
Universidade de Aveiro Departamento de Electrónica e Telecomunicações
2005

Cristiano Martins Pereira

Segurança em Redes de Comunicações de Área Local não-Cabladas IEEE 802.11

Universidade de Aveiro Departamento de Electrónica e Telecomunicações
2005

Cristiano Martins Pereira

Segurança em Redes de Comunicações de Área Local não-Cabladas IEEE 802.11

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. Rui Jorge Morais Tomaz Valadas, Professor Associado do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro

o júri

presidente

Prof. Dr. José Carlos Esteves Duarte Pedro
Professor Catedrático da Universidade de Aveiro

vogais

Prof. Dr. Rui Jorge Morais Tomaz Valadas
Professor Associado da Universidade de Aveiro (Orientador)

Prof. Dr. Carlos Nuno da Cruz Ribeiro
Professor Auxiliar do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. André Ventura da Cruz Marnôto Zúquete
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Gostaria, em primeiro lugar, de agradecer ao Professor Doutor Rui Jorge Morais Tomaz Valadas, Professor Associado do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro, na qualidade de orientador, pela oportunidade concedida para a realização do presente trabalho, assim como pelo rigor, crítica permanente, sessões de trabalho conjunto e por toda a disponibilidade e empenho que sempre demonstrou.

Ao Instituto de Telecomunicações – Pólo de Aveiro e ao Centro de Informática e Comunicações da Universidade de Aveiro pelas condições que colocaram à minha disposição.

O meu obrigado ao grupo de amigos que me dão o privilégio da sua convivência. A amizade e o apoio nos momentos difíceis criaram fortes laços que irão durar para todo o sempre.

Dedico esta dissertação aos meus Pais, António e Amélia, cujos ensinamentos marcaram a minha pessoa e serão a essência de toda a minha vida.

Finalmente gostaria de agradecer de forma especial à minha esposa Mabilde. Não é possível expressar em palavras a minha gratidão, por todos os sacrifícios, por toda a paciência, pela aceitação dos meus objectivos de vida, e pelo amor que me devota incondicionalmente.

palavras-chave segurança, redes não-cabladas, IEEE 802.11.

resumo

As redes de comunicações de área local não-cabladas tornaram-se nos últimos anos bastante populares, sendo a sua utilização hoje em dia comum tanto em redes privadas como públicas. A mobilidade que é permitida aos utilizadores e a simplicidade de implementação são os principais factores que contribuíram para a sua massificação. No entanto, existem problemas de segurança inerentes à utilização das redes não-cabladas. Alguns destes riscos são similares aos encontrados nas redes cabladas convencionais; outros advém do carácter não-guiado do meio de transmissão. Os ataques à autenticação e os ataques de negação de serviço são os principais riscos de segurança associados às redes de comunicações não-cabladas.

Esta dissertação aborda os aspectos de segurança das redes de comunicações de área local não-cabladas IEEE 802.11. Inicialmente faz-se uma introdução à problemática da segurança em redes de comunicações e descreve-se a norma IEEE 802.11, nomeadamente as diferentes camadas físicas e a sub-camada de controle de acesso ao meio. De seguida são estudados em detalhe os mecanismos de segurança incluídos nesta norma. Esta análise incluiu a validação em laboratório das várias vulnerabilidades conhecidas ao nível dos mecanismos de autenticação, confidencialidade e integridade das mensagens. São depois estudadas as tecnologias actualmente disponíveis para minimizar ou eliminar estas vulnerabilidades, nomeadamente os protocolos IEEE 802.1x e EAP (Extensible Authentication Protocol) para o processo de autenticação e o TKIP (Temporal Key Integrity Protocol) para o processo de confidencialidade, que em conjunto são conhecidos como especificação WPA (Wi-Fi Protected Access). O processo de autenticação foi também alvo de uma análise experimental. Finalmente são abordadas as tecnologias futuras actualmente em desenvolvimento por parte do grupo de trabalho IEEE 802.11i, para a implementação de segurança nas redes de comunicações não-cabladas IEEE 802.11.

keywords security, wireless networks, IEEE 802.11.

abstract

The wireless networks became, in the last years, very popular, being their utilization, nowadays, usual in the private and public areas. The mobility that is allowed to the user and the simplicity of implementation are the main factors that contributed to their massification. However, there are security problems related to the utilization of the wireless networks. Some of these risks are similar to those found in the conventional networks; others come from the widespread propagation of the radio waves. The attacks to the authentication and denial of service are the main problems of security related to the wireless networks.

This dissertation refers to the security aspects of the IEEE 802.11 wireless networks. Initially, it is made an introduction to the problematic of security in the wireless networks and the standard IEEE 802.11 is described, namely the different physical layers and the sub-layer of medium access control. Then, the security mechanisms, included in this standard, are studied in detail. This analysis includes the validation, in laboratory, of the widely known vulnerabilities at the level of authentication mechanisms, confidentiality and integrity of the messages. The technologies, nowadays available to minimize and to remove these vulnerabilities, are then studied, more specifically, the protocols IEEE 802.1x and EAP (Extensible Authentication Protocol), to the authentication process, and TKIP (Temporal Key Integrity Protocol), to the confidentiality process, that are together known as WPA (Wi-Fi Protected Access) specification. Finally, the future technologies, now in development by group work IEEE 802. 11i, for the implementation of security in the IEEE 802.11 wireless networks are mentioned.

Índice

Índice de Figuras

Índice de Tabelas

1 Introdução

1.1 Motivação e Enquadramento

O incremento exponencial ocorrido nos últimos anos relativamente à utilização de redes não-cabladas pode ser comparado ao crescimento da Internet verificado nas últimas décadas. Estas redes são cada vez mais consideradas por todos um complemento extremamente útil às redes convencionais, seja porque constituem hoje uma alternativa economicamente viável, seja pela capacidade que actualmente possuem de fornecerem taxas de transmissão comparáveis às das redes cabladas.

As redes não-cabladas, especialmente as redes IEEE 802.11 (*Institute of Electrical and Electronic Engineers*) tornaram-se comuns em redes de acesso para ambientes privados e públicos. A capacidade de mobilidade e a simplicidade de implementação tornaram estas redes muito populares quer para utilização doméstica, quer em sectores comerciais como sejam os ambientes de rede corporativa ou redes de acesso público, também designadas por *hotspots*.

A disponibilidade crescente deste tipo de redes exige uma elevada capacidade em fornecer comunicações seguras. Entenda-se comunicações seguras como comunicações que ocorrem em sistemas com serviços de segurança mínimos, nomeadamente, autenticação das entidades intervenientes na comunicação, confidencialidade dos dados transmitidos e integridade da informação.

Dadas as suas características específicas, nomeadamente o facto da transmissão de informação ser efectuada através de um meio físico não guiado, como é o caso da transmissão rádio, as redes não-cabladas devem possuir requisitos de segurança muito mais exigentes do que as redes convencionais. Os riscos, ou vulnerabilidades, numa rede não-cablada são o somatório das ameaças já existentes nas redes convencionais e dos novos riscos introduzidos pelas

possíveis falhas dos protocolos das redes não-cabladas. Quando uma transmissão é difundida através de ondas de rádio frequência, a intercepção e personificação tornam-se tarefas triviais para qualquer um que possua um dispositivo wireless. A constatação de tal facto exige a aplicação de mecanismos de protecção adicionais.

A norma IEEE 802.11 [2] especifica os mecanismos de segurança implementados nas redes não-cabladas de primeira geração através do protocolo WEP (*Wired Equivalent Privacy*). O objectivo do WEP era fornecer confidencialidade, autenticação e integridade dos dados transmitidos na rede. Infelizmente, o WEP falhou em todos os aspectos de segurança que se propunha solucionar.

Vários grupos de especialistas em segurança de redes de comunicações descobriram um vasto conjunto de problemas associados aos mecanismos implementados pelo WEP. Na realidade os problemas detectados comprometem todos os mecanismos de segurança propostos pela norma IEEE 802.11. Entre os ataques que podem ser desferidos incluem-se os seguintes: (i) ataques passivos para decifrar tráfego que circula na rede, o que compromete o mecanismo de confidencialidade; (ii) ataques activos que permitem a injecção de tráfego por parte de dispositivos não autorizados comprometendo os mecanismos de controlo de acesso e autenticação dos dados; (iii) ataques activos de intercepção e modificação dos dados transmitidos, o que compromete o mecanismo de integridade.

A divulgação pública das vulnerabilidades de segurança existentes na norma IEEE 802.11 levou ao surgimento de um novo movimento à escala global que se dedica à descoberta, mapeamento geográfico, divulgação pública e possível exploração maliciosa de redes não-cabladas, designado por *war driving*. Os resultados obtidos por este tipo de actividade demonstram inequivocamente, a falta de segurança existente em muitas das redes não-cabladas a operarem nos mais diversos cenários, indústria, comércio e serviços, pequenas redes não-cabladas domésticas, entre outros. Um *war drive*, efectuado e publicado pela empresa *Airdefense* [Link9] em Setembro de 2003, concluiu que 57 % das redes não-cabladas em funcionamento, não estavam protegidas com qualquer tipo de criptografia. Foram detectados um total de 1136 pontos de acesso, dos quais 650 não possuíam qualquer tipo de protecção de confidencialidade e cerca de 9% dos pontos de acesso possuíam as configurações por omissão proporcionadas pelos fabricantes.

Com estes e outros argumentos, os profissionais de segurança e os fabricantes de dispositivos wireless tomaram consciência de que a problemática de segurança associada às redes IEEE

802.11 era de extrema importância e optaram por "substituir" o mais rapidamente possível o protocolo WEP. Inicialmente foram introduzidas algumas medidas de protecção nas implementações comerciais do WEP, de modo a tornar inúteis as aplicações utilizadas pelos atacantes para quebrar os mecanismos de segurança. No entanto o problema persistia, e seria uma questão de tempo até que novas ferramentas de ataque fossem disponibilizadas.

A implementação de redes privadas virtuais foi uma das soluções adoptadas. Apesar de esta ser uma tecnologia bastante divulgada e aplicada, o facto de todo o tráfego wireless ser processado por uma *VPN gateway* implica imediatamente limitações em termos de desempenho[65], [66].

Para tentar solucionar alguns dos aspectos de segurança, mantendo a compatibilidade com os dispositivos wireless existentes surgiu o WPA (*Wi-Fi Protected Access*). O WPA é uma solução intermédia para a melhoria dos aspectos de segurança das redes IEEE 802.11. Esta solução adoptou dois novos protocolos para fornecerem mecanismos de confidencialidade e integridade: o TKIP (*Temporal Key Integrity Protocol*), que fornece confidencialidade, e o *Michael*, que fornece integridade de mensagens. O WPA implementa também um mecanismo de gestão e distribuição de chaves de cifra através da definição de uma hierarquia de chaves e de um protocolo de troca de mensagens designado por *4-Way Handshake*. Para os mecanismos de controlo de acesso e autenticação, o WPA adoptou, com as devidas alterações, um protocolo amplamente utilizado em redes convencionais, o IEEE 802.1x. A solução adoptada pelo WPA não restringe o processo de autenticação a um único protocolo. Este facto fez surgir um conjunto de protocolos de autenticação baseados em EAP *(Extensible Authentication Protocol)*: (i) o EAP-TLS *(EAP-Transport Layer Security)*; (ii) o protocolo proprietário LEAP (*Cisco Ligth EAP*); (iii) o PEAP (*Protected EAP*); (iv) e o EAP-TTLS (*EAP-Tunneled Transport Layer Security*). Os principais fabricantes de dispositivos wireless passaram a disponibilizar nos finais de 2003 actualizações de *software* que permitem a implementação destes novos mecanismos de segurança.

Em resposta à necessidade de uma norma que defina novos mecanismos de segurança que irão substituir o WEP surgiu o grupo de trabalho IEEE 802.11i, actualmente com um conjunto significativo de trabalho em fase de desenvolvimento [33]. Do trabalho desenvolvido até à data por este grupo destacam-se os mecanismos adoptados pelo WPA referidos anteriormente. Ao nível da arquitectura de controlo de acesso e de autenticação as diferenças entre o WPA e o IEEE 802.11i não são significativas. A principal diferença está no

mecanismo de confidencialidade que se baseia num novo protocolo mais robusto e seguro baseado no AES (*Advanced Encryption Standard*). A publicação desta nova norma irá definir uma arquitectura de segurança muito mais robusta e eficaz [52].

Durante esta fase de transição existe a necessidade de manter seguras as redes não-cabladas. Os mecanismos de segurança disponíveis actualmente, assim como os que estão em fase de desenvolvimento e serão adoptados nas futuras redes não-cabladas IEEE 802.11, são o tema principal desta dissertação.

1.2 Objectivos

A presente dissertação teve como principais objectivos:

- O estudo dos mecanismos de segurança implementados pela norma IEEE 802.11.
- Validação prática das vulnerabilidades apresentadas pelos mecanismos de autenticação, confidencialidade e integridade implementados pela norma IEEE 802.11.
- Estudo dos protocolos de autenticação EAP-TLS, LEAP, PEAP e EAP-TTLS e sua análise experimental.
- Estudo do trabalho em fase de desenvolvimento do grupo de normalização IEEE 802.11i e do conjunto de especificações baseadas neste, designado por WPA (*Wi-Fi Protected Access*).

1.3 Estrutura da Dissertação

A presente dissertação encontra-se estruturada na seguinte forma.

No capítulo 2 são apresentados conceitos básicos relacionados com a segurança em redes de comunicações, realçando os aspectos considerados relevantes para a compreensão dos assuntos abordados nesta dissertação.

O capítulo 3 descreve as redes de área local IEEE 802.11 e a sua relação com outras normas de rede do IEEE.

No capítulo 4 é feito um estudo detalhado dos mecanismos de autenticação, confidencialidade e integridade implementados pela norma IEEE 802.11. Neste capítulo é também efectuada a validação prática das vulnerabilidades existentes nos mecanismos estudados.

O capítulo 5 apresenta os novos mecanismos de segurança que pretendem substituir os mecanismos definidos na actual norma IEEE 802.11. Alguns destes mecanismos são analisados experimentalmente.

No capítulo 6 são apresentadas as conclusões finais do trabalho realizado e apontam-se tópicos relevantes para trabalho futuro.

2 Tecnologias de Segurança em Redes de Comunicações

A implementação de qualquer rede de comunicações comporta sempre um aumento de risco relativamente à utilização de máquinas isoladas, dado que os recursos de uma máquina ficam potencialmente acessíveis a utilizadores de outras.

Este capítulo pretende introduzir os conceitos de segurança fundamentais no âmbito das redes de comunicações. Na secção inicial são definidos os princípios básicos da segurança. Na secção 2.2 é apresentado um conjunto de medidas de segurança que devem ser tidas em consideração aquando da definição de uma política de segurança. Na secção 2.3 são abordados os principais elementos para o fornecimento de serviços de segurança como sejam a criptografia simétrica, a criptografia de chave pública e as funções de síntese. A secção 2.4 introduz o conceito de assinaturas digitais e os serviços de segurança que estas podem fornecer. Finalmente a secção 2.5 refere alguns dos principais métodos de gestão e distribuição de chaves, nomeadamente a distribuição de chaves públicas através de certificados digitais.

2.1 Conceitos

A segurança das redes de comunicações possui quatro objectivos fundamentais: a confidencialidade, a integridade, a disponibilidade e a utilização legítima. A confidencialidade é assegurada por um processo que evite a leitura da informação por parte de entidades ilegítimas. A integridade garante que os dados transmitidos ou armazenados não sofrem alterações não autorizadas durante o seu transporte ou existência. Com a disponibilidade pretende-se garantir que um determinado recurso, serviço ou informação, não é negado a utilizadores legítimos. A utilização legítima garante que determinados recursos não são acedidos por entidades não autorizadas.

De modo a alcançar os objectivos referidos anteriormente as medidas de segurança devem ser implementadas em conjunto. Não menos importante é a segurança física que deve ser implementada através do controlo de acesso físico aos sistemas e dispositivos de comunicações.

Para se compreender como pode ser implementada a segurança em redes de comunicações é necessário introduzir alguns conceitos elementares [59].

- Políticas de Segurança – São constituídas por um conjunto mais ou menos complexo de regras aplicadas a procedimentos e actividades relacionadas com as necessidades administrativas e de gestão da rede de comunicações à qual se aplicam.
- Autorização – Acção que atribui os direitos de acesso à rede. Normalmente está integrada numa determinada política de segurança que dite quem pode fazer o quê a que recursos.
- Ameaça – Identifica a entidade, pessoa, evento que constitui algum tipo de perigo. Este perigo pode estar relacionado com um ou vários dos objectivos de segurança fundamentais: confidencialidade, integridade, disponibilidade e uso legítimo.
- Ataque – É a concretização de uma determinada ameaça. Um ataque pode ser classificado em activo ou passivo. Um exemplo deste último tipo de ataque é a monitorização não autorizada ou *sniffing*. A intenção deste tipo de ataque é a obtenção de dados suficientes para proceder a um ataque activo que pode passar pela alteração ou destruição de dados e ou serviços disponibilizados na rede.
- Risco – Estimativa de custos das vulnerabilidades, tendo em conta o nível de probabilidade de um ataque ser bem sucedido. O risco é proporcional ao valor dos dados ou bens em causa e à probabilidade do ataque ser bem sucedido. Provavelmente não faz sentido dispender 80% do valor de um bem em segurança quando a probabilidade de ataque a este é diminuta.
- Medidas de protecção – Correspondem ao conjunto de políticas de segurança, controlo de acesso, procedimentos de monitorização e de reacção que diminuem a possibilidade de ocorrência de um ataque com sucesso. Entre as medidas de protecção

que devem ser implementadas destacam-se os mecanismos de autenticação, confidencialidade e integridade. Uma análise a estas e outras medidas de protecção é apresentada na secção 2.2.

A definição das medidas de protecção que devem ser integradas numa política de segurança é um dos aspectos fundamentais que deve ser objecto de estudo, pois o nível de segurança presente numa rede de comunicações depende em grande parte das opções tomadas.

2.2 Medidas de Protecção e Ameaças à Segurança

A segurança das redes de comunicações implica protecção de informação, ou seja deve prevenir e detectar acções não autorizadas, levadas a cabo por utilizadores ilegítimos. Esta "definição" de segurança foi-se tornando mais abrangente e actualmente inclui outros aspectos como sejam privacidade, confidencialidade e integridade. Esta definição revela a necessidade de avaliar a informação que se pretende proteger e determinar o seu valor por forma a desenvolver medidas protectoras adequadas.

As medidas ou acções a tomar para a protecção de dados em redes de comunicações podem ser classificadas nos seguintes grupos:

Prevenção: Medidas que previnam a informação a proteger de ser corrompida, alterada ou obtida de forma ilícita. Estas medidas de prevenção podem ir desde restringir o acesso físico às máquinas e servidores até regras de segurança restritivas no que respeita à manipulação das mesmas.

Detecção: As medidas de detecção devem permitir ao administrador obter informação relativamente à alteração dos dados, verificar se a informação que circula na rede foi danificada ou roubada, assim como saber quem o fez e o modo como tal acção foi efectuada. Existem actualmente um vasto conjunto de ferramentas que permitem detectar a corrupção dos dados, a presença de utilizadores ilegítimos na rede e a detecção de vírus nos sistemas.

Reacção: As medidas de reacção podem ser tomadas nos casos em que a informação foi de algum modo comprometida. Este tipo de medida permite a recuperação da informação, mesmo quando esta se perde ou corrompe.

As medidas acima apresentadas fazem todo o sentido, e tal como referido anteriormente são de extrema importância na definição da política de segurança a implementar. É necessário compreender como a informação que circula nas redes de comunicações pode ser comprometida. Só assim será possível tomar medidas para a proteger. Significa isto que o conjunto de medidas de segurança apresentadas anteriormente, devem ser implementadas em conjunto e ponderadamente.

No contexto desta dissertação, as medidas de protecção são, normalmente, chamadas serviços de segurança e podem ser divididos em quatro categorias:

- Serviço de Autenticação
- Serviço de Controlo de Acesso
- Serviço de Confidencialidade
- Serviço de Integridade de Dados
- Serviço de Não Repúdio.

A autenticação fornece mecanismos para assegurar a identidade dos intervenientes numa comunicações. A autenticação, confirma uma afirmação de identidade, seja por parte de um indivíduo ou aplicação, isto é verifica que um utilizador ou serviço é realmente quem dizem ser. Geralmente, isto é feito através de nomes associados a senhas. Métodos mais sofisticados recorrem ao uso de *smart-cards* e varrimentos à retina. O serviço de autenticação não garante ao utilizador o acesso aos recursos, isso é da responsabilidade dos mecanismos de autorização.

O serviço de controlo de acesso, pode ser definido como um mecanismo de protecção a acessos ilegítimos a determinado recurso da rede. Entenda-se por acesso, qualquer forma de utilização, modificação ou destruição. Este serviço não é mais do que um mecanismo usado para fornecer autorização.

Os mecanismos implementados para evitar que informações pertencentes a determinada comunicação não sejam lidos por entidades não autorizadas, são fornecidos pelo serviço de confidencialidade. Ou seja, este serviço previne a revelação não autorizada da informação.

O serviço de integridade previne a alteração dos dados. Entenda-se por alteração qualquer modificação, remoção, inserção ou reordenação dos dados originais.

O serviço de não–repúdio evita situações em que uma das partes envolvidas na comunicação possa, indevidamente, negar que esta mesma comunicação, alguma vez tenha ocorrido. Não se trata de uma protecção contra atacantes desconhecidos mas sim de utilizadores legítimos.

A razão da implementação dos serviços de segurança é pois proteger a informação mais sensível e valiosa de uma organização enquanto a torna disponível para quem a pode usar. Os agressores tentam normalmente prejudicar um sistema ou perturbar o seu normal funcionamento, explorando as vulnerabilidades dos sistemas usando várias técnicas, métodos e ferramentas. Os administradores de redes precisam pois de perceber os vários aspectos de segurança a ter em conta, para que as políticas e as medidas de segurança sejam definidas adequadamente.

As ameaças à segurança podem subdividir-se em duas categorias: maliciosas e não maliciosas. Os ataques não maliciosos vêm normalmente de utilizadores que não têm experiência com as tecnologias que operam ou não estão conscientes das várias ameaças à segurança existentes. Os ataques maliciosos são normalmente levados a cabo por ex-empregados, empregados descontentes ou outros agressores externos e que têm um objectivo específico a atingir.

De seguida apresentam-se os tipos de ameaças mais comuns às redes de comunicações. A enorme variedade de ameaças existentes actualmente pode ser dividida em quatro grupos principais:

- Reconhecimento.
- Acesso não Autorizado.
- Negação de Serviço.
- Ataques de Manipulação e Repetição.

O reconhecimento é a fase inicial de um qualquer ataque. Este tipo de ameaça prende-se com a descoberta de dispositivos e a monitorização das comunicações. O reconhecimento pode ser executado de forma activa ou passiva. A informação recolhida pode permitir ao atacante planear futuros ataques activos.

A violação de autorização permite a um atacante ganhar acesso a dispositivos ou serviços para os quais não está autorizado. O objectivo primeiro por parte do atacante é ganhar privilégios de acesso superiores aos que possui.

Os ataques de negação de serviço, DoS (*Denial of Service*) visam corromper ou tornar inoperacional um serviço ou uma rede. Este tipo de ameaça nega o serviço a utilizadores legítimos. Por exemplo, para determinado porto TCP são abertas um número elevado de conexões simultâneas, ou é gerada uma quantidade de tráfego elevada tornando o serviço inutilizável durante o ataque.

Exemplos de ameaças de manipulação e repetição, são os ataques conhecidos como *Man-in-the-Middle*, que consistem em deliberadamente interceptar e alterar partes de uma comunicação estabelecida entre duas entidades. Normalmente o intruso "coloca-se" entre uma comunicação que ocorre entre dois dispositivos, e faz acreditar ambas as entidades que estas estão a comunicar uma com a outra, quando na realidade, a comunicação é efectuada através do dispositivo atacante.

2.3 Introdução à Criptografia

Nesta secção pretende-se introduzir a terminologia básica da criptografia e apresentar os métodos mais implementados.

A criptografia é a ciência de protecção de dados e mensagens que fornece um conjunto de técnicas de codificação que permitem armazenar e transmitir dados e mensagens de forma considerada segura. As técnicas de criptografia moderna, incluindo muitas das utilizadas actualmente em redes de comunicações, são baseadas em técnicas desenvolvidas nos últimos 30 anos. Os livros [39], [40] permitem uma análise histórica à criptografia. Em [41] é apresentado uma análise técnica detalhada das técnicas criptográficas, do ponto de vista de redes de comunicações.

Tradicionalmente um sistema criptográfico integra os seguintes elementos (Figura 2-1):

- Algoritmo – Função matemática para transformação dos dados.
- Criptograma ou Texto Cifrado – Texto que sofreu a alteração imposta pelo algoritmo definido.

- Chave – Parâmetro que combinado com o algoritmo permite a transformação do texto plano em texto cifrado (operação de cifra), ou a operação inversa de transformar o criptograma em texto plano (operação de decifra).

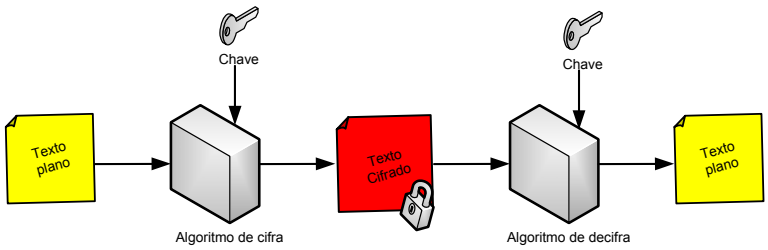


Figura 2-1 – Cifra/Decifra de um texto plano.

Uma utilização óbvia da criptografia é o serviço de confidencialidade. Um texto plano poderá ser transmitido por um canal inseguro desde que previamente cifrado, não tendo qualquer utilidade a quem o capture sem que possua a chave para o decifrar. No entanto como se verificará adiante neste capítulo, a criptografia não serve apenas para fornecer confidencialidade às comunicações.

Hoje em dia existem dois sistemas básicos de criptografia, ou criptosistemas: os sistemas de criptografia de chave simétrica e os sistemas de criptografia de chave assimétrica ou chave pública. As subsecções seguintes apresentam as características principais de cada um destes tipos de criptografia.

2.3.1 Criptografia de Chave Simétrica

Na criptografia tradicional, o emissor e receptor da mensagem conhecem e utilizam a mesma chave secreta; o emissor utiliza a chave para cifrar a mensagem e o receptor utiliza essa mesma chave secreta para decifrar a mensagem. Este método é conhecido por criptografia de chave secreta ou criptografia simétrica (Figura 2-2).

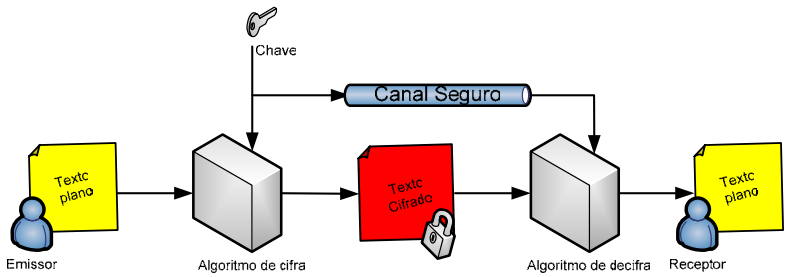


Figura 2-2 – Criptografia de chave simétrica.

O maior desafio que se coloca na utilização de criptografia de chave simétrica é fazer com que o emissor e o receptor concordem em utilizar uma chave secreta sem que ninguém a descubra. Se ambos se encontrarem em localizações físicas distintas, eles devem confiar num método de distribuição de chaves fiável para prevenir a revelação da chave secreta. Quem quer que seja que intercepte a chave em trânsito poderá mais tarde ler, modificar e forjar todas as mensagens cifradas ou autenticadas usando essa chave.

Apesar destes problemas, a criptografia com chave secreta apresenta algumas vantagens relativamente à criptografia de chave pública nomeadamente no que respeita à rapidez dos algoritmos de cifra e decifra. No entanto, tem como principal desvantagem, o facto de requerer $n \times (n-1)/2$ chaves para n interlocutores, assim como o problema da distribuição e gestão de chaves de modo a manter o seu secretismo.

Cifras de Blocos

Uma cifra de blocos é um tipo de algoritmo de cifra com chave simétrica que transforma um bloco de tamanho fixo de texto plano num bloco de texto cifrado do mesmo tamanho. Esta transformação decorre sobre a acção da chave secreta fornecida pelo utilizador. A decifra é efectuada aplicando a transformação inversa aos blocos de texto cifrado usando a mesma chave secreta.

O primeiro algoritmo de cifra por blocos a ser aplicado em larga escala foi o DES (*Data Encryption Standard*) [42], publicado pela primeira vez em 1977 e actualizado em 1993. Este algoritmo foi utilizado inicialmente para protecção de informação governamental americana e transações financeiras na indústria. Desde então tem sido adoptado largamente noutras áreas de aplicação.

O DES é um algoritmo de cifra em que os blocos têm 64 bits e a chave tem 56 bits [56]. Tal como ocorre em outros algoritmos de cifra por blocos, o DES deve operar num determinado modo de cifra sempre que aplicado a mensagens de tamanho superior a 64 bits. O documento [43] especifica vários modos de cifra para utilização com o DES, incluindo um para autenticação.

Actualmente o ataque mais efectivo ao DES é um ataque de força bruta de procura exaustiva de chaves. Tal facto deve-se à tecnologia de computação actual que permite atacar de forma fácil qualquer bloco de cifra com chaves de 56 bits. No entanto este não é o único método de ataque possível. Em 1993 foi anunciado por Biham e Shamir uma técnica conhecida por

criptoanálise diferencial [44] [57] e em 1994 foi desenvolvido por Mitsuru Matsui outro ataque conhecido por criptoanálise linear [58]. Recorrendo a este último método em 1999 conseguiu-se recuperar uma chave DES em apenas 22 horas [Link11].

Actualmente é reconhecido pela comunidade científica que o DES não é um algoritmo aceitável para protecção de documentos nem para aplicações de comércio electrónico. Reconhecendo tais vulnerabilidades, o departamento do comércio americano lançou em 1997 o projecto para a criação de um algoritmo mais forte que o DES designado por AES (*Advanced Encryption Standard*).

Em Novembro de 2001, o NIST anunciou a aprovação do AES, também conhecido como algoritmo de Rijndael. O AES é uma cifra de blocos que processa blocos de 128, 192 ou 256 bits e que pode operar com chaves de tamanho 128, 192 e 256 bits.

A Tabela 2-1 compara as principais características das cifras de blocos AES e DES.

	DES	**AES**
Comprimento Chave	56 bits	128, 192 ou 256 bits
Tipo de Cifra	Simétrica por blocos	Simétrica por blocos
Tamanho de Bloco	64 bits	128, 192 ou 256 bits
Nº Chaves possíveis	2^{56}	2^{128}, 2^{192}, 2^{256}
Tempo Necessário para obter a Chave	22 horas	$5x10^{21}$ anos (para chave de 128 bits)
Segurança	Considerado inadequado	Considerado seguro

Tabela 2-1 – Comparação entre DES e AES.

Mais informações relacionadas com algoritmos criptográficos e seus princípios de funcionamento pode ser encontrada em [LINK 8].

Cifra Contínua

Uma cifra contínua (*Stream Cipher*) consiste normalmente numa operação XOR, bit a bit, de uma chave contínua (*Keystream*) com um texto plano ou criptograma, consoante se esteja a cifrar ou decifrar. A chave contínua é produzida por um gerador de chaves que opera em função de uma chave de tamanho fixo e reduzido (Figura 2-3).

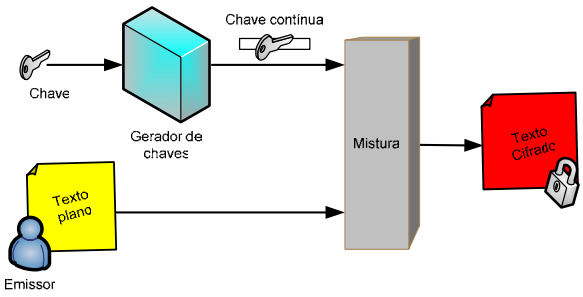


Figura 2-3 – Cifra contínua.

Até à presente data ainda não surgiu nenhuma cifra contínua que se tenha imposto "de facto" como *standard*. A cifra contínua mais utilizada actualmente é o RC4.

Algoritmo de Cifra Contínua RC4

O RC4 é um algoritmo de cifra contínua desenvolvido em 1987 por Ron Rivest para a *RSA Data Security* (actualmente a *RSA Security*). Uma das vantagens do RC4 é a relativa simplicidade de implementação e a não utilização de operações matemáticas complexas que tornam os algoritmos lentos, como é o caso da multiplicação. Este algoritmo pode considerar-se suficientemente forte se utilizado de forma correcta.

A cifra contínua RC4 é composta por dois componentes, representados na Figura 2-4: um algoritmo de mistura de chaves, KSA (*Key Scheduling Algorithm*), e um algoritmo que gera a chave contínua, designado por PRGA (*Pseudo Random Generation Algorithm*).

KSA(K, l)	*PRGA (l)*
#Preenche o vector S-Box com valores de 0 *a* 255 *for i*=0 *to* 255 *S*[*i*]= *i* *end for* *j*=0 *#Scrambling S-box usando a chave K* *for i*=0 *to* 255 *j*=*j*+*S*[*i*]+*K*[*i mod l*] *mod* 256 *swap*(*S*[*i*], *S*[*j*]) *end for*	*i*=0 *j*=0 *# Ciclo de geração de z* *for K*=0 *to l*-1 *i*=(*i*+1) *mod* 256 *j*=(*j*+*S*[*i*]) *mod* 256 *Swap*(*S*[*i*], *S*[*j*]) *z*=*S*[(*S*[*i*]+*S*[*j*]) *mod* 256] *return byte chave contínua z* *end for*

Figura 2-4 – Algoritmos KSA e PRGA.

O RC4 utiliza nos processos de cifra e decifra o conceito de estados. O valor do estado é mantido numa matriz 8×8 designada por *S-Box*, representada no pseudocódigo como *S* e o seu tamanho é igual a 256 bytes. A chave de cifra é representada por *K* e tem comprimento *l*. O algoritmo de mistura de chaves, KSA, é função do valor e do comprimento da chave *K*. Na fase inicial do KSA o vector *S* é preenchido com valores de 0 a 255. O processo de mistura

(*Scrambling*) do vector *S* ocorre na segunda fase do algoritmo, após 256 ciclos de operações de adição e troca (*swap*) o vector de estado *S* encontra-se perfeitamente misturado. Após o cálculo do vector de estado *S*, o RC4 passa ao processo de cifra. A geração de cada byte da chave de cifra contínua (representado na Figura 2-4 por z) recorre a um conjunto de operações simples de soma e troca dos elementos do vector *S*.

Teoricamente, o RC4 não é uma cifra completamente segura porque o processo de obtenção da chave contínua não gera valores totalmente aleatórios. Os valores gerados pelo algoritmo PRGA são valores pseudo-aleatórios. No entanto o RC4 pode ser considerado suficiente seguro em aplicações wireless, se aplicado correctamente.

Vector Inicialização utilizado no RC4

A utilização de um vector de inicialização, IV (*Inicialialization Vector*), pretende solucionar um problema inerente à utilização de algoritmos de chave simétrica de cifra contínua com tamanho de chave secreta fixo. Neste tipo de algoritmo se a chave de cifra contínua for utilizada mais do que uma vez para cifrar o mesmo texto plano resulta que o texto cifrado será o mesmo. Este resultado permite a um atacante determinar matematicamente o valor da chave contínua. A solução para este problema passa pela utilização de um vector de inicialização. Em vez de utilizar uma chave secreta de tamanho fixo, esta pode ser combinada com um vector de inicialização.

Uma vez que o valor do IV é alterado em cada pacote, a chave efectivamente utilizada para cifrar o pacote varia para cada pacote cifrado. Assim mesmo que o texto plano seja o mesmo, o texto cifrado é sempre diferente. Em teoria, o conhecimento do IV não tem qualquer utilidade sem o conhecimento da chave secreta. Para que tal seja verdade, o mesmo IV não deve ser utilizado duas vezes com a mesma chave secreta. Este aspecto será abordado posteriormente aquando da análise dos ataques aos mecanismo de segurança implementados na norma IEEE 802.11.

2.3.2 Criptografia de Chave Pública

Para resolver o problema da gestão de chaves associado à criptografia de chave simétrica, Whitfield Diffie e Martin Hellman [45] introduziram em 1976 o conceito de criptografia de

chave pública, ou assimétrica. Os sistemas criptográficos com chave pública têm duas utilizações primárias: confidencialidade e assinaturas digitais.

Segundo este sistema, cada pessoa possui um par de chaves, uma chamada de pública e outra de privada. A chave pública é do conhecimento geral enquanto que a chave privada permanece secreta. Basicamente tem-se que os dados cifrados com uma chave (chave pública) são decifrados com uma outra chave distinta (chave privada), ao contrário das chaves de cifra simétricas, onde a mesma chave é utilizada para cifrar e decifrar. As chaves utilizadas neste tipo de cifra estão relacionadas matematicamente, conseguindo uma decifrar aquilo que a outra cifrou. Uma das propriedades principais deste tipo de sistemas é que dada uma das chaves é computacionalmente impossível determinar a outra.

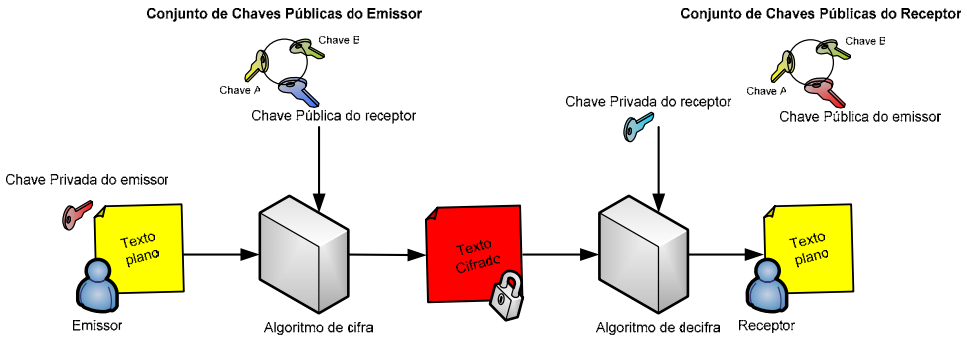


Figura 2-5 – Criptografia de chave pública.

Tome-se como exemplo a Figura 2-5. Cada utilizador possui um par de chaves, uma chave pública e uma chave privada distintas. A chave pública é disponibilizada aos utilizadores com quem se pretende comunicar. Assim é possível ao emissor enviar uma mensagem cifrada com a chave pública do receptor, com a garantia de que a mensagem é apenas decifrada por este, utilizando a sua chave privada.

Por outro lado, uma mensagem cifrada com a chave privada do emissor, apenas pode ser decifrada por quem possuir a chave pública do emissor, assegurando-se assim a origem da mensagem. Este é o mecanismo básico para as assinaturas digitais, discutidas mais adiante.

Uma vantagem da cifra de chave pública em relação aos sistemas de cifra simétrica é a possibilidade de publicitar uma das chaves, a chave pública, pelo que só serão necessárias n chaves para n interlocutores. Uma desvantagem que este tipo de cifra apresenta em relação à cifra de chave simétrica é o facto de o seu desempenho ser mais lento, por utilizar um processo algorítmico mais complexo. Por exemplo, o RSA [46] desenvolvido por Ronald

Rivest, Adi Shamir e Leonard Adleman em 1997 é aproximadamente 1500 vezes mais lento que o DES.

Por este motivo muitas vezes estes dois tipos de cifra operam em conjunto. Na prática a cifra dos dados é feita com um algoritmo de cifra simétrica, 3DES por exemplo, e é utilizado um algoritmo de chave pública para cifrar a chave simétrica, por exemplo o RSA. Neste caso a criptografia de chave pública é utilizada como canal seguro para troca de chaves simétricas. Deste modo é possível tirar partido do desempenho e robustez da cifra simétrica sem a desvantagem de ter de transmitir a chave secreta por um canal seguro externo. Este método é conhecido por criptografia híbrida e está representado na Figura 2-6.

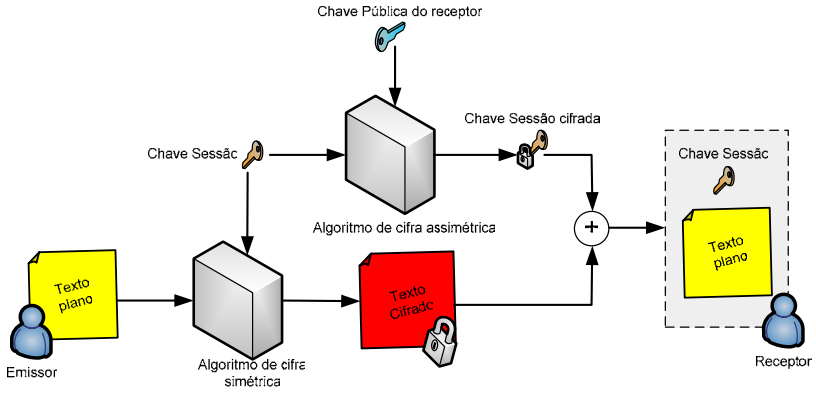


Figura 2-6 – Criptografia híbrida.

O emissor gera uma chave de sessão simétrica com a qual cifrará a mensagem. Cifra também com a chave pública do receptor essa chave de sessão. Finalmente envia ao receptor estes dois elementos cifrados. Para decifrar a mensagem o receptor deverá utilizar a sua chave privada em primeiro lugar para decifrar a chave de sessão. Com esta última, o receptor decifra a mensagem e tem acesso ao seu conteúdo.

2.3.3 Funções de Síntese

Uma função de síntese, ou função de *hash*, converte um bloco de dados de tamanho variável, num resultado de comprimento fixo, normalmente muito mais pequeno que o bloco de dados original.

Este tipo de funções são muito utilizadas em processos de criptografia assimétrica. Como principais propriedades destacam-se as seguintes: (i) é computacionalmente impraticável

encontrar dois blocos de dados diferentes cujo resultado de *hashing* seja igual; (ii) dada uma síntese é impraticável obter o bloco de dados que lhe deu origem.

As funções de síntese são também conhecidas por *Message Digest* ou *Fingerprint Algorithms*. Exemplos de implementações concretas de funções de síntese são o MD5 (*Message Digest 5*) definido no [RFC1321] da RSA Data Security, e o SHA-1 (*Secure Hashing Algorithm 1*) [47].

2.4 Assinaturas Digitais

As assinaturas digitais são um mecanismo de prova da origem dos dados (a assinatura é a do emissor dos dados) e um mecanismo de integridade dos dados transmitidos (a assinatura apenas é válida para os dados originais). A assinatura digital permite garantir o não repúdio, ou seja comprova o envio dos dados, por forma a proteger o receptor da negação de envio dos dados por parte do emissor.

Uma assinatura digital é criada e verificada criptograficamente. Um método de assinatura digital recorrendo a criptografia de chave pública consiste simplesmente em cifrar a mensagem com a chave privada do emissor. Neste caso o receptor da mensagem utiliza a chave pública do emissor para verificar a validade da assinatura digital. Com este processo a operação de cifra e decifra é aplicada a todo o texto plano da mensagem gerando uma assinatura do mesmo tamanho que os dados o que faz duplicar o volume de dados a transmitir.

Para optimizar este esquema introduziu-se uma função de síntese no processo de assinatura digital (Figura 2-7). Mais especificamente, a assinatura digital é o resultado da cifra da síntese da mensagem, efectuada com a chave privada. Qualquer alteração na mensagem produz uma síntese diferente e não é possível obter a mensagem através do valor da síntese.

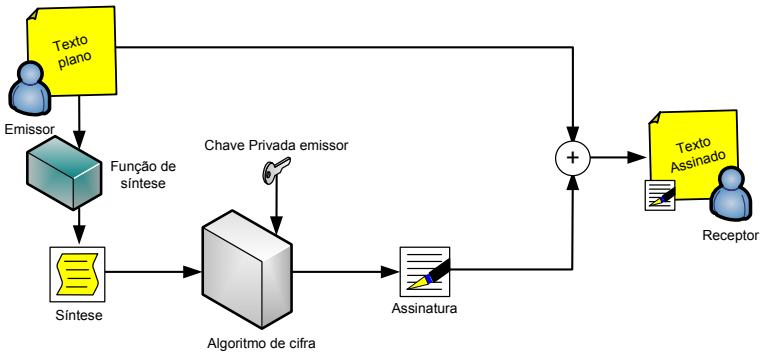


Figura 2-7 – Assinatura digital de um documento.

Ao observar a Figura 2-7, verifica-se que toda a mensagem em texto plano é submetida a uma função de síntese. De seguida a síntese da mensagem é cifrada com a chave privada do emissor, sendo o resultado designado por assinatura. O resultado enviado ao receptor da mensagem não é mais do que uma pequena sequência de bytes (assinatura) adicionada no fim da mensagem.

Quando o receptor recebe a mensagem, também cria uma síntese dessa mensagem. Decifra com a chave pública do emissor o valor da síntese enviada e compara ambos. Se forem idênticos significa que o conteúdo da mensagem não foi alterado, garantindo-se assim os serviços de integridade e autenticação. O processo de verificação da assinatura digital é representado na Figura 2-8.

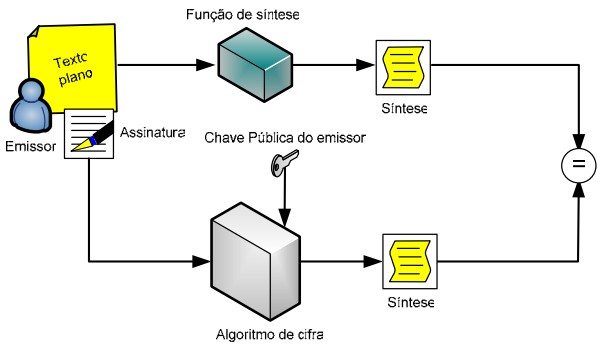


Figura 2-8 – Verificação da assinatura digital.

De referir que a assinatura digital por si só apenas fornece autenticidade e integridade, não confidencialidade. Para tal os dados terão de ser cifrados separadamente. Quando a confidencialidade e assinatura digital dos dados são combinados, obtém-se um serviço designado por envelope digital assinado (Figura 2-9).

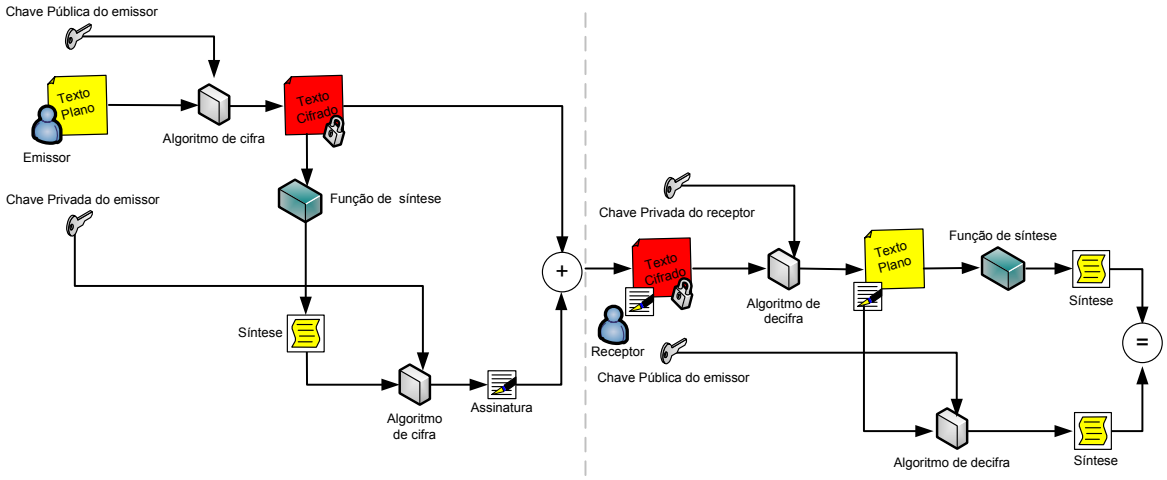


Figura 2-9 – Envelope digital assinado.

A robustez de uma assinatura digital depende da qualidade da função de síntese e da cifra assimétrica. Se a função de síntese puder ser invertida, então os dados originais, poderão ser alterados sem invalidar a assinatura digital. Se a cifra assimétrica não for suficientemente forte, então a probabilidade da assinatura ter sido feita por utilizadores ilegítimos aumenta.

2.5 Gestão e Distribuição de Chaves

Um grande número de mecanismos de segurança depende da utilização de chaves. Deste modo torna-se de extrema importância garantir a geração, distribuição e manutenção de chaves de modo seguro.

O processo de gestão de chaves inclui um conjunto diversificado de tarefas, tais como:

- Assegurar as propriedades desejáveis aquando da sua produção, por exemplo a aleatoriedade das chaves de cifra.
- Divulgar as chaves, de forma segura, a todas as entidades que delas legitimamente necessitem.
- Proteger as chaves contra substituição ou exposição ilegítima a terceiros.

O método de gestão e distribuição de chaves é dependente do tipo de chaves que se pretende gerir, isto é, conforme se trata de chaves simétricas ou chaves assimétricas.

2.5.1 Distribuição de Chaves Simétricas

Os requisitos de distribuição de chaves de um sistema criptográfico simétrico são diferentes dos requisitos dos sistemas assimétricos ou de chave pública. Nos sistemas simétricos, é necessário fazer chegar uma chave secreta a dois intervenientes de forma a estes poderem proteger as comunicações entre si. Esta chave deverá ser distribuída de forma a continuar secreta para todos os restantes sistemas da comunidade em questão.

Existem vários métodos de distribuição de chaves simétricas. No entanto apresentam-se apenas dois, o método de distribuição Diffie-Helman e o método de distribuição com RSA.

Distribuição Diffie-Hellman

Um método de duas entidades acordarem chaves simétricas de sessão é o método de Diffie-Hellman, desenvolvido por Diffie e Hellman em 1976 [48]. Esta técnica permite a duas entidades a criação e troca de uma chave secreta que poderá ser utilizada como chave de sessão num qualquer sistema de cifra simétrica.

O protocolo tem dois parâmetros de sistema p e g. Ambos os parâmetros são públicos e podem ser utilizados por todos os utilizadores de um sistema. O parâmetro p é um número primo maior que 2 e o parâmetro g (normalmente chamado de gerador) é um inteiro menor que p.

O parâmetro g possui ainda a seguinte propriedade:

- Para cada número n entre 1 e p-1 inclusivé, existe uma potência k de g tal que $n = g^k \bmod p$

De seguida apresenta-se um exemplo que pretende descrever o processo de estabelecimento de chave Diffie-Hellman.

1. O sistema emissor e o sistema receptor utilizam os valores 17 e 12 para os parâmetros públicos p e g respectivamente.
2. Cada entidade gera uma chave privada menor que p-1. Por exemplo o emissor gera x_a=9 e o receptor gera x_b=3.
3. Em seguida cada entidade deriva as chaves públicas $n_{emissor}$ e $n_{receptor}$ utilizando os parâmetros p e g e as suas chaves privadas. O valor público do emissor é dado

por $n_{emissor} = g^{x_a} \bmod p$ e o do receptor é dado por $n_{receptor} = g^{x_b} \bmod p$ $\Rightarrow n_{emissor}$=5; $n_{receptor}$=1.

4. Nesta fase as entidades trocam as chaves públicas o que permitirá a cada uma determinar a chave de sessão. A chave de sessão é calculada no emissor com $Z_{emissor} = (n_{receptor})^{x_a} \bmod p$ e no receptor com $Z_{receptor} = (n_{emissor})^{x_b} \bmod p$ e $Z_{emissor} = Z_{receptor}$. Neste caso $Z_{emissor} = Z_{receptor} = 6$.

A norma ANSI X9.42 descreve variantes deste modelo de forma adaptada a vários cenários de aplicação.

Distribuição com RSA

Os sistemas de chaves públicas também podem ser usados para implementar sistemas de distribuição de chaves simétricas de sessão. Esta abordagem é particularmente atraente para cenários onde é complicado manter pares de chaves para todos os intervenientes num centro de chaves da confiança de toda a comunidade.

O RSA fornece um mecanismo de confidencialidade nas comunicações. Assim, basta cada entidade estabelecer o seu par de chaves privada/pública para poder trocar chaves secretas simétricas com todas as outras entidades intervenientes na comunicação.

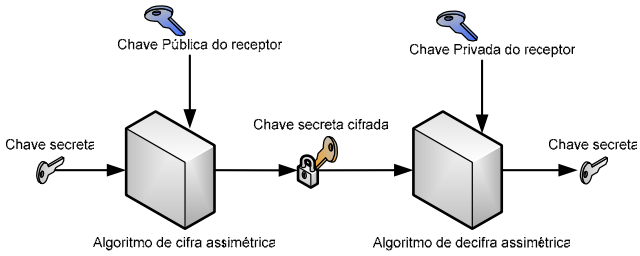


Figura 2-10 – Distribuição com RSA.

Como se verifica pela Figura 2-10, a chave secreta é cifrada recorrendo ao algoritmo assimétrico RSA. Este cifra com a chave pública do receptor a chave secreta simétrica. Para decifrar a chave secreta o receptor deverá utilizar a sua chave privada.

2.5.2 Distribuição de Chaves Públicas

O principal problema associado aos mecanismos de cifra assimétrica é o da distribuição das chaves públicas. No processo de distribuição das chaves públicas, não é necessário garantir confidencialidade. No entanto devem ser assegurados os serviços de autenticidade e integridade na transferência destas chaves entre as entidades intervenientes.

Não deve ser possível que uma entidade ilegítima possa substituir ou alterar a chave pública que está a ser anunciada. Se tal acontecesse um emissor poderia ser levado a crer que determinada chave pública correspondia ao seu titular legítimo, quando na realidade corresponderia por exemplo a uma chave privada na posse de uma entidade atacante.

Têm sido propostas um conjunto vasto de técnicas para a distribuição de chaves públicas. No entanto apresentam-se apenas dois, anúncio público e certificados digitais.

Anúncio Público

O próprio utilizador anuncia a sua chave pública, através dos mais diversos meios, nomeadamente, disponibilizando a chave na sua página pessoal ou incluindo-a em cada mensagem de correio electrónico que envie. Esta abordagem apesar de conveniente, não é considerada segura, pois qualquer utilizador mal intencionado pode falsificar a identidade de outro.

Certificados Digitais

Uma nova abordagem, que recorre a certificados digitais, foi proposta por Kohnfelder [50]. O conceito de certificado digital pode ser definido como uma estrutura de dados que tem como objectivo associar uma chave pública ao seu titular e garantir a autenticidade da mesma [49].

Um certificado digital, não é mais do que um conjunto de dados assinados digitalmente contendo, tipicamente, a chave pública de uma entidade. A assinatura digital garante, por um lado a integridade da chave pública, por outro, a sua autenticidade. Os certificados digitais, obedecem, em geral, à recomendação X.509 do ITU-T.

Para que este método de distribuição de chaves públicas funcione correctamente têm que se verificar os requisitos inicialmente propostos por Kohnfelder: (i) qualquer entidade pode ler um certificado, para obter o nome e chave pública do titular; (ii) qualquer entidade pode verificar a legitimidade da entidade certificadora CA (*Certification Authoritie*) que emitiu o

certificado; (iii) apenas a componente entidade certificadora, pode actualizar, revogar ou emitir um certificado.

Posteriormente Denning [51] acrescentou o seguinte requisito: qualquer entidade, pode verificar a validade do certificado. Entenda-se validade, como sendo um período temporal no qual o certificado é válido, expirando a sua validade no final do prazo definido no certificado.

O formato dos certificados digitais são definidos na norma X.509. Esta especificação faz parte da série de recomendações X.500 que definem um serviço de directório. Até à data foram definidas três versões de certificados digitais.

Em 1998 surgiu a versão X.509 v1. Esta versão não era suficientemente flexível dada a impossibilidade de adicionar novos atributos ao certificado. A versão X.509 v2 surge algum tempo mais tarde, sendo introduzido o conceito de identificador único para a entidade titular e para a entidade emissora do certificado. Esta versão não foi no entanto muito utilizada. A versão actual (desde 1996) e mais divulgada é a versão X.509 v3. Esta versão suporta o conceito de extensão, onde qualquer um pode definir uma nova extensão e inclui-la nos certificados.

A versão actual dos certificados digitais X.509, apresenta um vasto conjunto de dados associados ao titular do certificado e à CA que o emitiu. Os dados são os seguintes [RFC2459]:

- Versão do certificado – Verifica a versão do formato do certificado digital.
- Número de série – Identificador único do certificado emitido pela CA.
- Algoritmo de assinatura – Indica qual o algoritmo utilizado para assinar digitalmente o certificado.
- Nome do emissor – Fornece o nome da CA que gerou e assinou o certificado.
- Período de validade – Consiste em duas data que definem o período temporal de validade do certificado.
- Identificação do titular – Indica o nome da entidade a quem o certificado pertence.
- Chave pública – Chave pública do titular do certificado.
- Identificador único da CA – Fornece um valor que identifica inequivocamente a CA que gerou e emitiu o certificado no caso do nome da CA ser partilhado por diferentes entidades.

- Identificador único da entidade – Valor que identifica inequivocamente a entidade a quem o certificado pertence caso o nome da entidade seja partilhado por diferentes entidades.

- Extensões – Consiste num conjunto de campos que permitem incluir informação adicional que as versões 1 e 2 não implementam. Algumas extensões vulgarmente utilizadas são a *Key Usage*, que limita o uso das chaves a determinadas funcionalidades como por exemplo apenas assinaturas, *Signing Only*, e a extensão *Subject Alternative Name*, que permite que outras identificações como o nome de DNS, o endereço de correio electrónico, ou o endereço IP possam ser associadas à chave pública do certificado.

- Assinatura – Contém a síntese relativa a todo o certificado. Este código é cifrado com a chave privada da CA que gerou o certificado. Também indica o algoritmo utilizado na criação da assinatura. Ao gerar esta assinatura a CA está a certificar a integridade da informação apresentada nos dados, bem como a ligação entre a chave pública e o titular do certificado.

A Figura 2-11 apresenta um exemplo da informação disponibilizada por um certificado emitido por uma CA. Neste caso a CA é a *VeriSign* e o certificado é o da própria CA.

Figura 2-11 – Exemplo de um certificado.

Com a informação apresentada neste capítulo tem-se presente os conceitos e mecanismos mais utilizados na segurança de redes de comunicações. Pode assim avançar-se para os aspectos específicos relacionados com as redes de comunicações de área local não-cabladas IEEE 802.11. No próximo capítulo é apresentada a tecnologia associada a este tipo particular de redes de comunicações.

3 Redes não-Cabladas IEEE 802.11

Actualmente é possível identificar uma quantidade considerável de cenários de redes não-cabladas que podem ser classificados em três grupos de acordo com a área de cobertura proporcionada: redes não-cabladas de ampla cobertura WWAN (*Wireless Wide Area Network*), redes não-cabladas de cobertura local WLAN (*Wireless Local Area Network*) e redes não-cabladas de cobertura pessoal WPAN (*Wireless Personal Area Network*). O primeiro grupo inclui as redes celulares de segunda geração, GSM, CDPD, e Mobitex. Tecnologias como IEEE 802.11, HomeRF ou HiperLan surgem em redes não-cabladas de cobertura local. O grupo de redes WPAN engloba tecnologias wireless como Bluetooth e Infravermelhos.

Este capítulo fornece uma visão histórica das tecnologias associadas a redes de cobertura local, em particular da tecnologia IEEE 802.11 e a sua relação com outras normas de rede do IEEE. Na secção 3.2 são abordados alguns aspectos da camadas de controle de acesso ao meio especificada na norma IEEE 802.11. A arquitectura definida no protocolo IEEE 802.11 é abordada na secção 3.3. A secção 3.4 descreve os detalhes fundamentais do protocolo IEEE 802.11. Finalmente a secção 3.5 apresenta um cenário descritivo do princípio de funcionamento de uma rede não-cablada IEEE 802.11.

3.1 Descrição Histórica

De forma idêntica às redes de área local, muitas das WLANs iniciais eram proprietárias. Antes de 1998 muitas dessas redes não-cabladas eram caracterizadas pelo seu elevado custo, largura de banda limitada e pela ausência de uma norma que permitisse a sua aplicação a nível mundial. Estas redes foram utilizadas nas mais diversas áreas, nomeadamente na industria e comércio. Foi nesta conjuntura que o IEEE iniciou o desenvolvimento de normas para as WLANs.

3.1.1 Família de Protocolos IEEE 802.11

O IEEE iniciou o esforço na criação de normas na área de redes não-cabladas em Maio de 1991, com um pedido de autorização para a criação de um grupo de trabalho designado por IEEE 802.11. A primeira norma definida por este grupo é designada por IEEE 802.11-1997. Esta norma descreve uma camada de controle de acesso ao meio MAC (*Media Access Control*) e três camadas físicas distintas que permitem taxas de transmissão de 1 a 2 Mbps, duas baseadas em tecnologias de radio frequência (RF) e uma terceira baseada em infravermelhos.

Com o decorrer do tempo foram introduzidas várias actualizações à versão inicial, surgindo em 1999 uma nova versão, referida como IEEE 802.11-1999. Esta última versão define duas novas camadas físicas que permitem taxas de transmissão de 5.5 Mbps ou 11 Mbps.

Na secção 3.1.2 é apresentada uma descrição do modelo básico do protocolo IEEE 802.11.

3.1.2 Modelo do Protocolo IEEE 802.11

A norma IEEE 802.11 adere à arquitectura genérica das redes IEEE 802. Nas especificações IEEE 802 uma das camadas protocolares chave é a camada LLC (*Logical Link Control*). Esta cria a transparência entre os protocolos da camada de rede, como o IP, e os mecanismos de nível inferior, a camada de ligação de dados e a camada física.

A camada protocolar LLC isola os vários protocolos dos protocolos da camada de rede, permitindo assim a execução de aplicações, de protocolos de nível superior e de mecanismos de gestão sobre IEEE 802.11 de forma transparente. Esta camada protocolar torna o IEEE 802.11 indistinto de outros protocolos IEEE 802. À semelhança da arquitectura IEEE 802, o modelo do protocolo IEEE 802.11 inclui duas camadas protocolares abaixo da LLC, a camada de controle de acesso ao meio e a camada física, tal como se verifica na Figura 3-1.

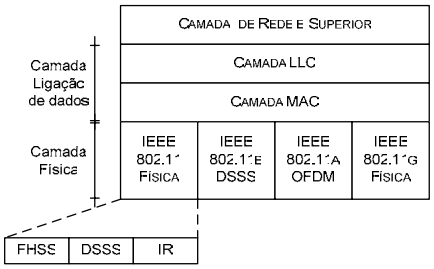


Figura 3-1 – Modelo IEEE 802.11.

3.1.3 Características do Protocolo Inicial IEEE 802.11

O protocolo IEEE 802.11 inicial aprovado em 1997 introduziu a utilização de uma nova camada de acesso ao meio e especificou novas camadas físicas. Define ainda a arquitectura lógica que permite a comunicação entre os dispositivos que suportam o protocolo. Foi introduzido a noção de comunicação de dispositivos com um ponto de acesso, AP (*Access Point*), o qual por sua vez estabelece ligação a uma rede cablada

As camadas físicas definidas no protocolo IEEE 802.11-1997 permitem taxas de transmissão desde 1 Mbps até um máximo de 2 Mbps. O primeiro valor é obrigatório enquanto que o valor máximo é opcional, dependendo do tipo de mecanismo de transmissão utilizado. Tal como referido inicialmente existem três tecnologias de transmissão do nível físico que foram aprovados na norma IEEE 802.11. Dois desses métodos são baseados em tecnologias de radio frequência, o terceiro é baseado em infravermelhos [54].

Quanto aos mecanismos baseados em RF estes empregam duas técnicas de espalhamento espectral: espalhamento espectral com salto de frequência FHSS (*Frequency Hopping Spread Spectrum*) e espalhamento espectral de sequência directa DSSS (*Direct Sequence Spread Spectrum*). O mecanismo de transmissão definido para a camada física baseada em infravermelhos opera tipicamente com um alcance até 10 m e não requer que o emissor e o receptor estejam em linha de vista. Tipicamente é utilizado em ambientes interiores tais como salas de aula, dada a incapacidade da luz infravermelha penetrar em obstáculos e igualmente devido à elevada atenuação que esta facilmente sofre. A tecnologia IR não é tão popular como as técnicas de FHSS e DSSS dado que não possui as características desejáveis à utilização, nomeadamente de utilizadores móveis.

Um dos requisitos do IEEE 802.11 prendia-se com a necessidade de compatibilidade a nível global, o que motivou a utilização de um espectro de frequências em torno dos 2.4 GHz espectro esse não licenciado e disponível em praticamente todo o mundo. Esta banda de frequências é conhecida por banda ISM (*Industrial, Scientific, and Medical*). A largura de banda disponível neste espectro de frequência é variável, dependendo o seu valor de país para país. Por exemplo, nos Estados Unidos estão disponíveis 79 MHz. A tecnologia baseada em infravermelhos opera na gama dos 850 nm com um nível de potência de transmissão máximo de 2 watts de potência óptica.

3.1.4 Características do Protocolo IEEE 802.11b

Uma das desvantagens do protocolo inicial IEEE 802.11-1997 foi a taxa de transmissão que poderia ser alcançada. O limite máximo de 2 Mbps era demasiado baixo para certas situações em que a necessidade de transmitir blocos de dados de tamanho elevado era um requisito, como por exemplo em aplicações multimédia. Esta necessidade resultou no aparecimento do protocolo IEEE 802.11b, aprovado em Setembro de 1999.

Em Outubro de 1997, o comité executivo do IEEE 802 permitiu a criação de dois novos projectos para incrementar a taxa de transmissão da norma IEEE 802.11 inicial. Isto levou ao desenvolvimento de dois protocolos o IEEE 802.11b e o IEEE 802.11a. O último será abordado na secção 3.1.5. Ambos mantêm a mesma camada de controlo de acesso ao meio, e apenas diferem na camada física.

A melhoria mais importante introduzida pelo IEEE 802.11b foi a especificação de uma nova camada física que suporta taxas de transmissão mais elevadas que o protocolo IEEE 802.11-1997. O IEEE 802.11b suporta taxas de transmissão de 5.5 Mbps e 11 Mbps, recorrendo à banda ISM. O valor de 11 Mbps é atingido em condições ideais. Em condições não ideais, podem se utilizadas as taxas de transmissão de 5.5 Mbps, 2 Mbps e 1 Mbps.

O IEEE 802.11b é também conhecido como Wi-Fi (*Wireless Fidelity*). De modo a certificar a interoperabilidade entre os vários produtos 802.11b, várias companhias formaram o WECA (*Wireless Ethernet Compatibility Alliance*).

3.1.5 Características do Protocolo IEEE 802.11a

O protocolo IEEE 802.11a foi aprovado em Dezembro de 1999. Este protocolo define uma camada física que permite uma taxa de transmissão máxima de 54 Mbps, a operar na recentemente criada banda de frequência UNII (*Unlicensed National Information Infrastructure*). O IEEE 802.11a suporta taxas de transmissão de 6, 9, 12, 18, 24, 36, 48, e 54 Mbps. O suporte dos valores de 6, 12 e 24 Mbps é obrigatório.

Quando comparada com a banda de frequência dos 2.4 GHz a banda de frequência de operação do protocolo IEEE 802.11a, 5 GHz, não é tão afectada pelo problema da interferência dada a pequena quantidade de tecnologias a operarem nesta frequência.

A tecnologia adoptada para transmissão foi a OFDM (O*rthogonal Frequency Division Multiplexing*). Esta é uma técnica recente que utiliza um esquema de codificação que oferece benefícios se comparada com as técnicas de espalhamento de espectro anteriormente referidas, nomeadamente no que se refere às elevadas taxas de transmissão alcançadas e maior imunidade a interferências associadas à reflexão do sinal rádio. O IEEE 802.11a proporciona características adicionais à camada física, tais como a possibilidade de correcção de erros.

Para finalizar esta secção a Tabela 3-1 apresenta um resumo comparativo de diversos protocolos da família IEEE 802.11.

Características	IEEE 802.11	IEEE 802.11b	IEEE 802.11a	IEEE 802.11g	IEEE 802.11h
Taxa Transmissão	1-2 Mbps	1-11 Mbps	Até 54 Mbps	54 Mbps	54 Mbps
Normalização	IEEE 802.11-1997	IEEE 802.11b-1999	IEEE 802.11a-1999	IEEE 802.11g-2003	IEEE 802.11h-2003
Banda de Frequência	2.4 Ghz	2.4 Ghz	5 Ghz	2.4 Ghz	5 Ghz

Tabela 3-1 – Protocolos da família IEEE 802.11.

3.2 Camada de Controle de Acesso ao Meio

Para além da função básica de coordenação de acesso ao meio a camada MAC inclui um conjunto de novas funcionalidades e mecanismos: fragmentação e reconstrução, sincronização temporal e gestão de consumo de potência. Na secção 3.2.1 descreve-se o mecanismo de acesso ao meio definido na norma IEEE 802.11. Na secção 3.2.2 é abordado o problema do terminal escondido. O mecanismo de gestão do consumo de potência é referido na secção 3.2.3.

3.2.1 Mecanismos de Acesso ao Meio

O controle de acesso ao meio, tal como o nome indica, fornece um método consistente e justo de acesso dos utilizadores ao meio partilhado. As redes IEEE 802.3 introduziram o conceito de detecção de colisões através do protocolo CSMA/CD (C*arrier Sense Multiple Access with Collision Detection*), suportado por dois mecanismos essenciais: detecção de actividade (*Carrier Sense*) e detecção de colisão (*Collision Detection*). No entanto o CSMA/CD não é

adequado para operar como mecanismo de controle de acesso no meio wireless, devido à dificuldade em detectar colisões. Para ultrapassar esta limitação a norma IEEE 802.11 define uma função de coordenação de acesso ao meio DCF (*Distributed Coordination Function*) que determina quando uma estação pode transmitir. Esta coordenação de acesso ao meio baseia-se no protocolo CSMA/CA (*Sense Multiple Access with Collision Avoidance*), que além de implementar o mecanismo de detecção de actividade CS adoptou um mecanismo de esquiva de colisão CA (*Collision Avoidance*).

De acordo com o principio de funcionamento do CSMA/CA uma estação que pretende transmitir tem de verificar primeiro a actividade do canal de transmissão. Se o canal estiver disponível durante um determinado intervalo de tempo chamado DIFS (*Distributed Inter-Frame Spacing*), então a estação está livre para transmitir. Se, pelo contrário, o canal estiver ocupado a estação aguarda um intervalo de tempo superior a DIFS e atrasa a transmissão de um intervalo de tempo aleatório, até fazer nova tentativa de transmissão. Quando a estação encontra o canal livre então transmite. A estação receptora verifica a trama recebida e envia uma trama de confirmação ACK (*Acknowledgment*) após um intervalo de tempo designado por SIFS (*Short Inter-Frame Spacing*). A recepção do ACK por parte da estação emissora indica que não existiram colisões. Se a estação emissora não receber um ACK irá retransmitir o pacote até receber um ACK, ou até um determinado número de reenvios.

Um dos problemas associados a este mecanismo básico de acesso ao meio é que este assume que todas as estações emissoras conseguem escutar todas as outras estações. No entanto a existência de obstáculos físicos ou o simples facto de determinada estação não estar na área de cobertura do sinal RF da estação emissora impede que todas as estações se escutem umas às outras. Este problema é conhecido como terminal escondido ou *Hidden Node*.

3.2.2 Problema do Terminal Escondido

A Figura 3-2 ilustra o problema do terminal escondido bem como a forma de o resolver. A estação A e a estação B estão na mesma área de cobertura do sinal, o mesmo acontecendo com as estações B e C. Consequentemente se a estação A pretende transmitir para a estação B, qualquer transmissão da estação C que ocorra no mesmo instante de tempo vai corromper as tramas que chegam à estação B. A estação C não está ciente da existência de que outras estações também estão a transmitir para a estação B.

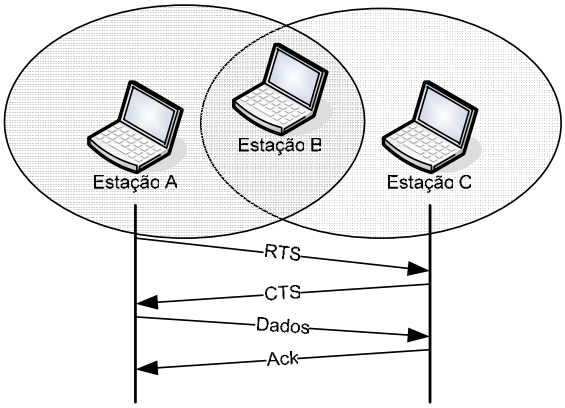


Figura 3-2 – Problemática do terminal escondido.

Para minimizar este problema, são trocadas mais duas mensagens sobre o mecanismo de acesso básico ao meio referido anteriormente. Isto exige que a estação transmissora, na Figura 3-2 a estação A, em primeiro lugar envie uma mensagem de pedido de transmissão RTS (*request to send*) para a estação B que inclui a identificação do transmissor e do destinatário e a duração da transmissão. Se a estação B receber correctamente o pedido, responde com uma mensagem de pronto para enviar CTS (*clear to send*), que também inclui a duração da transmissão. Esta mensagem é escutada por ambas as estações A e C. Deste modo apenas a estação A irá transmitir neste instante, uma vez que recebeu a mensagem CTS da estação B como resposta ao seu pedido.

As estações que recebem um RTS e/ou um CTS actualizam um temporizador, designado por NAV (*Network Allocation Vector*), que indica o tempo remanescente de ocupação do canal desde o fim da trama RTS (ou CTS) até ao fim da trama de confirmação ACK.

Com esta informação as estações escondidas da estação de origem ou da estação de destino, que não recebem o RTS mas recebem o CTS ou vice-versa, só voltam a escutar o canal para competir pelos seus recursos após terminar o período de tempo indicado pelo NAV.

3.2.3 Gestão de consumo de potência

A camada de controle de acesso ao meio fornece funcionalidades de gestão de consumo de potência apenas para redes não-cabladas que operam em modo infra-estrutura. A topologia de uma rede não-cablada é apresentada na secção 3.3.2.

As estações wireless podem operar em um de dois modos de gestão de consumo de potência: o modo activo, designado normalmente por CAM (*Continuos Aware Mode*), e o modo de conservação de potência com interrogação PSP (*Power Save Polling*). As estações podem estar no estado "acordado" ou no estado "adormecido". As estações no estado adormecido têm apenas um conjunto mínimo de circuitos ligados e não podem transmitir nem receber, enquanto as estações em modo acordado têm todos os circuitos activos.

Modo activo

No modo activo as estações wireless permanecem no estado "acordado" e mantêm uma comunicação regular com o ponto de acesso, o que lhes permite receber tramas a qualquer instante. As tramas com destino a estações que operam neste modo não são armazenada pelo ponto de acesso.

Modo de conservação de potência com interrogação

Quando estão a operar em modo PSP, as estações clientes informam o ponto de acesso que vão entrar no estado "adormecido". Após receber essa informação o ponto de acesso regista as estações clientes que estão nesse estado. Todas as tramas com destino a uma estação no estado "adormecido" são armazenadas no ponto de acesso. O ponto de acesso difunde informação em cada *Beacon*. As estações clientes sabem exactamente quando recebem os *Beacons*. Assim, em intervalos de tempo regulares as estações em estado "adormecido" passam ao estado "acordado" para receberem informação, que inclui um mapa de indicação de tráfego TIM (*Traffic Indication Map*). Se uma estação verifica que está listada no TIM, então fica no estado "acordado" e informa o ponto de acesso que está preparada para receber as tramas armazenadas. Após a recepção de todas as tramas a estação volta ao estado "adormecido". O processo repete-se continuamente. Este modo de funcionamento cria algum *overhead* nas comunicações que não ocorre se este modo de gestão de consumo de potência não for utilizado.

3.3 Arquitectura

Os diversos componentes definidos na arquitectura IEEE 802.11 são apresentados na secção 3.3.1. A arquitectura IEEE 802.11 define duas topologias de rede distintas, rede em modo *Ad-Hoc* e em modo infra-estrutura. A distinção destes modos de operação é feita na subsecção 3.3.2. Na secção 3.3.3 são definidos os serviços lógicos e a sua relação com os estados de uma estação.

3.3.1 Componentes

As redes não-cabladas IEEE 802.11 são redes baseadas numa arquitectura celular onde o sistema é subdividido em células. A célula, também designada por conjunto básico de serviço BSS (*Basic Service Set*), é o componente base de uma rede IEEE 802.11. A área ocupada pelas estações de uma BSS é designada por área básica de serviço BSA (*Basic Service Area*). Embora uma rede não-cablada possa ser constituída por uma única célula é comum encontrar redes formadas por várias células, em que pontos de acesso estão ligados através de uma rede principal (*Backbone*) designado por sistema de distribuição DS (*Distribution System*). O sistema de distribuição é tipicamente uma rede *Ethernet*, mas pode em alguns casos ser um sistema wireless.

Um outro conceito que deriva do conceito de BSS é o de BSS independente IBSS (*Independent Basic Service Set*), ou célula única. O IBSS permite a definição do tipo de rede 802.11 mais básico, que pode ser constituída apenas por duas estações. Esta topologia de rede conhecida como rede *ad-hoc* é abordada na secção 3.3.2.

O conjunto dos componentes sistema de distribuição e BSS permite a implementação de uma rede com área de cobertura e complexidade variável. A norma IEEE 802.11 refere-se a este tipo de rede como conjunto estendido de serviço ESS (*Extended Service Set*).

O conceito chave inerente a esta é o de que uma rede do tipo ESS é vista pela camada LLC como uma rede do tipo IBSS, o que permite a uma estação comunicar ou mover-se de um BSS para outro de forma transparente para a camada LLC. Esta topologia de rede é conhecida também como rede em modo infra-estrutura e é analisada na secção 3.3.2.

A norma IEEE 802.11 define um último componente lógico na sua arquitectura, designado por Portal. Este componente é o ponto lógico no qual os MSDUs (*Medium Access Control Service Data Unit*) provenientes de uma rede não IEEE 802.11 entram no sistema de distribuição. Basicamente um Portal fornece uma integração lógica entre a arquitectura IEEE 802.11 e as actuais redes cabladas. A funcionalidade de um Portal é normalmente integrada numa entidade física, que apresenta simultaneamente as funcionalidades de ponto de acesso e Portal.

Os componentes abordados nos parágrafos anteriores estão representados na Figura 3-3.

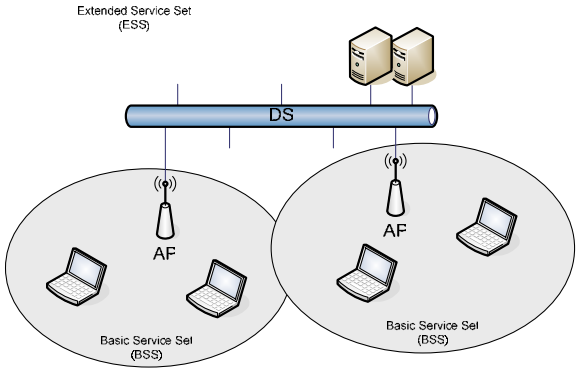


Figura 3-3 – Componentes da arquitectura IEEE 802.11.

3.3.2 Topologia de Rede

Na secção 3.3.1 foram definidos os conceitos de BSS independente e conjunto estendido de serviço. Estes conceitos correspondem a dois tipos de redes, *redes ad-hoc* e redes em modo infra-estrutura, descritas nas secções seguintes.

Ad-hoc

Uma rede *ad-hoc* é uma rede que é criada de forma espontânea entre estações. Normalmente qualquer estação é capaz de comunicar com as restantes estações da rede. A Figura 3-4 representa uma topologia de rede em modo *ad-hoc*.

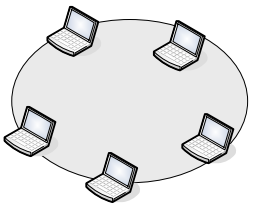


Figura 3-4 - Rede não-cablada em modo Ad-hoc.

Em redes *ad-hoc*, a troca de dados entre estações é possível no estado inicial, ao contrário das redes em modo infra-estrutura onde a troca de dados só é possível no estado três. O conceito de estado de uma estação assim como a sua análise detalhada é feita na secção 3.3.3.

Um cenário em que este tipo de rede tem aplicação é, por exemplo, uma reunião de trabalho onde os intervenientes possuam computadores portáteis, que comuniquem entre si para troca de informação.

Infra-estrutura

O segundo tipo de topologia definido na norma 802.11 é o modo infra-estrutura. Uma topologia deste tipo fornece comunicação entre os componentes de redes não-cabladas e os recursos de uma rede convencional.

A troca de dados da rede não-cablada para o meio cablado é efectuado através de um ponto de acesso. A área de cobertura deste tipo de topologia é definida pelos pontos de acesso e pelas estações associadas a estes. Uma topologia em modo infra-estrutura é bastante similar às actuais redes celulares. A Figura 3-5 ilustra uma rede não-cablada em modo infra-estrutura.

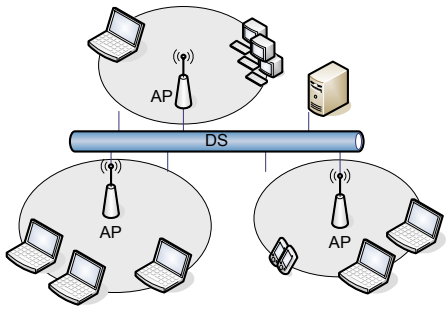


Figura 3-5 - Rede não-cablada em modo infra-estrutura.

Do ponto de vista de segurança, uma rede *ad-hoc* apresenta características e desafios completamente distintos os quais devem ser tratados separadamente. Uma vez que a grande maioria das redes não-cabladas são utilizadas em modo infra-estrutura e oferecendo este uma plataforma de segurança mais completa, todo o trabalho apresentado nesta dissertação referir-se-á a este modo de operação.

3.3.3 Serviços Lógicos

A arquitectura da norma IEEE 802.11 possibilita que o sistema de distribuição seja constituído por diversas tecnologias incluindo as actuais redes cabladas IEEE 802. A norma IEEE 802.11 não especifica os detalhes de implementação do sistemas de distribuição. Em vez de especificar o sistema de distribuição, a norma especifica serviços. São definidas duas categorias de serviços lógicos: Serviços de estação SS (*Station Services*), e serviços do sistema de distribuição DSS (*Distribution System Service*). Ambas as categorias são utilizadas pela camada de controle de acesso ao meio.

Os serviços de estação estão presentes em todas as estações IEEE 802.11, incluindo pontos de acesso que possuam funcionalidades de estação. Qualquer estação conforme a norma deve prestar os seguintes serviços:

- Autenticação
- Desautenticação
- Privacidade
- Entrega de MSDUs

Os serviços proporcionados pelo sistema de distribuição são os seguintes:

- Associação
- Desassociação
- Distribuição
- Integração
- Reassociação

Antes da descrição dos serviços lógicos definidos na norma, apresentam-se na Figura 3-6 os componentes e as duas categorias de serviços lógicos que caracterizam por completo a arquitectura IEEE 802.11.

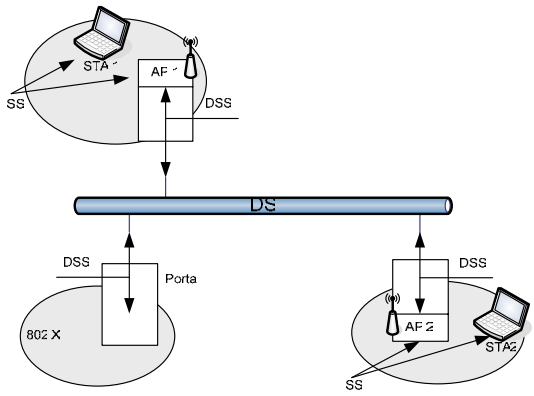


Figura 3-6 – Arquitectura IEEE 802.11.

A descrição da relação entre os diversos serviços lógicos, apresentada nas subsecções seguintes, é feita por uma ordem que permite uma compreensão mais efectiva do funcionamento das redes 802.11. As duas primeiras subsecções abordam serviços de entrega ou distribuição de mensagens através do sistema de distribuição. Após estes são abordados três serviços de suporte ao serviço de distribuição. Finalmente são analisados os serviços de controlo de acesso e confidencialidade. Os exemplos apresentados assumem uma topologia de rede em modo infra-estrutura.

Distribuição

Este é o serviço principal utilizado por uma estação. Conceptualmente este serviço é invocado por todas as mensagens de dados, de e para uma estação a funcionar em modo ESS, quando um pacote de dados é enviado através do sistema de distribuição.

De modo a clarificar a funcionalidade do serviço de distribuição atente-se no seguinte exemplo. Recorrendo à rede da Figura 3-6, considere-se uma mensagem de dados enviada da estação STA 1 para a estação STA 2. A mensagem é enviada da estação STA 1 e recebida pelo ponto de acesso AP 1 que entrega a mensagem ao serviço de distribuição. A função do serviço de distribuição é entregar a mensagem através do sistema de distribuição, DS, de forma a que esta alcance o seu destino. Neste exemplo a mensagem é distribuída para o ponto de acesso, AP 2, este acede ao meio wireless para enviar a mensagem para a estação, STA 2.

A forma como a mensagem é distribuída através do sistema de distribuição não é especificada na norma IEEE 802.11. A norma apenas define a informação que deve ser fornecida ao sistema de distribuição, para que este determine o AP de saída correspondente ao destinatário

da mensagem. A informação que é necessário fornecer ao sistema de distribuição é dada pelos serviços de Associação, Desassociação e Reassociação abordados nas secções seguintes.

Integração

Se o serviço de distribuição determinar que o destinatário de uma mensagem é um componente de uma rede convencional, então o ponto de saída do sistema de distribuição será um Portal e não um ponto de acesso.

As mensagens distribuídas para um Portal obrigam o sistema de distribuição a invocar o serviço de integração (conceptulamente após o serviço de distribuição). Uma mensagem proveniente de um Portal recebida pelo sistema de distribuição com destino a uma estação, vai invocar o serviço de integração antes da mensagem ser distribuída.

Um exemplo em que o serviço de integração é invocado é o envio de uma mensagem da estação STA 1 para uma máquina M1, localizada numa rede cablada. A mensagem é enviada da estação STA 1 e recebida pelo ponto de acesso AP 1. De seguida o serviço de distribuição é invocado e determina que o ponto de saída da mensagem é o Portal. Após esta fase o serviço de integração é invocado e toda a informação necessária para entregar a mensagem do meio do serviço de distribuição DSM (*Distribution Service Medium*) para o meio cablado é recolhida.

Os detalhes do serviço de integração não são especificados na norma IEEE 802.11, dada a dependência que este tem das possíveis implementações do sistema de distribuição.

Associação

Para encaminhar uma mensagem através do sistema de distribuição este necessita de conhecer o ponto de acesso a que deve aceder para alcançar a estação destinatária da mensagem. Esta informação é fornecida ao sistema de distribuição pelo serviço de associação.

Antes do envio de uma mensagem por parte de uma estação através de um ponto de acesso devem ser cumpridos um conjunto de pré-requisitos, designadamente a autenticação da estação no ponto de acesso através do serviço de autenticação e associação da estação no ponto de acesso utilizando o serviço de associação. A análise da relação dos vários serviços lógicos é efectuada no final desta secção. O serviço de associação tem como funcionalidade principal fornecer a informação necessária ao sistema de distribuição que a utiliza sempre que o serviço de distribuição de mensagens é invocado. A informação fornecida pelo serviço de associação consiste na indicação do conjunto de estações associadas a cada ponto de acesso.

Conclui-se que uma estação não pode estar em determinado instante associada a mais do que um ponto de acesso. Pelo contrário um ponto de acesso pode estar em determinado momento associado a várias estações. Este facto garante que o sistema de distribuição possui a informação exacta quando questionado pelo serviço de distribuição relativamente ao ponto de acesso que está a servir uma determinada estação. O serviço de associação é sempre invocado por uma estação que pretende associar-se a um ponto de acesso no qual está autenticado. A associação deve ser entendida como um pedido e não como uma notificação. Assim um pedido de associação pode ser rejeitado pelo ponto de acesso.

Reassociação

O serviço de associação é suficiente para a troca de mensagens entre estações estáticas, sem movimento, ou cuja mobilidade está dentro do mesmo espaço físico, isto é na área de cobertura da célula, BSA.

Em cenários ou situações de mobilidade entre BSSs, são necessárias funcionalidades adicionais. Essas funcionalidades são fornecidas pelo serviço de reassociação. O serviço de reassociação é invocado sempre é necessário "mover" a associação de uma estação de um ponto de acesso para outro ponto de acesso. Assim a actualização da informação dos atributos de associação de uma estação é garantida. Isto permite manter o sistema de distribuição informado sobre o mapeamento entre os pontos de acesso e as estações que se movem de um BSS para outro BSS pertencente ao mesmo ESS. Uma outra funcionalidade fornecida pelo serviço de reassociação é a alteração por parte de uma estação dos atributos de uma associação já estabelecida.

A reassociação é sempre iniciada pela estação. À semelhança do serviço de associação o serviço de reassociação é um pedido e não uma notificação pelo que este pode ser rejeitado por um ponto de acesso.

Desassociação

Um outro serviço lógico de apoio ao serviço de distribuição é o serviço de desassociação. Este serviço é invocado sempre que se pretende terminar uma associação existente.

Num cenário ESS este serviço informa o sistema de distribuição que deve eliminar toda a informação referente a uma determinada associação. Previne-se deste modo a tentativa de envio de uma mensagem através do serviço de distribuição para uma estação desassociada.

O serviço de desassociação pode ser invocado quer por um ponto de acesso quer por uma estação. Ao contrário dos serviços referidos anteriormente uma desassociação é uma notificação não um pedido, pelo que esta não pode ser rejeitada por qualquer das partes.

Autenticação

Em redes convencionais, o controlo de acesso à rede pode ser implementado recorrendo a mecanismos de segurança físicos, que não permitam o acesso ao meio de transmissão. Em redes não-cabladas este conceito de segurança é impraticável uma vez que os limites do meio de comunicação não são definidos com precisão.

O controlo de acesso a redes não-cabladas é fornecido pelo serviço de autenticação. Este serviço é utilizado por todas as estações para estabelecerem a sua identidade perante as restantes estações (móveis ou pontos de acesso) com as quais vão manter comunicação, independentemente da rede operar em modo *ad-hoc* ou em modo infra-estrutura.

A norma IEEE 802.11 define dois mecanismos de autenticação, não exigindo a utilização de um em particular. Os mecanismos de autenticação definidos são:

- Autenticação aberta (*Open System Authentication*)
- Autenticação de chave partilhada (*Shared Key Authentication*)

Na secção 4.1 é feita uma análise detalhada destes mecanismos de autenticação. O serviço de autenticação é simplesmente utilizado para "activar" a ligação wireless. A utilização deste serviço de autenticação é independente de qualquer processo de autenticação implementado num nível protocolar superior.

O serviço de autenticação definido não fornece autenticação ponto-a-ponto nem autenticação de utilizadores, apenas fornece autenticação dos dispositivos.

Desautenticação

O serviço de desautenticação é invocado sempre que se pretende cessar uma autenticação existente. Num cenário ESS o facto de o serviço de desautenticação ser invocado provoca automaticamente a desassociação da estação respectiva.

Tal como o serviço de desassociação também o serviço de desautenticação é uma notificação e não um pedido pelo que não pode ser rejeitado por nenhum dos componentes, estação ou

ponto de acesso. Sempre que um ponto de acesso envia uma notificação de desautenticação a uma estação associada, também o serviço de desassociação deve ser invocado.

Privacidade

Para fornecer um nível de privacidade equivalente às redes cabladas, a norma IEEE 802.11, fornece funcionalidades de cifra do conteúdo das mensagens. Esta funcionalidade é fornecida pelo serviço lógico de privacidade.

O mecanismo de confidencialidade especificado na norma é o WEP (*Wired Equivalent Protocol*). Este mecanismo é descrito em pormenor na secção 4.2.

De notar que o serviço de privacidade apenas pode ser invocado para pacotes de dados e para alguns pacotes de gestão. Durante a fase em que são estabelecidos os serviços de autenticação e privacidade todas as estações devem funcionar sem cifra.

Sempre que uma estação configurada com o serviço de privacidade receber um pacote de dados em texto plano ou com dados cifrados com uma chave errada, então esses pacotes devem ser descartados, sem necessidade de indicação para camada LLC.

Relação entre serviços lógicos

A norma IEEE 802.11 define que cada estação deve manter duas variáveis de estado que são dependentes dos serviços de autenticação, desautenticação e dos serviços de associação, reassociação e desassociação.

As duas variáveis de estado são: o estado da autenticação e o estado da associação das estações. Estas variáveis são utilizadas numa máquina de estados simples, por forma a determinar em que ordem devem ser invocadas e em que momento podem ser utilizados por uma estação os dados entregues por determinado serviço lógico.

A máquina de estados possui três estados possíveis para cada uma das estações. Os três estados são:

- Estado 1: Estado inicial, estação desautenticada e desassociada
- Estado 2: Estado intermédio, estação autenticada e desassociada
- Estado 3: Estação final, estação autenticada e associada.

A relação entre variáveis de estado e serviços lógicos é representada na Figura 3-7.

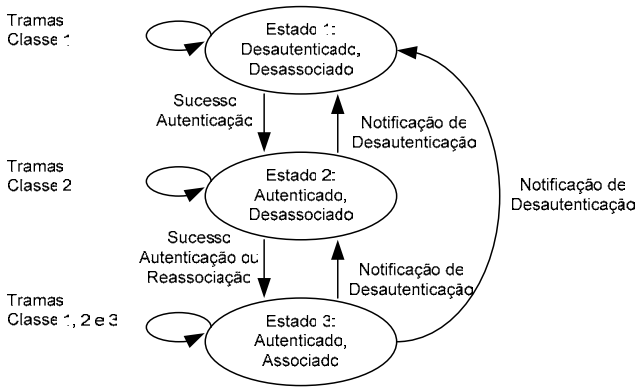


Figura 3-7 – Relação entre as variáveis de estado e os serviços lógicos.

O estado no qual se encontra uma estação, determina o tipo de tramas que podem ser trocadas entre estações.

No estado inicial é apenas permitida a troca de tramas cuja funcionalidade implementa o serviço de autenticação. As tramas que podem ser trocadas no estado 1 são designadas por tramas de classe 1. No estado 2 ou estado intermédio, são permitidas tramas adicionais que implementam os serviços de associação, desassociação e reassociação. Estas são conhecidas como tramas de classe 2. No terceiro e último estado, são permitidos todos os tipos de tramas, incluindo aquelas que permitem a utilização do serviço de dados.

3.4 Detalhes do Protocolo IEEE 802.11

Tal como referido inicialmente não é objectivo desta dissertação apresentar todos os detalhes de funcionamento do protocolo IEEE 802.11. Os conceitos básicos foram apresentados nas secções anteriores. Para detalhes mais específicos relacionados com o mecanismos de controlo de acesso ao meio definido na norma IEEE 802.11 aconselha-se a leitura de bibliografia especifica como por exemplo [1] [2]. É no entanto imperativo para a compreensão dos aspectos de segurança abordados nesta dissertação, fazer referência a alguns dos detalhes definidos na norma IEEE 802.11, nomeadamente o formato da trama MAC, assim como alguns campos e subcampos presentes nesta.

3.4.1 Formato Geral de uma Trama 802.11

Todas as transmissões sobre o meio wireless têm um formato semelhante, que se representa na Figura 3-8. À semelhança das redes *Ethernet* também os pacotes IEEE 802.11 são iniciados com um campo designado ***Preamble*** que permite ao receptor sincronizar-se com a taxa de transmissão do emissor. O ***PLCP Header***, (*Physical Layer Convergence Protocol*) é o cabeçalho da camada física e não têm qualquer intervenção nos aspectos de segurança. A seguir ao ***PLCP Header*** vêm o cabeçalho ***MAC*** seguido pelo campo ***Frame Body*** que contém os dados a transmitir. O último campo é o ***FCS*** (*Frame Check Sequence*) que possui o valor do CRC (*Cyclic Redundancy Check*) usado para detecção de erros.

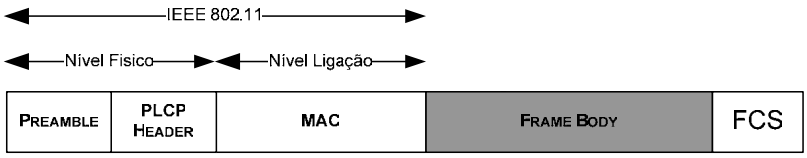


Figura 3-8 – Estrutura de um pacote IEEE 802.11.

Na secção 3.4.2 é efectuada uma análise ao formato da trama de controlo de acesso ao meio, ***MAC Header***, dada a importância da informação relativa ao controlo e gestão da rede que este pacote possui, nomeadamente ao nível de controlo de acesso, endereçamento, integridade e confidencialidade dos pacotes.

3.4.2 Formato da Trama MAC

O formato da trama MAC é especificado na norma IEEE 802.11. A Figura 3-9 descreve o seu formato geral.

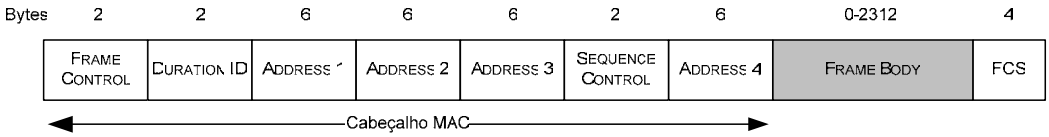


Figura 3-9 – Formato genérico da trama MAC.

Cada trama MAC é constituída pelos seguintes componentes básicos:

- Cabeçalho MAC, que integra campos de controlo, duração e vários campos com informação relativa a endereçamento.

- Um campo de tamanho variável designado por ***Frame Body*** o qual contém informação específica do tipo de pacote e os dados a transmitir.
- Um campo que verifica a integridade da sequência de dados designado por ***FCS*** (*Frame Check Sequence*) o qual contém o CRC (*Cyclic Redundancy Code*).

Os campos de endereço 2, endereço 3, controlo de sequência e ***Frame Body*** surgem apenas em alguns tipos de pacotes.

Nas secções seguintes são descritos com detalhe os campos que compõem a trama MAC.

3.4.3 Campo Frame Control

Este campo é obrigatoriamente enviado em todas as tramas MAC, uma vez que o seu conteúdo compreende toda a informação respeitante ao pacote.

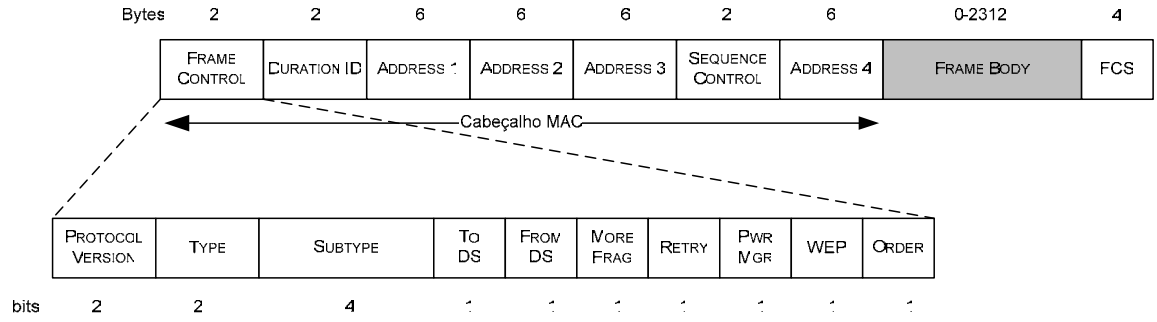


Figura 3-10 – Campo Frame Control.

O campo ***Frame Control*** é formado pelos seguintes subcampos (Figura 3-10): ***Protocol Version***, ***Type***, ***To DS***, ***From DS***, ***More Fragments***, ***Retry***, ***Power Management***, ***More Data***, ***WEP*** e ***Order***.

O campo ***Protocol Version*** é um campo composto por dois bits que identifica a versão do protocolo implementado pela estação de origem. Os dispositivos existentes no mercado compatíveis com a versão mais recente do protocolo devem utilizar o valor zero neste campo. O campo ***Type*** tem comprimento de dois bits enquanto o campo ***Subtype*** é composto por quatro bits. Em conjunto estes dois campos indicam a função e a interpretação da trama. O campo ***Type*** identifica três tipos de tramas: controle, dados e gestão. Cada um destes tipos de trama possui diversos subtipos definidos pelo campo ***Subtype***. Exemplos de subtipos de

tramas de gestão são a ***Association Request*** e ***Association Response***. Exemplos de subtipos de tramas de controle são ***RTS***, ***CTS*** e ***ACK***.

A Tabela 3-2 ilustra diversas combinações válidas para os valores dos campos ***Type*** e ***Subtype***.

Type	Descrição	SubType	Descrição
00	Gestão	0000	Association Request
00	Gestão	0001	Association Response
00	Gestão	0010	Reassociation Request
00	Gestão	0011	Reassociation Response
00	Gestão	0100	Probe Request
00	Gestão	0101	Probe Response
00	Gestão	1000	Beacon
00	Gestão	1010	Disassociation
00	Gestão	1011	Authentication
00	Gestão	1100	Deauthentication
01	Controlo	1011	Request to Send (RTS)
01	Controlo	1100	Clear to Send (CTS)
10	Dados	0000	Data

Tabela 3-2 Combinações válidas para os campos Type e Subtype.

Os bits ***To DS*** e ***From DS*** estão relacionados com o encaminhamento das tramas. Num cenário ESS, uma trama pode ser encaminhada de uma estação, através de um ponto de acesso, para outra estação associada a outro ponto de acesso. Estes bits descrevem o encaminhamento das tramas através do sistema de distribuição e informam o dispositivo destinatário sobre como deve interpretar os campos de endereçamento. A Tabela 3-3 apresenta a interpretação destes dois bits.

To DS	From DS	Descrição
0	**0**	Pacote de dados é encaminhado directamente de uma estação para outra estação num cenário IBSS. Pacotes do tipo controlo e gestão.
1	**0**	Pacote de dados com destino ao sistema de distribuição
0	**1**	Pacote de dados com origem no sistema de distribuição
1	**1**	Sistema de distribuição wireless (WDS), pacotes transmitidos de um ponto de acesso para outro ponto de acesso

Tabela 3-3 – Combinação dos bits To/ From DS nos pacotes de dados.

O bit ***More Frag*** é colocado a um em todas as tramas de gestão e de dados fragmentados, e indica que não é o último fragmento transmitido. Este bit tem o valor zero sempre que a trama transmitida é o último fragmento, ou a trama nunca foi fragmentada. A fragmentação é

implementada dinamicamente pelo protocolo IEEE 802.11 com o objectivo de minimizar o impacto das retransmissões.

Sendo a retransmissão de pacotes uma ocorrência comum é importante assinalar quais as tramas que são retransmitidas. Para tal é utilizado o bit ***Retry*** que é colocado a um em todas as tramas retransmitidas. A principal utilidade deste bit é a possibilidade de eliminação por parte das estações receptoras de pacotes repetidos. O bit ***Power Management***, tal como o seu nome indica codifica o modo de gestão de consumo de potência de uma estação. Se o seu valor for um significa que a estação está em modo de conservação de potência com interrogação. O valor zero indica que a estação está em modo activo. O bit ***Order*** é colocado a um para indicar que um MSDU está a ser transmitido usando a classe de serviço *Strictly Order* (por exemplo uma transmissão de voz). Este bit dá a indicação que nenhum outro MSDU pode ser enviado para uma estação até que o MSDU marcado com o bit a um seja completamente transmitido. Em todas as tramas que não usem a classe de serviço *Strictly Order* o seu valor é zero. O bit ***WEP*** possui o valor um se o ***Frame Body*** contêm informação que foi processada pelo algoritmo de privacidade WEP. Este bit apenas é colocado a um em tramas de dados e em tramas de autenticação. Em todos os outros pacotes o seu valor é zero. Quando este bit toma o valor um, o campo ***Frame Body*** toma a forma definida em 4.2.2.

3.4.4 Campo Duration ID

Um outro campo com funcionalidade de controlo de transmissão de pacotes é o campo ***Duration ID***. Este campo possui um comprimento de 16 bits tal como é representado na Figura 3-11. Para auxiliar no sincronismo e acesso ao meio rádio, as tramas a transmitir contêm o valor de tempo exacto de ocupação do canal rádio pela transmissão da trama.

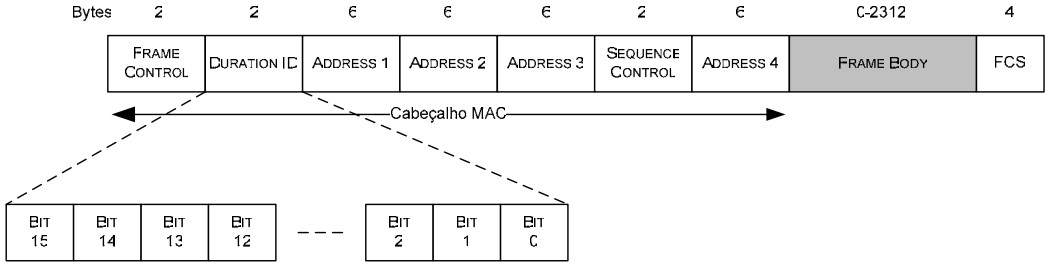


Figura 3-11 – Campo Duration ID.

3.4.5 Campo Sequence Control

O campo ***Sequence Control*** tem comprimento de 16 bits e é composto por dois campos. O campo ***Fragment Number*** com tamanho de 4 bits identifica o fragmento de uma trama. O seu valor incrementa de um por cada fragmento transmitido, mas a retransmissão de um fragmento não provoca alteração no seu valor. O outro campo é designado por ***Sequence Number***. Este campo é composto por 12 bits que indicam o número de sequência de uma trama. A cada trama transmitida é atribuído um número de sequência. Este valor incrementa de um por cada trama transmitida. A retransmissão ou fragmentação de um pacote não altera o valor deste campo. Estes dois campos são utilizados de forma combinada na reassemblagem de tramas que tenham sido fragmentadas.

O formato do campo ***Sequence Control*** é ilustrado na Figura 3-12.

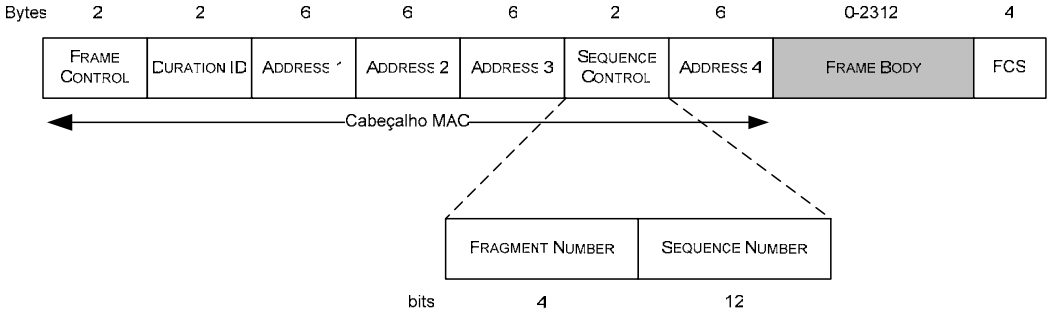


Figura 3-12 – Campo Sequence Control.

3.4.6 Campo Frame Body

O ***Frame Body*** é um campo de tamanho variável que contém informação especifica relacionada com os campos ***Type*** e ***Subtype***. Este campo não é utilizado nas tramas de controle. Nas tramas de dados transporta o MSDU enviado à MAC, ou um seu fragmento. Nas tramas de gestão é formado por um conjunto de campos de comprimento fixo e por um conjunto de comprimento variável. Os campos de comprimento variável tomam a forma de uma lista ligada e são designados por elementos (*Elements*) [3]. A secção 3.4.9 descreve com mais detalhe o conceito de elemento.

O tamanho mínimo deste campo é zero bytes. O tamanho máximo é obtido pelo tamanho máximo do conjunto de campos, (MSDU+ICV+IV). Para redes IEEE 802.11 o tamanho

máximo de um MSDU é 2304 bytes. Os campos ICV (*Integrity Check Value*) e IV (*Initialization Vector*) são campos definidos no protocolo WEP. Uma análise a estes campos é efectuada na secção 4.2.2

3.4.7 Campo FCS

O campo ***FCS*** tem comprimento de 32 bits que contém o resultado do algoritmo de detecção de erros CRC – 32. O seu valor é calculado sobre o conteúdo do cabeçalho MAC e do campo ***Frame Body***. Estes campos são definidos na norma como ***Calculation Field***.

O FCS é calculado recorrendo ao polinómio gerador de grau 32 seguinte:

$$G(x)=x^{32}+x^{26}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1 \quad \text{Equação 3-1}$$

O FCS é o complemento para um (modulo 2) da soma de a) e b), onde :

a) Resto da divisão de $x^{k}.(x^{31}+x^{30}+x^{29}+...+x^{2}+x+1)$ por $G(x)$ onde k é o número de bits dos campos ***Calculation Field***.

b) Resto da divisão do resultado da multiplicação dos conteúdos dos campos ***Calculation Field*** por x^{32} com $G(x)$.

Os bits que chegam dos campos ***Calculation Field*** e do campo ***FCS***, quando divididos por $G(x)$, resultam na ausência de erros de transmissão se o resto dessa divisão for único e diferente de zero. O resultado do resto de valor único deve ser o polinómio seguinte:

$$x^{31}+x^{30}+x^{26}+x^{25}+x^{24}+x^{18}+x^{15}+x^{14}+x^{12}+x^{11}+x^{10}+x^{8}+x^{6}+x^{5}+x^{4}+x^{3}+x+1 \quad \text{Equação 3-2}$$

3.4.8 Campos de Endereçamento

Existem quatro campos de endereço, ***Address***, no cabeçalho da trama MAC. Estes campos são utilizados para indicar o endereço da estação de origem SA (*Source Address*), o endereço da estação de destino DA (*Destination Address*), o endereço da estação transmissora TA (*Transmitter Address*), o endereço da estação receptora RA (*Receiver Address*) e a identificação BSS-ID (*Basic Service Set*). Este último campo contém o endereço MAC do ponto de acesso do BSS. De referir ainda que nem todos os pacotes possuem os quatro campos de endereço.

O conteúdo dos campos de endereço de uma trama de dados está directamente relacionado com os valores dos campos ***To DS*** e ***From DS*** definidos na Tabela 3-3. A Tabela 3-4 apresenta os valores possíveis dos campos de endereços de acordo com os valores dos campos ***To DS*** e ***From DS***.

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSS-ID	Não utilizado
0	1	DA	BSS-ID	SA	Não utilizado
1	0	BSS-ID	SA	TA	Não utilizado
1	1	RA	TA	DA	SA

Tabela 3-4 – Conteúdo dos campos Address.

Cada campo ***Address*** tem um tamanho de 48 bits tal como definido em [4]. Na norma IEEE 802.11 é dada aos endereços a seguinte designação:

- Endereços individuais – São endereços associados a uma estação em particular.
- Endereços de grupo – São endereços multi-destino, estão associados a uma ou mais estações. Estes endereços estão divididos em dois grupos distintos: endereços de *multicast* e endereços de *broadcast*. O campo endereço de destino DA com todos os bits a um é interpretado como um endereço de *Broadcast*.

3.4.9 Tramas de Gestão

Como foi referido na secção 3.4.3, existem três categorias de tramas MAC: controlo, gestão e de dados. Nesta secção são abordadas as tramas de gestão dada a sua relação com os protocolos de segurança abordados no decorrer desta dissertação.

A norma IEEE 802.11-1997, refere as seguintes tramas de gestão para utilização em modo infra-estrutura.

- *Beacon* (Notificação)
- Probe Request e Probe Response
- Authenticate Request e Authenticate Response
- Associate Request e Associate Response
- Reassociate Request e Reassociate Response

- Dissassociate (Notificação)
- Deauthenticate (Notificação).

As tramas de gestão identificadas como sendo de notificação são tramas que são transmitidas e não esperam qualquer tipo de resposta por parte da estação receptora.

O conteúdo de uma trama de gestão compreende duas partes: a primeira é composta por um conjunto de campos fixos que variam com o tipo de trama de gestão; a segunda é constituída por um conjunto de elementos. Um elemento (*Element*) é um campo que contém informação que pode, ou não, ser relevante para a estação receptora.

A Figura 3-13 apresenta o formato genérico de uma trama de gestão.

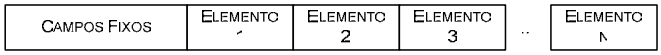


Figura 3-13 – Formato genérico de uma trama de gestão.

O recurso a elementos é uma solução bastante eficiente e flexível com vários benefícios. A utilização de elementos torna a actualização da norma mais fácil de realizar. A introdução de informação que implementa novos métodos de segurança pode ser introduzida recorrendo a novos elementos. A vantagem é a compatibilidade com sistemas anteriores que não compreendam a nova informação, uma vez que estes simplesmente ignoram todos os elementos que não compreendam.

Os fabricantes por vezes têm benefícios em utilizar esta capacidade de extensão dos elementos, pois podem deste modo fornecer algumas características especificas dos seus produtos. Um exemplo é o elemento utilizado por alguns fabricantes na trama de gestão ***Beacon*** que inclui informação relativa ao estado de ocupação de um ponto de acesso. Isto permite a implementação de partilha de carga (*Load Balance*) na qual as estações se distribuem pelos diversos pontos de acesso.

Os elementos possuem uma estrutura similar. O primeiro byte identifica o tipo de elemento e é designado na norma por ***Element ID***. O segundo byte, ***Length***, indica tamanho em bytes ocupados pela informação que se segue no campo ***Information***. A Figura 3-14 apresenta a estrutura geral de um elemento.

Figura 3-14 – Estrutura geral de um Elemento.

Um dos tipos de trama de gestão mais importantes no que se refere aos aspectos de segurança é a trama de gestão de subtipo autenticação. A subsecção seguinte descreve a estructura de uma mensagem deste tipo.

Mensagem de Autenticação

Para uma visão genérica do formato de uma mensagem de autenticação, a Figura 3-15 ilustra os campos que compõem uma trama de gestão de subtipo autenticação. O campo ***Subtype*** possui o valor decimal 11 de acordo com a Tabela 3-2.

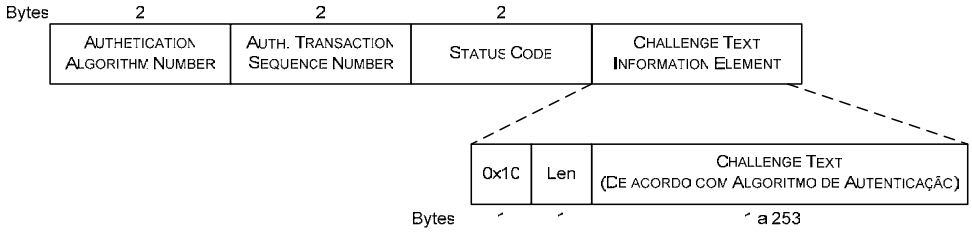


Figura 3-15 – Formato geral de uma mensagem de autenticação.

Os campos de uma trama de autenticação têm o seguinte significado:

- ***Authentication Algorithm Number*** – Identifica o algoritmo de autenticação utilizado.
- ***Authentication Transaction Sequence Number*** – Indica a ordem da trama correspondente na sequência de autenticação.
- ***Status Code*** – Permite verificar do sucesso ou insucesso de um pedido de autenticação. Sempre que um pedido de autenticação ocorra com sucesso este campo deve ter o valor 0.
- ***Challenge Text*** – Apresenta o desafio (texto gerado pelo gerador de números pseudo-aleatórios). Este campo apenas está presente nas tramas com ***Authentication Transaction Sequence Number*** igual a 2 e 3 do algoritmo de autenticação de chave partilhada.

3.5 Cenário Descritivo do Funcionamento IEEE 802.11

Esta secção examina e descreve o modo de funcionamento dos dispositivos da família IEEE 802.11 em modo infra-estrutura sem abordar particularmente os aspectos de segurança. Embora existam mecanismos bastantes similares para IBSS, esta secção foca apenas o cenário genérico de BSS. Para obter um determinado serviço num ambiente de rede não-cabladas, cada estação deve executar algumas funções ou serviços descritos na norma IEEE 802.11.

O cenário explorado nesta secção descreve as diversas etapas de uma estação que pretende o acesso a um servidor Web localizado na Internet.

3.5.1 Sincronização e Detecção

A primeira fase consiste na sincronização e detecção. A estação IEEE 802.11 em primeiro lugar necessita de procurar um ponto de acesso na área de cobertura do sinal RF e em seguida sincronizar-se com o BSS especifico. Como podem existir diversos sistemas IEEE 802.11 na mesma área, e a estação necessita de encontrar o BSS da sua preferência, os pontos de acesso enviam informação periódica que é transmitida para todas as estações na sua área de cobertura.

Esta informação periódica é enviada na trama de gestão ***Beacon*** e contém informação como seja, a identificação do BSS (BSS-ID), identificação do ESS (SSID ou ESSID), parâmetros referentes à camada física e o relógio (*Clock*) nativo do ponto de acesso transmitido num elemento designado *Timestamp*. A estação pode escutar passivamente todos os ***Beacons*** enviados pelos pontos de acesso da sua área de cobertura e sincronizar o seu relógio com um dos pontos de acesso. Uma vez obtida uma lista válida de ***Beacons***, a estação selecciona o ponto de acesso. Esta selecção baseia-se em vários critérios (parâmetros da camada física, SSID...). Este método de escolha do ponto de acesso é conhecido como passivo (*Passive Scanning*).

As estações IEEE 802.11 podem ainda utilizar um outro método designado como activo (*Active Scanning*). Neste a estação de forma pró-activa procura os pontos de acesso na área de cobertura do sinal, através do envio de tramas do tipo ***Probe Request***. Os pontos de acesso que receberem estas tramas devem responder com uma trama do tipo ***Probe Response***. Estas

tramas contêm informação idêntica aos ***Beacons*** referidos no parágrafo anterior. Uma vez mais a estação escolhe o ponto de acesso que satisfaz os critérios pretendidos pela estação.

3.5.2 Autenticação

A próxima etapa é a autenticação. Existem dois tipos de autenticação tal como anteriormente referido: (i) autenticação aberta – este mecanismo retorna uma resposta positiva a todos os pedidos de autenticação e deve ser apenas utilizado em situações que não requerem a validação de uma estação cliente; (ii) autenticação de chave partilhada – mecanismo que suporta a autenticação de uma estação cliente que tenha conhecimento prévio de uma chave secreta partilhada com o ponto de acesso. A secção 4.1 descreve com detalhe estes mecanismos de autenticação.

3.5.3 Associação

Até esta etapa, a estação é desconhecida do sistema de distribuição DS. Após o processo de autenticação é necessário activar o encaminhamento de pacotes para o destino apropriado, o que só é possível após a associação da estação com o ponto de acesso. Esta associação, cria uma ligação lógica entre a estação e o ponto de acesso. Se a associação for aceite, o ponto de acesso irá comunicar o endereço MAC da estação (fornecido aquando do pedido de associação, através de uma trama ***Association Request***) ao sistema de distribuição.

O sistema de distribuição toma conhecimento da associação da estação num BSS especifico para proceder ao encaminhamento de pacotes no interior do ESS. O processo de associação termina com o envio de uma trama do tipo ***Association Response*** onde é enviado o identificador da associação, que será utilizado pela estação para comunicações adicionais com o ponto de acesso.

3.5.4 Transferência de dados

Concluídos os processos de autenticação e associação os dados podem ser enviados para o mesmo BSS, para o mesmo ESS ou para outros ESS. O sistema de distribuição auxilia o AP

no encaminhamento dos pacotes para o destino apropriado. Quando o AP recebe um pacote de uma estação, este verifica se o destino é o mesmo BSS.

Se o destino for o mesmo BSS, o AP encaminha-o para o BSS. Se o destino é outro BSS, utiliza a informação fornecida pelo sistema de distribuição para encaminhar o pacote para outro BSS ou para um Portal, para que este o envie para a rede cablada.

3.5.5 Reassociação

Uma das características do IEEE 802.11 é a possibilidade de *Roaming* entre vários BSSs. O processo de registo ou de estabelecer uma nova ligação lógica com outro BSS é designado por processo de reassociação (*Reassociation*).

O processo de reassociação ocorre devido à possível mobilidade das estações IEEE 802.11 entre vários pontos de acesso. Esta mudança pode ocorrer devido a variações no canal rádio ou devido ao aumento de carga no ponto de acesso inicial. Este processo de reassociação permite extender as áreas de cobertura do sinal RF, tão necessária em aplicações empresariais bem como em acessos wireless públicos conhecidos como *Hot Spots*.

O processo de reassociação auxilia o sistema de distribuição a manter actualizada toda a informação referente às estações associadas aos respectivos pontos de acesso, permitindo desta forma que o encaminhamento da informação de e para uma estação possa ocorrer em qualquer momento. Em determinado momento uma estação apenas pode estar associada a um ponto de acesso.

O novo ponto de acesso ao qual a estação se associou contacta o ponto de acesso anterior de modo a que a associação respeitante à estação seja retirada e para obter todos os pacotes que ainda não tenham sido entregues à estação após esta se associar ao novo AP.

O mecanismo de comunicação entre dois pontos de acesso não está normalizado. O grupo de trabalho 802.11f do IEEE está a desenvolver um protocolo com essa finalidade designado IAPP (*Inter-Access Point Protocol*).

4 Mecanismos de Segurança da Norma IEEE 802.11b

Este capítulo aborda as características de segurança integradas na norma IEEE 802.11b-1999 [2]. Nos capítulos seguintes esta norma será apenas referida como norma IEEE 802.11, uma vez que os assuntos abordados nesta dissertação são comuns às diversas camadas físicas referidas na secção 3.2.

A norma identifica e define um conjunto de serviços de segurança para tornar as comunicações sobre um canal de transmissão seguras. A segurança e protecção dos dados é efectuada ao nível da ligação e durante a transmissão entre estações e pontos de acesso, ou seja a protecção ponto-a-ponto entre estações wireless não é fornecida por tais mecanismos de segurança.

Os serviços de segurança base definidos na norma para implementação numa rede não-cablada são idênticos aos serviços de segurança de uma rede de comunicações cablada: a integridade, a autenticação e a confidencialidade. Este são os únicos serviços de segurança abordados pela norma IEEE 802.11. O serviço de não repúdio não é implementado.

A norma especifica um meio de verificar a integridade de uma mensagem transmitida entre uma estação e o ponto de acesso. Este mecanismo de segurança foi desenvolvido com o objectivo de rejeitar qualquer mensagem alterada durante a fase de transmissão. O mecanismo de integridade não é mais do que o algoritmo de detecção de erros CRC-32 abordado na secção 3.4.7.

Os mecanismos de autenticação e confidencialidade definidos na norma IEEE 802.11 são analisados nas secções 4.1 e 4.2 respectivamente. A secção 4.3 descreve as vulnerabilidades associadas aos diversos mecanismos de segurança do IEEE 802.11. Finalmente a secção 4.4 apresenta a demonstração prática de algumas das vulnerabilidades analisadas na secção 4.3.

4.1 Autenticação

A norma IEEE 802.11 define dois métodos de validação da estação que pretende autenticar-se com um ponto de acesso que são conhecidos como autenticação por chave partilhada (*Shared-Key Authentication*) e autenticação aberta (*Open System Authentication*). Este dois métodos de autenticação distinguem-se respectivamente pela utilização ou não de chaves criptográficas.

Actualmente existem além destes, outros dois métodos de autenticação que apesar de não serem definidos pela norma IEEE 802.11 são implementados pela maioria dos fabricantes de dispositivos wireless. Esses mecanismos são conhecidos como, autenticação por endereço MAC e autenticação pelo identificador SSID (*Service Set Identifier*).

O SSID é um mecanismo que permite essencialmente a separação lógica de várias redes não-cabladas. Uma estação deve estar configurada com o SSID referente à rede à qual pretende aceder. Este mecanismo não deve funcionar por si só como um mecanismo de autenticação dada a facilidade com que pode ser atacado. Uma análise detalhada de possíveis ataques a redes que recorrem a este mecanismo de autenticação é efectuada na secção 4.4.1.

O mecanismo de autenticação baseado no endereço MAC de uma estação também é um mecanismo de autenticação vulnerável a ataques quando utilizado isoladamente. A secção 4.4.2 faz uma análise de ataques a este mecanismo de autenticação.

Os mecanismos de autenticação da norma IEEE 802.11 referem-se à autenticação de estações ou dispositivos wireless e não a autenticação de utilizadores. Este facto implica no caso de roubo de uma estação móvel que estes possam ser utilizados por um atacante, para acesso indevido à rede wireless.

O processo de autenticação consiste na troca da informação ilustrada na Figura 4-1.

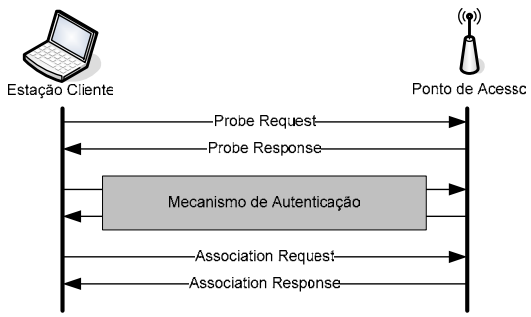


Figura 4-1 – Processo de autenticação de uma estação 802.11.

A estação quando activada envia tramas do tipo ***Probe Request*** em busca de um ponto de acesso e os pontos de acesso na área de cobertura do sinal rádio respondem com uma trama do tipo ***Probe Response***. De seguida a estação decide, de acordo com vários critérios (parâmetros da camada física, SSID...), a que ponto de acesso se pretende autenticar e dá inicio ao pedido de autenticação. As tramas trocadas nesta fase dependem do mecanismo de autenticação utilizado.

Após receber a indicação de sucesso a estação dá inicio ao processo de associação enviando uma trama ***Association Request***. O ponto de acesso deve reponder com uma trama do tipo ***Association Response*** com a indicação que a associação foi aceite. A partir deste instante o sistema de distribuição tem conhecimento da "localização" da estação e pode encaminhar informação de e para a estação.

As duas secções seguintes apresentam detalhadamente o processo de autenticação utilizando respectivamente os mecanismos de autenticação aberta e de chave partilhada.

4.1.1 Autenticação Aberta

O método de autenticação aberta é o mais simples. Essencialmente é um mecanismo de autenticação nulo. Se um ponto de acesso está a operar em modo de autenticação aberta, este irá aceitar todos os pedidos de autenticação.

Basicamente corresponde a um simples esquema do tipo pergunta-resposta, ilustrado na Figura 4-2. A primeira etapa da sequência corresponde ao envio da mensagem ***Authentication Request*** por parte da estação. A segunda fase, corresponde à resposta positiva do ponto de acesso através da mensagem ***Authentication Response***.

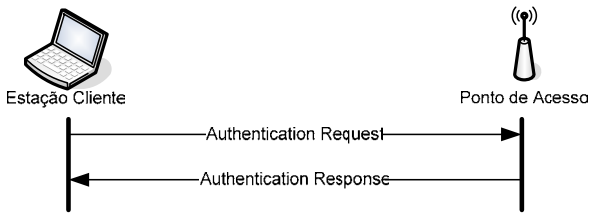


Figura 4-2 – Mensagens trocadas no mecanismo de autenticação Open System.

A Figura 4-3 e Figura 4-4 mostram dois pacotes capturados na fase de autenticação de uma estação utilizando o mecanismo de autenticação aberta, do tipo ***Authentication Request*** e ***Authentication Response*** respectivamente.

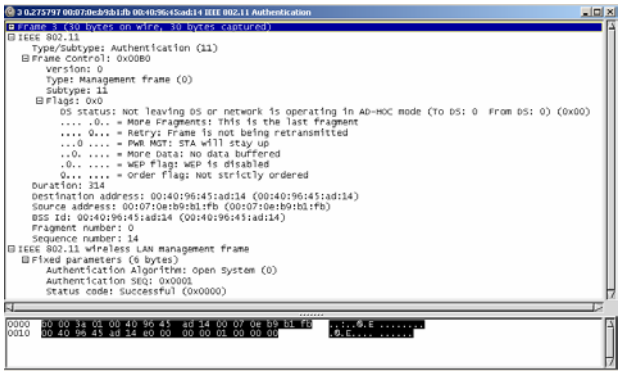


Figura 4-3 –Authentication Request.

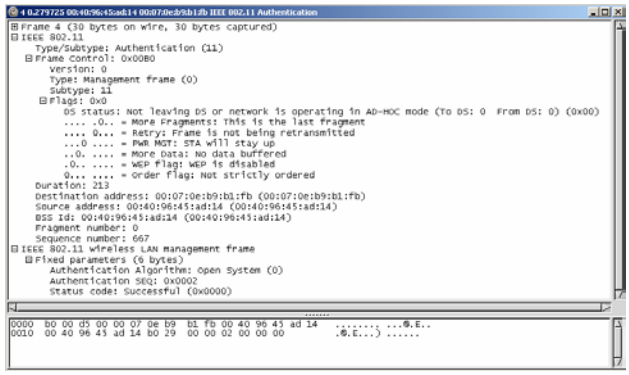


Figura 4-4 – Authentication Response.

O método de autenticação é identificado no campo ***Authentication Algorithm*** que neste caso toma o valor 0. A ordem da trama de autenticação é representada pelo campo ***Authentication Transaction Sequence Number***. O sucesso do pedido de autenticação é confirmado no campo ***Status Code*** que possui o valor 0 em ambas as tramas de autenticação.

4.1.2 Autenticação de Chave Partilhada

O mecanismo de autenticação de chave partilhada suporta a autenticação de estações, que tenham conhecimento prévio de uma chave secreta partilhada com o ponto de acesso ao qual pretendem autenticar-se. O método de distribuição da chave secreta utilizada não é definido na norma IEEE 802.11. A chave secreta é previamente entregue de forma segura a todas as estações intervenientes. Este método recorre ao mecanismo de confidencialidade WEP.

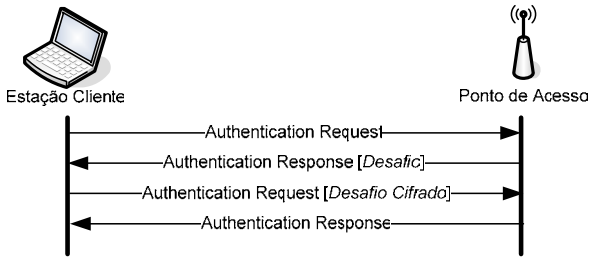


Figura 4-5 – Mensagens trocadas no mecanismo autenticação de chave partilhada.

A Figura 4-5 descreve o processo de autenticação de chave partilhada. A estação envia um pedido de autenticação, ***Authentication Request*** (Figura 4-6) para o ponto de acesso com a identificação do algoritmo de autenticação que pretende utilizar.

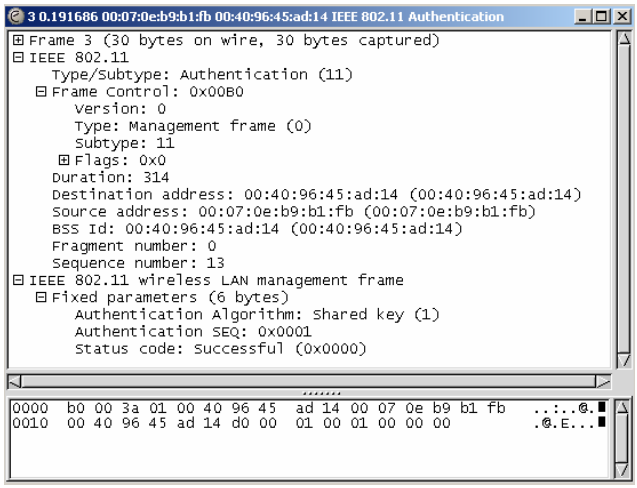


Figura 4-6 – Authentication Request.

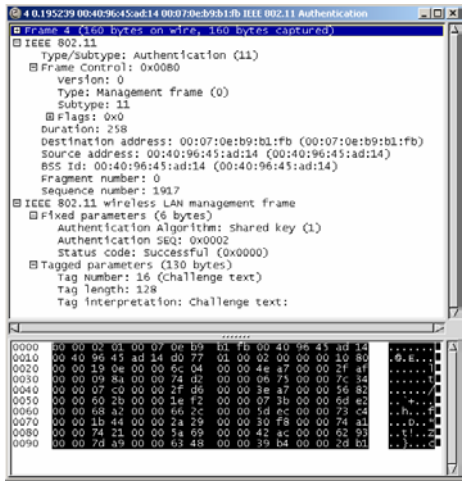


Figura 4-7 – Authentication Response.

O ponto de acesso responde enviando um ***Authentication Response*** (Figura 4-7) que contêm um desafio (*Challenge text*) com tamanho 128 bits.

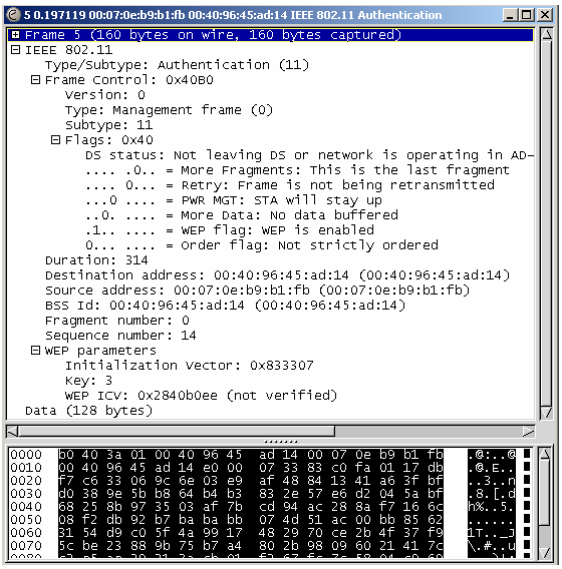


Figura 4-8 – Authentication Request.

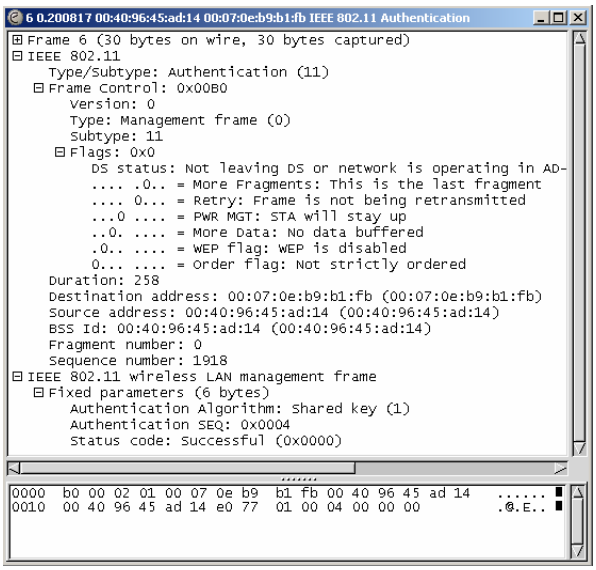


Figura 4-9 – Authentication Response.

A estação responde com uma nova trama, do tipo ***Authentication Request***, com o desafio recebido inicialmente cifrado com a chave WEP. Como se verifica pela Figura 4-8 o bit ***WEP*** tem valor 1, o que significa que o campo ***Frame Body*** desta trama de autenticação contém informação cifrada com o algoritmo WEP, no caso o desafio.

O ponto de acesso decifra a trama recebida com a sua chave e compara o desafio decifrado com aquele que foi enviado inicialmente. Tal operação é executada com sucesso se a chave secreta da estação coincidir com a chave secreta do ponto de acesso. Se esta operação é

efectuada com sucesso é transmitida à estação uma trama ***Authentication Response*** (Figura 4-9), com o valor do campo ***Status Code*** igual a 0.

4.2 Wired Equivalent Privacy WEP– Confidencialidade

Um problema comum aos utilizadores de tecnologias de rede cabladas e não-cabladas, é a privacidade ou confidencialidade dos dados que circulam na rede. A norma IEEE 802.11 especifica um algoritmo que pretende fornecer confidencialidade ao nível da ligação, designado WEP (*Wired Equivalent Privacy*). No entanto a sua utilização não é obrigatória.

Este serviço tem como objectivo, fornecer à rede não-cablada as mesmas funcionalidades presentes nas redes convencionais. A confidencialidade dos dados está dependente de um serviço externo de gestão de chaves, que permita a distribuição e gestão das chaves de cifra/decifra da informação que circula na rede.

4.2.1 Operação do WEP

À semelhança dos algoritmos criptográficos convencionais, o WEP recorre a uma chave secreta, k, para alterar os resultados na sua saída. Simbolicamente tem-se que uma função de cifra E que opera sobre o texto plano P de modo a produzir o texto cifrado C.

$$E_k(P)=C \qquad \text{Equação 4-1}$$

O processo inverso, função de decifra D, opera sobre o texto cifrado C, para recuperar o texto plano P.

$$D_k(C)=P \qquad \text{Equação 4-2}$$

Tal como referido anteriormente o WEP utiliza o algoritmo de chave simétrica RC4. Assim a mesma chave secreta k é utilizada para cifrar e decifrar os dados. Simbolicamente tem-se que:

$$D_k(E_k(P))=P \qquad \text{Equação 4-3}$$

O mecanismo de confidencialidade, proporcionado pelo protocolo WEP é ilustrado conceptualmente na Figura 4-10.

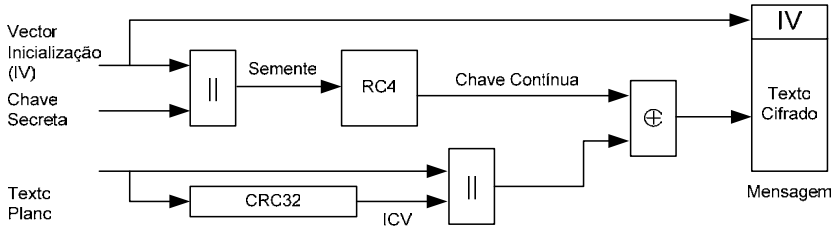


Figura 4-10 – Diagrama de blocos do codificador WEP.

Efectuando a análise à Figura 4-10, da esquerda para a direita e atendendo a que a chave secreta *k* é partilhada entre as partes que pretendem estabelecer uma comunicação entre si, o processo de cifra de uma trama processa-se do seguinte modo.

A chave secreta *k* é concatenada com o vector de inicialização IV. O resultado desta operação, designada na norma por *seed* ou "semente" é utilizada como entrada do algoritmo de cifra contínua RC4. Esta função vai gerar na sua saída uma chave contínua *RC4(IV,k)*, de bytes pseudo aleatórios com comprimento igual ao número de bytes de dados que serão transmitidos na trama, mais quatro bytes referentes ao valor de verificação de integridade do texto plano, ICV. Este valor é obtido através do algoritmo de integridade aplicado ao texto plano *P*. Neste caso, o algoritmo de integridade é o algoritmo de detecção de erros CRC-32. Como se verifica pela Figura 4-10, são aplicados dois processos ao texto plano, um processo de integridade de dados e um processo de cifra.

Para a protecção da alteração de dados não autorizada é utilizado o algoritmo de detecção de erros CRC-32 tal como referido no parágrafo anterior. O processo de cifra é aplicado combinando matematicamente a chave contínua com a saída da função de concatenação do ICV com o texto plano *P*. A combinação matemática é simplesmente uma operação ou-exclusivo, XOR, representada simbolicamente por ⊕. O resultado deste processo de cifra é uma mensagem composta pelo vector de inicialização IV e pelo texto cifrado *C*.

O vector de inicialização tem como funcionalidade principal o aumento do tempo de vida útil da chave secreta *k*. Normalmente a chave secreta mantém o seu valor constante durante um determinado período de tempo que pode ser pequeno ou elevado. Este valor depende directamente da política de gestão de chaves implementada pelos responsáveis.

A norma não define como deve ser alterado o valor do vector de inicialização. No entanto alerta para o perigo de utilização do mesmo par *(IV,k)* em pacotes sucessivos. Uma vez que a chave secreta *k* se mantém constante, a repetição de um vector de inicialização origina a

repetição da chave contínua *RC4(IV,k)* gerada pelo algoritmo RC4. A ocorrência de uma situação deste tipo pode reduzir significativamente o grau de confidencialidade oferecido pelo protocolo WEP, permitindo a um atacante decifrar os dados, sem o conhecimento da chave secreta *k*. A problemática relacionada com a repetição de chaves contínuas é abordada na secção 4.3.1.

Terminado o processo de cifra, o texto cifrado *C* e o vector de inicialização (não cifrado) são enviados através do canal de transmissão. Uma análise detalhada do formato de uma trama cifrada com o algoritmo WEP é apresentada na secção 4.2.2.

O processo inverso, é iniciado aquando da recepção da mensagem *M*. O processo é ilustrado na Figura 4-11.

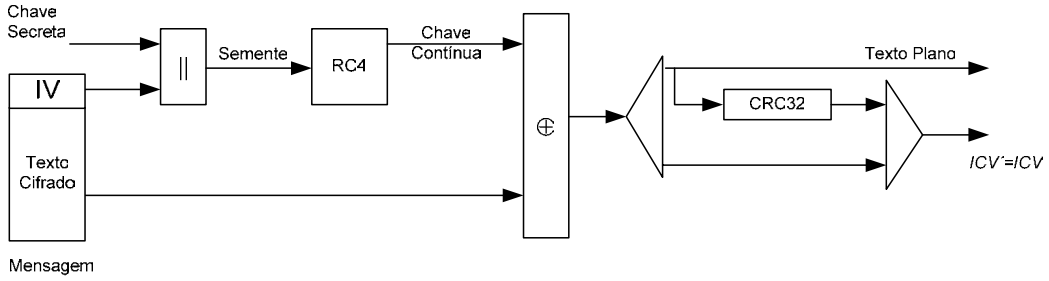


Figura 4-11 - Diagrama de blocos do descodificador WEP.

O vector de inicialização enviado na mensagem *M* e a chave secreta *k* previamente distribuída são utilizados para gerar a chave contínua *RC4(IV,k)*, necessária para decifrar a mensagem. Para tal é combinado matematicamente, recorrendo à operação XOR, o texto cifrado *C* e a chave contínua. Desta forma é obtido o texto plano *P* referente ao texto cifrado *C* assim como o vector de verificação de integridade do texto plano, ICV.

Para verificar a integridade dos dados é executado o algoritmo de integridade CRC-32 sobre o texto plano *P* obtido. O resultado é um vector de verificação de integridade ICV' que é comparado com o ICV transmitido na mensagem. Se o ICV não é igual a ICV' então as tramas não são encaminhadas para a camada LLC, assegurando-se desta forma que apenas tramas com *checksum* válido são aceites pelo receptor.

4.2.2 Formato da Trama Cifrada

Como foi referido anteriormente o algoritmo WEP é aplicado ao campo ***Frame Body*** de um MPDU (*MAC layer protocol data units*). O conjunto {IV+PAD+KeyID, *Frame Body*, ICV}

constitui os dados transmitidos numa trama de dados. A Figura 4-12 ilustra o formato de uma trama de dados cifrada com o algoritmo WEP.

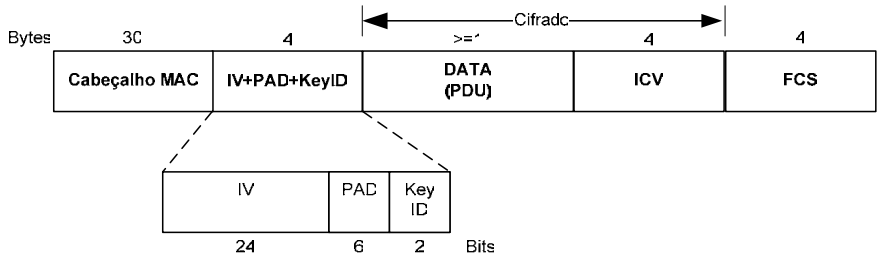


Figura 4-12 – Formato do pacote cifrado com algoritmo WEP.

Uma trama de dados cifrada inclui um campo de 32 bits, imediatamente antes do campo original de dados. O campo ***IV+PAD+KeyID*** contém três sub-campos: um campo inicial com tamanho de 3 bytes que contém o valor do vector de inicialização ***IV***, o campo ***PAD*** é composto por 6 bits com valor igual a zero e o campo ***KeyID*** com 2 bits cujo valor identifica a chave secreta utilizada para decifrar os dados. De acordo com a norma podem ser definidas nos dispositivos wireless até quatro chaves secretas. O campo ***ICV*** (*Integrity Check Value*) é composto por quatro bytes e contém o resultado da aplicação do algoritmo CRC-32 calculado sobre o campo ***PDU*** (*Protocol Data Unit*).

Valor de verificação de integridade (Integrity Check Value)

O objectivo do ICV é detectar a alteração de uma mensagem em trânsito. Em todas as tramas transmitidas, cifradas ou não, é efectuada uma verificação para detectar se quaisquer bits foram corrompidos durante a transmissão, recorrendo ao algoritmo CRC-32.

A integridade dos dados cifrados é fornecida pelo valor do ICV calculado sobre a trama de dados, antes do processo de cifra. A protecção de integridade do MPDU é fornecida através do campo ***FCS*** (secção 3.4.7).

A justificação inicial para a utilização do CRC-32 para protecção de integridade é que, uma vez que o ICV é cifrado, um atacante não pode recalculá-lo quando tenta alterar uma mensagem. Tal não é correcto como se verificará na análise dos ataques ao WEP.

4.3 Vulnerabilidades do WEP

Esta secção analisa em detalhe alguns ataques ao protocolo WEP. O WEP deve ser entendido como um termo genérico que engloba uma série de mecanismos de segurança. Para avaliar o WEP quanto à segurança proporcionada é necessário analisar os aspectos de autenticação, prevenção de repetição de mensagens, detecção de modificação de mensagens, confidencialidade das mensagens e protecção da chave secreta.

Uma primeira apreciação a estes mecanismos sugere a sua fiabilidade. No entanto uma análise mais detalhada permite concluir que tal não é verdade. Infelizmente o WEP falhou na implementação de todos os mecanismos de segurança referidos anteriormente.

Em meados de 2000, aquando do aparecimento em grande escala das redes não-cabladas IEEE 802.11, a comunidade cientifica da área de segurança tomou um contacto directo com a tecnologia, o que permitiu uma análise crítica e o levantar de questões pertinentes relativamente às questões de segurança associadas ao protocolo WEP. Nesse mesmo ano surgiram uma série de relatórios e artigos enumerando diversas falhas. A primeira vulnerabilidade do protocolo WEP foi publicada em [7] onde é abordada a implementação do vector de inicialização. Em [8] novas falhas no desenvolvimento do protocolo WEP são referidas. São ainda descritos alguns ataques que provam as vulnerabilidades detectadas. A descrição de um ataque ao mecanismo de autenticação é efectuada em [9]. A combinação destes três artigos mostra que os mecanismos de segurança implementados pelo protocolo WEP podem ser facilmente comprometidos. A última análise [10] veio colocar a descoberto uma grave vulnerabilidade no desenvolvimento do protocolo WEP. Finalmente foi tornado público um ataque que prova a possibilidade de se obter as chaves secretas utilizadas pelo WEP em apenas algumas horas. Este ataque é conhecido como FMS (*Fluhrer Mantin and Shamir*) [11] [Link1].

4.3.1 Riscos Associados à Repetição de Chaves Contínuas

Analisa-se nesta secção a vulnerabilidade associada à repetição de chaves contínuas que pode ocorrer na implementação do mecanismo de confidencialidade definido no protocolo WEP.

Como referido anteriormente a função de cifra é implementada através da operação XOR entre a chave contínua gerada *RC4*(*IV*,*k*) e o texto plano *P* a cifrar. A função inversa, função de decifra, consiste em gerar uma chave contínua idêntica baseada no vector de inicialização e na chave secreta *k*. Posteriormente aplica-se a função matemática XOR entre a chave contínua e o texto cifrado *C*.

Uma vulnerabilidade bem conhecida das funções de cifra contínua é o facto de ser possível obter informação relevante do texto plano de duas mensagens quando elas são cifradas usando a mesma chave contínua.

Simbolicamente, C_i representa o texto cifrado, o texto plano é representado por P_i e a chave contínua gerada é representada por *RC4*(*IV*,*k*). Atente-se no seguinte:

Se

$$C_1 = P_1 \oplus RC4(IV,k) \text{ e } C_2 = P2 \oplus RC4(IV,k) \qquad \text{Equação 4-4}$$

Então

$$C_1 \oplus C_2 = (P_1 \oplus RC4(IV,k)) \oplus (P_2 \oplus RC4(IV,k)) = P_1 \oplus P_2 \qquad \text{Equação 4-5}$$

Como pode verificar-se a operação de XOR sobre dois textos cifrados com a mesma chave contínua origina a anulação desta, resultando que o XOR dos textos cifrados é igual ao XOR dos textos planos.

Conclui-se assim que a reutilização da chave contínua permite a obtenção do valor da operação de XOR de dois textos planos, sem que para tal exista conhecimento da chave secreta. Se um dos textos planos é conhecido então o outro texto plano é obtido imediatamente. Existem também técnicas que permitem recuperar P_1 e P_2 a partir apenas de $P_1 \oplus P_2$. Por exemplo, no caso em que P_1 e P_2 são textos em inglês é possível determiná-los a partir de $P_1 \oplus P_2$ explorando as redundâncias existentes na linguagem [12]. No caso geral, este tipo de ataque obriga a duas condições [8]: (i) captura de vários textos cifrados com a mesma chave contínua; (ii) conhecimento parcial de alguns dos textos planos.

Existem diversas formas de obter textos planos conhecidos. Muitos campos do tráfego IP são previsíveis, uma vez que a estrutura das mensagens destes protocolos são bem definidas, e os conteúdos das mensagens são normalmente previsíveis. Por exemplo, numa sequência de *login* o *prompt* da mensagem contêm normalmente o texto "Password".

Para prevenir este tipo de ataques, o protocolo WEP utiliza um vector de inicialização por trama com o objectivo de variar o valor da entrada do algoritmo RC4 que gera a chave contínua. O WEP gera a chave contínua em função da chave secreta *k* (a mesma para todas as tramas) e um vector de inicialização público (que varia para cada trama). Desta forma cada uma das tramas transmitidas será cifrada com uma chave contínua diferente.

Repetição do Vector de Inicialização

A utilização de um vector de inicialização por trama pretende prevenir ataques baseados na reutilização de chaves contínuas. No entanto esse objectivo não é alcançado devido a falhas nas implementações existentes no mercado e ao tamanho reduzido do vector de inicialização, de apenas 24 bits. A repetição do vector de inicialização é uma das principais causas para a repetição da chave contínua.

A norma IEEE 802.11 recomenda que o valor do vector de inicialização seja alterado para cada trama, mas tal não é obrigatório. Além do mais não especifica como deve ser alterado esse valor. Assim algumas implementações fazem-no de forma deficiente.

Vários dispositivos wireless atribuem o valor zero ao vector de inicialização sempre que estes são reinicializados, incrementando de um o seu valor para cada trama transmitida. Num cenário de rede não-cablada com um número elevado de estações, a reinicialização pode ocorrer com uma frequência considerável. Uma consequência imediata é a repetição das chaves contínuas geradas correspondentes aos valores iniciais do vector de inicialização durante o tempo de utilização da chave secreta *k*.

Um outro método implementado pelos fabricantes é a atribuição de valores aleatórios ao vector de inicialização. No entanto esta abordagem também apresenta as suas limitações devido ao que se designa por *Birthday Paradox* [8].

A designação deste paradoxo advém do facto de num grupo de 23 ou menos pessoas existir 50% de probabilidade do dia do aniversário de dois membros desse grupo ser o mesmo. Genericamente, se um conjunto tiver *n* elementos e estes forem seleccionados um de cada vez com reposição, então a probabilidade de haver uma repetição após duas extracções é $p_2=1/n$ e a probabilidade de haver pelo menos uma repetição após $k\geq 3$ extracções é dada por:

$$p_k=p_{k-1}+(k-1).1/\mathrm{n}.(1-p_{k-1}) \qquad \text{Equação 4-6}$$

Cálculos efectuados em [7] mostram que a probabilidade de repetição de vectores de inicialização, ou colisão de IVs, é de 99% após serem transmitidas 12430 tramas, o que representa 2 ou 3 segundos de tráfego a 11 Mb/s. Existe 10% de probabilidade de colisão após 1881 tramas transmitidas, 1% após a transmissão de 582 tramas, 0.1% após 184 tramas e 0.001% após 19 tramas.

Isto permite concluir que a atribuição aleatória de valores ao IV por forma a manter uma probabilidade de colisão de IVs de apenas 0.000001% requer a alteração da chave secreta após todos os membros de uma BSS transmitirem apenas seis pacotes com a mesma chave, o que em implementações práticas não acontece. Verifica-se assim que o tamanho de 24 bits do vector de inicialização é demasiado pequeno para proteger contra ataques por repetição de chaves contínuas.

4.3.2 Decifra por Ataque de Dicionário

Sempre que é obtido o texto plano de uma mensagem interceptada o atacante toma conhecimento do valor da chave contínua utilizada para cifrar a mensagem. Assim é possível utilizar essa chave contínua para decifrar qualquer outra mensagem que utilize o mesmo vector de inicialização.

Ao longo do tempo um atacante pode construir uma tabela de chaves contínuas correspondente a cada IV. O tamanho máximo duma tabela destas é de aproximadamente 24 GB, considerando que cada um dos 2^{24} IVs ocupa aproximadamente 1500 bytes. Um atacante após algum esforço consegue obter dados suficientes para construir um dicionário de decifra. Este tipo de ataque é particularmente fiável numa situação em que a chave secreta é alterada com pouca frequência.

Obviamente que o esforço necessário para implementar um dicionário deste tipo limita este ataque apenas aos atacantes mais persistentes. Pode argumentar-se que o protocolo WEP não foi desenvolvido com o intuito de proteger contra este tipo de atacantes. Dado que o tamanho da sua chave é de apenas 40-bits esta pode ser obtida através de um ataque de força bruta num espaço de tempo relativamente pequeno. Apesar dos fabricantes desenvolverem dispositivos que suportam tamanhos de chaves WEP superiores, esta solução não proporciona uma melhor protecção contra este tipo de ataques dado que o tamanho do dicionário a construir

não é superior. O tamanho do dicionário depende apenas do tamanho do vector de inicialização, especificado na norma com 24 bits, e não do tamanho da chave de cifra.

4.3.3 Alteração de Mensagens

Como referido na secção 3.4.7, a integridade das tramas é protegida através do algoritmo CRC-32. Este tipo de algoritmo não é suficiente para assegurar que um atacante não consiga alterar o conteúdo de uma trama, pois foi desenvolvido com a única intenção de detectar erros na transmissão. Não sendo um mecanismo de integridade baseado em criptografia, naturalmente possui vulnerabilidades que não permitem a sua utilização com eficácia para detectar violações à integridade.

Para compreender a vulnerabilidade inerente ao mecanismo de integridade CRC-32 utilizado, atente-se na propriedade genérica de linearidade presente no *checksum* do protocolo WEP. O *checksum* do WEP é uma função linear da mensagem transmitida. Isto significa que dadas duas mensagens x e y:

$$CRC(x \oplus y) = CRC(x) \oplus CRC(y) \qquad \text{Equação 4-7}$$

para qualquer valor de x e y. Esta é uma propriedade comum a todos os algoritmos de detecção de erros deste tipo. Uma consequência imediata desta propriedade é a possibilidade de efectuar alterações, ainda que controladas, ao texto cifrado, sem corromper o seu *checksum*. Atente-se no seguinte exemplo.

Considere-se C o texto cifrado, que é interceptado antes de alcançar o seu destino. Assumindo que C corresponde a uma mensagem M desconhecida e que $CRC(M)$ é o seu *checksum*, tem-se:

$$C = RC4(IV,k) \oplus M \,||\, CRC(M) \qquad \text{Equação 4-8}$$

Segundo [8] é possível encontrar um novo texto cifrado C' que quando decifrado origina uma nova mensagem M' dada por:

$$M' = M \oplus \Delta \qquad \text{Equação 4-9}$$

A alteração desejada no texto plano, dada por Δ, é escolhida pelo atacante.

Resta descrever a obtenção do novo texto cifrado C' a partir do texto cifrado original C de modo a que quando C' for decifrado origine a mensagem alterada M' e não a mensagem original M, mantendo válido o *checksum* da mensagem alterada.

Dada a linearidade dos algoritmos de cifra contínua como o RC4, em [8] é sugerida a aplicação da função matemática XOR de $\Delta||CRC(\Delta)$ aos dois lados da Equação 4-8 para obter o novo texto cifrado *C'*, onde $CRC(\Delta)$ representa o *checksum* da alteração na mensagem original. Note-se que $P=M||CRC(M)$. Simbolicamente tem-se:

$$\begin{aligned} C' &= C \oplus \Delta||CRC(\Delta) \\ &= RC4(IV,k) \oplus M||CRC(M) \oplus \Delta||CRC(\Delta) \\ &= RC4(IV,k) \oplus (M \oplus \Delta||CRC(M) \oplus CRC(\Delta) \\ &= RC4(IV,k) \oplus M'||CRC(M \oplus \Delta) \\ &= RC4(IV,k) \oplus M'||CRC(M') \end{aligned} \qquad \text{Equação 4-10}$$

Este resultado demonstra que é possível alterar um texto cifrado *C* para obter um novo texto cifrado *C'* que quando decifrado origina $P \oplus \Delta$. Um ataque deste tipo pode ser aplicado sem um conhecimento da mensagem *M*. O atacante apenas tem que conhecer o texto cifrado original *C* e a diferença desejada do texto plano, Δ, de modo a calcular $C'=C \oplus \Delta||C(\Delta)$.

4.3.4 Introdução de Mensagens

Uma outra vulnerabilidade descrita em [8] é a possibilidade de introduzir mensagens, nomeadamente mensagens associadas ao mecanismo de autenticação, como se pode verificar na secção 4.3.5. Os autores de [8] tiram partido de uma propriedade da função de *checksum* do WEP (CRC-32), que é a sua independência da chave secreta utilizada para cifrar a mensagem, e aproveitam o comportamento dos pontos de acesso que não alertam o receptor da mensagem aquando da reutilização de um IV. Estes aspectos são analisados em seguida para provar a incapacidade do WEP na protecção contra introdução de mensagens.

Tal como referido anteriormente, o conhecimento de um texto plano *P* e do respectivo texto cifrado *C* permite a obtenção da chave contínua correspondente. Esta pode ser reutilizada para gerar uma nova trama, utilizando o mesmo vector de inicialização. Isto é, se o atacante tomar conhecimento do texto plano *P* e do respectivo texto cifrado *C*, então pode recuperar a chave contínua $RC4(IV,k)$ utilizada no processo de cifra. Simbolicamente:

$$P \oplus C = P \oplus (P \oplus RC4(IV,k)) = RC4(IV,k) \qquad \text{Equação 4-11}$$

Assim um atacante pode construir um texto cifrado *C'* de uma nova mensagem *M'*, utilizando a mesma chave contínua

$$C'=M'||CRC(M') \oplus RC4(IV,k) \qquad \text{Equação 4-12}$$

De notar que a mensagem alterada *M'* utiliza o mesmo IV da mensagem original *M*. No entanto tal facto não constitui um impedimento para efectuar este ataque, dada a incapacidade dos dispositivos receptores para detectarem a reutilização de um IV. Uma vez conhecidos o vector de inicialização e a correspondente chave contínua *RC4*(*IV*,*k*) é possível utilizar a chave contínua indefinidamente para a introdução de novas mensagens cifradas.

4.3.5 Spoofing de Autenticação

Um caso especial em que a vulnerabilidade analisada na secção anterior pode ser explorada é na implementação de ataques ao mecanismo de autenticação de chave partilhada abordado na secção 4.1.2. Como se analisa na Figura 4-5, após uma estação efectuar um pedido de autenticação o ponto de acesso envia um desafio em texto plano, composto por uma *string* aleatória de 128 bits. A estação tem então que responder com o mesmo desafio cifrado. A autenticação tem sucesso se o desafio decifrado corresponder ao desafio enviado inicialmente.

No entanto tal como descrito na secção 4.3.4 é possível a introdução de mensagens cifradas, sem o conhecimento da chave secreta *k*. Para um ataque deste tipo (introdução de mensagens cifradas) é suficiente o conhecimento do conjunto texto plano/texto cifrado. Esta informação é facilmente obtida através da análise das tramas trocadas no processo de autenticação, descrito na secção 4.1.2 .

Em primeiro lugar o atacante captura a segunda e a terceira trama trocada durante a sequência de autenticação. A segunda trama contém o texto plano do desafio. A terceira, por sua vez, contém o mesmo desafio mas cifrado com a chave secreta *k*. Assim o atacante tem conhecimento do texto plano *P* e do correspondente texto cifrado *C*. O vector de inicialização IV também é do conhecimento do atacante, uma vez que este é transmitido em texto plano. Desta forma o atacante pode obter a chave contínua produzida com a chave secreta *k* e com o vector de inicialização IV, de acordo com a Equação 4-11.

O atacante tem agora os elementos necessários para se autenticar na rede com sucesso, sem ter conhecimento da chave secreta *k*. Para ultrapassar o mecanismo de autenticação de chave partilhada o atacante envia um pedido de autenticação ao ponto de acesso, em seguida este responde ao atacante com um desafio de autenticação em texto plano. Nesta fase o atacante já

calculou uma chave contínua *RC4(IV,k)* através do processo descrito no parágrafo anterior e já possui o valor do desafio em texto plano *R*. Em seguida o atacante calcula uma resposta de autenticação válida através do XOR do texto plano com a chave contínua *RC4(IV,k)* resultando no desafio cifrado. Finalmente o atacante procede ao cálculo de um CRC válido e responde com uma mensagem ***Authentication Request*** válida. Ultrapassado o mecanismo de autenticação o atacante pode associar-se à rede.

De referir que este ataque apenas funciona porque o texto desafio tem sempre um tamanho de 128 bytes e porque tal como referido anteriormente, é possível a reutilização e repetição do vector de inicialização.

4.3.6 Obtenção da Chave Secreta

Existem actualmente diversas ferramentas disponíveis na Internet, tais como Airsnort [Link1], ou WEPCrack [Link2], que permitem a obtenção da chave secreta utilizada pelo WEP para cifrar os dados. Estas ferramentas recorrem ao chamado ataque FMS (Fluhrer, Mantin, Shamir) [10] [11]. O ataque FMS processa a informação capturada e determina byte a byte o valor provável da chave secreta. Se a quantidade de tramas capturadas for suficientemente elevada (entre 2,000,000 e 5,000,000) o valor da chave é determinado com uma margem de erro bastante pequena [13].

O ataque FMS recorre essencialmente aos seguintes aspectos:

1. O formato de alguns IVs permite a obtenção da chave secreta byte a byte com um grau de probabilidade relativamente elevado (5%). Estes são os chamados IVs fracos.
2. O primeiro byte do texto plano sujeito à operação de cifra é normalmente igual a 0xAA (SNAP Header). Deste modo, através da captura do 1º byte do texto cifrado é possível determinar o primeiro byte da chave contínua.

Este ataque é relativamente complexo. Nesta secção iremos apenas ilustrar a forma como o 1º byte da chave secreta pode ser descoberto.

Ataque ao 1º byte da chave secreta

Um IV fraco que permite o ataque ao 1º byte da chave secreta é o representado na Figura 4-13. Nesta figura *N* corresponde ao número de ciclos do algoritmo KSA do RC4 (tipicamente assume o valor 256) e *X* representa qualquer valor entre 0 e 255.

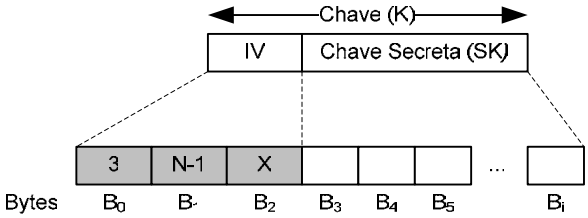


Figura 4-13 – Formato de um vector de inicialização fraco.

Para facilitar a compreensão do ataque ao 1º byte da chave secreta *SK* recordam-se na Tabela 4-1 os algoritmos KSA e PRGA que foram apresentados na secção 2.3.1. Recorde-se também que *l* corresponde ao comprimento em bytes da chave *K*.

KSA(K, l)	*PRGA (l)*
#Preenche o vector S-Box com valores de 0 *a N* *for i*=0 *to N*-1 *S*[*i*]= *i* *end for* *j*=0 *#Scrambling S-box usando a chave K* *for i*=0 *to N*-1 *j*=*j*+*S*[*i*]+*K*[*i mod l*] *mod N* *swap*(*S*[*i*], *S*[*j*]) *end for*	*i*=0 *j*=0 # *Ciclo de geração de z* *for K*=0 *to l*-1 *i*=(*i*+1) *mod N* *j*=(*j*+*S*[*i*]) *mod N* *Swap*(*S*[*i*], *S*[*j*]) *z*=*S*[(*S*[*i*]+*S*[*j*]) *mod N*] *return byte chave contínua z* *end for*

Tabela 4-1 – Algoritmos KSA e PRNG

Para efeitos de demonstração assume-se que *N*=16 e consideram-se os seguintes valores para *IV* e *K*:

- *IV*=[3, 15, 7]
- *SK* = [1, 2, 3, 4, 5]

Deste modo:

- *K* = [3, 15, 7, 1, 2, 3, 4, 5]

Considera-se também que o 1º byte do texto cifrado capturado pelo atacante tem o valor 164.

Assume-se que os elementos de *S* não são trocados quando *i* > 3 (5% probabilidade de ocorrência).

Inicialmente o atacante tem conhecimento do IV e consequentemente dos 3 primeiros bytes da chave secreta *K*. O primeiro byte da chave secreta *SK*, valor que o atacante pretende obter é representado por *K*[3]. Assim,

$$K[0]=3 \; ; \; K[1]=15; \; K[2]=7; \; K[3]=?$$

As três primeiras iterações do algoritmo KSA são facilmente simuladas pelo atacante dado o conhecimento dos valores anteriores. Em seguida são apresentadas estas três iterações.

KSA_0 (1ª Iteração)

A primeira fase do algoritmo corresponde à inicialização do vector de estados *S*. Para $N = 16$ tem-se que:

$$S_0=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$

Após a inicialização do vector de estados, dá-se inicio ao processo de mistura (*scrambling*) do vector *S*. Para os índices $i_0=0$ e $j_0=0$ tem-se.

$$j_1=j_0+S_0[i_0]+K[i_0 \; mod \; l] \; mod \; N = 0+S_0[0]+K[0 \; mod \; 8] \; mod \; 16 = 0+0+3 \Rightarrow j_1=3$$

Note-se que no caso do IV fraco o valor de j_1 será sempre igual a 3.

O processo de troca (*swap*) utiliza então os valores $i_0=0$ e $j_1=3$ para determinar que posições do vector *S* devem ser trocadas. Os valores de $S_0[0]$ e $S_0[3]$ são trocados entre si. No final da primeira iteração o vector de estados *S* passa a ter o seguinte valor.

$$S_1=[\mathbf{3}, 1, 2, \mathbf{0}, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$

Neste momento o atacante avança para a segunda iteração.

KSA_1 (2ª Iteração)

Para $i_1=1$ e $j_1=3$ tem-se que o novo valor de j_2 é dado por:

$j_2=j_1+S_1[i_1]+K[i_1 \; mod \; l] \; mod \; N = 3+S_1[1]+K[1 \; mod \; 8] \; mod \; 16 = 3+1+\mathrm{K}[1] \; mod \; 16 = [4+15] \; mod \; 16 \Rightarrow j_2=3$

Note-se que no caso do IV fraco j_2 será sempre igual a 3 uma vez que

$$j_2=3+1+N\text{-}1 \; mod \; N = 3$$

Assim os valores de $S_1[1]$ e $S_1[3]$ são trocados entre si. Assim no final da segunda iteração o vector de estados *S* passa a ter o seguinte valor.

$$S_2=[3, \mathbf{0}, 2, \mathbf{1}, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$

KSA_2 (3ª Iteração)

Para $i_2=2$ e $j_2=3$ têm-se que o novo valor de j_3 é dado por:

$j_3=j_2+S_2[i_2]+K[i_2 \bmod l] \bmod N = 3+S_2[2]+K[2 \bmod 8] \bmod 16 = 3+2+K[2] \bmod 16 = [5+7] \bmod 16 \Rightarrow j_3=12$

Os valores de de $S_2[2]$ e $S_2[12]$ são trocados entre si. Assim no final da terceira iteração o vector de estados S passa a ter o seguinte valor.

$$S_3=[3, 0, \mathbf{12}, 1, 4, 5, 6, 7, 8, 9, 10, 11, \mathbf{2}, 13, 14, 15]$$

Note que, no caso do IV fraco, após esta iteração será sempre $S_3[0] = 3$ e $S_3[1] = 0$.

Se o atacante avançar para a simulação da quarta iteração do KSA não conseguirá prosseguir uma vez que não tem conhecimento do valor de $K[3]$. Como referido previamente este ataque assume que para $i > 3$, os elementos do vector S não são trocados

Para perceber o ataque ao 1º byte é preciso entender a 1ª iteração do PRGA que conduz à determinação do 1º byte da chave contínua que designaremos por z.

$PRGA_0$ (1ª Iteração)

O vector de estados S é dado por:

$$S=S_3=[3, 0, 12, 1, 4, 5, 6, 7, 8, 9, 10, 11, 2, 13, 14, 15]$$

Para $i_0=0$ e $j_0=0$ tem-se que:

$$i_1=(i_0+1) = 0+1 \Rightarrow i_1=1$$

$$j_1=(j_0+S_3[i_1]) = 0+S_3[1] = 0+0 \Rightarrow j=0$$

O processo de troca é entre as posições $S_3[1]$ e $S_3[0]$. Assim S passa a ter o seguinte valor.

$$S=[\mathbf{0}, \mathbf{3}, 12, 1, 4, 5, 6, 7, 8, 9, 10, 11, 2, 13, 14, 15]$$

Finalmente é calculado o primeiro byte da chave de cifra contínua z.

$$z= S[(S[i_1]+S[j_1])] = S[(S[1]+S[0])] = S[0+3] = S[3]$$

Conclui-se então, no caso do IV fraco, o valor do 1º byte da chave contínua é dado por $z=S[3]$.

O atacante pode determinar o 1º byte da chave contínua através do conhecimento do 1º byte do texto cifrado e do 1º byte do texto plano, ou seja

$$z = \textit{byte texto cifrado} \oplus \textit{byte texto plano} = 164 \oplus 0xAA = 14.$$

É na 4ª iteração do KSA que intervém o 1º byte da chave secreta que o atacante pretende determinar. Analisemos então esta iteração.

KSA_3 (4ª Iteração)

Inicialmente $i_2 = 3$ e $j_2 = 12$. Assim,

$$j_3 = j_2 + S_2[3] + K[3]$$

pelo que

$$K[3] = j_3 - j_2 - S_2[3]$$

O valores de j_2 e $S_2[3]$ são conhecidos mas o valor de j_3 não. Este valor pode ser determinado analisando a operação de SWAP. Note-se que

$$SWAP\ (S_2[3], S_2[j_3])$$

resulta em $S_3[3] = S_2[j_3]$ e $S_3[j_3] = S_3[3]$. Uma vez que da 1ª iteração do PRGA verificámos que $z = S[3]$ conclui-se que $S_2[j_3] = z$. Assim, j_3 corresponde à posição do vector S em que estiver armazenado o valor z. Neste caso,

$$j_3 = S_2^{-1}[z] = S_2^{-1}[14] = 14$$

Finalmente,

$$K[3] = 14 - 12 - 1 = 1$$

O atacante conseguiu determinar o valor do 1º byte da chave secreta.

O ataque aos restantes bytes da *SK* exige uma análise mais complexa e recorre a IVs com um padrão distinto do apresentado inicialmente. Em [Link 12] é descrito um algoritmo de análise para detectar o padrão de IVs mais adequado à obtenção dos restantes bytes da *SK*. A secção 4.4.5 apresenta o resultado da obtenção de uma chave secreta recorrendo ao programa *Airsnort*. Este programa explora a vulnerabilidade analisada nesta secção.

4.4 Ataques Práticos

No âmbito desta dissertação, foram realizadas experiências práticas com o objectivo de analisar as vulnerabilidades dos diversos mecanismos de segurança implementados pelo protocolo IEEE 802.11. Os ataques apresentados foram realizados em laboratório recorrendo à configuração apresentada na Figura 4-14, excepto quando referido o contrário.

Na secção 4.4.1 apresenta-se um ataque ao mecanismo de autenticação aberta. Na secção 4.4.2 é apresentado um ataque ao mecanismo de autenticação por listas de acesso de endereços MAC. Na secção 4.4.3 apresenta-se um ataque de intercepção (*Man in the Middle*) da comunicação. Na secção 4.4.4 é descrito um ataque de negação de serviço e finalmente na secção 4.4.5 apresenta-se um ataque ao mecanismo de confidencialidade WEP.

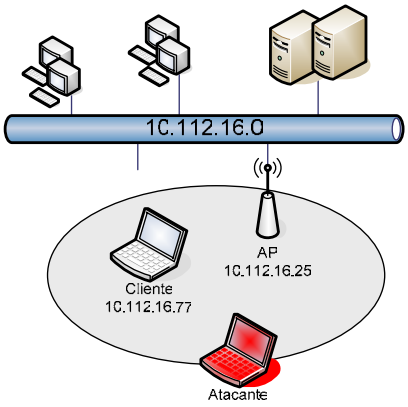


Figura 4-14 – Configuração utilizada em laboratório.

Para a realização das experiências recorreu-se a um ponto de acesso da série 350 comercializado pela empresa *Cisco Systems, Inc.* A versão de *firmware* instalada corresponde à versão 11.10t. O endereço MAC deste é 00:40:96:45:ad:14.

Relativamente à estação, cliente, que representa um utilizador legítimo da rede, possui um cliente wireless da *Cisco Systems, Inc*, mais concretamente uma placa PCMCIA PCM-352, com endereço MAC, 00:07:0e:b9:b1:fb. Esta estação possui instalado o sistema operativo Microsoft Windows 2000.

A estação utilizada para implementar os ataques, atacante, possui duas placas, com diferentes características, nomeadamente no que respeita ao *chipset* utilizado. Uma das placas é uma PCMCIA Compaq WL110 com um *chipset* Hermes, e endereço MAC 00:02:a5:6e:8a:53. A outra placa é uma PCMCIA Linksys WPC11 com um *chipset* PrismII, e endereço MAC

00:07:0e:b9:69:fc. O sistema operativo, utilizado pela estação atacante é baseado na distribuição do Red Hat Linux 7.3 com versão de Kernel 2.4.16.

Nas experiências práticas foram utilizadas diversas ferramentas que permitem implementar um conjunto diversificado de ataques: Analisador de protocolos, Ethereal versão 0.9.11 [Link4]; Ferramenta para captura de tramas IEEE 802.11, *Kismet* versão 2.8.1 [Link5]; Aplicação *Airjack* versão 0.6.2-alpha [Link6], desenvolvida originalmente por Robert Baird e Michael Lynn (esta é a única versão disponibilizada que permite implementar os ataques descritos neste documento; as versões posteriores têm funcionalidades limitadas e não permitem a implementação de alguns ataques); finalmente a aplicação já referida anteriormente e que permite determinar a chave secreta utilizada pelo algoritmo WEP, Airsnort versão 0.2.1.b [Link1].

4.4.1 Mecanismos de Autenticação

Esta experiência visa estudar as vulnerabilidades do mecanismo de autenticação aberta descrito na secção 4.1.1.

Inicialmente configurou-se o ponto de acesso para que este utilizasse o método de autenticação aberta. Desta forma o ponto de acesso permite a autenticação de qualquer estação que tenha conhecimento do seu SSID. Para esta experiência o SSID foi configurado no ponto de acesso com o valor "WLAN". Em seguida configurou-se a estação legítima, estação cliente, com o SSID respectivo. Esta acção permitiu à estação legítima a associação e respectiva autenticação ao ponto de acesso AP.

O que se pretende observar com esta experiência é a possibilidade da estação atacante obter o valor do SSID e posteriormente autenticar-se no ponto de acesso. Para efectuar este ataque é suficiente a utilização, por parte do atacante, de uma ferramenta que permita a captura de tramas IEEE 802.11, no caso a aplicação *Kismet*.

A Figura 4-15 apresenta uma trama de gestão IEEE 802.11 do tipo ***Beacon*** capturada pela estação atacante. Esta trama foi enviada pelo ponto de acesso para o endereço de *broadcast*.

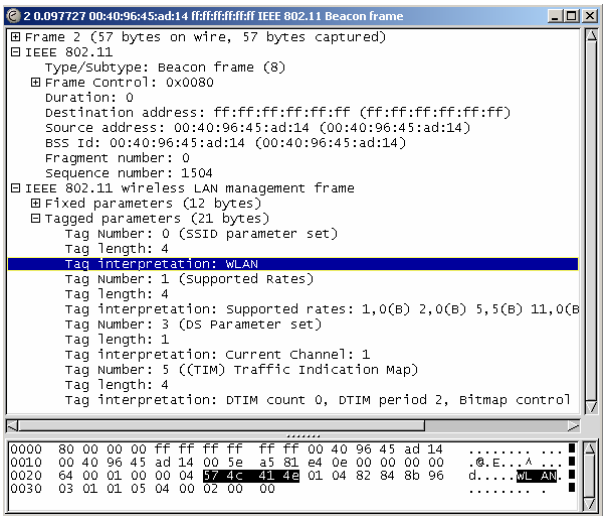


Figura 4-15 – Beacon Frame enviado pelo AP para o endereço de *Broadcast*.

Analisando a informação relativa ao SSID presente no campo ***Frame Body***, é possível verificar que o SSID configurado no ponto de acesso tem o valor "WLAN". Conclui-se assim com esta experiência que o envio do valor do SSID nas tramas de gestão do tipo ***Beacon*** compromete o mecanismo de autenticação aberta, pois após a obtenção desta informação a estação atacante autentica-se facilmente no ponto de acesso AP.

Para ultrapassar esta vulnerabilidade vários fabricantes de dispositivos wireless desenvolveram mecanismos que permitem configurar um ponto de acesso de modo a que este oculte o valor do SSID nas tramas de gestão do tipo ***Beacon***.

A experiência seguinte, descreve o ataque efectuado ao mecanismo de autenticação aberta com a funcionalidade referida no parágrafo anterior activa no ponto de acesso. A Figura 4-16 ilustra as diversas fases da experiência.

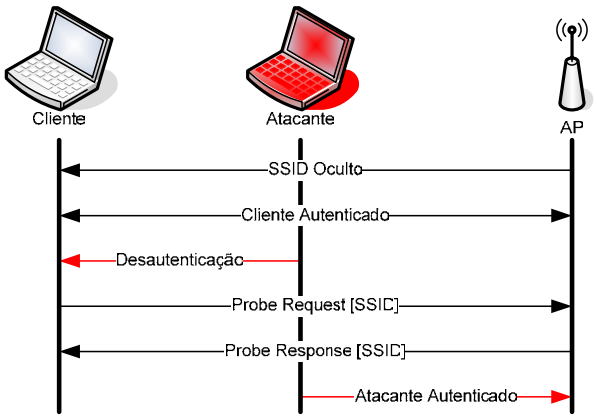


Figura 4-16 – Ataque ao mecanismo de autenticação com SSID oculto.

Procedeu-se inicialmente à configuração do ponto de acesso para ocultar o valor do SSID. A estação legítima, com conhecimento prévio do valor do SSID, é configurada com um SSID igual a "WLAN" e autentica-se no ponto de acesso. Nesta fase o atacante não possui qualquer conhecimento do SSID configurado na rede. A primeira tentativa para obter o SSID passou pela captura de tramas de gestão ***Beacon***. Uma trama capturada pela estação atacante é apresentada na Figura 4-17. Como se verifica o valor do SSID é ocultado neste tipo de tramas de gestão pelo ponto de acesso.

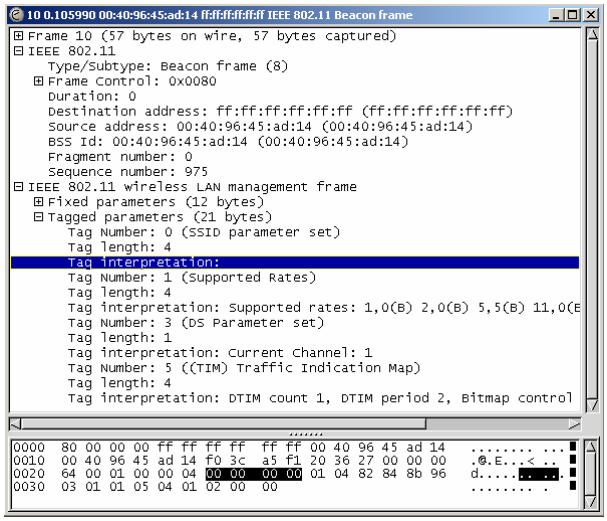


Figura 4-17 – Beacon Frame com SSID oculto.

Ao contrário do ataque anterior onde a simples análise do SSID transmitido nas tramas de gestão ***Beacon*** é suficiente, em pontos de acesso com capacidade para ocultar o SSID, a abordagem do atacante tem que ser outra. Esta passa pela análise de tramas de gestão do tipo ***Probe Request*** e ***Probe Response*** trocadas entre um ponto de acesso e uma estação na fase de autenticação.

Para tal o atacante tem duas opções distintas: (i) aguardar que um utilizador legítimo se autentique com o AP, analisando em seguida o valor do SSID presente nas tramas capturadas durante a fase de autenticação; (ii) forçar a desautenticação de um cliente legítimo e aguardar que este se autentique novamente.

A última opção é bastante mais rápida e igualmente eficaz. Para a experiência realizada optou-se por esta segunda opção. Para implementar este ataque configurou-se a estação atacante com a aplicação *Airjack*. O funcionamento desta ferramenta tira partido do facto das tramas de gestão enviadas e recebidas pelos dispositivos wireless não serem autenticadas. Inicialmente esta ferramenta faz-se passar pelo ponto de acesso legítimo dada a sua capacidade para forjar o

endereço MAC do ponto de acesso. Esta operação é também conhecida como *MAC Spoofing*. Em seguida envia uma trama de gestão do tipo ***Deauthentication*** para um utilizador legítimo, ou então para o endereço de *Broadcast*.

Na preparação do ataque o atacante têm que obter informação necessária para executar o ataque com sucesso, ou seja, o endereço MAC do ponto de acesso e o endereço MAC da estação cliente. Este último dado não é obrigatório uma vez que as tramas de desautenticação podem ser enviadas o endereço de *Broadcast* tal como referido anteriormente. Estes dados são obtidos pela captura de algumas tramas trocadas entre o ponto de acesso e um cliente autenticado.

Obtida a informação necessária executa-se o comando *essid_jack*. O resultado deste comando é apresentado na Figura 4-18. Como se verifica este comando retorna de imediato o valor do SSID dada a sua capacidade de captura e processamento de tramas IEEE 802.11.

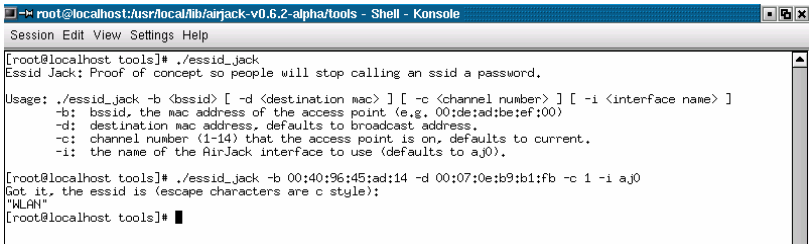


Figura 4-18 – Comando essid_jack.

Para compreender todo o processo de ataque apresentam-se de seguida as capturas efectuadas. A sequência inicial de tramas trocadas entre a estação atacante, a estação cliente e o ponto de acesso é apresentada na Figura 4-19.

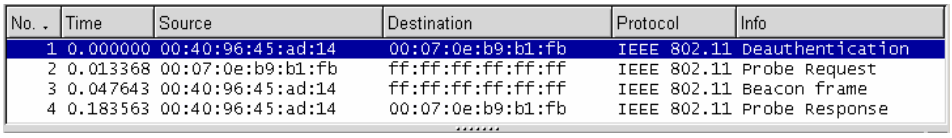


Figura 4-19 – Sequência de tramas trocadas na fase inicial do ataque.

A trama do tipo ***Deauthentication*** forjada pelo atacante é apresentada na Figura 4-20. Como se verifica pelos campos de endereço ***Destination Address*** e ***Source Address*** a trama é supostamente enviada do ponto de acesso que possui o endereço MAC 00:40:96:45:ad:14 para a estação com o endereço MAC 00:07:0e:b9:b1:fb. O campo ***Frame Body*** desta trama permite verificar o motivo da desautenticação. Neste caso tem o código 0x0002 que indica que a autenticação já não é válida.

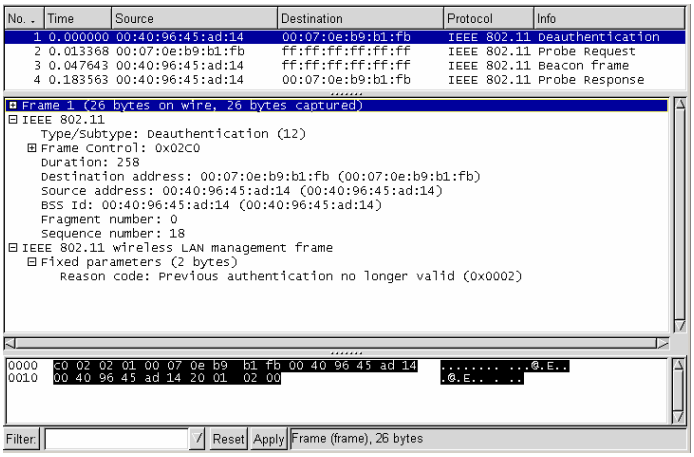


Figura 4-20 – Trama de desautenticação forjada pelo atacante.

O valor do SSID pode ser obtido pela análise das tramas de gestão do tipo ***Probe Request*** ou ***Probe Response*** enviadas pela estação vítima e pelo ponto de acesso, respectivamente. Como se verifica na Figura 4-21 e Figura 4-22 o valor do SSID é igual "WLAN".

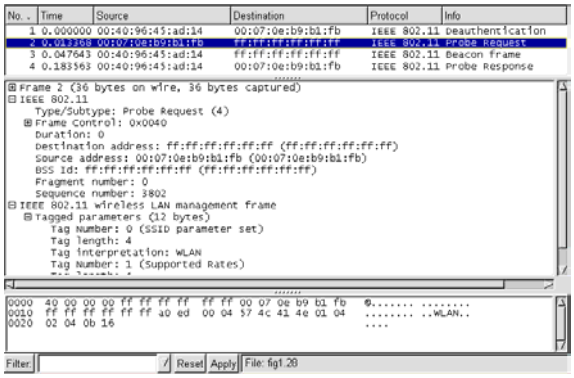


Figura 4-21 – Probe Request.

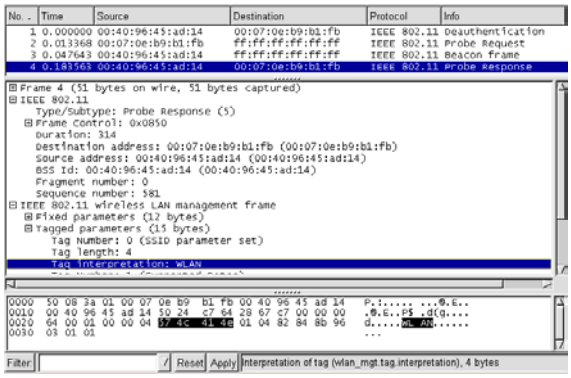


Figura 4-22 – Probe Response.

Os testes efectuados com esta aplicação vêem uma vez mais demonstrar as vulnerabilidades existentes no mecanismo de autenticação aberta. É possível concluir que mesmo a implementação de funcionalidades não definidas na norma IEEE 802.11, como é o caso da capacidade de ocultar o SSID nas tramas de gestão do tipo ***Beacon*** permite a um atacante comprometer o mecanismo de autenticação baseado em SSID.

Um outro mecanismo de autenticação não especificado na norma mas que foi implementado por um grande conjunto de fabricantes de dispositivos wireless é o mecanismo de autenticação por listas de acesso de endereços MAC. A secção seguinte descreve algumas experiências efectuadas com este mecanismo.

4.4.2 Ataque ao Mecanismo de Autenticação por Listas de Acesso de Endereços MAC

A experiência laboratorial descrita nesta secção pretende demonstrar as vulnerabilidades existentes no mecanismo de autenticação baseado em listas de acesso por endereço MAC. Para realizar esta experiência recorreu-se ao cenário da Figura 4-23.

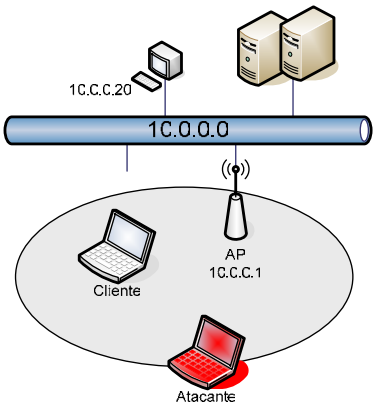


Figura 4-23 – Cenário implementado em laboratório.

As características dos dispositivos utilizados nesta experiência diferem dos descritos na secção 4.4. Utilizou-se um ponto de acesso *Cisco Aironet 1100* com endereço MAC igual a 00:0D:29:3B:73:04 e configurado com endereço IP 10.0.0.1.

A estação atacante possui um dispositivo wireless da empresa *Cisco System, Inc*, concretamente uma placa PCMCIA PCM-352, com endereço MAC 00:0B:46:65:63:2F. A estação cliente possui um dispositivo *TrueMobile 1300 Mini PCI*, comercializado pela empresa DELL com endereço MAC 00:90:4B:2F:A3:D4. O sistema operativo instalado em ambas as estações é o *Microsoft Windows XP*. O endereço IP de cada uma das estações wireless é atribuido pelo ponto de acesso por DHCP.

O ponto de acesso foi configurado com um SSID igual a "WLAN". A autenticação permitida é baseada numa lista de endereços MAC. Esta lista contém os endereços MAC das estações autorizadas a autenticar-se com o ponto de acesso. Optou-se por permitir apenas o acesso à estação cliente, tal como representado na Figura 4-24.

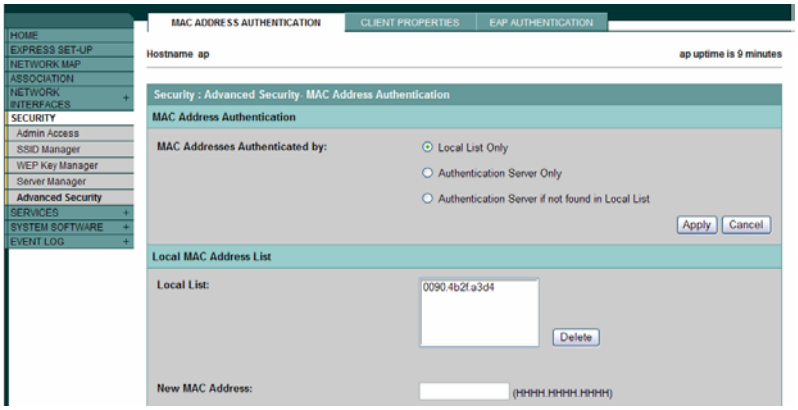


Figura 4-24 – Configuração do ponto de acesso.

Com esta configuração a estação cliente efectua um pedido de autenticação ao ponto de acesso. Este após comparar o endereço MAC da estação cliente com os endereços definidos na sua lista de endereços efectuou a autenticação da estação cliente.

Este ataque é ilustrado na Figura 4-25. Numa primeira fase a estação atacante sem conhecimento do mecanismo de autenticação baseado em lista de endereços MAC faz uma tentativa de autenticação, a qual ocorre sem sucesso. Posteriormente o atacante altera o seu endereço MAC (*MAC Spoofing*) fazendo-se passar pela estação cliente, de modo a obter sucesso na autenticação.

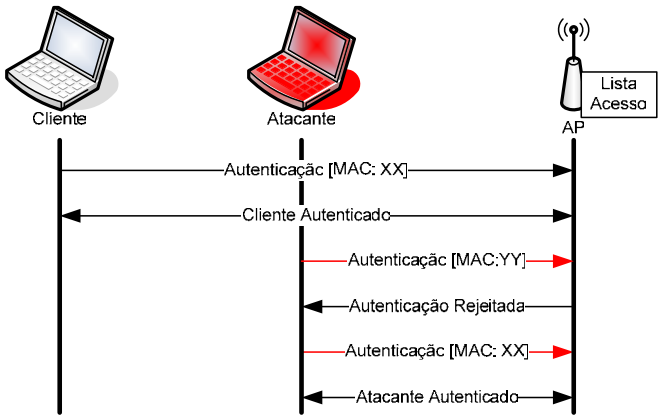


Figura 4-25 – Ataque ao mecanismo de autenticação por listas de acesso MAC.

Assumindo que o atacante não conhece qualquer informação relativa às estações legítimas nem ao ponto de acesso, a primeira acção a tomar será a captura de tramas IEEE 802.11 para obter o máximo de informação relevante, nomeadamente, o SSID configurado no ponto de acesso e o endereço MAC de um utilizador devidamente autenticado no ponto de acesso. Para obter esta informação o atacante procede à captura de tramas IEEE 802.11 trocadas entre o

ponto de acesso e um cliente autenticado. Esta informação pode ser obtida recorrendo à aplicação *Kismet*. Esta fase do ataque é normalmente conhecida por fase de reconhecimento.

Configurou-se a estação cliente com um SSID igual a "WLAN". A autenticação ocorreu com sucesso uma vez que o seu endereço MAC coincidia com o endereço definido na lista configurada previamente no ponto de acesso. Para demonstrar que a estação cliente está autenticada no ponto de acesso efectuou-se o comando *ipconfig/all* e o comando *ping* para a máquina 10.0.0.20 localizada na rede cablada. Estes comandos são ilustrados na Figura 4-26.

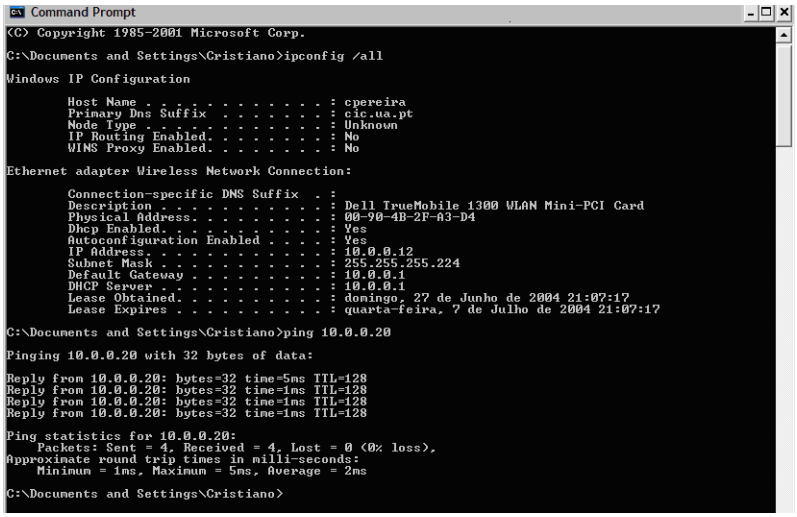


Figura 4-26 – Comandos ipconfig /all e ping 10.0.0.20 da estação cliente.

Como se verifica pela Figura 4-26 o ponto de acesso atribuiu o IP 10.0.0.12 à estação cliente. A máquina com o endereço IP 10.0.0.20 respondeu aos vários *ICMP Echo Request* enviados pela estação cliente.

Na segunda fase da experiência configurou-se a estação atacante com um SSID igual a "WLAN". A primeira tentativa de autenticação ocorreu sem sucesso, tal como se verifica através do comando *ipconfig/all* representado na Figura 4-27.

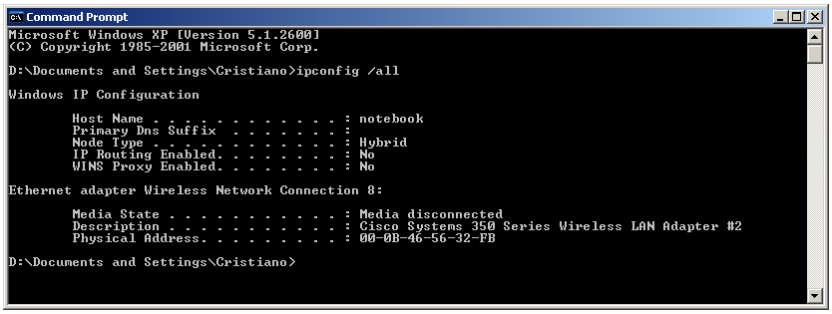


Figura 4-27 – Comando ipconfig /all na estação atacante.

Para ultrapassar o mecanismo de autenticação por listas de endereços MAC efectuou-se a configuração da placa com o mesmo endereço MAC da placa wireless de um utilizador legítimo, neste caso com o mesmo endereço MAC da placa utilizada pela estação cliente. Utilizou-se a aplicação AMAC [Link 10], para alterar o endereço MAC da placa wireless. A configuração desta aplicação é representada na Figura 4-28.

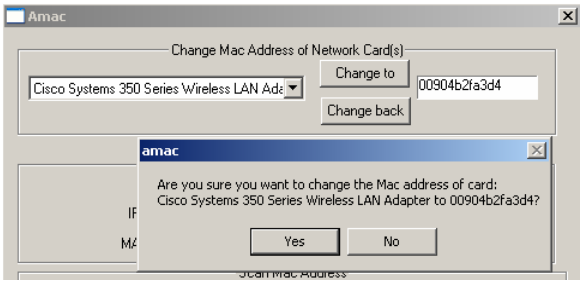


Figura 4-28 – Aplicação AMAC – alteração do endereço MAC.

Para confirmar a alteração do endereço MAC da estação atacante executou-se o comando *ipconfig/all*. Como se verifica na Figura 4-29 o endereço MAC da placa Cisco PCMCIA 350 utilizada pelo atacante passou a ser idêntico ao endereço MAC da estação cliente.

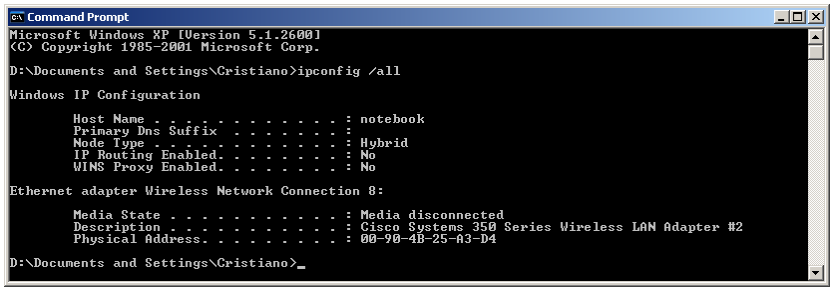


Figura 4-29 – Endereço MAC alterado na estação Atacante.

Após a alteração do endereço MAC a estação atacante pode autenticar-se no ponto de acesso ultrapassando o mecanismo de autenticação por lista de endereços MAC. Para verificar que a estação atacante se autenticou no ponto de acesso efectuou-se o comando *ipconfig/all* para verificar a atribuição de um endereço IP por parte do ponto de acesso, e em seguida o comando *ping*, tal como representado na Figura 4-30.

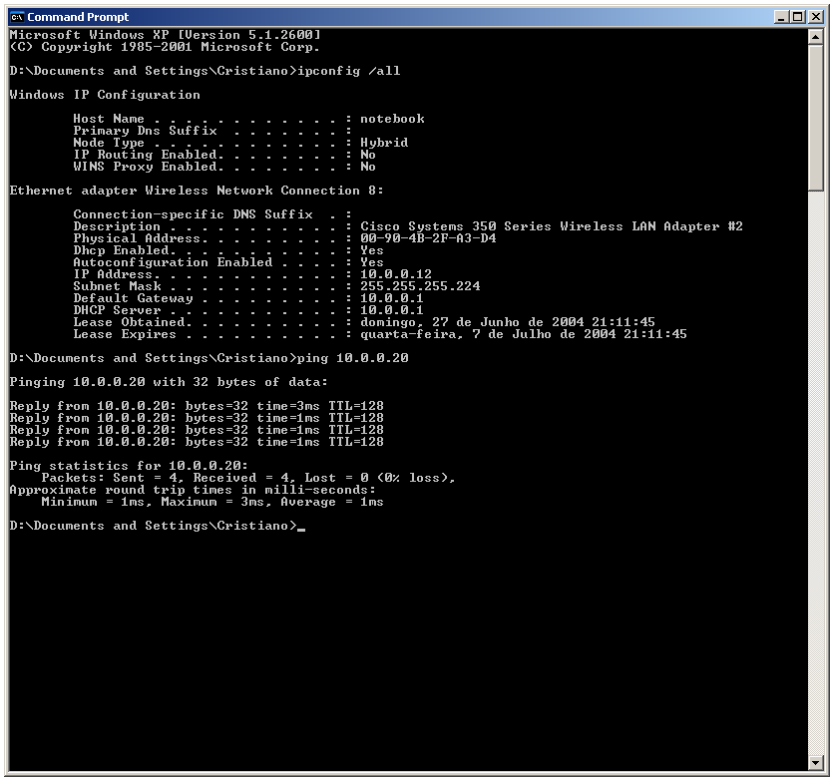


Figura 4-30 - Comandos ipconfig /all e ping 10.0.0.20 da estação atacante.

Como se verifica o endereço IP atribuído à estação Atacante é idêntico ao endereço atribuído à estação cliente, que nesta experiência é igual a 10.0.0.12. Este facto deve-se ao simples facto de o servidor de DHCP incorporado no ponto de acesso detectar um endereço MAC ao qual já havia atribuído um endereço IP. A máquina com o endereço IP 10.0.0.20 respondeu com sucesso aos vários *ICMP Echo Request* enviados pela estação atacante.

Demonstra-se assim que a alteração do endereço MAC da estação atacante para um valor de endereço presente na lista de acessos, no caso desta experiência o valor do endereço MAC da estação cliente, permite ultrapassar o mecanismo de autenticação baseado em listas de acesso de endereço MAC. De referir por último que ocorrerão problemas ao nível da comunicação na rede se ambas as estações estiverem autenticadas. Estes problemas estão relacionados com a duplicação de endereços MAC e respectiva corrupção das tabelas ARP do ponto de acesso.

4.4.3 Ataque Man-in-the Middle

A experiência descrita nesta secção pretende demonstrar a implementação de um ataque do tipo *Man in the Middle* em redes IEEE 802.11. Para realizar estas experiências recorreu-se ao cenário da Figura 4-14.

Inicialmente configurou-se o ponto de acesso para que este utilizasse o método de autenticação aberta. Para esta experiência o SSID foi configurado no ponto de acesso com o valor "WLAN", a operar no canal rádio 1. Em seguida configurou-se a estação legítima, estação cliente, com o SSID respectivo. Esta acção permitiu à estação legítima a associação e respectiva autenticação ao ponto de acesso AP.

A estação atacante recorre à aplicação *Airjack* para implementar o ataque. Este, ao contrário das experiências anteriores, exige a utilização de duas placas PCMCIA por parte da estação atacante. Uma das placas funciona como ponto de acesso forjado ao qual a estação cliente se associa e autentica. A outra placa associa-se e autentica-se no ponto de acesso legítimo. Este ataque é ilustrado na Figura 4-31.

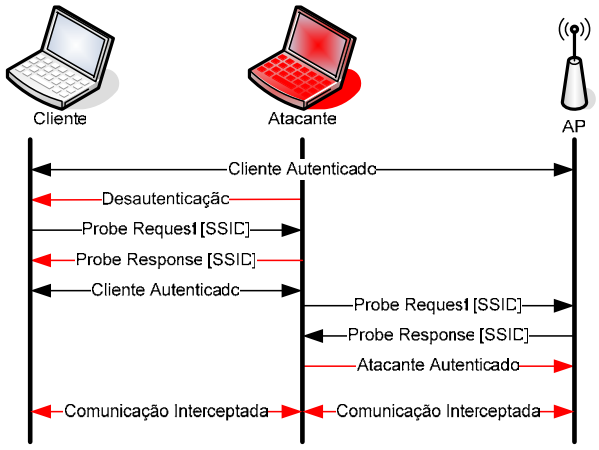


Figura 4-31 – Ataque Man in the Middle.

O ataque *Man in the Middle* baseia-se no facto das tramas de gestão IEEE 802.11 não serem autenticadas. Na fase de reconhecimento o atacante recolhe o máximo de informação relacionada com os dispositivos presentes na rede, endereços MAC da estação cliente e ponto de acesso e SSID da rede. A recolha de informação pode ser obtida recorrendo à aplicação *Kismet*.

Para proceder ao ataque propriamente dito, a estação atacante envia um conjunto de tramas de gestão do tipo ***Deauthentication*** para o endereço da estação cliente. O endereço origem

destas tramas é o endereço do ponto de acesso no qual a estação cliente está autenticada, o AP legítimo. Após a desautenticação a estação cliente vai procurar autenticar-se novamente enviando ***Probe Request*** para o meio físico. O atacante vai responder ao pedido com um ***Probe Response*** forjado, fazendo passar-se pelo ponto de acesso legítimo, a operar num canal rádio diferente, para evitar interferências. Nesta fase a estação cliente autentica-se no ponto de acesso forjado.

Finalmente a estação atacante procede à autenticação no ponto de acesso legitimo, fazendo passar-se pela estação cliente. Após esta última fase todos os dados transmitidos entre a estação cliente e o ponto de acesso legítimo são interceptados e reencaminhados pela estação atacante.

Como referido anteriormente para implementar este ataque recorreu-se à aplicação *Airjak*, e em particular ao comando *monkey_jack* que permite a implementação do ataque. A Figura 4-32 apresenta as opções de configuração deste comando.

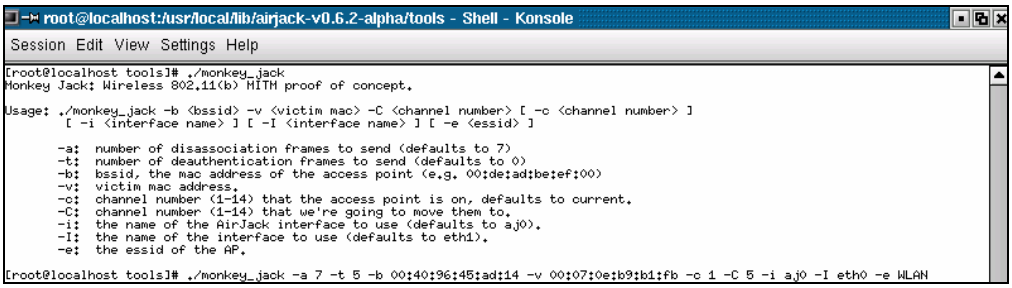


Figura 4-32 – Configuração da aplicação Monkey-Jack.

A opção –a indica o número de tramas de desassociação enviadas pela estação atacante; a opção –t o número de tramas de desautenticação enviadas (para a experiência efectuada configurou-se esta opção com o valor 7); as opções –b e –v o endereço MAC do ponto de acesso legítimo e da estação vítima respectivamente; o canal rádio em que o ponto de acesso está a operar é dado pela opção –c; o canal rádio do ponto de acesso forjado pela estação atacante é dado pela opção –C (na experiência foi atribuído o canal 5); as opções –i e –I identificam as interfaces configuradas na estação atacante; finalmente a opção –e identifica o valor do SSID configurado no ponto de acesso (neste caso o SSID possui o valor "WLAN").

A Figura 4-33 apresenta as de tramas de gestão do tipo ***Deauthentication*** enviadas para a estação cliente com origem na estação atacante. Como se verifica o endereço MAC de origem das tramas foi forjado de modo a coincidir com o endereço MAC do ponto de acesso.

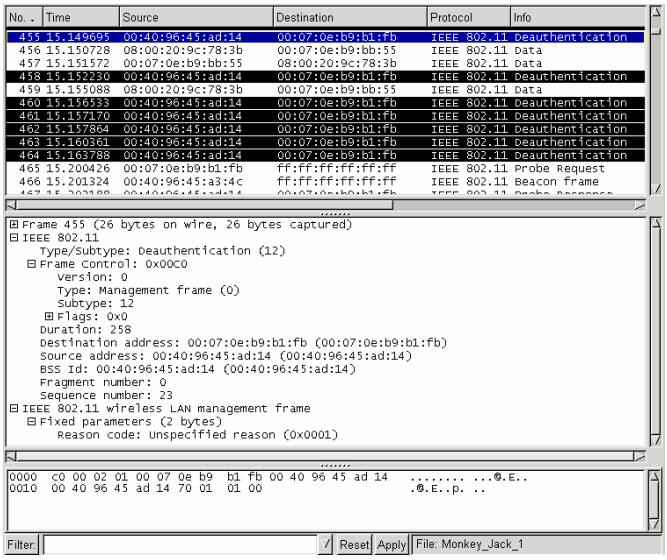


Figura 4-33 – Tramas Deauthentication enviadas pela aplicação Monkey_Jack.

Após o envio das tramas de desautenticação, a estação cliente vai tentar autenticar-se novamente num ponto de acesso disponível na área de cobertura do seu sinal RF. Para tal vai efectuar um varrimento activo, através do envio de um ***Probe Request*** em cada canal rádio. Neste processo o dispositivo wireless aguarda o tempo suficiente para receber um ***Probe Response*** antes de testar a presença de um ponto de acesso enviando um ***Probe Request*** no canal seguinte. Como se verifica pelas tramas capturadas, 465 e 474 apresentadas na Figura 4-34.

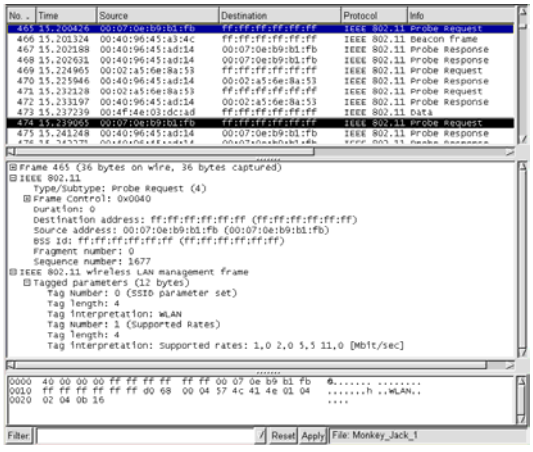


Figura 4-34 – Probe Request enviado pela estação cliente.

A estes pedidos de autenticação quer o ponto de acesso legítimo, quer o ponto de acesso forjado pela estação atacante, vão responder com tramas do tipo ***Probe Response***. A trama 475, representada na Figura 4-35 é enviada pelo ponto de acesso legítimo, o que pode ser confirmado analisando o campo com a indicação do canal RF.

A trama 476, representada na Figura 4-36, é enviada pelo ponto de acesso forjado pela estação atacante. Tal confirma-se analisando o canal RF em que este é enviado, no caso o canal 5 configurado inicialmente com a opção –C do comando *monkey_jack*.

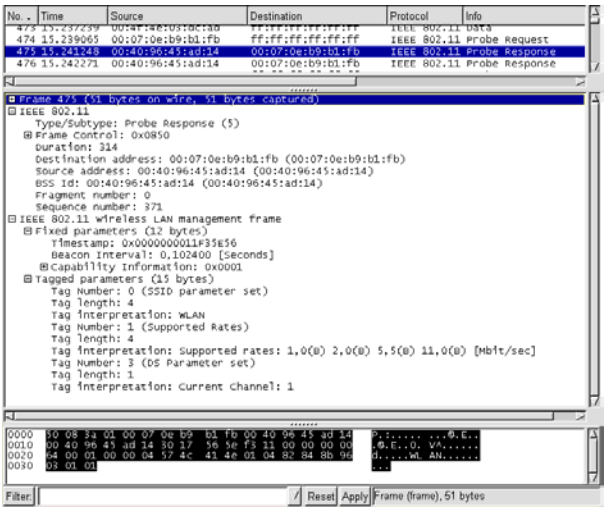


Figura 4-35 – Probe Response AP legítimo.

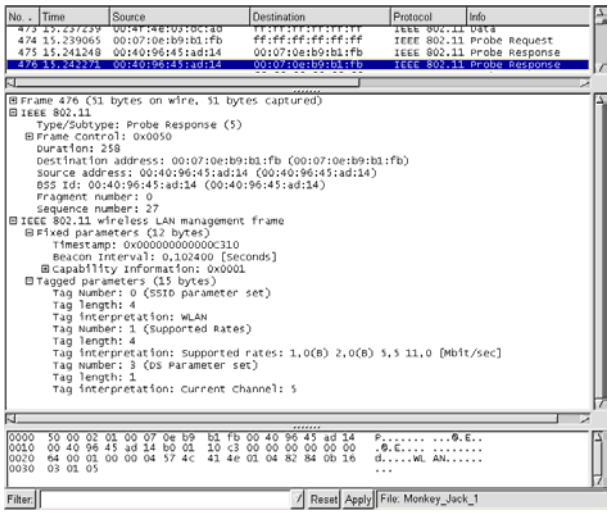


Figura 4-36 – Probe Response do AP forjado.

A trama 526 (Figura 4-37) apresenta o sucesso de autenticação da estação cliente no ponto de acesso forjado pela estação atacante.

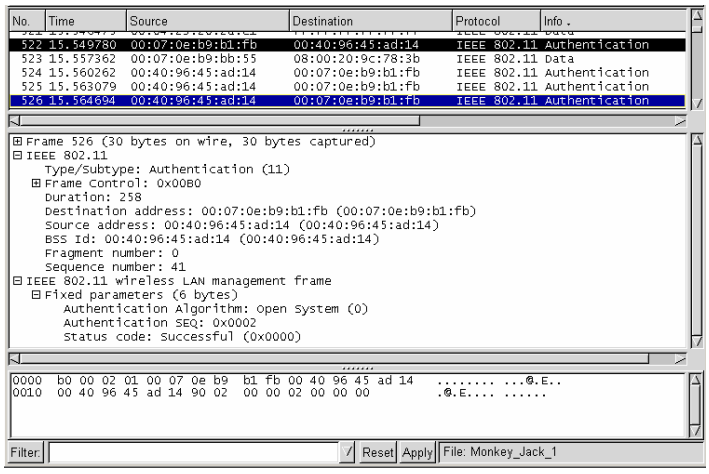


Figura 4-37 – Autenticação com sucesso da estação cliente no ponto de acesso forjado.

A última fase do ataque efectuado por esta aplicação é a autenticação da estação atacante no ponto de acesso legitimo forjando o endereço MAC da estação vítima.

As tramas de gestão ***Probe Request*** (trama 660) e ***Probe Response*** (trama 670) trocadas entre a estação atacante e o ponto de acesso legítimo, são apresentada na Figura 4-38 e Figura 4-39 respectivamente.

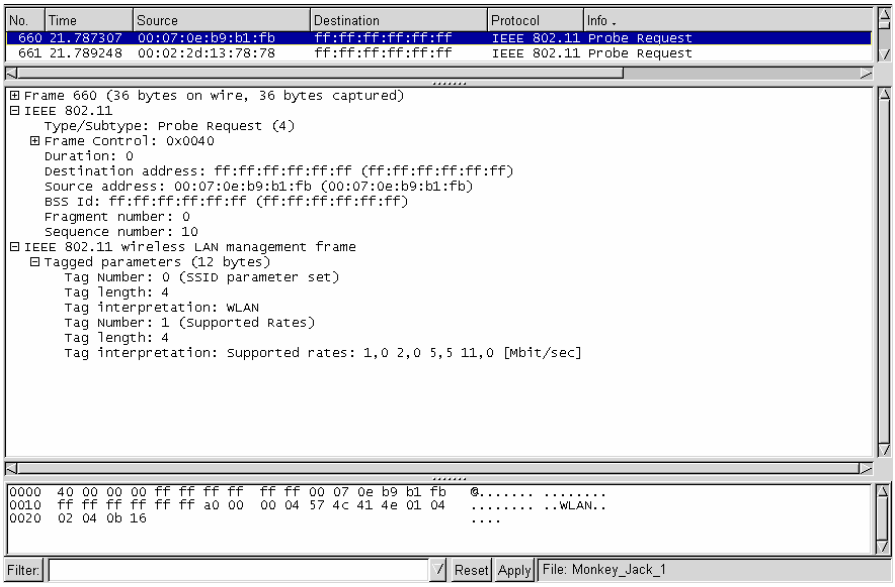


Figura 4-38 – Probe Request enviado pela estação atacante.

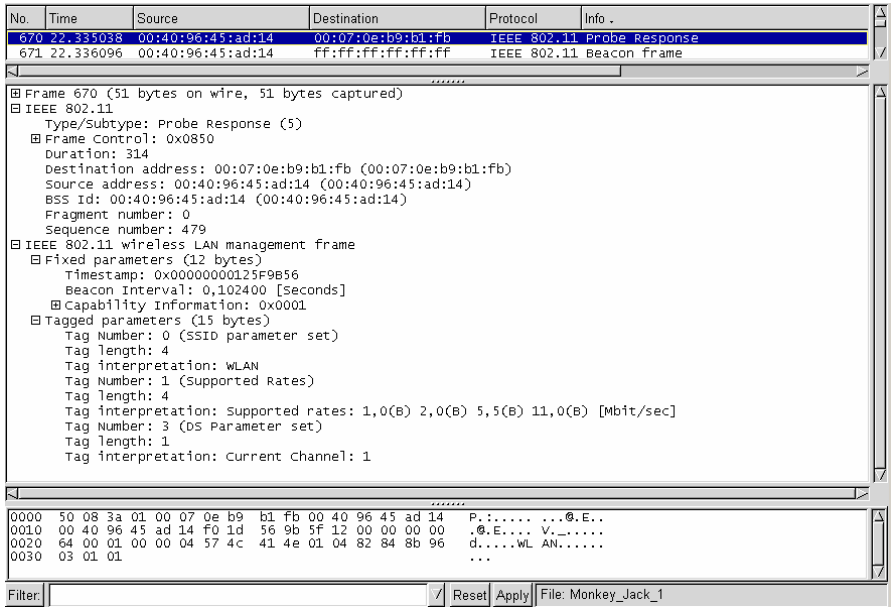


Figura 4-39 – Probe Response enviado pelo ponto de acesso AP.

A trama 698 (Figura 4-40) apresenta o sucesso de autenticação da estação atacante no ponto de acesso legítimo.

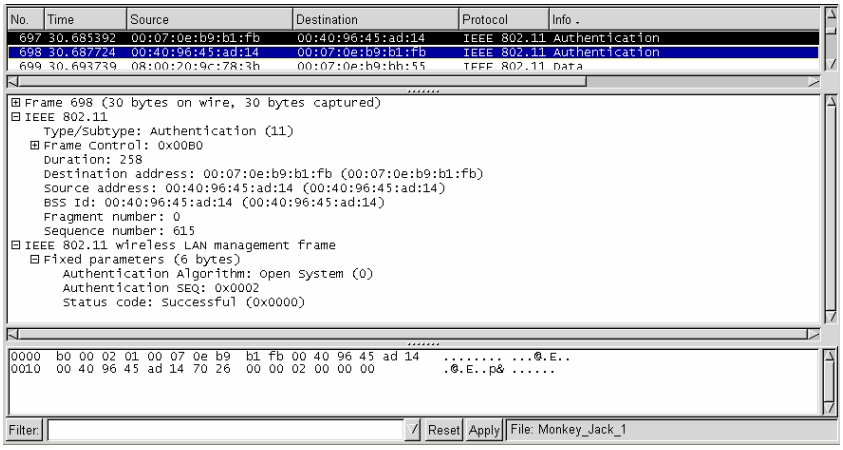


Figura 4-40 – Sucesso da autenticação da estação atacante no AP legítimo.

O comando *monkey_jack* confirma o sucesso do ataque, tal como se verifica pela Figura 4-41.

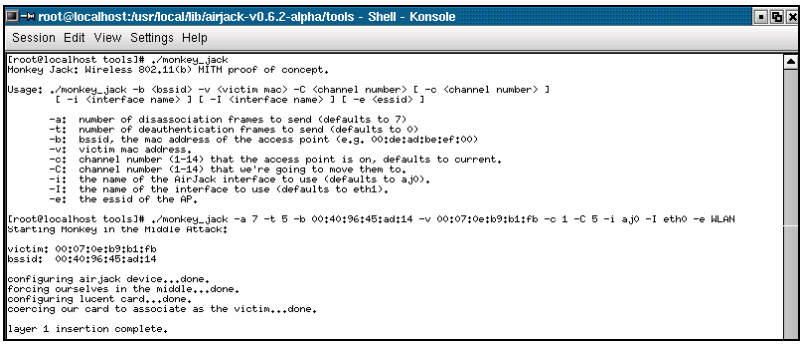


Figura 4-41 – Sucesso do ataque Man-in-the Midlle.

A implementação deste ataque permitiu verificar uma vez mais as vulnerabilidades inerentes ao mecanismo de autenticação aberta e ao facto de as tramas de gestão IEEE 802.11 não serem autenticadas.

De referir que a ferramenta de captura *Kismet* possui funcionalidades que permitem a captura de tramas IEEE 802.11 em todos os canais rádio. No ficheiro de configuração é possível definir quais os canais rádio que a aplicação deve "percorrer". No caso das experiências configurou-se a aplicação para proceder a capturas em todos os canais rádio.

4.4.4 Ataque de Negação de Serviço

Para implementar este ataque de negação de serviço recorreu-se ao comando *wlan_jack*, da aplicação *Airjack*. Basicamente esta aplicação permite o envio contínuo de tramas de gestão IEEE 802.11 do tipo ***Deauthentication*** para um determinado endereço MAC de um utilizador legítimo ou para o endereço de *broadcast*.

Uma vez mais a aplicação tira partido do facto das tramas de gestão não serem autenticadas, o que possibilita que as tramas sejam enviadas com endereço MAC de origem forjado. A fase inicial deste ataque corresponde à recolha de informação relacionada com os dispositivos wireless presentes na rede. Para efectuar esta experiência é necessário obter o endereço MAC do ponto de acesso e da estação cliente que se pretende atacar. A informação é obtida recorrendo à aplicação para captura de tramas IEEE 802.11 *Kismet*.

A captura efectuada (Figura 4-42) apresenta um conjunto de tramas de gestão do tipo ***Deauthentication*** com endereço de origem MAC forjado pela estação atacante com valor igual ao do ponto de acesso legítimo.

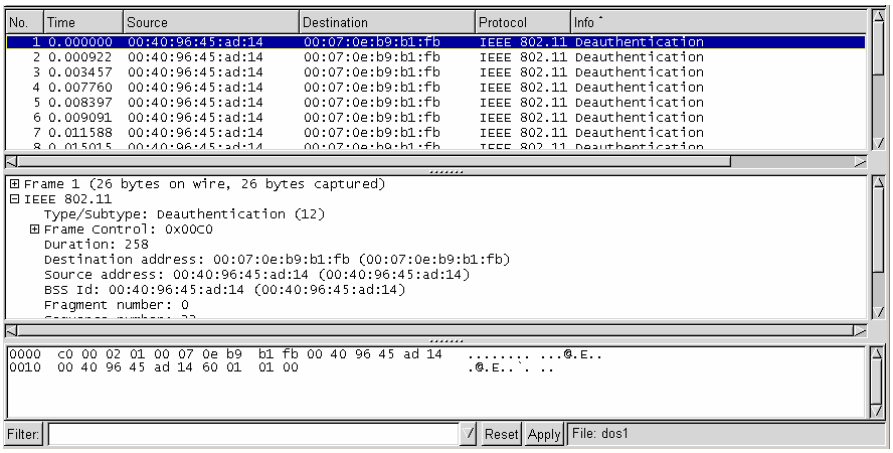


Figura 4-42 – Sequência de tramas do tipo Deauthentication.

Para esta experiência o endereço de destino das tramas definido no comando *wlan_jack* foi o endereço MAC da estação cliente, o que originou um ataque de negação de serviço apenas à estação cliente. No entanto é possível configurar o endereço de *broadcast* como endereço de destino das tramas de gestão, originando um ataque de negação de serviço a todas as estações associadas ao ponto de acesso legítimo.

4.4.5 Ataque ao Mecanismo de Confidencialidade WEP

Esta secção descreve a experiência prática do ataque ao mecanismo de confidencialidade WEP da norma IEEE 802.11. A ferramenta utilizada nesta experiência, *Airsnort*, explora as vulnerabilidades descritas teoricamente na secção 4.3.6. O cenário implementado em laboratório é apresentado na Figura 4-43. Para a realização desta experiência recorreu-se a um ponto de acesso da marca Intel 2011B com endereço MAC 00:02:B3:AE:F9:64, e a duas estações wireless associadas e autenticadas no ponto de acesso com o objectivo de gerarem tráfego suficiente na rede não-cablada. O cliente 1 utiliza uma placa PCMCIA Cisco - PCM-352; o cliente 2 uma placa PCMCIA Compaq - WL110.

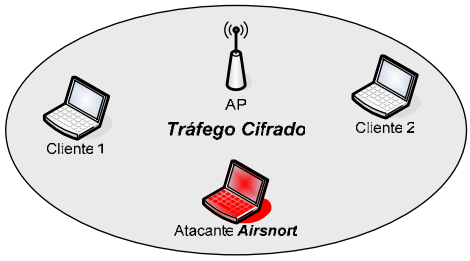


Figura 4-43 – Cenário laboratório.

A estação designada no cenário como atacante, é responsável pela captura de todo o tráfego que circula na rede. Esta estação tem instalado o sistema operativo Red Hat Linux 7.3 com versão de Kernel 2.4.16. As ferramentas utilizadas para a captura das tramas IEEE 802.11 e respectiva análise e processamento para obtenção do valor da chave WEP foram o *Kismet* e *Airsnort* respectivamente.

Segundo informação disponibilizada em [Link1], a aplicação *Airsnort* opera em modo passivo, e possui a capacidade para capturar o tráfego transmitido e efectuar os cálculos necessários para obter a chave WEP. No entanto a tentativa para colocar o *Airsnort* a capturar em tempo real as tramas transmitidas pelas estações clientes não ocorreu com sucesso. Para solucionar este problema optou-se pela utilização da aplicação *Kismet* para efectuar as capturas e armazenamento das tramas IEEE 802.11 em ficheiros, que serão posteriormente lidos pelo *Airsnort*. O *Kismet* gera dois tipos de ficheiros, um ficheiro onde são armazenadas todas as tramas capturadas e um segundo onde são armazenadas as tramas classificadas como de interesse para serem analisadas por uma ferramenta de obtenção da chave WEP, como o *Airsnort*. O *Kismet* classifica uma trama como trama de interesse sempre que esta possui um vector de inicialização fraco.

Note-se que o critério de classificação de IV fraco do *Kismet* é ligeiramente diferente do critério de classificação utilizado pelo *Airsnort*. Esta abordagem possui como principal inconveniente a dificuldade de medir o tempo mínimo necessário à obtenção da chave WEP, assim como a quantidade mínima de tramas que é necessário capturar.

Tendo em conta as limitações referidas nos parágrafos anteriores, a leitura dos resultados em termos de tempo de captura e número de tramas necessárias à obtenção da chave WEP devem ser entendidos apenas como valores máximos indicativos.

Esta experiência foi efectuada em duas fases distintas. A primeira fase pretendia apenas a captura de tramas cifradas. Na segunda fase da experiência procedeu-se à obtenção do valor da chave secreta WEP configurada no ponto de acesso com tamanho de 40 bits.

A chave secreta configurada no ponto de acesso e nos clientes para esta primeira experiência tinha o valor "chave" (valor ascii). O valor do SSID configurado foi "surete". Após a configuração de todos os dispositivos procedeu-se ao início da experiência. O tráfego gerado entre as estações cliente foi forçado com o comando *ping –f*. Este comando inunda a rede com tráfego ICMP em modo contínuo. Para esta fase da experiência foram capturadas 14 326 985 tramas IEEE 802.11 em aproximadamente 5 horas.

Em seguida foi utilizado o *Airsnort* para analisar os dados capturados, armazenados num ficheiro, e proceder à obtenção da chave WEP configurada. A Figura 4-44 apresenta os resultados obtidos.

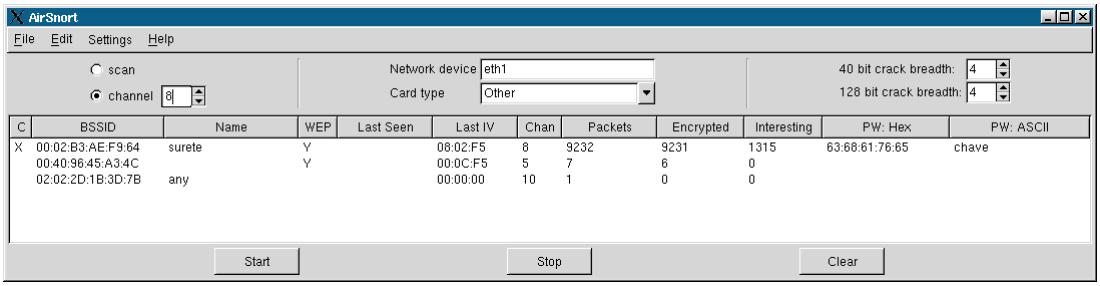


Figura 4-44 – Airsnort - chave WEP de 40 bits.

Dos resultados fornecidos pelo *Airsnort* sobressai de imediato o valor da chave WEP que se pretendia obter e que está representado na coluna PW:ASCII. Podemos observar outras informações que no essencial estão relacionados com o princípio de funcionamento explorado pelo *Airsnort* para decifrar a chave WEP.

A coluna "Interesting", que na Figura 4-44 possui o valor 1315, apresenta a quantidade de vectores de inicialização considerados fracos pelo *Airsnort*. O *Kismet* por sua vez classifica 9231 tramas com IVs fracos.

As colunas "Packets" e "Encrypted" deveriam apresentar os valores referentes à quantidade de tramas capturadas. No entanto os valores apresentados nesta experiência referem-se ao número de tramas armazenadas no ficheiro de tramas interessantes gerado pelo *Kismet*, e que foi submetido à analise do *Airsnort*. Se tivesse sido utilizado o *Airsnort* para efectuar as capturas das tramas estes valores seriam muito superiores aproximando-se do valor 14 326 985, referido anteriormente.

Para demonstrar que este tipo de ataque continua a ser possível com as chaves de tamanho maior, procedeu-se a uma nova experiência onde se pretendeu determinar o valor de uma chave WEP de 104 bits. Para tal configurou-se uma chave WEP no ponto de acesso e nos clientes com o valor ascii "ABCDEFGHIPQRS".

Procedeu-se como na experiência anterior, isto é, recorreu-se ao *Kismet* para efectuar a captura e armazenamento das tramas IEEE 802.11 e ao *Airsnort* para obter o valor da chave WEP. O tráfego gerado entre as estações clientes foi igualmente forçado com o comando *ping –f*. A quantidade de tramas capturadas nesta segunda fase foi de 15 328 037.

Observando os resultados obtidos pelo *Airsnort* (Figura 4-45) é possível verificar que foram necessárias 1858 tramas (classificação *Airsnort*) cifradas com vectores de inicialização fracos para que fosse obtido o valor correcto da chave WEP inicialmente configurada. Nesta experiência o *Kismet* classificou 1866 tramas com IVs fracos.

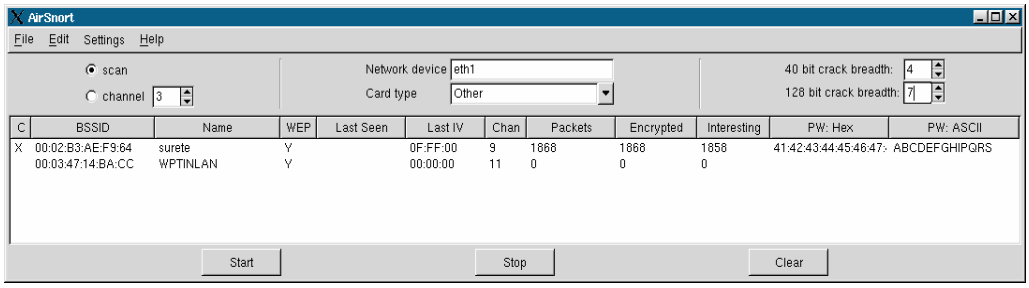


Figura 4-45 – Airsnort – chave WEP de 104 bits.

Como referido na secção 4.3.6 existe 95 % de probabilidade de um vector de inicialização "fraco" não fornecer qualquer informação válida relacionada com um byte da chave. Em [Link1] afirma-se que o número de tramas necessárias para revelar um byte da chave pode ser

pequeno, mas pode também ser muito elevado. Afirma-se ainda que em alguns casos o *Airsnort* determinou rapidamente informação de 12 bytes de uma chave de 13 bytes e levou muito tempo a determinar o valor do décimo terceiro byte. Estas afirmações permitem concluir da dificuldade em determinar valores de tempo ou mesmo quantidades de tráfego necessárias à obtenção da chave WEP. No entanto em resposta à questão de, quanto tempo é necessário para determinar uma chave WEP com o *Airsnort*, os responsáveis pela ferramenta afirmam que dos cerca de 16 milhões de vectores de inicialização que podem ser gerados por um dispositivo wireless, cerca de 9 mil são fracos (para chaves de 104 bits), e que a grande maioria das chaves podem ser adivinhadas com cerca de 2000 tramas cifradas com IVs fracos, outras podem ser adivinhadas com apenas 1200-1500 e outras bastante mais 3500-4000. Afirmam ainda que à medida que uma rede se aproxima da saturação de tráfego, o tempo de captura necessário aproxima-se das 24 horas.

Estas afirmações vêm em certa medida confirmar os resultados verificados nas experiências práticas realizadas. A quantidade de tramas "interessantes" enquadra-se nos valores de referência indicados em [Link1]. O relativo pouco tempo necessário à obtenção das chaves WEP, na ordem das 5 horas fica-se a dever à saturação da rede com tráfego gerado propositadamente. Testes realizados numa rede real com pouco tráfego aumentariam obrigatoriamente estes tempos.

Interessa finalmente referir que os testes realizados nesta secção recorreram a um ponto de acesso sem qualquer actualização de *firmware* à data da realização destas experiências. Experiências realizadas com pontos de acesso com *firmware* actualizado não obtiveram sucesso na obtenção da chave em tempo útil. Este facto fica a dever-se às actualizações introduzidas pelos principais fabricantes de dispositivos wireless, para reduzir o número de vectores de inicialização fracos gerados, dificultando desta forma a obtenção das chaves WEP por parte do *Airsnort*.

As experiências laboratoriais apresentadas nas secções anteriores vêm provar que o protocolo de segurança desenvolvido pelo IEEE 802.11 não garante segurança em nenhum dos serviços em que se propunha fazê-lo, nomeadamente, autenticação, integridade e confidencialidade dos dados

5 Mecanismos de Segurança Alternativos

Este capítulo apresenta um conjunto de mecanismos e protocolos adoptados de modo a solucionar as vulnerabilidades das redes não-cabladas IEEE 802.11. Alguns dos mecanismos apresentados são propostas em fase de desenvolvimento pelo que a sua implementação está ainda sujeita a rectificações futuras por parte dos organismos responsáveis pela sua aprovação, nomeadamente o IEEE e o IETF.

Alguns mecanismos apresentados pretendem ser uma solução transitória até que o novo protocolo de segurança, o IEEE 802.11i seja normalizado e possa ser adoptado em larga escala por todos os sistemas de rede não-cabladas.

A secção 5.1 aborda a utilização de Redes Privadas Virtuais em cenários de redes não-cabladas. Na secção 5.2 é apresentada a relação entre os mecanismos de segurança WPA (*Wi-Fi Protected Access*), redes RSN (*Robust Security Network*) e o protocolo IEEE 802.11i. Nesta secção é ainda apresentado o modelo de segurança em camadas adoptado para as redes wireless. Os mecanismos de controlo de acesso implementados em redes não-cabladas são apresentados na secção 5.3. Os mecanismos de autenticação e confidencialidade são apresentados nas secções 5.4 e 5.5 respectivamente.

5.1 Redes Privadas Virtuais

As soluções baseadas em redes privadas virtuais, VPN (*Virtual Private Networks*), têm vindo a aumentar nos últimos anos. O aperfeiçoamento e uniformização das suas características de segurança, bem como a redução de custos de implementação, incrementaram a sua utilização.

Uma VPN combina os serviços de autenticação, confidencialidade e *tunneling* (transferência de dados entre dois pontos, utilizando uma rede intermediária não segura e mantendo a privacidade e o controlo dos dados originais) para criar um canal de comunicação seguro entre um utilizador e uma rede corporativa, ou entre duas redes.

Nos últimos anos têm surgido um conjunto de tecnologias de *tunneling*. Destacam-se entre as mais recentes as seguintes: (i) PPTP (*Point-to-Point Tunneling Protocol*); (ii) L2TP (*Layer 2 Tunneling Protocol*); (iii) IPSec (*IP Security Protocol*).

Para estabelecer um túnel entre duas entidades, estação cliente e servidor VPN, ambas devem usar o mesmo protocolo de *tunneling*. As tecnologias de *tunneling* podem ser baseadas em protocolos de nível protocolar 2 ou protocolos de nível de rede. O PPTP e o L2TP são protocolos de *tunneling* de nível 2, ambos encapsulam o *payload* em tramas PPP (*Point-to-Point Protocol*). O IPSec é um protocolo de nível 3 que encapsula pacotes IP com um cabeçalho IP adicional.

Na secção 5.1.1 são apresentados os protocolos de *tunneling* baseados no protocolo PPP. Na secção 5.1.2 aborda-se o protocolo IPSec. Finalmente na secção 5.1.3 é apresentado um cenário de implementação de uma solução de acesso wireless baseada em redes privadas virtuais.

5.1.1 Soluções Baseadas em PPP

Uma vez que o PPTP e o L2TP são dois protocolos baseados nas características originais do PPP faz-se inicialmente uma análise a este protocolo.

O protocolo PPP definido em [RFC1661] é utilizado para fornecer um acesso remoto seguro a uma rede corporativa estabelecendo uma ligação autenticada e confidencial de nível 2 entre dois pontos, por exemplo entre uma estação remota e uma rede corporativa. Em redes IP, o PPP encapsula os pacotes IP numa trama PPP, e em seguida transmite os pacotes encapsulados através de uma ligação ponto-a-ponto.

O [RFC1661] define as várias fases do processo de configuração, manutenção e terminação de uma ligação PPP, tal como descrito na Figura 5-1.

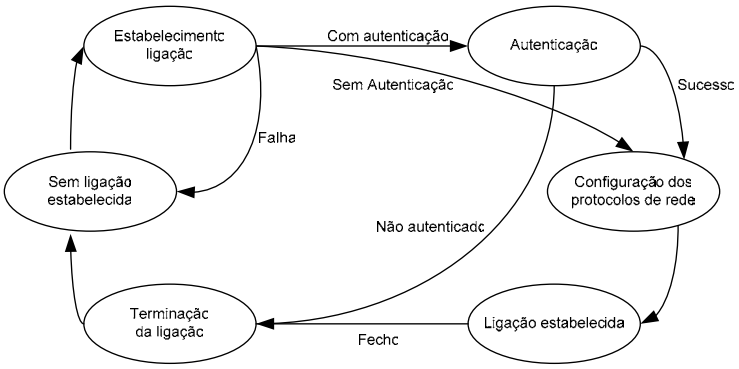


Figura 5-1 – Diagrama de fases do PPP.

Na fase de estabelecimento da ligação o PPP utiliza o protocolo LCP (*Link Control Protocol*) para estabelecer, manter e terminar a ligação ponto-a-ponto. Nesta fase, são definidas as opções básicas de comunicação. Por exemplo, são escolhidos os protocolos de autenticação a utilizar na fase seguinte.

A fase de autenticação é responsável pela autenticação do cliente perante o servidor remoto, de acordo com o método negociado na fase anterior. O nível de segurança varia de acordo com o método de autenticação escolhido e pode variar desde um método de autenticação por senha em texto plano, como é o caso do protocolo PAP (*Password Authentication Protocol*), até um método de autenticação do tipo desafio – resposta, como por exemplo o protocolo CHAP (*Challenge-Handshake Authentication Protocol*) ou o MS-CHAP-v2 (*Microsoft- Challenge-Handshake Authentication Protocol-version2*). Durante esta fase o servidor de VPN recebe os dados de autenticação do cliente e procede à sua validação através de uma base de dados interna com os dados do utilizador, ou pode recorrer a um servidor RADIUS para validar os dados do utilizador. Esta fase é opcional não é obrigatório a autenticação do cliente para que uma ligação seja estabelecida.

A fase seguinte é responsável pela configuração dos protocolos de rede. Após a conclusão das fases anteriores, cada protocolo de rede deve ser configurado separadamente pelo respectivo NCP (*Network Control Protocol*). Por exemplo, no caso do IP o protocolo IPCP (*Internet Protocol Control Protocol*) é utilizado para atribuir um endereço dinâmico ao cliente PPP. No caso particular da implementação PPP da Microsoft, o protocolo de controlo de compressão CCP (*Compression Control Protocol*) é usado para negociar a compressão de dados (usando MPPC) e para a cifra dos dados, recorrendo ao MPPE (*Microsoft Point-To-Point Encryption Protocol*) [60].

Após o sucesso das fases anteriores a ligação está estabelecida e o PPP começa a transmitir dados através do túnel estabelecido. Cada pacote de dados transmitido é encapsulado num cabeçalho PPP que é posteriormente desencapsulado pelo sistema receptor.

A fase final é a terminação da ligação. O PPP pode terminar uma ligação a qualquer momento. Isto pode acontecer após perca do sinal de linha, falha na autenticação, má qualidade da ligação, expiração de um temporizador de inactividade ou intenção do utilizador.

O PPP foi originalmente desenvolvido para utilização em ligações *dial-up*. A utilização por si só deste protocolo não é suficiente para proporcionar uma solução de redes privadas virtuais segura. Para ultrapassar esta limitação foram criados os protocolos PPTP e L2TP, ambos baseados no PPP.

Point to Point Tunneling Protocol (PPTP)

O protocolo PPTP [RFC2637] foi desenvolvido em 1996 dois anos antes dos protocolos IPSec e L2TP. Os principais objectivos a alcançar com o desenvolvimento deste protocolo eram a simplicidade e o suporte multi-protocolo.

Tal como é implementado actualmente, o PPTP encapsula pacotes PPP utilizando uma versão modificada do protocolo GRE (*Generic Routing Encapsulation*) [RFC2784], o qual fornece a flexibilidade necessária para que o PPTP suporte outros protocolos além de IP, tais como IPX (*Internet Packet Exchange*) e NetBEUI (*NetBios Extended User Interface*).

A ideia base deste protocolo é o transporte de tramas PPP em pacotes IP. As tramas PPP são encapsulados em pacotes GRE, e estes são inseridos em pacotes IP, tal como representado na Figura 5-2.

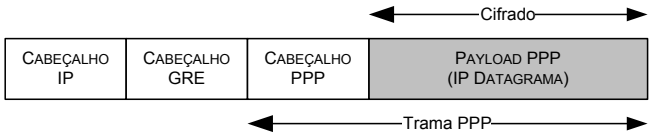


Figura 5-2 – Encapsulamento PPTP.

Dada a dependência relativamente ao PPP, os mecanismos de autenticação proporcionados inicialmente foram os mesmos que o PPP, protocolo PAP e CHAP. Posteriormente foram definidos outros protocolos de autenticação baseados em EAP (*Extensible Authentication Protocol*). A implementação PPTP da Microsoft implementa o protocolo de autenticação MS-CHAP-v2, este também é suportado por outras implementações.

Após o processo de autenticação é possível a negociação de mecanismos de cifra de dados. São suportados três protocolos: o protocolo (*Encryption Control Protocol* - ECP) [RFC1968], DESE (*PPP DES Encryption Protocol*) [RFC1969] e 3 DESE (*PPP Triple-DES Encryption Protocol*) [RFC2420]. A implementação PPTP da Microsoft cifra os dados através do protocolo MPPE.

Foram detectadas algumas falhas de segurança no PPTPv1 e PPTPv2 [14] o que levou os vários grupos de trabalho do IETF a não adoptar o PPTP como norma. No entanto este vem sendo utilizado em cenários de redes privadas virtuais devido em grande parte ao facto de ser um dos protocolos de *tunneling* suportados pela Microsoft.

Layer 2 Tunneling Protocol (L2TP)

O protocolo L2TP [RFC2661], é uma combinação do protocolo PPTP e da tecnologia proposta pela empresa *Cisco Systems*, L2F (*Layer 2 Forwarding*) [RFC2341]. O L2TP pretende reunir as melhores características presentes nos dois protocolos, ao mesmo tempo que tenta fornecer interoperabilidade entre diversos dispositivos.

O L2TP usa UDP para transmitir tramas PPP encapsuladas em pacotes L2TP. Este protocolo de *tunneling* pode utilizar os mesmos protocolos de autenticação do PPTP incluindo PAP, CHAP e MS-CHAP-V2. Como mecanismo de cifra [61] recomenda a utilização do protocolo IPSec. A implementação L2TP da Microsoft não recorre ao protocolo MPPE mas sim ao protocolo IPSec, de acordo com o [RFC3193]. A Figura 5-3 apresenta uma trama PPP encapsulada num pacote L2TP.

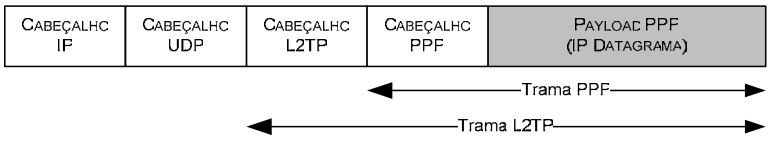


Figura 5-3 – Encapsulamento L2TP.

Uma das principais diferenças entre o L2TP e o PPTP é que o L2TP, ao contrário do PPTP, pode ser usado sobre redes não baseadas em IP.

5.1.2 IPSec

A arquitectura base do protocolo IPSec é definida em [RFC2401]. O IPSec é um protocolo largamente utilizado em redes privadas virtuais que implementa canais seguros sobre redes não seguras. Este protocolo fornece serviços de segurança ao nível da camada de rede.

Para fornecer estes serviços de segurança, o IPSec estabelece associações de segurança SA (*Security Association*). Uma SA é um canal lógico unidirecional através do qual se descrevem e definem os mecanismos de segurança a utilizar numa comunicação entre duas entidades. Cada entidade gere um conjunto de SAs, no mínimo uma por cada comunicação. As associações de segurança definem o modo de protecção do tráfego, que tráfego proteger e com quem realizar a protecção do tráfego.

Outro componente essencial na arquitectura IPSec, é a base de dados das políticas de segurança SPD (*Security Policy Database*). A SPD opera em conjunto com a SAD (*Security Association Database*) para processar os pacotes. Se não existe nenhuma SA na base de dados SAD que respeite a política definida na SPD então será necessário criar uma nova SA.

De modo a negociar e estabelecer uma SA entre dois sistemas é necessário um sistema de gestão de chaves partilhadas eficaz. O IPSec recorre aos procedimentos de criação e troca de chaves definidos no protocolo IKE (*Internet Key Exchange*) [RFC2409]. O IKE opera sobre uma estrutura base que define os mecanismos de gestão das associações de segurança, sendo esta estrutura definida pelo protocolo ISAKMP (*Internet Association and Key Management Protocol*) [RFC2408]).

Existem dois protocolos chave, aos quais o IPSec pode recorrer: o protocolo AH (*IP Authentication Header*) [RFC2402] e o ESP (*IP Encapsulating Security Payload*) [RFC2406]. O primeiro fornece integridade de dados, autenticação da origem dos dados e protecção contra ataques de repetição. O protocolo ESP fornece os mesmos serviços de segurança do AH e adicionalmente implementa confidencialidade.

O IPSec pode operar em dois modos de funcionamento:

- Modo de Túnel – concebido para ser utilizado entre dois sistemas intermédios (como por exemplo firewalls ou routers). No entanto nada impede a sua utilização entre dois sistemas finais. Neste modo apenas os dois extremos do túnel têm que suportar IPSec.

- Modo de Transporte – só pode ser utilizado entre dois sistemas finais, normalmente um cliente e um servidor. Os sistemas finais têm obrigatoriamente que suportar IPSec. Os sistemas intermédios não necessitam desse requisito, pois limitam-se a encaminhar os pacotes.

As secções seguintes apresentam uma breve descrição dos protocolos AH e ESP respectivamente.

Authentication Header (AH)

Este protocolo é responsável por fornecer o serviço de autenticação e integridade dos pacotes IP. No entanto não fornece confidencialidade dos conteúdos. Como referido anteriormente o IPSec pode operar em dois modos de funcionamento, modo transporte e modo de túnel. O formato de um pacote IP com protecção AH é distinto de acordo com o modo de funcionamento. A Figura 5-4 e Figura 5-5 ilustram respectivamente um pacote IP com protecção AH em modo de transporte e em modo de túnel.

Figura 5-4 – Pacote IP com protecção AH em modo de transporte.

Como se verifica da Figura 5-4 no modo de transporte o AH protege os protocolos transportados pelo IP. Para garantir autenticação e integridade o AH adiciona entre o cabeçalho IP e o cabeçalho do protocolo da camada superior (UDP, TCP, ICMP), um cabeçalho IPSec AH.

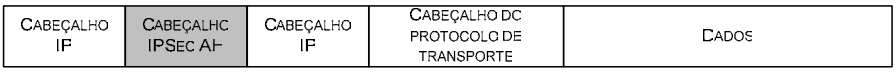


Figura 5-5 – Pacote IP com protecção AH em modo de túnel.

No modo de túnel (Figura 5-5) todo o pacote IP é encapsulado num novo cabeçalho IP. O cabeçalho IP "interior" contém o endereço de destino final dos dados. O cabeçalho exterior pode conter endereços IP intermédios, que podem ser alterados de acordo com o encaminhamento do pacote até ao seu destino.

Encapsulating Security Payload (ESP)

Este é o protocolo responsável pela cifra dos dados transportados num pacote IP. As suas funcionalidades permitem a implementação dos serviços de autenticação, confidencialidade e

integridade. O serviço de autenticação é de implementação opcional. Tal como o AH, também o ESP suporta os modos de transporte e túnel. De acordo com o [RFC2406], nas implementações do ESP os algoritmos de cifra obrigatórios são o DES e o 3DES. No entanto podem ser utilizados outros algoritmos, por exemplo, RC5, IDEA ou Blowfish.

A Figura 5-6 e Figura 5-7 ilustram respectivamente um pacote IP protegido pelo ESP em modo de transporte e em modo de túnel.

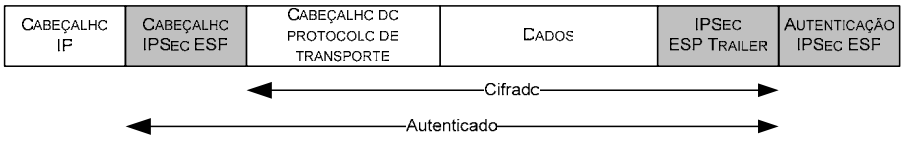


Figura 5-6 – Pacote IP com protecção ESP em modo de transporte.

Como se verifica pela Figura 5-6 os dados que se situam entre o cabeçalho ESP e o campo de autenticação ESP são cifrados, garantindo assim confidencialidade e integridade dos dados a transportar. O campo de autenticação ESP é opcional.

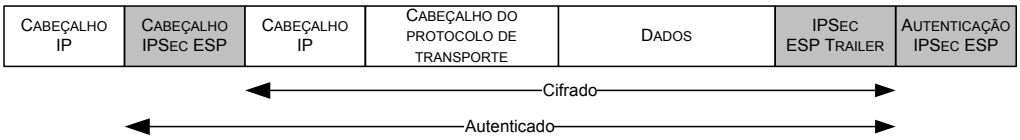


Figura 5-7 – Pacote IP com protecção ESP em modo túnel.

O modo túnel do ESP é utilizado para cifrar e opcionalmente autenticar todo o pacote IP. Como se verifica pela Figura 5-7, o cabeçalho ESP é adicionado ao pacote IP original sendo este cifrado juntamente com o *trailer* ESP.

Sempre que se pretende uma protecção excepcional, podem-se juntar os serviços prestados por ambos os protocolos AH e ESP. Nestes casos o pacote ESP é encapsulado dentro de um pacote AH e este por sua vez é encapsulado num pacote IP, como se verifica pela Figura 5-8.

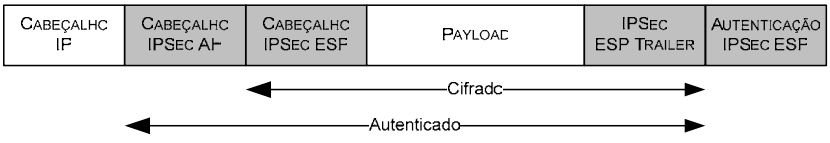


Figura 5-8 – Pacote IP com protecção simultânea do AH e ESP.

Para concluir a análise efectuada aos protocolos de *tunneling* mais divulgados, importa referir uma implementação do protocolo L2TP com cifra IPSec, baseada no [RFC3193]. Esta

consiste essencialmente no encapsulamento de pacotes L2TP num cabeçalho ESP de um pacote IP. A Figura 5-9 representa um pacote L2TP protegido pelo protocolo ESP.

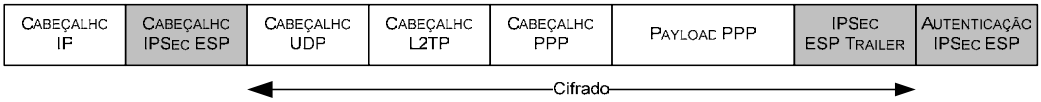


Figura 5-9 – Pacote L2TP cifrado com ESP.

Combinando as propriedade do L2TP com as do IPSec, são conseguidas as funcionalidades do PPTP e adicionados um conjunto de serviços de segurança e controlo fornecidos pelo IPSec.

5.1.3 Cenário de acesso VPN a uma rede não-cablada

Nesta secção apresenta-se um cenário de acesso VPN aos recursos de uma rede não-cablada, como por exemplo uma rede corporativa ou a rede interna de uma universidade. A solução representada na Figura 5-10 adiciona um servidor de VPN com funcionalidade de NAS (*Network Access Server*) entre a rede corporativa e a rede não-cablada (rede insegura).

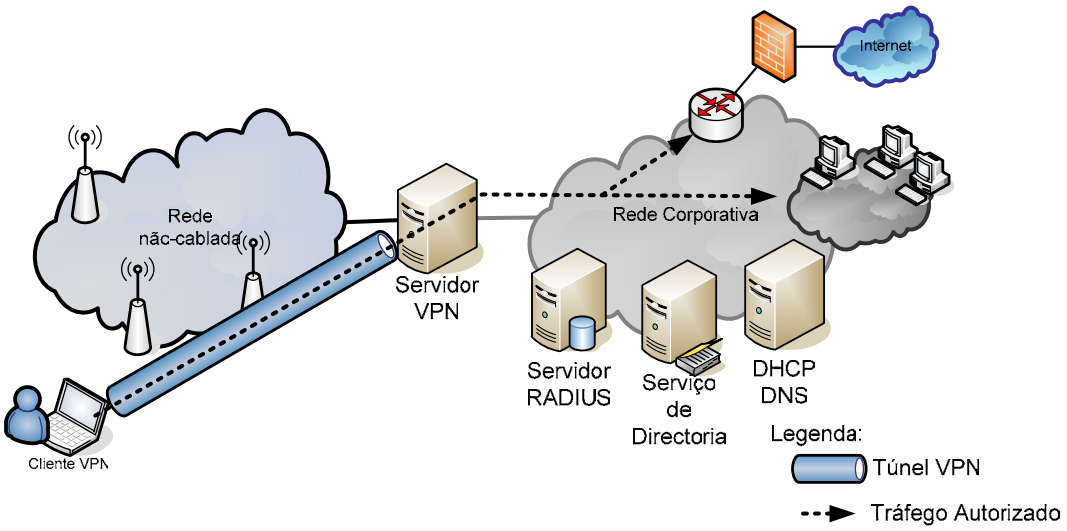


Figura 5-10 – Cenário de acesso VPN a uma rede não-cablada.

Considera-se para esta solução que a rede não-cablada é anunciada através de um SSID público e é configurada com autenticação aberta. O utilizador tem configurado um cliente VPN, que permitirá estabelecer um túnel entre as duas entidades (cliente e servidor). As

configurações do cliente VPN devem indicar qual o protocolo de *tunneling* a usar, por exemplo, PPTP e o endereço do servidor de VPN.

Após a estação wireless se associar à rede não-cablada o cliente VPN inicia o processo para estabelecer uma ligação VPN através de um pedido encaminhado para o NAS. Em seguida o servidor VPN encaminha as credencias de autenticação (nome de utilizador/senha) para o servidor RADIUS. O servidor RADIUS verifica se o utilizador cumpre as condições necessárias definidas nas políticas de autorização e no caso de as cumprir envia as credenciais para o serviço de directoria. Finalmente o serviço de directoria certifica a validade das credenciais e no caso destas serem válidas o processo de autenticação é concluído com sucesso e é estabelecido um túnel VPN entre o cliente e o servidor VPN. A atribuição do IP à estação wireless é efectuada por um servidor DHCP, através de um pedido feito por um *Relay Agent* configurado no servidor de VPN. Com o túnel estabelecido o utilizador tem acesso aos recursos da rede corporativa, com a garantia que os dados transmitidos na rede não-cablada são cifrados e transportados de forma segura entre a estação cliente e o NAS.

5.2 IEEE 802.11i, RSN e WPA

Na secção 5.1 abordou-se um mecanismo alternativo de segurança para redes não-cabladas IEEE 802.11 recorrendo a uma tecnologia consolidada como é o caso das redes privadas virtuais. Esta secção apresenta alguns conceitos referentes ao desenvolvimento de novas soluções, algumas das quais transitórias. Será feita a distinção entre o que de facto foi desenvolvido como solução transitória e o que está em fase de desenvolvimento e que pretende num futuro próximo ser o novo protocolo de segurança das redes não-cabladas IEEE 802.11.

5.2.1 Relação entre 802.11i, RSN e WPA

Tal como referido anteriormente, o IEEE é responsável pelo desenvolvimento e aprovação das diversas normas da família IEEE 802, *Local Area and Metropolitan Area Networks*. O grupo IEEE 802 é constituído por diversos grupos de trabalho, cada um deles com áreas de actuação e normalização especificas. O grupo de trabalho responsável pelas normas referentes a redes de área local não-cabladas é o 802.11. A norma original desenvolvida por este grupo em 1997 veio a tornar-se uma norma internacional em 1999. No entanto o trabalho continua e foram surgindo várias actualizações ao longo do tempo. Entre estas encontra-se em fase de desenvolvimento a norma IEEE 802.11i [16] que especifica novos mecanismos de segurança para redes não-cabladas 802.11.

O IEEE 802.11i define um novo tipo de rede não-cablada designada por rede de segurança robusta ou RSN (*Robust Security Network*). De acordo com [16] uma RSN é uma rede que permite apenas associações robustas e seguras ou RSNAs (*Robust Security Network Associations*). Isto é, numa RSN as associações entre todos os dispositivos wireless presentes numa rede, incluindo os pontos de acesso, são construídas sob uma associação/autenticação robusta designada RSNA. De acordo com [16] uma RSNA depende do IEEE 802.1X para transportar os seus serviços de autenticação e entregar os serviços de gestão de chaves.

A estrutura da futura norma IEEE 802.11i define duas classes de segurança para as redes não-cabladas 802.11: as redes RSN e as redes pré-RSN. São consideradas redes pré-RSN aquelas que possuam dispositivos wireless que não suportam os mecanismos de segurança específicos

das redes RSN. As redes RSN são compostas apenas por dispositivos que suportam os mecanismos de segurança definidos no IEEE 802.11i, a saber:

- Mecanismo de autenticação – Suportar IEEE 802.1X para os serviços de autenticação e gestão de chaves.
- Estabelecimento e gestão de chaves de cifra – Suportar gestão automática de chaves de cifra. Esta gestão depende dos serviços de gestão de chaves fornecidos pelo IEEE 802.1X, mais especificamente do mecanismo de troca de mensagens *4-way handshake* utilizado para estabelecer as chaves de cifra temporárias.
- Mecanismo de cifra dos dados – Suportar os algoritmos de cifra AES-CCMP (*Advanced Encryption Standard - Counter mode CBC-MAC Protocol*) e TKIP (*Temporal Key Integrity Protocol*). A implementação do TKIP é opcional.

À data não existem no mercado produtos com capacidades RSN. Tais produtos não deverão ser comercializados até que o protocolo IEEE 802.11i seja aprovado. A grande maioria dos dispositivos wireless existentes no mercado não poderão ser actualizados para suportar as capacidades RSN, uma vez que os requisitos das operações de criptografia não são suportados pelo hardware actual. Isto permite concluir que decorrerá algum tempo até que existam redes RSN.

Dados os factos anteriores, os fabricantes de dispositivos wireless optaram por "substituir" o mais rapidamente possível o protocolo WEP. Para endereçar esta necessidade, o grupo de trabalho do IEEE 802.11i em conjunto com a confederação *Wi-Fi Alliance* [Link3], iniciaram o desenvolvimento de uma solução de segurança baseada nas capacidades dos dispositivos actuais. Isto levou à definição de um novo protocolo baseado no WEP, o TKIP (*Temporal Key Integrity Protocol*) descrito na secção 5.5.2.

Devido às pressões do mercado, a *Wi-fi Alliance* optou pela definição de um conjunto de especificações baseadas no futuro protocolo IEEE 802.11i, designado por WPA (*Wi-Fi Protected Access*). O WPA partilha uma arquitectura de segurança comum às redes RSN para fornecer os serviços de autenticação, distribuição e gestão de chaves de cifra. A distinção entre WPA e RSN ocorre ao nível do serviço de confidencialidade e integridade. O WPA define o TKIP, enquanto as redes RSN incluem suporte para o algoritmo AES-CCMP, apenas disponível em hardware futuro.

Uma vez que o WEP é actualmente o protocolo mais implementado, a evolução natural é adoptar o WPA de imediato e avançar para a implementação de soluções baseadas em redes RSN à medida que os dispositivos forem surgindo no mercado. Note-se que, um dos principais problemas associados ao WEP, que é a inexistência de um mecanismo de gestão de chaves, é resolvido tanto pelo WPA como pelas futuras redes RSN.

5.2.2 Camadas de segurança

No contexto da segurança de redes de comunicações podem ser claramente identificadas três camadas [21]: (i) camada física ou wireless; (ii) camada de controlo de acesso; (iii) camada de autenticação (Figura 5-11). Note-se que este modelo de segurança em camadas não é específico das redes não-cabladas.

A camada física é responsável pelas comunicações no meio wireless. Esta camada é também responsável pela cifra e decifra dos dados uma vez estabelecido um contexto de segurança. Entenda-se por contexto de segurança o conjunto de mecanismos de segurança necessários a um sistema para que este possa ser considerado seguro.

A camada intermédia, controlo de acesso, só deve permitir a comunicação a uma entidade que prove possuir os mecanismos de segurança necessários a uma comunicação segura. Esta camada comunica com a camada de autenticação para decidir quando pode permitir o estabelecimento de um novo contexto de segurança a uma entidade. Os mecanismos de segurança adoptados para a camada de controlo de acesso são abordados na secção 5.3.

Na camada de autenticação, tal como o seu nome indica, são verificadas as provas de identidade fornecidas. Na secção 5.4 são analisados vários protocolos de autenticação propostos para utilização em redes não-cabladas.

A camada física reside no dispositivo wireless integrado no ponto de acesso e na placa de rede e nos *drivers* de *software* associados à estação cliente. Usualmente a camada de controlo de acesso está completamente integrada no ponto de acesso, que assume o papel de autenticador. O pedido de acesso é efectuado pela entidade suplicante presente na estação cliente. Apesar de em sistemas de pequena dimensão ser possível encontrar a camada de autenticação num ponto de acesso, normalmente esta é implementada num servidor de autenticação separado do ponto de acesso. A camada de autenticação é constituída pelo servidor de autenticação e pelo cliente de autenticação presente na estação cliente.

A Figura 5-11 representa a relação entre todas as camadas de segurança e os respectivos dispositivos.

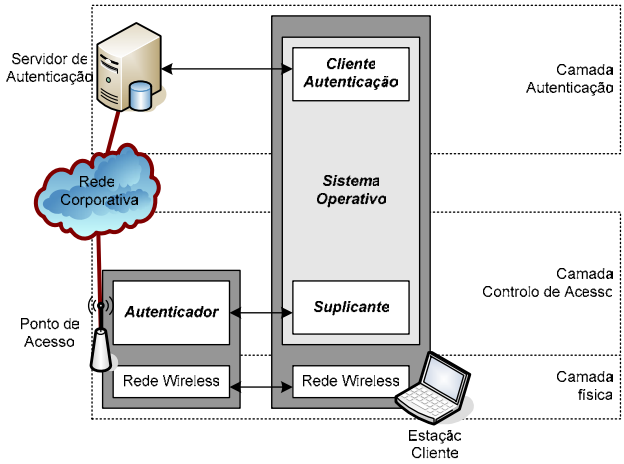


Figura 5-11 – Relação entre camadas de segurança.

O protocolo IEEE 802.11 especifica apenas os níveis físicos e de ligação de dados de redes não-cabladas, não sendo o seu grupo de trabalho responsável pela definição de camadas protocolares superiores. Isto representa um problema quando se pretende desenvolver um sistema que necessita da cooperação de várias camadas para o seu funcionamento. Esta foi uma das razões pela qual o protocolo WEP tentou definir todos os aspectos de segurança na camada wireless.

Para evitar este problema o grupo de trabalho IEEE 802.11i não especificou protocolos para a camada de controlo de acesso e para a camada de autenticação, optando antes pela utilização de protocolos e arquitecturas de segurança desenvolvidos por outros. Nos casos em que era necessário proceder a alterações devido às especificidade das redes não-cabladas o grupo IEEE 802.11i contactou os respectivos grupos de trabalho para que as alterações necessárias fossem efectuadas.

Relativamente à camada de controlo de acesso, o protocolo IEEE 802.1X [17], foi seleccionado como o mais apropriado. No entanto este teve que sofrer modificações para que todos os requisitos de segurança exigidos pelo grupo de trabalho do IEEE 802.11i fossem atingidos. O documento [62] em desenvolvimento reflecte as alterações pretendidas.

A camada de autenticação foi muito mais problemática. A dificuldade deveu-se às diferentes possibilidades existentes. No final a decisão tomada pelo grupo IEEE 802.11i é que este não indicaria nenhum método de autenticação obrigatório, mas o desenvolvimento da arquitectura

RSN deveria decorrer de tal modo que qualquer um dos métodos de autenticação que cumpram requisitos mínimos de segurança (por exemplo suporte de autenticação mútua) pudesse ser aplicado a uma RSN.

5.3 Mecanismos de controlo de acesso

Uma das funcionalidades chave que deve estar presente num contexto de segurança é o controlo de acesso. As soluções apresentadas nesta secção foram desenvolvidas em torno do protocolo IEEE 802.1X, apresentado na secção 5.3.1, que é um protocolo especificamente desenhado para implementar um mecanismo de controlo de acesso.

Esta secção apresenta ainda os protocolos EAP e RADIUS, respectivamente nas secções 5.3.2 e 5.3.5, que em combinação com os protocolos IEEE 802.11 e IEEE 802.1X proporcionam uma solução de controlo de acesso flexível que pode ser implementada tanto em redes não-cabladas de pequena dimensão como em redes corporativas de dimensão bastante considerável.

Na secção 5.3.3 é descrito o protocolo EAPoL (*EAP Over LAN*). Este especifica o método de encapsulamento das mensagens EAP, trocadas durante o processo de controlo de acesso IEEE 802.1X. A secção 5.3.4 apresenta as mensagens genéricas trocadas durante o processo de controlo de acesso IEEE 802.1X.

5.3.1 IEEE 802.1X

O protocolo IEEE 802.1X define uma estrutura de controlo de acesso em redes de comunicações. O controlo de acesso é implementado no ponto onde o utilizador se liga à rede. Este ponto de entrada é designado na norma IEEE 802.1X por *Port* ou porto. O controlo de acesso baseado no porto fornece compatibilidade entre mecanismos de autorização e autenticação (certificados digitais, cartões inteligentes, senhas únicas) utilizados nas mais diversas tecnologias de rede, não-cabladas IEEE 802.11, *Token Ring*, FDDI e redes locais 802.3.

5.3.1.1 Arquitectura

A análise efectuada nesta secção descreve os conceitos essenciais do protocolo de controlo de acesso IEEE 802.1X tendo em vista uma utilização no contexto de redes não-cabladas. As duas subsecções seguintes descrevem respectivamente, as entidades definidas na arquitectura IEEE 802.1X e o modelo funcional *Dual Port*. Essencialmente são definidos os conceitos e a terminologia associada aos diversos dispositivos intervenientes numa sessão de controlo de acesso IEEE 802.1X.

Entidades

Os dispositivos que fazem parte de uma rede são definidos no protocolo IEEE 802.1X como sistemas. Estes sistemas possuem um ou mais pontos de comunicação com a rede, denominados *Network Access Ports*, ou simplesmente *Ports*.

Outro conceito definido é o de PAE (*Port Access Entity*), que é a componente lógica que suporta o IEEE 802.1X e está associada a um porto.

A PAE tem a capacidade de adoptar um de dois papéis dentro do processo de interacção do controlo de acesso:

- Autenticador – O porto assume este papel sempre que pretende reforçar a autenticação antes de permitir o acesso a serviços que estão acessíveis através deste. Os serviços que um sistema fornece incluem, por exemplo, a função de encaminhamento de um *router*, ou a funcionalidade de um servidor de ficheiros, ou a funcionalidade de um ponto de acesso. No contexto das redes IEEE 802.11 podemos identificar este papel com o papel assumido por um ponto de acesso.

- Suplicante – O porto que pretende aceder a serviços disponibilizados pelo autenticador, adopta este papel. Num cenário de rede não-cablada a estação cliente assume o papel de suplicante.

Uma outra entidade definida na norma IEEE 802.1X e que apresenta um papel essencial no processo de controlo de acesso é o servidor de autenticação:

- Servidor de autenticação – Este sistema é responsável por verificar as credenciais do suplicante com interesse para o autenticador. Deve indicar ao autenticador se o

suplicante possui ou não autorização para acesso aos serviços disponibilizados pelo autenticador. Este papel é normalmente atribuído a um servidor AAA genérico.

As entidades definidas anteriormente são fundamentais para uma execução completa do processo de autenticação. Um dado sistema é capaz de assumir um ou mais destes papeis. Por exemplo, o autenticador e o servidor de autenticação podem estar associados ao mesmo sistema. Este é um dos aspectos que permitem a utilização do IEEE 802.1X, em cenários de redes não-cabladas de menor dimensão. Uma vez que na prática o servidor de autenticação pode ser um simples processo a operar no ponto de acesso, como por exemplo uma lista de utilizadores e respectivas palavra-chave, o princípio do IEEE 802.1X pode ser aplicado tanto a redes não-cabladas corporativas como a redes domésticas. Se o servidor de autenticação for implementado no ponto de acesso não existe a necessidade de recorrer a um servidor exterior. No entanto fica-se limitado aos métodos de autenticação suportados pelo fabricante do ponto de acesso.

De forma semelhante um porto pode adoptar o papel de suplicante em algumas fases do processo de autenticação, e assumir o papel de autenticador durante outras. Esta situação ocorre, por exemplo, em situações em que o protocolo de autenticação utilizado requer autenticação mútua.

Modelo Dual Port

A Figura 5-12, ilustra o conceito de autenticação baseado em portos, ou modelo *Dual Port*. Este conceito assenta na definição de dois portos de acesso distintos entre o sistema autenticador e o seu ponto de ligação à rede. Estes portos são designados por porto controlado (*Controlled Port*) e porto não controlado (*Uncontrolled port*).

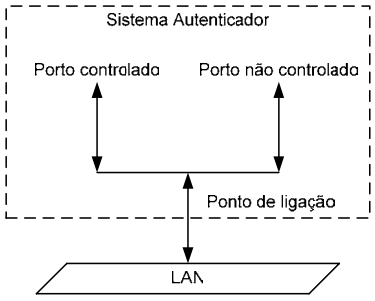


Figura 5-12 – Porto controlado e porto não controlado.

O porto não controlado permite a troca não controlada de pacotes entre sistemas independentemente do seu estado de autenticação. O porto controlado, permite a troca de informação entre sistemas apenas se o estado de autenticação do porto o permitir.

Ambos os portos fazem parte de um mesmo ponto de ligação do sistema à rede, isto é, qualquer informação recebida no porto físico do sistema é disponibilizada a ambos os portos ficando sujeita ao estado de autorização associado ao porto controlado.

O ponto de ligação deve ser entendido como um porto lógico ou físico capaz de fornecer uma ligação ponto-a-ponto entre o sistema autenticador e o sistema suplicante.

No cenário particular de uma rede não-cablada, o autenticador deve permitir a troca de pacotes EAP antes de proceder à autenticação do suplicante. O modelo *Dual-Port* definido anteriormente é utilizado neste processo. A Figura 5-13 apresenta o conceito deste modelo num sistema autenticador.

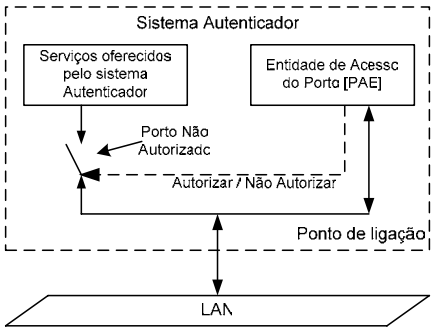


Figura 5-13 – Modelo Dual Port no sistema autenticador.

O sistema autenticador possui dois portos para acesso à rede: o porto não controlado e o porto controlado. O primeiro deve filtrar todo o tráfego da rede e permitir apenas a passagem de pacotes EAP para autenticação com servidor de autenticação. Quanto ao porto controlado este permite ou não o acesso aos serviços fornecidos pelo sistema autenticador. Esta permissão é dada pela mensagem de autorização enviada pelo servidor de autenticação ao autenticador.

A PAE do autenticador recorre ao porto não controlado para a troca de informação do protocolo de autenticação com o suplicante. A troca de mensagens entre o autenticador e o servidor de autenticação (no caso destes estarem localizados em sistemas diferentes) pode ser conduzida por um ou mais portos do sistema.

A Figura 5-14 ilustra a relação entre as três entidades definidas anteriormente.

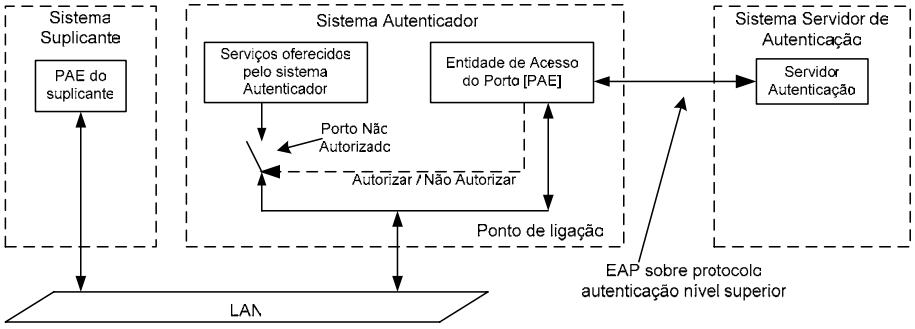


Figura 5-14 – Relação entre autenticador, suplicante e servidor de autenticação.

Como se verifica pela análise da Figura 5-14 o porto controlado do autenticador está no estado não autorizado e do ponto de vista de acesso aos serviços fornecidos pelo sistema autenticador encontra-se desactivado. A PAE do autenticador recorre ao porto não controlado para comunicar com a PAE do suplicante. Esta comunicação recorre ao protocolo EAPoL (*Extensible Authentication Protocol Over LANs*). O EAPoL é definido na norma IEEE 802.1X. Uma descrição deste protocolo é efectuada na secção 5.3.3.

A comunicação entre a PAE do autenticador e o servidor de autenticação, no caso deste estar localizado remotamente, recorre ao EAP sobre um protocolo de camada superior. O WPA especificou o protocolo RADIUS (*Remote Authentication Dial In User Service*) para efectuar esta operação. No entanto outros protocolos podem ser utilizados como por exemplo o DIAMETER [18].

A partir do momento em que o servidor de autenticação confirma a validade das credenciais do suplicante, a PAE deste tem acesso autorizado aos serviços fornecidos pelo sistema autenticador.

5.3.2 Protocolo EAP – Extensible Authentication Protocol

O objectivo desta secção é o de descrever de uma forma não exaustiva o protocolo EAP (*Extensible Authentication Protocol)*, apresentando alguns conceitos importantes à sua compreensão quando utilizado em conjunto com o protocolo IEEE 802.1X para implementar um mecanismo de autenticação eficaz em redes IEEE 802.11. A descrição é organizada em três subsecções. A primeira descreve o modelo genérico do protocolo EAP. De seguida é apresentado o formato do pacote EAP e os respectivos campos que o compõem. Finalmente enumeram-se alguns dos métodos de autenticação suportados pelo protocolo EAP.

O protocolo EAP encontra-se definido em [RFC2284]. O EAP permite a troca de informação especifica do método de autenticação utilizado entre duas entidades. O conteúdo desses métodos de autenticação não é definido no EAP. De facto, estes métodos de autenticação podem ser completamente proprietários, ou totalmente novos. A razão pela qual o EAP é extensível prende-se com o facto de os detalhes das suas mensagens não serem definidos pelo EAP, mas sim por RFCs próprios. Por exemplo, existe um RFC que define como utilizar TLS (*Transport Layer Security*) sobre EAP e outro (*draft*) que define a utilização de TTLS (*Tunneled Transport Layer Security*).

Dada a importância que este protocolo vem assumindo actualmente, houve a necessidade de o actualizar. Esta actualização está em fase de conclusão. O trabalho em desenvolvimento pode ser consultado em [19].

Modelo

Conceptualmente, as implementações do EAP, aderem a um modelo composto por três níveis protocolares: físico, EAP e nível de autenticação ou de métodos EAP (Figura 5-15).

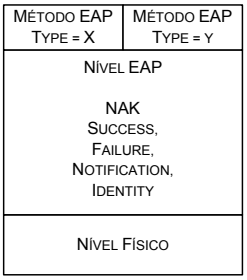


Figura 5-15 – Modelo protocolo EAP.

Cada um dos níveis tem funções específicas:

- Nível físico – Nível responsável pela transmissão e recepção de pacotes EAP entre duas entidades, por exemplo entre o suplicante (ou *peer* como é denominado no RFC2284) e o autenticador. O EAP têm sido implementado numa série de níveis físicos distintos incluindo, redes cabladas IEEE 802 e redes não-cabladas IEEE 802.11.
- Nível EAP – Nível referente ao EAP propriamente dito. É responsável por receber e transmitir pacotes EAP através do nível físico. Este nível implementa a máquina de estados do protocolo EAP com a transmissão e recepção de pacotes EAP de e para o nível de métodos EAP.

- Nível de métodos – Nível que implementa os algoritmos de autenticação suportados pelo EAP.

Formato dos pacotes

O formato típico de um pacote EAP está apresentado na Figura 5-16.

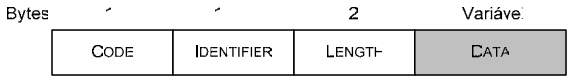


Figura 5-16 – Formato do pacote EAP.

O campo ***Code*** identifica o tipo de pacote EAP. Este campo pode tomar um dos valores apresentados na Tabela 5-1. Todos os pacotes que contenham um valor diferente dos presentes na Tabela 5-1 são descartados pelas entidades intervenientes no processo de autenticação.

Código	Pacote EAP
1	Request
2	Response
3	Success
4	Failure

Tabela 5-1 – Códigos possíveis para os pacotes EAP.

O campo ***Identifier*** tem as funcionalidades de identificação dos pacotes, o valor do campo ***Length*** representa o comprimento do pacote EAP e finalmente o campo ***Data*** ou campo de dados tem um formato que varia de acordo com o valor do campo ***Code***.

O formato dos pacotes ***Request*** e ***Response*** é o mesmo e pode ser observado na Figura 5-17. Os pacotes ***Request*** são enviados do autenticador para o suplicante com o valor do campo ***Code*** igual a 1. Os pacotes ***Response*** são enviados do suplicante para o autenticador, com o valor do campo ***Code*** igual a 2 e o valor presente no campo ***Identifier*** deve corresponder ao do pacote ***Request*** ao qual está a responder. Os pacotes ***Response*** só devem ser enviados mediante a recepção de um ***Request*** e nunca através da expiração de um temporizador.

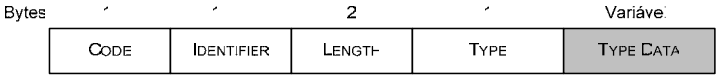


Figura 5-17 – Formato dos pacotes EAP Request e Response.

O formato dos pacotes ***Success*** e ***Failure*** é o mesmo e é representado na Figura 5-18.

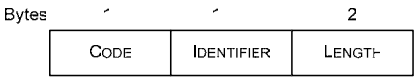


Figura 5-18 – Formato dos pacotes EAP Success e Failure.

Os pacotes ***Success*** e ***Failure*** são enviados do autenticador para o suplicante dependendo se a autenticação é aceite ou não. É possível que um autenticador envie múltiplos pedidos ao suplicante antes de enviar um pacote de ***Failure***, com o intuito de proporcionar novas tentativas de autenticação.

Campo Type

Os campos ***Type*** e ***Type-Data*** têm um papel de destaque nos pacotes EAP pois é através destes que é identificado o método de autenticação a utilizar pelo EAP.

O campo ***Type*** indica o tipo de pedido ou resposta, isto é, qual o método de autenticação a utilizar. Os diversos métodos de autenticação definidos no [RFC2284] são apresentados na Tabela 5-2. Apenas um tipo pode ser especificado por cada pacote ***Request*** ou ***Response***.

Normalmente o tipo especificado num pacote de resposta é o mesmo que foi especificado no pacote de pedido. Existem, no entanto, situações em que tal não acontece pelo facto de o suplicante não suportar o tipo de autenticação enviado no pacote de pedido.

Nestes casos o pacote ***Response*** especifica o tipo ***Nak***, podendo ainda ser indicado um tipo alternativo de autenticação suportado pelo suplicante. Relativamente ao campo ***Type-Data*** o seu valor varia de acordo com o tipo de pedido e respectiva resposta.

Valor	Type
1	Identity
2	Notification
3	Nak
4	EAP/MD5
5	One Time Password (OTP)
6	Generic Token Card (GTC)
13	EAP-TLS
21	EAP-TTLS
25	PEAP

Tabela 5-2 – Tipos possíveis para os pacotes EAP Request e Response.

O campo ***Type*** com valor igual a 1, ***Identity*** , pode ser considerado um tipo especial de mensagem ou um método de autenticação bastante simples. Um ***Request*** com campo ***Type***

do tipo ***Identity*** é frequentemente o primeiro a ser enviado pelo autenticador para obter uma resposta do suplicante com informação da sua identidade.

Originalmente este processo era visto como um procedimento posterior à fase de autenticação principal. No entanto foi ligeiramente alterado no documento de revisão do EAP [19], que considera a troca de mensagens ***Identity*** um processo de autenticação por si só. Quando se executa a troca de mensagens do tipo ***Identity*** seguida por outro método de autenticação como por exemplo TLS, na realidade estão a executar-se dois métodos de autenticação. Este conceito de autenticação em série é generalizado na revisão do EAP [19]. Este simplesmente identifica as mensagens do tipo ***Identity*** como um método de autenticação básico e afirma que podem ser executados, previamente às mensagens finais de ***EAP-Success*** ou ***EAP-Failure***, tantos métodos de autenticação em sequência quantos os desejados.

A capacidade para executar múltiplos métodos de autenticação sequencialmente é explorada em novos métodos de autenticação que permitem ao cliente a autenticação na rede antes mesmo de revelar a sua identidade. Uma aproximação deste conceito é abordada na secção 5.4.3, através do protocolo PEAP (*Protected EAP*).

O número de métodos de autenticação não se esgota na lista apresentada na Tabela 5-2. Uma série de novos métodos têm sido propostos e aprovados, além de outros proprietários que são suportados pelo protocolo EAP.

Métodos Autenticação suportados pelo protocolo EAP

Alguns dos principais métodos de autenticação EAP utilizados actualmente em cenários de rede não-cablada são:

- **EAP-MD5** – Equivalente ao protocolo CHAP [RFC1994]. O suplicante é autenticado pelo servidor de autenticação através de uma senha fornecida pelo cliente. Este método não efectua autenticação mútua, pelo que o suplicante não consegue autenticar o servidor de autenticação. Não são geradas chaves de cifra durante o processo de autenticação.
- **EAP-TLS** – Fornece autenticação mútua do suplicante e do servidor de autenticação recorrendo a certificados digitais. A aplicação deste método implica a necessidade de uma infra-estrutura de chave pública. As chaves de cifra são geradas durante o processo de autenticação. Este protocolo será descrito em detalhe na secção 5.4.1.3.

- **Protected EAP (PEAP)** – Fornece autenticação mútua das entidades intervenientes no processo de autenticação, recorrendo obrigatoriamente a certificados digitais para autenticar o servidor de autenticação, a autenticação do suplicante pode recorrer a outros métodos de autenticação suportados pelo protocolo EAP. Este protocolo será descrito em detalhe na secção 5.4.3.
- **EAP-Tunneled TLS (EAP-TTLS)** – Mecanismo de autenticação proprietário da empresa *Funk*. Combina o método EAP-TLS com métodos de autenticação tradicionais como sejam, CHAP, PAP [RFC1334], MS-CHAP [RFC2759]. A utilização de certificados digitais por parte do suplicante não é obrigatória. A autenticação pode ser por senha. O servidor de autenticação é autenticado através de certificado digital. As chaves de cifra são geradas durante o processo de autenticação. Este protocolo será descrito em detalhe na secção 5.4.4.

A Figura 5-19 reflecte a tradução dos diversos níveis protocolares do modelo EAP num diagrama de blocos representativo dos protocolos suportados em cada um desses níveis.

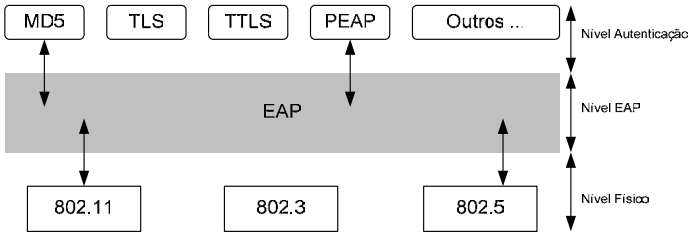


Figura 5-19 – Diagrama de blocos do EAP.

5.3.3 Protocolo EAPoL – EAP Over LAN

A arquitectura IEEE 802.1X define a técnica de encapsulamento que deve ser utilizada para transmitir pacotes EAP entre a PAE do suplicante e a PAE do autenticador. Esta técnica de encapsulamento é designada por EAPoL (*EAP Over LAN*).

As entidades EAPoL definidas no IEEE 802.1x são o suplicante EAPoL e o autenticador EAPoL. O servidor de autenticação não deve ser visto como uma entidade EAPoL, a menos que este esteja integrado no mesmo dispositivo que o autenticador.

O documento [802.1X-2001] descreve o formato dos pacotes EAPoL para redes Ethernet/ IEEE 802.3 e Token Ring. O formato de uma trama EAPoL é representado na Figura 5-20 começando a representação com o campo ***Length/Type*** de uma trama MAC.

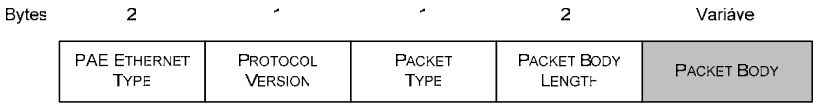


Figura 5-20 – Formato de uma trama EAPoL.

O campo ***PAE Ethernet Type*** contêm o valor hexadecimal 88-8E que define uma PAE do tipo 802.1X. O campo ***Protocol Version*** contêm o valor que identifica a versão do EAPoL. Os valores deste campo são definidos em [17]. O campo ***Packet Type*** descreve o tipo de pacote EAPoL. A Tabela 5-3 apresenta os cinco tipos de pacotes EAPoL definidos em [17]. O valor presente no campo ***Packet Body Length*** indica o tamanho em bytes do campo ***Packet Body***. Se este valor for igual a zero significa que o pacote EAPoL não possui o campo ***Packet Body***. Este último campo está presente se o pacote EAPoL for do tipo ***EAP-Packet***, ***EAPoL-Key*** ou ***EAPoL–Encapsulated-ASF- Alert***. Para os restantes pacotes EAPoL este campo não existe.

Valor	Tipo de Pacote
0x00	EAPoL – Packet
0x01	EAPoL – Start
0x02	EAPoL – Logoff
0x03	EAPoL – Key
0x04	EAPoL – Encapsulated-ASF- Alert

Tabela 5-3 – Tipo de pacotes EAPoL.

Sempre que uma estação cliente (suplicante) estabelece uma ligação à rede não-cablada não tem conhecimento do endereço MAC do autenticador. Na fase inicial do processo de controlo de acesso o suplicante envia um pacote do tipo ***EAPoL-Start*** para um endereço MAC do tipo *multicast*. Este endereço é reservado para os autenticadores 802.1X. Com esta mensagem o suplicante informa os autenticadores que está preparado para se identificar.

Os pacotes ***EAPoL-Packet*** servem para transportar pacotes do tipo EAP. Por exemplo o pedido de identificação enviado do autenticador para o suplicante, pacote EAP do tipo ***Request-Identity***, é enviado num pacote ***EAPoL-Packet***.

Para terminar uma ligação o suplicante envia um pacote do tipo ***EAPoL-Logoff***. As chaves de cifra obtidas do processo de autenticação de camada superior (na secção 5.4 são descritos

diversos processos de autenticação de camada superior), são enviadas do autenticador para o suplicante num pacote ***EAPoL-Key***.

Obviamente é necessário cifrar as chaves antes destas serem enviadas. Este processo de cifra não é definido pelo IEEE 802.1X, mas está a ser considerado pelo grupo de trabalho IEEE 802.1AA [20]. Este é um dos aspectos fundamentais tidos em conta no desenvolvimento de redes RSN. Na secção 5.5.1 é abordada a forma como o pacote ***EAPoL-Key*** é utilizado para estabelecer chaves de cifra.

O último tipo de pacote, ***EAPoL-Encapsulated-ASF-Alert*** está relacionado com a possibilidade de uma entidade enviar alertas de gestão para o sistema. Este tipo de pacote não é utilizado em redes RSN nem no WPA.

Um exemplo de uma trama EAPoL capturada durante o processo de controlo de acesso a uma rede não cablada é apresentado na Figura 5-21.

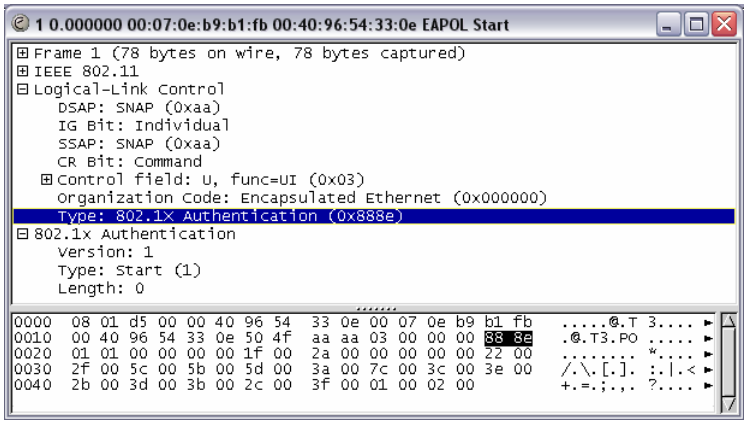


Figura 5-21 – Exemplo de uma trama EAPoL.

Como se verifica o ***PAE Ethernet Type*** possui o valor hexadecimal 88-8E. O valor do campo ***Type*** identifica o tipo de pacote EAPoL que neste exemplo é igual a 1 o que significa, de acordo com a Tabela 5-3, que é um pacote do tipo ***EAPoL-Start***. Actualmente a versão do EAPoL é igual a 1. O campo ***Length*** possui o valor zero, uma vez que este tipo de pacote EAPoL não possui o campo ***Packet Body***.

5.3.4 Mensagens Trocadas no IEEE 802.1X

Após a descrição dos protocolos intervenientes no processo de controlo de acesso IEEE 802.1X, interessa descrever de forma genérica a aplicação da estrutura 802.1X a um cenário de

rede não-cablada IEEE 802.11. A troca de mensagens entre as entidades intervenientes num processo genérico de controlo de acesso é apresentada na Figura 5-22.

A sequência de autenticação tem início no pedido da estação móvel, o suplicante, através de uma mensagem ***EAPoL-Start*** enviada para o ponto de acesso, o autenticador. Assim que o autenticador detecta o pedido do suplicante envia um pedido de identificação do utilizador através de uma mensagem ***EAP Request-Identity***. A resposta do suplicante é dada através de um pacote ***EAP Response-Identity***. Após identificar o utilizador que pretende acesso aos recursos da rede não-cablada, o servidor de autenticação transmite ao autenticador o protocolo de autenticação suportado. O autenticador informa o suplicante através de uma mensagem ***EAP Request***. De acordo com a Tabela 5-2 o campo ***Type*** desta mensagem identifica o protocolo de autenticação a ser utilizado pelo suplicante. Se o suplicante suporta esse método de autenticação, então responde com uma mensagem ***EAP-Response***.

Após esta fase segue-se a troca de mensagens referentes ao protocolo de autenticação negociado previamente. Na Figura 5-22 não é considerado nenhum mecanismo de autenticação em particular. Uma análise a diversos mecanismos de autenticação de camada superior utilizados em redes não-cabladas IEEE 802.11 é efectuada na secção 5.4.

A conclusão do processo de controlo de acesso ocorre quando o servidor de autenticação informa a estação móvel do resultado do processo de autenticação. No exemplo da Figura 5-22 a autenticação do utilizador é concluída com sucesso, através de uma mensagem ***EAP-Success***. O ponto de acesso regista a autorização dada pelo servidor de autenticação e passa a disponibilizar os seus serviços à estação autenticada.

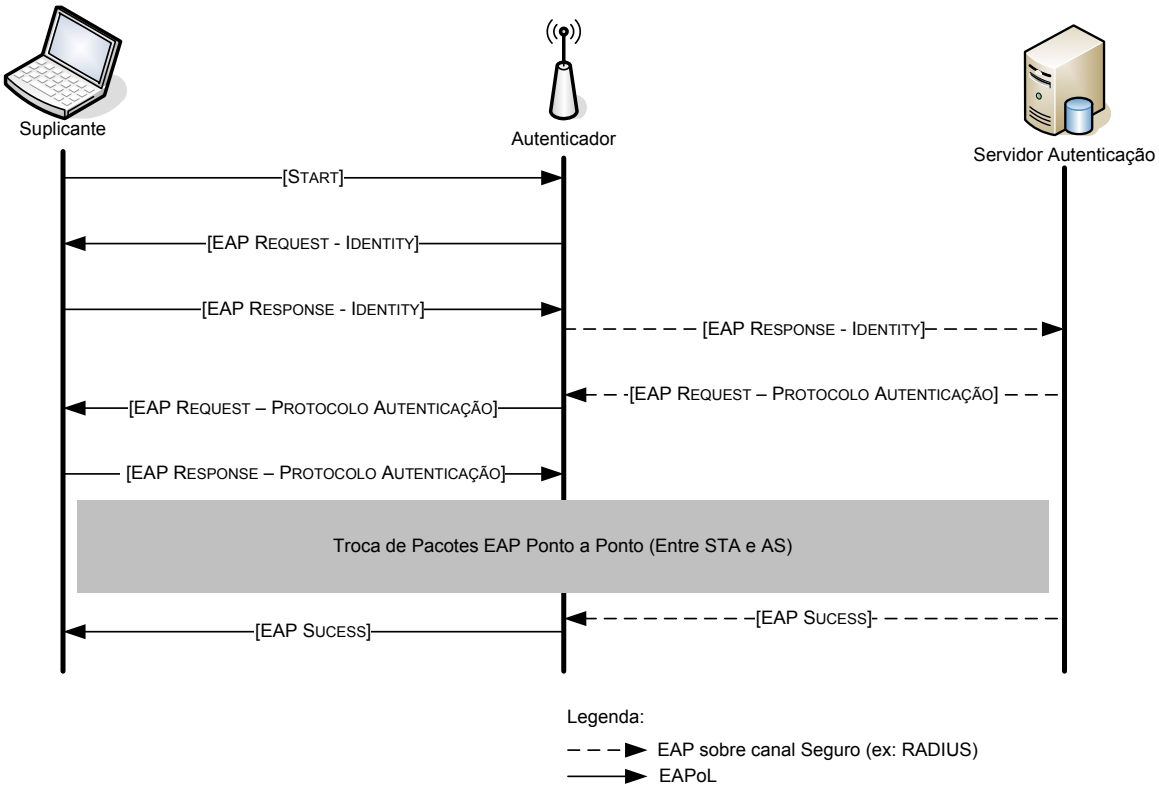


Figura 5-22 – Processo de controlo de acesso IEEE 802.1X.

Importa ainda referir que o IEEE 802.1X não define por si só como são trocadas as mensagens entre o autenticador e o servidor de autenticação. No entanto tem sido sugerida a utilização do protocolo RADIUS. O WPA vai mais longe e define como obrigatório o suporte ao protocolo RADIUS, de modo a tornar possível a interoperabilidade entre as diversas implementações.

5.3.5 RADIUS (Remote Authentication Dial In User Service)

A implementação de uma arquitectura de autenticação, autorização e contabilização, ou arquitectura AAA (*Authentication, Authorization e Accounting*) em redes não-cabladas IEEE 802.11 recorrendo ao protocolo RADIUS deve obedecer a alguns requisitos mínimos. Por exemplo, o servidor AAA deve suportar EAP sobre RADIUS [RFC2869]. Esta especificação não está incluída nas especificações originais do protocolo RADIUS [RFC2865]. Nesta secção não se pretende descrever em detalhe o protocolo RADIUS. A abordagem feita refere apenas os aspectos principais das funcionalidades utilizadas em redes não-cabladas.

Um estudo mais completo deste protocolo é apresentado em [31]. As especificações publicadas pelo IETF, relevantes para redes não-cabladas são: i) *Remote Authentication Dial-In User Radius* [RFC2865], *RADIUS Accounting* [RFC2866], *RADIUS Accounting for Tunneling* [RFC2867], *RADIUS Authentication for Tunneling* [RFC2868], *RADIUS Extensions* [RFC2869], *Microsoft Vendor-Specific RADIUS Attributes* [RFC2548]. A especificação [RFC2869] é particularmente relevante porque possui informação de como é implementado o EAP sobre RADIUS. Uma actualização a este documento pode ser encontrada em [RCF3579].

5.3.5.1 Arquitectura AAA

A implementação de uma arquitectura AAA permite a utilizadores legítimos o acesso aos recursos disponibilizados numa rede, impedindo o acesso não-autorizado a esses mesmos recursos. A Figura 5-23 apresenta um exemplo de implementação de uma arquitectura AAA.

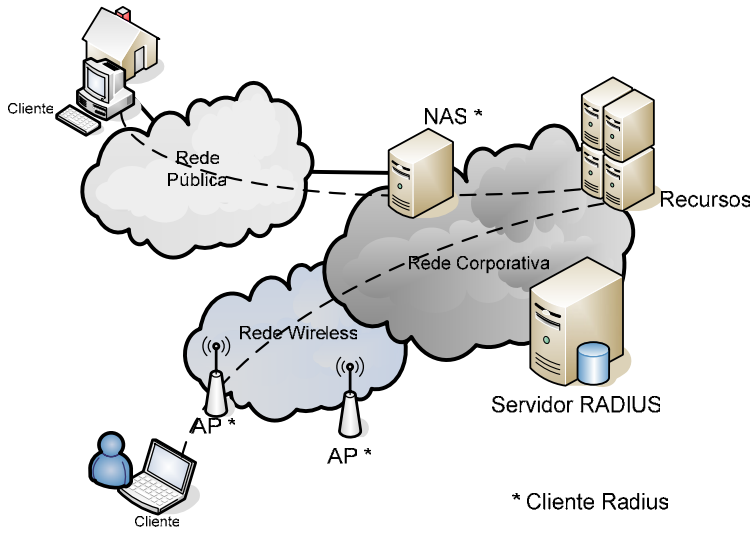


Figura 5-23 – Exemplo de implementação de uma arquitectura AAA.

Como se pode verificar esta arquitectura tem um vasto campo de aplicações, que podem ir desde a protecção para acessos remotos, através de um NAS (*Network Access Server*) até à utilização na autenticação e controlo de acesso a redes não-cabladas. A permissão para aceder aos recursos da rede corporativa é fornecida após a autorização dada pelo servidor de AAA. Neste exemplo, o servidor de AAA é um servidor RADIUS. Existem outros protocolos de AAA nomeadamente, TACACS+ e DIAMETER. Destes apenas será abordado o protocolo RADIUS por ser de suporte obrigatório no WPA e ainda ser o mais utilizado neste tipo de arquitectura.

Da arquitectura representada destacam-se dois componentes fundamentais: (ii) os clientes RADIUS, no exemplo o NAS para acessos remotos e os pontos de acesso da rede wireless, que são responsáveis por enviar a informação dos "clientes" para o servidor RADIUS; (ii) o servidor RADIUS responsável por receber os pedidos de ligação por parte dos "clientes", autenticar os "clientes", informar os clientes RADIUS para que estes permitam ou não o acesso aos "clientes" e opcionalmente iniciar o serviço de contabilização.

5.3.5.2 Mecanismos RADIUS

Esta secção descreve a operação do protocolo RADIUS. O formato das mensagens RADIUS é extremamente simples. A complexidade maior recai nas mensagens que contêm os atributos RADIUS.

O RADIUS utiliza o UDP como protocolo de transporte. Apenas um pacote RADIUS pode ser transportado no campo de dados de um pacote UDP. Os portos UDP utilizados por omissão são o 1812 para serviços de autenticação e autorização e o 1813 para o serviço de contabilização.

A Figura 5-24 mostra o formato típico de um pacote RADIUS.

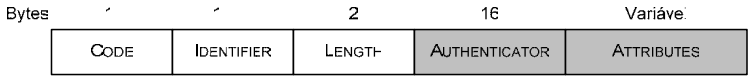


Figura 5-24 – Formato de um pacote RADIUS.

O campo ***Code*** identifica o tipo de pacote RADIUS. Os valores permitidos encontram-se na Tabela 5-4.

Código	Tipo de Pacote RADIUS
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reservado

Tabela 5-4 – Códigos possíveis para os pacotes RADIUS.

O campo ***Identifier*** serve para estabelecer a correspondência entre os pedidos e as respostas. O campo ***Length*** indica o tamanho total do pacote RADIUS. Existem dois tipos específicos de campo ***Authenticator***, o ***Request-Authenticator*** e o ***Response-Authenticator***. O primeiro é utilizado nas mensagens ***Access-Request*** e ***Accounting-Request***. O segundo é utilizado nas mensagens do tipo ***Access-Accept***, ***Access-Reject*** e ***Access-Challenge***.

O campo ***Request-Authenticator*** é um desafio de 16 bytes. Este valor deve ser único durante o tempo de vida da chave simétrica partilhada entre o servidor e o cliente RADIUS. O valor deste campo e a chave simétrica são utilizados como parâmetros de entrada numa função de síntese (algoritmo MD5) que gera um código de 16 bytes.

O campo ***Response-Authenticator*** é usado para autenticar a resposta do servidor. O seu valor corresponde ao resultado da função de síntese MD5, obtida a partir dos campos ***Code***, ***Identifier***, ***Length***, ***Request-Authenticator*** e ***Attributes*** seguidos da chave de cifra. A expressão seguinte apresenta o que foi referido:

Response-Authenticator =MD5 (Code||Identifier||Length||Request-Authenticator||Attributes||chave)

O campo ***Attributes*** tem comprimento variável e contém uma lista de atributos necessários à comunicação.

Atributos RADIUS

Os atributos são transportados pelos pacotes RADIUS e contêm informação de autenticação, autorização e detalhes de configuração específicos a uma determinada ligação. Um pacote RADIUS pode transportar múltiplos atributos, sendo o último atributo definido pelo comprimento do pacote RADIUS, campo ***Length***.

O formato típico de um atributo RADIUS é mostrado na Figura 5-25.

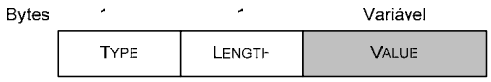


Figura 5-25 – Formato de um atributo RADIUS.

O campo ***Type*** define o tipo de atributo. Valores recentes atribuídos a este campo, podem ser encontrados no [RFC1700]. Um servidor RADIUS e um cliente RADIUS podem ignorar atributos cujo tipo é desconhecido. Alguns dos atributos mais importantes encontram-se listados na Tabela 5-5.

Valor	Nome	Descrição
1	User-Name	Identificação do utilizador
2	User-Password	Transporta informação de autenticação que o utilizador fornece para aceder aos serviços da rede. O conteúdo deste atributo poderá ser uma senha cifrada mas também pode ser a resposta do cliente a um desafio enviado por um pacote *Access-Challenge*.
3	CHAP-Password	Durante o CHAP (*Challenge-Handshake Authentication Protocol*), o desafio recebido pelo cliente é cifrado e enviado ao servidor de autenticação neste atributo.
4	NAS-IP Address	Endereço IP do NAS ao qual o servidor RADIUS deve responder.
5	NAS-Port	O valor deste atributo representa o porto ao qual o cliente está ligado.
26	Vendor-Specific	Este atributo permite a fabricantes a implementação de funcionalidades especificas relevantes apenas para os seus equipamentos. Se estes atributos se tornarem públicos, outros fabricantes podem suportar as mesmas funcionalidades.
32	NAS-Identifier	Identificação do NAS.
40	Acct-Session-Time	Este atributo pode ser encontrado nos pacotes do tipo *Accounting-Request*. O seu valor indica o tempo de ligação do cliente em segundos.
48	Acct-Output-Packets	Este atributo pode ser encontrado nos pacotes do tipo *Accounting-Request*. O valor deste atributo representa a quantidade de pacotes transmitidos por um cliente durante uma sessão.

Tabela 5-5 – Alguns tipos de atributos incluídos nos pacotes RADIUS.

O campo ***Length*** indica o tamanho do atributo, incluindo o campo ***Type***, ***Length*** e ***Value***. Se um atributo, contido num pacote ***Access-Request***, tiver um comprimento incorrecto, o servidor RADIUS deve enviar um ***Access-Reject***. O campo ***Value*** contém informação específica ao atributo. O formato e comprimento do campo ***Value*** é determinado pelos campos ***Type*** e ***Length***. De qualquer forma, o formato do campo ***Value*** pode ser um dos seguintes:

- Text - 1 a 253 bytes, contendo caracteres codificados UTF-8;
- String - 1 a 253 bytes, contendo dados binários;
- Address - valor de 32 bits;
- Integer - valor de 32 bits (positivo)
- Time - valor de 32 bits (positivo), a usar por futuros atributos.

Uma descrição completa de cada um dos atributos é apresentada em [RFC2865].

5.3.5.3 EAP sobre RADIUS

O [RFC2869] define um conjunto de extensões ao protocolo RADIUS, nomeadamente o suporte do EAP em mensagens RADIUS. Assim as mensagens de autenticação EAP são enviadas para o servidor numa mensagem ***Access-Request*** e a resposta é retornada para o NAS, ou ponto de acesso no caso de redes não-cabladas, numa mensagem ***Access-Challenge***.

Tome-se como referência a Figura 5-22, onde é ilustrado a troca de mensagens entre o autenticador e o servidor de autenticação, num processo de autenticação genérico. A mensagem de resposta do cliente, ***EAP Response-Identity***, ao pedido de identificação ***EAP Request-Identity*** enviada pelo ponto de acesso, é transmitida para o servidor de autenticação numa mensagem RADIUS do tipo ***Access-Request***. A resposta de sucesso de autenticação, ***EAP-Success*** transmitida pelo servidor é enviada numa mensagem RADIUS do tipo ***Access-Challenge***.

5.3.5.4 Utilização do RADIUS em redes RSN e WPA

Em cenários de rede não-cablada devem ser considerados alguns aspectos na implementação do protocolo RADIUS. Ao contrário das redes *dial-up*, onde a principal utilização está relacionada com a autenticação dos utilizadores, as redes RSN e WPA têm requisitos mais exigentes. Além da autenticação é necessário que sejam fornecidos mecanismos de integridade e autenticação por pacote transmitido. Para fornecer estes mecanismos de protecção, o servidor de autenticação deve enviar uma chave de cifra para o ponto de acesso. O processo de geração e obtenção da chave de cifra é descrito em detalhe na secção 5.5.1. Os antigos servidores RADIUS baseados nos RFC2865-2869, não forneciam um método de enviar a chave de cifra de um servidor para o NAS. Assumia-se neste caso o envio da chave de cifra para validação através de outro método. Para solucionar este problema, foi adoptado um atributo específico de fabricante, concretamente o atributo proprietário da Microsoft designado MS-MPPE-Recv-Key, desenvolvido especificamente para a entrega de chaves ao NAS. Este atributo é definido em [RFC2548] e foi adoptado no WPA como o método para a troca da chave de cifra, entre o servidor RADIUS e o ponto de acesso.

Conclui-se assim que os requisitos para utilização de RADIUS em redes RSN/WPA são o suporte do protocolo RADIUS e do atributo MS-MPPE-Recv-Key por parte do ponto de acesso e do servidor de autenticação, incluindo as extensões EAP. O suporte RADIUS não é referido actualmente nos trabalhos do grupo IEEE 802.11i como obrigatório. No entanto as especificações WPA obrigam à utilização deste protocolo.

5.4 Mecanismos de Autenticação

Esta secção apresenta os protocolos de autenticação de camada superior adoptados no WPA e nas redes RSN.

Na secção 5.4.1 é abordado o método adoptado por omissão pelo WPA, o TLS (*Transport Layer Security*). Na secção 5.4.2 é apresentado um protocolo de autenticação proprietário, o LEAP (*Cisco Ligth EAP*), o que se justifica devido à sua enorme penetração no mercado. Os protocolos de autenticação PEAP (*Protected EAP*) e EAP-TTLS (*EAP-Tunneled Transport Layer Security*), actualmente em fase de desenvolvimento, são analisados na secção 5.4.3 e 5.4.4 respectivamente.

5.4.1 TLS – Transport Layer Security

5.4.1.1 Funcionalidades

O protocolo TLS completo fornece serviços de autenticação, cifra e funções de compressão de dados. De acordo com [21] esta última funcionalidade nunca terá sido implementada na prática, nem possui qualquer interesse para utilização em redes wireless, uma vez que estas não especificam qualquer tipo de compressão de dados. A funcionalidade de cifra do TLS também não é importante, dado que o WPA e as redes RSN utilizam métodos próprios, analisados posteriormente na secção 5.5. No entanto a sua funcionalidade de autenticação é bastante apropriada para implementação no modelo 802.1X abordado na secção 5.3.

O protocolo TLS é dividido em duas camadas, que representam dois protocolos: o protocolo de registo (*Record Protocol*) e o protocolo de troca de mensagens (*Handshake Protocol*). O primeiro é responsável pela transferência de dados numa ligação entre duas entidades; o segundo define os parâmetros das mensagens utilizadas no protocolo de registo.

A Figura 5-26 apresenta as duas camadas protocolares definidas pelo TLS.

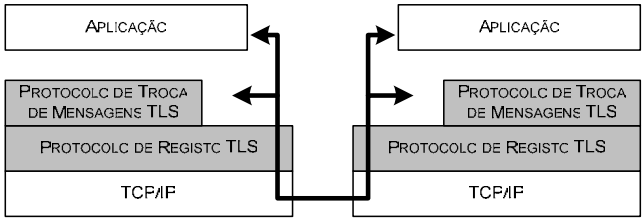


Figura 5-26 – Camadas protocolares do TLS.

Os dados são encaminhados desde a camada de aplicação para o protocolo de registo do TLS, onde são sujeitos às funções de cifra antes de serem enviados para a entidade correspondente. O processo inverso é efectuado na entidade receptora dos dados.

A camada do protocolo de registo opera de acordo com os parâmetros definidos pelo protocolo de troca de mensagens. Estes são designados por parâmetros de estado da ligação e, tal como o seu nome indica, incluem informação relativa ao estado da ligação, por exemplo o algoritmo de cifra e chaves de cifra utilizadas. O protocolo de registo pode armazenar até quatro estados da ligação: estado actual e estado pendente da transmissão dos dados; estado actual e estado pendente da recepção dos dados. O estado actual é o estado efectivo da ligação. O estado pendente é um estado com um conjunto de parâmetros que serão utilizados na próxima mudança de estado.

O TLS não utiliza cifra de chave pública para transferência de grandes volumes de dados da camada de registo. Pelo contrário, utiliza algoritmos de chave simétrica definidos entre as entidades. O protocolo de troca de mensagens recorre a certificados digitais, durante o processo de autenticação. Esta abordagem permite reduzir a carga de processamento no processo de autenticação, o que se traduz numa vantagem para utilização em dispositivos wireless.

5.4.1.2 Troca de Mensagens

A relação estabelecida entre duas entidades pelo TLS envolve a troca de uma série de mensagens entre estas, conforme ilustrado na Figura 5-27.

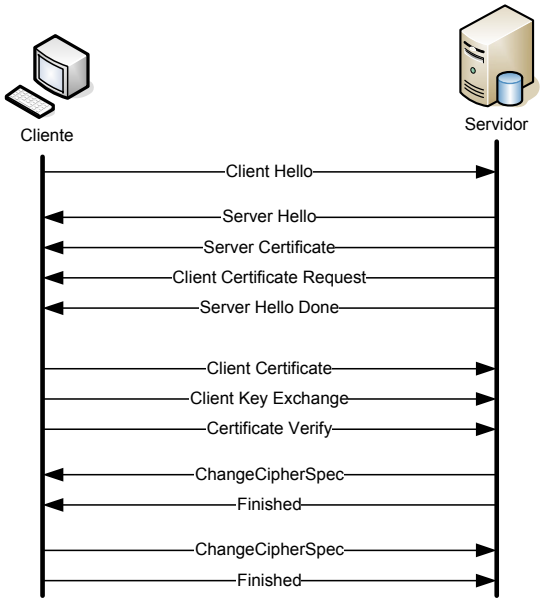


Figura 5-27 – Troca de mensagens TLS.

A primeira mensagem enviada pelo cliente é uma mensagem ***Client Hello***. Esta contém o parâmetro *CipherSuites* com um conjunto de métodos criptográficos, que definem os aspectos relacionados com os serviços de segurança a fornecer, nomeadamente, tipo de certificado digital, método de cifra e método de verificação de integridade. Esta mensagem transporta ainda um valor aleatório, designado *ClientHello.random*.

Após a recepção da mensagem ***Client Hello*** o servidor verifica se suporta os métodos criptográficos referidos pelo cliente. Em seguida responde com uma mensagem ***Server Hello***. Esta mensagem contém, entre outros, dois parâmetros relevantes. O primeiro parâmetro é um outro número aleatório, designado *ServerHello.random*. O segundo parâmetro, contém o identificador da sessão *Session ID*, utilizado por ambas as entidades para identificar a sessão estabelecida. O parâmetro *CipherSuite* identifica o método criptográfico a utilizar durante a sessão. O servidor escolhe um dos métodos apresentados no parâmetro *CipherSuites* da mensagem ***Client Hello***.

Uma das características do TLS é que após ser estabelecida uma sessão segura, esta pode ser reiniciada várias vezes pelo cliente. Para tal o cliente envia o parâmetro *Session ID* numa mensagem ***Client Hello***. Uma vantagem desta característica é, por exemplo, o envio imediato, por parte de um servidor, de uma página HTML visitada anteriormente pelo cliente, sem que tenha que ser estabelecida uma nova sessão.

Nesta fase do processo de troca de mensagens, o cliente e o servidor trocaram mensagens que permitiram sincronizar o estado das duas entidades, estabelecer um identificador de sessão, definir o conjunto de métodos criptográficos a utilizar e trocar números aleatórios.

A próxima fase envolve a troca de certificados digitais. O servidor envia o seu certificado para o cliente através da mensagem ***Server Certificate***. Esta mensagem contém informação relacionada com a chave pública do servidor. O formato exacto desta informação depende do algoritmo de chave pública utilizado. O cliente irá utilizar a chave pública do servidor para cifrar a chave de sessão, que ambos utilizarão para cifrar os dados da sessão. Após esta mensagem o servidor pede ao cliente o seu certificado digital (com informação da chave pública do cliente) através da mensagem ***Client Certificate Request***. Após estas etapas, o servidor envia uma mensagem ***Server Hello Done***, indicando ao cliente que terminou o processo de negociação inicial e fica a aguardar que o cliente tome a próxima acção.

Ao pedido de certificado digital o cliente responde com a mensagem ***Client Certificate***. Em seguida o cliente envia a mensagem ***Client Key Exchange*** para o servidor. Com esta mensagem o cliente fornece ao servidor a informação necessária para estabelecer uma comunicação segura. Com a chave pública do servidor o cliente cifra e envia na mensagem uma PMS (*Pre-Master Secret*) que será utilizada posteriormente para calcular a chave mestra MS (*Master Secret*) da qual as chaves de sessão serão obtidas. A PMS pode ser calculada com Diffie-Hellman (secção 2.5.1), ou por um valor escolhido pelo cliente. É necessário calcular 6 funções de síntese para obter o valor da MS. O valor da MS é dado por:

MS = MD5 [PMS || SHA ('A' || PMS || *ClientHello.random* || *ServerHello.random*)] ||
MD5 [PMS || SHA ('BB' || PMS || *ClientHello.random* || *ServerHello.random*)] ||
MD5 [PMS || SHA ('CCC' || PMS || *ClientHello.random* || *ServerHello.random*)]

onde 'A', 'BB' e 'CCC' são *strings*.

As chaves de sessão utilizadas para proteger a comunicação entre o cliente e o servidor são calculadas a partir da MS e dos valores aleatórios *ClientHello.random* e *ServerHello.random* através do código de autenticação de mensagens HMAC (*Keyed-Hashing for Message Authentication*) [RFC2104] e da função de síntese MD5 (*Message Digest 5*) [RFC1321].

A verificação do certificado do cliente é efectuada nesta fase do processo de troca de mensagens. O cliente prova a sua identidade submetendo todas as mensagens enviadas e

recebidas a uma função de síntese. Note-se que ambas as entidades mantêm cópias de todas as mensagens trocadas durante o processo de troca de mensagens. O resultado é enviado para o servidor através de uma mensagem ***Certificate Verify***, que é assinada com a chave privada do cliente. O servidor verifica a assinatura utilizando a chave pública do cliente. Se a assinatura corresponder então o servidor calcula a síntese das mensagens armazenadas por este e compara o resultado ao enviado pelo cliente. Quando este resultado coincide o servidor tem a garantia que está a comunicar com um cliente legítimo.

Os objectivos do processo de troca de mensagens do TLS são autenticar as entidades intervenientes na comunicação e criar um novo estado de comunicação seguro. Na fase inicial do processo não é utilizado qualquer tipo de cifra de dados. Durante esta fase são negociados os métodos de cifra para criar o novo estado da comunicação, estado seguro. Após serem negociados todos os métodos de cifra a utilizar no estado seguro este pode ser assumido pelas entidades através do envio de uma mensagem ***ChangeCipherSpec***.

Para verificar que a negociação ocorreu com sucesso o cliente e o servidor enviam uma mensagem ***Finished***. Esta mensagem contém uma síntese de todas as mensagens trocadas anteriormente (excluindo a mensagem ***Finished***). A entidade receptora pode calcular essa mesma síntese a partir das mensagens que tem armazenadas e verificar se o resultado é coincidente. Se assim for, todo o processo é validado, dando-se inicio à troca de dados cifrados, entre as entidades, com recurso aos métodos de cifra negociados.

Processo de troca de mensagens TLS no WPA e em redes RSN

O processo de troca de mensagens descrito anteriormente, cumpre três funções: (i) autenticar o servidor (e opcionalmente o cliente); (ii) gerar a chave de cifra da sessão; (iii) iniciar e efectivar a utilização de um método de cifra para protecção da comunicação.

As funcionalidades do TLS a adoptar pelo WPA e pelas redes RSN restringem-se à função de autenticação e à geração da chave de cifra (*Master Key*). Relativamente aos métodos de cifra o WPA e as redes RSN utilizam os seus próprios métodos WEP, TKIP ou AES-CCMP. Basicamente o WPA e as redes RSN utilizam a chave secreta de cifra gerada pelo TLS para derivar um conjunto de chaves de cifra para cifrar os dados de uma comunicação wireless.

O protocolo TLS integra-se perfeitamente no modelo 802.1X, tendo sido especificado para operar sobre o protocolo EAP. Este modo de funcionamento é obrigatório nas especificações do WPA.

5.4.1.3 EAP-TLS

O [RFC2716] define a troca de mensagens TLS sobre EAP. Este documento considera apenas cenários de autenticação de acesso usando PPP (*Point to Point Protocol*). Este pode no entanto ser adoptado para utilização em cenários de rede não-cablada que recorram ao IEEE 802.1X e em redes RSN.

Esta secção descreve a utilização do protocolo EAP-TLS num cenário em que é implementado o modelo IEEE 801.1X. Numa primeira fase é apresentado o formato genérico das mensagens EAP-TLS. Em seguida apresentam-se os resultados de uma experiência prática realizada em laboratório e que permite analisar a troca de mensagens durante o processo de autenticação de um cliente.

Formato de mensagens

Esta subsecção descreve sumariamente o formato das mensagens EAP-TLS ***Request*** e EAP-TLS ***Response*** trocadas durante a fase de autenticação.

O formato genérico de um pacote EAP-TLS é idêntico ao da Figura 5-16. No entanto o valor do campo ***Type*** é fixo e tem o valor 13, de acordo com a Tabela 5-2. O formato de um pacote EAP-TLS ***Request*** ou ***Response*** é representado na Figura 5-28.

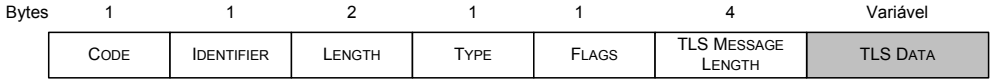


Figura 5-28 – Pacote EAP-TLS Request / Response.

Os quatro primeiros campos ***Code***, ***Identifier***, ***Length*** e ***Type*** que compõem o pacote EAP-TLS ***Request*** têm a mesma função que os campos descritos na secção 5.3.2.

O campo ***Flags*** ilustrado na Figura 5-29 é composto por oito bits, onde os três bits mais significativos têm as seguintes funcionalidades:

- O bit L (*Length included*) é colocado a um para indicar a presença do campo ***TLS Message Length***.
- O bit M (*More Fragments*) é colocado a um sempre que o pacote seja um fragmento de uma mensagem EAP-TLS.
- O bit S (*EAP-TLS Start*) é colocado a um sempre que a mensagem EAP-TLS é do tipo *Start*.

L	M	S	R	R	R	R	R

Figura 5-29 – Campo Flags do pacote EAP-TLS Request.

Os restantes bits, R, estão reservados para uso futuro.

O campo ***TLS Message Length*** tem tamanho de quatro bytes e representa o tamanho total da mensagem EAP-TLS. Actualmente este campo é opcional e normalmente não é incluído se os dados EAP-TLS não necessitarem de fragmentação. O campo ***TLS Data*** é constituído pelos dados no formato definido pelo protocolo TLS.

Mensagens trocadas na fase de autenticação

A troca de mensagens na fase de autenticação utilizando o protocolo EAP-TLS é ilustrada na Figura 5-30.

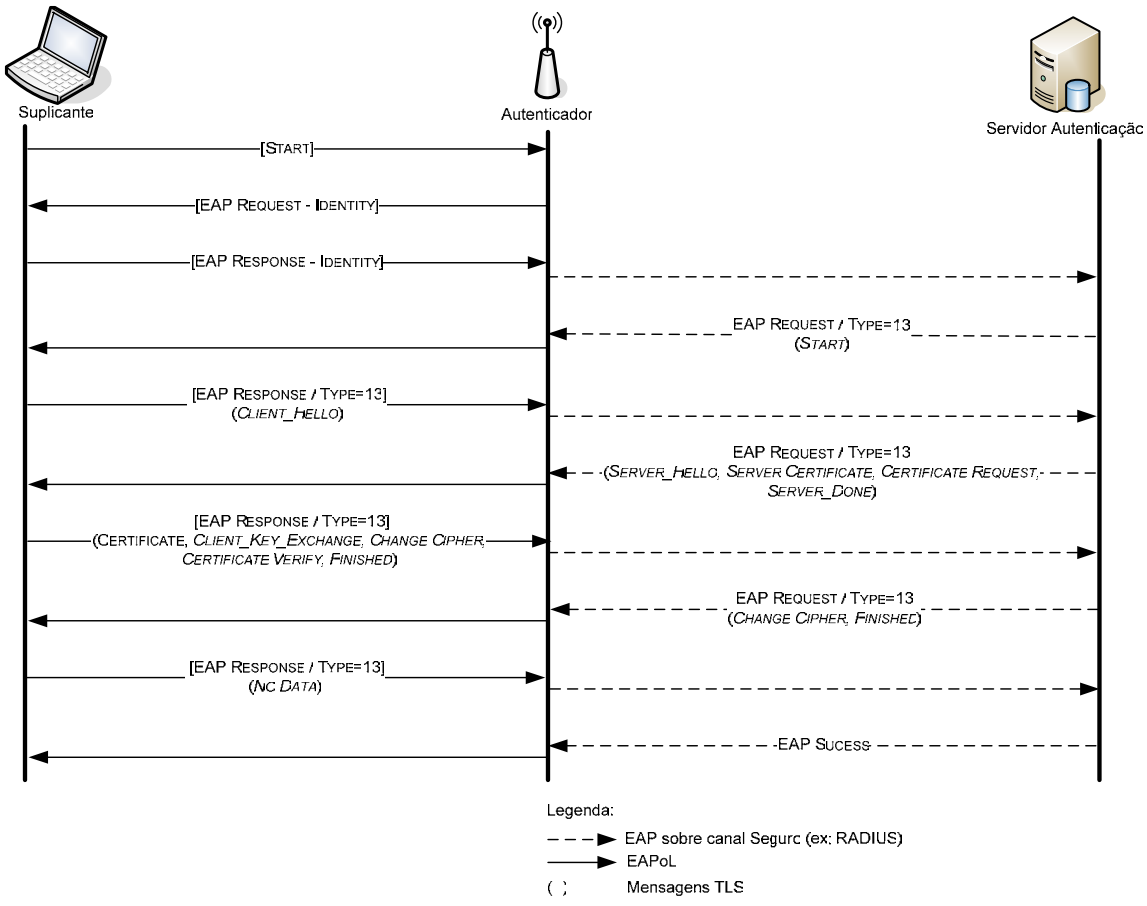


Figura 5-30 – Troca de mensagens EAP-TLS.

O pedido de acesso à rede é tipicamente iniciado pela estação cliente através do envio de um pacote ***EAPoL - Start***. Após este pedido, o cliente transmite para o servidor de autenticação a indicação que possui as credenciais necessárias à sua autenticação na rede.

Numa fase posterior, após o envio das credenciais por parte da estação cliente, o servidor procede à verificação destas. Para tal consulta as suas políticas de acesso e em seguida garante ou nega autorização ao cliente.

Se a estação cliente estiver autorizada, é permitido o acesso. Se o cliente não for autorizado, a comunicação é interrompida. Se o processo de troca de mensagens EAP-TLS terminar com sucesso é transmitida uma mensagem do tipo ***EAP Success***; caso ocorra alguma falha é enviada uma mensagem ***EAP Fail***.

Experiência Prática- Análise de mensagens trocadas na fase de autenticação EAP-TLS

A experiência prática efectuada visa estudar na prática o funcionamento do protocolo EAP-TLS aplicado ao modelo IEEE 802.1X.

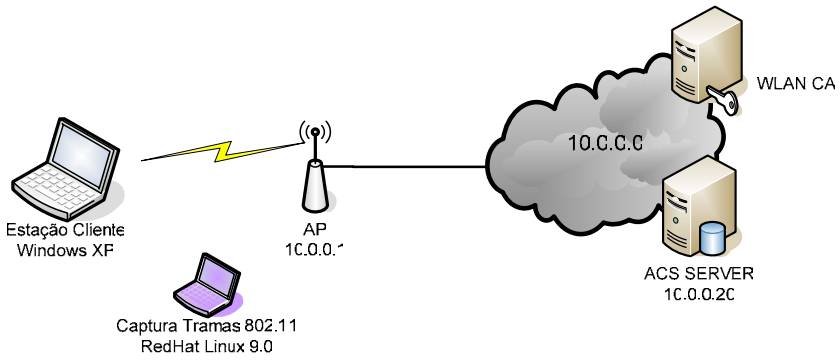


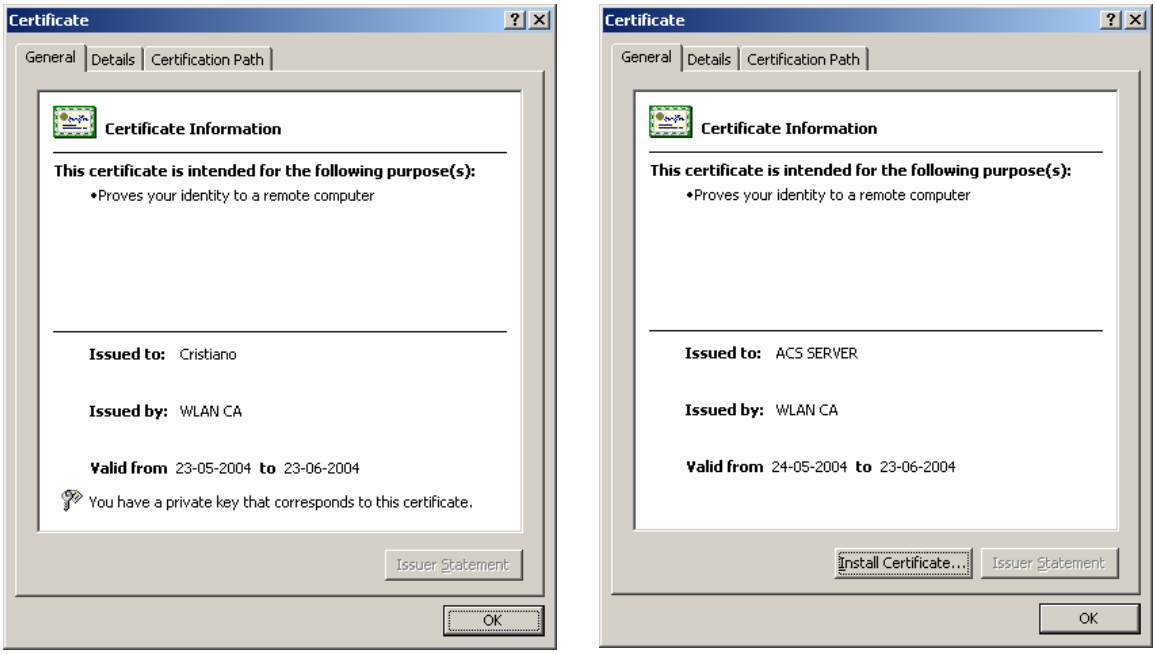
Figura 5-31 – Configuração utilizada em laboratório.

Antes de descrever a experiência prática baseada no protocolo EAP-TLS, importa referir que o cenário implementado em laboratório apresentado na Figura 5-31, serve de base a várias experiências realizadas no âmbito do estudo dos protocolos de autenticação abordados nesta dissertação. As diferenças nas respectivas configurações são referidas sempre que necessário.

A estação cliente possui uma placa PCMCIA Cisco Aironet PCM-350 com endereço MAC 00:0B:46:56:30:BB. O *software* que implementa o suplicante na estação cliente é fornecido nativamente pelo sistema operativo Windows XP. A máquina responsável pela captura de tramas IEEE 802.11 está a executar a ferramenta *Kismet*, versão 2004-04-R1, e com sistema operativo RedHat Linux 9.0. O ponto de acesso utilizado é um Cisco Aironet 1100 com

endereço MAC igual a 00:02:8A:78:B6:F0 e configurado com endereço IP 10.0.0.1. Para o servidor de autenticação a opção recaiu no *software* comercializado pela empresa *Cisco System* ACS Server versão 3.2 para Windows. O servidor de autenticação está a ser executado num PC com sistema operativo Windows 2000 Server. Este servidor foi configurado com endereço IP 10.0.0.20. O último componente representado no cenário da Figura 5-31 é um servidor designado por WLAN CA e cuja única funcionalidade é fornecer o serviço de entidade certificadora, responsável por emitir os certificados digitais a cada uma das entidades, cliente e servidor de autenticação, intervenientes no processo de autenticação.

A Figura 5-32 apresenta os certificados emitidos pela CA ao cliente e ao servidor de autenticação. A informação contida nestes certificados digitais é descrita na secção 2.5.2.



a) Certificado digital do cliente.

b) Certificado digital do servidor autenticação.

Figura 5-32 – Certificados Digitais.

Nesta experiência configurou-se o utilizador "cristiano" na estação cliente. As credenciais do utilizador para se autenticar no servidor de autenticação estão no certificado digital, emitido pela entidade certificadora WLAN CA. Procedeu-se à configuração das políticas de acesso para este cliente, no servidor de autenticação, ACS SERVER. As políticas de acesso definidas permitem o acesso aos recursos da rede após a autenticação com sucesso do cliente. Finalmente configurou-se o ponto de acesso de modo a suportar o mecanismo de controlo de acesso IEEE 802.1X.

Pretende-se observar o processo de autenticação quando este ocorre com sucesso. Para tal efectuaram-se capturas das mensagens trocadas, nomeadamente tramas IEEE 802.11 entre o ponto de acesso e a estação cliente (comunicação wireless) e pacotes RADIUS entre o servidor de autenticação e o ponto de acesso (comunicação *wired*) representados respectivamente na Figura 5-33 e Figura 5-34. Uma vez que as mensagens capturadas no meio físico wireless e cablado são idênticas, apenas serão apresentadas mensagens transmitidas no meio wireless.

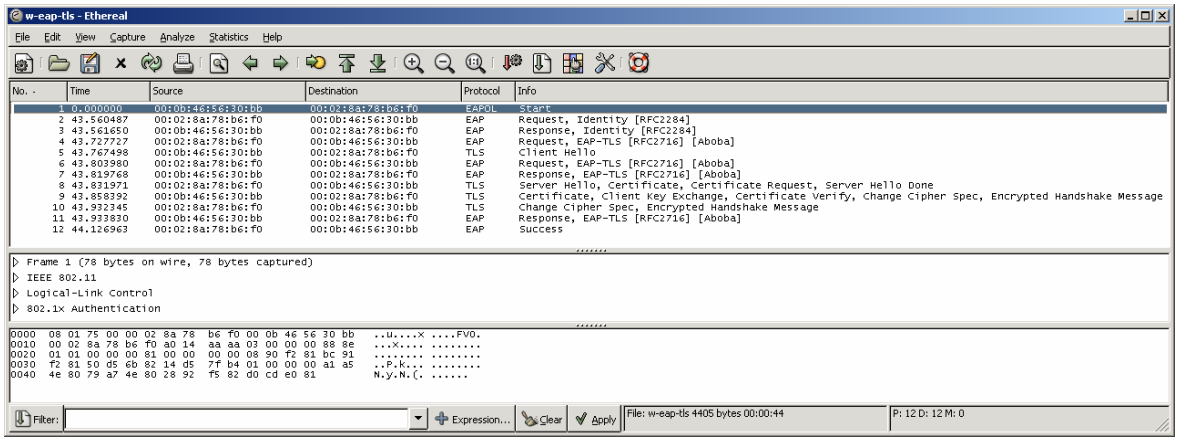


Figura 5-33 – Comunicação wireless – autenticação EAP-TLS

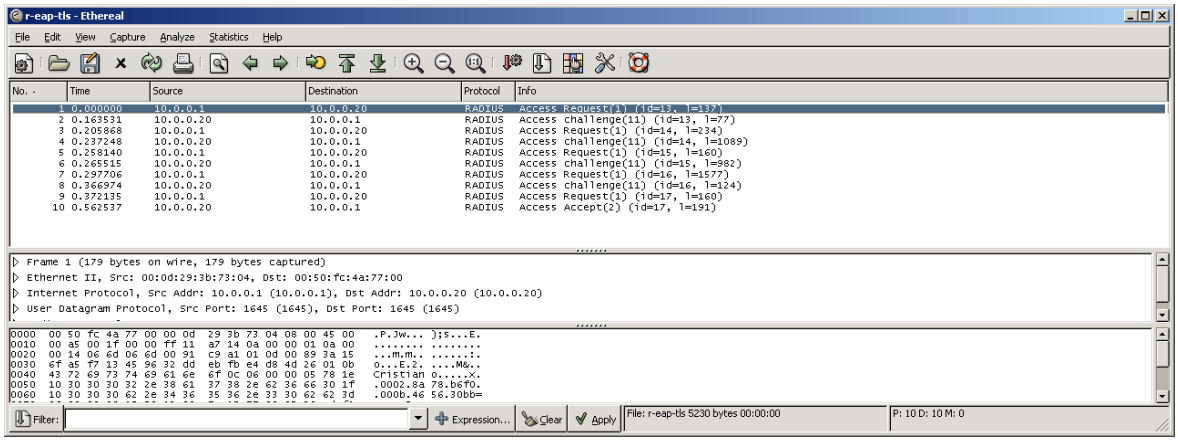


Figura 5-34 – Comunicação wired – autenticação EAP-TLS.

Passando a uma análise mais detalhada de algumas das mensagens capturadas, podemos reparar que após o pedido de acesso à rede wireless iniciado pela estação cliente segue-se a fase de troca de informação relativa ao método EAP a utilizar pelas entidades intervenientes no processo de autenticação.

O ponto de acesso envia para a estação cliente um pedido de identificação através de um pacote ***EAP Request*** (Figura 5-35) em que o campo ***Type*** tem valor igual a um (*Identity*) de acordo com a Tabela 5-2

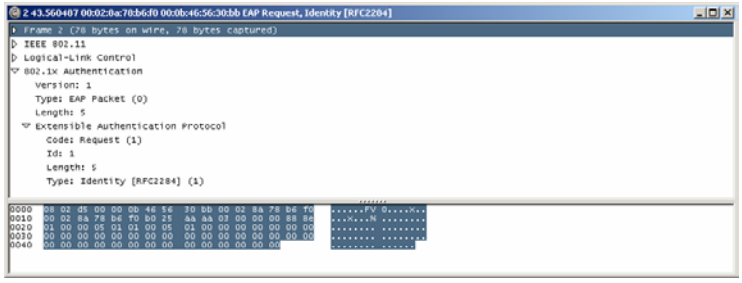


Figura 5-35 – Pacote EAP-Request enviado pelo ponto de acesso.

Em resposta ao pedido de identificação a estação cliente responde com um pacote ***EAP Response*** (Figura 5-36) onde é enviada a identidade do utilizador. Observa-se que o campo ***Identity*** possui o valor "cristiano". Esta informação pode identificar o certificado de cliente que vai ser enviado posteriormente. Se o cliente não pretende enviar um certificado, então será como um cliente anónimo, e aqui qualquer informação pode ser enviada, tal como uma *string Anonymous*.

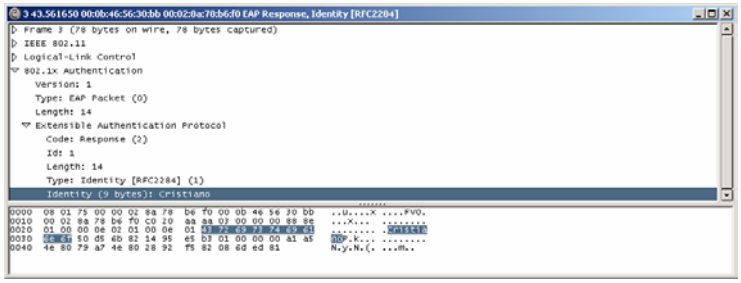


Figura 5-36 – Pacote EAP-Response enviado pela estação cliente.

Quando o servidor de autenticação receber a identidade do utilizador deve responder com um pacote ***EAP-Request*** (Figura 5-37). O campo ***Type*** identifica o método de autenticação suportado pelo EAP a utilizar. Neste caso possui o valor 13, que refere o método EAP-TLS de acordo com Tabela 5-2. Verifica-se ainda que o bit S (*start*) está activo. Esta é a única mensagem trocada em que o valor do bit S é um.

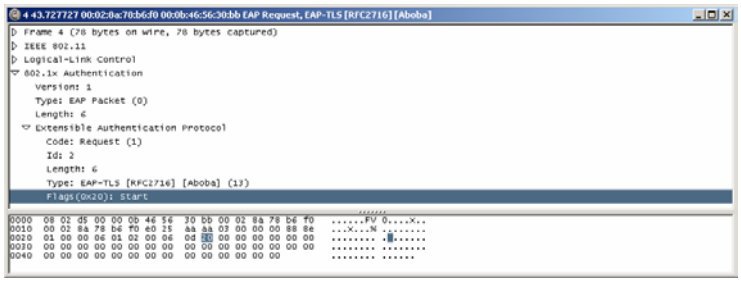


Figura 5-37 – Pacote EAP-Request enviado pelo servidor de autenticação.

Nesta fase dá-se inicio à conversação EAP-TLS. A estação cliente envia um pacote ***EAP-Response*** (Figura 5-38). Este pacote transporta uma mensagem ***Client Hello***, que contém a mesma informação que uma mensagem TLS normal. Para uma breve descrição do seu conteúdo consultar secção 5.4.1.

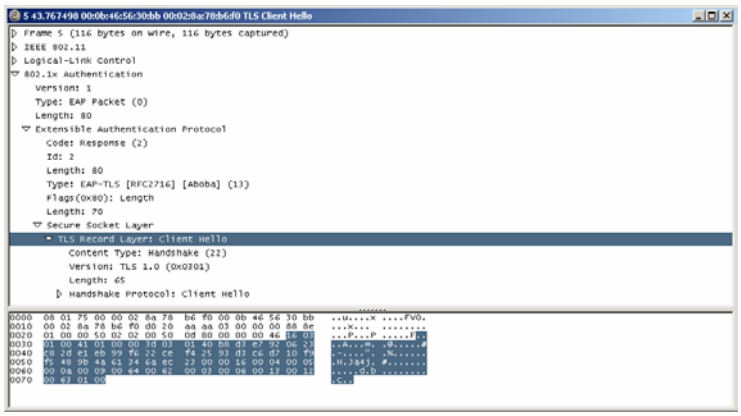


Figura 5-38 – Mensagem Client Hello enviada pela estação cliente.

A resposta do servidor de autenticação é dada através de uma mensagem ***EAP-Request*** (Figura 5-39). Normalmente o servidor envia várias mensagens TLS numa única, nomeadamente, a mensagem ***Server Hello***, o seu certificado através da mensagem ***Server Certificate***, o pedido de certificado do cliente, ***Certificate Request*** e a finalizar uma mensagem ***Server Hello Done***. De referir que o conteúdo desta mensagem é fragmentado nas tramas 6 e 8 conforme indicado na Figura 5-33.

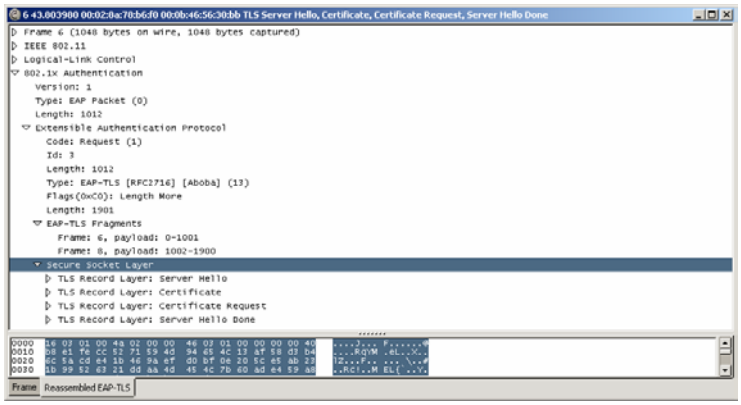


Figura 5-39 - Mensagens TLS enviada pelo servidor de autenticação

Antes de analisar a resposta da estação cliente à mensagem ***EAP-Request*** enviada pelo servidor de autenticação, apresenta-se alguma informação relativa ao certificado digital do servidor de autenticação. Uma breve análise à Figura 5-40 permite determinar imediatamente o nome da entidade certificadora que emitiu o certificado que, como referido anteriormente, se designa WLAN CA. O proprietário do certificado também é identificado facilmente, no caso o nome dado ao servidor de autenticação, ACS SERVER .

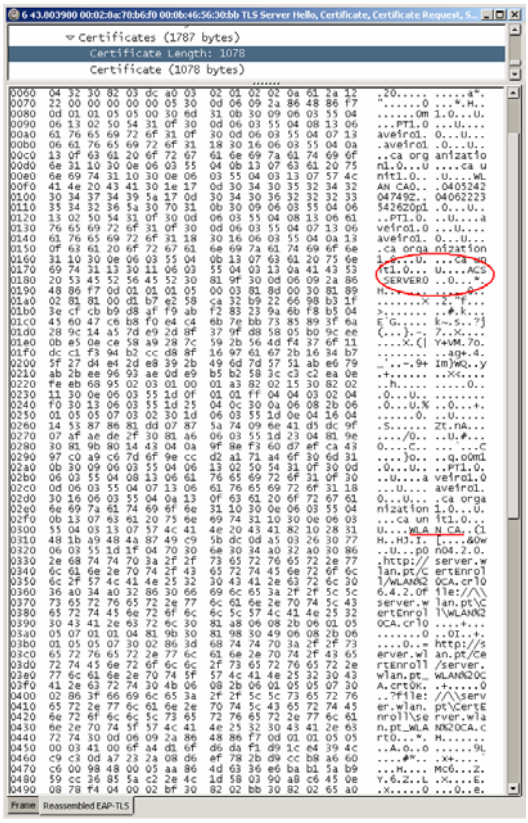


Figura 5-40 – Certificado digital do servidor de autenticação.

A resposta da estação cliente é dada através de um pacote do tipo ***EAP-Response*** (Figura 5-41). Este pacote transporta um conjunto de mensagens TLS, nomeadamente, a mensagem ***Client Certificate*** enviada em resposta ao pedido do servidor, ***Client Key Exchange*** que transporta o valor da PMS cifrado com a chave pública do servidor por forma a que este possa gerar as chaves de sessão a utilizar no estado seguro (secção 5.4.1.2) e a mensagem ***Certificate Verify*** com o resultado de uma função de síntese de todas as mensagens recebidas e enviadas. Esta mensagem é assinada com a chave privada do cliente o que permite ao servidor ter a garantia que está a comunicar com um cliente legítimo.

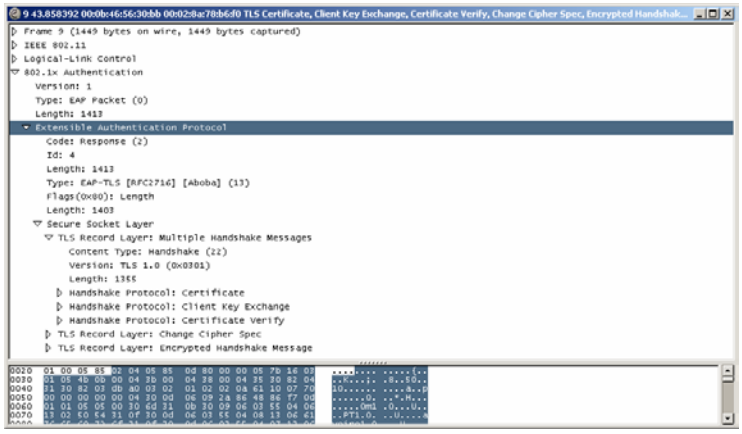


Figura 5-41 – Mensagens TLS enviadas pela estação cliente.

São ainda trocadas mais algumas mensagens TLS entre o servidor de autenticação e a estação cliente, referentes à troca de material criptográfico de modo a efectivar a utilização de um método de cifra para proteger a comunicação. Estes aspectos não são relevantes para o processo de autenticação abordado nesta secção.

Se a autenticação do servidor ocorrer com sucesso, a estação cliente deve enviar um pacote ***EAP-Response*** com o campo ***Type*** igual a 13 e sem quaisquer outros dados. (Figura 5-42)

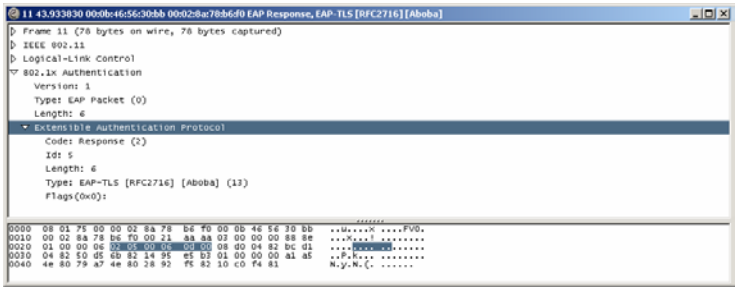


Figura 5-42 – EAP-Response enviado pela estação cliente.

O servidor de autenticação responde então com um pacote do tipo ***EAP-Success*** (Figura 5-43). Esta última mensagem indica o sucesso do processo de autenticação mútua entre a estação cliente e o servidor de autenticação.

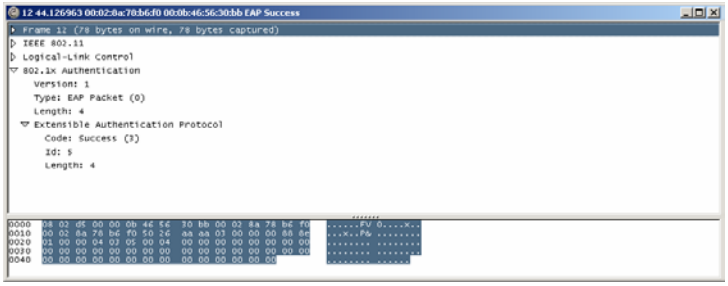


Figura 5-43 – EAP Success enviado pelo servidor de autenticação.

A experiência apresentada nesta secção descreve o funcionamento do protocolo EAP-TLS. Este não foi especificamente desenvolvido para ser utilizado em redes wireless. É um protocolo baseado no SSL (*Secure Sockets Layer*), um mecanismo de segurança desenvolvido para operar ao nível da aplicação. No entanto, o desenvolvimento de um método que suporta TLS sobre EAP, combinado com algumas alterações no protocolo RADIUS para suportar EAP sobre RADIUS, permitiu a utilização destes em redes wireless.

Com esta experiência ficou demonstrada a utilidade do EAP-TLS como mecanismo de autenticação, capaz de fornecer autenticação mútua em redes wireless.

5.4.2 LEAP

Um dos protocolos de autenticação baseados no protocolo EAP que é mais implementado é o LEAP, apesar da sua condição de ser um protocolo proprietário da empresa *Cisco Systems, Inc.* A análise a este protocolo justifica-se nesta dissertação pelo facto de ter sido efectivamente a primeira implementação comercial do modelo IEEE 802.1X e EAP em redes não-cabladas.

O LEAP não é um protocolo normalizado e os principais detalhes relativos ao seu funcionamento não foram publicados. No entanto o protocolo foi analisado e tornado público em [23], permitindo a outros fabricantes a implementação de dispositivos compatíveis com este protocolo.

De forma consistente com o modelo IEEE 802.1X, o LEAP divide um sistema em três entidades, suplicante, autenticador e servidor de autenticação. À data do desenvolvimento do LEAP os sistemas operativos existentes não suportavam IEEE 802.1X, pelo que era

necessária a sua actualização através de *drivers* e *software* específicos de modo a tornar operacionais as funcionalidades do modelo IEEE 802.1X.

Apenas os pontos de acesso comercializados pela empresa Cisco suportam LEAP. De referir que um autenticador 802.1X genérico não é suficiente dada a forma como as chaves de cifra são administradas. O ponto de acesso tem que ter suporte específico para LEAP assim como para IEEE 802.1X. O servidor de autenticação é implementado por um servidor RADIUS. O LEAP utiliza atributos RADIUS proprietários para receber as chaves de cifra do servidor.

A Figura 5-44 apresenta as mensagens trocadas entre o servidor de autenticação e a estação cliente.

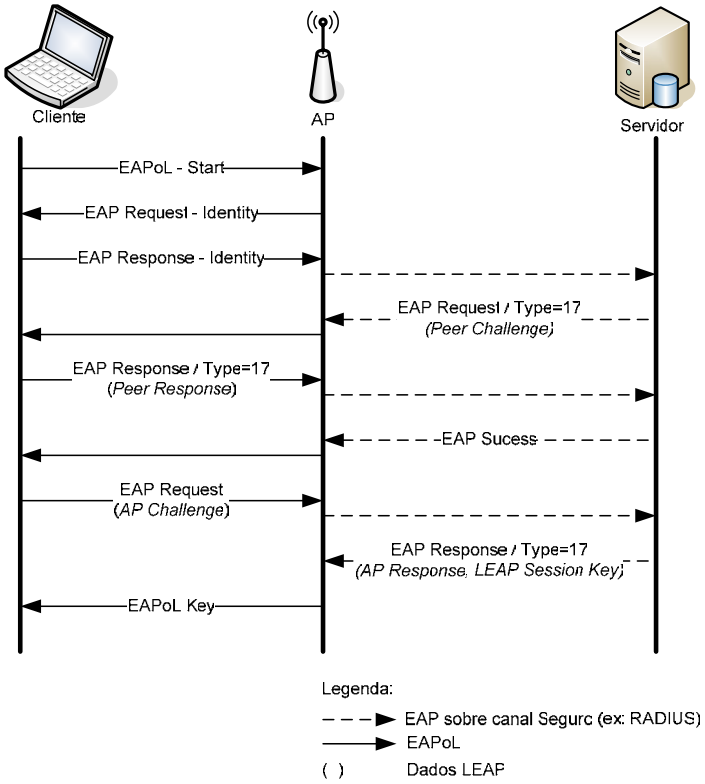


Figura 5-44 – Processo de autenticação LEAP.

O LEAP é um protocolo do tipo pergunta-resposta entre as duas entidades intervenientes no processo de autenticação, cliente e servidor de autenticação, com base numa chave secreta partilhada. O LEAP é baseado no protocolo de autenticação MS-CHAPv1 [RFC2433] mas ao contrário deste o LEAP fornece autenticação mútua através de desafios distintos, *Peer challenge* e *AP challenge*, enviados respectivamente pelo servidor e pelo cliente.

Uma vez concluída a autenticação mútua das entidades, o servidor de autenticação e o cliente calculam uma chave de sessão. Esta chave é distribuída do servidor para o ponto de acesso através de um atributo RADIUS da Cisco designado por *Leap:session-key*. Este atributo é cifrado utilizando a chave secreta partilhada configurada entre o ponto de acesso e o servidor de autenticação.

De acordo com [23] o valor da chave de sessão é obtido através do mesmo algoritmo utilizado para cifrar o atributo *MS-MPPE-Send-Key*. Este algoritmo é descrito no [RFC2548]. Genericamente tem-se que:

$$Chave\ de\ sessão = f\{SS, Auth, MD5(MPPEHash||APC||APR||PC||PR)\}$$

onde:

- SS (*Shared Secret*) – Chave secreta configurada no ponto de acesso e no servidor de autenticação. Esta chave garante ao servidor de autenticação que está a comunicar com um ponto de acesso legítimo.
- Auth – Atributo RADIUS *Message-Authenticator*.
- MPPEHash – Função de Síntese da Microsoft *NtPasswordHash* [RFC2759].
- APC (*Access Point Challenge*) – Desafio enviado pelo cliente ao servidor de autenticação; é um valor aleatório de 8 bytes.
- APR (*Access Point Response*) – Resposta enviada pelo servidor ao desafio do cliente; é o resultado da função *ChallengeResponse (APC, MPPEHash)* [RFC2759].
- PC (*Peer Challenge*) – Desafio enviado pelo servidor de autenticação ao cliente; é um valor aleatório de 8 bytes.
- PR (*Peer Response*) – Resposta enviada pelo cliente ao desafio do servidor; é o resultado da função *NTChallengeResponse (PC, Password)* [RFC2759].

O símbolo || representa a função de concatenação. A função *f* representa o algoritmo descrito em [RFC2548].

A mensagem final ***EAPoL Key*** enviada pelo ponto de acesso para o cliente não contém o valor da chave de sessão, apenas serve para notificar o cliente que deve activar a cifra de dados a partir desse instante.

Experiência Prática- Análise de mensagens trocadas na fase de autenticação LEAP

A experiência prática efectuada visa estudar na prática o funcionamento do protocolo proprietário LEAP. O cenário implementado para esta experiência é o representado na Figura 5-31.

Todos os componentes descritos anteriormente foram utilizados nesta experiência, à excepção do *software* que implementa o suplicante. No caso recorreu-se a um conjunto de *software* fornecido pela empresa *Cisco System Inc*, composto por diversos componentes nomeadamente: (i) Cisco Aironet Client Adapter Installation Wizard version 1.0 - 802.11b; (ii) Windows NDIS Driver 8.3; (iii) Aironet Client Utility (ACU) 6.0.

Inicialmente configurou-se o utilizador "cristiano" na estação cliente, com auxílio da ferramenta ACU. As credenciais do utilizador são fornecidas por nome de utilizador/senha. O seu valor é definido previamente na base de dados do servidor de autenticação ACS SERVER. O ponto de acesso foi configurado de modo a suportar mecanismos de controlo de acesso IEEE 802.1X. Configurou-se ainda uma chave secreta (*Shared Secret*) comum entre o ponto de acesso e o servidor de autenticação. Esta chave garante que o ponto de acesso é um cliente RADIUS legítimo do servidor de autenticação. Finalmente procedeu-se à configuração das políticas de acesso no servidor de autenticação. A política definida garante o acesso aos recursos da rede no caso do utilizador "cristiano" se autenticar com sucesso.

De seguida é apresentado em detalhe todo o processo de autenticação. São apresentadas mensagens capturadas no meio wireless e no meio cablado.

Na fase inicial, a informação é transferida entre o cliente e o ponto de acesso. A estação cliente envia um pacote ***EAPoL Start*** para dar inicio ao processo de autenticação. Em resposta é enviado um pacote ***EAP-Request Identity*** para o cliente com um pedido de identificação do utilizador que pretende autenticação.

A resposta do cliente ao pedido anterior é dada através do envio de um pacote ***EAP-Response*** (Figura 5-45). A estação cliente responde com a identidade do utilizador previamente configurado no caso "cristiano".

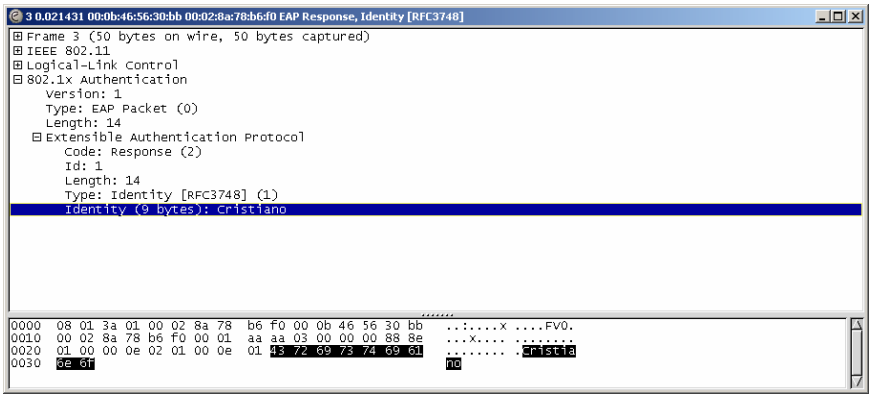


Figura 5-45 – Mensagem EAP Response enviada pela estação cliente.

Em resposta à identidade do utilizador enviada pela estação cliente, o servidor de autenticação envia uma mensagem ***EAP Request*** com o desafio (*Peer Challenge*) para a estação cliente. (Figura 5-46)

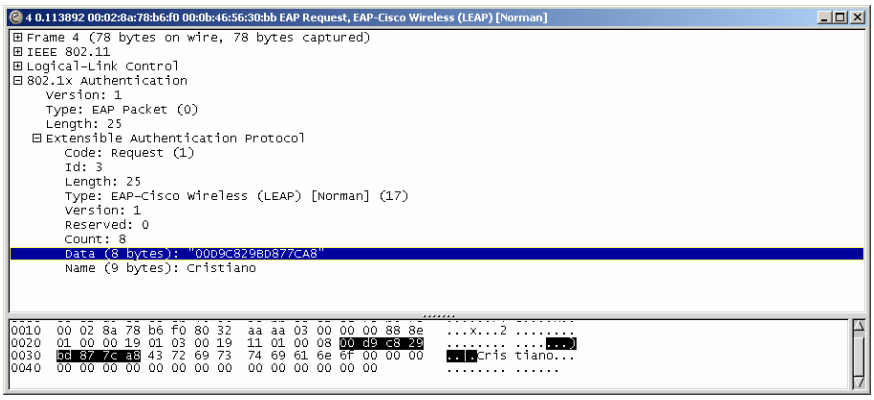


Figura 5-46 – Mensagem com o desafio enviada pelo servidor ao cliente.

A resposta do cliente (*Peer Response*) ao desafio enviado pelo servidor é dada através da mensagem ***EAP Response*** (Figura 5-47).

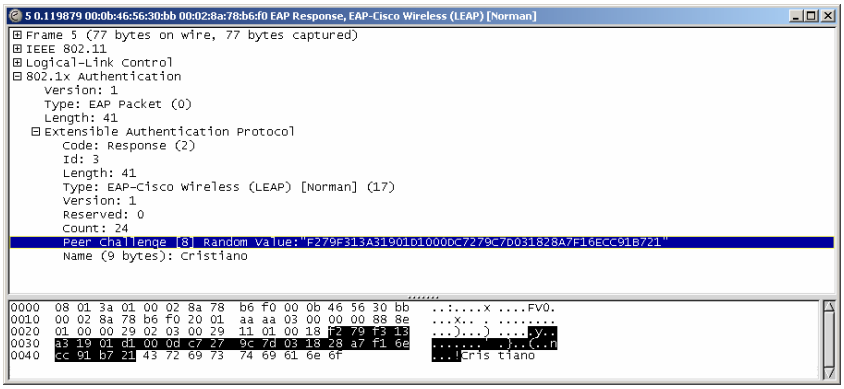


Figura 5-47 – Resposta do cliente ao desafio do servidor.

Se o cliente for considerado inválido, o servidor de autenticação enviará um pacote RADIUS contendo um pacote ***EAP Fail***. Para esta experiência o utilizador é válido e desse modo o servidor envia um ***EAP Success*** (Figura 5-48).

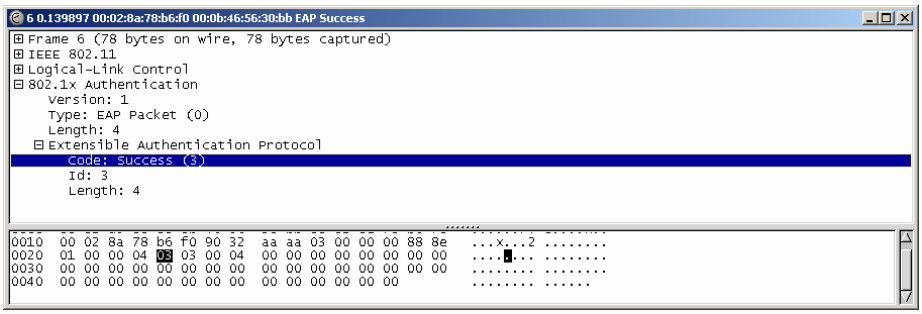


Figura 5-48 – Mensagem EAP Success enviado pelo servidor de autenticação.

Como referido anteriormente o LEAP fornece autenticação mútua. Nesta fase o cliente já se autenticou perante o servidor de autenticação. A fase seguinte é a autenticação do servidor perante o cliente. Para tal o cliente envia para o servidor um desafio (*AP Challenge*) através de uma mensagem ***EAP Request***. (Figura 5-49)

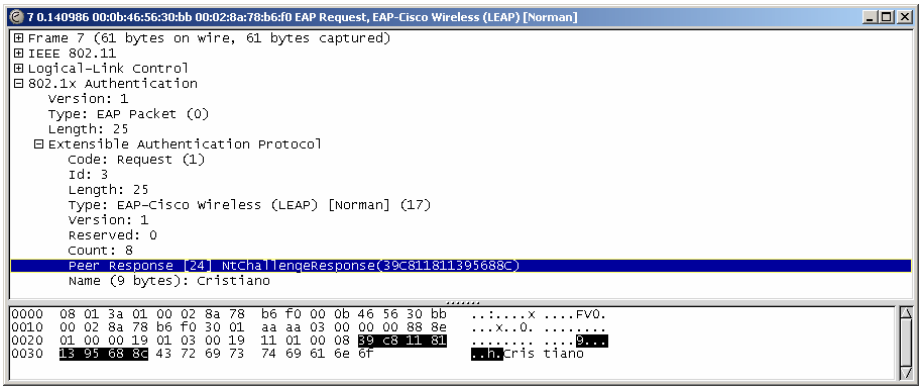


Figura 5-49 – Pacote Radius Access Challenge enviado pelo ponto de acesso.

Por fim o servidor de autenticação envia para o ponto de acesso um pacote ***RADIUS-Access-Accept*** (Figura 5-50). Esta mensagem contém a resposta (*AP Response*) ao desafio enviado anteriormente pelo cliente. Esta mensagem transporta ainda o valor da chave de sessão através do atributo proprietário da Cisco designado *Leap:session-key*. A chave de sessão é cifrada com a chave (SS) configurada entre o servidor de autenticação e o ponto de acesso.

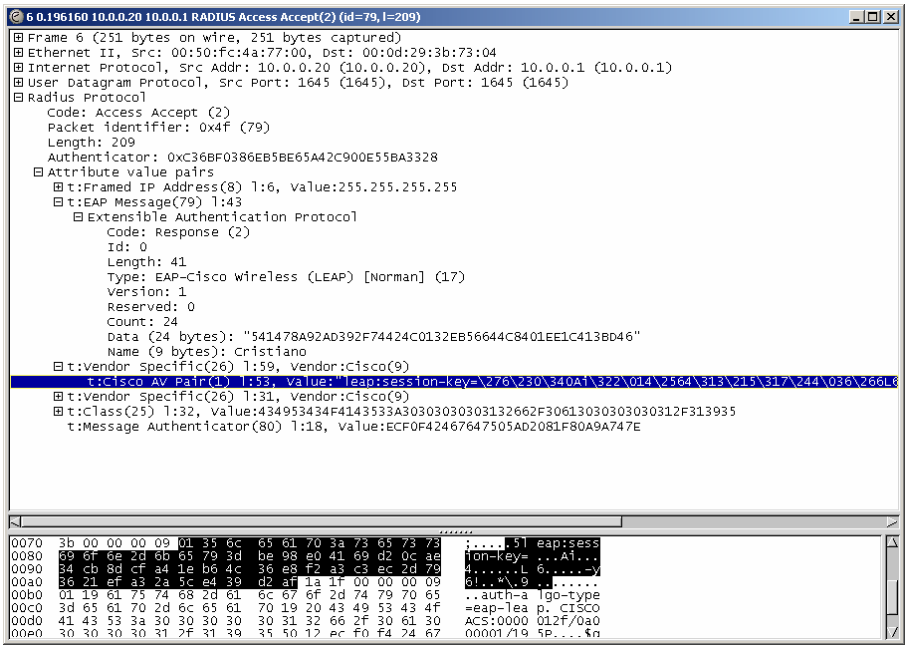


Figura 5-50 – Pacote Radius Access Accept enviado pelo servidor de autenticação.

Quando o ponto de acesso recebe do servidor de autenticação o valor da chave de sessão, envia uma mensagem ***EAPoL Key*** (Figura 5-51) para a estação cliente. Esta mensagem informa a estação cliente do índice e tamanho da chave a utilizar na comunicação com o ponto de acesso.

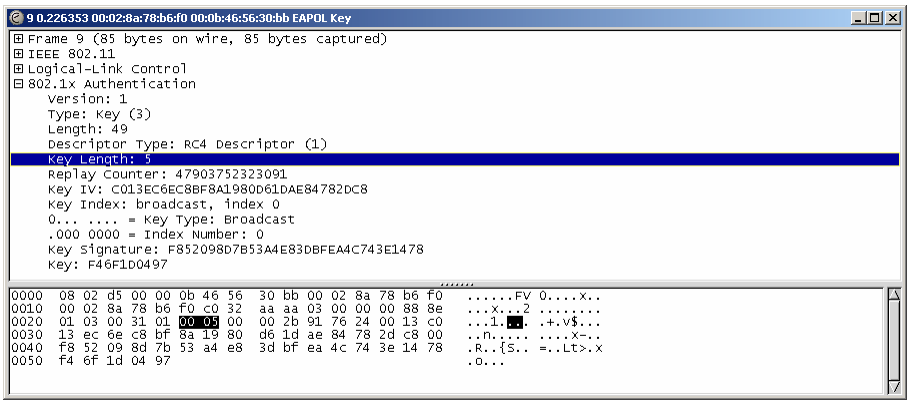


Figura 5-51 – Mensagem EAPoL Key enviada pelo ponto de acesso.

Apesar do LEAP apresentar uma série de melhorias de segurança, aquando do seu desenvolvimento, nomeadamente, autenticação mútua, chaves de sessão temporárias e gestão centralizada de chaves, não é um mecanismo de autenticação isento de vulnerabilidades. O LEAP utiliza o protocolo MS-CHAPv1, o qual é conhecido por ser vulnerável a alguns ataques de dicionário [24]. Não é objectivo desta secção descrever as vulnerabilidades

inerentes a este protocolo. Uma descrição detalhada destas vulnerabilidades pode ser encontrada em [25] [26].

5.4.3 PEAP

O protocolo PEAP (*Protected EAP*) é mais um protocolo de autenticação suportado pelo EAP. Foi desenvolvido em parceria pelas empresas *Microsoft*, *Cisco* e *RSA Security*. Este método de autenticação encontra-se actualmente em fase de desenvolvimento tendo sido proposto como norma ao IETF [27].

O PEAP pretende solucionar uma série de aspectos de segurança, nomeadamente protecção contra ataques de dicionário em mecanismos de autenticação baseada em senha, como é o caso do LEAP ou do EAP-MD5. Um outro aspecto que este protocolo pretende solucionar é a vulnerabilidade associada à componente comum a todos os métodos do protocolo EAP, ou seja à fase de identificação do cliente e às mensagens finais ***EAP-Success*** e ***EAP-Fail***. Estas duas componentes introduzem as seguintes falhas de segurança:

- A mensagem ***EAP-Identity*** transmitida na fase inicial não é protegida, o que permite a um atacante obter informação relacionada com a identidade do utilizador para utilizar em ataques posteriores.
- As mensagens ***EAP-Success*** e ***EAP-Fail*** não são protegidas e podem ser forjadas por um atacante.

A protecção destas ameaças é alcançada com o PEAP através do estabelecimento inicial de um túnel TLS para a troca de mensagens EAP. Ao contrário do método EAP-TLS, em que a presença de uma infra-estrutura de chave pública (PKI) para gestão de certificados digitais dos clientes e do servidor de autenticação é um requisito, o PEAP permite a autenticação do cliente recorrendo a outros métodos de autenticação não mútua e que não exigem a utilização de certificados digitais, como por exemplo EAP-GTC, EAP-OTP, EAP-MSCHAP-v2 ou ainda o método EAP MD5 para autenticação baseada em senha.

A utilização de certificados digitais é apenas obrigatória na autenticação do servidor de autenticação, uma vez que esta é baseada em EAP-TLS. Esta característica é uma mais valia para cenários de aplicação wireless em que a implementação e gestão de certificados digitais nas estações cliente se torna uma tarefa complexa.

Tal como referido anteriormente, o objectivo do EAP é fornecer autenticação. O objectivo do PEAP, tal como o seu nome indica é fornecer autenticação de forma privada. Para alcançar este objectivo, em primeiro lugar é estabelecida a componente de privacidade sem autenticação; em seguida a autenticação é efectuada através da ligação privada. Ou seja, o protocolo PEAP compreende duas fases distintas:

- Na primeira fase ou fase 1, recorre ao EAP de forma convencional de modo a estabelecer uma ligação segura, através de uma sessão TLS. Nesta fase apenas o servidor de autenticação é autenticado perante o cliente. A restante troca de mensagens é então cifrada com a chave negociada previamente.
- A fase 2 ocorre assumindo que a sessão TLS foi estabelecida. Esta sessão segura é utilizada para completar uma nova troca de mensagens EAP onde o cliente é autenticado. O método de autenticação EAP utilizado nesta fase é negociado entre a estação cliente e o servidor de autenticação.

Apresenta-se na Figura 5-52 a troca de mensagens durante a fase de autenticação recorrendo ao protocolo PEAP.

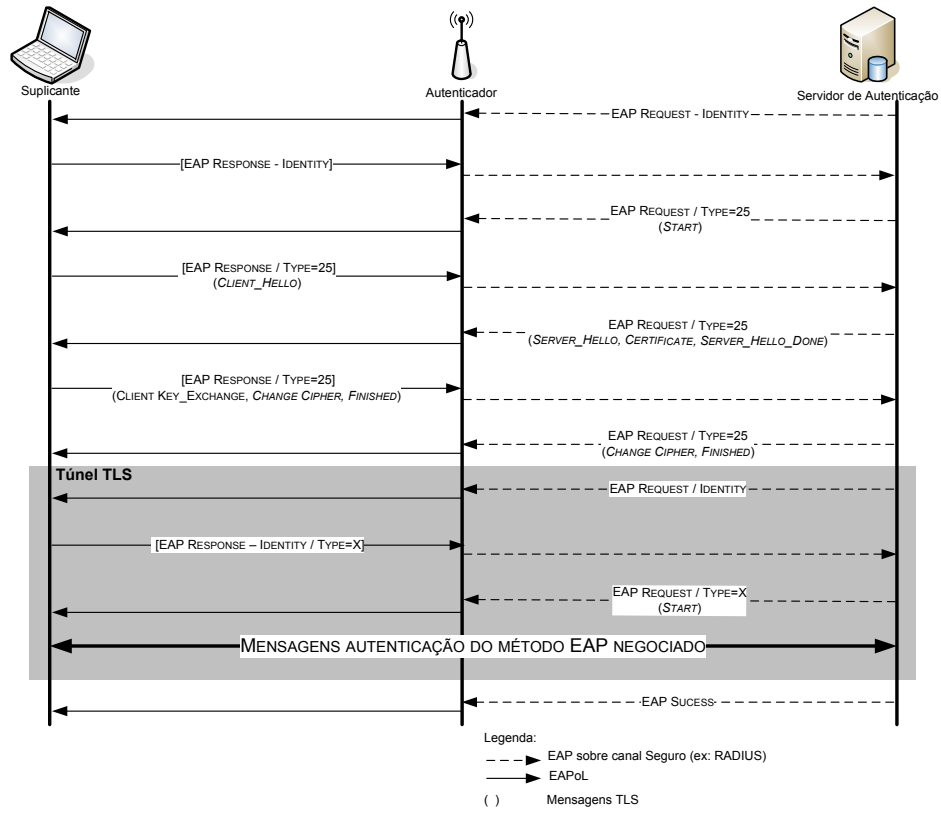


Figura 5-52 – Fase de autenticação PEAP.

Experiência Prática- Análise de mensagens trocadas na fase de autenticação PEAP

A experiência prática efectuada visa estudar na prática o funcionamento do protocolo PEAP. O cenário implementado para esta experiência é representado na Figura 5-53.

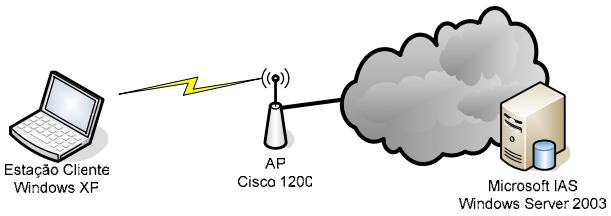


Figura 5-53 – Cenário de laboratório.

A estação cliente possui um dispositivo wireless Dell TrueMobile 1300 WLAN com endereço MAC 00:90:4B:25:A3:D4. O *software* que implementa o suplicante na estação cliente é fornecido nativamente pelo sistema operativo Windows XP. A captura das tramas de autenticação foram efectuadas com o *software* Ethereal versão 0.10.6. O ponto de acesso utilizado é um Cisco Aironet 1200 com endereço MAC igual a 00:0E:84:AB:0D:60. O servidor de autenticação utilizado para esta experiência prática é o servidor RADIUS comercializado pela empresa Microsoft, o IAS (*Internet Acess Server*). O PC que está a executar este *software* tem instalado o sistema operativo Microsoft Windows Server 2003. O certificado utilizado pelo servidor de autenticação foi emitido por uma CA designada "CICUA". O nome do servidor de autenticação é "galadriel.servers.ua.pt".

Esta experiência é baseada no protocolo PEAP-MSCHAPv2 [53]. Inicialmente configurou-se na estação cliente o utilizador "cpereira". As credenciais do utilizador são fornecidas por nome de utilizador/senha. O seu valor é também definido na base de dados do servidor de autenticação. Finalmente procedeu-se à configuração das políticas de acesso no servidor de autenticação.

A Figura 5-54 representa as mensagens capturadas no meio wireless.

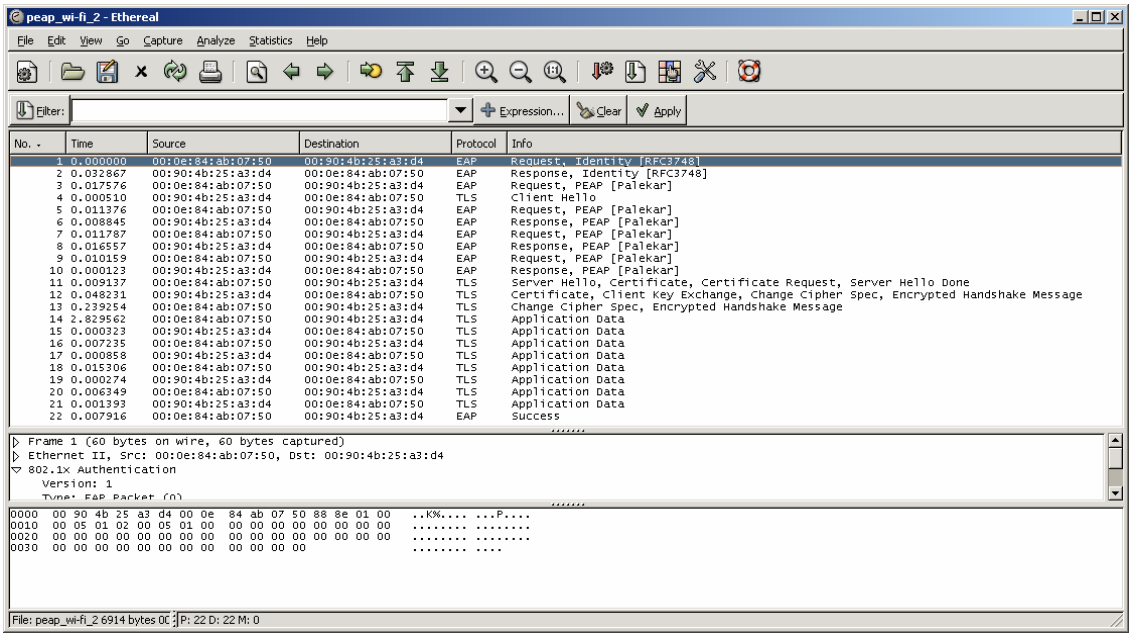


Figura 5-54 - Comunicação wireless – autenticação PEAP-MSCHAPv2.

À semelhança de outros métodos de autenticação EAP já abordados nesta dissertação, também no PEAP a troca de mensagens inicia-se com o envio de uma mensagem ***EAP-Request Identity***, à qual o cliente deve responder com um ***EAP-Response Identity***. Estas mensagens são representadas respectivamente nas tramas 1 e 2 da Figura 5-54.

No entanto, aqui a estação cliente não é obrigada a enviar a sua verdadeira identidade, o que garante a protecção da identidade do cliente. A verdadeira identidade só é obrigatória na segunda fase do processo de autenticação, a qual será protegida pelo túnel TLS.

Nesta experiência a entidade do utilizador não foi alterada, como se pode verificar pela Figura 5-55, o seu valor corresponde ao nome do utilizador "cpereira".

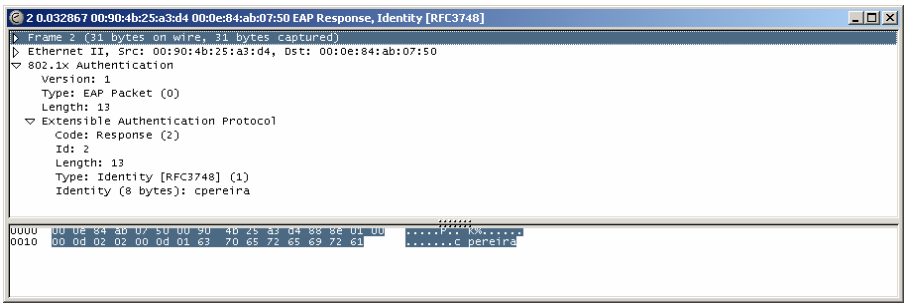


Figura 5-55 – Mensagem EAP-Response Identity enviada pelo cliente.

Após o envio da identidade por parte do cliente, dá-se inicio à negociação do túnel TLS. Basicamente a negociação TLS permite autenticar o servidor de autenticação e

simultaneamente estabelecer uma sessão TLS segura entre a estação cliente e o servidor de autenticação.

As tramas 3 a 13 da Figura 5-54, representam a negociação do túnel TLS e a respectiva autenticação do servidor. Como esperado são pacotes EAP que transportam um conjunto de mensagens TLS.

A trama 3 transporta uma mensagem ***Start***. A trama 4 enviada pelo cliente é uma mensagem ***Client Hello*** (Figura 5-56). Esta mensagem contêm um conjunto de métodos criptográficos, que definem os aspectos de segurança a fornecer pelo TLS. Neste caso são definidos 11 tipos de métodos criptográficos (*Ciphersuites*) distintos. O servidor de autenticação irá escolher um para estabelecer um canal seguro entre as duas entidades.

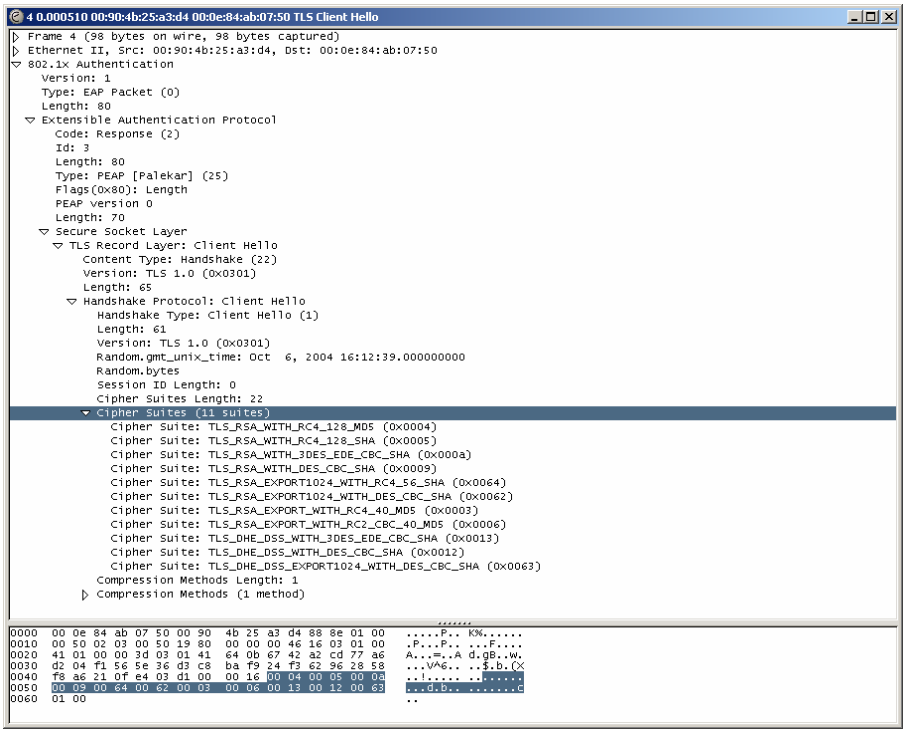


Figura 5-56 – Mensagem TLS Client Hello enviada pela estação cliente.

Em seguida o servidor responde com uma mensagem ***EAP-Request***. Como se verifica pela Figura 5-57 as tramas 5, 7, 9 e 11 são tramas fragmentadas que em conjunto compõem a mensagem EAP-TLS enviada pelo servidor de autenticação.

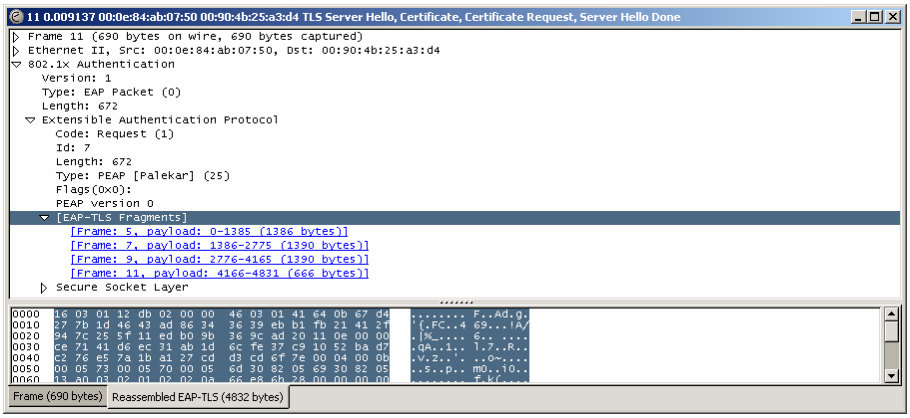


Figura 5-57 – Fragmentos da mensagem EAP-Request enviada pelo servidor de autenticação.

Esta mensagem transporta quatro mensagens TLS, a mensagem ***Server Hello***, ***Certificate***, ***Certificate Request*** e ***Server Hello Done***. Na primeira mensagem, ***Server Hello*** é indicado o método criptográfico escolhido pelo servidor de autenticação para estabelecer uma sessão segura. Nesta experiência o método escolhido foi o *TLS_RSA_WITH_RC4_128_MD5* definido em [RFC2246] como se verifica pela Figura 5-58.

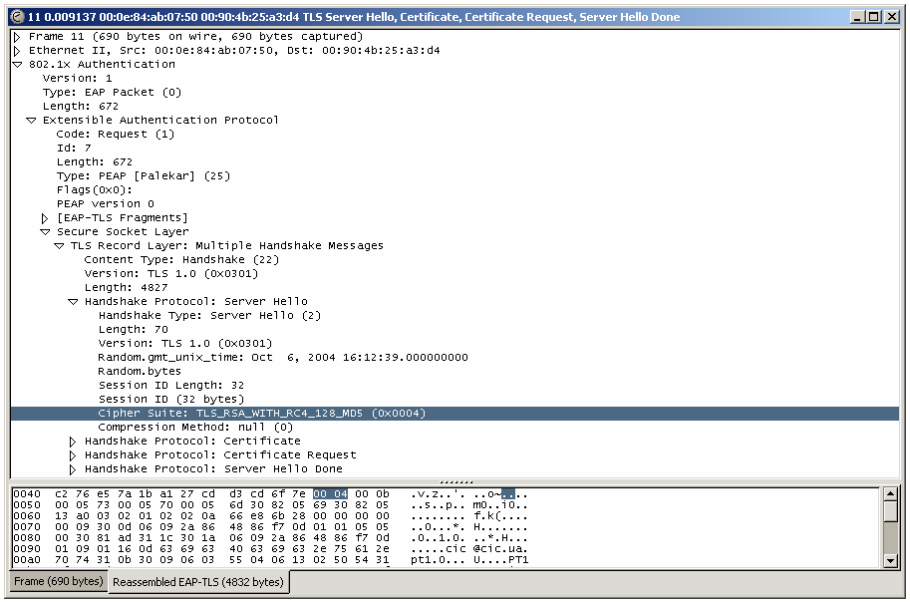


Figura 5-58 – Mensagem TLS Server Hello enviada pelo servidor de autenticação.

Interpretando o método criptográfico escolhido é possível concluir que o protocolo utilizado para gerar as chaves é o TLS com distribuição de chaves simétricas RSA (secção 2.5.1), o algoritmo de cifra a utilizar é o algoritmo de cifra contínua RC4 com chave de tamanho 128 bits, e a função de síntese é a MD5.

A informação relativa ao certificado do servidor é enviada na mensagem ***Certificate***. Entre outra informação é possível observar na Figura 5-59 o nome atribuído ao servidor de autenticação. A mensagem ***Certificate Request***, é ignorada pelo cliente, uma vez que o PEAP, ao contrário do EAP_TLS não exige a utilização de certificados digitais na autenticação do cliente. A mensagem TLS, ***Servel Hello Done***, finaliza a mensagem ***EAP Request*** enviada pelo servidor de autenticação.

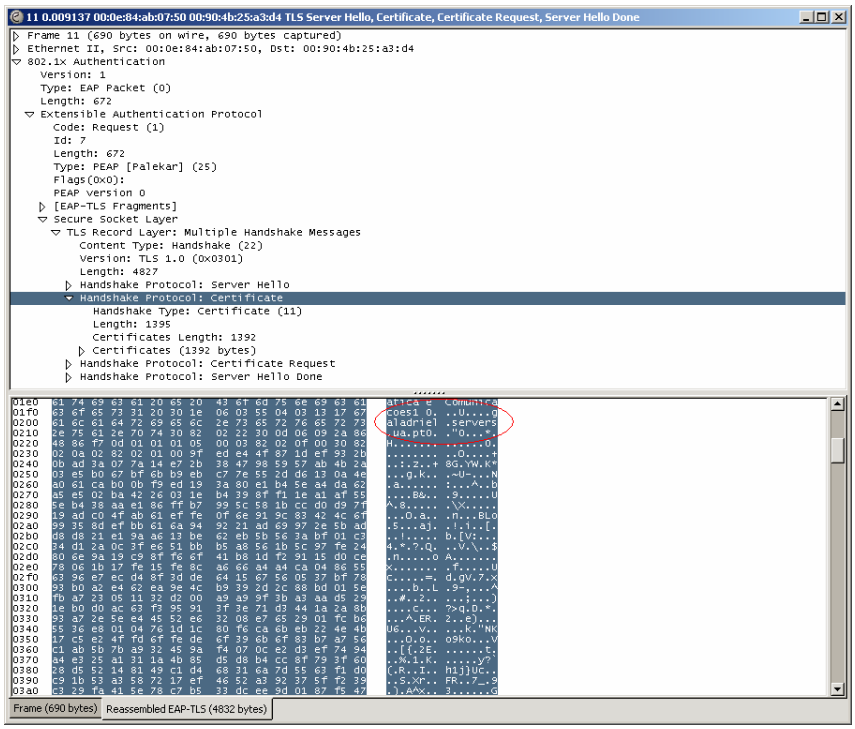


Figura 5-59 – Certificado digital do servidor de autenticação.

As tramas 12 e 13 da Figura 5-54, concluem a primeira fase do processo de autenticação PEAP. O cliente envia uma mensagem TLS do tipo ***Change Cipher Specification***. De acordo com o [RFC2246] esta mensagem é enviada para notificar o servidor que as mensagens seguintes serão protegidas com os novos métodos criptográficos negociados previamente. A mensagem TLS ***Change Cipher Specification*** enviada pelo servidor na trama 13 confirma a utilização dos novos métodos criptográficos.

A autenticação mútua fornecida pelo PEAP, é concluída na fase seguinte com a autenticação do cliente. Esta segunda fase do processo de autenticação consiste numa nova conversação EAP que ocorre sobre uma sessão TLS estabelecida na fase anterior.

À semelhança do que vem sendo referido, a fase dois inicia-se também com a troca inicial de mensagens ***EAP-Request Identity***, e ***EAP-Response Identity*** entre a estação cliente e o servidor de autenticação.

A troca de pacotes nesta fase será de acordo com o método de autenticação negociado entre as entidades intervenientes no processo de autenticação. Neste caso o método utilizado é o MSCHAP-v2. Esta troca de mensagens é representada na Figura 5-54, pelas tramas 14 a 21.

De referir que a captura do conjunto de tramas 14 a 21, efectuada durante a experiência, não permite a análise do conteúdo das mensagens EAP transmitidas. A razão para o sucedido resulta do facto de o túnel TLS negociado e estabelecido na primeira fase do processo de autenticação PEAP-MSCHAPv2, cifrar o seu conteúdo. Como exemplo apresenta-se na Figura 5-60, uma mensagem, ***EAP Response*** , cifrada e transmitida pela estação cliente.

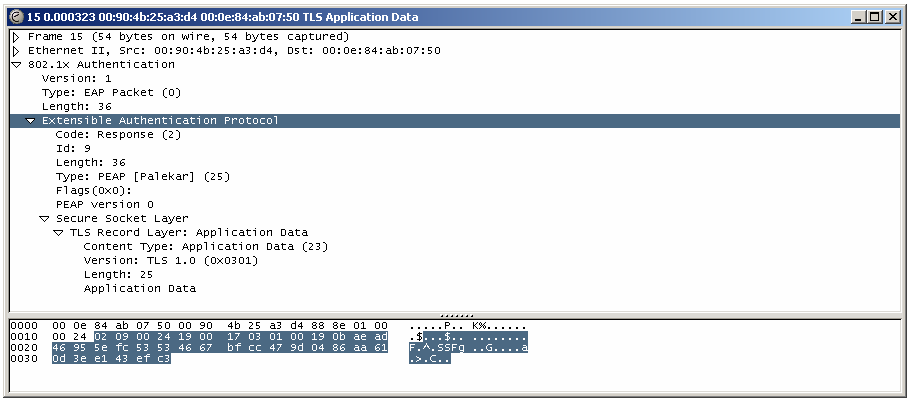


Figura 5-60 – Mensagem EAP Response cifrada enviada pela estação cliente.

Esta situação permite concluir que a captura desta informação por parte de um atacante não permite a este a obtenção de qualquer informação relevante relativamente às credenciais de autenticação de um utilizador legitimo. Mesmo que o método de autenticação utilizado seja baseado em senha.

Para concluir o processo de autenticação PEAP com sucesso, o servidor de autenticação deve enviar uma mensagem ***EAP Success*** (Figura 5-61) para a estação cliente.

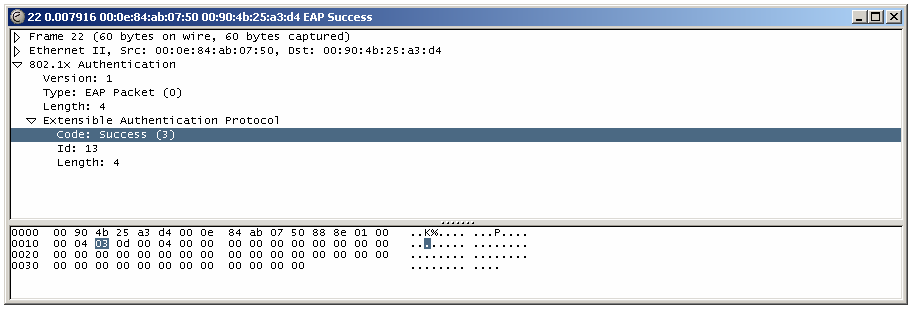


Figura 5-61 – Mensagem EAP Success enviada pelo servidor de autenticação.

Finalmente importa referir que as implementações práticas utilizadas nesta experiência não solucionam uma das vulnerabilidades referidas inicialmente nesta secção. Mais especificamente a protecção das mensagens finais ***EAP-Success*** e ***EAP-Fail***.

No documento [53] este facto é justificado com limitações da implementação do *software* do suplicante fornecido pelo Windows XP SP1, o qual apenas permite o envio de mensagens ***EAP Request*** e ***EAP Response*** através do túnel TLS.

Outro factor que neste momento inviabiliza a protecção das mensagens finais está relacionado com as especificações IEEE 802.1X. Esta exige que um ponto de acesso "produza" uma mensagem ***EAP-Success*** em texto plano, sempre que recebe um ***RADIUS Access Accept*** por parte do servidor de autenticação. Como resultado se numa determinada implementação do protocolo PEAP fosse enviado um ***EAP Success*** ou ***EAP Fail*** cifrado, essa implementação não estaria conforme as especificações do protocolo IEEE 802.1X. Esta problemática está a ser solucionada pelo grupo de trabalho IEEE 802.1aa [20].

5.4.4 EAP-TTLS

Um outro método de autenticação baseado no protocolo EAP é o EAP–TTLS (*EAP-Tunneled Transport Layer Security*). Este mecanismo foi desenvolvido pela empresa *Funk Software*. Foi proposto ao IETF para aprovação como RFC através do documento [28]. O EAP-TTLS é uma extensão ao protocolo EAP-TLS. À semelhança do PEAP analisado na secção 5.4.3, o EAP-TTLS estende o processo de autenticação à utilização de uma ligação segura, estabelecida na fase inicial.

O EAP-TTLS permite autenticação mútua, ou apenas autenticação do servidor perante a estação cliente. A autenticação do cliente pode ser obtida recorrendo a vários métodos EAP, ou a outros protocolos de autenticação tais como PAP, CHAP, MS-CHAP ou MS-CHAP-V2.

Assim, o EAP-TTLS permite a utilização de protocolos de autenticação baseados em senha facilitando a utilização de bases de dados de autenticação já existentes, protegendo estes protocolos contra ataques criptográficos e de intercepção.

À semelhança do que ocorre com o PEAP, também o processo de autenticação EAP-TTLS compreende duas fases.

- A fase inicial é responsável pela negociação TLS de modo a estabelecer uma ligação segura, através de uma sessão TLS. Nesta fase apenas o servidor de autenticação é autenticado. Opcionalmente a estação cliente também pode ser autenticada. A restante troca de mensagens é então cifrada com a chave negociada previamente.
- A fase 2 ocorre assumindo que a sessão TLS foi estabelecida. Esta sessão segura é utilizada para completar uma nova troca de mensagens EAP onde o cliente é autenticado. A informação trocada nesta fase inclui a autenticação do utilizador, a negociação de funcionalidades de segurança e a distribuição de chaves.

Modelo de níveis protocolares

Os pacotes EAP-TTLS são encapsulados em pacotes EAP. O protocolo EAP por sua vez requer um protocolo portador que o transporte. Os próprios pacotes EAP-TTLS são encapsulados em pacotes TLS, usados para encapsular informação de autenticação.

As mensagens EAP-TTLS podem ser descritas usando um modelo de níveis protocolares, onde cada um dos níveis encapsulam o nível abaixo destes. Os diagramas representados na Figura 5-62 clarificam a relação entre os diversos protocolos.

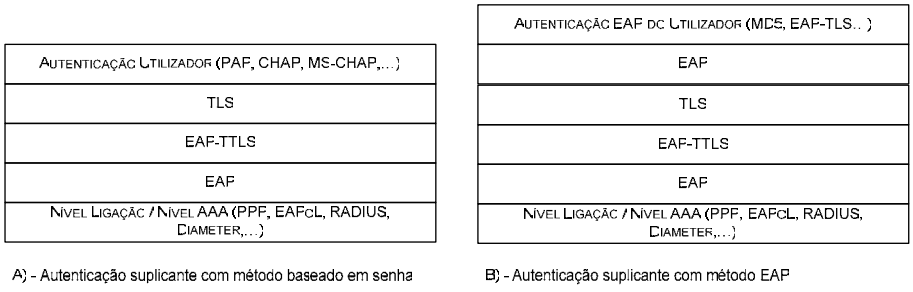


Figura 5-62 – Modelo de níveis protocolares EAP-TTLS.

À semelhança dos protocolos de autenticação abordados nas secções anteriores, também o EAP-TTLS recorre aos mesmos métodos de encapsulamento das mensagens EAP. Isto é o protocolo EAPoL é utilizado para transportar as mensagens trocadas entre o ponto de acesso e a estação cliente. O protocolo RADIUS é utilizado para transportar as mensagens EAP entre o ponto de acesso e o servidor de autenticação.

A utilização de certificados digitais é apenas obrigatória na autenticação do servidor. Esta característica é uma mais valia para cenários de aplicação wireless em que a implementação e gestão de certificados digitais nas estações clientes se torna uma tarefa complexa.

Antes de descrevermos a experiência prática realizada, apresenta-se na Figura 5-63 a troca de mensagens durante a fase de autenticação recorrendo ao protocolo EAP-TTLS.

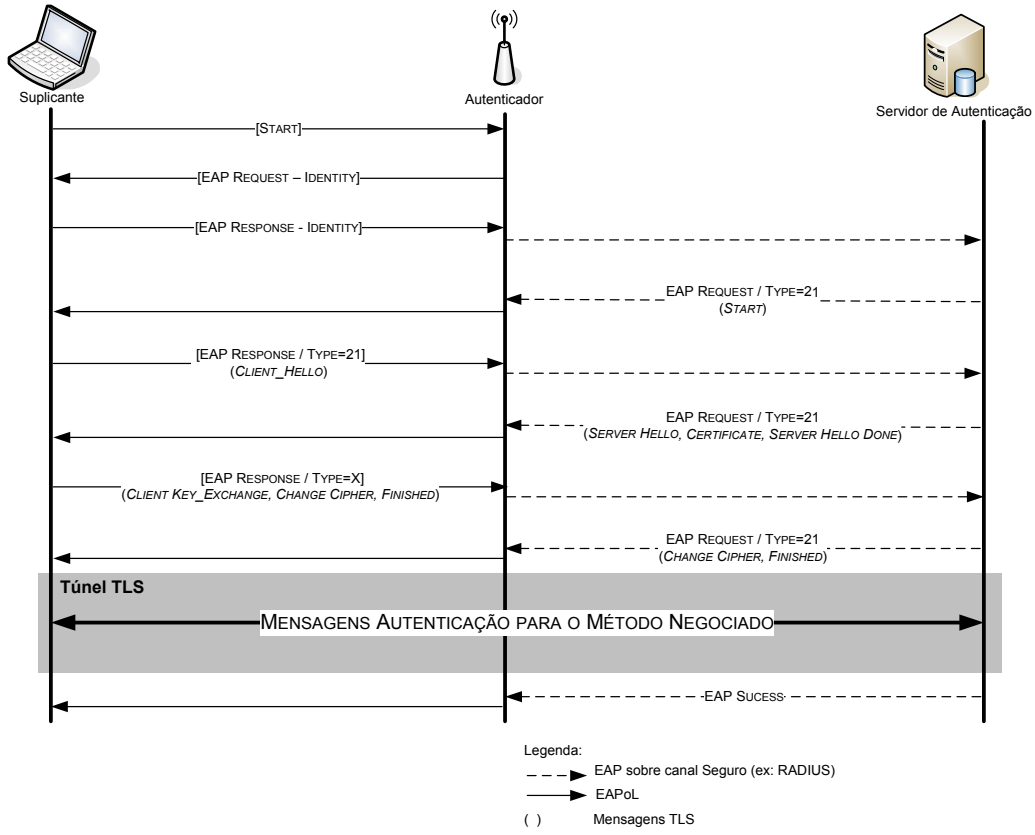


Figura 5-63 – Autenticação EAP-TTLS.

Experiência Prática- Análise de mensagens trocadas na fase de autenticação EAP-TTLS

A experiência prática efectuada visa estudar o funcionamento do protocolo EAP-TTLS. O cenário implementado para esta experiência é representado na Figura 5-31. A estação cliente possui uma placa PCMCIA Cisco Aironet PCM-350 com endereço MAC 00:0B:46:56:30:BB.

O ponto de acesso utilizado é um Cisco Aironet 1100 com endereço MAC igual a 00:02:8A:78:B6:F0 e configurado com endereço IP 10.0.0.1. O servidor de autenticação está a ser executado num PC com sistema operativo Windows 2000 Server. Este servidor foi configurado com endereço IP 10.0.0.20.

O certificado utilizado pelo servidor de autenticação foi emitido pela entidade certificadora WLAN CA. Relativamente ao *software* que implementa o suplicante na estação cliente optou-se pela implementação comercial da empresa *Funk Software*, designada por, *Odyssey Client* versão 3.00. Esta aplicação é compatível com o sistema operativo Windows XP. A implementação adoptado para a realização desta experiência foi o *software* designado *Odyssey Server* versão 2.01, também este comercializado pela empresa *Funk Software*. Ambas as aplicações utilizadas são versões de demonstração.

O protocolo de autenticação da estação cliente utilizado nesta experiência foi o MSCHAPv2. Configurou-se o utilizador "cristiano". As credenciais do utilizador são fornecidas por nome de utilizador/senha. O seu valor é definido na base de dados do servidor de autenticação. de modo a proteger a identidade do utilizador definiu-se o valor *anonymous*, a ser enviado pela estação cliente na fase inicial do processo de autenticação. Finalmente procedeu-se à configuração das políticas de acesso no servidor de autenticação.

As mensagens capturadas no meio wireless são representadas na Figura 5-64.

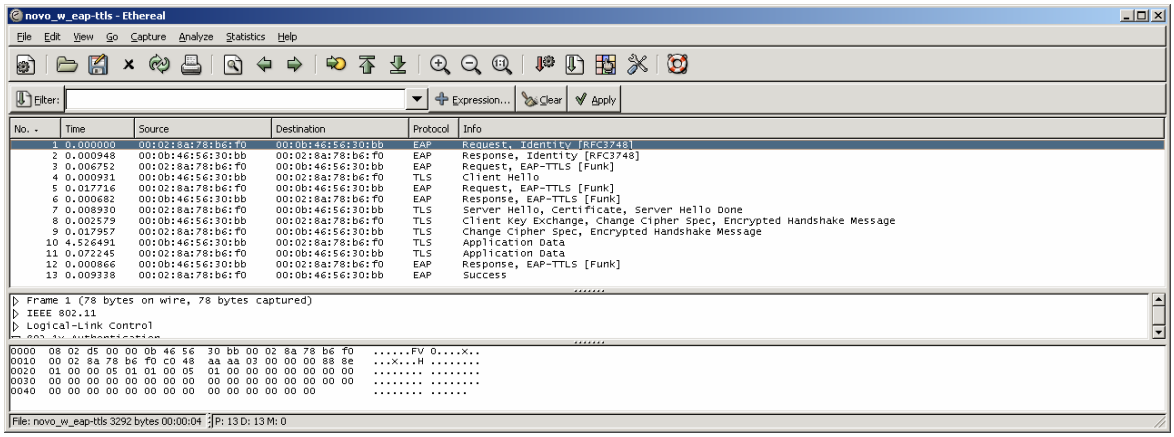


Figura 5-64 - Comunicação wireless – autenticação EAP-TTLS.

O processo de autenticação inicia-se com a negociação EAP entre o ponto de acesso e a estação cliente. À semelhança do que ocorre com o método EAP-TLS o ponto de acesso envia um pedido de identificação para a estação cliente através de uma mensagem ***EAP-Request Identity*** (Figura 5-65).

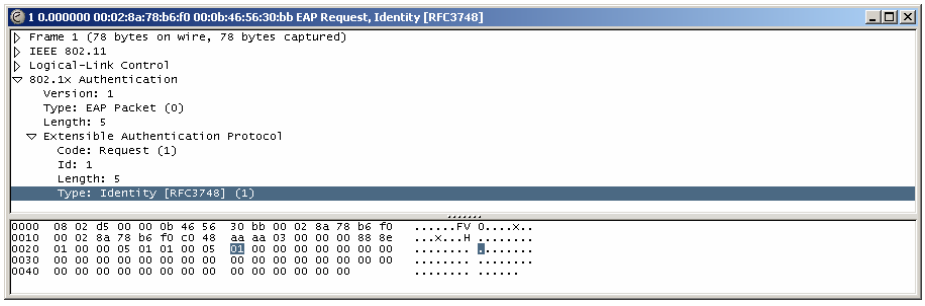


Figura 5-65 – EAP-Request Identity enviado pelo ponto de acesso.

Em resposta a este pedido a estação não envia a identificação do utilizador mas sim um valor definido previamente, e que pretende ocultar a verdadeira identidade do utilizador. Como referido anteriormente nesta experiência definiu-se o valor *anonymous*. Esta informação é enviada através de um pacote ***EAP-Response Identity*** (Figura 5-66).

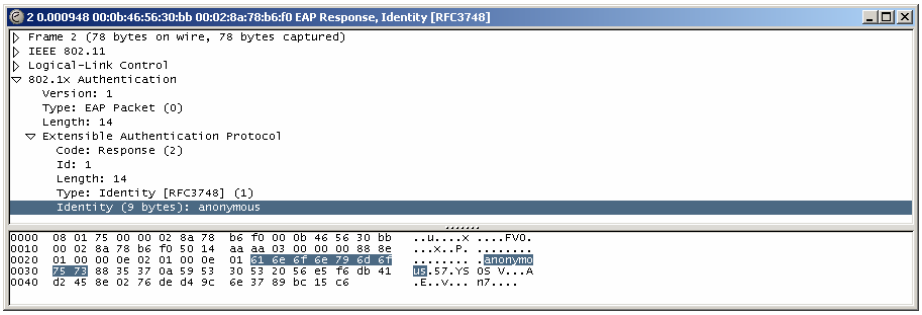


Figura 5-66 - EAP-Response enviado pela estação cliente.

Após a recepção da identificação da estação cliente por parte do servidor de autenticação, inicia-se a primeira fase de autenticação, com o envio por parte do servidor de uma mensagem ***EAP-Request*** (Figura 5-67). O campo ***Type*** identifica o método de autenticação suportado pelo EAP a utilizar, neste caso possui o valor 21, que se refere ao método EAP-TTLS de acordo com Tabela 5-2. Verifica-se ainda que o bit S (*start*) está activo. Esta é a única mensagem trocada em que o valor do bit S é um.

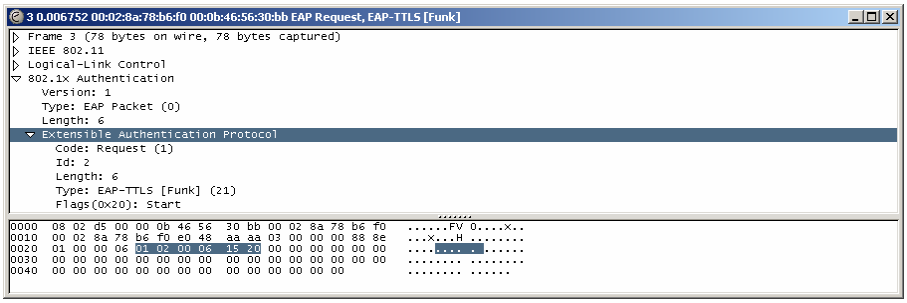


Figura 5-67 – EAP-Request Start enviado pelo servidor de autenticação.

Na fase 1, dá-se inicio à troca de mensagens TLS utilizada para autenticar o servidor perante a estação cliente. A estação cliente envia um pacote ***EAP-Response***, que transporta uma mensagem ***Client Hello*** (Figura 5-68), e corresponde à trama 4 da Figura 5-63. O conteúdo desta mensagem é idêntico ao referido em experiências anteriores, isto é contem o conjunto de métodos criptográficos, que define os aspectos de segurança a fornecer pelo TLS.

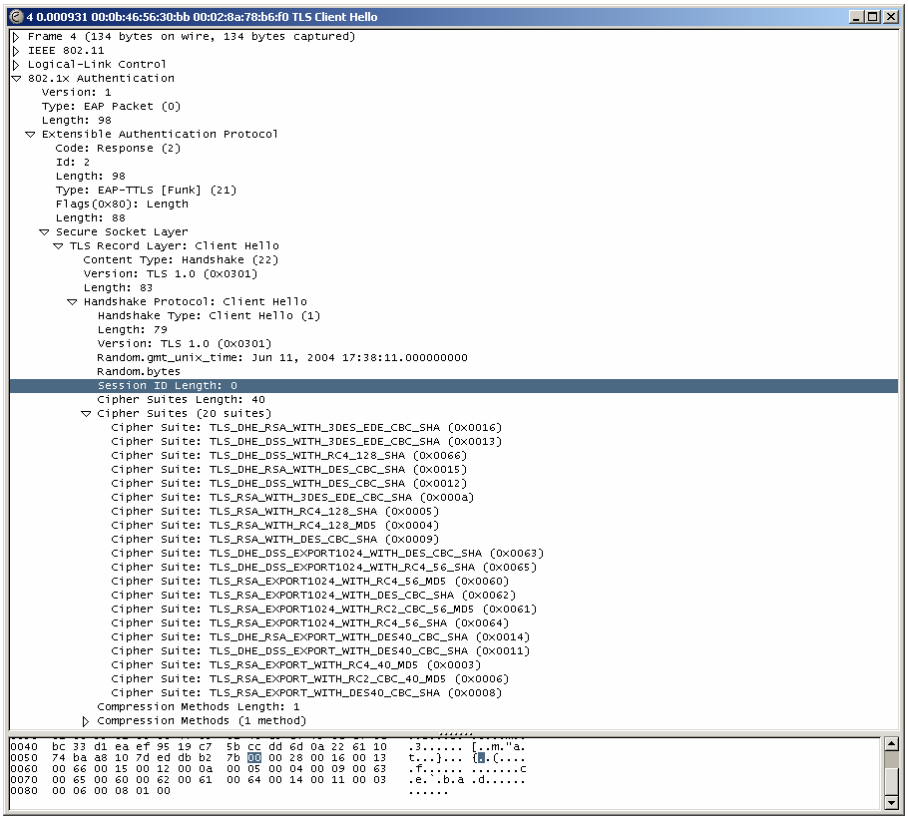


Figura 5-68 – Mensagem Client Hello enviada pela estação cliente.

A aplicação utilizada nesta experiência suporta 20 métodos criptográficos distintos, como se pode verificar através do campo ***Cipher Suite*** representado na Figura 5-68.

A resposta do servidor de autenticação é dada através de uma mensagem ***EAP-Request*** (Figura 5-69). O servidor envia várias mensagens TLS numa única trama EAP, nomeadamente a mensagem ***Server Hello***, o seu certificado através da mensagem ***Server Certificate*** e a finalizar uma mensagem ***Server Hello Done***. De referir que o conteúdo desta mensagem é fragmentado nas tramas 5 e 7 da Figura 5-64.

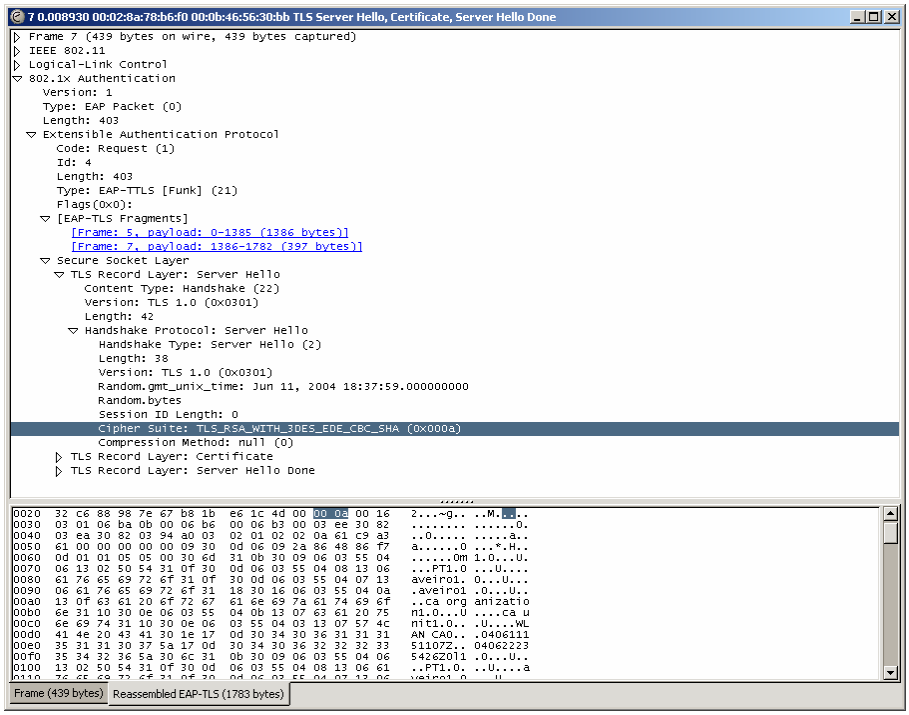


Figura 5-69 – Mensagem Server Hello enviada pelo servidor de autenticação.

Nesta experiência o método criptográfico escolhido pelo servidor de autenticação foi o *TLS_RSA_WITH_3DES_EDE_CBC_SHA* definido em [RFC2246]. Interpretando o método criptográfico escolhido é possível concluir que o protocolo utilizado para gerar as chaves é o TLS com distribuição de chaves simétricas RSA (secção 2.5.1), o algoritmo de cifra a utilizar é o algoritmo de cifra simétrica por blocos 3DES com chave de tamanho 168 bits e a função de síntese a utilizar é SHA. De forma idêntica às experiências anteriores, as tramas 8 e 9 concluem o processo de negociação do túnel TLS. A partir deste momento informação relativa à autenticação da estação cliente pode ser trocada entre os intervenientes de uma forma segura.

A troca de pacotes nesta segunda fase será de acordo com o método de autenticação negociado entre as entidades intervenientes no processo de autenticação. De referir que a captura das tramas 10 e 11, efectuada durante a experiência, não permite a análise do conteúdo das mensagens EAP transmitidas. A razão para o sucedido resulta do facto de o túnel TLS negociado e estabelecido na primeira fase do processo de autenticação cifrar o seu conteúdo.

O sucesso da autenticação do utilizador é confirmado através da mensagem ***EAP-Success*** (Figura 5-70) enviada pelo servidor de autenticação, após confirmar as políticas de acesso definidas para o utilizador "cristiano".

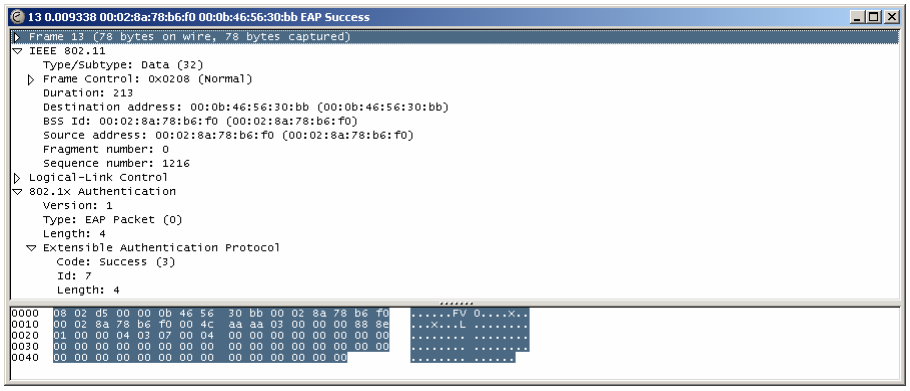


Figura 5-70 – Mensagem EAP Success enviada pelo servidor de autenticação.

5.4.5 Conclusões das Experiências Práticas

Estas experiências permitem concluir que não existem diferenças significativas entre os protocolos de autenticação PEAP e EAP-TTLS. Importa referir que em termos de mercado, são disponibilizadas um conjunto superior de implementações no que se refere ao protocolo PEAP. Em contrapartida este suporta um conjunto mais limitado de protocolos de autenticação da estação cliente, quando comparado com o EAP-TTLS.

O facto de ambos os protocolos não exigirem certificados digitais para autenticação da estação cliente, ao contrário do método EAP-TLS, torna-os do ponto de vista de gestão de redes wireless uma solução mais interessante.

Refira-se ainda que à exepção do protocolo EAP-TLS, os restantes protocolos, PEAP e EAP-TTLS, são um conjunto de especificações ainda em fase de desenvolvimento, o que não permite às aplicações desenvolvidas e utilizadas nestas experiências uma completa fiabilidade. Algumas das implementações são versões de demonstração de versões comerciais, com as devidas limitações.

5.5 Mecanismos de Confidencialidade e Integridade

Esta secção descreve os protocolos adoptados pelo WPA e pelo grupo IEEE 802.11i para implementar os mecanismos de confidencialidade e integridade em redes IEEE 802.11. Na secção 5.5.1 são apresentados conceitos que permitem uma compreensão da hierarquia de chaves utilizadas pelo WPA e pelas redes RSN. A secção 5.5.2 apresenta o protocolo de confidencialidade TKIP. Este protocolo foi desenvolvido de modo a permitir a utilização de um mecanismo de confidencialidade mais seguro que o WEP nos dispositivos wireless actuais. Um novo protocolo capaz de fornecer um mecanismo de confidencialidade robusto e suficientemente seguro está a ser desenvolvido pelo IEEE 802.11i, designado AES-CCMP. Este protocolo é abordado genericamente na secção 5.5.3. Um estudo mais aprofundado do AES-CCMP não é feito nesta dissertação dado o carácter de trabalho em desenvolvimento deste protocolo.

5.5.1 Hierarquia e Distribuição de Chaves

As mensagens *unicast* e *multicast* têm diferentes requisitos de segurança. As mensagens *unicast* trocadas entre dois dispositivos necessitam de um mecanismo de privacidade comum a ambos. Este é alcançado utilizando uma chave secreta para cada par de dispositivos que comunicam entre si. Estas chaves secretas são designadas por chaves do par (*pairwise key*s). Normalmente este tipo de chaves são utilizadas para proteger a comunicação entre uma estação cliente e um ponto de acesso. Isto significa que ambas as entidades armazenam o valor desta chave. Uma estação cliente armazena apenas uma chave enquanto o ponto de acesso deve armazenar um conjunto de chaves do par, respeitante a cada uma das estações associadas a este.

Relativamente a dados *multicast*, estes são recebidos por diversos dispositivos wireless em simultâneo formando um grupo de confiança. Este grupo deve partilhar uma chave comum a todos os membros. Esta chave é designada por chave de grupo (*Group Key*). Cada estação cliente e respectivo ponto de acesso apenas armazenam uma chave de grupo comum a todos.

O conceito de chave do par e de chave de grupo é ilustrado na Figura 5-71 .

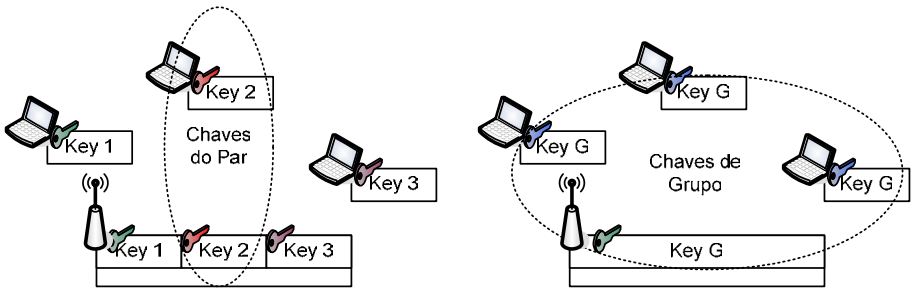


Figura 5-71 – Chave de grupo e chaves do par.

Os métodos de gestão e distribuição de chaves de grupo diferem dos das chaves do par. Esta diferença originou a definição de duas hierarquias de chaves distintas:

1. Hierarquia de chaves do par – refere todas as chaves utilizadas para protecção de mensagens *unicast*.
2. Hierarquia de chaves de grupo – refere as chaves necessárias à protecção de mensagens *multicast*.

Importa ainda definir o conceito de chave pré - partilhada (*Preshared key*) e de chave baseada no servidor (*Server Based Key*). Tal como o seu nome sugere, uma chave pré – partilhada é distribuída e instalada num ponto de acesso e numa estação cliente por um qualquer método externo ao WPA e RSN. Um exemplo da utilização de chaves pré-partilhadas são os sistemas baseados no WEP. Nestes a chave utilizada é configurada em todos os dispositivos (estações clientes e pontos de acesso) que pretendem comunicar. As chaves pré – partilhadas contornam completamente o conceito de autenticação de camada superior, uma vez que a autenticação é conseguida simplesmente com a prova de posse da chave secreta. Este tipo de sistema apenas permite a autenticação do dispositivo cliente e não do utilizador. De acordo com [63] não se deve optar pela utilização de chaves pré-partilhas em redes WPA, dadas as vulnerabilidades detectadas. Alternativamente, as chaves baseadas no servidor exigem um processo de autenticação de camada superior que permita à estação cliente e ao servidor de autenticação a criação de chaves secretas comuns. O processo de obtenção das chaves de cifra a partir da autenticação de camada superior no WPA é referido na secção 5.5.1.4.

5.5.1.1 Hierarquia de Chaves do Par

No topo da hierarquia de chaves do par, tem-se a chave do par mestra PMK (*Pairwise Master Key*). Esta chave é obtida a partir do processo de autenticação de camada superior. Para cada

estação cliente é gerada uma PMK distinta a partir da qual as restantes chaves da hierarquia são obtidas. No caso de se utilizarem chaves pré-partilhadas a chave configurada em cada um dos dispositivos wireless assume o papel de PMK.

A autenticação de camada superior ocorre entre o suplicante e o servidor de autenticação, resultando na geração de chaves PMK idênticas em cada uma destas entidades. Para que a ligação wireless esteja protegida é necessário fornecer a PMK ao ponto de acesso. O IEEE 802.11i não especifica claramente como deve ser efectuada a transferência da PMK do servidor de autenticação para o ponto de acesso. No entanto recomenda a utilização do atributo MS-MPPE-Recv-Key (vendor_id =17, RFC 2548 secção 2.4.3) no caso de ser utilizado o protocolo RADIUS entre estas duas entidades. Este atributo é obrigatório no WPA.

A chave PMK não é utilizada directamente em nenhuma operação de segurança. Esta é apenas utilizada para obter um conjunto de chaves temporárias PTK (*Pairwise Transient Key*) que serão então utilizadas na protecção da comunicação entre dois dispositivos.

São necessárias quatro chaves temporárias PTK distintas para proceder à protecção da comunicação:

- Chave de cifra dos dados (*DataEncr*) (128 bits).
- Chave de integridade dos dados (*DataMIC*) (128 bits).
- Chave de cifra *EAPoL Key* (*EAPOLEncr*) (128 bits).
- Chave de integridade *EAPoL Key* (*EAPOLMIC*) (128 bits).

As duas primeiras chaves temporárias são utilizadas respectivamente nas funções de cifra e integridade dos dados. A chave *EAPOLMIC* é utilizada para fornecer integridade às mensagens durante a troca inicial de mensagens (*4-way handshake*). Esta troca de mensagens é abordada na secção 5.5.1.3. A chave *EAPOLEncr* é utilizada para cifrar a chave de grupo obtida no final da troca de mensagens (*4-way handshake*).

As chaves temporárias são partilhadas entre os dois dispositivos uma vez que ambos calculam um valor idêntico para essas mesmas chaves (as chaves calculadas não são transmitidas entre os dispositivos).

Cálculo das chaves do par temporárias

Para efectuar o cálculo das chaves do par temporárias, o dispositivo wireless utiliza um conjunto de parâmetros como entrada numa função PRG (*Pseudo Random Generator*). Os parâmetros de entrada são: (i) PMK - valor da chave do par mestra; (ii) MAC1 e MAC2 - endereços MAC da estação cliente e do autenticador; (iii) *A-Nonce* – valor aleatório (*nonce*) gerado pelo suplicante na fase de autenticação; (iv) *S-Nonce* – valor aleatório gerado pelo suplicante durante a fase de autenticação. Os valores do *A-Nonce* e *S-Nonce* são transmitidos entre os dispositivos durante o processo de troca de mensagens *4-way handshake*, descrito na secção 5.5.1.3.

A função PRG não é utilizada apenas no cálculo das chaves do par temporárias. Por exemplo, também é utilizada para efectuar o cálculo dos valores aleatórios (*nonces*) ou para o cálculo das chaves temporárias de grupo como será descrito na secção 5.5.1.2. A acumulação de todas estas funcionalidades numa só função pode criar um potencial problema, caso o mesmo resultado da função seja utilizado para propósitos distintos. Por outro lado para garantir uma maior eficiência das implementações pretende-se utilizar apenas uma função PRG. Basicamente é pretendida uma função PRG que garanta resultados distintos para propósitos diferentes, mesmo quando a "semente" da PRG é a mesma.

Para atingir este objectivo o WPA e as redes RSN definiram um conjunto de funções pseudo – aleatórias PRF (*Pseudo Random Function*). Cada uma das funções produz um determinado número de bits. Estas funções são referidas como *PRF-n* onde *n* é o número de bits gerados. O documento [16], define as seguintes funções pseudo-aleatórias:

- *PRF-128*
- *PRF-192*
- *PRF-256*
- *PRF-384*
- *PRF-512*

Os três parâmetros de entrada de uma função PRF são:

- K – representa um valor aleatório ou chave secreta.
- A – *String* que especifica a aplicação da função.

- B – valor da concatenação (||) de dados específicos para cada caso, por exemplo, endereços MAC e valores aleatórios.

A notação utilizada para estas funções pseudo-aleatórias é :

- *PRF-n* (*K,A,B*)

Deste modo para o cálculo das chaves do par temporárias do TKIP, a função PRF utilizada é:

PRF-512 (*PMK, "Pairwise key expansion", MAC1||MAC2||SNonce||ANonce*)

onde PMK é o valor da chave do par mestra, *"Pairwise key expansion"* é o valor da string que especifica a aplicação da função e (*MAC1||MAC2||SNonce||ANonce*) é a concatenação dos restantes parâmetros de entrada necessários para calcular as chaves do par temporárias.

Para calcular os valores aleatórios *A-Nonce* ou *S-Nonce*, a função PRF utilizada é:

PRF-256 (*valor aleatório, "Init Counter", MAC||Time)*

O parâmetro *valor aleatório* é um valor aleatório numérico de tamanho 256 bits, a *string* que especifíca a aplicação da função é igual a "*Init Counter*", o parâmetro *Time* é o valor de tempo actual e deve ser obtido preferencialmente através do protocolo NTP (*Network Time Protocol*) e o parâmetro MAC é o endereço MAC do dispositivo. Os valores aleatórios *A-Nonce* e *S-Nonce* têm uma dependência relativamente ao tempo, o que garante a não repetição deste valor. Por sua vez o valor das chaves do par temporárias são dependentes do *A-Nonce* e *S-Nonce* garantindo-se assim a utilização de chaves temporárias únicas.

O processo de obtenção de chaves referido anteriormente completa a hierarquia de chaves do par descrita nesta secção. A hierarquia de chaves do par implementada nos protocolos TKIP e AES é ilustrada na Figura 5-72 e Figura 5-73 respectivamente.

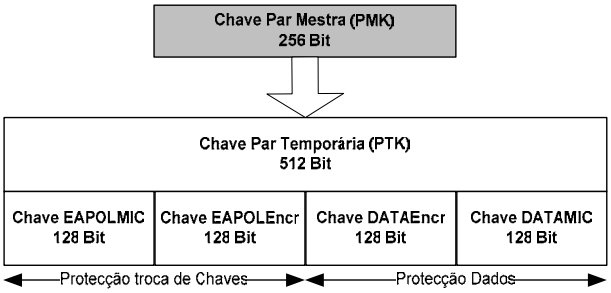


Figura 5-72 – Hierarquia chaves do par – TKIP.

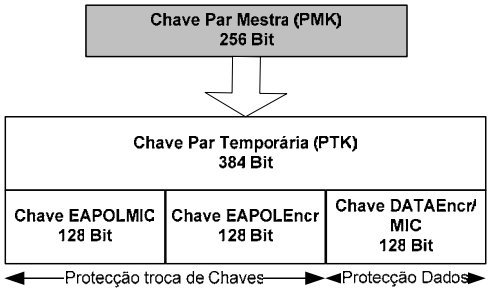


Figura 5-73 – Hierarquia chaves do par – AES.

5.5.1.2 Hierarquia de Chaves de Grupo

Como se observa pela Figura 5-71 ao contrário da hierarquia de chaves do par todos os dispositivos partilham um conjunto comum de chaves de grupo, o que permite a todas as estações decifrar mensagens *multicast* enviadas pelo ponto de acesso. Esta abordagem resolve o problema de armazenamento de chaves. No entanto surge um outro problema sempre que uma estação abandona o grupo *multicast*.

Quando a estação abandona o grupo, esta mantém o conhecimento da chave de grupo podendo decifrar todas as mensagens *multicast* trocadas entre os elementos do grupo, o que não é aceitável do ponto de vista de segurança. A solução passa pela troca das chaves de grupo sempre que um dispositivo abandone o grupo. Para tal é enviada a nova chave de grupo directamente a cada uma das estações clientes pertencentes ao grupo. O processo de distribuição e actualização das chaves temporárias de grupo é descrito na secção 5.5.1.3. A informação é enviada de forma segura uma vez que estão estabelecidas chaves temporárias para a comunicação *unicast* entre o ponto de acesso e cada uma das estações associadas.

Gerada a chave GMK (*Group Master Key*) é necessário obter as chaves de grupo temporárias GTK (*Group Transient key*). São necessárias duas chaves temporárias distintas para proceder à protecção da comunicação *multicast*:

- Chave de cifra de grupo (*DataEncr*) (128 bits).
- Chave de integridade de grupo (*DataMIC*) (128 bits).

O conjunto de chaves GTK é obtido da chave GMK, combinando um valor aleatório (*nonce*) e o endereço MAC do ponto de acesso.

Cálculo das chaves de grupo temporárias

Para obter o valor das GTK o documento [16] aconselha a utilização das funções *PRF-256* ou *PRF-128* para TKIP e AES respectivamente. Deste modo para o cálculo das chaves de grupo temporárias, a função PRF a utilizar é:

$$PRF\text{-}n\ (GMK,\ \text{“Group key expansion”},\ MAC||GNonce)$$

onde GMK é o valor da chave de grupo mestra, *“Group key expansion”* é o valor da string que especifica a aplicação da função e (*MAC*||*GNonce*) é a concatenação dos restantes parâmetros

de entrada necessários para calcular as chaves de grupo temporárias. O número de bits gerados, *n*, é igual a 256 ou 128 de acordo com o algoritmo de cifra utilizado.

O processo de obtenção de chaves referido anteriormente, completa a hierarquia de chaves de grupo descrita nesta secção. A Figura 5-74 e a Figura 5-75 ilustram a hierarquia de chaves de grupo implementada nos protocolos TKIP e AES-CCMP respectivamente.

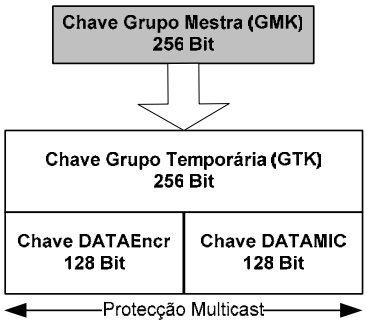


Figura 5-74 – Hierarquia chaves grupo – TKIP.

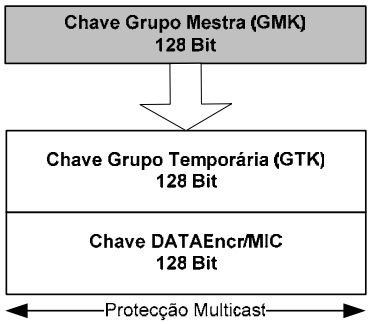


Figura 5-75 – Hierarquia chaves grupo AES.

5.5.1.3 Troca de Mensagens "4-Way Handshake"

A troca de mensagens *4-way handshake* definida no WPA e no futuro protocolo IEEE 802.11i [16] possui as seguintes funcionalidades: confirmação do estado activo das estações clientes, actualização das chaves temporárias e processo de instalação e confirmação das chaves temporárias. Esta troca de mensagens recorre ao protocolo IEEE 802.1X, mais especificamente a mensagens do tipo ***EAPoL-Key*** .

Antes da análise do processo de troca de mensagens *4-way handshake* apresentaremos em detalhe os campos que constituem uma mensagem ***EAPoL-Key*** (Figura 5-76) definida no WPA. Esta mensagem é ligeiramente diferente da mensagem ***EAPoL-Key*** definida na arquitectura IEEE 802.1X [17], possuindo alguns campos adicionais, nomeadamente os campos ***Key Information***, ***Key Nonces***, ***Key RCS*** e ***Key Data Length***.

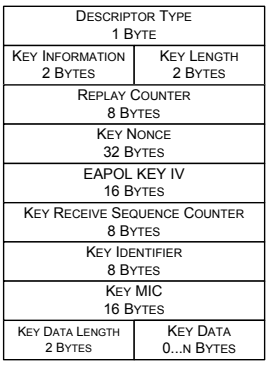


Figura 5-76 - Versão WPA da mensagem EAPol Key.

O campo ***Descriptor Type*** possui o valor 254 que identifica a variante WPA do descritor. O campo ***Key Information*** ilustrado na Figura 5-77 é composto por diversos bits de controlo, que auxiliam no processo de troca de mensagens, e subcampos com informação acerca do tipo de chaves. Este campo é descrito posteriormente. O comprimento da chave em bytes é armazenado no campo ***Key Length***. O valor do campo ***Replay Counter*** é incrementado em todas as mensagens de modo a detectar um possível reenvio de mensagens antigas. A excepção é quando uma mensagem é enviada em resposta a um pedido de confirmação ***Ack Request***, onde o valor do campo ***Replay Counter*** é o mesmo da mensagem ***Ack Request***. O valor dos números aleatórios utilizados para gerar as chaves temporárias é armazenado no campo ***Key Nonce***. O campo ***EAPoL-Key IV*** contém o valor do vector de inicialização a ser utilizado com a chave do par temporária *EAPOLEncr*. O campo ***Key Receive Sequence Counter*** contém o valor do número de sequência esperado na primeira trama após a instalação das chaves temporárias. Este campo permite uma protecção contra ataques de repetição. O campo ***Key Identifier*** actualmente não é utilizado no WPA. No futuro poderá ser utilizado para identificar as chaves temporárias, num cenário em que a implementação de chaves múltiplas seja possível. O campo ***Key MIC*** tal como o seu nome indica contém o valor do código de integridade de mensagem calculado sobre toda a trama ***EAPoL Key***. Os últimos dois campos, ***Key Data Length*** e ***Key Data*** apresentam respectivamente o comprimento em bytes do campo ***Key Data*** e o material relativo às chaves temporárias. Por exemplo, o valor da chave de grupo temporária cifrada é transmitido neste campo. O campo ***Key Data*** é também utilizado para transmitir elementos de informação IE (*Information Elements*). Na secção 3.4.9 foi feita uma descrição dos elementos de informação definidos para o WEP. O WPA e as redes RSN definem novos elementos de informação para negociar os parâmetros criptográficos [33].

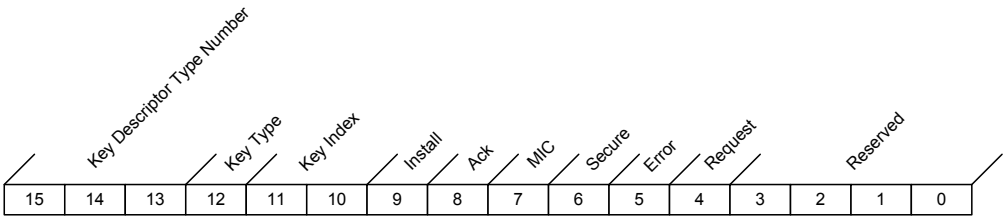


Figura 5-77 – Campo Key Information.

O campo ***Key Information*** é composto pelos bits e subcampos representados na Figura 5-77. Os ***bits 0 1 2*** e ***3*** são reservados para uso futuro e possuem o valo zero. O bit ***Request*** é colocado a um pelo suplicante para pedir ao autenticador que dê início ao processo de troca de mensagens de modo a actualizar as chaves temporárias. O bit de controlo ***Error*** é colocado a um para assinalar um erro na confirmação do código de integridade da mensagem. O bit ***Secure*** é colocado a um quando o processo de troca de mensagens está completo, indicando que a ligação passa a ser segura. A indicação de que o código de integridade de mensagens foi calculado e armazenado no campo ***Key MIC*** é efectuada colocando o valor do bit de controlo ***MIC*** a um. O bit ***Ack*** é colocado a um nas mensagens enviadas pelo autenticador que requerem uma resposta do suplicante. O bit ***Install*** indica que a nova chave do par temporária deve ser instalada para ser utilizada. Para chaves de grupo este bit tem o valor zero. Os bits 10 e 11, ***KeyIndex***, identificam em conjunto o índice da chave de grupo que deve ser utilizada. Este campo é utilizado para proceder à actualização da chave de grupo. O bit ***KeyType*** distingue entre mensagens de chaves do par, valor igual a um, ou mensagens de chaves de grupo, valor igual a zero. Os três últimos bits ***KeyDescriptorTypeNumber*** identificam a função de síntese e o algoritmo criptográfico utilizados na protecção das mensagens ***EAPoL-Key***. No caso do WPA o seu valor é igual a 1 e indica que o código de integridade da mensagem é calculado usando HMAC-MD5, e o algoritmo de cifra utilizado para protecção do campo ***Key Data*** é o RC4.

Após a recepção da PMK, gerada no processo de autenticação, e enviada do servidor de autenticação para o autenticador, dá-se inicio ao processo de troca de mensagens *4-way handshake* ilustrado na Figura 5-78. Um exemplo detalhado do processo *4-way handshake* e da distribuição da chave temporária de grupo no WPA é apresentado na secção 5.5.1.4.

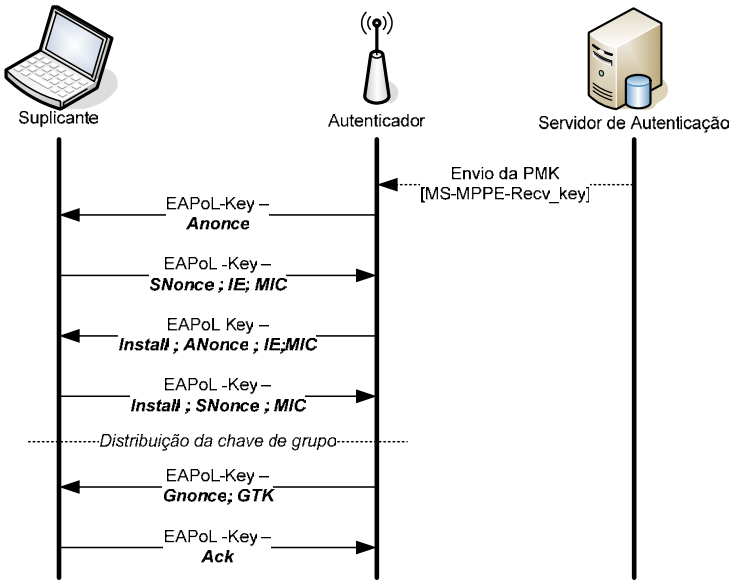


Figura 5-78 – Troca de mensagens *4-way Handshake*.

O processo de troca de mensagens *4-way handshake* é iniciado após a recepção da chave do par mestra PMK. Esta chave é entregue pelo servidor de autenticação ao autenticador através do atributo RADIUS, MS-MPPE-Rec-Key.

A primeira mensagem ***EAPoL-Key*** é enviada do autenticador para o suplicante. Esta mensagem transporta o valor do número aleatório (*A-Nonce*) gerado pelo autenticador. Após a recepção desta primeira mensagem o suplicante possui a informação necessária (chave PMK, *A-Nonce*, *S-Nonce* e endereços MAC) para gerar o conjunto de chaves do par temporária PTK utilizando as funções PRF descritas anteriormente.

O autenticador apenas pode calcular a chave temporária PTK após a recepção da segunda mensagem ***EAPoL-Key***. Esta mensagem é enviada pelo suplicante e contêm a informação em falta: o valor do *S-Nonce* e o endereço MAC do suplicante. Esta mensagem é enviada em texto plano. No entanto possui um código de integridade da mensagem MIC que não estava presente na primeira mensagem. Esta é a primeira utilização de uma chave temporária calculada pelo suplicante. Em particular, é utilizada a chave *EAPOLMIC* para gerar o código de integridade da mensagem MIC, recorrendo ao código de autenticação de mensagens HMAC-MD5. O campo ***Key Data*** desta mensagem é transmitido em texto plano e contém os elementos de informação IE negociados durante a fase de associação. Entre outra informação estes elementos identificam os protocolos de autenticação e gestão de chaves AKMP (*Authentication and Key Management Protocol*) a utilizar e identificam os algoritmos de cifra a utilizar, por exemplo TKIP ou AES-CCMP. A inclusão destes elementos previne que um

dispositivo wireless altere os parâmetros de segurança após a negociação inicial. Os IE não vão cifrados mas estão protegidos com o código de integridade da mensagem, de modo a não serem alterados.

A terceira mensagem, enviada do autenticador para o suplicante informa que o autenticador está preparado para utilizar as chaves temporárias calculadas na fase anterior. Esta terceira mensagem é também protegida com o código MIC calculado pelo autenticador. É também enviado um número de sequência nesta mensagem, valor este que será utilizado na primeira mensagem cifrada.

O autenticador apenas instala as chaves temporárias quando receber a quarta e última mensagem do processo *4-way handshake*. Este momento de espera é necessário para garantir que o suplicante recebeu a terceira mensagem, pois se esta falhar deve ser enviada novamente. Acontece que se as chaves já estivessem instaladas, a mensagem reenviada iria cifrada, e consequentemente seria rejeitada pelo suplicante.

A quarta mensagem enviada pelo suplicante confirma a conclusão do processo de troca de mensagens *4-way handshake*, e indica que o suplicante vai instalar as chaves temporárias para dar inicío à cifra de dados. Após a recepção desta mensagem, o autenticador também procede à instalação das chaves.

Distribuição e actualização das chaves de grupo

Após o processo de *4-way handshake* o autenticador procede à distribuição das chaves temporárias de grupo. O valor da chave de grupo é calculado pelo autenticador recorrendo às funções PRF referidas anteriormente. Em seguida a chave de grupo e o valor aleatório *G-Nonce* são transmitidos do autenticador para o suplicante através de uma mensagem ***EAPoL-Key***. O campo ***Key Data*** desta mensagem transporta o valor cifrado da chave GTK. A chave de grupo GTK é cifrada com a chave do par temporária *EAPOLEncr*.

A conclusão do processo de distribuição da chave de grupo é feita pelo suplicante com uma confirmação de recepção da chave de grupo. Para tal envia uma mensagem ***EAPoL-Key*** com o bit ***Ack*** (Figura 5-77) igual a 1.

Dada a necessidade da troca periódica das chaves de grupo, é necessário um método que proporcione esta actualização sem afectar o normal funcionamento da rede. Este aspecto seria problemático se as estações cliente não possuíssem capacidade para armazenar mais do que uma chave de grupo. Aproveitando a característica do protocolo WEP que define a

capacidade para armazenar até quatro chaves secretas num dispositivo, torna-se possível guardar até três chaves de grupo de uma só vez. A quarta chave será a chave do par utilizada para cifrar as mensagens *unicast*.

Tal como referido na secção 4.2.2, cada trama 802.11 possui um campo de tamanho dois bits designado por ***Key ID*** que especifica qual das quatro chaves deve ser utilizada para cifrar os dados. Esta funcionalidade garante a actualização, sempre que uma estação abandona a rede, das chaves temporárias de grupo de modo transparente para o funcionamento da rede.

Admitamos a título de exemplo que a chave de grupo utilizada num dado momento está armazenada em ***KeyID*** igual a 1. Quando for necessário efectuar a sua actualização o autenticador envia a nova chave de grupo GTK numa mensagem ***EAPoL-Key***. Os bits ***KeyIndex*** (Figura 5-77) desta mensagem têm o valor igual a 2. Assim o suplicante deve armazenar a nova chave de grupo no ***KeyID*** 2. Durante a fase de actualização as mensagens *multicast* continuam a ser enviadas recorrendo à chave de grupo armazenada em ***KeyID*** igual a 1 até que todas as estações associadas sejam informadas da nova chave a utilizar, isto é recebam a mensagem ***EAPoL-Key*** com a nova chave de grupo. As estações cliente confirmam a recepção da nova chave de grupo através de uma mensagem ***EAPoL-Key*** com o bit ***Ack*** a 1. Quando o ponto de acesso receber a confirmação de todas as estações clientes então passa a utilizar as novas chaves de grupo, armazenadas em ***KeyID*** igual a 2.

5.5.1.4 Detalhe da Obtenção de Chaves WPA

A Figura 5-79 apresenta um exemplo detalhado do processo de obtenção de chaves do par e das chaves de grupo temporárias no protocolo WPA. Na Figura 5-76 estão representados os diversos campos das mensagens ***EAPoL Key*** .

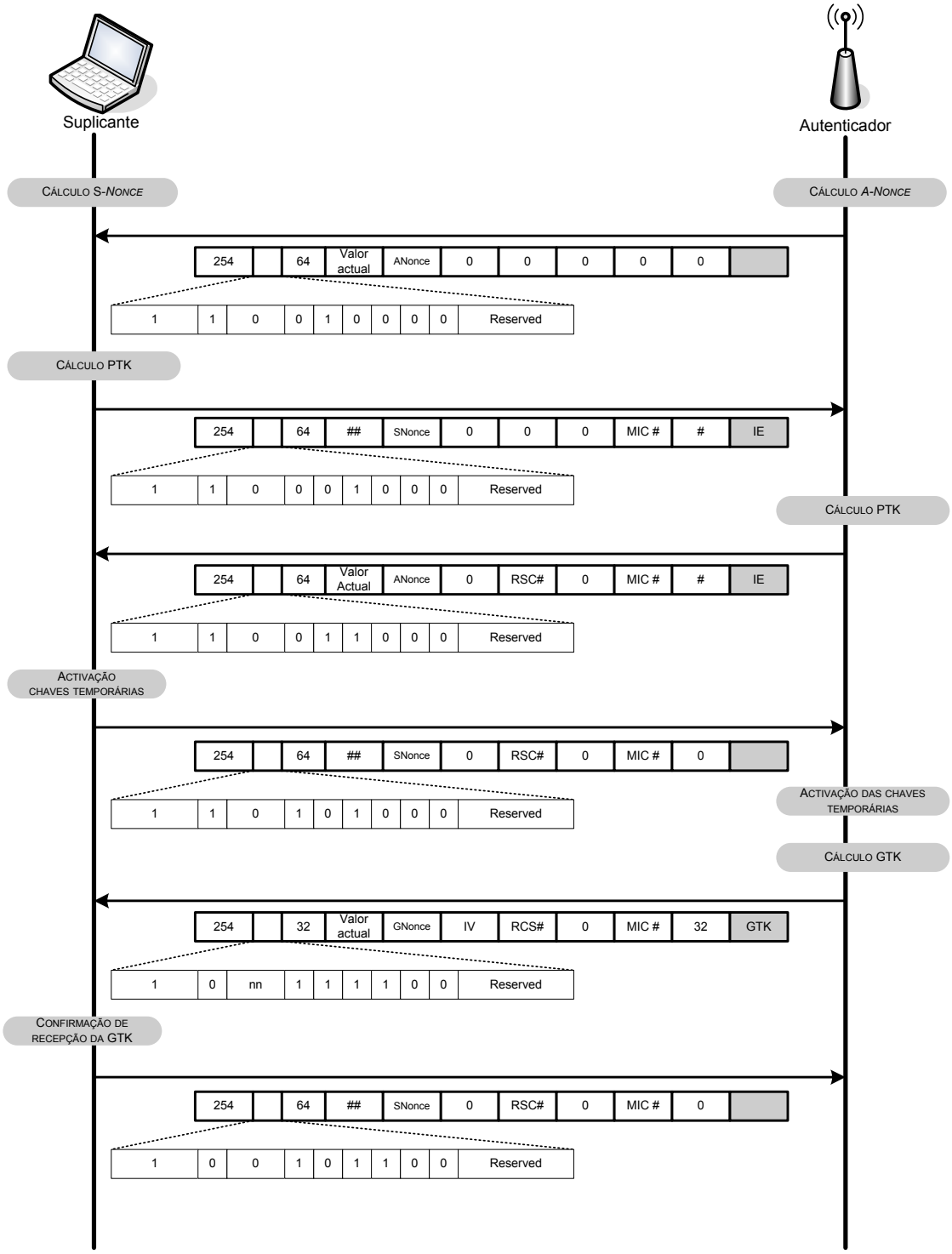


Figura 5-79 – Obtenção das chaves temporárias WPA.

Para o exemplo apresentado nesta secção considera-se que o autenticador já possui o valor da chave par mestra (PMK) atribuída pelo servidor de autenticação ou definida por uma chave pré-partilhada. Previamente ao início da troca de mensagens *4-way handshake* para o cálculo das

chaves do par temporárias, ou chaves de sessão, o suplicante e o autenticador procedem ao cálculo dos valores aleatórios *S-Nonce* e *A-Nonce* respectivamente.

Observando o campo ***Key Information*** da primeira mensagem ***EAPoL-Key*** verifica-se que o valor do bit ***Error*** é igual a zero e, uma vez que o processo *4-way handshake* não está concluído, o valor do bit ***Secure*** também é igual a zero. O valor do bit ***MIC*** igual a zero significa que o código de integridade da mensagem não foi aplicado a esta mensagem o que justifica o valor nulo do campo ***Key MIC***. Como o processo de obtenção de chaves do par temporárias não está concluído o valor do bit ***Install*** é igual a zero. O campo ***Key Type*** possui valor igual a um, indicando que o processo se refere à obtenção das chaves do par. O campo ***Replay Counter*** é o valor de um contador iniciado a zero quando a PMK é estabelecida e é incrementado em mensagens sucessivas. O valor aleatório *A-Nonce* calculado previamente é transportado no campo ***Key Nonce***. Os restantes campos possuem o valor zero.

Após a recepção da primeira mensagem o suplicante calcula o conjunto de chaves do par temporárias (PTK) e envia uma mensagem ***EAPoL-Key*** de resposta com informação necessária para que o autenticador também calcule as chaves do par temporárias. Como a mensagem anterior não possui código de integridade o valor do bit ***Error*** é igual a zero. O bit ***Secure*** mantém o valor zero até ao final do processo de troca de mensagens. Esta segunda mensagem possui um código de integridade pelo que o valor do bit ***MIC*** é um. O campo ***Key MIC*** da mensagem ***EAPoL-Key*** possui o valor do código de integridade da mensagem. O bit ***Install*** continua a ter o valor zero. A instalação das chaves negociadas só efectuada no final do processo. O campo ***Key Descriptor Type Number*** tem valor igual a um, indicando que o código de integridade de mensagem é calculado através do código de autenticação de mensagens HMAC-MD5. O valor do campo ***Replay Counter*** tem o mesmo valor da mensagem anterior uma vez que esta é uma resposta à mensagem anterior. O campo ***Key Nonce*** transporta o valor aleatório *S-Nonce*. O campo ***Key Data*** contém os elementos de informação IE utilizados para negociar os parâmetros criptográficos durante a fase de autenticação. Neste momento o autenticador procede ao cálculo das chaves do par PTK

A terceira mensagem ***EAPoL-Key*** tem duas funções. Em primeiro lugar garantir perante o suplicante que o autenticador tem conhecimento da chave PMK. Se assim não fosse o valor do código de integridade da mensagem calculado pelo autenticador não era igual ao calculado pelo suplicante. Em segundo lugar indicar ao suplicante que o autenticador está preparado

para instalar e utilizar as chaves do par temporárias. Observando o campo ***Key Information*** verifica-se que o bit ***MIC*** possui o valor um. O valor do código de integridade calculado é armazenado no campo ***Key MIC***. O campo ***Key RSC*** indica ao suplicante o valor inicial do número de sequência que o autenticador pretende utilizar. O campo ***Key Data*** possui os elementos de informaçao que o autenticador utilizou na fase de autenticação.

A quarta mensagem indica ao autenticador que as chaves do par temporárias foram instaladas pelo suplicante e que pode fazer o mesmo com as cópias das suas chaves. Esta mensagem é em tudo idêntica à segunda mensagem ***EAPoL-Key***, mas com o campo ***Key Data*** sem qualquer informação. O bit ***Secure*** continua a ter valor igual a zero até ao momento em que as chaves do par temporárias do suplicante e autenticador estejam instaladas.

Após o processo *4-way handshake*, dá-se a distribuição das chaves temporárias de grupo GTK. Esta distribuição é composta por duas mensagens. A primeira mensagem transmitida do autenticador para o suplicante é uma mensagem ***EAPoL-Key*** com o valor da chave GTK calculada previamente pelo autenticador através de uma função PRF. Observando o campo ***Key Information*** verifica-se que o bit ***Key Type*** é igual a zero indicando que o processo se refere à obtenção das chaves de grupo. O bit ***MIC*** possui o valor um. O bit ***Ack*** possui o valor 1 indicando que esta mensagem necessita de uma resposta de confirmação. Os bits ***Key Identifier*** têm o valor da localização onde deve ser armazenada a nova chave de grupo. De referir que o bit ***Secure*** está activo indicando que a comunicação se processa de forma segura, uma vez que as chaves do par já estão activas. O campo ***KeyNonce*** possui o valor aleatório *G-Nonce*. Finalmente importa referir que a chave GTK é cifrada e transmitida no campo ***Key Data***. Como o campo ***Key Descriptor Type Number*** tem valor igual a um, a chave GTK é cifrada recorrendo ao algoritmo de cifra RC4.

Após a recepção da mensagem anterior o suplicante conclui o processo enviando uma mensagem ***EAPoL-Key*** de confirmação para o autenticador. Esta mensagem é idêntica à quarta mensagem do processo *4-way handshake* à excepção do campo ***Key Information*** onde o bit ***Secure*** está activo e o bit ***Key Type*** é igual a zero.

Esta secção conclui a descrição dos processos de distribuição de chaves do par e de grupo temporárias. As hierarquias de chave apresentadas permitem solucionar o aspecto da gestão e atribuição de chaves de cifra para os mecanismos de segurança implementados nas redes não-cabladas. Este era um dos pontos fracos da solução apresentada no protocolo WEP que não fornecia qualquer capacidade de gestão ou distribuição de chaves de cifra.

5.5.2 Temporal Key Integrity Protocol – TKIP

Esta secção aborda um novo protocolo desenvolvido especificamente para utilização em dispositivos wireless. O protocolo TKIP (*Temporal Key Integrity Protocol*) fornece melhorias significativas em termos de confidencialidade e integridade relativamente ao WEP. Tal como o WEP, o TKIP baseia-se no algoritmo de cifra continua RC4. O principal objectivo do TKIP é incrementar o nível de confidencialidade e integridade das redes 802.11 através de uma simples actualização de *firmware* dos dispositivos wireless que existem actualmente no mercado e que não são compatíveis com as futuras redes RSN. Este protocolo foi adoptado e é obrigatório na certificação WPA. A futura norma IEEE 802.11i refere o TKIP como opcional. Este deve ser entendido como uma solução transitória a adoptar em cenários de rede pré-RSN.

Na secção 5.5.2.1 é feita uma introdução ao protocolo TKIP, com referência às novas medidas implementadas por este. A secção 5.5.2.2 apresenta a implementação do algoritmo. Finalmente a secção 5.5.2.3 descreve os detalhes do código de integridade de mensagens *Michael* e um conjunto de contramedidas implementadas pelo TKIP, para proteger alguns aspectos de segurança relacionados com a utilização do *Michael*.

5.5.2.1 Introdução

O TKIP pretende solucionar algumas das vulnerabilidades detectadas no protocolo WEP e que foram analisadas na secção 4.3. A lista seguinte apresenta as vulnerabilidades específicas do protocolo WEP que o TKIP pretende solucionar:

- Riscos associados à repetição de chaves contínuas devido ao tamanho e reutilização do vector de inicialização.
- Obtenção da chave secreta através de métodos de ataque interactivos à chave WEP.
- Alteração do conteúdo das mensagens.
- Repetição e introdução de mensagens.
- Falta de mecanismos de actualização e gestão da chave secreta.

Para solucionar estas vulnerabilidades com um nível de segurança aceitável sem que ocorram constrangimentos de desenvolvimento, o TKIP aplica um conjunto de medidas correctivas e algoritmos compatíveis com o *hardware* actual nomeadamente: (i) um novo código de integridade de mensagens; (ii) alteração das regras de selecção e utilização do valor do vector de inicialização; (iii) uma nova função de mistura de chaves; (iv) incremento do tamanho do vector de inicialização.

Integridade de Mensagens

Na secção 4.2.1 foi descrita a utilização do ICV (*Integrity Check Value*) utilizado no WEP para detecção de modificação de mensagens. Como se verificou na secção 4.3.3 este mecanismo não fornece uma protecção eficaz.

O novo código de integridade de mensagens MIC (*Message Integrity Code*), desenvolvido por Niels Ferguson [29] baseia-se num método designado por *Michael*. Tal como o WEP este método não recorre à operação de multiplicação no cálculo do MIC e utiliza apenas operações de XOR, substituição e de deslocamento. Este facto permite a sua implementação em dispositivos wireless com baixa capacidade de cálculo sem afectar o seu desempenho.

Selecção e utilização do vector de inicialização

De modo a solucionar as ameaças relacionadas com o tamanho e reutilização do vector de inicialização o TKIP introduz um conjunto de novas regras para a utilização do vector de inicialização. Essencialmente são três as diferenças:

- O tamanho do vector de inicialização é incrementado de 24 bits para 48 bits.
- O IV passa a ter a funcionalidade de contador de sequência.
- O método de obtenção do IV é alterado para não serem gerados IVs fracos.

No desenvolvimento do TKIP o tamanho do vector de inicialização foi incrementado de 32 bits. Estes, somados aos 24 bits originais, resultariam num IV de tamanho 56 bits. No entanto na prática apenas são utilizados 48 bits uma vez que o primeiro byte é rejeitado para prevenir IVs fracos. Este incremento no tamanho do IV elimina as vulnerabilidades associadas à repetição de vectores de inicialização descritas na secção 4.3.1.

Para prevenir ataques de repetição, o TKIP utiliza o IV como um contador de sequência de tramas transmitidas designado TSC (*TKIP Sequence Counter*). Na realidade o contador TSC e o

IV são o mesmo. O TKIP define que este valor é sempre inicializado a zero e incrementado de 1 em cada trama transmitida. Sempre que se verifica a repetição do TSC, a trama é descartada prevenindo-se assim ataques de repetição.

Quando o valor do TSC chega ao fim, isto é, atinge o valor máximo permitido pelos 48 bits, as opções propostas em [33] são a substituição das chaves de cifra de sessão ou então terminar a comunicação entre os dispositivos. A reutilização de qualquer valor do TSC compromete o tráfego já enviado. Como o TSC e o IV são o mesmo, a repetição deste valor implica as mesmas vulnerabilidades associadas à repetição de IVs referidas no WEP.

Função de mistura de chaves

Na secção 4.3.6 foi descrito o método de ataque FMS para obtenção da chave secreta baseado na observação dos primeiros bytes de dados cifrados. Para evitar este ataque o TKIP implementa uma nova função de mistura de chaves que garante uma nova chave de cifra RC4 para cada trama transmitida. No WEP a chave de cifra muda em cada trama transmitida devido ao IV, mas a sua componente secreta (excluindo o IV) mantém-se constante.

O esquema de mistura e obtenção de chaves de cifra RC4 (Figura 5-80) apresentado ao IEEE 802.11i e adoptado pelo WPA, divide o processo de cálculo em duas fases. Os dados de entrada da fase inicial são a chave de cifra de sessão, o endereço MAC do dispositivo emissor, e os 32 bits mais significativos do contador de sequência TSC. Os parâmetros de entrada da fase 2 são o resultado da fase inicial designado por TTAK (*TKIP mixed Transmit Address and Key*), a chave de cifra de sessão e os 16 bits menos significativos do contador TSC. O tamanho da chave TTAK é de 80 bits.

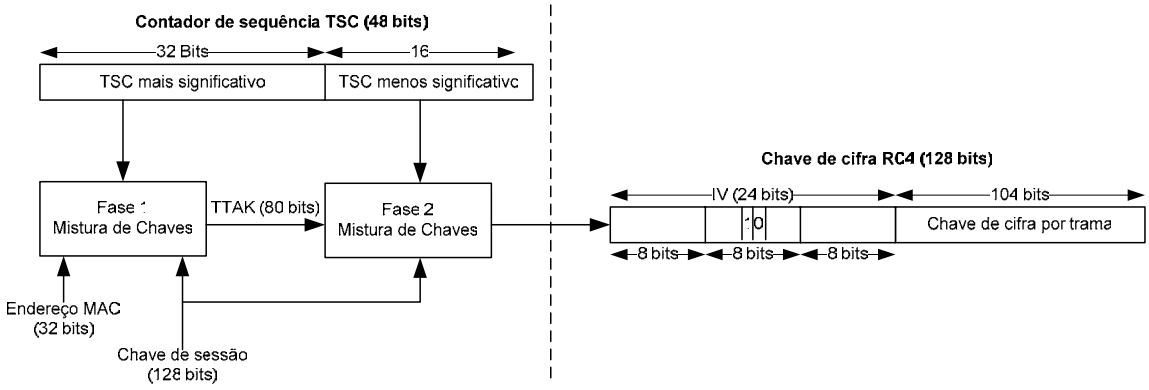


Figura 5-80 – Obtenção e mistura da chave RC4.

O resultado da fase 2 da função de mistura de chaves é a chave de cifra RC4, com tamanho 128 bits. Esta função de mistura de chaves resulta numa chave de cifra RC4 para cada trama transmitida. O documento [30] apresenta uma análise detalhada da função de mistura de chaves.

A estrutura interna da chave resultante, obedece às especificações do WEP. Isto é, os primeiros 3 bytes são transmitidos em texto plano tal como o IV do WEP. No IV o primeiro byte contém os 8 bits mais significativos do TSC menos significativo; o terceiro byte contém os 8 bits menos significativos do TSC menos significativo; finalmente o segundo byte, designado no TKIP por ***WEP Seed1*** (Figura 5-83), é idêntico ao primeiro byte excepto os bits 5 e 4 que são forçados a 1 e zero respectivamente. Estes dois bits são forçados com estes valores para evitar a geração de chaves de cifra contínuas conhecidas como chaves fracas.

5.5.2.2 Implementação do Algoritmo

Esta secção apresenta em detalhe a implementação do algoritmo TKIP. Para a descrição efectuada nesta secção considera-se a atribuição da chave mestra (PMK) através de um mecanismo de autenticação de camada superior ou através de uma chave pré-partilhada. As chaves de sessão ou chaves do par temporárias são calculadas de acordo com a hierarquia de chaves do par descrita na secção 5.5.1.1.

O algoritmo do protocolo TKIP recorre a duas chaves do par temporárias: (i) chave de cifra dos dados (*DataEncr*); (ii) chave de integridade dos dados (*DataMIC*). A primeira é utilizada no processo de cálculo da chave contínua RC4. A chave *DataMIC* é utilizada no cálculo do código de integridade de mensagens *Michael*. Na Figura 5-81 as chaves do par temporárias *DataEncr* e *DataMIC* são representadas respectivamente por TK (*Temporal Key*) e chave MIC.

Encapsulamento

O TKIP complementa o encapsulamento WEP com uma série de funções adicionais representadas na Figura 5-81. O encapsulamento WEP é o representado na Figura 4-10.

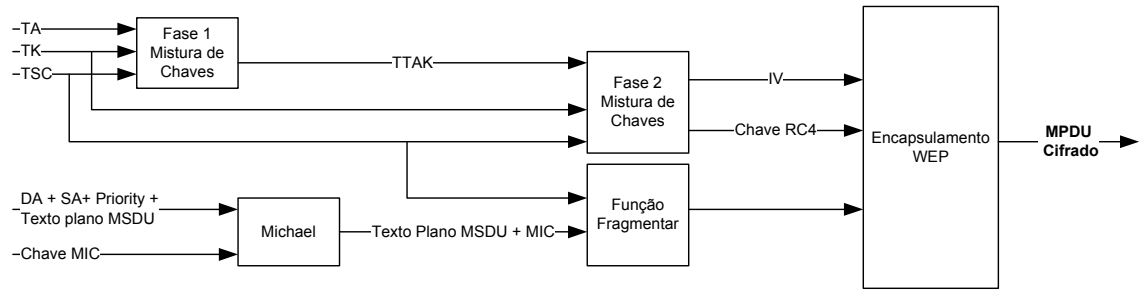


Figura 5-81 – Diagrama de blocos do processo de encapsulamento TKIP.

Quando uma estação pretende transmitir um MSDU, a implementação TKIP utiliza a chave do par temporária *Chave MIC* para calcular o código de integridade da mensagem sobre os campos ***SA*** (*Source MAC Address*), ***DA*** (*Destination MAC Address*), ***Priority*** e ***Data***. O MIC calculado é adicionado ao campo de dados do MSDU, o que aumenta o tamanho do campo ***Data*** em 8 bytes. Uma análise ao código de integridade *Michael* é efectuada na secção 5.5.2.3.

Se necessário o protocolo IEEE 802.11 fragmenta um MSDU e o respectivo MIC em dois ou mais MPDUs. A função fragmentar (Figura 5-81) associa um valor incremental do contador TSC a cada MPDU resultante da fragmentação.

Como referido anteriormente a obtenção da chave RC4 é dividida em duas fases. A primeira fase combina a chave do par temporária TK (*DataEncr*), o endereço MAC do emissor TA e os 16 bits menos significativos do contador TSC para obter o valor intermédio da chave TTAK. A chave intermédia TTAK, a chave temporária TK e os 16 bits menos significativos do TSC são os valores de entrada da segunda fase da função de mistura de chaves. O resultado é a chave RC4 utilizada pelo encapsulamento WEP.

Desencapsulamento

O processo de desencapsulamento do TKIP é representado na Figura 5-82.

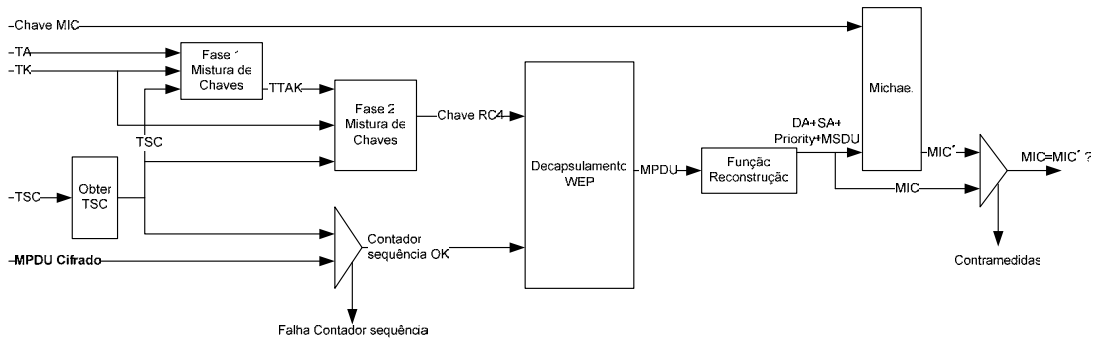


Figura 5-82 – Diagrama de blocos do processo de desencapsulamento TKIP.

Antes do processo de desencapsulamento de uma trama de dados cifrados o TKIP obtém o valor do contador de sequência TSC. O valor do contador de sequência é transmitido no MPDU (Figura 5-83). Em seguida verifica se o valor do contador de sequência é válido. O dispositivo receptor deve manter para cada sessão um contador TSC único cujo valor é incrementado de 1 para cada trama recebida. Para que o valor do TSC recebido seja válido este é comparado com o valor TSC do último MPDU e deverá ter um valor superior. No caso do valor ser igual ou inferior então a protecção de repetição é activada e o MPDU é descartado. Se o valor do TSC for válido então o TKIP utiliza a função de mistura de chaves para associar o número de sequência com a chave de cifra temporária de modo a determinar a chave de cifra RC4.

O processo de desencapsulamento WEP, representado na Figura 4-11, valida o vector de verificação de integridade ICV. Se a validação do ICV ocorre com sucesso, então a trama de dados MPDU é reconstruída num MSDU. Após a reconstrução do MSDU é efectuada a verificação do código de integridade de mensagens MIC. A verificação do MIC consiste no cálculo de um novo código de integridade de mensagem (MIC') sobre o endereço de origem ***SA***, endereço de destino ***DA***, campo ***Priority*** e o campo de dados. Em seguida compara o resultado obtido (MIC') com o código recebido (MIC).

Se o código de integridade calculado localmente for idêntico ao código da mensagem recebida, então a verificação ocorre com sucesso e o TKIP deve passar o MSDU ao nível protocolar superior. Se os dois diferirem então a verificação falha e o receptor deve descartar a trama recebida e aplicar as contramedidas referidas na secção 5.5.2.3.

Formato dos MPDUs

O formato de um MPDU gerado com o protocolo TKIP (Figura 5-83) difere de um MPDU WEP (Figura 4-12) em 4 bytes adicionados pelo TKIP para albergar a extensão ao vector de inicialização WEP. Esta extensão é designada no protocolo TKIP por ***Extended IV*** e aumenta o tamanho do MSDU em 8 bytes referentes ao novo campo ***MIC***. De referir que a adição destes 8 bytes não vão aumentar o tamanho máximo de um MSDU.

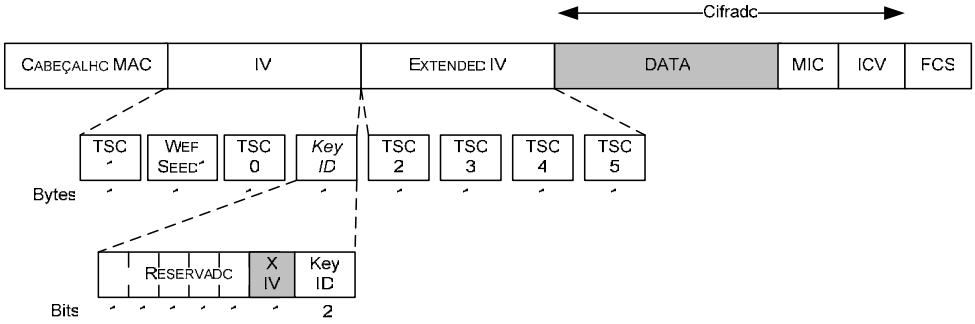


Figura 5-83 - Formato de um MPDU cifrado com TKIP.

O bit ***XIV*** do campo ***Key ID*** indica a presença ou não do campo ***Extended IV***. Se este bit possui o valor zero apenas o vector de inicialização não estendido é transferido. Se o bit ***XIV*** possui o valor um, então um vector de inicialização de 32 bits é associado ao vector de inicialização original. Para utilizar o TKIP, o bit ***XIV*** deve possuir o valor um e o campo ***Extended IV*** deve ser fornecido. Como foi referido na secção 4.2.2 para tramas cifradas com WEP os bits reservados e o bit ***XIV*** são designados por ***PAD*** e têm valor igual a zero. Tal como no WEP o campo ***Key ID*** armazena o valor do parâmetro *Key ID* que representa o índice de uma das quatro chaves secretas possíveis de definir no protocolo WEP. O campo ***WEP Seed1*** é o segundo byte do vector de inicialização representado na Figura 5-80, o seu valor é igual ao campo TSC1 excepto os bits 5 e 4 que são forçados a um e zero. Os campos ***TSC5*** e ***TSC0*** representam respectivamente o byte mais significativo e menos significativo do contador de sequência TSC. Os bytes **TSC0** e **TSC1** formam em conjunto parte do número de sequência do vector de inicialização e são utilizados na fase 2 da função de mistura de chaves. Os bytes **TSC2** a **TSC5** são utilizados na fase 1 da função de mistura de chaves, e em conjunto formam o campo ***Extended IV***. O campo ***Extended IV*** não deve ser cifrado. O TKIP deve cifrar todos os MPDUs gerados de um MSDU com a mesma chave do par temporária.

5.5.2.3 Código de Integridade de Mensagens – Michael

Antes de apresentar os detalhes do *Michael* é interessante rever o contexto no qual este mecanismo actua. Os ataques activos permitidos pelo desenvolvimento original do protocolo WEP incluem: alteração do conteúdo das mensagens, ataques iterativos para obtenção da chave WEP, redirecionamento das mensagens modificando os endereços de destino e finalmente ataques de personificação modificando os endereços de origem dos MPDUs. Estes

ataques operam ao nível do MPDU. No entanto o *Michael* aplica-se ao MSDU bloqueando com sucesso ataques ao nível do MPDU. O TKIP aplica o MIC ao MSDU no emissor da mensagem e verifica-o no receptor ao nível do MSDU. Se a verificação do código de integridade de mensagem ao nível MSDU falhar, a implementação deve descartar o MSDU e invocar as contramedidas necessárias.

O código de integridade de mensagem é calculado sobre o valor dos seguintes campos de um MSDU:

- Endereço de destino do MSDU, ***DA***;
- Endereço de origem do MSDU, ***SA***;
- ***Priority*** (reservado para uso futuro);
- Campo de dados em texto plano, ***Data***.

Os campos processados pelo código de integridade de mensagens *Michael*, são representados na Figura 5-84.

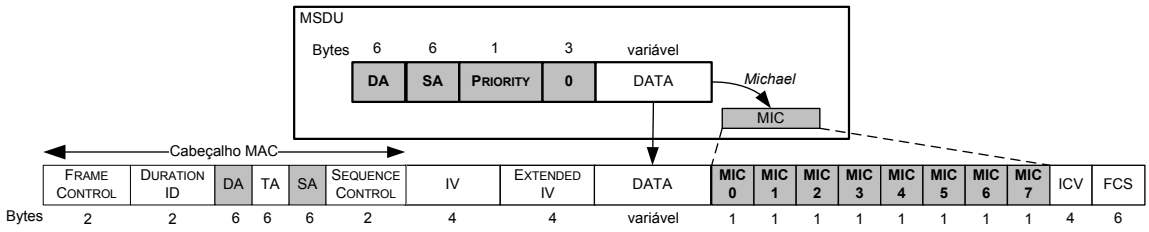


Figura 5-84 - Processamento TKIP MIC.

Os campos ***DA***, ***SA***, ***Priority***, os 3 ***Bytes*** do campo reservado (0) e o campo de dados ***DATA*** são utilizados para calcular o MIC. O campo ***Priority*** possui o valor zero e é reservado para utilização futura. O protocolo TKIP adiciona o código MIC com tamanho de 8 bytes no final do campo de dados do MSDU. O documento [55] descreve em detalhe a implementação do algoritmo *Michael*.

Contramedidas

O desenvolvimento do *Michael* comprometeu alguns aspectos de segurança em favor da capacidade de implementação em dispositivos já existentes no mercado. O *Michael* fornece um fraco nível de protecção contra ataques activos. De acordo com o grupo de trabalho IEEE 802.11i, sempre que o TKIP detectar uma falha na análise do MIC devem ser tomadas contramedidas. Estas contramedidas alcançam os seguintes objectivos:

- Registar os eventos de falhas no *Michael*. Normalmente estes são uma indicação da ocorrência de um ataque activo o que justifica uma análise por parte do administrador do sistema.
- Contabilizar a taxa de ocorrência de falhas. Esta ocorrências devem ser inferiores a duas por minuto, isto implica que os dispositivos wireless que detectem duas falhas no MIC num intervalo de tempo de 60 segundos devem desactivar a recepção de qualquer trama durante um período mínimo de 1 minuto.
- Não comprometer as chaves temporárias. Sempre que for detectada uma falha na análise do MIC deve iniciar-se um novo processo de negociação de chaves para a obtenção de novas chaves temporárias PTK e GTK.

A quantidade de eventos de falhas do MIC é acumulada independentemente da chave de sessão utilizada. Qualquer falha detectada por um ponto de acesso ou por uma estação é tratada como um evento de falha do MIC *Michael*. A estação cliente informa o ponto de acesso da ocorrência de um evento de falha através de uma trama de registo de falha do MIC. Esta é uma trama EAPoL com os bits ***MIC***, ***Error*** e ***Request*** a um.

Um suplicante deve activar as contramedidas definidas no TKIP em duas situações distintas: na recepção de uma trama com erro no código MIC e na recepção de uma primitiva designada por MLME-MICHEALMICFAILURE [33]. Esta primitiva é utilizada pela camada MAC 802.11 para indicar ao suplicante ou ao autenticador 802.1X uma falha no MIC.

As contramedidas adoptadas pelo suplicante são as seguintes:

1. Para um suplicante que recebe uma trama com erro no código MIC:
    a. Incrementar o contador de eventos de falha.
    b. Descartar a trama recebida com erro.
    c. Gerar uma primitiva MLME-MICHEALMICFAILURE.
2. Para um suplicante que recebe a primitiva MLME-MICHEALMICFAILURE:
    a. Enviar uma trama de registo de falha do MIC para o ponto de acesso.
    b. Se for a primeira falha no MIC no último minuto, inicalizar o temporizador de contramedidas.

c. Se tiver passado menos de 60 segundos desde o último evento de falha então as chaves temporárias (PTK) e (GTK) devem ser apagadas. Deve também se desautenticado o ponto de acesso e deve aguardar-se 60 segundos antes de inicializar um novo processo de troca de mensagens *4-way handshake*.

À semelhança do suplicante também o autenticador pode detectar eventos de falha do MIC e activar as respectivas contramedidas, que são as seguintes:

1. Para um autenticador que recebe uma trama com erro no código MIC:
    a. Descartar a trama recebida.
    b. Incrementar o contador de eventos de falha.
    c. Gerar uma primitiva MLME-MICHEALMICFAILURE.
2. Para um autenticador que recebe a primitiva MLME-MICHEALMICFAILURE:
    a. Se a trama recebida é uma trama de registo de falha do MIC, incrementar o contador de evento de falhas.
    b. Se for a primeira falha no MIC no último minuto, inicializar o temporizador de contramedidas.
    c. Se tiver passado menos de 60 segundos desde o último evento de falha então o autenticador deve desautenticar todas as estações cliente e eliminar as chave do par temporárias. A chave temporária de grupo actual deve ser descartada e uma nova GTK deve ser gerada mas não utilizada durante 60 segundos. O autenticador deve rejeitar durante 60 segundos qualquer tentativa de estabelecer novas chaves temporárias. No final dos 60 segundos, o autenticador aceita novamente processos para estabelecer as chaves temporárias.

As contramedidas implementadas pelo autenticador perante um evento de falha MIC poderão dar origem a um novo tipo de ataque de negação de serviço. Se um atacante enviar uma mensagem *multicast* forjada a cada 59 segundos a rede estará permanentemente desactivada. No entanto é importante compreender que na prática este tipo de ataque é extremamente complexo de ser executado [21]. Em primeiro lugar a trama deve ser forjada com um contador de sequência TSC correcto para que não seja descartada de imediato devido à recepção de uma

trama com TSC inválido. Como referido anteriormente o valor do TSC é utilizado como vector de inicialização para gerar uma chave de cifra por trama. Assim se existirem alterações ao TSC a trama não é decifrada correctamente pelo receptor. Neste caso o valor do ICV não será correcto pelo que a trama será descartada.

Para que um ataque de negação de serviço deste tipo possa ser implementado tem que ser capturada uma trama válida durante a transmissão, prevenir que esta seja entregue ao destino, alterar o MIC de modo a torná-lo inválido, calcular o ICV de acordo com a alteração efectuada e finalmente entregar a mensagem no destino para que o evento de falha MIC seja activado.

Conclui-se assim que este tipo de vulnerabilidade não é impeditivo para a implementação das contramedidas propostas no *Michael* pelo simples facto de que existem outros métodos bastante mais simples de implementar um ataque de negação de serviço, como por exemplo, o envio sucessivo de tramas de desassociação, tal como é descrito na secção 4.4.4.

5.5.3 AES-CCMP

Esta secção apresenta genericamente o protocolo AES-CCMP (*Advanced Encryption Standard - Counter mode CBC-MAC Protocol*).

Na secção 5.5.2 descreveu-se o algoritmo de cifra TKIP, o mecanismo de cifra e integridade das mensagens implementado no WPA e que é opcional nas redes RSN. O protocolo de segurança adoptado por omissão pelo IEEE 802.11i para implementação de mecanismos de cifra e integridade é o CCMP (*Counter mode CBC-MAC Protocol*) baseado no modo de operação CCM (*Counter with CBC-MAC*) [36] do algoritmo de cifra de blocos AES (*Advanced Encryption Standard*) [34].

5.5.3.1 Modos de Operação do AES

O AES é baseado no algoritmo de Rijndael, inventado por Joan Daeman and Vincent Rijmen. Este algoritmo está documentado em [35]. O algoritmo de Rijndael suporta uma variedade de tamanhos de blocos e de chaves. As escolhas possíveis para o tamanho de cada um destes parâmetros são 128, 160, 192, 224 e 256 bits (o tamanho do bloco e da chave pode ser diferente). Quando o NIST (*National Institute of Standards and Technology*) adoptou o Rijndael

para o AES, especificou apenas a cifra de bloco de tamanho 128 bits mantendo a opção de escolha do tamanho da chave em 128, 192 ou 256 bits. O IEEE 802.11i ao adoptar o AES restringe o tamanho da chave a 128 bits. Esta opção é justificada na facilidade de implementação do algoritmo evitando-se mais uma opção de configuração por parte dos utilizadores durante a fase de instalação.

As mensagens trocadas numa rede não-cablada não possuem comprimento fixo variando tipicamente entre 512 e 12000 bits. Consequentemente para fazer uso de uma cifra de blocos como o AES é necessário definir um método de converter mensagens de tamanho variável numa sequência de blocos de tamanho fixo antes do processo de cifra e vice versa. O método de conversão entre tramas de tamanho variável e blocos de comprimento fixo é designado modo de operação da cifra de blocos.

O AES define um conjunto de modos de operação com graus de complexidade distintos. O NIST apresenta em [Link7] uma lista de modos de operação do AES. O protocolo CCMP baseia-se no modo de operação CCM, que por sua vez recorre ao modo CTR (*Counter Mode*) [37]. Estes dois modos de operação da cifra de blocos AES são apresentados nas subsecções seguintes.

Modo de Operação Counter Mode – CTR

O princípio de funcionamento do modo de operação CTR, representado na Figura 5-85, passa por cifrar com o AES um valor aleatório designado *Counter* e em seguida submeter o resultado e um bloco da mensagem a uma operação de XOR. De referir que por uma questão de simplicidade não é representada na Figura 5-85 a chave de cifra temporária que é um parâmetro de entrada do AES.

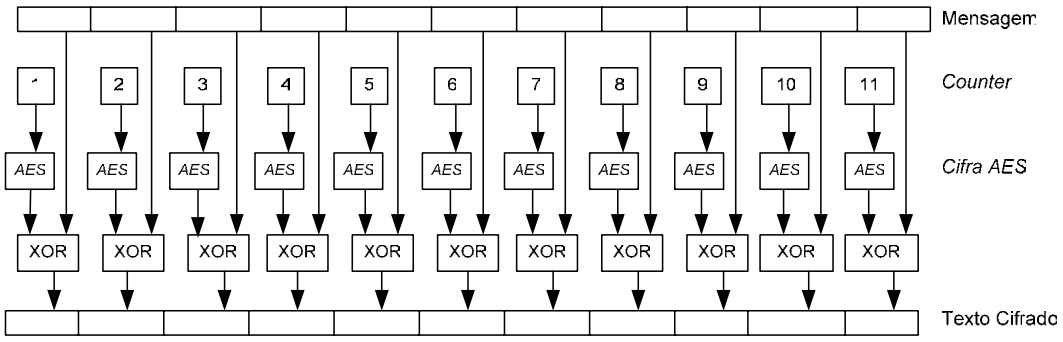


Figura 5-85 – Exemplo do modo de operação CTR.

No exemplo, o valor do *Counter*, é iniciado a 1 e é incrementado até ao valor 11. Na prática este valor não é obrigatoriamente iniciado a 1. O valor inicial do *Counter* pode ser arbitrário e o incremento pode ter um outro valor ou padrão. Geralmente a inicialização do *Counter* é feita com um valor diferente para cada trama transmitida de modo a evitar a cifra de duas mensagens idênticas com o mesmo valor do *Counter*. O importante é que o dispositivo receptor tenha conhecimento do valor inicial e das regras de incremento do valor do *Counter*. Em [37] são propostos métodos para o cálculo do valor inicial do *Counter* e respectiva função de incremento.

O recurso à operação de XOR torna o processo de cifra igual ao processo de decifra, uma vez que efectuando um XOR duas vezes a um mesmo valor é obtido o valor original. Tal facto permite aos dispositivos implementarem apenas a componente de cifra do AES, não sendo necessário implementar a componente de decifra.

Este modo de operação é utilizado há mais de vinte anos. É um método bastante conhecido e da confiança da comunidade cientifica. A sua simplicidade e robustez torna-o uma opção atractiva para implementação em redes RSN. No entanto não fornece qualquer tipo de autenticação de mensagens. O IEEE 802.11i chegou então à conclusão que seriam necessárias funcionalidades adicionais recorrendo para tal ao modo CCM abordado na subsecção seguinte.

Modo de Operação Counter with CBC-MAC – CCM

O modo de operação CCM foi desenvolvido especialmente para utilização em redes RSN. Foi submetido ao NIST como um modo genérico de operação do AES. Foi também submetido ao IETF através do documento [RFC3610]. Note-se que também na Figura 5-86 a chave de cifra temporária não é representada como parâmetro de entrada do AES.

O CCM combina o modo de operação CTR e o modo CBC-MAC (*Cipher Block Chaining - Message Authentication Code*) [38]. O primeiro fornece confidencialidade e o segundo autenticação e integridade. O princípio de funcionamento do modo de operação CCM, é representado na Figura 5-86.

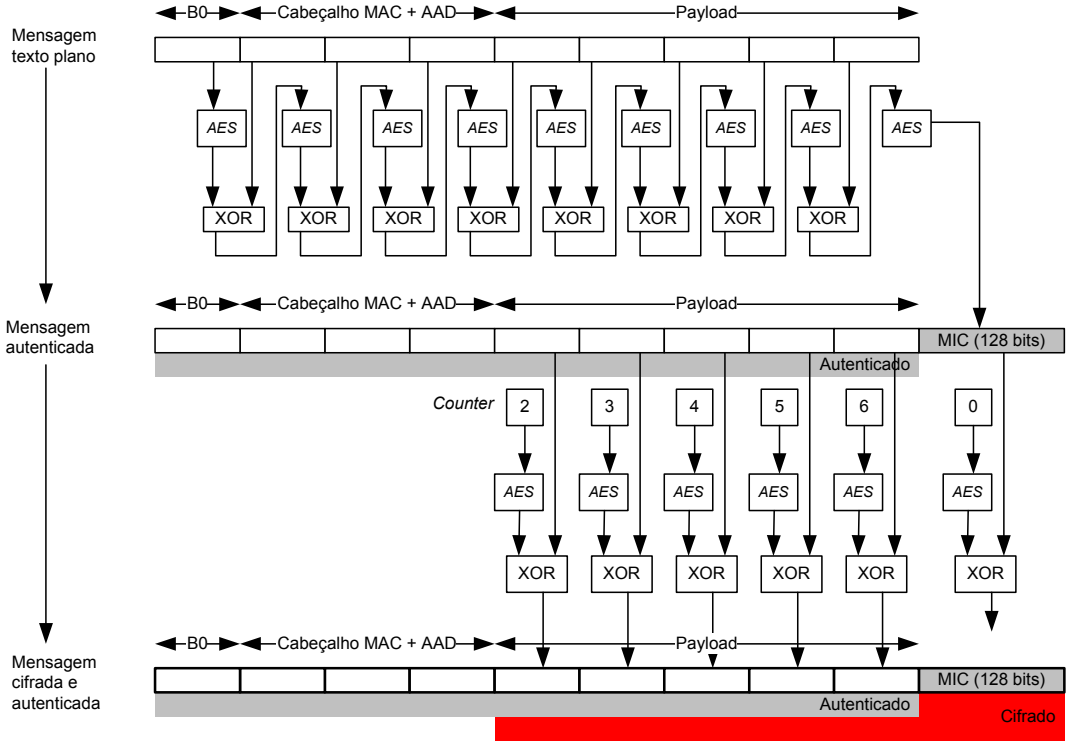


Figura 5-86 – Exemplo do modo de operação CCM.

O CCM fornece autenticação e integridade à mensagem, submetendo esta ao modo de operação CBC-MAC. O resultado desta operação é um código de integridade de mensagens MIC com comprimento de 128 bits, calculado sobre B0, o cabeçalho MAC, o *payload* da mensagem em texto plano e opcionalmente sobre o AAD (*Additional Authentication Data*).

O formato do bloco B0 é apresentado na Figura 5-87.

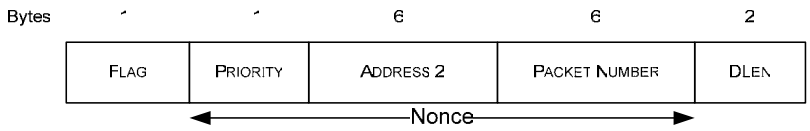


Figura 5-87 – Formato do primeiro bloco para modo CBC-MAC.

O campo ***Flag*** tem o valor fixo 01011001 para utilização em redes RSN. Para outras aplicações do CCM, este campo pode ter valores diferentes. Os campos ***Priority***, ***Address 2*** e ***Packet Number*** compõem o *nonce*. O *nonce* garante que cada mensagem cifrada com a mesma chave temporária utiliza alguma informação que nunca foi utilizada anteriormente. O campo ***DLen*** representa o comprimento dos dados da mensagem. O documento [38] define como opcional a utilização de blocos de dados adicionais de autenticação AAD. O grupo IEEE 802.11i propõe em [33] a construção do AAD com campos do cabeçalho MAC (Figura 5-88) .

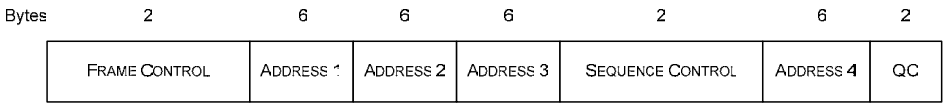


Figura 5-88 – Construção do AAD.

No AAD vários bits dos subcampos do campo ***Frame Control*** e do campo ***Sequence Control*** são colocados a zero devido à variação do seu valor durante o normal funcionamento da rede. Relativamente ao campo ***Frame Control*** os bits 4, 5 e 6 do subcampo ***SubType*** (Figura 3-10) são colocados a zero. Os bits ***Retry***, ***Power Management*** e ***More Data*** também são colocados a zero. Os bits 4 a 15 do campo ***Sequence Control*** são forçados com o valor zero. O AAD é ainda composto pelos campos de endereço ***Address 1***, ***Address 2***, ***Address 3*** e ***Address 4*** e pelo campo ***Quality of Service Control*** (QC) este campo é reservado para utilização futura.

Como se observa da Figura 5-86, o primeiro bloco é cifrado recorrendo ao AES e em seguida este resultado é submetido a uma operação de XOR com o segundo bloco. O resultado da operação de XOR é depois cifrado e submetido a nova operação de XOR com o próximo bloco. O processo repete-se até que todos os blocos sejam processados. O resultado final é concatenado à mensagem em texto plano para que esta seja submetida ao processo de cifra. A confidencialidade do modo CCM é obtida através do modo de operação CTR. Como se verifica pela Figura 5-86 toda a mensagem é cifrada (incluindo o MIC) excepto o cabeçalho.

O modo de operação CCM, por si só apenas pode ser aplicado a mensagens com um número de blocos múltiplo do comprimento do bloco. Para solucionar este problema o grupo IEEE 802.11i apresenta uma solução no protocolo CCMP baseada no *padding* da mensagem, isto é define um mecanismo que adiciona o número suficiente de bytes com valor igual a zero até que a mensagem tenha o comprimento desejado.

A utilização deste modo de operação proporciona algumas funcionalidades para as redes RSN, nomeadamente:

- Especificação de um valor distinto por trama cifrada com uma chave temporária. Assim duas mensagens sucessivas são cifradas com um valor de chave distinto.
- Utilização de apenas uma chave de cifra para fornecer integridade e confidencialidade das mensagens.

- Extensão do mecanismo de autenticação aos dados da mensagem que não podem ser cifrados.

Esta última funcionalidade tem particular interesse em redes não-cabladas. Por exemplo, o cabeçalho das tramas 802.11 contêm o endereço MAC que é sempre transmitido em texto plano. Como o modo CCM calcula o MIC sobre o cabeçalho da mensagem, tanto a autenticação como a integridade do endereço MAC são garantidas.

Abordados genericamente os modos de operação do AES utilizados pelo CCMP a secção seguinte apresenta o trabalho em desenvolvimento pelo IEEE 802.11i referente ao protocolo CCMP que será obrigatório nas redes RSN.

5.5.3.2 Protocolo CCMP

Esta secção descreve o protocolo CCMP. Os dados são cifrados ao nível do MPDU. A Figura 5-89 caracteriza um MPDU quando é utilizado o protocolo CCMP.

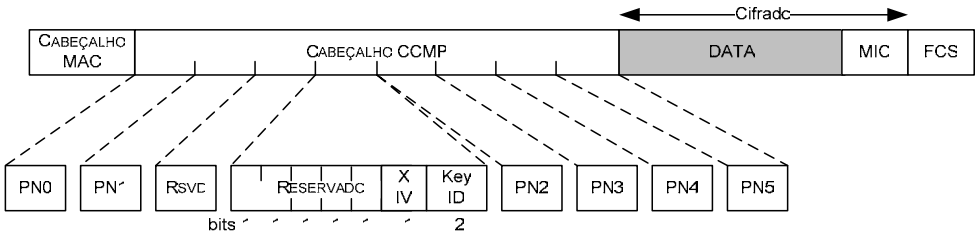


Figura 5-89 – Formato de um MPDU CCMP.

O CCMP expande o tamanho original de um MPDU em 16 bytes: 8 bytes constituem o cabeçalho do CCMP e os restantes 8 bytes são para o código de integridade da mensagem.

O cabeçalho CCMP tem duas funcionalidades. Em primeiro lugar fornece um valor de 48 bits que identifica o pacote, designado por PN (*Packet Number*), para protecção contra ataques de repetição. Segundo, no caso de tramas *multicast* o cabeçalho indica aos dispositivos receptores qual a chave de grupo utilizada.

Observando os 6 bytes que compõem o PN representados na Figura 5-89 tem-se que ***PN5*** é o byte mais significativo e ***PN0*** o byte menos significativo. O cabeçalho CCMP tem um campo reservado para uso futuro, ***Rsvd***. O campo ***KeyID*** contém informação relativa à identificação da chave. O campo ***ExtIV*** assinala que o cabeçalho CCMP aumenta o cabeçalho do MPDU em 8 bytes quando comparado com os quatro bytes acrescentados no caso da utilização do

WEP. O bit ***ExtIV*** tem o valor 1 quando o CCMP é utilizado. Os bits reservados para uso futuro têm valor zero e são ignorados pelo dispositivo receptor.

Processo de cifra CCMP

O processo de cifra de uma trama IEEE 802.11 recorrendo ao CCMP é representado na Figura 5-90.

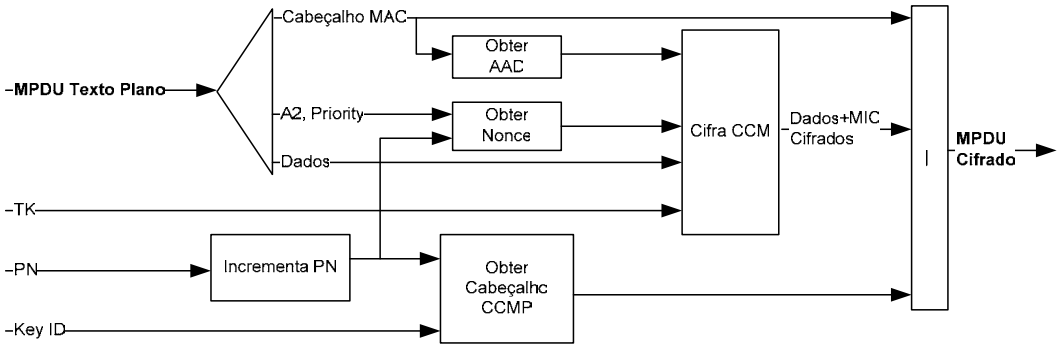


Figura 5-90 – Processo de cifra CCMP.

O CCMP cifra o *payload* de um MPDU em texto plano e encapsula o texto cifrado resultante, através do seguinte processo:

1. O valor do PN é incrementado para obter um novo valor PN por cada MPDU.
2. Os vários campos do cabeçalho MAC são utilizados para construir os dados adicionais de autenticação AAD para o modo de operação do CCM.
3. Um valor distinto *nonce* é gerado a partir do PN, do campo de endereço ***A2*** e do campo ***Priority***. O campo ***Address 2*** é representado na Figura 3-9. O campo ***Priority*** é um valor reservado para utilização futura com valor zero.
4. O novo valor do PN e a identificação da chave ***KeyID*** são inseridos no cabeçalho CCMP.
5. O valor da chave de cifra temporária TK, o valor *nonce*, os dados adicionais de autenticação AAD e os dados do MPDU são utilizados pelo modo CCM para cifrar o MIC e os dados da mensagem.
6. Finalmente, o cabeçalho CCMP é concatenado ao cabeçalho MAC e ao resultado da cifra CCM resultando no MPDU cifrado.

Processo de Decifra CMMP

A Figura 5-91 apresenta o diagrama de blocos que descreve o processo de decifra do protocolo CCMP.

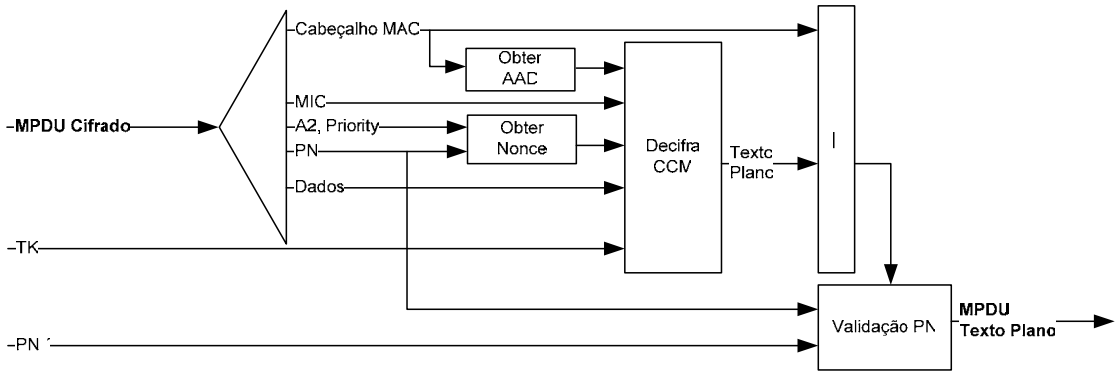


Figura 5-91 – Processo de decifra CCMP.

O CCMP decifra o *payload* de um MPDU e desencapsula o texto plano através do seguinte processo:

1. O MPDU cifrado é separado de modo a construir os valores dos dados adicionais de autenticação AAD e do valor aleatório *nonce*.
2. O AAD é construído a partir do cabeçalho MPDU do MPDU cifrado.
3. O *nonce* é calculado a partir do campo ***Address 2***, do PN, e do campo ***Priority***.
4. O valor do código de integridade de mensagem MIC é extraído para ser utilizado na verificação de integridade do algoritmo CCM.
5. O algoritmo CCM utiliza os valores da chave temporária TK, AAD, *nonce*, MIC e o campo do MPDU ***Data*** cifrado para obter os dados em texto plano e também para verificar a integridade do AAD e do texto plano obtido.
6. O cabeçalho do MPDU recebido e os dados em texto plano obtidos do algoritmo CCM podem ser concatenados para formar o MPDU em texto plano.
7. Finalmente o processo de decifra previne ataques de repetição, através da validação do valor do PN que deve ser superior ao contador *Replay Counter* mantido para a sessão.

O processo de decifra é bastante semelhante ao processo de cifra, permitindo a simplificação das implementações que surgirão no futuro. Uma vez obtido o MPDU, o MIC e o cabeçalho

CCMP podem ser removidos. Os restantes dados são facilmente adicionados com outros fragmentos de modo a reconstituírem o MSDU final.

6 Conclusões

Neste capítulo são apresentadas as principais conclusões resultantes do trabalho efectuado e são apontados os tópicos deixados em aberto e considerados importantes para trabalho futuro.

6.1 Principais Conclusões

A presente dissertação teve como principal objectivo o estudo de mecanismos de segurança implementáveis em redes não-cabladas IEEE 802.11. Dada a enorme diversidade de tecnologias de segurança existentes actualmente é extremamente difícil, senão impossível, garantir a adopção de uma solução de segurança para redes de comunicações totalmente segura e isenta de quaisquer falhas. Esta afirmação aplica-se integralmente às redes não-cabladas IEEE 802.11. Os inúmeros e distintos mecanismos de segurança apresentados nesta dissertação vêm comprovar que não existem garantias de que um mecanismo de segurança hoje considerado seguro, o será no futuro.

Numa primeira fase foi realizado um estudo dos mecanismos de segurança implementados pela norma IEEE 802.11. Durante o período inicial de estudo, um conjunto de vulnerabilidades foram apontadas aos mecanismos de autenticação, confidencialidade e integridade. Assim optou-se por uma pesquisa e validação prática dos possíveis ataques a que as redes não-cabladas IEEE 802.11 estavam sujeitas. Os mecanismos de autenticação aberta (*Open System*), autenticação de chave partilhada (*Shared Key*) e de controlo de acesso por endereços MAC assim como o mecanismo de confidencialidade baseado em chaves WEP estáticas, ofereciam um nível de segurança bastante rudimentar, como ficou demonstrado nas experiências laboratoriais efectuadas.

Estas conclusões não permitem afirmar que os mecanismo implementados pelo WEP são totalmente inúteis. Verificou-se que tomadas algumas medidas é possível minimizar os riscos associados aos ataques efectuados. Quando os dispositivos wireless não permitem a implementação de mecanismos de segurança recentes, ou os requisitos de segurança da rede não-cablada são mínimos, a implementação do protocolo WEP pode ser apropriada. Recomenda-se neste último caso a utilização de WEP com chave de tamanho 128 bits, de modo a dificultar ataques de força bruta. A alteração da chave secreta regularmente é outra das medidas que deve ser implementada.

À medida que as vulnerabilidades aos mecanismos de segurança foram sendo identificadas, novos mecanismos de segurança começaram a ser estudados e desenvolvidos. A segunda fase desta dissertação consistiu essencialmente num trabalho de pesquisa e análise destes novos mecanismos. Algumas das soluções estudadas devem ser apenas compreendidas como soluções intermédias, e não como soluções definitivas.

A adopção da tecnologia de redes privadas virtuais foi uma das primeiras alternativas ao WEP. Posteriormente surgiram propostas por parte dos principais fabricantes, nomeadamente os mecanismos de autenticação PEAP, EAP-TLS e EAP-TTLS baseados nos protocolos IEEE 802.1x e EAP. O estudo experimental destes protocolos de autenticação permitiu concluir que não existem diferenças significativas entre o PEAP e o EAP-TTLS. Ambos os protocolos não exigem certificados digitais para autenticação da estação cliente, ao contrário do método EAP-TLS, o que os torna do ponto de vista de gestão de redes wireless, uma solução mais interessante. Outra solução adoptada pelos fabricantes para melhorar o mecanismo de confidencialidade e integridade das redes não-cabladas foi o protocolo TKIP, cujo desenvolvimento teve em conta a compatibilidade com os dispositivos wireless existentes actualmente no mercado.

O estudo dos mecanismos de segurança alternativos efectuado nesta dissertação permite concluir que estes são considerados seguros, quando comparados com os mecanismos implementados pelo WEP. Como principal melhoria aponta-se o mecanismo de obtenção e gestão de chaves de cifra secretas proporcionados pelo WPA, ao contrário do WEP que apenas permitia a utilização de chaves de cifra estáticas.

Paralelamente, no âmbito do IEEE uma nova norma de segurança para redes não-cabladas vem sendo elaborada, a norma IEEE 802.11i. Esta norma define o conceito de rede robusta segura ou rede RSN. Os mecanismos de segurança das futuras redes RSN assentam

essencialmente no protocolo AES-CCMP que proporciona integridade e confidencialidade e no protocolo IEEE 802.1X que define uma estrutura de controlo de acesso e uma arquitectura que suporta um vasto conjunto de protocolos de autenticação como o EAP-TLS, PEAP ou EAP-TTLS.

De referir finalmente que a grande maioria dos dispositivos wireless utilizados actualmente nos diversos cenários de redes de comunicações não-cabladas IEEE 802.11 não são compatíveis com a futura norma IEEE 802.11i, devido aos requisitos de processamento de cálculo exigidos pelo AES.

6.2 Trabalho Futuro

Nesta dissertação, foram abordados vários mecanismos de segurança entre os quais o mecanismo de confidencialidade TKIP, e os mecanismos de autenticação PEAP e EAP-TTLS. Actualmente estes mecanismos encontram-se implementados por diversos fabricantes. No entanto existem pequenas diferenças entre as diversas implementações o que por vezes leva à incompatibilidade das implementações. Esta problemática deve-se ao facto destes mecanismos não serem actualmente considerados como normas efectivas mas antes trabalho em desenvolvimento. A abordagem efectuada aos mecanismos de segurança a implementar pela futura norma em desenvolvimento pelo grupo de trabalho IEEE 802.11i, teve apenas como objectivo introduzir conceitos que em breve deverão ser familiares aos responsáveis pela implementação de segurança em redes de comunicações não-cabladas. Por estas razões, será de extrema importância introduzir como tópico de trabalho futuro o acompanhamento e estudo exaustivo daqueles que serão os novos mecanismos de segurança da futura norma, nomeadamente os mecanismos de confidencialidade e integridade proporcionados pelo protocolo AES-CCMP.

Referências

[1] Al Petrick and Bob O'Hara, "*IEEE 802.11 Handbook: A Designer's Companion*", Standards Information Network IEEE Press, 1999.

[2] ANSI/IEEE Std 802.11-1999, "*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*" Institute of Electrical and Electronics Engineering, Inc, March 1999.

[3] Rui Jorge M.T. Valadas, "*Redes de Comuniações de Área Local Não-Cabladas por Raios Infra-Vermelhos*" Tese de Doutoramento, Universidade de Aveiro, pág. 144, Novembro 1995.

[4] IEEE Standards, "*Overview and Architecture*", 1990.

[5] Standards Information Network, "*IEEE Wireless LAN Edition A compilation based on IEEE Std 802.11TM-1999 (R2003) and its amendments*", Standards Information Network IEEE Press, 19 September 2003.

[6] B. Schneier, "*Applied Cryptography*", 2nd Edition, Wiley, 1997.

[7] Jesse R. Walker, "*Unsafe at any key size; An analysis of WEP encapsulation*", IEEE doc 802.11-00/362, October 2000.

[8] Nikita Borisov, Ian Goldberg, David Wagner, "*Intercepting Mobile Communications The insecurity of 802.11*", International Conference on Mobile Computing and Networking Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 180-189, 2001.

[9] William A. Arbaugh, Narenda Shankar, and Y.C.Justin Wan, "*Your 802.11 Wireless Network Has No Clothes*", Department of computer Science, University of Maryland, March 2001.

[10] Scott Fluhrer, Itsik Mantin, Adi Shamir, "*Weakness in key scheduling RC4 algorithm*", Lecture Notes in Computer Science, 2001.

[11] Adam Stubblefield, John Ioannidis, Aviel D. Rubin, "*Using the Fluhrer, Mantin, and Shamir Attack to Break WEP*", AT&T Labs Technical Report TD-4ZCPZZ, August 21, 2001.

[12] E. Dawson and L. Nielsen, "*Automated cryptanalysis of XOR plaintext strings*", Cryptologia, pp. 165-181, April 1996.

[13] Cyrus Peikari and Seth Fogie, "*Maximum Wireless Security*", Sams, December 1, 2002.

[14] B. Schneier and Mudge, *"Cryptanalisys of Microsoft´s Point-to-Point Tunneling Protocol (PPTP)".* Proceeding of the 5th ACM Conference on Communications and Computer Security, ACM Press, pp. 132-141, 1998.

[15] Microsoft Corporation. "*Microsoft Privacy Protected Network Access: Virtual Private Networking and Internet Security*", Technical Report, Microsoft, 1999.

[16] IEEE Std 802.11i/D7.0**, "***Draft Amendment to ISO/IEC 8802-11/1999(I) ANSI/IEEE Std 802.11, 1999 edition*", October 2003.

[17] IEEE P802.1X, "*Standards for Local and Metropolitan Area Networks: Standard for Port Based Network Access Control*", October 25 2001.

[18] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, "*Diameter Base Protocol*", RFC3588, September 2003.

[19] L. Blunk , J. Vollbrecht , Bernard Aboba , "*Extensible Authentication Protocol (EAP)*", draft-ietf-pppext-rfc2284bis-07.txt, October 2002.

[20] IEEE P802.1aa/D6.1, "*Draft Standard for Local and Metropolitan Area Networks—Port Based Network Access Control—Amendment 1: Technical and Editorial Corrections*", June 16, 2003.

[21] Jon Edney, William A. Arbaugh, "*Real 802.11 Security Wi-Fi Protected Access and 802.11i*", Addison-Wesley Pub Co, 1st edition, July 2003.

[22] Cisco Systems, Inc., "*Extensible Authentication Protocol Transport Layer Security Deployment Guide for Wireless LAN Networks"*, 2002.

[23] Cameron Macnally, "*Cisco LEAP protocol description*", 2001.

[24] Product Bulletin Nº 2331, "*Cisco Response to Dictionary Attacks on Cisco LEAP*", Cisco Systems, Inc., 2003.

[25] Joshua Wright, "*Abusing 802.11: Weaknesses in LEAP Challenge/Response*", Defcon, 2003.

[26] Sharad Ahlawat, "*Weakness in LEAP Challenge/Response*", Security Focus, October 2003.

[27] Ashwin Palekar , Dan Simon, Glen Zorn, Joe Salowey, Hao Zhou, S. Josefsson, "*Protected EAP Protocol (PEAP) Version 2*", draft – josefsson-ppext-tls-eap-07.txt, 26 October 2003.

[28] Paul Funk, Simon Blake-Wilson, "*EAP Tunneled TLS Authentication Protocol (EAP-TTLS)*", draft-ietf-pppext-eap-ttls-05.txt, July 2004.

[29] Niels Ferugson, MacFergus, "*Michael: an improved MIC for 802.11 WEP*", IEEE doc 802.11-02/020r0, Jan 17, 2002.

[30] Russ Housley, Doug Whiting, "*Temporal Key Hash"*, IEEE doc 802.11-01/550r3, December 20, 2001.

[31] Jonathan Hassel, "*RADIUS- Securing Public Access to Private Resources*" O'Reilly & Associates, 1st edition, October 8, 2002.

[32] B. Aboba, P. Calhoun, "*RADIUS Support for Extensible Authentication Protocol (EAP)*", 2 November 2002.

[33] IEEE Std. 802.11i/D8.0**, "***Draft Amendment to ISO/IEC 8802-11/1999(I) ANSI/IEEE Std. 802.11, 1999 edition*", February 2004.

[34] FIPS197, "*Advanced Encryption Standard*", Federal Information Processing Standard, May 2002.

[35] Daemen, J., and V. Rijmen. 2001. Rijndael, "*The Advanced Encryption Standard*", Dr. Dobb's Journal, Vol.26, No.3, pp. 137–139, March 2001.

[36] R. Housley, D. Whiting, N. Ferguson, "*Counter with CBC-MAC*", June 3 2002.

[37] Morris Dworkin, "*Recommendation for Block Cipher Modes of Operation – Methods and Techniques* ", NIST Special Publication 800-38a, pp.15-16, December 2001.

[38] Bellare, M., J. Kilian, and P. Rogaway, "*The security of the cipher block chaining message authentication code*", Journal of Computer and System Sciences Vol.61, No.3, pp. 362–399, 2000.

[39] D. Kahn, "*The Codebreakers, the story of secret writing*", The Macmilliam Company, 1967.

[40] S. Singh, "*The Codebook: The Evolution of secrecy from May, Queen of scots to Quantum Cryptography*", Doubleday Press, 1999.

[41] C. Kaufman, R. Perlman, M. Spencier, "*Network Security, Private Communication in a Public World*", Pretince Hall, Englewood Cliffs, 1995.

[42] National Institute of Standards and Technology, "*Data Encryption Standard*", Processing Standards Publication, pp. 46-2, 1993.

[43] Federal Information Processing Standards Publication 81, "*DES Modes of Operation*", 1980.

[44] E. Biham and A. Shamir, "*Differential cryptanalysis of the full 16-round DES, Advances in Cryptology*", Springer-Verlag, pp. 487-496, 1993.

[45] W. Diffie and M.E. Hellman, "*New directions in cryptography, IEEE Transactions on Information Theory*", pp. 644-654, 1976.

[46] R.L. Rivest, A. Shamir, and L.M. Adleman, "*A method for obtaining digital signatures and public-key cryptosystems*", Communications of the ACM, pp. 120-126, 1978.

[47] National Institute of Standards and Technology (NIST), "*Announcement of Weakness in the Secure Hash Standard*", 1994.

[48] W. Diffie and M.E. Hellman, "*New directions in cryptography, IEEE Transactions on Information Theory*", pp. 644-654, 1976.

[49] Microsoft Corporation "*Overview of Certificates and Authentication*", 1997.

[50] Kohnfelder, L, "*Towards a Practical Public-Key Cryptosystem*", M.I.T., May 1978.

[51] Denning, D., "*Protecting Public Keys and Signature Keys*", Computer, February 1983.

[52] William A. Arbaugh, "*Wireless Security Is Different*", Univ. of Maryland, August 2003.

[53] Vivek Kamath, Ashwin Palekar, Mark Wodrich, "*Microsoft's PEAP version 0 (Implementation in Windows XP SP1*)", Microsoft, 25 October 2002.

[54] Valadas, R.; A. Moreira; C. Lomba; A. Tavares; A. M. O. Duarte; " *The IEEE 802.11 Infrared Physical Layer* ", IEEE Comm. Magazine , Vol. 36 , No. 12 , pp. 107 - 112 , December , 1998 .

[55] Niels Ferugson, "*Michael: an improved MIC for 802.11 WEP*", IEEE doc 802.11-02/020r0, January 2002.

[56] Federal Information Processing Standards Publication 74, "*Guidelines for Implementing and Using the NBS Data Encryption Standard*", 1981.

[57] Eli Biham, Adi Shamir, "*Differential Cryptanalysis of the Data Encryption Standard*", Springer Verlag, 1993.

[58] Mitsuru Matsui, "*The First Experimental Cryptanalysis of the Data Encryption Standard*", CRYPTO 1994: pp1-11

[59] Luis Carlos Azevedo, "*Infra-estructuras de Chave Pública*", Dissertação de Mestrado em Engenharia Electrónica e de Telecomunicações, Universidade de Aveiro, 2001.

[60] Joseph Davies, Elliot Lewis, "*Deploying Virtual Private Networks with Microsoft Windows Server 2003*" Microsoft Press, 2004.

[61] Charlie Scott, Paul Wolfe, Mike Ervin, "*Virtual Private Networks 2nd Edition*", O´Reilly, 1999.

[62] IEEE P802.1x-REV/D11 "*Draft Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control (Revision)",* July 22, 2004.

[63] Robert Moskowitz "*Wekness in Passphrase Choise in WPA Interface*", ICSA Labs, November 2003.

[64] Scott Fluhrer, Itsik Mantin, Adi Shamir "*Attacks On RC4 and WEP*", Cryptobytes 2002.

[65] Paula Thomas, Khalid AL-Begain, John Hughes "*The impact of security measures on the performance of IEEE802.11 networks* ", 3rd Internacional Workshop in Wireless Security Technologies, April, 2005.

[66] Jenne Wong "*Performance Investigation of Secure 802.11 Wireless Lans: Raising the Security Bar to Wich Level?*" Master of Commerce in Accountancy, Finance and Information Systems, University of Canterbury, 2003.

[Link1] http://airsnort.shmoo.com/

[Link2] http://sourceforge.net/projects/wepcrack

[Link3] http://www.wi-fi.com/OpenSection/index.asp

[Link4] http://www.ethereal.com/

[Link5] http://www.kismetwireless.net/

[Link6] http://802.11ninja.net/airjack/

[Link7] http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/

[Link8] http://www.faqs.org/faqs/cryptography-faq/part10/

[Link9] http://www.itsecurity.com/tecsnews/sep2003/sep193.htm

[Link 10] http://amac.paqtool.com/

[Link 11] http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/

[Link 12] http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt

Acrónimos

ACL	*Access Control List*
AES	*Advanced Encryption Standard*
AH	*Authentication Header*
AP	*Access Point*
AVP	*Attribute Value Pair*
BSA	*Basic Service Area*
BSS	*Basic Service Set*
CA	*Certification Authority*
CBC-MAC	*CBC Message Authentication Code*
CDPD	*Cellular Digital Packet Data*
CHAP	*Challenge Handshake Authentication Protocol*
CRC	*Cyclic Redundancy Code*
CSMA/CD	*Carrier Sense Multiple Access with Collision Detection*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
CTR	*Counter mode*
DCF	*Distributed Coordination Function*
DES	*Data Encryption Standard*
DNS	*Domain Name System*
DoS	*Denial of Service*
DS	*Distribution System*

DSSS *Direct Sequence Spread Spectrum*

EAP *Extensible Authentication Protocol*

EAPoL *EAP over LANs*

ESP *Encapsulation Security Payload*

ESS *Extended Service Set*

FEC *Forward Error Correction*

FHSS *Frequency Hopping Spread Spectrum*

GMK *Group Master Key*

GRE *Generic Routing Encapsulation*

GSM *Global System for Mobile communication*

GTK *Group Transient Key*

IAPP *Inter Access Point Protocol*

IBSS *Independent Basic Service Set*

ICV *Integrity Check Value*

IEEE *Institute of Electrical and Electronic Engineers*

IETF *Internet Ingineering Task Force*

IP *Internet Protocol*

IPSec *Internet Protocol Security*

IPX *Internet Packet Exchange*

IR *Infra-Red*

ISM *Industrial, Scientific and Medical*

ITU-T *International Telecommunications Union-Telecommunication*

IV *Initialization Vector*

KSA *Key Scheduling Algorithm*

L2TP *Layer 2 Tunnelling Protocol*

LAN	*Local Area Network*
LLC	*Logical Link Control*
MAC	*Medium Access Control*
MD5	*Message Digest 5*
MIC	*Message Integrity Code*
MPDU	*MAC Layer Protocol Data Unit*
MSDU	*Medium Access Control Service Data Unit*
NIST	*National Institue of Standards and Technology*
OFDM	*Orthogonal Frequency Division Multiplexing*
PAE	*Port Access Entity*
PAP	*Password Authentication Protocol*
PCMCIA	*Personal Computer Memory Card International Association*
PMK	*Pairwise Master Key*
PPP	*Point to Point Protocol*
PPTP	*Point to Point Tunnelling Protocol*
PRNG	*Pseudo-Random Number Generator*
PSK	*Pre-Shared Key*
PTK	*Pairwise Transient Key*
RF	*Radio Frequency*
RFC	*Request For Comments*
RSN	*Robust Security Network*
SHA-1	*Secure Hashing Algorithm 1*
SS	*Station Service*
SSID	*Service Set Identifier*
STA	*Station*

TCP	*Transmission Control Protocol*
TKIP	*Temporal Key Integrity Protocol*
TSN	*Transition Security Network*
UNII	*Unlicensed National Information Infrastructure*
VPN	*Virtual Private Network*
WEP	*Wires Equivalent Privacy*
WECA	*Wireless Ethernet Compatibility Alliance*
Wi-Fi	*Wireless Fidelity*
WLAN	*Wireless Local Area Network*
WM	*Wireless Medium*
WPA	*Wi-Fi Protected Access*
WPAN	*Wireless Personal Area Network*
WWAN	*Wireless Wide Area Network*