Segurança em VoIP

28838 - André Lima - alima@alunos.isel.ipl.pt
29786 - Herman Duarte - hcduarte@alunos.isel.ipl.pt

ISEL - Redes e Serviços de Comunicação Multimédia

4 de Janeiro de 2009

1 Introdução

As tecnologias VoIP têm tido uma enorme adopção. Essa adopção, e previsão de que muitos mais o pretendem fazer, tem trazido uma crescente preocupação no que toca aos aspectos relativos à segurança desse tipo de comunicações. Existem várias formas de diminuir drasticamente a probabilidade de sucesso de um atacante. No caso de se ter uma rede presumivelmente segura, é importante prevenir acessos não autorizados e, caso se pretenda comunicar com o exterior, que se utilizem tecnologias VPN, nomeadamente a *Voice and Video-enabled VPN* (V3PN)(2). É também aconselhável que se isole o tráfego VoIP em VLANs distintas às dos restantes tipos de tráfego. Esta abordagem traz a vantagem de se isolarem os potenciais danos causados por um ataque/falha, e também traz a vantagem que advém do facto de se poderem atribuir diferentes prioridades ao tráfego de acordo com as VLANs. Contudo, esta abordagem requer total controlo sobre a transição de pacotes de uma VLAN para a outra.

Para além dos cenários já apresentados, é importante notar que existem outros cenários onde existe comunicação em redes não seguras, nomeadamente na Internet, e há quem possa desejar acrescentar mais uma camada de segurança, no caso do controle de acessos, da rede presumivelmente segura, não funcionar. Para esses casos existe a opção de se cifrarem os dados (voz, vídeo e sinalização). Apesar de existirem opções de cifra na camada de Internet, é sugestão do grupo que se utilizem protocolos da camada aplicacional. Esta abordagem traz aquilo a que se chama de segurança *end-to-end* pois, mesmo que existam nós intermediários (*proxies*), o conteúdo da mensagem estará seguro até que chegue ao destinatário. A solução adoptada para a segurança *end-to-end* é a utilização do protocolo Secure RTP(10).

Contudo, não é apenas o *payload* da comunicação VoIP que necessita de serviços de segurança, mas também a sinalização. Pelo facto de estar a ser globalmente adoptado sendo já visto como o protocolo de sinalização VoIP do futuro, o grupo decidiu escolher o *Session Initiation Protocol* (SIP)(3) para a análise de segurança. Começa-se por realçar algumas das características do SIP, que o tornam mais vulnerável do ponto de vista de segurança. De seguida apresentam-se as opções de segurança recomendados pelo protocolo SIP bem como uma pequena comparação entre estas. Alguns ataques ao protocolo SIP são apresentados de seguida, demonstrando assim a exploração das vulnerabilidades presentes. Em relação ao SIP do ponto de vista de segurança, conclui-se com algumas soluções que visam mitigar as vulnerabilidades presentes consequentemente eliminando a maioria dos ataques.

2 Segurança no SIP

O protocolo SIP tem as mesmas vulnerabilidades a nível da camada de rede e da camada aplicacional que outros protocolos VoIP. As seguintes características do protocolo o tornam ainda mais vulnerável:

- Imaturidade - o protocolo SIP é relativamente novo.
- Extensibilidade - o protocolo suporta extensões que normalmente são recentes e fracos do ponto de vista de segurança.
- Complexidade - o protocolo é relativamente complexo, e quando se tem em conta as extensões necessárias, torna-se bastante complexo.
- Codificação - a codificação é textual logo é mais fácil de se obter com um *sniffer*.

Muitas implementações utilizam o UDP *(Universal Datagram Protocol)* como protocolo de transporte. Do ponto de vista da segurança o UDP é menos seguro que o TCP *(Transport Control Protocol)*. O TCP é um protocolo orientado a ligação (existindo assim a noção de sessão), garante a entrega de pacotes e utiliza números de sequência. Todas estas características tornam mais difícil o trabalho necessário a um atacante, para introduzir um pacote fabricado na rede, passível de ser aceite por qualquer dispositivo como sendo um pacote válido.

2.1 Mecanismos de segurança fornecidos pelo protocolo SIP

As mensagens do protocolo SIP derivam do modelo pergunta/resposta do protocolo HTTP, logo todos os mecanismos de segurança disponibilizados ao HTTP são aplicáveis ao SIP. Existe também a possibilidade de utilizar mecanismos de segurança semelhantes ao email, nomeadamente o S/MIME (Secure MIME) e PGP (Pretty Good Privacy) nas mensagens SIP, visto que o protocolo suporta mensagens do tipo MIME. Existe também a possibilidade de utilizar SIPS URI (SIP Secure URI) que faz com que o sistema tente criar uma ligação segura utilizando o protocolo TLS (Transport Layer Security). Por último há a possibilidade de se utilizar o mecanismo genérico de segurança IPsec (IP security) que opera na camada de rede.

De seguida apresenta-se um quadro com a comparação dos mecanismos de segurança do ponto de vista da autenticação, integridade e confidencialidade recomendados na versão 1 do SIP actualizados para a versão 2 do protocolo:

Authentication methods: PSK Pre-Shared Keys PKI Public Key Infrastructure	Authentication	Data Integrity	Confidentiality	
HTTP 1.0 Basic Authentication	PSK	-	-	Deprecated by SIPv2 Insecure transmission of password
HTTP 1.1 Digest Authentication	PSK	-	-	Challenge/response exchange based on MD5 hash of [strong] password
Pretty Good Privacy (PGP)	PKI	✓	✓	Deprecated by SIPv2
Secure MIME (S/MIME)	PKI	✓	✓	For encryption the public key of the recipient user agent must be known
SIPS URI (TLS)	PKI	✓	✓	SIP application and proxies must tightly integrate TLS
IP Security (IPsec)	PKI	✓	✓	Integration with SIP application not required but proxies must be trusted

Figura 1: Mecanismos de segurança do protocolo SIP

2.1.1 HTTP 1.1 Digest Authentication

Neste mecanismo é enviado ao SIP UA um desafio quando este tenta registar, iniciar uma sessão, etc. O desafio consiste num *nounce* (valor aleatório), indicação de qual o algoritmo de *hash* a utilizar (normalmente MD5 ou SHA-1) e um *realm*. A resposta ao desafio consiste em produzir o *hash* da seguinte combinação com o algoritmo indicado: *username + secret password + nounce + SIP method + requested URI*. A password tem de ser conhecida pelo servidor que lança o desafio para poder verificar a resposta.

Este mecanismo permite ataques de dicionário *off-line* com bastante probabilidade de sucesso no caso de passwords fracas. Basta a captura das respostas aos desafios, porque estas contêm toda a informação utilizada para o calculo do *hash*, excepto a password. Para além desta desvantagem este mecanismo não oferece confidencialidade nem integridade às mensagens.

A ferramenta *authtool* (desenvolvida pelos autores do livro(1)) permite efectuar ataques de dicionário mediante a indicação de uma password ou uma lista de passwords. As ferramentas *SiVus* e *Cain & Abel* também são capazes de efectuar este tipo de ataque e de forma mais automática.

2.1.2 Secure MIME (S/MIME)

Uma das vantagens deste mecanismo face aos restantes é que permite obter segurança *end-to-end* além da segurança ponto-a-ponto. Esta solução, também como qualquer outro sistema de autenticação baseada em chave pública, sofre do problema de distribuição de certificados(chave pública) podendo levar a ataques de MiTM (Man-in-The-Middle).

2.1.3 SIPS URI (TLS)

Quando se utiliza um SIPS URI em vez de um SIP URI, todos os dispositivos desde a origem até o destino têm de suportar o protocolo TLS (automaticamente o TCP, pois o TLS depende deste). A desvantagem do TLS consiste em ser um mecanismo de segurança ponto-a-ponto levando a que as mensagens sejam decifradas em cada ponto/dispositivo, possivelmente modificadas e cifradas novamente para o próximo dispositivo/ponto.

2.1.4 IP Security (IPsec)

Este é um mecanismo de segurança genérico que opera na camada de rede. Este mecanismo pode estar activo permanentemente nas máquinas/dispositivos que compõem a rede SIP ou pode ser activado por estes quando necessário. A última opção é a mais desejada do ponto de vista funcional, no entanto exige interacção e conhecimento da aplicação/dispositivo em relação ao IPsec.

Este mecanismo suporta ambas as formas de autenticação (PSK e PKI), no entanto a autenticação com PSK neste ambiente, tem muitos problemas de autenticação associado logo não é recomendado a sua utilização.

2.2 Alguns ataques que exploram as vulnerabilidades do SIP

2.2.1 Ataques de inundação

Os ataques de inundação têm por objectivo interromper o serviço nos dispositivos SIP, através da inundação de mensagens na rede. Se o ataque for bem efectuado, estes dispositivos são levados a processar estas mensagens o que faz com que os recursos destes se esgotem, deixando assim de poder processar a maioria das mensagens legítimas dos utilizadores da rede VoIP. Alguns dispositivos conseguem operar parcialmente durante este tipo de ataque, enquanto que outros acabam por deixar de responder mesmo depois do ataque ter terminado sendo então necessário serem reiniciados.

Este tipo de ataques tem maior probabilidade de sucesso se os dispositivos da rede SIP aceitarem mensagens transportadas por UDP. Pode-se inundar a rede com mensagens UDP, com mensagens de INVITE (estas têm maior probabilidade de sucesso) e com mensagens RTP.

No caso de se utilizar o protocolo de transporte TCP pode-se sempre recorrer a inundações de mensagens SYN.

As seguintes ferramentas foram modificadas/desenvolvidas pelos autores do livro(1):

- *udpfloodtool* - utilizada para efectuar ataques de inundação através de mensagens UDP.
- *invitefloodtool* - utilizada para efectuar ataques de inundação através de mensagens INVITE.
- *rtpfloodtool* - utilizada para efectuar ataques de inundação através de mensagens RTP.

2.2.2 Ataques de remoção de registos

REGISTER, é uma das primeiras mensagens enviadas por um UA *(User Agent)*. Esta mensagem é enviada ao servidor SIP *registrar* responsável pelo domínio onde este dispositivo se pretende registar. Um atacante pode enviar uma mensagem de REGISTER ao ao servidor SIP *registrar*, onde os cabeçalhos *To* e *From* contêm o SIP URI do dispositivo alvo; o cabeçalho *Contact* tem como valor o símbolo especial *, que indica todos os contactos; e o cabeçalho *Expires* com o valor 0 (zero).

Este ataque é classificado como um ataque de DoS *(Denial of Service)* na medida em que impossibilita o alvo de receber chamadas. Este ataque é válido até que o dispositivo reenvie uma nova mensagem REGISTER para renovar o registo. No entanto o atacante pode sempre enviar uma nova mensagem, logo

a seguir à mensagem enviada pelo dispositivo, anulando assim o registo novamente.

Através das seguintes aplicações é possível efectuar este tipo de ataque: *erase_registrations* (desenvolvida pelos autores do livro(1)) e SiVus.

2.2.3 Ataques de adição registos

Este método consiste no envio de uma mensagem REGISTER ao servidor SIP *registrar* com o objectivo de adicionar um ou mais contactos, ao conjunto de contactos de um determinado alvo. Com este método o atacante pode provocar ataques de DoS ao alvo, por exemplo, ser o primeiro a atender quando uma chamada é recebida; adicionando aos contactos do dispositivo alvo, os contactos de outros dispositivos, por exemplo do mesmo piso num escritório, faz com que esses dispositivos começam a tocar quando uma chamada é efectuada ao dispositivo alvo. Quanto maior o número de contactos adicionados, maior é a probabilidade de não ser o utilizador do dispositivo a atender a chamada. Além deste aspecto funcional, é muito irritante ter vários telefones a tocar ao mesmo tempo quando era suposto tocar somente um.

De realçar que nem todos os SIP *registrars* tem o mesmo comportamento após o recebimento de uma mensagem REGISTER. Os autores do livro(1) utilizaram dos SIP *proxy/registrars*: SER *(SIP Express Router)* e *Asterisk*, em que este último faz a substituição do registo, e não a adição.

Através das seguintes aplicações é possível efectuar este tipo de ataque: *add_registrations* (desenvolvida pelos autores do livro(1)) e SiVus.

2.2.4 Ataques de hijacking de registos

Hijacking de registo, consiste em substituir o contacto do dispositivo alvo, no servidor SIP *registrar*, por um contacto que o atacante controla. Com esta técnica o atacante pode provocar ataques de DoS, *Phishing* - onde o atacante/programa faz-se passar pelo utilizador do dispositivo alvo. Por exemplo utilizando uma aplicação de *voicemail* o atacante pode ficar na posse de informações valiosas.

Entre os ataques que podem ser explorados por esta técnica, o ataque de MiTM *(Man-in-The-Middle)* aplicacional é o mais poderoso e perigoso. Com este ataque o atacante faz o papel de intermediário entre o chamador e o servidor SIP *proxy* ou entre SIP *proxies*, podendo visualizar e modificar o conteúdo da sinalização e o conteúdo do fluxo multimédia trocado entre os *endpoints*. A seguinte figura ilustra este tipo de ataque:

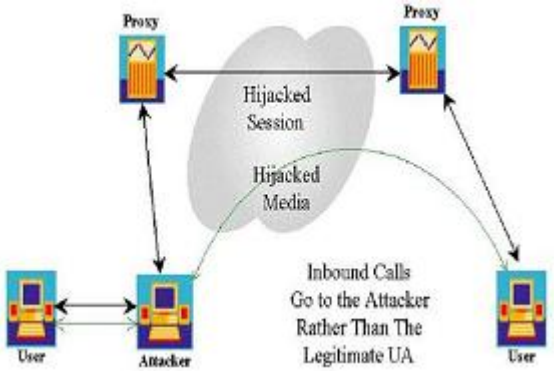

Figura 2: Hijacking de registo - Ataque MITM

A ferramenta *reghijacker* (desenvolvida pelos autores do livro(1)) permite efectuar este ataque em ambientes autenticados e não autenticados.

2.2.5 Ataques de redireccionamento

As respostas *301 Moved Permanently* e *302 Moved Temporarily* podem ser enviadas por qualquer UA ou por um servidor SIP *proxy/redirect*. No envio de uma mensagem INVITE por parte de um dispositivo, um atacante que esteja a escuta pode responder com uma das mensagens acima referidas, levando o chamador a entrar em contacto com por exemplo uma aplicação/dispositivo controlada pelo atacante, podendo então realizar vários tipos de ataques como por exemplo: DoS, MiTM, *Phishing*, etc.

Os autores do livro(1) desenvolveram uma aplicação - *redirectpoison* - que tira partido desta técnica. Esta aplicação aumenta o seu nível de prioridade por forma a enviar a resposta o mais rápido possível, ganhando assim a possível *race condition* entre esta mensagem e a mensagem enviada pelo servidor SIP *proxy/redirect*.

2.2.6 Ataques à sessão

A mensagem BYE é utilizada pelos *endpoints* para terminar uma sessão previamente estabelecida. A mensagem BYE tem de transportar a informação necessária para identificar a sessão para qual se destina, nomeadamente: *Call-ID*, *FromTag* e *ToTag*. Basta ao atacante identificar estes parâmetros, que normalmente se encontram na resposta OK enviada pelo chamado, e enviar uma mensagem BYE a um dos *enpoints*.

A ferramenta *teardown* (desenvolvida pelos autores do livro(1)) permite explorar esta técnica causando assim ataques de DoS.

A seguinte figura ilustra este ataque:

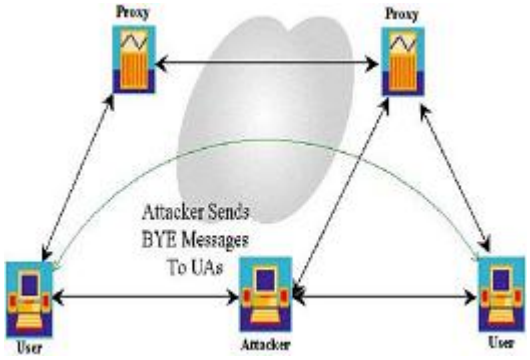

Figura 3: Ataque que termina um sessão entre dois utilizadores

2.2.7 Ataques à telefones SIP

Uma das extensões ao protocolo bastante implementada, é a funcionalidade SUBSCRIBE/NOTIFY que permite aos dispositivos receberem notificações assíncronas após a subscrição há um determinado tipo de evento.

Muitos dos telefones SIP implementam esta funcionalidade. Os autores do livro(1) concluíram que muitos dos telefones aceitam mensagens NOTIFY mesmo sem terem enviado uma mensagem SUBSCRIBE para o evento em causa. Um dos eventos que pode afectar de forma errada os telefones SIP é o *check-sync*. A ferramenta *check_sync_reboot* (desenvolvida pelos autores do livro(1)) permite enviar este tipo de mensagens. Este tipo de ataque pode levar ao *reboot* do telefone ou então provocar um *crash* que leva a que o telefone seja reiniciado.

2.2.8 Medidas para mitigar os ataques acima referidos:

De seguida apresenta-se algumas medidas à utilizar com o objectivo de mitigar a maioria dos ataques acima referidos.

- utilização do protocolo TCP na sinalização SIP entre TODOS os dispositivos.
- utilização de VLANs para separar a sinalização do fluxo multimédia.
- aplicar as últimas actualizações ao software e *firmware*.
- utilizar autenticação SIP com passwords que são consideradas fortes (tornando assim os ataques de dicionário mais difícil).
- mudar as passwords por omissão que vêm configuradas com os dispositivos.
- remover ou limitar o acesso aos serviços disponibilizados pelos dispositivos.
- utilizar *firewalls* SIP.

3 Segurança no RTP

3.1 Segurança nativa

Os protocolos das camadas inferiores, por exemplo o IP, poderão fornecer serviços de segurança, como a confidencialidade, a autenticidade e a integridade. Contudo, antes destes serviços terem sido especificados para IP(4), a sua necessidade por parte de aplicações áudio e vídeo, já era um facto real. Esta é a razão pela qual teve-se a necessidade de definir na própria especificação do RTP(5), como disponibilizar o serviço de confidencialidade. Essa confidencialidade é dada pela utilização, por omissão, do algoritmo *Data Encryption Standard* (DES) em modo *Cipher Block Chaining* (CBC), permitindo assim que não sejam

detectados padrões nas mensagens cifradas. No RFC do RTP(5), é aconselhada a utilização de esquemas de cifra mais fortes como o *Triple-DES*, ou mesmo a utilização de extensões a esta especificação, como é o caso do padrão *Secure RTP* (SRTP). De realçar o facto dos serviços de integridade e autenticidade não serem abordados na especificação do RTP. Isto acontece porque, ao contrário da confidencialidade que é implementada com esquemas de cifra simétricos, a implementação de integridade e autenticidade é feita através de cifras assimétricas, tipicamente utilizando certificados para o efeito. Isto traria a complexidade, do problema da distribuição de chaves, para o protocolo RTP, tornando-o mais complexo e potencialmente menos seguro. De acordo com o a especificação do RTP (5), é assumido que, caso necessários, este serviços deverão ser disponibilizados pelos protocolos das camadas inferiores.

3.2 Análise à segurança nativa

A utilização do DES(6), como algoritmo por omissão, por si só é considerada uma fraqueza devido ao pequeno tamanho das suas chaves (56 bits). Para além disso, o modo CBC requer a utilização de um *Initialization Vector* (IV) para a cifra do primeiro bloco. Para que seja considerado seguro, este IV deverá ser aleatório e independente de mensagem para mensagem, ou seja, não dedutível. Isto acontece no caso do RTCP, que permite o anexo de um valor de 32 bits em cada *frame*. Contudo, no caso do RTP, este valor é calculado a partir dos valores do *timestamp* e do *sequence number* e apesar destes realmente consistirem em valores aleatórios, no primeiro *frame*, nos *frames* seguintes, são calculados relativamente ao primeiro, perdendo a desejada independência entre os IVs. Outro problema deste modo, é que são necessários todos os blocos cifrados para a decifra destes, o que obriga ao reenvio de pacotes mesmo que já não sejam necessários, por exemplo se o tempo em que deveriam ser apresentados passou.

Uma das funcionalidades(7) dos *translators* é a de poder comprimir cabeçalhos RTP. Isto é particularmente útil em casos onde o corpo da mensagem é relativamente pequeno, como é o caso de um corpo de 20 *bytes* em relação ao *header* de 12(mínimo - assumindo que CC=0), e em que a comunicação esteja a ser feita sobre linhas de baixa velocidade, por exemplo em *modems* de 14.4 e 28.8 Kbps. A compressão de cabeçalhos pode ser feita apenas ao RTP (*end-to-end*) ou simultaneamente nos cabeçalhos IP, UDP e RTP (*link-to-link*). Visto que o número de *bytes* resultante de ambas as compactações são relativamente semelhantes (2-4 *bytes*), e visto que a compactação em massa (*link-to-link*) é mais eficiente devido ao menor número de *bytes* transmitidos, a compactação em massa é a mais utilizada.

Voltando à análise, a compactação previamente descrita não é aplicável caso se disponha do serviço de confidencialidade. Apesar de ser possível, cifrando e decifrando a mensagem em cada *translator*, isto traz enorme perda de *performance* à comunicação.

Outro problema significativo consiste no facto de, para haver comunicação, com confidencialidade, todos os potenciais comunicadores (incluindo os *mixers*) deverão ter na sua posse a chave (simétrica) necessária à cifra e decifra das mensagens. Isto resulta no facto de, caso um dos participantes seja atacado com sucesso, todas as comunicações tornar-se-ão vulneráveis daí em diante. Isto significa que, na prática, este serviço não é aplicável para sessões com grande número de participantes, visto que a probabilidade de toda a sessão estar vulnerável aumenta com o número de participantes.

3.3 Solução: Secure Real-Time Transport Protocol (SRTP)

O *Secure Real-Time Transport Protocol* (SRTP)(10) consiste numa extensão do RTP que disponibiliza confidencialidade, autenticidade, e protecção contra ataques de *replay*, não só para o RTP, como também para o RTCP(5). A confidencialidade é aplicada a nível do corpo da mensagem permitindo, como já vimos antes, a compressão do cabeçalho por parte dos *translators*. De salientar que os serviços de segurança disponibilizados pelo SRTP ao RTP, são os mesmos que os disponibilizados pelo SRTCP ao RTCP. Também é importante notar que o facto de se disponibilizar autenticidade, significa implicitamente que se disponibiliza integridade, pois caso contrário a informação relativa à origem da

mensagem poderia facilmente ser adulterada. O pacote terá o seguinte formato:

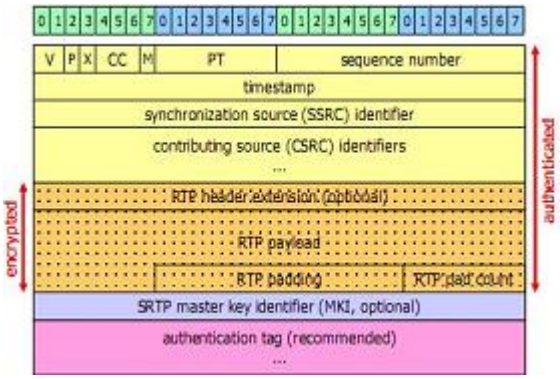

Figura 4: Formato do Pacote SRTP

3.3.1 Confidencialidade

O algoritmo de cifra utilizado por omissão é o *Advanced Encryption Standard* (AES)(8) e são definidos dois modos de utilização: *Segmented Integer Counter Mode* e o modo *f8-mode*. O *Segmented Integer Counter Mode* consiste num típico *Counter Mode*. Isto significa que a mensagem cifrada C não resulta directamente das entradas da Mensagem *ClearText* M e da chave K, mas sim de M e de uma *keystream* que, por sua vez, resulta das entradas K e de uma *Nonce* do tipo *counter*, como demonstrado na figura 2.2.

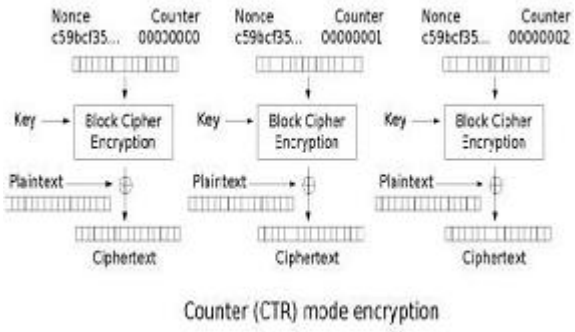

Figura 5: modo counter

Ao contrário de outros modos, como o CBC, este modo permite o acesso aleatório aos dados cifrados. Isto consiste num factor crucial neste cenário de comunicação via RTP sobre UDP, visto que pacotes podem se perder, deixando de ter a necessidade de ter obrigatoriamente de requisitar repetidamente esses mesmos pacotes perdidos. O *nonce* na figura 2.2, é especificado no SRTP como sendo um do tipo incremental. O tamanho por omissão utilizado no SRTP, para a chave e para o *salt key*, são 128 e 112 *bits* respectivamente. Este *salt key* tem essencialmente a mesma função que o *nonce*, ou seja, a de prevenir que o *output* de uma determinada mensagem seja sempre o mesmo, caso a chave também seja a mesma. Isto atribui um muito maior nível de dificuldade a ataques de pré-computação.

Quanto ao modo *f8-mode*, este é uma variante do modo *Output Feed Back* (OFB) demonstrado na figura 2.3.

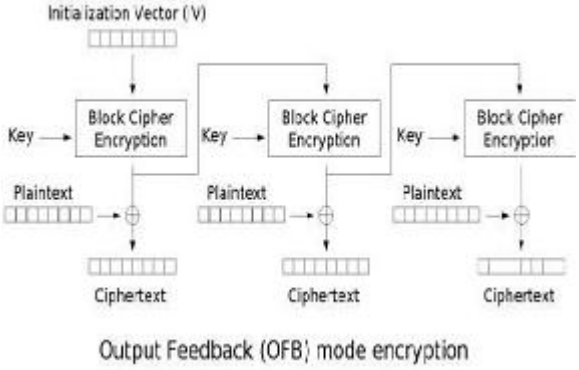

Figura 6: modo output feedback

O algoritmo f8 foi desenvolvido para cifrar dados UMTS (Universal Mobile Telecommunications System, como redes 3G).

A desactivação da confidencialidade, podendo manter a autenticidade e integridade, é outra opção dada pelo SRTP. Esta opção é denominada por *NULL Cipher*, o que faz com que o AES devolva a entrada M como sendo C.

3.3.2 Autenticidade

Para que melhor se entenda a seguinte explicação, vamos referenciar a porção do pacote SRTP (fig. 1) a ser autenticada, como sendo MA. O que acontece é que o emissor cria uma *tag* relativa a MA e anexa-a ao pacote (*authentication tag*). Por sua vez, o receptor calcula a *tag* do MA do pacote recebido e compara-o à *authentication tag* contida no pacote. Caso sejam iguais, então a mensagem é válida; caso contrário, o pacote não é validado e é retornado ao emissor a mensagem de erro '*AUTHENTICATION FAILURE*'. O HMAC-SHA1(9) é a cifra utilizada para autenticação.

3.3.3 Protecção contra ataques de replay

Ataques de *replay* consistem no atacante armazenar pacotes de dados enviados por um emissor e recebidos pelo respectivo receptor, e posteriormente reenviá-los ao receptor em causa fazendo-o pensar que o emissor enviou um pacote válido. Estes ataques não podem ser detectados pela simples verificação de integridade. Não obstante, a integridade é parte crucial na detecção dos mesmos. Essa detecção é feita armazenando uma lista com os índices dos últimos pacotes recebidos e autenticados. Estes índices consistem em números sequenciais, o que significa que, quando o atacante injectar o pacote armazenado de novo na rede, o receptor apenas terá de verificar se o índice do pacote recebido é maior que o último inserido na lista, ou é um pacote que ainda não tenha sido recebido. Este último cenário basea-se no tamanho da lista que é configurável, sabendo que existe um limite para além do qual já não mais interessa receber um pacote. Os índices de cada pacote são calculados com base nos respectivos *sequence numbers*, daí que a integridade dos mesmos seja fundamental à detecção deste tipo de ataques.

3.3.4 Gestão de chaves

Uma função para geração de chaves, conhecida por *Key Derivation Function* (KDF), é utilizada para gerar as diferentes chaves necessárias, como as de cifra, os *salt keys*, e as chaves de autenticação, apartir de uma *master key* e uma *master salt*. Esta geração é melhor explicada através da figura 2.4.

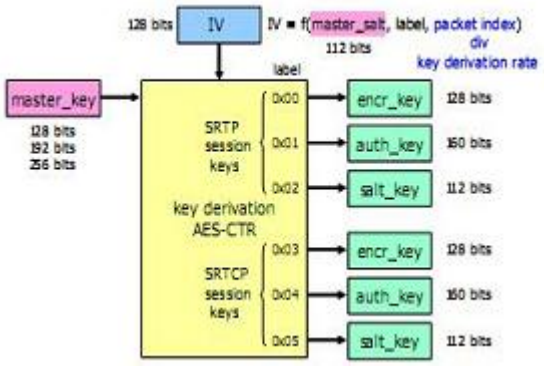

Figura 7: função de geração de chaves

Os pacotes SRTP têm a si anexados o MKI, que consiste num identificador da *master key* utilizada para gerar as chaves utilizadas na cifra e no processo de autenticação do pacote em causa. Esse MKI é importante para que, no caso de renovação da *master key*, o receptor possa identificar que chaves utilizar para a decifra e verificação de autenticidade e integridade do pacote.

O SRTP depende de um protocolo externo de gestão de chaves para negociar e transportar de forma segura a *master key* e a *master salt* iniciais. Dois protocolos especificamente concebidos para esta função são o ZRTP(11) e o MIKEY(12).

4 Conclusão

Tem-se a noção de que não existe nenhum sistema 100% seguro. Porém, aquando da implementação de um sistema VoIP numa rede, conclui-se que têm-se várias formas de aumentar as probabilidades de sucesso, na protecção do sistema, até muito perto desses 100%. A utilização do SRTP na protecção do *payload* revela-se bastante satisfatória, garantindo os principais serviços de segurança que se podem desejar no transporte do fluxo multimédia.

A segurança de uma rede VoIP está dependente do nível de segurança empregue na infraestrutura que a suporta. A utilização do TCP como protocolo de

transporte é recomendada. Deverá ser utilizada a técnica de protecção em profundidade tornando assim o sistema mais robusto do ponto de vista de segurança.

Referências

[1] Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions by David Endler and Mark Collier *http://www.hackingvoip.com/*

[2] Cisco, *www.cisco.com/univercd/cc/td/doc/solution/v3pnload.pdf*

[3] J. Rosenberg, Jun 2002, *SIP: Session Initiation Protocol*

[4] Kent, S. e R. Atkinson, Nov 1998, *Security Architecture for the Internet Protocol*

[5] H. Schulzrinne, Jul 2003, *RTP: A Transport Protocol for Real-Time Applications*

[6] NIST, 1993, *Data Encryption Standard - www.itl.nist.gov/fipspubs/fips46-2.htm*

[7] C. Casner, Cisco Systems, V. Jacobson, Feb 1999, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*

[8] NIST, Nov 2001, *Advanced Encryption Standard - csrc.nist.gov/publications/fips/fips197/fips-197.pdf*

[9] H.Krawczyk, Feb 1997, *HMAC: Keyed-Hashing for Message Authentication*

[10] M. Baugher, Mar 2004, *The Secure Real-Time Transport Protocol (SRTP)*

[11] Phil Zimmermann, Nov 2008, *http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-11*

[12] J. Arkko, Aug 2004, *MIKEY: Multimedia Internet KEYing*