Instituto Politécnico do Porto

Redes, comunicações e segurança informática

Jorge Pinto Leite

Trabalho apresentado para a satisfação parcial dos requisitos do grau de Especialista

Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto

Porto – Julho de 2013

I. Resumo

Este trabalho começa por apresentar uma instalação de rede informática local efetuada sob supervisão do candidato e parcialmente por ele, descrevendo de seguida algumas considerações que na altura se lhe levantaram. Sendo ligada à saúde, para além das preocupações normais de instalação de qualquer solução, a questão humana e social foi particularmente sentida.

Várias organizações e pessoas individuais necessitam de implementar e utilizar soluções de comunicações informáticas, pelo que a privacidade e confiabilidade das informações processadas se torna um assunto premente.

No sentido de obter uma resposta para este problema, o candidato evoluiu para a área da criptografia e segurança de comunicações, implementado alguns dos protocolos existentes e estudando-os com maior rigor.

É justamente a descrição dessa evolução e da segurança com que se pode contar quando se utilizam meios informáticos o âmbito deste trabalho, focando principalmente o impacto que tem na sua utilização.

Palavras Chave: segurança; criptografia; lan

II. Índice geral

III. Lista de figuras

IV. Lista de tabelas

V. Lista de acrónimos não *standard* utilizados

AES	Advanced Encryption Standard
AH	Authentication Header
AVP	Attribute-Value Pair
CHAP	Challenge-Handshake Authentication Protocol
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DCE	Data Communication Equipment
DTE	Data Terminal Equipment
ESP	Encapsulating Security Payload
GRE	Generic Routing Encapsulation
IANA	Internet Assigned Numbers Authority
ICL	International Computers Ltd
IEFT	Internet Task Engineering Force
IP	Internet Protocol
IPsec	IP security
IV	Initialization Vector
L2F	Layer 2 Forwarding Protocol
L2TP	Layer 2 Tunneling Protocol
MAC Address	Endereço físico de uma placa de rede
MAC	Message Authentication Code
MD5	Message Digest 5
MPPE	Microsoft Point-to-Point Encryption
MS-CHAP	Microsoft CHAP

MTU	Maximum Transfer Unit
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
Radius	Remote Authentication Dial In User Service
RFC	Request For Comments
RS232	Norma de interligação entre um DTE e um DCE definido pela TIA
Rx	Receive
SA	Security Association
SAD	Security Association Database
SHA	Secure Hash Algorithm
SPD	Security Policies Database
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TLS	Transport Layer Security
TLSHP	TLS Handshake Protocol
TLSRP	TLS Record Protocol
Tx	Transmit
UDP	Used Datagram Protocol
VPN	Virtual Private Network
X.509	Certificado de chave pública

1. Introdução

No âmbito de uma das ocupações profissionais desempenhadas pelo candidato em 1996, foi responsável pela instalação da rede informática do Hospital de Mirandela e pela formação dos seus utilizadores, bem como, depois, pela assistência técnica de segunda linha à rede informática e à aplicação integrada hospitalar.

O objetivo final era a informatização e interligação dos dados entre todos os serviços existentes: urgência, consulta externa, imagiologia, serviço de atendimento permanente, análises (laboratório), sem esquecer os serviços não orientados ao utente mas que se tornam imprescindíveis tais como o economato, farmácia, arquivo, faturação, entre outros.

A dimensão física do Hospital de Mirandela não é minimamente comparável com a de hospitais centrais como o Hospital de S. João no Porto, contudo é mesmo assim considerável. Atendendo à maior disponibilidade de terreno disponível, são os hospitais do interior normalmente construídos sob a forma plana e não em altura. O Hospital de Mirandela era composto por uma estrutura mista, uma vez que o bloco central era constituído por 5 ou 6 pisos de altura, dividindo-se ainda por quatro edifícios independentes. A estrutura do hospital representa uma facilidade em muitos aspetos orientada para o utente, nomeadamente pela facilidade de locomoção. No entanto, uma estrutura deste tipo pode representar diversos inconvenientes quando orientado às tecnologias, como por exemplo a inerente maior dimensão física das cablagens associadas.

À data (1996) ainda as empresas que comercializavam equipamentos informáticos, concretamente a ICL (*International Computers Limited*) de que a empresa em que o candidato trabalhava era agente, instalavam tecnologias proprietárias. No caso concreto e no que toca à rede local instalada, a solução recomendada pela ICL e implementada consistia numa tecnologia de rede local denominada *Microlan*.

As redes *Microlan* são, ou eram já que a ICL foi adquirida por uma outra multinacional tendo acabado por desaparecer, uma tecnologia tipo CSMA/CD[1] em barramento onde os terminais (*dummys* – não inteligentes) eram conectados através de caixas de derivação com um máximo de dois terminais por caixa. Cada caixa continha para além das tomadas de ligação dois *dip switch*, um de cada lado e onde encaixavam os condutores do cabo *Microlan*, devendo na última caixa o *dip switch* oposto, onde não há cabo interligado, ser ativado. O comprimento máximo de cabo era 1000m o que permitia à partida e sem necessidade de mais equipamentos uma dimensão real da rede local bastante elevada face aos parâmetros atuais. A performance da rede *Microlan* era bastante razoável para a tecnologia existente na altura, e a elevada dimensão máxima de cabo tornava-a suficiente para praticamente todas as implementações de rede.

Um problema que se levantou, mal a instalação estava efetuada e testada, foi devido ao pedido da Administração para tentar aproveitar todo ou parte de um sistema anterior que

[1] *Carrier Sense Multiple Access with Collision Detection*

nunca tinha sido instalado. O referido sistema consistia num servidor com um sistema operativo baseado em Unix e terminais não inteligentes tipo RS232[2]. A Administração solicitou durante a instalação da solução o aproveitamento da maior parte desse equipamento.

Aproveitando as portas RS232 existentes na solução instalada concluiu-se que os terminais não funcionavam quando ligados ao sistema então instalado, apenas quando ligados ao sistema cujo *hardware* se tentava aproveitar.

Socorrendo-se nesta altura de um analisador de sinais RS232 detetou-se que se tratava apenas de uma troca dos pinos *Transmit* (Tx) e *Receive* (Rx) face ao *standard*. A solução encontrada que a primeira vista parece simples, mas não evidente, foi a de incluir mais portas RS232 no novo sistema e alterar os cabos de forma a refletir essa troca. A única desvantagem é que os terminais reaproveitados precisavam estar numa zona relativamente próxima do servidor. Uma cablagem de maior qualidade, que permitiria em teoria uma dimensão máxima de cabo na ordem dos 300m, seria necessária para ultrapassar esta limitação, mas era insustentável, quer pelo custo à data do cabo em si, quer pela passagem de cabos em tubagens no interior das paredes e que não tinham diâmetro para esse efeito. Ainda se equacionou a instalação de *multiplexers*/*demultiplexers* mas também o custo à data o tornava proibitivo.

A solução encontrada, para resolver o referido problema, foi o de utilizar as portas RS232 disponíveis, tendo-se construído também um cabo adicional para permitir a interligação direta entre o novo sistema e o antigo. Através dos comandos de programação de tarefas (*cron*[3]) e de um *script* criado para o efeito configurou-se o sistema antigo para que a horas determinadas efetuasse uma ligação ao novo sistema e copiasse para o seu disco as cópias de segurança da aplicação hospitalar lá existentes, diferenciando-as por data. O objetivo desta diferenciação era garantir a possibilidade de se consultar ou recuperar informação com uma data razoavelmente anterior à data atual – dependendo apenas do volume de dados gerados em cada dia e da capacidade do disco do sistema antigo.

O conjunto destas questões e numa altura em que a Internet era quase incipiente em Portugal (se bem que já existisse) levou o candidato a considerações relativas à confidencialidade e segurança da informação, principalmente, e como é natural dado o local onde a solução foi implementada, na temática da saúde.

A principal questão recaia sobre "*Qual a dificuldade de algum elemento aceder à rede e obter acesso a dados não autorizados?*". É claro que à data, utilitários como *sniffers*[4] não estavam ainda disponíveis, pelo menos da forma como estão atualmente, não havia interligação à Internet e o acesso de estranhos às zonas do Hospital onde se encontravam terminais, tal como sucede hoje, era bastante condicionado.

[2] *Standard* definido pela TIA (*Telecommunications Industry Association*) para interligação entre dispositivos, normalmente um DTE (*Data Terminal Equipment*) e um DCE (*Data Communication Equipment*)
[3] Sistema de agendamento de ações existente em sistemas operativos tipo Unix
[4] Aplicações que capturam a informação que circula numa rede, permitindo a sua análise

Num campo diametralmente oposto a essas preocupações era um problema grave um utente deslocar-se a um Hospital fora da sua área habitual de residência e/ou trabalho – em gozo de férias, por exemplo – e ser complicado (nalgumas situações, impossível) para o médico que o atendesse obter um historial médico completo do utente.

Estas duas considerações são no entender do candidato opostas, mas necessárias. *Opostas* porque a privacidade do utente pode ficar comprometida caso o seu registo de saúde eletrónico seja capturado por terceiros, *necessárias* porque o desconhecimento de eventuais problemas que o utente tenha – como alergias, por exemplo – pode levar a medicamentação incorreta. Com esse objetivo o candidato preparou uma apresentação que vaticinava um conjunto de sistemas distribuídos que guardassem o historial clínico de cada utente e que poderia mediante graus de autorização definidos ser consultado por qualquer médico do País. A apresentação deveria ter sida apresentada num congresso de funcionários informáticos hospitalares organizado pelo Hospital de Macedo de Cavaleiros, que esteve agendado mas que acabou por não se realizar.

Entretanto, deu-se a explosão da infraestrutura *Internet Protocol* (IP) e da Internet. Por essa altura já as redes proprietárias tinham caído praticamente em desuso e todas as empresas apostavam nos *standards* como o modelo *Transmission Control Protocol* (TCP)[5].

A evolução da experiência profissional do candidato, nomeadamente no apoio a inúmeras instalações de sistemas de comunicação e de redes locais (muitas das quais em gabinetes de radiologia, outras na extensão da rede ou dos serviços existentes em hospitais como o de Oliveira de Azeméis, Vila Nova de Gaia, Macedo de Cavaleiros, entre outros), encaminhou-o no sentido em se especializar nas novas tecnologias de redes locais ligadas à internet, onde os servidores centrais com terminais não inteligentes foram sendo progressivamente substituídos por redes de computadores pessoais, com ou sem servidor central, e para o domínio do endereçamento IPv4. Durante esta evolução, várias foram as tecnologias de rede local em muitas das quais o candidato esteve envolvido se bem que em redes de dimensão inferior à acima descrita, como a Novell[6], que instalou em empresas comerciais, e a Lan Manager[7], utilizada em soluções de retalho. No entretanto, esteve também envolvido em sistemas abertos, tendo sido inclusive convidado para testar algumas delas (Lindows[8] por exemplo, no que tocava à integração com sistemas Windows).

Mas as considerações relativas à segurança, confidencialidade e privacidade nunca o abandonaram. A evolução do candidato deu-se então no estudo e análise de sistemas de segurança em redes informáticas em geral, e nas comunicações em particular. Essa experiência levou a que fosse convidado para lecionar aulas no Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto, normalmente em áreas ligadas a redes de computadores (Redes de Computadores 1 e 2, mais tarde, após a introdução do Tratado de

[5] RFC793, http://www.rfc-editor.org/pdfrfc/rfc761.txt.pdf
[6] http://www.novell.com
[7] Sistema operativo de rede desenvolvido pela Microsoft (http://www.microsoft.com) e pela 3Com Corporation (http://´www.3com.com)
[8] Sistema operativo com base em Unix que pretendeu competir contra a Microsoft; posteriormente alterou o seu nome para *Linspire* e finalmente foi adquirida e desapareceu

Bolonha, Redes de Computadores) e à administração de sistemas informáticas (Administração de Sistemas Informáticos 2, mais tarde Administração de Sistemas).

Uma das áreas de especialização do candidato é focada nos protocolos de segurança mais usuais (PPTP, L2TP, IPsec e SSL/TLS) tendo implementado para efeitos de teste, demonstração e análise alguns deles, nomeadamente o PPTP e o IPsec.

De salientar a preocupação do candidato em aplicar a sua experiencia em prol da sociedade a titulo voluntário, nomeadamente presta apoio informático na infraestrutura de rede e comunicações do Banco Alimentar Contra a Fome do Porto, rede com domínio Windows 2008 Server R2 (à data), e acessos remotos acessíveis a apenas alguns dos seus colaboradores.

Nesta atividade, mas desta vez não relativamente a questões ligadas à saúde, também agora se levantam questões relativas à segurança, confidencialidade e privacidade dos dados. Apesar de ser um sistema completamente diferente, também é verdade que existem hoje em dia vários *scripts* disponíveis na Internet fáceis de descarregar e de usar por motivos por vezes ilegais.

Dentro da evolução do candidato para questões relativas à segurança, adotou a trilogia das propriedades de confidencialidade[9], integridade[10] e disponibilidade [2], resultando a confidencialidade da garantia de a mensagem só ser percetível para o seu destinatário, a integridade da garantia de a mensagem ser realmente proveniente de quem o diz ser e o seu conteúdo ser o que foi realmente enviado, e a disponibilidade da garantia de que um sistema a que se pretende aceder está em condições de responder a tal acesso.

De facto estes três vetores são cruciais para um pleno funcionamento confiável de um sistema. A título de exemplo, se uma comunicação for intercetada por alguém e adulterada na sua forma ou conteúdo, é claramente uma situação de quebra da integridade da mensagem, ou seja, uma quebra de segurança já que o objetivo dessa adulteração pode ser induzir em engano o destinatário. Essa possível alteração da informação implica que a mensagem rececionada pelo destinatário não seja proveniente de quem ele espera que seja mas do atacante, o que viola a autenticidade da mesma uma vez que já não é exatamente a mensagem que o sistema devolveu. Por força da sensibilidade de uma informação – como por exemplo, a troca de credenciais para obter acesso a um sistema ou a troca de dados com uma instituição ligada à saúde, entre outros exemplos – também a confidencialidade é intuitiva. Por último, a indisponibilidade de um sistema fere a sua utilização inibindo os utilizadores legítimos do mesmo de lhe obterem acesso e, por inerência, efetuarem as operações pretendidas.

[9] A confidencialidade engloba a privacidade, o secretismo e o anonimato.

[10] A integridade engloba a autenticidade, a autenticação, a autorização de acesso e o controlo de acessos; alguns autores incluem a integridade isoladamente e a autenticidade como um quarto vetor, que inclui a autenticação, a autorização de acesso e o controlo de acessos. Neste trabalho optámos por considerar os vetores definidos em [2].

De forma a definir uma base para uma análise dos protocolos de segurança atuais e da sua interoperabilidade, é feita a definição do *Protocol Data Unit (*PDU) dos níveis 2 e 3 do modelo *Open Systems Interconnection* (OSI), respetivamente a *frame* e o pacote.

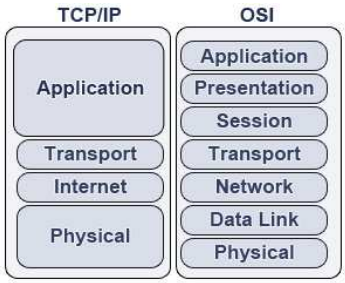

Figura 1 - Modelo OSI e modelo TCP
(Fonte: http://www.learn-networking.com))

O modelo de referência OSI nunca foi implementado realmente uma vez que pretendia ser apenas um modelo de referência. No entanto, serviu como base para o modelo TCP, também representado na Figura 1, que é utilizado atualmente.

Os níveis do modelo OSI (Figura 1) são normalmente designados por um número crescente desde o nível 1 (Físico) até ao nível 7 (Aplicação). Cada nível comunica com o nível imediatamente inferior e superior, adicionando ou retirando a informação que lhe diz respeito (associada ao controlo da informação), operação essa designada por encapsulamento e desencapsulamento, respetivamente. Os níveis Físico a Transporte são habitualmente apelidados de níveis baixos (*lower layers*) sendo os restantes apelidados de níveis altos (*higher layers*). Cada um dos níveis possui um PDU próprio para a informação após encapsulada pelos dados de controlo desse nível.

Mau grado o modelo existente ser o TCP que, como se observa na Figura 1 acima, possui como nível mais baixo um nível Físico que corresponde sensivelmente aos níveis Físico e Enlace de Dados do modelo OSI, a designação de *frame* para o PDU desse nível mantém-se.

O formato original da *frame* foi definido na RFC1222[11] e para as redes Ethernet encontra-se representado na Figura 2.

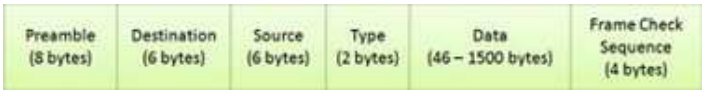

Figura 2 - Formato da frame
(Fonte: http://learncomputernetwork.blogspot.com)

O cabeçalho da *frame* é constituído pelos quatro primeiros campos da Figura 2, com as seguintes funções:

[11] http://tools.ietf.org/html/rfc1122, último acesso em 9 de Junho de 2010

- O campo *Preamble*, com uma dimensão de 8 bytes com o padrão 10101010, tem como objetivo sincronizar o *clock* entre o emissor e o recetor; usando a codificação Manchester (utilizada na Ethernet atual) a 10Mbps, este padrão gera uma onda quadrada de 10MHz;
- O campo *Destination* contém o endereço físico (***MAC Address***) do sistema a quem a frame se destina;
- O campo *Source* contém o endereço físico (***MAC Address***) do sistema que emitiu a *frame*;
- O campo *Type* identifica o protocolo encapsulado.

Segue-se ao cabeçalho da *frame* o campo *Data* (muitas vezes representado como *Payload*) que contém a informação do nível superior, no caso do modelo OSI, o nível de Rede. Este campo tem um limite mínimo e máximo para a sua dimensão, como se observa na Figura 2, podendo atualmente conter uma dimensão superior à mostrada como se explicará mais abaixo.

Finalmente, temos o *Frame Check Sequence* com 4 bytes. Este campo contém um *Cyclic Redundancy Check* (CRC) aplicado ao valor dos campos *Destination*, *Source*, *Type* e *Data*. O objetivo deste campo é permitir a deteção de erros de transmissão.

Como já referido, o único campo com tamanho variável mas com um limite máximo e mínimo é o campo *Data*. A RFC1222 determina que o tamanho máximo de um pacote (*Maximum Transfer Unit* MTU) é de 1500 bytes. No entanto, este valor deve ser entendido como um máximo e não como o tamanho que deve sempre ter, ou seja, se a informação que chega ao nível Enlace de Dados, constituída pelo cabeçalho IP e pelos dados das camadas superiores for superior ao MTU definido, o pacote é sucessivamente fragmentado em blocos com o tamanho do MTU até que seja integralmente transmitido. O último fragmento, tal como informação proveniente do nível de rede que não atinja o tamanho do MTU, é enviado numa *frame* cujo tamanho, excluído do cabeçalho do nível 2, será inferior.

Estudos efetuados [4] demonstram que o MTU mais eficiente para o estado atual de tecnologia é superior aos 1500 bytes, surgindo as designadas *Jumbo Frames* e *Super Jumbo Frames*, as primeiras com um MTU até 9000 bytes e as segundas que excedam essa dimensão.

O objectivo deste documento é então o de apresentar o estudo realizado pelo candidato no contexto de análise e apreciação dos protocolos de segurança atualmente existentes.

O documento é organizado da seguinte forma. Na primeira secção é feita uma introdução aos motivos que levaram ao interesse pela problemática da segurança. Na segunda secção descrevem-se os protocolos de segurança *standard* mais vulgares. Na terceira secção é feita a descrição da interoperabilidade entre eles. Na quarta secção efetua-se uma análise comparativa dos protocolos descritos na segunda secção e finalmente na quinta secção apresenta-se as conclusões.

2. Protocolos de segurança

Os protocolos de segurança normalizados e actualmente utilizados são o *Point-to-Point Tunneling Protocol* (PPTP), *Layer 2 Tunneling Protocol* (L2TP), *Internet Protocol Security* (IPsec) e *Secure Sockets Layer* (SSL). Este último foi desenvolvido pela Netscape e uma evolução da sua versão 3.0 foi padronizada sob a designação *Transport Layer Security* (TLS).

Para além da análise dos protocolos normalizados – excluindo-se portanto o SSL – interessou-nos a análise da sua interoperabilidade, isto é, a possibilidade e eventuais problemas inerentes à utilização conjunta de mais do que um deles em simultâneo, e o aumento da dimensão de uma mensagem (*overhead*) devido à sua utilização.

2.1 PPTP

O *Point-to-Point Tunneling Protocol* (PPTP) surgiu com o objectivo de permitir o transporte do *Point-to-Point Protocol* (PPP)[12] em redes IP possibilitando assim a ligação de um equipamento a um servidor, num ambiente cliente-servidor.

2.1.1 Descrição

O protocolo PPTP [5] é um protocolo que permite que conexões PPP (*Point-to-Point Protocol*) sejam encapsuladas numa rede IP, criando uma *Virtual Private Network* (VPN)[6].

Antes da disponibilização universal de infraestruturas de rede baseadas no protocolo IP, os sistemas remotos podiam-se conectar ligar a redes através do *Point-to-Point Protocol* (PPP). Servindo bem para esse objetivo, estas ligações dispunham de um conjunto de limitações associadas ao canal de comunicação que utilizavam, como por exemplo permitir um e um só sistema em comunicação. O protocolo PPTP, teve como objetivo permitir que as ligações PPP pudessem tirar partido da infraestrutura IP emergente.

Desenvolvido por um grupo de empresas que incluía a Microsoft e a 3Com entre outros, incluiu dois componentes [7]:

- Transporte, que mantém a conexão virtual; e
- Encriptação, que garante a confidencialidade.

O PPTP trabalha encapsulando os pacotes protegidos da VPN em pacotes PPTP que por sua vez são encapsulados em pacotes tipo *Generic Routing Encapsulation* GRE[13] que são enviados sobre IP da origem para o servidor de encaminhamento PPTP [6]. A estrutura de um pacote IP

[12] http://www.rfc-editor.org/rfc/rfc1172.txt
[13] Protocolo de encapsulamento e encaminhamento desenvolvido pela Cisco (http://www.cisco.com)

obtido por estes sucessivos encapsulamentos está representada na Figura 3 (por *Media Header* entende-se o cabeçalho específico do meio de comunicação).

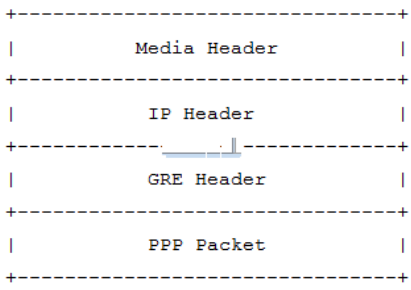

Figura 3 - Estrutura de um pacote PPTP transmitido
(Fonte: RFC2637, disponível em http://www.rfc-editor.org/rfc/rfc2637.txt)

Em conjunção com este canal de dados encapsulados existe uma sessão de controlo em TCP (esta sessão é utilizada para controlo de estado e para troca de sinalização entre os sistemas) [6].

A definição do PPTP não especifica algoritmos específicos para autenticação e encriptação. Ao invés, define um ambiente para negociação dos algoritmos.

Embora a definição do protocolo deixe espaço para qualquer tipo imaginável de encriptação e autenticação, a maior parte dos produtos comerciais utilizam a versão desenvolvida pela Microsoft para o Windows NT [6].

2.1.2 Formato

O formato do PPTP depende do tipo de mensagem que está a ser trocada [5]. De comum entre todos eles é o cabeçalho cuja composição se encontra na Tabela 1.

Tabela 1 - Cabeçalho PPTP

16	32 bits
Length	PPTP message type
Magic cookie	
Control message type	Reserved 0

O campo *Length* contém o comprimento total em bytes da mensagem PPTP incluindo o próprio cabeçalho.

O campo *PPTP message type* contém um identificador numérico com o valor 1 (um) se se trata de uma mensagem de controlo ou 2 (dois) se se trata de uma mensagem de gestão (estas mensagens não foram ainda definidas).

O campo *Magic Cookie* contém sempre o valor **1A2B3C4D**(16). Esta constante é usada para permitir que o sistema recetor se assegure que está corretamente sincronizado com o fluxo de dados TCP.

O campo *Control message type* contém um identificador numérico com os valores de 1 (um) a 15 (quinze) de acordo com o tipo de mensagem atual. Os tipos de mensagem definidos estão representados na Tabela 2.

Finalmente, o campo *Reserved 0* contém sempre obrigatoriamente o valor zero.

Tabela 2 - Tipos de mensagens PPTP definidas

Control	Message Code
(Control Connection Management)	
Start-Control-Connection-Request	1
Start-Control-Connection-Reply	2
Stop-Control-Connection-Request	3
Stop-Control-Connection-Reply	4
Echo-Request	5
Echo- Reply	6
(Call Management)	
Outgoing-Call-Request	7
Outgoing-Call-Reply	8
Incoming-Call-Request	9
Incoming-Call-Reply	10
Incoming-Call-Connected	11
Call-Clear-Request	12
Call-Disconnect-Notify	13
(Error Reporting)	
WAN-Error-Notify	14
(PPP Session Control)	
Set-Link-Info	15

Uma vez que o PPTP é encapsulado dentro de um pacote tipo GRE (aliás, a RFC2637[14] denomina-o de *enhanced GRE*) importa ter em mente a sua constituição, estando o seu cabeçalho representado na Figura 4.

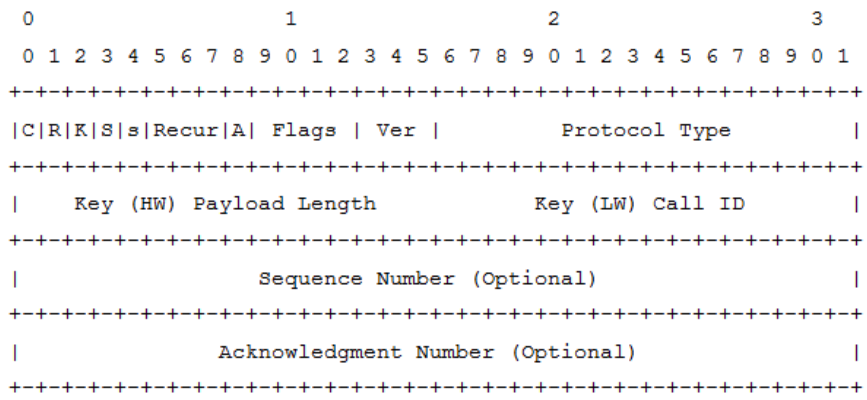

Figura 4 - Cabeçalho do pacote GRE utilizado no PPTP
(Fonte: RFC2637, disponível em http://www.rfc-editor.org/rfc/rfc2637.txt)

Dos campos não opcionais que constituem o cabeçalho GRE chama-se a atenção que alguns deles possuem um valor fixo e pré-determinado (caso dos campos **C**, **R**, **s**, **Recur** e **Flags** que contém o valor zero, os campos **K** e **Ver** que contém o valor um e o campo **Protocol Type** que contém o valor **0x880B** (16). O campo **S** conterá o valor zero se não existir informação a ser transmitida nesse pacote – caso em que a informação se limita a um **acknowledgment** – ou um se o pacote transmite dados. Da mesma forma, o campo **A** conterá o valor um se o campo **Acknowledgment Number** estiver preenchido para confirmar receção de dados anteriormente enviados.

Note-se que os campos **Sequence Number** e **Acknowledgment Number** surgem como opcionais na definição mas apenas o são realmente para mensagens de controlo. Nos pacotes que possuem dados enviados estes campos existem e são necessários, uma vez que um dos motivos para o duplo encapsulamento (primeiro PPTP e depois GRE) é permitir que as confirmações possam ser enviadas conjuntamente com mais dados, o que implica um melhor aproveitamento do canal de comunicação e, por inerência, maior eficiência do protocolo.

Ressalta-se que a única função do GRE é permitir o encapsulamento num formato compatível com uma infraestrutura de rede IP.

2.1.3 Overhead

Dado que o PPTP não especifica concretamente quais os algoritmos a usar para autenticação e encriptação, o *overhead* que implica é sempre dependente dos algoritmos escolhidos numa determinada implementação. Mas há sempre dois cabeçalhos que necessariamente são adicionados ao pacote IP de dados: o cabeçalho PPTP e o cabeçalho GRE.

[14] http://www.rfc-editor.org/pdfrfc/rfc2637.txt.pdf

No seu conjunto, estes cabeçalhos implicam um conjunto de dados adicionais de 28 bytes (16 pelo encapsulamento no pacote GRE e 12 pelo encapsulamento no PPTP).

2.1.4 Vantagens e desvantagens

Provavelmente uma das principais vantagens deste protocolo seja o relativamente baixo *overhead* que implica já que a ocupação da capacidade do canal de comunicação devida ao seu uso é mínima. A sua simplicidade e facilidade de utilização (pode ser utilizado por qualquer sistema) são também uma vantagem significativa já que não implica, de per si, a necessidade de qualquer sistema ou equipamento adicional. Uma outra vantagem é implicar um comportamento em rede como se o sistema remoto estivesse fisicamente ligado à rede a que se liga via VPN (vantagem que não é exclusiva do PPTP, como veremos adiante). Contudo, podemos equacionar a utilidade desta característica, isto é, pense-se por exemplo nas mensagens enviadas em *broadcast* (por exemplo, as associadas à resolução de nomes) numa rede que também serão encapsuladas no túnel PPTP [1]; poderá equacionar-se se o benefício devido ao pouco *overhead* não é desprezável pelo normalmente elevado número de mensagens habitualmente existentes

A generalização presente na definição deste protocolo – note-se que a sua especificação não refere os algoritmos a utilizar, como já referido – apresenta, do ponto de vista do candidato, uma desvantagem importante, uma vez que se torna demasiado dependente da implementação específica efetuada – e a única existente e vulgarmente utilizada é a implementada pela Microsoft.

Schneier [6] apresenta diversas vulnerabilidades nesta implementação. Nas suas conclusões refere que esta implementação *"…é frágil de um ponto de vista de implementação e apresenta sérias falhas graves de um ponto de vista de protocolo*". Frisa porém que a análise criptográfica efetuada não conseguiu quebrar o protocolo PPTP mas apenas esta implementação.

No entanto, note-se também que o PPTP permite a cifra dos dados em trânsito mas não a integridade dos mesmos [1] o que inviabiliza quer a correção dos dados quer a autenticidade dos mesmos. Além disso, não contempla um mecanismo de distribuição entre os intervenientes de chaves de sessão a ser usada pela cifragem.

A cifra é aplicada após implementação do protocolo de autenticação escolhido, o que significa que o processo de implementação propriamente dito não é protegido bem como as negociações de configuração do PPTP [1].

Finalmente, uma desvantagem do PPTP que não deve ser menosprezada é só funcionar sobre uma rede IP, o que impede a sua utilização noutros tipos de rede. Não há dúvida que a maioria das redes funciona sobre este protocolo, contudo e de um ponto de vista do candidato não se deve por isso considerar que é o único disponível.

Numa última conclusão, as implementações do PPTP preocupam-se com a confidencialidade e a autenticidade do emissor da mensagem mas não com a sua integridade nem com a

disponibilidade dos sistemas envolvidos na comunicação. Na implementação da Microsoft o algoritmo de autenticação é o *Microsoft Challenge-Handshake Authentication Protocol* (MS-CHAP) e o de cifra o *Microsoft Point-to-Point Encryption* (MPPE). Mau grado as versões mais atuais de ambos terem sido melhoradas em relação a defeitos funcionais prévios, alguns problemas subsistem. A autenticação MS-CHAP, por exemplo, é efectuada no início da comunicação considerando-se autenticados os interlocutores a partir desse momento (permitindo pois que um atacante capture a informação a partir desse instante e assuma a identidade de um deles). Na versão original deste protocolo as palavras-chave eram enviadas como um *hash*[15] em formato LAN Manager (que possui uma proteção fraca) e em formato Windows NT (mais forte). Na versão atual o formato LAN Manager já não é enviado, mantendo-se contudo a autenticação apenas no instante de estabelecimento da comunicação.

O MPPE utiliza por sua vez o *hash* da autenticação para efetuar a encriptação dos dados. Dois problemas se levantam com esta opção. Por um lado, mesmo sem descodificar a autenticação é possível capturar essa informação e utilizá-la de seguida para codificar comunicações forjadas, violando desta forma a segurança do canal. Por outro lado existem aplicações que permitem descodificar palavras-passe do Windows NT.

Algumas formas de intrusão formas de intrusão em comunicações protegidas por esta versão do PPTP são descritas no artigo *Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)* [8].

2.2 L2TP

O *Layer 2 Tunneling Protocol* (L2TP) foi definido com o objectivo de permitir o transporte do *Point-to-Point Protocol* (PPP)[16] entre redes IP. Complementava assim o objectivo do PPTP que se destinava a uma ligação cliente-servidor.

2.2.1 Descrição

O L2TP é um protocolo orientado à conexão que foi definido com o objetivo de facilitar a segurança de comunicações PPP de uma forma transparente quer para os utilizadores quer para as aplicações [9]. Um dos objetivos que presidiu ao seu desenvolvimento foi aproveitar os benefícios associados aos protocolos PPTP (secção 2.1) e *Layer 2 Forwarding Protocol* (L2F), um protocolo de *tunneling* desenvolvido pela Cisco[17], possibilitando dessa forma a utilização de qualquer tipo de infraestrutura de rede, baseada em IP ou não – ultrapassando dessa forma uma das limitações do PPTP.

[15] Resultado da aplicação de um método de cifra irreversível com o objetivo de detetar alterações à mensagem original
[16] http://www.rfc-editor.org/rfc/rfc1172.txt
[17] http://www.cisco.com/en/US/tech/tk827/tk369/tk387/tsd_technology _support_sub- protocol _home.html

O L2TP utiliza dois tipos de mensagens, as de controlo e as de dados. As primeiras (que utilizam o porto 1701 em TCP) são utilizadas para o estabelecimento, manutenção e limpeza (*reset*) dos túneis, enquanto as segundas (que utilizam o mesmo porto mas em *User Datagram Protocol (*UDP) são utilizadas para encapsular as *frames* PPP que são transportadas pelo túnel criado. As mensagens de controlo utilizam um canal isolado e de confiança dentro do L2TP para garantir a entrega das mensagens. Por seu lado, as mensagens de dados que sofram perda de informação não são retransmitidas, cabendo aos protocolos de nível superior detetarem e solicitarem a retransmissão das mensagens perdidas.

As mensagens de controlo utilizam números de sequência para garantir a entrega ordenada destas mensagens. As mensagens de dados podem ter ou não números de sequência que, caso existam, servirão para ordenação no recetor e facilitar a deteção de pacotes perdidos.

A Figura 5 mostra a estrutura do protocolo L2TP. Os dados provenientes do protocolo PPP são capturados e encapsulados com um cabeçalho L2TP, sendo enviados sobre um canal não confiável. As mensagens de controlo são transmitidas sobre um canal L2TP confiável.

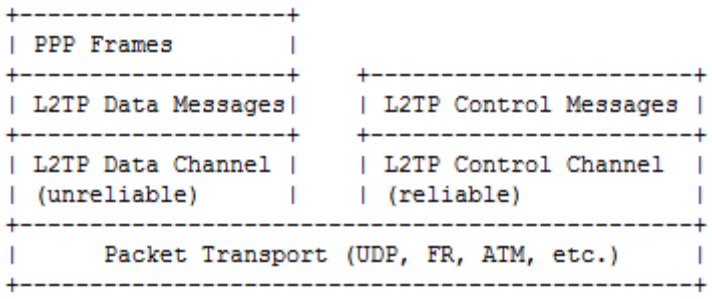

Figura 5 - Estrutura do protocolo L2TP
(Fonte: RFC2637, disponível em http://www.rfc-editor.org/rfc/rfc2637.txt)

2.2.2 Formato

O formato do cabeçalho L2TP é comum entre as mensagens de controlo e as mensagens de dados, sendo a sua estrutura apresentada na Figura 6.

O campo **T** indica o tipo da mensagem. É colocado em zero para uma mensagem de dados e em um para uma mensagem de controlo.

O campo **L** tem obrigatoriamente o valor de um para uma mensagem de controlo, sendo opcional para as mensagens de dados.

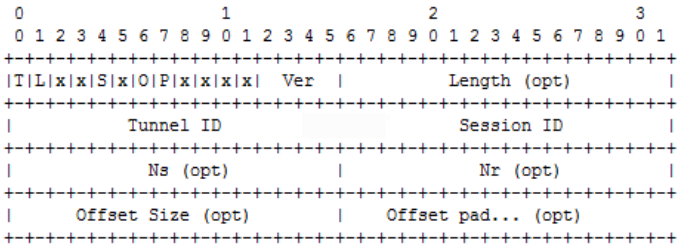

Figura 6 - Cabeçalho L2TP
(Fonte: RFC2637, disponível em http://www.rfc-editor.org/rfc/rfc2661.txt)

Os campos indicados por **x** são reservados para extensões futuras.

O campo **S**, obrigatoriamente com o valor de um para mensagens de controlo, indica a presença ou ausência (caso contenha o valor zero) dos campos **Ns** e **Nr**.

O campo **O** indica a presença (valor um) ou ausência (valor zero) do campo **Offset Size**. Nas mensagens de controlo contém o valor zero.

O campo **P** indica a prioridade da mensagem, sendo válido apenas para as mensagens de dados, caso em que contém o valor de um. Nesta situação a mensagem deverá beneficiar de um tratamento preferencial na fila de espera de transmissão. Para as mensagens de controlo tem que conter o valor zero.

O campo **Ver** indica a versão do protocolo, devendo conter o valor de dois. O valor de um é permitido para efeitos de deteção de pacotes do protocolo L2F. Qualquer outro valor presente neste campo implica a rejeição da mensagem.

O campo **Length** indica o tamanho total da mensagem em bytes. Este campo só estará preenchido se o campo **L** contiver o valor um, sendo irrelevante o seu conteúdo no caso contrário. Como o campo **L** é obrigatório para mensagens de controlo, intui-se que nessas mensagens o campo **Length** é também obrigatório.

O campo **Tunnel ID** contém o identificador para a conexão de controlo. Os túneis L2TP são identificados por um valor que só tem significado local, ou seja, os dois extremos da comunicação podem ter um identificador diferente para a mesma conexão. O valor presente neste campo é o do identificador do recetor, não o do emissor. A criação do identificador em cada extremo da comunicação é efetuada durante a criação do túnel sendo divulgado ao outro extremo como *Assigned Tunnel ID AVPs*.

O campo **Session ID** tem um significado similar ao anterior. Contém o identificador da sessão estabelecida pelo sistema recetor dentro do túnel. Os identificadores de sessão são próprios de cada sistema e só possuem significado local. A criação de uma sessão dentro do túnel é efetuada em cada um dos extremos após criação do túnel e apenas quando se pretende enviar uma comunicação, sendo trocadas entre os extremos como *Assigned Session ID AVPs*.

O campo **Ns** indica o número de sequência da mensagem atual, iniciando-se em zero e sendo incrementada de um módulo 2^{16} em cada mensagem subsequente. Juntamente com o campo

Nr e também opcional, o seu propósito é providenciar um mecanismo confiável para as mensagens de controlo.

O campo **Nr** contém o número de sequência esperado na próxima mensagem de controlo, sendo sempre incrementado por um módulo 2^{16}. Note-se que este campo só tem um valor obrigatório para as mensagens de controlo (aliás é opcional para as mensagens de dados). Caso exista nas mensagens de dados – o que é indicado pelo campo **S** – deve ser ignorado após receção da mensagem.

O campo **Offset Size**, caso presente, indica o byte após o cabeçalho L2TP a partir do qual (inclusive) a mensagem se inicia.

O campo **Offset Pad** tem como única função alinhar o tamanho do cabeçalho L2TP sendo o seu conteúdo indefinido e ignorado.

Tabela 3 - Tipos de mensagens de controlo do L2TP

Control Connection Management	
0	(reserved)
1 (SCCRQ)	Start-Control-Connection-Request
2 (SCCRP)	Start-Control-Connection-Reply
3 (SCCCN)	Start-Control-Connection-Connected
4 (StopCCN)	Stop-Control-Connection-Notification
5	(reserved)
6 (HELLO)	Hello
Call Management	
7 (OCRQ)	Outgoing-Call-Request
8 (OCRP)	Outgoing-Call-Reply
9 (OCCN)	Outgoing-Call-Connected
10 (ICRQ)	Incoming-Call-Request
11 (ICRP)	Incoming-Call-Reply
12 (ICCN)	Incoming-Call-Connected
13	(reserved)
14 (CDN)	Call-Disconnect-Notify
Error Reporting	
15 (WEN)	WAN-Error-Notify
PPP Session Control	

16 (SLI)	Set-Link-Info

As mensagens de controlo possuem vários tipos, indicados na Tabela 3.

Um método uniforme de codificação das mensagens de controlo está considerado na definição para maximizar a extensibilidade do protocolo mantendo a interoperabilidade entre as versões e evoluções. A este método foi atribuído o nome *Attribute-Value Pair* (**AVP)**.

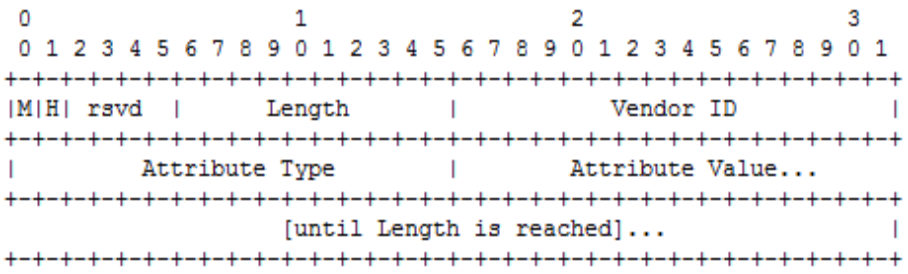

Figura 7 - Codificação AVP
(Fonte: RFC2637, disponível em http://www.rfc-editor.org/rfc/rfc2637.txt)

A Figura 7 mostra a codificação associada a este método. Os primeiros seis bits (campos **M**, **H**, e **rsvd**) descrevem os atributos gerais da AVP. Como se observa na Figura 7 apenas dois bits são utilizados à data, já se intui que o acrónimo **rsvd** significa reservado para extensões futuras (a definição especifica contudo que este campo deve conter o valor zero em todos os bits, indicando que qualquer AVP que não contenha esse valor no campo deve ser tratado como um AVP não reconhecido).

Os campos **M** e **H** podem ser utilizados para controlo de segurança da comunicação.

O campo **M** especifica o tratamento que deve ser dado a um AVP rececionado e não reconhecido. Se este bit estiver ativo num AVP não reconhecido a sessão – se se tratar de uma mensagem associada a uma sessão – ou o túnel – se se tratar de uma mensagem associada ao túnel criado – devem ser de imediato terminadas, o que, no caso do túnel, implica a finalização imediata de todas as sessões que existirem. Caso o campo não esteja ativo, o campo deve ser ignorado processando-se a mensagem de controlo normalmente.

O campo **H**, caso ativo, indica que o campo **Attribute Value** está encriptado. Esta opção permite evitar que informação sensível, como uma palavra-chave, por exemplo, seja transmitida em texto claro. Este campo deve estar ativo apenas se um segredo partilhado existe entre o emissor e o recetor. Este segredo partilhado é o mesmo que é utilizado para autenticação do túnel. O conteúdo do campo **Atribute Value** é obtido pela aplicação do algoritmo criptográfico irreversível MD5 ao qual é por sua vez aplicada uma operação lógica XOR.

É interessante notar que a definição prevê a utilização de um vetor de inicialização (*Initialization Vector* IV) para o cálculo de um *hash* pela aplicação do *Message Digest 5 (*MD5), método pelo qual a aplicação da cifra em bloco evita que blocos iguais produzam cifras iguais. O valor do vetor de inicialização inicial é definido numa mensagem obrigatória que precede o

primeiro AVP, sendo o seu valor para a operação do bloco N + 1 o resultado da operação XOR do bloco N.

A criação de um túnel é a primeira operação a ser efetuada numa comunicação L2TP e consiste numa sequência de mensagens de controlo SCCRQ, SCCRP, SCCCN e ZLB ACK, descritas na Tabela 3. A autenticação do túnel é opcional e baseada no protocolo CHAP [10]. Após criação do túnel podem ser estabelecidas sessões no túnel.

Para obviar períodos de inatividade do túnel, são enviadas mensagens de controlo **HELLO** que servem para detetar quebras de comunicação no túnel (caso em que o túnel e todas as sessões que se encontrem estabelecidas são terminadas).

A definição do L2TP não sofreu alterações desde a sua criação em 1999, existindo apenas extensões definidas para os vários canais de comunicação existentes (caso do FRAME RELAY), métodos alternativos de autenticação (caso do RADIUS) ou diferenciação de serviços.

2.2.3 Overhead

O *overhead* implícito na utilização do L2TP é de dezasseis bytes devidos apenas ao seu cabeçalho. Dado que todas as comunicações que possam existir e assumindo a utilização exclusiva do L2TP, este valor é devido ao cabeçalho identificador do túnel e da sessão existente. Contudo e mau grado ser objetivo do L2TP substituir várias conexões PPTP por uma só L2TP este valor será inerente a toda e qualquer mensagem que seja trocada pelo túnel.

2.2.4 Vantagens e desvantagens

O L2TP isolado apresenta o *overhead* mais baixo dos protocolos analisados. A possibilidade da sua utilização sobre qualquer tipo de canal de comunicação – ao contrário do PPTP – é também um factor que não deve ser desprezado. Sendo uma evolução natural do PPTP, a disponibilização de multi-sessão sob o túnel criado é uma vantagem significativa já que permite que equipamentos remotos à rede constituam uma rede encapsulada no L2TP (rede a rede) ao contrário do PPTP onde tal cenário obrigava a uma ligação de cada sistema remoto à rede pretendida.

Apresenta contudo como enorme inconveniente só providenciar segurança com a sua aplicação juntamente com o IPsec como se nota na sua definição [1], bem como tal como o PPTP transnportar todas as mensagens de controlo e pesquisa na rede local.

De facto o L2TP não oferece autenticidade nem privacidade, pelo que a sua própria definição sugere (ou recomenda) a integração com o IPsec para assegurar estes aspectos. A disponibilidade é mais uma vez ignorada, sendo até possível constatar que pode ser posta em causa pela aplicação do L2TP. Na descrição da constituição do pacote L2TP efectuada acima nota-se que o campo M do AVP incorretamente configurado implica o fecho imediato da sessão ou do túnel. Esse fecho provocará a necessidade de estabelecer nova sessão ou túnel e

a retransmissão de toda a mensagem. Um ataque continuado que altere esse valor evita a comunicação numa forma que fere a disponibilidade.

Não é de admirar que os inconvenientes em termos de segurança do PPTP subsistem neste protocolo, já que, como referido, o L2TP pretendeu combinar as funcionalidades do PPTP com o L2F. Desta forma, os extremos da comunicação podem opcionalmente utilizar um protocolo de autenticação quando do estabelecimento do canal que não é contudo utilizado em comunicações subsequentes. Para esse fim é recomendada a utilização conjunta com o IPsec para obter a autenticação e encriptação.

2.3 IPsec

O *Internet Protocol Security* (IPsec) foi definido com o objectivo de possibilitar um mecanismo de maior capacidade na segurança de comunicações ao operar num nível superior do modelo de referência OSI.

2.3.1 Descrição

O IPsec [11] é um protocolo desenvolvido para proporcionar segurança de comunicação no nível de rede do modelo de referência OSI.

A sua estrutura base é algo complexa pois permite uma variedade significativa de modos de operação, assentando em dois modos de utilização distintos entre si e não intermutáveis e dois protocolos de segurança que podem operar isolada ou concomitantemente.

Como parte integrante deste protocolo temos as associações de segurança (*Security Association* **SA**), unidirecionais – o que implica que para haver uma comunicação segura bidirecional é obrigatória a criação de duas **SA**'s – e duas bases de dados, a *Security Policies Database* (**SPD**) e a *Security Association Database* (**SAD**). A primeira define a política de segurança que deve ser aplicada a uma determinada mensagem, procurando depois na segunda, caso uma política tenha sido encontrada, os parâmetros de segurança a aplicar – isto é, as definições que estão implícitas na SA associada.

Os modos de funcionamento são designados por modo **transporte** e modo **túnel**. O primeiro é aplicado no encapsulamento IP que ocorre no nível 3 do modelo de referência OSI enquanto o segundo é aplicado após o encapsulamento IP ter ocorrido. Como diferenciador importante e visível entre os dois modos, basta atentar que o primeiro mantém o cabeçalho IP tal como é normalmente constituído enquanto o segundo encapsula o cabeçalho IP original num outro pacote IP com um novo cabeçalho (normalmente designado por cabeçalho IPsec).

Os parâmetros de segurança que fornece divergem consoante o protocolo de segurança aplicado. O *Authentication Header* (**AH**) [12] tem como objetivo assegurar a autenticidade do pacote IP em toda a sua extensão, excluindo apenas os campos do cabeçalho IP que se alteram em trânsito, através da aplicação de um algoritmo de *hash* cujo resultado é incluído no pacote

(para aplicação da função de *hash* os campos alteráveis do cabeçalho IP são considerados com todos os seus bits com o valor zero). O *Encapsulating Security Payload* (**ESP**) [13] preocupa-se mais com a confidencialidade, encriptando a informação em trânsito através da aplicação de uma cifra.

Quando aplicados conjuntamente e especialmente em modo túnel, o IPsec providencia a melhor segurança possível.

O IPsec pode ser configurado conjuntamente com um protocolo de troca automática de chaves que não é aqui abordado.

2.3.2 Formato

O formato do IPsec depende do modo de funcionamento e do protocolo de segurança pretendido.

Em modo transporte o pacote IP terá um dos formatos mostrados na Figura 8 e na Figura 9 consoante o protocolo de segurança adotado.

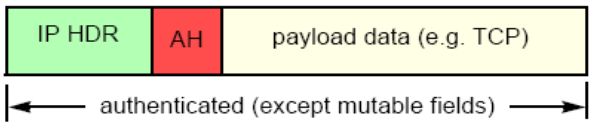

Figura 8 - AH em modo transporte
(Fonte: IP Virtual Private Networks [14])

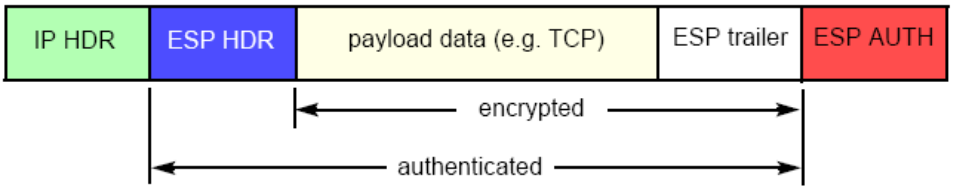

Figura 9 - ESP em modo transporte
(Fonte: IP Virtual Private Networks [14])

Os formatos do cabeçalho IP em modo túnel encontram-se representados na Figura 10 e na Figura 11.

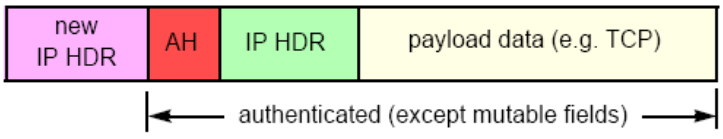

Figura 10 - AH em modo túnel
(Fonte: IP Virtual Private Networks [14])

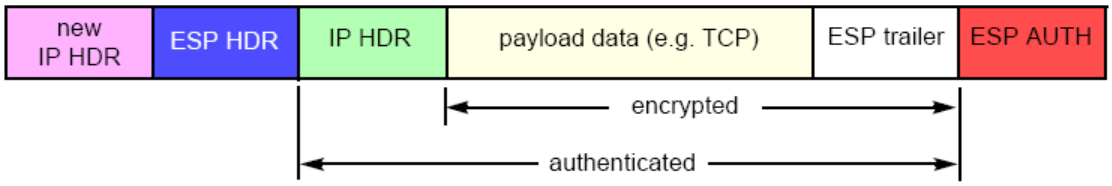

Figura 11 - ESP em modo túnel
(Fonte: IP Virtual Private Networks [14])

Caso o protocolo AH seja aplicado conjuntamente com o protocolo ESP, o formato resultante é similar aos apresentados se incluirmos após o cabeçalho AH (que é obrigatoriamente o primeiro a aplicar quando os dois estão em utilização conjunta) o cabeçalho ESP.

Como se nota nas figuras acima referidas importa verificar a constituição e formato do cabeçalho AH e do cabeçalho e rodapé ESP.

O cabeçalho AH tem uma dimensão que tem que ser múltipla de 32 bits e o seu formato encontra-se representado na Figura 12.

Next header	**Payload len**	**Reserved**
Security parameters index (SPI)		
Sequence number field		
Integrity Check Value (ICV) [variable]		

Figura 12 - Cabeçalho AH
(Fonte: RFC4302, disponível em http://www.rfc-editor.org/rfc/rfc4302.txt)

O campo **Next header** indica o protocolo que se segue ao AH.

O campo **Payload len**(gth) especifica o tamanho do cabeçalho AH em palavras de 32 bytes decrementado de dois (por exemplo, se a autenticação provoca um valor com 96 bits este campo conterá o valor quatro, sendo este resultado obtido de 3 palavras de 32 bytes devidos às três primeiras linhas da Figura 12 acrescido de 96/32 para o ICV menos 2).

O campo **Reserved** está reservado para evolução futura do protocolo, deve conter zeros e deve ser ignorado pelo sistema recetor. Note-se contudo que este campo de dezasseis bits é considerado na aplicação do algoritmo de *hash*.

O campo **SPI** contém um valor arbitrário que identifica a SA associada à mensagem. Numa SA *unicast* este valor pode ser utilizado *de per si* para especificar a SA ou pode ser utilizado conjuntamente com o tipo de protocolo (neste caso, AH). Em SA's *multicast* a definição do AH especifica o algoritmo que deve ser utilizado para associar pacotes IPsec à SA respetiva.

O campo **Sequence number** é um inteiro sem sinal de 32 bits que é incrementado por cada pacote enviado. Pode ser implementada uma extensão ao protocolo AH denominada **ESN** (*Extended Sequence Number*) que deverá ser negociada pelo protocolo de gestão da SA e que gera um campo de 64 bits. Contudo o cabeçalho AH mantém-se igual sendo os 32 bits mais significativos mantidos pelos sistemas em comunicação, mas nunca transmitidos – com o objetivo de minimizar o *overhead*.

Finalmente, o **ICV** tem de ter uma dimensão múltipla de 32 bits pelo que pode conter *padding* (preenchimento) e contém o resultado da operação de *hash* obtido para a mensagem.

Ao contrário do AH, o ESP envolve a informação em transmissão. Apresentamos na Figura 13 o seu formato.

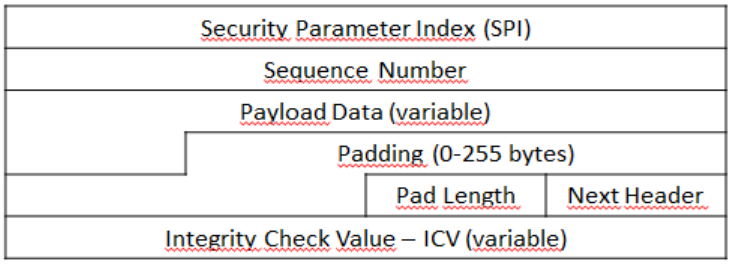

Figura 13 – Formato do ESP
(Fonte: RFC4303, disponível em http://www.rfc-editor.org/rfc/rfc4303.txt)

Alguns dos campos do ESP possuem um formato e função igual à do AH pelo que os omitiremos. Chama-se a atenção que, tal como o AH, o pacote ESP tem que ter uma dimensão múltipla de 32 bits. Também e tal como o AH o ESP suporta o ESN.

Mais interessante é o conjunto de campos **Payload data**, **Padding**, **PAD length** e **Next header** (estes últimos três campos constituem o ESP Trailer que é visível na Figura 9 e na Figura 11).

O campo **Payload Data** possui uma subestrutura que depende do algoritmo de encriptação e do modo de funcionamento escolhido. Esta subestrutura encontra-se representada na Figura 14. Chama-se a atenção que sendo a integridade opcional no ESP esse campo poderá não existir.

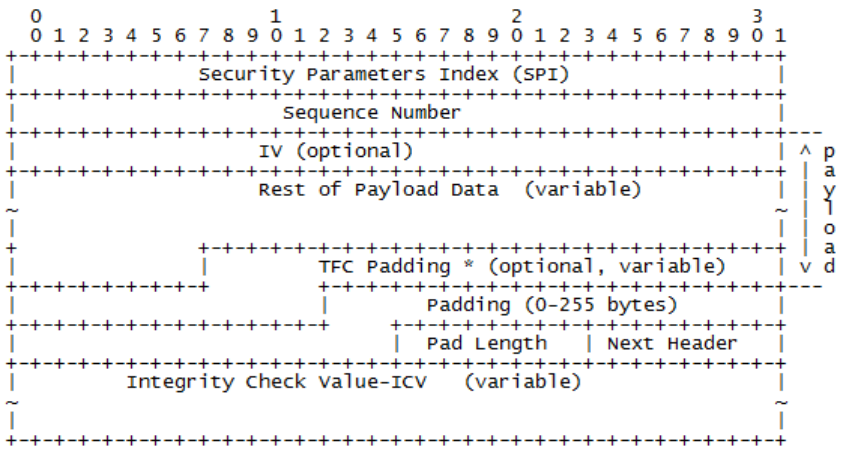

Figura 14 - Subestrutura do campo *Payload Data* do ESP
(Fonte: RFC4303, disponível em http://www.rfc-editor.org/rfc/rfc4303.txt)

Se o algoritmo de cifra exige um vetor de inicialização específico este é incluído nesta subestrutura.

Alguns algoritmos de cifra proporcionam integridade e encriptação numa só operação. Neste caso a integridade é validade apenas para os campos que estão encriptados, pelo que os campos **SPI** e **Sequence Number** são aqui repetidos.

O campo **Rest of Payload Data** inclui o pacote IP anterior à aplicação do protocolo ESP.

O campo **Next Header** é obrigatório e contém um código numérico que identifica o tipo de dados incluído no campo **Payload Data**, de acordo com a definição de número de protocolos definido pela *Internet Assigned Numbers Authority* (IANA).

O IPsec suporta e pode ser complementado com um protocolo automático de gestão de chaves *Internet Key Exchange* (IKE) aumentando desta forma a segurança proporcionada dado que, se corretamente configurado, implementa associações de segurança com cifras de autenticação e encriptação diferentes mas comuns aos dois extremos do canal [14].

2.3.3 Overhead

Vários estudos existem sobre o *overhead* implícito na utilização do IPsec. Este valor varia consoante o modo de utilização e o ou os algoritmos de segurança utilizados.

Um mínimo de 16 bytes é obrigatório caso se utilize o AH em modo transporte se assumirmos que o campo ICV obtido pela aplicação do algoritmo utilizado induz um resultado que não seja superior a 4 bytes. Mas com os protocolos de autenticação mais habituais atualmente essa assunção falha (basta pensarmos no *Advanced Encryption Standard* (AES)[18]. Em [15] é obtida uma expressão que permite calcular o valor do pacote após aplicação do IPsec (e por inerência a dimensão do *overhead*) em função do tamanho de pacote por omissão.

[18] http://www.rfc-editor.org/pdfrfc/rfc3268.txt.pdf

2.3.4 Vantagens e desvantagens

Em "*A Cryptographic Evaluation of IPsec*" [16], a análise do IPsec provoca dois tipos de conclusões díspares. Por um lado, é bem melhor do que qualquer outro protocolo de segurança que opere nos níveis mais baixos do modelo de referência OSI. Por outro lado estes autores não acreditam que alguma vez venha a resultar num sistema seguro.

Há falhas que podem ser consideradas graves. Assumindo a máxima que "*o pior inimigo da segurança é a complexidade*", a configuração do IPsec – os dois modos de funcionamento, os dois protocolos de segurança que oferece – pode implicar implementações frágeis. Além disso, a necessidade de o IPsec gerir a sua própria fragmentação para garantir a compatibilidade com os protocolos existentes – como o TCP – levam também a uma maior complexidade do protocolo, bem como a granularidade fina resultante da conjunção de **SA**'s, **SPD**'s e **SAD**'s.

O IPsec é tão-somente um protocolo de segurança do nível de rede, não do nível aplicacional. Tentar implementar autenticação de utilizador, por exemplo, que é manifestamente uma função de níveis superiores, ou outras funções que são tarefa de outros níveis só provoca confusão e, potencialmente, fragilidades.

É curioso notar que a definição mais atual do IPsec aponta para algumas das sugestões presentes em [16]. De facto, esta definição já indica que o protocolo AH pode existir nas implementações do IPsec enquanto o ESP deve existir. Note-se que a autenticação providenciada pelo AH abrange o cabeçalho do pacote IP enquanto a providenciada pelo ESP não. Mas, qual a necessidade de autenticar o cabeçalho? A autenticação da mensagem já prova que a informação é proveniente de alguém que sabe a chave de autenticação correta, pelo que a inclusão do cabeçalho IP nada lhe acrescenta. Nessa análise do protocolo [16] sugerem pois que o AH seja eliminado, alterando-se a definição do ESP para exatamente o oposto do modo atual – obrigatória a autenticação e opcional a encriptação.

Uma vantagem que se pode apontar ao modo transporte é implicar um menor *overhead*. Mas sugerem que no modo túnel seja aplicado um algoritmo de compressão que poderá implicar, apenas na opção de autenticação, um *overhead* equivalente ao atualmente implícito no modo transporte [16].

Em conclusão, o IPsec promete (ou prometia à data da sua aparição) um mecanismo de segurança eficaz e completo, excluindo o fator da disponibilidade. O passar dos anos e as lacunas apontadas em [16] aliadas à complexidade da sua implementação levam a que seja possível questionar atualmente qual a importância que se lhe deve ser atribuída. É provavelmente o protocolo de segurança mais eficaz, mas não oferece uma segurança (no sentido restrito do termo) completa.

2.4 TLS

O *Transport Layer Security* (TLS) foi definido com o objectivo de normalizar um protocolo que se tornou num *standard de facto* nas comunicações, o SSL (*Secure Sockets Layer*), bem como passarem a ser possíveis critérios de segurança de nível muito superior aos anteriormente descritos.

2.4.1 Descrição

O TLS é a versão padronizada [17] do protocolo SSL desenvolvido pela Netscape para proteger as comunicações HTTP. A versão 3.0 do SSL com algumas alterações – que implicaram a incompatibilidade – foi adotada pelo IETF com a designação *Transport Layer Security*). O seu objetivo é providenciar a transferência segura de informação sobre um canal inseguro, tipicamente a Internet. Permite que aplicações cliente-servidor comuniquem de forma a impedir espionagem, adulteração ou falsificação de mensagens.

É constituído por dois protocolos que cooperam entre si e que se descrevem em seguida:

- *TLS Handshake Protocol* (TLSHP); e
- *TLS Record Protocol* (TLSRP).

O TLSHP tem por função criar e gerir sessões seguras entre os sistemas enquanto o TLSRP gere o transporte seguro sobre um protocolo de transporte inseguro, usando para o efeito os algoritmos negociados pelo TLSHP.

O TLSRP é um protocolo desenvolvido em camadas, isto é, captura as mensagens que vão ser transmitidas e fragmenta-as em blocos, comprimindo-os opcionalmente, aplica a autenticação em cada bloco, encripta-os e transmite os blocos resultantes. Do lado recetor, cada bloco é desencriptado, verificado, expandido (se a compressão estiver ativada), agrupado com os demais blocos pela ordem e posição correta e finalmente fornecido ao nível a que se destina.

O objetivo do TLSRP é então providenciar uma comunicação segura através de duas propriedades básicas:

1. A comunicação é privada
   Aplicando criptografia simétrica para encriptação dos dados com uma chave partilhada gerada para cada conexão por um outro protocolo como o TLSHP.
2. A comunicação é de confiança
   A mensagem enviada inclui o resultado de uma cifra irreversível para garantir a sua integridade, gerada por uma função de *hash* segura como o MD5 ou o *Secure Hash Algorithm* (SHA).

Com estas propriedades o TLSRP encapsula a informação proveniente de protocolos de nível superior, um dos quais é o TLSHP que tem por objetivo criar um canal de autenticação entre o

sistema cliente e o sistema servidor e negociar um algoritmo de encriptação bem como as chaves criptográficas antes que qualquer comunicação de dados ocorra.

O TLSHP baseia-se em três propriedades:

1. Autenticação dos sistemas
   A identidade dos sistemas em comunicação pode ser autenticada através de criptografia assimétrica (chave pública).
2. Negociação segura
   A negociação de uma chave secreta é imune a espionagem e para qualquer conexão autenticada o segredo é imune a ataques
3. Confiança da negociação
   Nenhum atacante pode modificar a comunicação de negociação sem ser detetado pelos sistemas comunicantes.

A especificação atual do TLS (1.1) prioriza os objetivos que este protocolo pretende atingir, respetivamente:

1. Segurança criptográfica
   O TLS deve ser utilizado para estabelecer uma conexão segura entre dois sistemas.
2. Interoperabilidade
   Qualquer aplicação pode invocar o TLS que por sua vez pode estabelecer e negociar os parâmetros criptográficos de forma independente da aplicação.
3. Extensibilidade
   Foi objetivo da definição do TLS garantir um ambiente de desenvolvimento no qual novas chaves públicas e métodos de encriptação em bloco podem ser incorporados quando necessário.
4. Eficiência
   As operações criptográficas tendem a ser pesadas de um ponto de vista de processamento, principalmente se de chaves públicas. Para minimizar este aspeto o TLS incorpora um esquema de *cache* para reduzir o número de conexões necessárias.

O TLS especifica três modos de autenticação distintos [1]:

1. Sem autenticação (interações anónimas)
   Neste modo o canal seguro é criado usando uma chave partilhada negociada com o algoritmo de Diffie-Hellman[19]. Apresenta o inconveniente de a troca das chaves públicas não ser autenticada o que permite ataques por interposição.
2. Autenticação do servidor
   Neste modo o canal seguro é criado utilizando uma chave negociada entre os extremos usando um protocolo em que o servidor usa um par de chaves

[19] http://www.ietf.org/rfc/rfc2631.txt

assimétricas e um certificado X.509[20] da chave pública para se autenticar. A negociação da chave pode ser também efetuada segundo o algoritmo de Diffie-Hellman ou escolhida pelo sistema cliente, sendo neste caso enviada de forma secreta para o sistema servidor cifrada com a chave pública do recetor.

3. Autenticação mútua do cliente e do servidor
   Neste modo os extremos da comunicação criam um canal seguro usando uma chave negociada entre ambos usando um protocolo em que quer o sistema cliente quer o sistema servidor usam um par de chaves assimétricas e um certificado X.509 da chave pública para se autenticarem. De novo a chave pode ser negociada com o algoritmo de Diffie-Hellman ou simplesmente escolhida pelo sistema cliente e enviada de forma secreta para o sistema servidor cifrada com a chave pública do recetor.

As diferenças mais significativas da especificação 1.1 face à especificação 1.0 prendem-se com situações que se consideraram poder possibilitar anteriormente alguns tipos de ataques, nomeadamente no valor inicial do vetor de inicialização, a gestão da comunicação de erros de *padding* e alargamento da informação relativa a novos ataques. Para além disso foram definidos registos para parâmetros do protocolo e corrigido um problema no fecho prematuro da sessão que inviabilizava que fosse restaurada.

2.4.2 Formato

Há várias formas do protocolo TLS de acordo com a operação específica que ocorre, não sendo por isso relevante descriminar um formato. A própria definição do TLS não o faz, optando antes por definir as estruturas de comunicação – numa linguagem tipo C – utilizadas nos vários estágios do protocolo. Para além disso, os vários modos de autenticação implicam sobrecargas diferenciadas no canal de comunicação.

2.4.3 Overhead

Dado que este documento se debruça apenas sobre o *overhead* implícito na comunicação de dados, como anteriormente referido, analisemos então o bloco associado à comunicação de mensagens quando o canal seguro já está criado bem como a sessão associada. Da análise da RFC4346 vemos que a estrutura associada é a representada na Figura 15.

Assumindo que não é efetuada compressão o *Message Authentication Code* (MAC) é calculado com o algoritmo SHA-1 o que implica uma dimensão de vinte bytes. Se assumirmos também que a cifra negociada é AES-128 que tem um bloco de dezasseis bytes, então o *padding* poderá ter uma dimensão de quinze bytes já que a especificação obriga a que haja um alinhamento da dimensão total a um múltiplo de oito menos um byte (para o *padding_length*) o que perfaz um *overhead* de

[20] http://www.ietf.org/rfc/rfc3280.txt

20 + 15 + 1 = 36 (1)

Neste cálculo não foi considerado o campo **IV** que tem uma dimensão variável de acordo com a dimensão de bloco da cifra considerada.

```
block-ciphered struct {
    opaque IV[CipherSpec.block_length];
    opaque content[TLSCompressed.length];
    opaque MAC[CipherSpec.hash_size];
    uint8 padding[GenericBlockCipher.padding_length];
    uint8 padding_length;
} GenericBlockCipher;
```

Figura 15 - Bloco de comunicação de dados do TLS 1.1
(Fonte: RFC4346, disponível em http://www.rfc-editor.org/rfc/rfc4346.txt)

2.4.4 Vantagens e desvantagens

Uma vantagem do TLS é ser independente do protocolo de aplicação utilizado (deixando para esse protocolo a responsabilidade de tirar partido do TLS).

Não apresenta na sua especificação atual inconvenientes relevantes, com exceção do elevado *overhead* que representa.

A fragmentação implícita no TLS poderia chocar com o TCP, mas uma vez que o protocolo responsável pela transmissão de informação opera sobre o nível de transporte, a mensagem TLSRP é encapsulada num datagrama TCP pelo que esse potencial conflito não existe. Mas que sucede se um ataque de negação de disponibilidade injeta um pacote com um número de sequência TCP correto mas um número de sequência TLSRP incorreto (sendo este campo cifrado, o mais provável é que seja realmente incorreto)?

O TCP deverá aceitar o pacote já que a sua estrutura está correta. Mas o TLSRP deverá rejeitá-lo por conter um *sequence number* incorreto. Se assumirmos que o pacote correto é rececionado depois, o TCP deverá rejeitá-lo por conter um número de sequência já rececionado, não o passando nunca para o TLSRP.

No entanto, o *sequence number* do TLSRP nunca é transmitido, apenas faz parte do cálculo do MAC. Se um datagrama TLSRP com um MAC incorreto for rececionado é emitido através do TLSHP uma mensagem de erro de MAC incorreto (*bad_record_mac*). Esta mensagem é fatal, ou seja, força o fecho da conexão e obriga à negociação de nova conexão para retransmissão da informação o que pode conduzir, no limite, a um ataque de negação de disponibilidade.

3. Interoperabilidade entre protocolos

Dado que todos os protocolos possuem vantagens e inconvenientes pode-se colocar a questão de utilizar simultaneamente mais do que um só com o objetivo de mitigar a possibilidade de falha associada a um isoladamente. De seguida é efetuado uma análise teórica da utilização conjunta dos protocolos de segurança abordados.

Algumas das utilizações conjuntas não serão certamente utilizadas, mas tentou-se obter um resultado teórico o mais abrangente possível do ponto de vista do *overhead* implícito.

Atendendo que o PPTP e o L2TP operam no mesmo nível do modelo de referência OSI que é por sua vez diferente dos níveis em que operam o IPsec e o TLS, a interdependência de um ponto de vista funcional não apresenta qualquer inconveniente.

De um ponto de vista de *overhead* implícito o resultado é regra geral a soma dos *overheads* devidos a cada um dos protocolos isolados, quando a conjunção é efetuada por protocolos de diferentes níveis do modelo de referência OSI – caso do L2TP com IPsec e/ou com TLS.

Um caso diferente ocorre quando os protocolos de segurança utilizados operam no mesmo nível, o que só sucede com o PPTP e o L2TP. E aqui o *overhead* resultante difere consoante o cenário.

Admita-se que A está fora do seu local de trabalho habitual e estabelece um túnel PPTP com a rede da empresa. As mensagens trocadas com a rede sofrem do *overhead* implícito do protocolo de segurança aplicado. Admita-se que é então estabelecido um túnel L2TP da rede da empresa a um outro sistema. As mensagens que circulam da rede para o outro sistema serão encapsuladas no protocolo associado, mas apenas a partir da rede da empresa, pelo que não há operação conjunta dos protocolos.

Uma situação diferente ocorre se o túnel for estabelecido dentro da própria rede. Se estabelecermos um túnel PPTP numa rede local e iniciarmos depois um outro túnel L2TP a um sistema remoto a partir da mesma máquina – isto é, não de rede a rede – então é obtido um *overhead* que é igual à soma dos *overheads* isolados de cada um deles já que a mensagem PPTP será encapsulada num pacote L2TP.

Ora estes dois protocolos caracterizam-se por implicarem um *overhead* baixo, como descrito acima. Mas se o túnel para a rede remota for IPsec ou TLS? Ou se para prever a confidencialidade do L2TP for implementado um túnel IPsec, como sugerido na definição do L2TP? Não é possível, dada a multiplicidade de opções disponíveis, caracterizar de modo fixo o *overhead* resultante de qualquer destes últimos protocolos. Mas no melhor cenário de utilização conjunta de todos eles em simultâneo – e considerando apenas as mensagens de dados e ignorando as de controlo – teremos:

28 (PPTP) + 16 (L2TP) + 16 (IPsec em modo transporte e apenas com AH) + 36 (TLS) = 96 (2)

O valor obtido – que, recorde-se, pressupõe critérios de segurança que não são de forma alguma a melhor implementação possível – leva-nos a admitir que apesar da elevada capacidade dos canais de comunicação atuais o atraso provocado pela aplicação conjunta de todos os protocolos levaria a mal-estar por parte dos utilizadores. Se é uma máxima vulgarizada entre os especialistas de segurança que "*A adoção de protocolos de segurança implica um atraso na comunicação*" (tradução livre) nos não especialistas resume-se a uma frase crua: "*Os computadores estão lentos!*".

4. Análise comparativa

Do exposto nas secções anteriores ressaltam algumas das características que podem e devem ser comparadas se se pretender optar por um critério de comunicação segura entre dois sistemas.

Conclui-se ser manifesto que o PPTP e o L2TP são protocolos cuja definição de segurança é escassa (Secções 2.1 e 2.2). Não se espera que um protocolo de encapsulamento de comunicações para segurança se preocupasse com a disponibilidade (terceiro vetor da definição mais comummente aceite) já que esse vetor é mais corretamente associado a configurações a efetuar no sistema e/ou no dispositivo de segurança (tipo *firewall*[21]) que protege uma rede, pelo que a análise comparativa foi realizada tomando em consideração os outros dois vetores, confidencialidade e integridade.

O PPTP só assegura a confidencialidade. Implementações específicas – como a da Microsoft – incluem um protocolo de autenticidade de intervenientes. Mas a integridade é omissa quer na definição quer na implementação mais vulgar, pelo que a segurança que oferece é fraca. Contudo, é uma alternativa a considerar por um aspeto que em nada tem a ver com a segurança (e que é inclusive equacionada se é realmente uma vantagem): a capacidade de uma forma simples e sem necessidade de equipamentos ou aplicações adicionais possibilitar que um sistema remoto se comporte como se estivesse fisicamente ligado à rede de destino.

O L2TP não oferece autenticidade nem privacidade, como anteriormente referido. Se assumirmos que para obter esses aspetos teremos a necessidade de implementar IPsec juntamente com o L2TP, então para quê implementar o L2TP? As únicas vantagens que se nos afiguram para este protocolo são a não limitação a redes IP (o que, de novo, não é uma vantagem competitiva no tocante à segurança) e o estabelecimento de múltiplas sessões dentro do túnel criado, isto é, se considerarmos um túnel com L2TP entre duas redes qualquer sistema de qualquer uma das redes pode estabelecer uma sessão dentro desse túnel – sendo assim mais abrangente do que o PPTP em que é estabelecido um túnel exclusivamente entre dois sistemas.

O IPsec está muito próximo de ser um protocolo excelente. Mas, como referido, peca por vários defeitos ou excessos de flexibilidade que o tornam pouco suscetível de ser utilizado fora dos ambientes empresariais, onde normalmente existe uma equipa de informática dedicada. Apresenta ainda hoje – e recordemo-nos que foi lançado há mais de dez anos – lacunas e falhas que, na versão atual (3), já esperávamos que estivessem sanados. Tem no entanto algumas vantagens competitivas, como a multiplicidade de configurações possíveis e a facilidade de integrar autenticidade, integridade e confidencialidade. Mas parece-nos lícito duvidar que alguma vez constitua um protocolo perfeito. Com o tempo decorrido desde o seu aparecimento, sentimo-nos tentados a concordar com [16] quando afirma que o problema do IPsec não é o protocolo em si mas o método escolhido para o seu desenvolvimento que

[21] Um *firewall* é um sistema dedicado para controlo de acessos; podendo ser aplicacional ou físico, consideramos aqui o segundo caso

motivou que interesses díspares fossem contemplados e provocando por arrasto a elevada complexidade que o caracteriza.

O TLS é o único protocolo que opera nos níveis mais altos do modelo de referência OSI. Como tal, era lícita uma maior expectativa no seu funcionamento e operacionalidade. Parece-nos o protocolo mais perfeito dos estudados, mas o elevado *overhead* poderá causar alguns problemas de implementação principalmente em canais de comunicação mais fracos. A necessidade de obter certificados X.509 para o melhor funcionamento do sistema – ainda que só no lado do servidor – pode obstar nos tempos mais próximos à vulgarização da sua utilização por um efeito de custo-proveito.

Um aspeto que parece desconcertante nestes protocolos é que todos podem provocar uma quebra de segurança de disponibilidade. Com efeito a simples troca de um bit numa comunicação pode implicar a quebra do canal protegido o que, no limite, leva à indisponibilidade do sistema remoto. No mínimo, implicará a necessidade de renegociar o canal (túnel, sessão, chaves, etc.) com custos que se podem tornar significativos na eficácia da comunicação já que toda ela terá que ser repetida – e não retomada.

Finalmente, a robustez dos protocolos. Por robustez entendemos a impraticabilidade de obter a chave de cifra tendo a cifra. Um estudo de 2001 [18] aborda a ordem em que deve ser efetuada a aplicação do algoritmo de autenticação e de encriptação. Conclui-se que a forma mais segura para garantir a integridade e confidencialidade é aplicar a encriptação à mensagem e autenticar o resultado, ordem que é utilizada no IPsec e oposta à efetuada pelo TLS (curiosamente em [16] é analisado a sequência de autenticação e encriptação utilizada no IPsec concluindo-se que deveria ter sido definida a inversa, ou seja, a utilizada no TLS; refere que *"… Going by the 'Horton principle' … should authenticate what was meant, not what was said."*). A investigação realizada em 2001 contradiz esta afirmação, demonstrando como se consegue efetuar um ataque caso se aplica primeiro a autenticação e depois a encriptação.

5. Conclusões

Neste trabalho foram analisados os protocolos de segurança mais utilizados atualmente de um ponto de vista da constituição do seu cabeçalho e algumas especificidades do seu funcionamento. Foi também avaliado o *overhead* que implicam isoladamente mas apenas no tocante à transmissão de mensagens, nunca na ocupação do canal de comunicação e tráfego devido à negociação prévia de chaves de cifra, estabelecimento de canais, ou mensagens de controlo.

Pensamos que do estudo efetuado pode resultar uma melhor compreensão da problemática associada à implementação de um protocolo de segurança, sendo, esperamos, um auxiliar para um leigo poder tomar uma decisão fundamentada – ou no mínimo alertá-lo para consultar outros estudos.

Do estudo efetuado resulta que o PPTP está esgotado e deverá ser substituído pelo L2TP quando da aplicação de segurança de uma rede para outra, mantendo-se, mau grado as lacunas na implementação vulgarizada da Microsoft, quando da comunicação de um sistema isolado para um outro sistema. Contudo quer um quer o outro não garantem uma segurança de comunicação como as necessidades atuais exigem – o que é consubstanciado na própria definição do L2TP que recomenda a utilização do IPsec se o objetivo incluir também a encriptação.

O IPsec, sendo um protocolo que está "*…90% certo…*" como referido nas conclusões de [16] apresenta algumas lacunas, descritas na secção 2.3.4 que obstam a um funcionamento que proporcione a tão almejada segurança de comunicação – e convém não esquecer que este protocolo teve o seu início em 1995 [11]!

O TLS proporciona uma segurança mais vasta até porque opera nos níveis mais altos do modelo de referência OSI, permitindo assim alguns dos critérios que os mais atentos às questões de segurança pedem atualmente, como por exemplo a autenticação dos utilizadores. Mas também não oferece uma garantia de todos os critérios de segurança dada a ordem de aplicação de autenticação e depois encriptação aplicada, que já foi demonstrado que pode ser quebrado. Acresce que investigações mais recentes concluíram que as implementações de SSL/TLS em aplicações não orientadas para navegadores da Internet eram mal efetuadas o que levava à possibilidade de ataques [19].

Após terminar o estudo efetuado, algumas questões que foram levantadas e prendem-se com qual a real necessidade de protocolos de segurança dos níveis mais baixos do modelo de referência OSI. Porque não investir antes num protocolo que congregue parte das especificações do TLS juntamente com alguns dos aspetos do IPsec?

Realmente nenhum dos protocolos estudados preenche todos os requisitos de uma comunicação segura. Referindo-nos apenas aos dois últimos por serem os mais próximos dessa pretensão do nosso ponto de vista, a garantia da integridade e da confidencialidade está muito próxima do desejável mas não são imunes a ataques que a firam.

Após o estudo efetuado uma questão que se pode levantar é qual protocolo é que deverá ser utilizado. Depreende-se das secções anteriores que de um ponto de vista dos princípios comummente aceites para definição de segurança nenhum é perfeito – nem tão pouco o era esperado.

O PPTP e o L2TP poderão ser utilizados se é pretendido que o comportamento do sistema remoto (ou sistemas apenas no caso do L2TP) seja similar à ligação física à rede de destino. Dadas as falhas de segurança de que sofrem, poderão ser complementados com o IPsec ou o TLS, o que vai implicar um aumento do *overhead* e inerente custo de comunicação (largura de banda, tempo, processador).

O IPsec e o TLS apresentam os melhores comportamentos em termos de segurança, mau grado também apresentarem falhas – seja por princípios que nortearam a sua definição, seja pela complexidade de implementação. O TLS parece-nos ser a melhor opção, todavia para que todas as suas potencialidades em termos de segurança sejam utilizadas deverá cada extremo da comunicação possuir um certificado X.509. Dado que a utilização de um certificado válido implica custos, nem por vezes servidores possuem um certificado válido nem acreditamos que os clientes remotos os adquiram. Ademais, por ser um protocolo dos níveis superiores do modelo de referência OSI, possui propriedades típicas desses níveis, como a autenticação do utilizador, o que pode ser útil e vantajoso num determinado cenário.

Em jeito de conclusão, uma máxima que se costuma utilizar em segurança genérica, não estritamente informática: não faz sentido despender um custo para segurança superior ao valor do bem a proteger…

Referências

[1] Zúquete, A., "Segurança em Redes Informáticas" 2ª Ed., 2008

[2] Bishop, M., "Introduction to Computer Security", 2004

[3] Anderson, R., "Security Engineering" (second edition), 2008

[4] Dykstra, P., "Gigabit Ethernet Jumbo Frames", 1999

[5] RFC2637 disponível em http://www.rfc-editor.org/rfc/rfc2637.txt

[6] Schneier, B., "Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol", 1999

[7] Gregg, M, "Hack the Stack", 2006

[8] Schneier, B., Mudge, "Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)", 1999

[9] RFC2661disponível em http://www.rfc-editor.org/rfc/rfc2661.txt

[10] RFC1994 disponível em http://www.rfc-editor.org/rfc/rfc1994.txt

[11] RFC4301 disponível em http://www.rfc-editor.org/rfc/rfc4301.txt

[12] RFC4302 disponível em http://www.rfc-editor.org/rfc/rfc4302.txt

[13] RFC4303 disponível em http://www.rfc-editor.org/rfc/rfc4303.txt

[14] Hills, S., McGaughlin, D., Hanafi, N., "IP Virtual Private Networks", 2000

[15] Xenakis, C. Laoutaris, N., Merakos, L., Stavrakakis, I., "A Generic Characterization of the Overheads Imposed by IPsec and Associated Cryptographic Algorithms", 2005

[16] Ferguson, N. Schneier, B., "A Cryptographic Evaluation of IPsec", 2001

[17] RFC4346 disponível em http://www.rfc-editor.org/rfc/rfc4346.txt

[18] Krawczyk, H., "The Order of Encryption and Authentication for Protecting Communications (Or: How Secure is SSL?)" 2001

[19Georgiev, M., Iyengar, S., Jana, S., Anunhai, R., Boneh, D., Shmatikov, V., "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software" 2012