerentes aos algoritmos de criptografia adotados e de características dos pacotes que trafegam na rede. Este artigo propõe um mecanismo, denominado EPP (*Eliminador de Previsibilidade de Pacotes*), o qual visa solucionar as vulnerabilidades decorrentes da previsibilidade do tamanho e do texto-plano de pacotes criptografados. O mecanismo proposto atua antes do pacote ser criptografado pelo protocolo de segurança e utiliza um algoritmo criptográfico do tipo HMAC (*Hash-based Message Authentication Code*) para inserir, aleatoriamente, informações no pacote, alterando suas características previsíveis de tamanho e conteúdo em texto-plano. Ao que tudo indica, este é o primeiro trabalho que se propõe a resolver o problema em questão e que, adicionalmente, explora o uso de uma técnica criptográfica complementar ao protocolo de segurança que protege o tráfego da rede.

O restante deste artigo está organizado da seguinte forma: a Seção 2 descreve conceitos básicos relativos aos protocolos de segurança. A Seção 3 apresenta os problemas relacionados com a previsibilidade do texto-plano de pacotes criptografados e consequentes vulnerabilidades. A Seção 4 descreve o mecanismo proposto neste trabalho e sua aplicação com os protocolos de segurança de redes IEEE 802.11. A Seção 5 discute a segurança do mecanismo proposto e seu *overhead*. Finalmente, a Seção 6 apresenta as conclusões deste trabalho.

2. Protocolos de Segurança

Os algoritmos de criptografia são cruciais para a garantia da confidencialidade de informações trocadas utilizando-se os mais diversos protocolos de comunicação. A necessidade de algoritmos de criptografia que possuam uma grande confiabilidade se torna ainda maior em ambientes sem um meio guiado, visto que, nesses casos, os pacotes que trafegam na rede podem ser capturados facilmente. Existem basicamente dois tipos de algoritmos de criptografia: os algoritmos de cifra de fluxo e os algoritmos de cifra por blocos. Os algoritmos de cifra de fluxo realizam a cifragem dos dados *bit* a *bit* enquanto os de cifra por bloco realizam a cifragem de blocos de *bits* de tamanho fixo.

Em redes IEEE 802.11, os principais protocolos de segurança utilizados atualmente para a proteção do tráfego de dados são o WPA (*Wi-Fi Protected Access*) [Wi-Fi Alliance 2003] e WPA2 ou IEEE 802.11i [Kerry et al. 2004]. Tais protocolos ainda são considerados seguros devido às condições requeridas para a execução dos ataques existentes atualmente contra eles. Embora o protocolo WEP (*Wired Equivalent Privacy*) [Hayes et al. 1999] também tenha como objetivo a proteção do tráfego de dados em redes IEEE 802.11, seu uso não é recomendado devido a sua longa lista de vulnerabilidades [Beck and Tews 2009],[Tews et al. 2007].

O protocolo WPA utiliza como algoritmo de criptografia o RC4, sendo este um algoritmo de cifra de fluxo que é utilizado também por outros protocolos de segurança como o WEP, o SSL (*Secure Sockets Layer*) e o RDP (*Remote Desktop Protocol*). O RC4 não é considerado um algoritmo com nível de segurança elevado [Fluhrer et al. 2001], principalmente levando-se em consideração sua implantação em ambientes de redes sem fio, nas quais os dados podem ser facilmente capturados. Além de não prover um alto grau de segurança, o RC4 também sofre de uma vulnerabilidade inerente a todos os algoritmos de cifra de fluxo: o tamanho dos pacotes cifrados com esse algoritmo releva às entidades maliciosas o tamanho real dos pacotes correspondentes em texto-plano.

O padrão IEEE 802.11i especifica a utilização do AES (*Advanced Encryption Standard*) [Daemen and Rijmen 1998] como algoritmo de criptografia. O AES utiliza cifra por blocos e é considerado mais seguro e adequado para redes sem fio quando comparado ao RC4. Os algoritmos de cifra por bloco são mais seguros do que os algoritmos de cifra de fluxo no que diz respeito à dificuldade de se conhecer o tamanho original do pacote em texto-plano a partir do pacote criptografado. Isso é devido ao fracionamento das mensagens em blocos. No AES, cada bloco corresponde a 128 *bits*. A última fração da mensagem é preenchida com tantos *bits* quantos forem necessários para se completar o tamanho do bloco. Consequentemente, o tamanho do texto-cifrado será o menor múltiplo do tamanho do bloco que seja maior ou igual ao tamanho original do texto-plano.

3. Previsibilidade de Informações e Vulnerabilidades Associadas

Existem diversos exemplos na literatura de ataques que foram criados com base na previsibilidade de informações de determinados pacotes que trafegam criptografados [Tews et al. 2007], [Beck and Tews 2009], [Ohigashi and Morii 2009], [Paterson et al. 2009]. Geralmente, os ataques que utilizam esse tipo de técnica são baseados em pacotes de tamanho reduzido, como pacotes do TCP (*e.g SYN, ACK, RST e FIN*), pacotes ARP (*Address Resolution Protocol*) e pacotes DNS (*Domain Name System*). Em particular, esta seção motiva o problema em questão, apresentando como os ataques existentes para redes IEEE 802.11 utilizam a previsibilidade de tamanho e conteúdo de pacotes do tipo ARP.

3.1. Estrutura de Pacotes do Tipo ARP

A Figura 1 ilustra a estrutura dos pacotes do tipo ARP. Nessa estrutura existem campos de tamanho fixo e variáveis conforme descrito a seguir:

- **Tipo de Hardware** - Campo de 2 bytes que define o tipo de rede na qual o ARP é utilizado. Por exemplo, tipo 1 corresponde ao *Ethernet*;
- **Tipo de Protocolo** - Campo de 2 bytes que especifica o protocolo de rede. Por exemplo, o IPv4;
- **Tamanho do Endereço de Hardware** - Campo de 1 byte que especifica o tamanho do endereço físico;
- **Tamanho do Endereço de Protocolo** - Campo de 1 bytes que especifica o tamanho do endereço do protocolo de rede;
- **Opcode** - Campo de 2 bytes que especifica o tipo do pacote (*e.g.* requisição, resposta);
- **Endereço Físico de Origem** - Campo de comprimento variável que define o endereço físico do emissor. Em redes *Ethernet*, esse campo possui 6 bytes;

- **Endereço Lógico de Origem** - Campo de comprimento variável que define o endereço lógico do emissor. Sob o IPv4, esse campo possui 4 bytes;
- **Endereço Físico de Destino** - Campo de comprimento variável que define o endereço físico do destino. Em requisições ARPs, esse campo é preenchido com zeros já que se desconhece o endereço físico do destino;
- **Endereço Lógico de Destino** - Campo de comprimento variável que define o endereço lógico do destino.

Em particular, nas redes *Ethernet* com o uso do IPv4, o tamanho do pacote ARP é de 28 bytes.

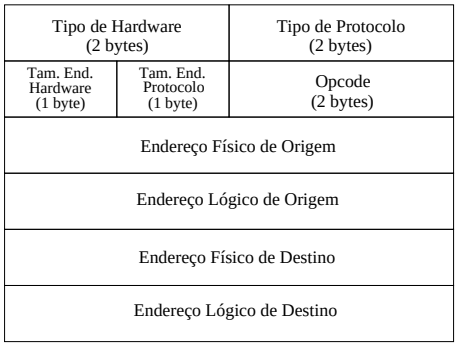

Figura 1. Estrutura de um Pacote ARP

3.2. Vulnerabilidades em Redes IEEE 802.11

Em [Tews et al. 2007] é apresentado o ataque mais recente contra redes IEEE 802.11 protegidas pelo WEP. Esse ataque explora a previsibilidade do conteúdo e tamanho de pacotes ARP. Para funcionar, o ataque precisa obter uma quantidade suficiente de *keystreams* do RC4 utilizados na criptografia de pacotes de interesse. Nesse ataque, os pacotes de interesse são pacotes do tipo ARP. Supondo o uso do IPv4 sobre uma rede IEEE 802.11, os primeiros 16 bytes em texto-plano de um pacote ARP são formados por 8 *bytes* fixos do cabeçalho LLC/SNAP 802.11 (AA:AA03:00:00:00:08:06) seguidos de 8 *bytes* específicos à mensagem ARP. Esses *bytes* também são fixos e iguais a (00:01:08:00:06:04:00:01) para resquições ARP e iguais a (00:01:08:00:06:04:00:02) para respostas ARP. A distinção entre pacotes ARPs e outros pacotes é feita pelo tamanho do quadro capturado. Tanto requisições quanto respostas ARP possuem tamanho pequeno e fixo.

A distinção entre requisições e respostas ARP é simples, pois as requisições são sempre enviadas para o endereço físico de *broadcast* da rede enquanto as respostas são enviadas para um endereço físico *unicast*. Tanto o endereço físico de origem quanto o de destino não são criptografados em redes IEEE 802.11, possibilitando a fácil identificação de um endereço *unicast* ou *broadcast*. Para recuperar os 16 primeiros *bytes* do *keystream* usado para cifrar o pacote ARP, realiza-se um XOR entre o ARP criptografado que foi capturado e o seu padrão inicial fixo de *bytes* em texto-plano. Ao se capturar um número suficiente de ARPs, o ataque consegue encontrar a chave WEP que protege a rede.

O reconhecimento de pacotes do tipo ARP criptogafados também é possível mesmo que o protocolo WPA seja utilizado para proteger a rede sem fio. Isso ocorre, pois conforme mencionado, o RC4 também é utilizado no WPA e, portanto, não há

modificações no tamanho dos pacotes quando criptografados. Isso permitiu recentemente a criação do primeiro ataque prático contra o WPA que ficou conhecido como ataque *Beck-Tews* [Beck and Tews 2009].

O ataque *Beck-Tews* realiza inicialmente a captura de pacotes ARP criptografados. Como os endereços físicos nos quadros não são cifrados e um quadro contendo um pacote ARP é facilmente reconhecido pelo seu tamanho, uma parte significativa do conteúdo dos pacotes ARP torna-se previamente conhecido pelo atacante. Para decifrar o restante do pacote, é realizado um ataque do tipo *chopchop* [KoreK 2004] que foi originalmente inventado para se descobrir o texto-plano de pacotes criptografados pelo WEP sem a necessidade da chave criptográfica.

O ataque *Beck-Tews* leva em consideração as medidas de prevenção ao *chopchop* inseridas no WPA: renegociação imediata das chaves no caso de ocorrerem dois erros de MIC (*Message Integrity Check*) no intervalo de 60 segundos; e descarte do quadro, caso o número de sequência TSC (*TKIP Sequence Counter*) seja menor que o TSC atual do canal [Kerry et al. 2004]. Após a obtenção de todo o texto-plano do pacote ARP e dos campos de verificação de integridade criptografados do quadro que carrega o ARP, o atacante é capaz de inverter o algoritmo de integridade *Michael*, levando-o a obtenção da chave de integridade MIC. De posse dessa chave, o atacante tem a possibilidade de enviar aos clientes da rede um determinado número de pacotes criptografados que serão considerados legítimos e íntegros por tais clientes. Dessa forma, explorando o conteúdo desses pacotes, é possível a criação de novos ataques.

Recentemente, Ohigashi e Morii [Ohigashi and Morii 2009] estenderam o ataque *Beck-Tews*. Nesse novo ataque, o atacante se posiciona fisicamente em um local onde possa se comunicar com cliente e o ponto de acesso, mas sem que tais entidades consigam se comunicar diretamente (ataque *man-in-the-middle*). Em seguida, o atacante executa um ataque *Beck-Tews* modificado. Esse ataque melhora o ataque *Beck-Tews* original ao eliminar o requisito da rede estar utilizando Qualidade de Serviço (IEEE 802.11e) no momento do ataque. Tanto esse novo ataque quanto o ataque *Beck-Tews* original pode explorar adicionalmente a previsibilidade de tamanho e conteúdo de pacotes DNS.

4. O Mecanismo Proposto e Seu Uso com Outros Protocolos

Esta seção apresenta o mecanismo de segurança proposto contra a previsibilidade de conteúdo e tamanho de pacotes e mostra como o mesmo pode ser implantado em redes IEEE 802.11 protegidas pelos protocolos WPA e IEEE 802.11i. Esse mecanismo é denominado EPP (*Eliminador de Previsibilidade de Pacotes*) e sua ideia básica é atuar antes do pacote ser criptografado, inserindo informações que alteram as características previsíveis do pacote e, em consequência, modificam o tamanho original do mesmo. A solução utiliza um algoritmo do tipo HMAC (*Hash-based Message Authentication Code*) [Krawczyk et al. 1997] cujos principais conceitos são apresentados na seção que segue.

4.1. HMAC - *Hash-based Message Authentication Code*

HMAC é uma classe algoritmos criptográficos utilizados originalmente para gerar códigos de autenticação de mensagens ou MACs (*Message Authentication Codes*). Esses códigos são utilizados para a verificação de integridade e autenticidade de pacotes. Esse tipo de

algoritmo combina uma função *hash* resistente à colisão com uma chave secreta para gerar os MACs. O grau de segurança desse tipo de algoritmo está diretamente relacionado à eficácia da função *hash* e à força da chave secreta utilizadas.

As funções *hash* resistentes à colisão mais utilizadas nos algoritmos HMAC são a SHA-1 e a MD5, as quais são consideradas bastante seguras atualmente. Essas funções recebem um bloco de tamanho fixo como entrada e quebram esse bloco em partes menores de mesmo tamanho. Essas partes são iteradas através de uma função de compressão, gerando uma saída pseudo-aleatória de tamanho fixo. O tamanho da saída é de 160 *bits* na SHA-1 e de 128 *bits* na MD5.

O algoritmo HMAC-SHA-1 é utilizado em diversos protocolos de segurança, como no TLS (*Transport Layer Security*), no IPSec (*IP Security Protocol*), no WPA e no IEEE 802.11i. No caso específico dos protocolos de segurança WPA e IEEE 802.11i de redes IEEE 802.11, esse algoritmo é utilizado para realizar a derivação de chaves de segurança durante o *4-Way Handshake* [Kerry et al. 2004],[Wi-Fi Alliance 2003].

4.2. O Mecanismo EPP

O EPP modifica o tamanho e conteúdo dos pacotes antes dos mesmos serem criptografados. Para isso, ele insere uma quantidade aleatória de *bytes* em posições também aleatórias no conteúdo dos pacotes. O mecanismo proposto é composto de dois procedimentos: procedimento de inserção de *bytes* e procedimento de remoção de *bytes*. Esses procedimentos serão detalhados nas próximas seções.

4.2.1. Procedimento de Inserção

Como mostrado na Figura 2, o procedimento de inserção do EPP é composto por dois módulos: *módulo gerador* e *módulo montador*. O módulo gerador define a quantidade de *bytes* (*dummy bytes*) a ser inserida e a posição na qual cada um desses *bytes* deve ser inserido no pacote. O módulo montador insere os *dummy bytes* nas posições corretas e repassa o pacote modificado para protocolo de segurança.

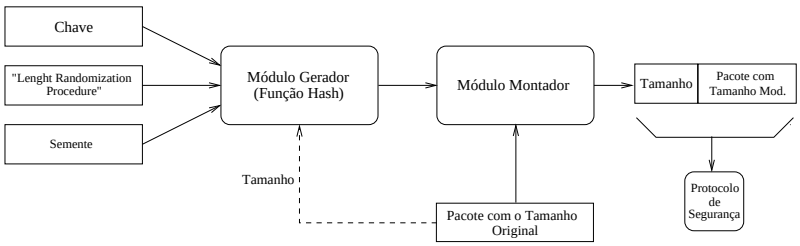

Figura 2. Procedimento de Inserção

No módulo gerador está presente um algoritmo HMAC que recebe como entradas: uma chave de segurança, cuja origem varia de acordo com o protocolo utilizado; uma *string* de diferenciação com o nome do procedimento (“*Length Randomization Procedure*”); o tamanho do pacote; e uma semente, que deve variar a cada pacote. A semente tem por objetivo fazer com que a saída do algoritmo seja sempre única para cada pacote a ser criptografado, mesmo que haja pacotes com conteúdos iguais. Os primeiros n *bits*

gerados como saída da função *hash* indicam a quantidade de *dummy bytes* a serem inseridos. Os *bytes* subsequentes da saída da função são utilizados para indicar as posições nas quais os *dummy bytes* devem ser inseridos no pacote. Esses *bytes* subsequentes são divididos em n grupos de k *bits*, sendo que cada grupo representa a posição de um dos n *dummy bytes*.

O módulo gerador do EPP pode ser configurado para inserir uma quantidade mínima de *dummy bytes* pré-configurada. Essa quantidade mínima deve ser utilizada sempre que o valor representado pelos primeiros n *bits* de saída da função *hash* seja menor que essa quantidade mínima. Esse módulo também pode ser configurado para inserir uma quantidade de *dummy bytes* que seja menor ou igual ao tamanho do pacote, em *bytes*, multiplicado por um determinado fator. Isso permite controlar o *overhead* máximo gerado pelos *dummy bytes* em função do tamanho original do pacote. Seja x o valor representado pelos n *bits*, α o fator multiplicativo e s o tamanho original do pacote. Nesse caso, o *overhead* máximo gerado é igual a $x \ mod \ \alpha s$.

O módulo montador recebe como entrada o pacote e a saída do modulo anterior. Os *dummy bytes* são inseridos no pacote nas posições indicadas de acordo com os n grupos de k *bits*. O pacote modificado é concatenado com 2 *bytes* que indicam o tamanho do pacote original. O resultado é dado como saída do EPP e recebido pelo protocolo de segurança responsável pela criptografia do pacote.

4.2.2. Procedimento de Remoção

A Figura 3 apresenta o mecanismo de remoção do EPP. Ele atua após o pacote ser decifrado pelo protocolo de criptografia. Há uma estrutura em dois módulos, semelhante ao mecanismo de inserção. O módulo gerador é utilizado novamente no mecanismo de remoção e recebe os mesmos parâmetros recebidos durante o procedimento de inserção. Assim sendo, ele obtém a mesma informação sobre a quantidade de *dummy bytes* inseridos no pacote e sobre o posicionamento deles. O módulo removedor recebe o pacote e a saída do módulo gerador e remove os *dummy bytes*, devolvendo o pacote com tamanho e conteúdo originais.

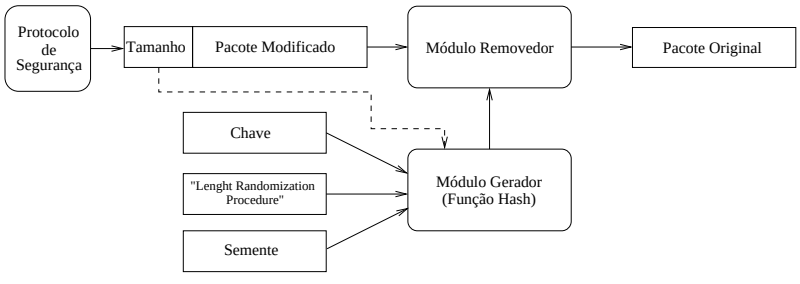

Figura 3. Procedimento de Remoção

4.3. Uso do EPP em Conjunto com Outros Protocolos

Como já apresentado, os algoritmos de cifra por fluxo sofrem significantemente mais com problemas de identificação dos pacotes a partir de seus tamanhos do que os algoritmos de cifra por blocos. Em função disso, o EPP é um mecanismo importante para fortalecer a

segurança de protocolos de segurança que utilizem algoritmos de cifra por fluxo. No entanto, a implantação do mecanismo em conjunto com protocolos que utilizem algoritmos de cifra por blocos pode ajudar de forma preventiva.

A maior parte dos ataques que exploram a previsibilidade do conteúdo e tamanho originais de pacotes criptografados ocorre em redes sem fio. Isso acontece pela simplicidade de se capturar pacotes nessas redes, bastando que para isso, o atacante escute o canal de comunicação. Em função disso, o EPP é proposto com grande motivação em redes sem fio IEEE 802.11 e leva em consideração os recursos e limitações dos protocolos já utilizados por tais redes. Porém, a aplicabilidade do mecanismo proposto não se limita a tais redes, visto que também existem ataques a protocolos em camadas mais elevadas da pilha de protocolos que se baseiam na previsibilidade de informações de pacotes [Paterson et al. 2009]. As duas seções seguintes apresentam como o EPP pode ser aplicado em conjunto com o WPA e o IEEE 802.11i, respectivamente.

4.3.1. Implantação com o WPA

Os protocolos WPA e IEEE 802.11i utilizam duas chaves de criptografia: a chave TK ou TEK (*Temporal Encription Key*) para pacotes destinados a um endereço físico *unicast* e a chave GTK (*Group Transient Key*) para pacotes destinados ao endereço físico de *broadcast*. Essas chaves são obtidas durante a autenticação do cliente na rede e são válidas, temporariamente, até que ocorra um novo processo de derivação de chaves.

Quando o EPP é implantado junto ao WPA, o módulo gerador recebe como entrada a chave TK ou a chave GTK de acordo com o endereço de destino do pacote (*unicast* ou *broadcast*). Além disso, O EPP também recebe como entradas a *string* "*Length Randomization Procedure*", o tamanho do pacote original e o contador TSC, que é utilizado pelo WPA como um contador de pacotes. O valor do TSC atua como semente para o EPP. Além disso, a função *hash* utilizada pelo EPP deve ser a *HMAC-SHA-1*, pois a mesma já é utilizada no WPA.

Ao se implementar o mecanismo proposto com o WPA, inserindo uma quantidade "suficiente" de *dummy bytes* nos pacotes, evita-se os ataques *Beck-Tews* e *Ohigashi-Morii*. Isso ocorre, pois o atacante não terá como ter certeza sobre o tipo do pacote capturado. Mesmo que o atacante ainda suponha corretamente o tipo do pacote capturado, haverá alterações importantes de conteúdo, pois há vários *dummy bytes* inseridos aleatoriamente. Na prática, o uso do EPP faz com que a descoberta da chave do MIC se torne difícil, pois ela depende da correta avaliação do tipo do pacote capturado e do conhecimento prévio de seu conteúdo original em texto-plano. No caso específico dos dois ataques citados, eles se tornam inoperantes sem a chave *MIC*. A Seção 5 apresenta uma discussão sobre a segurança do EPP e sobre a quantidade de *dummy bytes* a serem inseridos nos pacotes.

4.3.2. Implantação com o IEEE 802.11i

A implantação do EPP em conjunto com o protocolo IEEE 802.11i é parecida com a apresentada para o WPA. O IEEE 802.11i também utiliza, por padrão, a função *hash HMAC-SHA-1* para a derivação das chaves. Sua hierarquia de chaves é semelhante à hi-

erarquia de chaves do WPA e, desse modo, a chave TK (*unicast*) e a chave GTK (*broadcast*) são utilizadas na função de geração dos *dummy bytes* de acordo com o endereço de destino do quadro que encapsula o pacote. No entanto, existem algumas diferenças para a implantação do mecanismo em conjunto com o IEEE 802.11i. O contador *TSC* não é mais utilizado pelo EPP. Em seu lugar, utiliza-se o *Packet Number* do IEEE 802.11i que possui função semelhante ao TSC do WPA. Outra diferença é o fato do IEEE 802.11i utilizar uma cifra por bloco, de 16 *bytes*, para criptografar os pacotes. Para esse tipo de cifra, o tamanho de cada pacote, em *bytes*, é modificado para o primeiro múltiplo de 16 que seja superior ao seu tamanho. Portanto, o tamanho final de um pacote cifrado só é modificado pelo EPP caso o número de *dummy bytes* inseridos seja suficiente para transpor o limite de *bytes* do último bloco.

5. Controle do *Overhead* e Segurança do EPP

Foi realizada uma captura de tráfego na rede do Centro de Informática da UFPE. Dentre os pacotes capturados, mais de 25% possuiam tamanho entre 26 e 42 *bytes*. Esse valor expressivo de pacotes pequenos é devido, em parte, pela presença de ACKs do protocolo TCP (40 *bytes*, em geral) e ARPs (28 *bytes* fixos). Como os ataques baseados em previsibilidade de tamanho e conteúdo de pacotes, geralmente, utilizam pacotes pequenos, é possível minimizar o *overhead* global do EPP no tráfego da rede. Para isso, o percentual de *dummy bytes* inseridos pode ser reduzido a medida que o tamanho do pacote original aumenta. Nesse caso, uma maior proteção seria dada a pacotes pequenos que são mais previsíveis e vulneráveis do que a pacotes grandes que, em geral, são mais imprevisíveis devido a maior variabilidade de conteúdo.

A eficácia do EPP contra ataques que tentam advinhar o tipo de pacote criptografado depende do tipo de cifra utilizada, do número mínimo de *dummy bytes* inseridos e da dificuldade de se conhecer tal número, uma vez que ele é criptografado. Suponha o uso de cifra de fluxo e a inserção de z *bytes* em um pacote ARP. Ao se supor adicionalmente que o atacante tenha capturado esse pacote criptografado e acertado o tipo do pacote e o número de *dummy bytes*, ele teria ao todo $\binom{28+z}{28}$ pacotes para analisar e escolher 1 como sendo o pacote ARP original. Para $z = 10$, por exemplo, o atacante precisaria analisar mais de 472 milhões de pacotes. Certamente, quanto mais próximo de 1 for o valor de z, mais fácil será para um atacante conseguir advinhar o tipo de pacote criptografado. Um estudo mais aprofundado sobre a menor quantidade de *dummy bytes* a serem inseridos nos pacotes de forma a garantir a eficácia do EPP será realizado em trabalhos futuros.

A resistência do EPP a ataques depende da escolha do algoritmo HMAC a ser utilizado e do quão difícil é a recuperação da chave secreta empregada pelo mecanismo. Assim sendo, ao se utilizar o EPP em conjunto com os protocolos WPA e IEEE 802.11i, sua resistência a ataques depende da segurança do algoritmo HMAC-SHA-1 e da dificuldade de se encontrar as chaves TK e GTK usadas nesses dois protocolos. O algoritmo HMAC-SHA-1 é considerado bastante seguro atualmente. Em redes IEEE 802.11 protegidas pelo WPA ou pelo IEEE 802.11i, é possível encontrar as chaves TK e GTK através de um ataque de dicionário quando a rede usa o método de autenticação pessoal em conjunto com uma *passphrase*. Contudo, esse ataque só é praticável se a *passphrase* possuir menos de 20 caracteres [Fogie 2005]. Essa vulnerabilidade é considerada do usuário/administrador e não do protocolo de segurança.

6. Conclusões

A previsibilidade de informações em alguns tipos de pacotes oferece riscos à segurança das próprias redes. Essa característica foi utilizada em ataques contra diversos protocolos de segurança, principalmente em redes IEEE 802.11. Com base nesse problema, esse artigo propôs um mecanismo, denominado EPP, capaz de eliminar a previsibilidade de tamanho e conteúdo de pacotes através da inserção de uma quantidade aleatória de *bytes* em posições igualmente aleatórias. O EPP permite um ajuste fino de seu *overhead* de acordo com o tamanho original dos pacotes a serem protegidos.

Um fator relevante sobre o EPP é o fato de sua arquitetura ser significativamente simples e a mesma poder ser adaptada para operar em conjunto com outros protocolos de segurança de redes de computadores. Quando usado em conjunto com o WPA, o EPP é capaz de evitar os ataques *Beck-Tews* e *Ohigashi-Morii*. A implantação do EPP com o IEEE 802.11i é uma medida preventiva, dificultando a criação de ataques futuros contra esse protocolo e que venham a se basear na previsibilidade de informações de pacotes.

Referências

Beck, M. and Tews, E. (2009). Practical Attacks Against WEP and WPA. In *Proceedings of the Second ACM Conference on Wireless Network Security - WiSec'09*, pages 79–86.

Daemen, J. and Rijmen, V. (1998). AES Proposal: Rijndael. Technical report.

Fluhrer, S., Mantin, I., and Shamir, A. (2001). Weakness in the Key Scheduling Algorithm of RC4. *Lecture Notes in Computer Science - Selected Areas in Cryptography*, (2259):1–24.

Fogie, S. (2005). Cracking Wi-Fi Protected Access (WPA), Part 2. `http://www.fermentas.com/techinfo/nucleicacids/maplambda.htm`.

Hayes, V. et al. (1999). IEEE Standard 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Computer Society*.

Kerry, S. J. et al. (2004). IEEE Standard 802.11i, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Medium Access Control (MAC) Security Enhancements . *IEEE Computer Society*.

KoreK (2004). Chopchop (Experimental WEP Attacks).

Krawczyk et al. (1997). HMAC: Keyed-Hashing for Message Authentication. RFC 2104.

Ohigashi, T. and Morii, M. (2009). A Practical Message Falsification Attack on WPA. In *Procedings of Joint Workshop on Information Security, Cryptography and Information Security Conference System*.

Paterson, M. R. et al. (2009). Plaintext Recovery Attacks Against SSH. In *Procedings of IEEE Symposium on Security and Privacy*, pages 16–26.

Tews, E., Weinmann, R.-P., and Pyshkin, A. (2007). Breaking 104 Bit WEP in Less Than 60 Seconds. *Lecture Notes in Computer Science - Information Security Applications*, (4867):188–202.

Wi-Fi Alliance (2003). Wi-Fi Protected Access: Strong, Standards-based, Interoperable Security for Today's Wi-Fi Networks. Technical report.