



JANET Technical Guides

Secure Virtual Private Networks

**Dr John S Graham
University of London
Computer Centre**

GD/JANET/TECH/004 (03/05)

JANET Technical Guides

JANET Technical Guides are one of a series of user guides available to JANET customers. Technical Guides contain detailed technical information and are intended for technical support staff at JANET sites, network specialists or those with a particular interest in the specialist area.

If you have any queries or comments about the Technical Guides or would like to obtain copies, please contact:

JANET Customer Service

UKERNA

Atlas Centre, Chilton, Didcot

Oxfordshire, OX11 0QS

Tel: 01235 822212

Fax: 01235 822397

E-mail: service@janet.ac.uk

Further details of the documents in this series are available at:

<http://www.ja.net/documents/>

Contents

1. Introduction	5
2. An Overview of VPN Technology	9
2.1 Motivation	11
2.1.1 Remote Site VPN	11
2.1.2 Remote User VPN	11
2.2 Elements	11
2.2.1 Tunnelling	11
2.2.2 Confidentiality	12
2.2.3 Integrity	12
2.2.4 Authentication	13
3. An Introduction to Cryptographic Techniques	15
3.1 Symmetric Ciphers	17
3.1.1 Stream Ciphers	17
3.1.2 Block Ciphers	18
3.2 Asymmetric Ciphers	18
3.3 Key Exchanges	19
3.3.1 RSA Key Exchange	19
3.3.2 Diffie-Hellman Exchange	19
4. Public Key Infrastructure and Authentication	21
4.1 Digital Signatures	23
4.2 Message Integrity	24
4.3 Digital Certificates	24
4.3.1 Certification Authorities (CAs)	24
4.3.2 Registration Authorities (RAs)	24
4.3.3 Anatomy of a Certificate	25
4.3.4 Trust Relationships	25
4.3.5 Trust Models	26
4.3.6 Authentication by Certificates	27
4.4 Authentication Models	27
5. Tunnelling	29
5.1 Layer III Tunnelling	31
5.1.1 GRE Tunnelling	31
5.2 Layer II Tunnelling	33
5.2.1 Compulsory Tunnelling	33
5.2.2 Voluntary Tunnelling	34
5.2.3 Layer II Tunnelling Protocol (L2TP)	34
5.2.4 Point-to-Point Tunnelling Protocol (PPTP)	34
6. IP Security: overview and architecture	37
6.1 IPSec Security Associations (SAs)	39
6.1.1 Security Parameter Index (SPI)	40
6.1.2 Sequence Numbers	40
6.1.3 Management of SAs	40
6.2 IPSec Modes	40
6.2.1 Tunnel Mode	41
6.2.2 Transport Mode	41
6.3 IPSec Protocols	42
6.3.1 Authentication Header	42
6.3.2 Encapsulating Security Payload (ESP)	42
6.3.3 Internet Key Exchange (IKE)	43
6.3.4 IPSec Domain of Interpretation (DOI)	44

	/continued	
7.	Implementation and Worked Examples	45
7.1	Configuring IPSec on Cisco® Routers	47
7.1.1	Configure Crypto Lists	47
7.1.2	Configure Transform Sets	47
7.1.3	Apply Crypto Maps	48
7.1.4	Configure Key Exchange Policies	48
7.2	Configuring IPSec on Windows® 2000	48
7.2.1	Filter Lists	49
7.2.2	Filter Actions	50
7.2.3	Authentication Methods	50
7.2.4	Tunnel Setting	51
7.3	Configuring an IPSec-Protected GRE Tunnel	51
7.4	Configuring Tunnel Mode IPSec	53
8.	Glossary	57

Introduction

1

1. Introduction

Universities or colleges consisting of multiple campuses, each with a Local Area Network (LAN), traditionally connect geographically diverse 'islands' by means of private leased lines. If the connected site is small and consumes little bandwidth, the costs of such Wide Area Network (WAN) links do not necessarily represent value for money. Many organisations also wish to offer their staff the facility to connect to their central network remotely, either from their houses or when travelling on business. Operating a corporate dial-up service is unattractive due to the capital equipment costs and the burden of supporting the service on home computers of unknown provenance.

For both requirements there is now an alternative to the expensive dedicated services of a true private network: a collection of technologies that can be used to construct a Virtual Private Network (VPN). A VPN carries an organisation's private network traffic securely over networks such as JANET and the Internet. Instead of using a dedicated, real-world connection such as a leased line, a VPN uses 'virtual' connections routed through the Internet from the organisation's private network to the remote site or employee. This guide discusses techniques for securely extending the reach of an organisation's network beyond the boundaries of a single physical campus by utilising existing JANET connectivity. Two examples illustrate the need for this wide area connectivity.

The technologies required to support a secure VPN exist as approved standards, and as manufacturers are now implementing these standards in commonly available products, the prospects for implementing stable VPN solutions without purchasing expensive items of equipment or additional software have never been better. However designing, implementing and running a VPN still requires an understanding of those fundamental technologies. Just as a network manager needs to understand addressing, routing and Transmission Control Protocol/Internet Protocol (TCP/IP), a VPN manager needs to understand what makes the network tick.

A secure VPN involves:

- encryption, to ensure that private traffic sent over shared networks cannot be read by other users of those networks;
- authentication, so that the systems at the two ends of the VPN can be confident of each other's identity;
- tunnelling, so that packets that would normally be contained within a single physical LAN can be carried over a WAN to reach remote segments of that LAN.

This guide brings together these diverse subject areas required to implement secure VPNs into a coherent account of the subject. The general principles of each topic are described along with the implementation of each used by the Secure Internet Protocols (IPSec), which are the most widely used international standard. Finally two practical examples illustrate the need for wide area connectivity and how VPNs can be created to provide it.

An Overview of VPN Technology 2

Motivation

Elements

2. An Overview of VPN Technology

A number of diverse computing topics contribute to the subject of VPNs, and this can make the subject appear daunting to a newcomer. This chapter seeks to provide a roadmap for a reader interested in implementing a straightforward VPN with minimal background reading. A discussion of the practicalities of implementing VPNs using Windows® 2000 and Cisco® routers is provided in *Chapter 7* along with some examples.

2.1 Motivation

There are two general networking scenarios for which a VPN solution is appropriate. Both concern the extension of a LAN over existing links to the Internet.

2.1.1 Remote Site VPN

In this scenario, a college wishes to connect a small remote office to the main campus without investing in a dedicated leased line to link the two. Both the main campus and the satellite are connected to the Internet, for example by means of a high-bandwidth leased line at the main campus and an Asymmetric Digital Subscriber Line (ADSL) connection at the satellite. Inter-campus network traffic can flow between the two sites with data encryption ensuring that confidentiality is maintained.

2.1.2 Remote User VPN

A college may wish to offer its staff the facility to log on to the local network from their houses without incurring the expense and support burden of operating a private dial-up service. These remote users would have access using normal domestic Internet Service Provider (ISP) connections to the local file servers and databases that would ordinarily not be reachable from the Internet for obvious security reasons. Certificate-based authentication ensures that only approved remote computers are able to negotiate a connection.

These two scenarios have been identified in order to illustrate the main uses of the technology. The processes involved in realising either of the above two scenarios are very similar, if not identical.

2.2 Elements

A secure VPN consists of two Internet-connected devices that, after having authenticated one another, exchange data over the Internet in a secure fashion. The four processes that comprise a secure VPN are tunnelling, confidentiality, integrity and authentication.

2.2.1 Tunnelling

This is the defining characteristic of a VPN as it allows packets to travel to destinations that would not ordinarily be reachable over the Internet. This allows existing Internet infrastructure to replace a dedicated intersite leased line or dial-up service. A VPN tunnel consists of two Internet-connected devices, one at either end. These tunnel endpoints both dispatch packets to the other endpoint and receive packets, sent by the peer, that are emerging from the tunnel.

In order to send a packet down the tunnel, it is first placed within another packet. This has the effect of creating a new outermost Internet Protocol (IP) header whose source and destination fields are filled with the addresses of the sending and receiving tunnel endpoints. When this packet is received at the far end of the tunnel, the additional headers concerned with delivery via the tunnel are stripped away and the original packet is regenerated. This mechanism can be used to dispatch two types of packet over the Internet that would, by their very nature, ordinarily be undeliverable.

2.2.1.1 Invalid Protocols

Some sites may employ network-level protocols, such as AppleTalk® or Novell's Internet Packet Exchange (IPX), on disparate LANs. The Internet, by definition, only routes IP packets, and so other Layer III protocols cannot be carried in their native form. Encapsulating an IPX packet within an IP 'envelope' would permit the two campuses to use the Internet, rather than private leased lines, to exchange Netware traffic.

2.2.1.2 Invalid Addresses

Many sites employ IP addresses from the designated private ranges on their local networks. A college with several campuses, each using private IP numbers, could use tunnelling to allow the campuses to exchange these packets via the Internet.

There are two broad classes of tunnelling methods that work by either encapsulating Layer II frames (usually Point-to-Point Protocol - PPP) or Layer III packets.

2.2.2 Confidentiality

A VPN causes traffic local to an organisation to be transmitted over infrastructure that carries general Internet traffic. It is essential to guard against the remote possibility that these packets could be intercepted and examined by some third party. Data confidentiality may be achieved by encrypting the payload of any packets that are destined for the remote end of a VPN tunnel. The encryption process is a compromise between the inevitable increase in transmission delays and the strength of the cryptographic cipher employed. There are two categories of encryption algorithm, and both are used to secure packets that travel over a VPN.

2.2.2.1 Symmetric

These algorithms rely upon the two security endpoints agreeing upon a secret phrase that is used for all subsequent encryptions and decryptions. Although they operate quickly, the great drawback of these encryption algorithms is that the shared key must be agreed in advance over the insecure medium. If this initial exchange were conducted in plaintext, any third party that managed to intercept it would be able to decode all the subsequent encrypted data.

2.2.2.2 Asymmetric

These algorithms do not require a secret phrase to be shared between the security peers. Each peer generates two keys, one of which (the *Public Key*) is published while the other (the *Private Key*) is kept secret. A message that has been encrypted with a peer's Public Key can only be decrypted by means of the partnering Private Key. Despite their great security, these algorithms are slow and therefore unsuitable for ongoing encryption of a stream of data such as IP packets.

A useful compromise between the speed of the symmetric algorithms and the security of the asymmetric type is readily achieved. A fast symmetric algorithm is used for securing the data stream with the shared secret (the *session key*) being encrypted using an asymmetric cipher. This means that transmission times for packets traversing the VPN are kept to a minimum without compromising security by exchanging the session key in plaintext. It is normal practice for the session key to be assigned a limited lifetime so that it must be periodically renewed. This further increases security, as an attacker would have insufficient time to discover the session key by means of some brute force attack before it expires and is replaced with a completely new key.

2.2.3 Integrity

It is vital that any data arriving at one of the endpoints of a VPN is guaranteed to have originated from the recognised security peer and not to have been modified en-route. Both of these assurances can be provided by use of digital signatures.

Passing a message through a mathematical function called a hash function produces a short, fixed-length digest. If even one bit of the original message is changed, then a different digest will be produced. Data integrity can be assured by attaching a digest to an outgoing message. When a message that has been transmitted via a VPN is received, the recipient applies the hash function to the data that was sent and compares the resulting digest to one that was generated by the sender and attached to the message. If the two digests differ, the recipient endpoint will know the message has been modified.

The problem with this scheme is that the sender's digest that accompanies the message could easily be replaced with one that had been calculated from the modified data. The recipient would then be unaware that the sender's message had been changed. The digest can be guaranteed to have originated from the sender by instead using a *keyed* hash function that uses the message and a key as the input. A symmetric hash function is one where the key is a secret phrase that the two security peers have previously agreed upon. A slower, but more secure, asymmetric hash function employs the sender's Private Key. An on-going stream of data will be authenticated using the fast symmetric variant and the key will be the same one used for the symmetric encryption. Because only the two security peers know the key, third parties will not be able change the digest and the recipient can be confident that the message has not been changed en-route from the sender.

2.2.4 Authentication

By introducing VPN technology into the network, servers that would otherwise be shielded from the dangers of exposure to the Internet can be rendered vulnerable. It is absolutely essential therefore that measures be taken to ensure that only approved remote stations are able to inject packets via a tunnel into the local network. Two different techniques can be used to identify approved stations.

2.2.4.1 Shared Secret

A password is configured on both of the stations that are acting as the tunnel endpoints. The authentication process requires each endpoint to check that the peer's copy of the secret matches its own.

2.2.4.2 Certificate Authentication

A digital certificate is installed on each of the tunnel endpoints. Providing the two stations have been configured so as to 'trust' the issuer of the peer's certificate then authentication will occur. These certificates can either be purchased from a commercial Certification Authority (CA), or the college can configure a server to generate their own local certificates.

The scenario, as described in *Section 2.1*, will probably be a factor when selecting an authentication method. If a number of remote users require access to the college LAN, then issuing certificates (which can later be revoked if a staff member leaves the college's employment) allows the network manager more control over who has access to the VPN facilities. For a single remote site, a shared secret is simpler as it does not require any supporting infrastructure and can be quite secure as the two tunnel endpoints can also be statically configured with the other's IP address as an additional identity check.

An Introduction to Cryptographic Techniques 3

Symmetric Ciphers

Asymmetric Ciphers

Key Exchanges

3. An Introduction to Cryptographic Techniques

Cryptography is one of the essential technologies used in building a secure VPN. Different applications of the same basic algorithms can provide both encryption that keeps data secret and authentication that ensures the two security peers in a VPN are who they claim to be. This chapter introduces some basic concepts in cryptography and demonstrates how they can be used in practice to provide data confidentiality. The next chapter continues this theme with a discussion of mutual authentication using cryptographic algorithms.

Data confidentiality may be provided by one of two categories of encryption algorithm, namely symmetric cryptography and asymmetric cryptography. Symmetric, or conventional, cryptography requires that the sender and receiver share a key, which is an item of secret information used to encrypt and decrypt data. The process by which two peers agree upon a key over an insecure medium can be problematic as, until the key is agreed, the peers have no way to communicate in secret. Asymmetric, or Public Key, cryptography solves the key exchange problem by using two keys, either of which may be used to encrypt a message. The encrypted data may then only be decrypted by means of the other key. Messages may be received securely by publishing one of the keys (for example, in the footer of an e-mail message) as a Public Key and keeping the second, the Private Key, secret. Anyone wishing to send a secure communication may then encrypt the message with the recipient's Public Key and, providing the Private Key has not been disclosed, only the intended recipient will be able to decrypt the encrypted text and recover the original message.

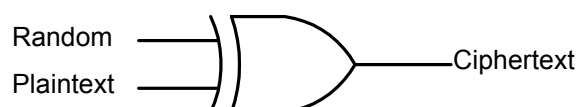
Throughout this discussion, the original unencrypted data will be referred to as *plaintext* and the encrypted form as *ciphertext*.

3.1 Symmetric Ciphers

Symmetric ciphers employ the same key to encrypt the plaintext and to decrypt the ciphertext. The sender and the recipient must therefore agree upon this key, which must be known to no one else, in advance. The cryptographic strength of a symmetric algorithm may be gauged by the size of the key it employs. Two examples are Data Encryption Standard (DES) and Blowfish. The DES algorithm uses a 64-bit key, of which 8 bits are reserved leaving 56 variable bits. It is now common practice to protect information with Triple DES (3DES) instead of DES. This means that the information is subjected to three successive encryptions. The use of multiple encryption cycles does not necessarily offer a concomitant increase in security, and may be viewed as a waste of computing power for many applications. Blowfish allows implementers to select a key length of between 32 and 448 bits; commercially available implementations often use 128-bit keys. Symmetric algorithms are popular because their speed enables them efficiently to encrypt large quantities of plaintext. There are two subcategories of symmetric cipher, stream and block ciphers.

3.1.1 Stream Ciphers

These algorithms operate upon one bit at a time. A stream of plaintext flows into the cipher and a stream of ciphertext emerges as the output. Messages encrypted with a stream cipher are always the same size as the original plaintext. The encryption takes place by means of an operation in which each bit of the plaintext is XORed (i.e. manipulated by the Boolean operator eXclusive OR - XOR) with a random bit to produce the ciphertext. The essence of a stream cipher concerns the methods by which the shared key is used to generate the stream of random



bits. Cracking attempts centre on analysing this random bit generator.

3.1.2 Block Ciphers

These ciphers encrypt data in blocks of bytes, rather than a single bit at a time. Block sizes vary according to the algorithm, 64 bits being the commonest. Because the plaintext is unlikely to be a multiple of the algorithm's block size, it is often necessary to pad the input. For example, if the block length is 64 bits and the last block contains only 40 bits, then 24 bits of padding must be added. The padding string can consist of all zeros, alternating zeros and ones, random bits, or some other sequence. Some encryption standards specify a particular padding scheme.

There are two methods for encrypting a sequence of blocks. Either the blocks are treated independently and the cipher is used on each block without reference to what has gone before, or the results of encrypting previous blocks affect the encryption of the current block. These two methods are known as the Electronic CodeBook (ECB) mode and Cipher Block Chaining (CBC) mode respectively.

3.1.2.1 Electronic CodeBook (ECB) Mode

In ECB, identical blocks of plaintext will clearly generate identical blocks of ciphertext. A cracker can therefore exploit repetition in the ciphertext to release the plaintext version.

3.1.2.2 Cipher Block Chaining (CBC) Mode

In CBC, a feedback mechanism is added so that the results of the encryption of previous blocks are fed back into the encryption of the current block. Each ciphertext block is made dependent not only on the plaintext block that generated it, but also on all previous plaintext blocks. This ensures that even if the plaintext contains many identical blocks, they each encrypt to a different ciphertext block. Prior to encryption, CBC takes the last block of ciphertext and XORs it with the current block of plaintext. Although CBC mode forces identical plaintext blocks to encrypt to different ciphertext blocks, messages that start with the same data will encrypt in the same way up until the first difference, since the initial plaintext blocks are identical. Encrypting random data, called the Initialisation Vector (IV), as the first block can prevent this. Decryption is the exact opposite in that the ciphertext is decrypted and then XORed with the previous block of ciphertext (or IV for the first block) to release the plaintext version.

Other modes are available, but they are not discussed here because all the block ciphers currently used in VPN technology operate exclusively in CBC mode

3.2 Asymmetric Ciphers.

As noted in the introduction to this chapter, the great advantage of asymmetric ciphers is that a shared secret key does not have to be exchanged over an insecure medium such as the public Internet. A pair of keys is generated and one of them is nominated as the Public Key and is published. Any parties wishing to communicate securely with the key's owner encrypt the message using the recipient's Public Key. The decryption can only be accomplished by knowing the second, Private, key, which the owner ensures is never released.

The most popular asymmetric block cipher is RSA (named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman). The keys of the RSA algorithm are composed of two parts. The first part is called the modulus. It is usually a 512-bit number and is the product of two 256-bit primes.

$$N = p \times q$$

The Public and Private Keys share the same modulus. The second part of an RSA key is called the exponent. This is a variable-length number, different for the two keys, with the exponent of the Public Key usually being the smaller of the two. RSA encryption works as follows. The plaintext (viewed as a binary number) is raised to the power of the Public exponent, and the remainder after dividing by the modulus is the ciphertext. To decrypt, the ciphertext is raised

to the power of the Private exponent, and the remainder after dividing by the modulus is the plaintext again. The RSA encryption and decryption functions are as follows:

$$C = T^k \bmod N \qquad T = C^l \bmod N$$

where C is the ciphertext, T is the plaintext, k and l are the public and private exponents and N is the modulus.

The security of asymmetric encryption stems from the difficulty of factoring this modulus back into its constituent primes. Without knowing the two primes used to generate the modulus, it is not possible to calculate the Private exponent from the Public one and therefore an RSA-encrypted message is secure from anyone save the holder of the relevant Private Key.

3.3 Key Exchanges

Although the asymmetric encryption algorithms are more secure than the symmetric types, they are also far slower and it is not feasible to use them to secure large quantities of data, as the consequent increase in transmission times would be excessive. Similarly, although chained mode symmetric algorithms can process large quantities of plaintext at speed, they do not offer the requisite level of security because the key is a shared secret that must be exchanged over the insecure medium prior to the transmission of the ciphertext. This paradox may be resolved as follows. A random secret, known as the *session key*, is generated and an asymmetric cipher secures this small piece of data for exchange over the Internet. A fast symmetric cipher then uses the session key, known only to the two security peers, to encrypt their exchanges of bulk data.

3.3.1 RSA Key Exchange

The RSA algorithm may be employed to provide a simplistic form of secure key exchange. If Alice wishes to secure some large quantity of data with a fast algorithm such as DES before transmitting the data to Bob, she first chooses some random 56-bit number as the DES key and encrypts it using Bob's RSA Public Key. Only Bob will be able to decrypt this exchange using his private RSA key. The drawback with this approach is that anybody, including the cracker Mallory, can encrypt anything using Bob's Public Key. Bob therefore has no proof that it is indeed Alice with whom he is communicating. The communications channel is only secure if Alice digitally signs the DES key and encrypts both the key and her signature with Bob's Public Key. The problem with this approach is that the signature can be too big to secure in a single RSA operation.

3.3.2 Diffie-Hellman Exchange

The Diffie-Hellman key exchange was the first Public Key cryptosystem and it underpins the entire framework by which IP packets may be securely transmitted over the Internet. The participants in the exchange must first agree upon a *group*, which defines the prime p and generator g that should be used. In the first part of the exchange, Alice and Bob each select a random private number (indicated by the lowercase initial of each party) and exponentiate to produce a corresponding public value (uppercase initial of the party):

$$\text{Alice: } A = g^a \bmod p \qquad \text{Bob: } B = g^b \bmod p$$

Alice and Bob exchange these two public values and they exponentiate again, using the other party's public value as the generator to produce the shared secret:

$$\text{Alice: } g^{ab} = (g^b)^a \bmod p \qquad \text{Bob: } g^{ba} = (g^a)^b \bmod p$$

Alice and Bob now have a shared secret:

$$g^{ab} = g^{ba} = k$$

The significant property of this exchange is that the public values A and B can be exchanged over an insecure public network without reducing the security of the exchange. An eavesdropper (conventionally known as Eve) could know g and p and intercept the exchange of public values and still not be able to discover the key because one of the private values must be known to generate the shared secret.

The Diffie-Hellman exchange is vulnerable to a man-in-the-middle attack in which Mallory impersonates Bob to Alice and Alice to Bob. Both Alice and Bob believe they are performing a key exchange with one another, but in reality are doing so with Mallory. When Alice sends secured data to Bob, Mallory can intercept the traffic and decrypt it before passing the packets on to Bob. Neither Alice nor Bob would notice anything out of the ordinary. This type of attack may be thwarted if Alice and Bob both digitally sign their public values.

Public Key Infrastructure and Authentication

4

Digital Signatures

Message Integrity

Digital Certificates

Authentication Models

4. Public Key Infrastructure and Authentication

The degree of security of a system is largely governed by the quality of the authentication procedures that are employed. Authentication may be defined as the process by which proof of identity or of integrity is established in response to some form of challenge. This chapter examines authentication methodologies based on asymmetric algorithms and their application to VPNs.

During the initial negotiation phases between two security endpoints, each peer should authenticate the other by some means. Failure to implement such peer authentication could allow an unknown system to masquerade as the remote end of a VPN in order to acquire confidential data. For example, if a university has provided a remote user VPN facility, only recognised computers should be allowed to establish a tunnel with the remote access server otherwise any Internet-connected computer could inject packets into the private campus LAN. Mutual authentication of the security endpoints is always the first stage of establishing a VPN connection. If this fails, all succeeding processes must stop so that the VPN connection is not established.

This form of device-authentication is completely separate from the more familiar user-authentication whereby a human submits their username and password in order to gain access to network resources. Device authentication is concerned with determining whether the basic network infrastructure of a VPN should be constructed. Only if these communication channels have been established can user authentication take place. Mistakenly deploying user-authentication instead of device-authentication can result in catastrophic security flaws.

Once identities have been established, data can be transmitted over the VPN. All packets arriving at the security endpoints must be submitted to an integrity authentication procedure to ensure that the security peer (and not some other device) sent them and that they were not modified during the course of their transmission over the public infrastructure. This is accomplished by attaching a digital signature to each packet, which is then checked by the recipient endpoint.

4.1 Digital Signatures

Public Key cryptography can be used to authenticate, as well as encrypt, a message by transmitting a digital signature along with the message data. The previous chapter described how messages may be encrypted with a recipient's Public Key. Only the holder of the corresponding Private Key is able to decrypt the message. If these actions are performed in the opposite sense, then a simple form of digital signature has been achieved, implying that the messages originate from the sender and not an impersonator. The sender encrypts messages with his or her Private Key and the recipient decrypts them with the sender's Public Key.

This simplistic attempt at authentication relies upon the transformation of encrypted gibberish into legible plaintext. While a human reader can tell the difference between ciphertext and plaintext, a machine cannot be expected to make this distinction and so something further is required to provide a worthwhile signature. The signatory passes the message through a one-way hash function that produces a unique fixed-length summary called the message *digest* as its output. This is encrypted with the sender's Private Key to produce a digital signature. A hash function produces a unique thumbprint that is characteristic of the message from which it is derived. It is safe to assume that a signed digest produced in this manner is a trustworthy digital signature. Furthermore, it is amenable to machine verification. The signature, in the form of an encrypted digest, is appended to the message it is authenticating. The recipient verifies the signature by passing the message through the same hash function and decrypting the signature. If the two digests are identical, then the signature is valid.

Digital signatures constructed by encrypting a message digest with the sender's Private Key are extremely difficult to forge, cannot later be repudiated and are non-transferable. This makes them ideal vehicles for guaranteeing the authenticity of the sender.

4.2 Message Integrity

A digital signature can guarantee the integrity of the message it accompanies as well as the identity of the sender because any alterations to a signed message will produce an entirely different hash value, rendering the signature invalid. However, digital signatures are slow to calculate because they are generated by means of asymmetric ciphers. For an ongoing stream of data (such as a series of IP packets), signing each protected packet would be as onerous as encrypting the payload with the recipient's Public Key rather than using a symmetric block cipher.

Fortunately, there are symmetric and asymmetric hash functions in the same way as there are symmetric and asymmetric ciphers. A digital signature uses a slow, highly secure asymmetric hash function. Once the peers have authenticated one another by these means, the sender of the data is trusted and no longer needs to be authenticated. A faster symmetric hashing function is sufficient to guarantee the integrity of any received packets. Symmetric hash functions where a single shared key is used to sign the input are known as Message Authentication Codes (MACs).

Two commonly implemented hash functions are the Secure Hash Algorithm (SHA) and Message Digest-5 (MD5). These functions simply produce a fixed-length digest of a variable length input. It is important that the keyed variant of these functions is employed otherwise an attacker could alter the payload of a packet and simply recompute the digest. A special type of keyed hash, known as the Hash-Keyed Message Authentication Code (HMAC) exists that can be utilised with any existing hash function and so the keyed versions of the aforementioned hash functions are known as HMAC-SHA and HMAC-MD5.

4.3 Digital Certificates

Digital signatures prove conclusively that a message was received in an unmolested state from the peer holding a given Public Key. However, there is a strong element of trust still involved because the message's recipient has no proof of the sender's identity. A certificate is a device, issued and digitally signed by a trusted third party, that binds the holder's identity to a Public Key. There is an agreed standard called X.509 (which is a member of the family of X.500 directory standards), that governs the properties a digital certificate must implement. Certificates are non-transferrable, non-forgable files that act as a digital identity badge or passport to help ensure that users or computers are who they say they are.

4.3.1 Certification Authorities (CAs)

A certificate may be obtained by applying to some issuing party, called a CA. This may either be a company that specialises in issuing digital certificates or a private organisation that requires users to submit a certificate it has previously issued as proof of digital identity. The CA should take reasonable steps to confirm the identity of applicants before issuing a certificate and should undertake various maintenance tasks throughout the lifetime of the certificate. These include the revocation of any certificate whose Private Key has been compromised or whose holder has left the issuing organisation. Most importantly, the CA should ensure that its own Private Keys are never revealed as this would allow an attacker to issue bogus certificates under the issuer's name thereby casting doubt on every certificate emanating from that CA.

4.3.2 Registration Authorities (RAs)

The purpose of the optional RA is to verify the information supplied by an applicant for a certificate. The CA may delegate this task to the RA or perform the checks itself. If the certificate confirms the identity of a person (as opposed to a computer), such verification might comprise checking the supplied e-mail address or for more demanding applications, requiring a personal visit to the RA's offices to prove identity by displaying a document such as a driving license or passport. Once the various checks have been conducted, the RA approves the request for a certificate by submitting it to the CA which then issues the certificate. The certificate will usually contain an indication of the identity checks performed.

4.3.3 Anatomy of a Certificate

Version 3 of the X.509 standard is most commonly implemented. It allows more detailed identifying information, and requires that a certificate bears the following items of information:

- the holder's identifiers (name, organisation, address, etc.);
- the holder's Public Key;
- the certificate's validity dates;
- the certificate's serial number;
- the name and signature of the issuer;
- the signature algorithm (usually SHA-1);
- the applicable X.509v3.

4.3.4 Trust Relationships

When two security endpoints exchange certificates, they are delegating the process of confirming the peer's identity to a CA. Acceptance of a certificate implies that the authenticator trusts the issuing authority, which in turn implies that the authenticator must have some existing knowledge of the CA.

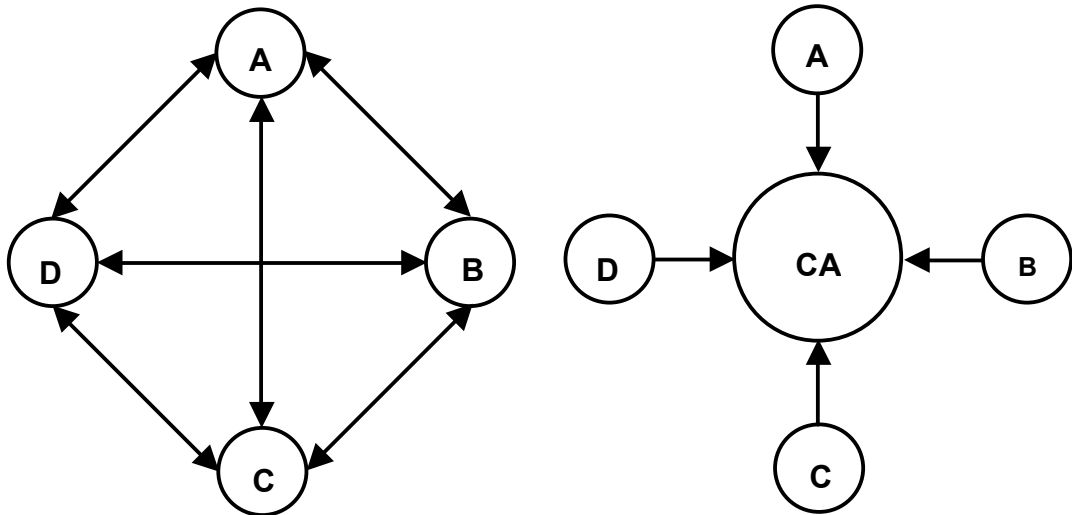
As well as confirming the identities of others, every CA publishes a special type of certificate that contains its own Public Key and is self-signed. This is tantamount to declaring one's own identity without any supporting proof. A node wishing to authenticate a peer must first establish a trust relationship with the CA that issued the peer's certificate. Installing the CA's self-signed certificate into the node's *trusted root* certificate store achieves this end. The identity of any third party that subsequently presents a certificate from this now trusted CA would then be assumed valid. When a group of nodes establish individual trust relationships with a common CA, and are all issued certificates by that CA, the participants form a *Public Key Infrastructure* (PKI).

The participation of a CA greatly reduces the number of trust relationships that must be formed. Without a CA, a fully meshed arrangement in which each node establishes direct one-to-one trust relationships with every other node in the PKI would be required. The number of trust relationships can be calculated from:

$$\frac{(N^2 - N)}{2}$$

where N is the number of nodes.

For a PKI consisting of just 25 members, no fewer than 300 direct trust relationships would be required, as opposed to 25 (the same as the number of nodes) with a CA at the centre of the PKI. The two diagrams overleaf graphically demonstrate two PKI structures with and without a central CA. The direction of the arrows indicates the direction of the trust.



The left-hand diagram depicts a PKI based upon a fully meshed structure, and the right-hand diagram shows how the presence of a CA reduces the number of trust relationships. The solid arrows indicate a trust relationship between either two nodes or between a node and the CA. In the second diagram, even though there is no direct trust relationship between any two nodes, this is implied because they are members of a common PKI. Consequently, any two of the nodes will successfully authenticate one another.

4.3.5 Trust Models

The type of PKI hitherto described is a very simple system in which a node is only able to authenticate other members of the same PKI. Suppose that Alice and Bob wish to authenticate one another but are members of different PKIs. By establishing trust relationships between CAs, the trust model can be rapidly extended. Two different methods are recognised.

4.3.5.1 Cross-certification Method

A cross-certification system can exist where Alice's CA signs the certificate of Bob's CA and vice versa. Because Alice trusts her CA not to misuse its signature, when it signs a certificate bearing the identity and Public Key of another CA, Alice will then trust that CA in exactly the same manner as she would any other node in her PKI. By extension, therefore, she trusts any node whose certificate has been signed by the alien, but now trusted, CA.

4.3.5.2 Hierarchical Method

Another trust model is based on the familiar hierarchical method with the pinnacle represented by a 'root' CA that is implicitly trusted by all the other subservient CAs. Immediately below the root, there is a level of CAs whose own certificates the root has signed. Further levels of CAs may be added whose own certificates a CA in the level immediately above has signed.

This type of hierarchical CA is suitable for constructing an overarching PKI that covers an entire organisation, such as a university, that consists of partially autonomous faculties and departments. The cross certification system is clearly better suited for use by a number of completely autonomous organisations such as separate universities.

4.3.6 Authentication by Certificates

Consider a situation where a server provides some form of network access to remote clients. While in practice both devices are peers of equal standing and check one another's certificates, it is convenient to consider just one of these transactions (the access 'server' authenticating the remote 'client') for illustrative purposes.

The client digitally signs (but does not encrypt) its certificate and sends it to the server. The server extracts the Public Key from the certificate and uses this to decrypt the signature. By comparing the digest from the decrypted signature with a digest it computes, the server verifies that the sender of the certificate is also its holder. A similar check is now performed on the CA's digital signature that has been extracted from the client's certificate. This second check can only be performed if the server knows the CA's Public Key and this will only be the case if the server holds a copy of the CA's self-signed certificate. The server must trust the client's CA in order to verify the certificate that it issued to the client. It should now be evident why the two security peers must belong to a common PKI in order for the mutual authentication process between them to conclude successfully.

An additional optional step in the authentication procedure involves the peers consulting the CA's Certificate Revocation List (CRL). This is a mechanism by which a CA can revoke a certificate prior to its expiry date. Such a mechanism allows a college to issue certificates to staff members, and then withdraw the certificate should a member of staff leave the college's employment.

When two security peers exchange certificates, it is worth noting that they obtain one another's Public Keys that can be used for the subsequent strong encryptions of, for example, a shared DES key.

4.4 Authentication Models

VPNs have already been classified according to whether they are used to connect a remote site or a remote user to the main campus. Different models of identity authentication are most appropriate for for these two types of VPN.

Peer authentication within the *remote site* category of VPN can generally be accomplished by configuring a shared secret on both of the peers. Cisco® routers allow the secret phrase to be mapped to the peer endpoint's identity (usually its hostname or IP address). This is akin to the Challenge Handshake Authentication Protocol (CHAP) authentication method associated with PPP and may be considered secure providing both of the endpoint routers reside within the same management domain (for example the college's computer services department). Although local circumstances may dictate otherwise, it should not be necessary to resort to the expense and complexity of installing certificates on the peer routers.

An extensive *remote user* VPN requires a more rigorous approach because individual users, not the college authorities, will control the remote computers. It is important that, should the need arise, the remote access facility can be immediately withdrawn without requiring access to the device concerned. User authentication is insufficiently secure because suspension of an account cannot guarantee that the user would not be able to gain remote access to the network using a colleague's password. If the remote computer is authenticated using a digital certificate, then access can be withdrawn by placing the certificate on the CA's revocation list.

Tunnelling

5

Layer III Tunnelling

Layer II Tunnelling

5. Tunnelling

Many corporate networks are shielded from the outside world by firewall devices or by the simple expedient of running the network on private IP addresses that are not routed over the global Internet. Either or both of these measures may be present at both ends of a VPN, preventing external packets from reaching systems connected to the LAN. However the purpose of a VPN is to allow a remote host or site to become part of the LAN, and so the security measures used to guard against intrusion from the Internet must be selectively circumvented to allow the VPN to work.

The solution is to use tunnelling, in which a packet destined for the remote site is placed inside another (IP) packet with globally routable source and destination addresses. A VPN consists of two stations, known as the tunnel endpoints, that perform the necessary operations. When a tunnelled packet is received by the destination end point, the headers concerned with the tunnel are stripped away to reveal the original packet that is delivered to the final destination.

All tunnelling involves encapsulation of the original packet or frame before it is released onto the Internet. Three protocols are involved in the process:

- the **passenger protocol** is the original packet and will either be a network layer protocol (IP, IPX, AppleTalk® etc.) or a PPP frame;
- the **encapsulating protocol** is the transport-layer protocol that is wrapped around the original data;
- the **carrier protocol** is used by the network carrying the tunnelled packet and must be IP if tunnelling is to take place over the Internet.

There are two types of tunnelling, distinguished by the nature of the passenger protocol. In a Layer III tunnel, the passenger protocol is a network-layer protocol such as IP or AppleTalk®. In Layer II tunnelling, it is a data-link protocol (usually PPP).

5.1 Layer III Tunnelling

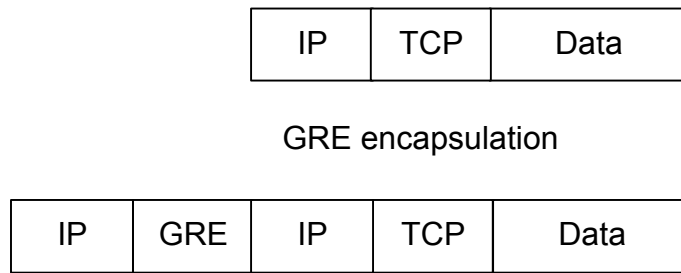
An IP datagram is encapsulated and this takes place before any Layer II components (headers and trailers) have been applied. This type of tunnelling is often associated with inter-site VPNs in which the two tunnel endpoints are both perimeter routers.

There are a number of types of Layer III tunnelling that are distinguished by the encapsulating protocol employed. This guide concerns itself with the two commonest types; Generic Routing Encapsulation (GRE) tunnels and IPSec tunnelling mode, which is covered in *Chapter 6*. The more obscure protocols are too specialised to warrant further mention.

5.1.1 GRE Tunnelling

This is a protocol developed by Cisco® that can encapsulate a wide variety of packet types within IP tunnels. It is a transport (Layer IV) protocol operating at the same level as TCP, User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) and is assigned IP protocol number 47. When employed natively, GRE is suited to static tunnels that remain configured on the two tunnel endpoints, often Internet-connected routers, regardless of whether data is flowing at any given instant. Several other tunnelling technologies also use enhanced versions of GRE as the encapsulating protocol, including the Layer II tunnelling technology Point-to-Point Tunnelling Protocol (PPTP) that is discussed in *Section 5.2.4*.

The encapsulation process is illustrated below and involves the prepending of a GRE header and a fresh IP header to the original datagram.

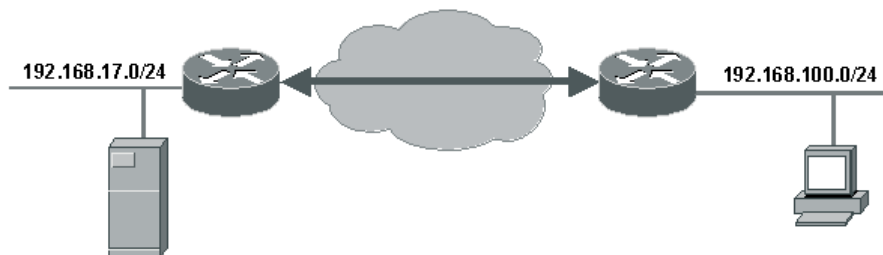


Usually, the destination of the original packet is not deliverable over the Internet because it is an address from a private IP range. The tunnelling process creates a new IP header with a destination address that is the remote end-point of the tunnel and that *is* reachable over the Internet. When the peer receives this packet, it is decapsulated (the IP and GRE headers are removed), thereby exposing the original IP header. The reconstituted datagram is then routed within the internal network and delivered to the specified station.

The GRE protocol header supports checksums, keys and sequencing, all of which are optional.

- The **checksum field** contains the IP (one's complement) checksum sum of all the 16-bit words in the GRE header and the payload packet.
- The **key field** contains a four-octet number that was inserted by the encapsulator. The key can identify an individual traffic flow within a tunnel should this facility be required. It should *never* be used as a form of weak security as simple packet sniffing will reveal its value.
- The intended use of the four-byte **sequence field** is to provide unreliable but in-order delivery. If a key is used, the sequence number is specific to the traffic flow identified by the key field.

As a simple example of a GRE Layer III tunnel in action, consider two colleges each with connections to JANET.



One of the colleges operates a student records database server on their privately numbered administrative LAN and wishes to allow the other college access to the server. For security reasons, it is decided not to create a network address translation mapping to the server's IP address within the border router. Instead, a GRE tunnel is established between the two sites by configuring logical 'tunnel' interfaces on each of the routers and then routing traffic to the peer site's privately numbered network via the tunnel interface. Elements of the configuration of the routers are shown on the following page.

Traffic destined for the internal (privately-addressed) network at the far end of the tunnel is statically routed via the tunnel interface on both routers. Any packets leaving a tunnel interface have a GRE and a new IP header added. The source and destination fields of the outermost IP header are filled with the IP addresses of the serial interfaces of the two routers. The tunnelled packet then leaves the source router via its serial interface and is routed over the Internet to the peer router.

```

configuration of router A
!
interface Tunnel0
 ip unnumbered FastEthernet0/0
 tunnel source Serial0/0
 tunnel destination 193.61.71.250
!
interface FastEthernet0/0
 ip address 14.83.103.186
 duplex auto
 speed auto
!
interface Serial0/0
 bandwidth 2048
 ip address 193.61.71.246 255.255.255.252
!
ip route 0.0.0.0 0.0.0.0 Serial0/0
ip route 192.168.100.0 255.255.255.0 Tunnel0

configuration of router B
!
interface Tunnel0
 ip unnumbered FastEthernet0/0
 tunnel source Serial0/0
 tunnel destination 193.61.71.246
!
interface FastEthernet0/0
 ip address 192.168.100.254 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0
 bandwidth 2048
 ip address 194.83.103.186 255.255.255.252
!
ip route 0.0.0.0 0.0.0.0 Serial0/0
ip route 192.168.17.0 255.255.255.0 Tunnel0

```

5.2 Layer II Tunnelling

In Layer II tunnelling, the encapsulation process takes place after the data-link header and trailer have been applied. The corollary of this is that the Layer II tunnelling process cannot be applied selectively (deciding whether to tunnel a packet based on, for example, its source or destination IP addresses). A station with active Layer II tunnelling blindly dispatches all outbound traffic to its tunnelling peer.

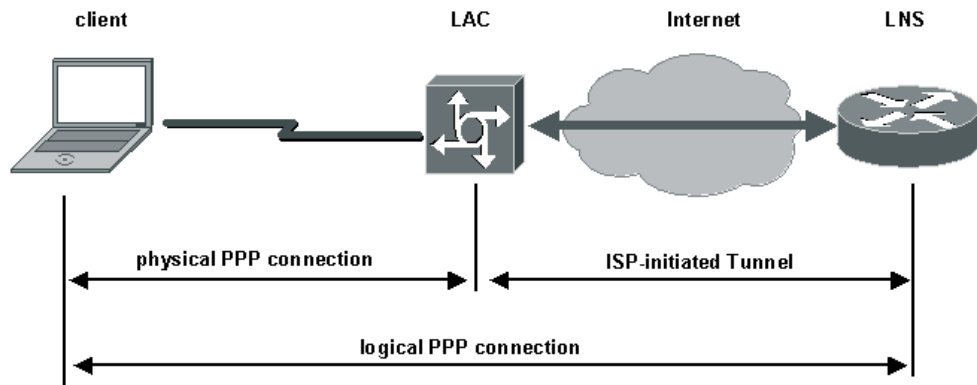
There are three Layer II tunnelling protocols, the most recent of which, Layer II Tunnelling Protocol (L2TP), is a synthesis of the two earlier protocols, Layer II Forwarding (L2F) and PPTP. This document discusses L2TP and PPTP.

There are two possible modes of Layer II tunnelling. Compulsory tunnelling means that the tunnel is always active while voluntary tunnelling allows the user to dial the ISP and then decide whether to initiate a Layer II tunnel.

5.2.1 Compulsory Tunnelling

This type of tunnelling requires the active participation of the ISP. The client station forms a connection to the L2TP Access Concentrator (LAC) via a media that supports PPP such as a dial-up modem or an ADSL connection. The LAC is operated by the ISP and would most often be a dial-up server of some description. Upon receipt of frames from the client station, the LAC

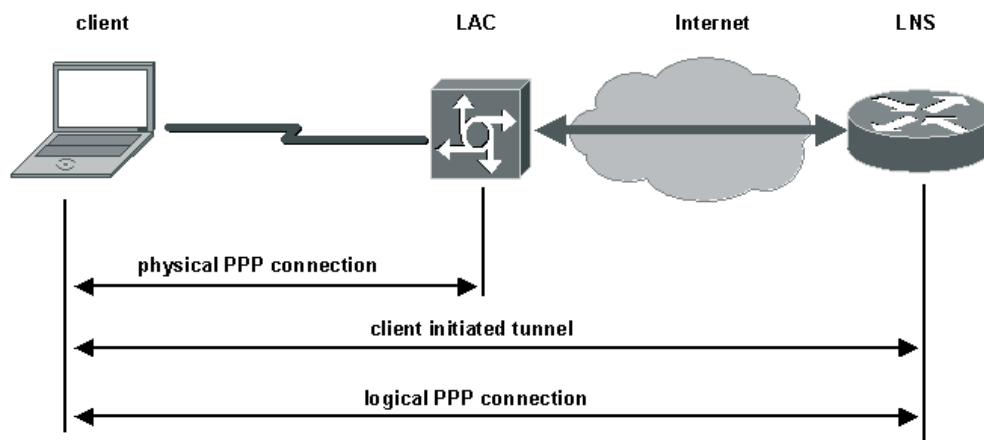
securely tunnels these to the L2TP Network Server (LNS) without any required knowledge or participation by the client. The LNS would be, for example, an Internet-connected router residing at a company's offices with the client station representing a teleworking employee.



Because every frame is dispatched via the tunnel, then from the perspective of the client station there exists a direct PPP connection between itself and the remote tunnel endpoint, the intervening infrastructure being completely transparent. This arrangement can be highly advantageous as it allows organisations to provide the benefits of a dial-up service to their remote employees without incurring the support burden of actually operating the dial-up aspect of the service. It does, however, require a special ISP service.

5.2.2 Voluntary Tunnelling

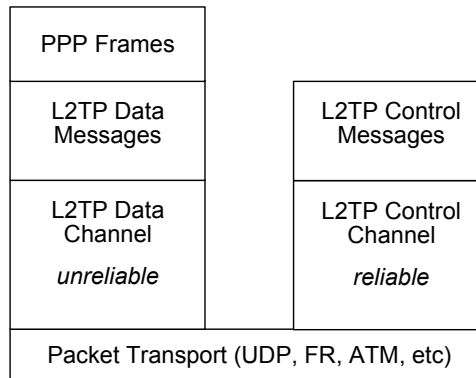
If it is deemed preferable for the client station to initiate the tunnelling process, the dial-up ISP need play no role other than providing normal Internet connectivity.



The encapsulation process is similar to the compulsory tunnelling scenario except that the client station assumes the tunnelling role that would otherwise be performed by the ISP's dial-up server hardware. The advantage of voluntary tunnelling is that the client can obtain general Internet connectivity by dialling the ISP and need only initiate the tunnel when it is needed.

5.2.3 Layer II Tunnelling Protocol (L2TP)

This is the most recent member of the family of tunnelling protocols for Layer II frames. L2TP is not limited to IP networks, being able to tunnel non-IP protocols such as AppleTalk® and IPX over non-IP networks such as Frame Relay (FR) or Asynchronous Transfer Mode (ATM).



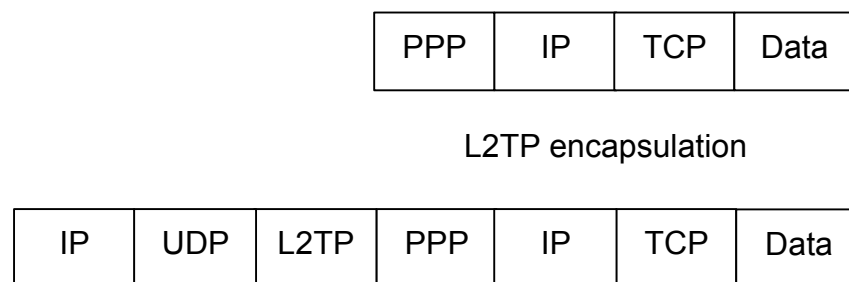
L2TP utilises two types of messages, control messages and data messages. Control messages are used in the establishment, maintenance and clearing of tunnels and calls. Data messages are used to encapsulate PPP frames being carried over the tunnel. Control messages utilise a reliable channel within L2TP to guarantee delivery. Data messages are not retransmitted by L2TP when packet loss occurs. Sequence numbers (within the L2TP header) are required for control messages so that their delivery is guaranteed. L2TP messages include a *Next-Received* field and a *Next-Sent* field, which are comparable to TCP's *Acknowledgement Number* field and *Sequence Number* field, respectively. Delivery of control messages has to be managed in-band by L2TP itself because the carrier network may not offer reliable delivery. These messages take the form of Attribute-Value Pairs (AVPs).

5.2.3.1 Security

No specific allowance is made within the L2TP specification for security. However, since L2TP can run over IP, it is possible to deploy transport mode IPSec to provide security services.

5.2.3.2 L2TP Over IP

L2TP data tunnelling begins with a PPP payload. The PPP frame is encapsulated with a L2TP header. The encapsulating protocol, UDP, is applied. L2TP uses UDP port 1701 for both the source and destination port. Finally, the outermost IP header can be applied.



5.2.3.3 Availability

Because L2TP is a synthesis of Cisco®'s earlier L2F and Microsoft®'s PPTP, both these vendors offer L2TP natively with their operating systems. Although Cisco® routers support L2TP as part of the standard Internetworking Operating System (IOS), most users will wish to deploy the new security features. IPSec is only provided as part of the 3DES Firewall feature set, which is a chargeable option. Microsoft® provides a full implementation of L2TP with Windows® 2000 and later. It is available as part of the dial-up networking upgrade for Windows® 9x.

5.2.4 Point-to-Point Tunnelling Protocol (PPTP)

This protocol was developed jointly by an alliance of vendors. In contrast to L2TP, it can only tunnel PPP frames over an IP carrier network. Two connections are required for a PPTP tunnel; a data tunnel and a separate control connection operating on TCP port 1723. After the two peers have established a tunnel, they send PPTP control-connection packets back and forth to maintain the connection. These control-connection packets consist of PPTP Echo-Request and PPTP Echo-Reply messages. The data connection utilises GRE as the transport protocol and it is this fact that restricts the carrier protocol to IP. The separate control channel is required because GRE lacks the functionality to set up a session. Secure authentication is provided by means of Microsoft® Challenge Handshake Authentication Protocol (MS-CHAP) and the data stream may be secured using Microsoft® Point-to-Point Encryption (MPPE) that uses RC4 (Rivest's Cipher 4, also known as Ron's Code 4), a stream cipher, to encrypt the PPP datagrams.

Security concerns have dogged PPTP since its inception. It is the author's opinion that PPTP is inherently insecure because there are too many unauthenticated control packets that are readily spoofed. The availability of L2TP secured with IPSec has now rendered PPTP obsolete.

IP Security: overview and architecture

6

IPSec Security Associations (SAs)

IPSec Modes

IPSec Protocols

6. IP Security: overview and architecture

There are no security provisions within the IP standard that guarantee that received packets:

- originate from the claimed sender;
- have not been inspected or modified by a third party during transmission;
- have not been replayed from earlier transmissions.

Most of the traffic sent over JANET or the Internet is not of a nature to warrant concerns about these matters, or alternative application-level methods are available to verify the origin or encrypt transmissions. For example, HyperText Transfer Protocol Secure (HTTPS) is a secure version of HTTP available for transmitting sensitive data (such as credit card details) over the worldwide web.

A VPN, however, will cause internal data, which is assumed to be sensitive, to be transmitted over an external shared network. Furthermore, users will expect access to all of the available local network services, and so a solution that relies upon individual application security features will not be suitable. This chapter examines the security extensions to the IP standard, IPSec, that provide a framework within which encryption and authentication algorithms may be applied to IP packets.

IPSec is a suite of three transport-level protocols used for authenticating the origin and content of IP packets and, optionally, for the encryption of their data payload. Two of the protocols, Encapsulating Security Payload (ESP) and Authentication Header (AH), provide authentication, comprising proof of data source, data integrity and anti-replay protection. Additionally, ESP (but not AH) provides data encryption. The third IPSec protocol, Internet Key Exchange (IKE), is a complex hybrid protocol used for the peer authentication and key exchange processes that necessarily precede the services provided by AH and ESP.

The IPSec protocols do not define *which* algorithms should be used for the computations involved in encryption or in generating digital signatures. This renders the protocol definitions completely generic, meaning they can accommodate developments such as new cryptographic techniques as they become available. The algorithms to be used are specified separately as part of the overall security policy configured at each of the peer stations. The initial IKE negotiations allow the peers to agree on the particular combination of protocols and algorithms to be used for all subsequent IPSec processing.

6.1 IPSec Security Associations (SAs)

These are fundamental to the operation of IPSec. Prior to the transmission of protected datagrams, the two peer stations must reach an agreement on how the conversations between them should be processed by IPSec. They agree upon the protocol, the transform, the keys and the key lifetime. This ‘contract’ between a pair of IPSec peers is the SA.

The SAs are simplex (i.e., unidirectional) in nature. If two stations, X and Y, are communicating securely, then each will maintain two SAs, one for outbound and one for inbound traffic. It should be apparent therefore that $(SA_{out})^x$ would share the same cryptographic parameters with $(SA_{in})^y$.

The defining characteristics of an active SA are the result of a negotiation between two IPSec peers. Each peer must be configured in advance with the selection of protocols and transforms it is willing to accept from or is able to offer to a peer. Suppose station X is configured to offer ESP with the DES encryption and the MD5 integrity algorithms and to offer AH with the MD5 algorithm, while station Y is only configured to use AH with the MD5 algorithm, then the SA negotiation between these two peers would result in each agreeing to protect their conversations with the AH protocol and the MD5 algorithm. Because station Y has not been configured to use ESP, then it will not form SAs employing this protocol. These predefined policies are known as the Security Policy Database (SPD). Gross failures in IPSec processing are often caused by a lack of any common protocol or algorithm in the SPDs of two peers. Thus the initial IKE main mode negotiations fail to reach an agreement and no SAs can be established.

6.1.1 Security Parameter Index (SPI)

When an IPSec-protected datagram is received, it is clearly important that the station to which the packet has been sent is able to determine which of its SAs it should use when processing the secured packet. The SPI is an arbitrary 32-bit value that, together with the destination IP address of the outer IP header and the protocol (AH or ESP), uniquely identifies the SA to the receiving station. The sending station includes an SPI with each packet it dispatches identifying the receiving peer's inbound SA. If the SA cannot be recognised, the then receiving station drops the packet without attempting any further processing. In database terminology, the *<SPI, destination IP address, protocol>* tuple¹ may be regarded as the primary key of the receiving station's SA Database (SADB).

6.1.2 Sequence Numbers

The sender dispatches this unique and monotonically increasing 32-bit unsigned integer with each secured packet to guard against replay attacks. The receiving station checks this field to determine whether the packet is a duplicate of one that has already been received. The sequence number is incremented by one for each packet processed through a given SA. The SAs are usually renegotiated before this number overflows.

Sequence numbers are required to protect against a *replay attack* in which an attacker intercepts and stores packets emanating from the sending station. The attacker then floods the receiving station by repeatedly resending these intercepted packets. This is a form of denial-of-service attack.

6.1.3 Management of SAs

The management of SAs concerns their creation and deletion. Creating security associations is a two-stage process in which parameters are first negotiated with the IPSec peer and the SADB is then updated with the new SA. It is mandatory for all implementations of IPSec to support manual keying in which all the necessary parameters are agreed off-line (e.g. by means of a telephone conversation). Manually constructed SAs never expire and the processes involved in their creation are cumbersome and insecure. It is preferable, therefore, that a key management protocol such as IKE is deployed to create the SAs. Various triggers cause an SA to be deleted. These comprise:

- expiration of the key lifetime;
- compromised keys;
- the number of bytes processed through the SA reaching a threshold;
- the security peer requesting that the SA is torn down.

In order to avoid unnecessary interruptions, a new replacement SA will be negotiated a short time before the existing one is deleted. Once this new SA has been fully established, it will be used immediately and the older one will be deleted shortly thereafter.

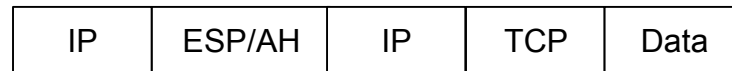
6.2 IPSec Modes

The principal function of IPSec is to provide a standards-based framework defining how IP packets may be protected during their transmission. There is also an important ancillary function whereby IPSec provides the functionality to tunnel the packets it is protecting. Prevailing circumstances determine whether it is necessary to tunnel the protected packets, and so IPSec may be operated in two different modes wherein the tunnelling functionality is either active or suppressed. The difference between the two modes concerns the location of the IPSec header within the original IP packet. This in turn affects the degree of protection that IPSec affords.

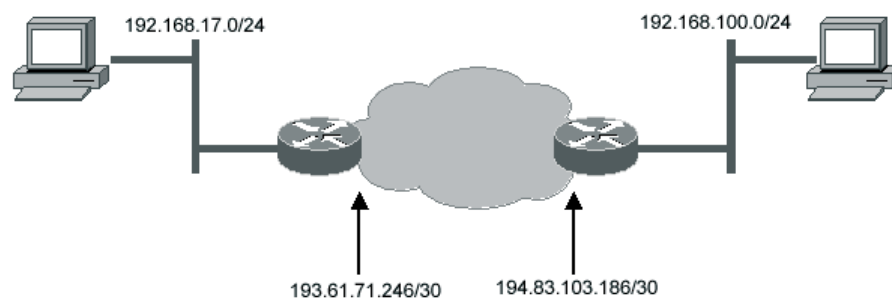
¹A tuple is an ordered set of values (often separated by commas) and is analogous to a record in a non-relational database.

6.2.1 Tunnel Mode

When the destination of the original, unsecured packet is not the same as the remote security endpoint, IPSec must operate in tunnel mode. The IPSec header and a new IP header are prepended to the original packet as shown.



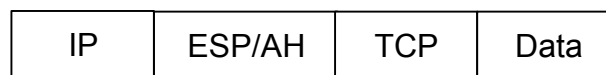
The corollary is that in tunnel mode, protection is afforded to the entire IP packet and not just its payload between the tunnel end points. The diagram below depicts a typical scenario where IPSec would be required to operate in tunnel mode.



The two communicating stations are connected to privately numbered networks and cannot therefore communicate directly. The Internet-connected routers act as the security endpoints. Packets are received from the sending station and encapsulated by the source router. The new outer IP header has the Internet-valid router addresses as the source and destination. When the tunnelled packets arrive at the destination router, the outer IP header is removed and the original IP packet is regenerated as a result of the IPSec processing. This packet is then transformed into an Ethernet frame and delivered to the local destination in the usual manner.

6.2.2 Transport Mode

Here, the IPSec header (AH or ESP) is inserted between the original packet's network and transport headers as shown below.



Consequently, the IP *payload* is protected in transport mode, but the *header* is not. Transport mode is suitable under two circumstances.

Firstly, if the two stations participating in the underlying communication are also the IPSec peers, then there is clearly no need to tunnel the secured packets. This will usually be the case if the two stations are able to communicate directly because they are operating on Internet-valid IP addresses.

Secondly, circumstances may dictate that while IPSec is used to provide security services, an alternative tunnelling technology should be deployed. For instance, IPSec may be used to secure packets that have been routed through a tunnel interface on a Cisco® router. The Layer III tunnel is here accomplished by means of some other (normally GRE) encapsulation and it is therefore appropriate for IPSec to operate in transport mode, as an outer IP header has already been provided. This may seem to be a contrived method of achieving the same object as tunnel-mode IPSec, but there are cases when such techniques are required, and a full example is discussed in *Section 7*.

6.3 IPsec Protocols

The two IPsec protocols, AH and ESP, both operate at the same transport layer of the Open Systems Interconnection (OSI) model as the more familiar protocols such as TCP and UDP. In order to protect a packet with IPsec, that packet must be encapsulated with one of these two protocols.

6.3.1 Authentication Header

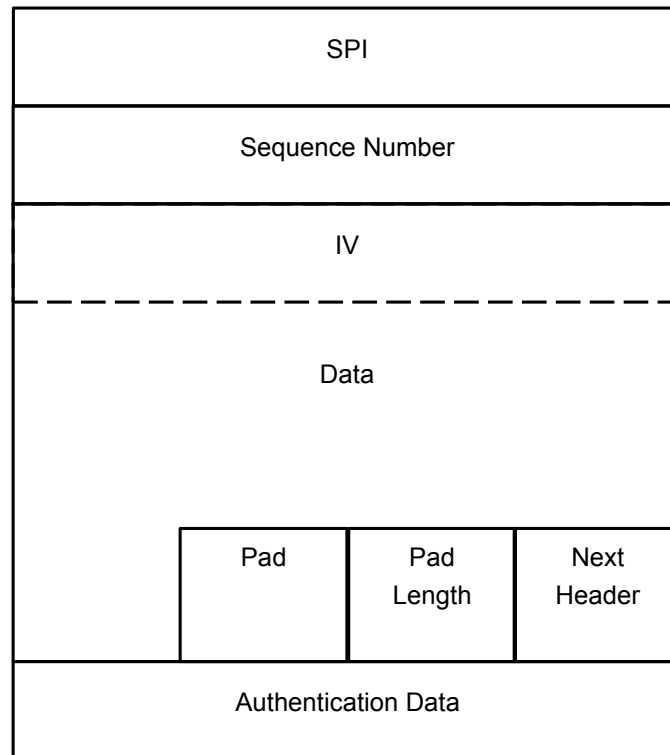
Because AH provides only authentication (and not confidentiality), it is a fairly simple protocol creating just a header and no trailer. The header contains the SPI, the sequence number and the authentication data.

This latter field contains the Integrity Check Value (ICV), which offers assurance that the packet has not been altered during its transmission. The SA specifies the algorithm used to calculate this value. A family of suitable algorithms is the MAC such as MD5 or SHA-1. The Next Header is an 8-bit field indicating what protocol follows the AH header. When IPsec is running in tunnel mode, the value of this field will be 4 (IP-in-IP). In transport mode, it will be the upper-layer protocol being protected (usually UDP or TCP).

Next Header	Payload Length	Reserved
SPI		
Sequence Number		
Authentication Data		

6.3.2 Encapsulating Security Payload (ESP)

This protocol must be used when data encryption is required. Because it is an IPsec header, ESP provides SPI and sequence numbers whose purposes have been discussed in the preceding section. Padding (to a maximum of 255 bytes) is used by ESP to preserve byte boundaries. If an encryption algorithm is employed that requires the plaintext to be a multiple of some number of bytes (for example the block size of a block cipher), the padding field is used to fill the plaintext to the requisite size. Padding may also be required, irrespective of encryption algorithm requirements, to ensure that the resulting ciphertext terminates on a 4-byte boundary, thereby right-aligning the trailer.



6.3.3 Internet Key Exchange (IKE)

A suitable SA must exist before packets can be secured using AH or ESP. The third member of the IPsec family of protocols is IKE and it is used for peer authentication, negotiation of keys and for the dynamic construction of SAs. Although the fine details of IKE are complex and lie outside the scope of this guide, an understanding of the methods used in the establishment of IPsec SAs is invaluable when troubleshooting.

A hybrid protocol, IKE combines parts of the Oakley key determination protocol and the Security Key Exchange Mechanism (SKEME), both key exchange protocols, with the Internet Security Association Key Management Protocol (ISAKMP). The latter defines a framework for peer authentication, key exchange and SA management over an IP network and operates on UDP port 500.

An IKE negotiation consists of two phases in which an IKE SA is first established (Phase 1) to provide a secure communications channel through which SAs for other protocols (e.g. IPsec) may be constructed (Phase 2). There are two mutually exclusive methods by which a Phase 1 IKE negotiation may proceed; main mode or aggressive mode. The phase 2 negotiations are conducted using a quick mode exchange.

6.3.3.1 Main Mode

A main mode exchange uses six messages in three round trips. In the first exchange, the peers negotiate the parameters of the IKE SA and how the remaining exchanges will be accomplished. The next two exchanges are used for the exchange of the Diffie-Hellman keying material. The final pair of main mode messages is used for authenticating the peers' identities. This last exchange is encrypted using the previously negotiated key, thereby protecting the identities of the peers from eavesdroppers.

6.3.3.2 Aggressive Mode

An aggressive mode exchange accomplishes the same things as main mode, but using only half the number of messages by embedding some of the messages within others. Consequently, the negotiating abilities of aggressive mode are limited and peer identities are not concealed.

6.3.3.3 Quick Mode

After an IKE SA has been established by means of a Phase 1 exchange (in either main or aggressive mode), it can be used to construct the IPSec SAs using a Phase 2 exchange in quick mode. With the exception of the ISAKMP header, all quick mode exchanges are encrypted and authenticated.

The security peers exchange the IPSec keys (one for each SA) as part of the quick mode exchange. This is done by first exchanging pseudo-random nonces¹ that are hashed with the IKE shared secret. The resulting unique IPSec keys do not have the property of Perfect Forward Security (PFS) because they are all derived from the same IKE shared secret. With PFS, if one key is compromised, previous and subsequent keys remain secure, because subsequent keys are not derived from previous keys. An additional Diffie-Hellman exchange is performed with each quick mode exchange if PFS is specified in the IPSec policy. Because each Diffie-Hellman exchange requires large exponentiations, PFS will exact a performance cost. The shared secret derived from a quick mode Diffie-Hellman exchange is used to generate IPSec keys that have no 'memory' of their predecessors.

6.3.4 IPSec Domain of Interpretation (DOI)

The IKE protocol defines *how* security parameters are negotiated and shared keys are established for other protocols. It does not define *what* to negotiate. That is the function of a DOI document that defines, amongst other things, the attributes that IKE negotiates in quick mode. There currently exist DOI documents for IPSec and for the routing protocols Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

In summary, IKE may be regarded as an authenticated Diffie-Hellman exchange with ISAKMP providing the necessary network framework.

¹ A nonce is a parameter that varies with time.

Implementation and Worked Examples

7

Configuring IPSec on Cisco® Routers

Configuring IPSec on Windows® 2000

Configuring an IPSec-Protected GRE Tunnel

Configuring Tunnel Mode IPSec

7. Implementation and Worked Examples

This chapter shows how the facilities provided by IPSec can be used in practice to create secure VPNs. The examples use Cisco® routers and Windows® 2000 workstations. These devices have been chosen because they are widely used and most readers will have access to hardware similar to that discussed in the examples. Two common requirements are discussed: providing a secure VPN tunnel between two private networks, for example a remote site or office and a main campus, and providing a secure remote access service for staff working at home.

A VPN must be prepared in advance by defining its policy and the security technologies it will support. Each participant in a VPN has its own set of policies and technologies, described in an SPD. The policies in the SPD define what traffic should be secured and how that security should be applied. For example the policy may define certain characteristics, such as source and destination addresses, that require encryption and/or authentication. In this way IPSec processing may be restricted to certain packets only. The technologies part of the SPD defines what protocols and algorithms the device will offer to its peer during the negotiation phase, both for authentication and encryption.

When a VPN connection is created, the end points negotiate on the basis of their own SPDs and, provided agreement can be reached, an SA is created that defines the connection. If the peers cannot agree a set of technologies, for example because there are no encryption algorithms supported by both ends or because the digital signature offered as proof of identity by one peer is considered too weak by the other, then the VPN will not be established.

7.1 Configuring IPSec on Cisco® Routers

The Cisco® IOS is not generally supplied with IPSec, as it is a chargeable option. An approved vendor's pre-sales advice should always be sought before purchasing any new equipment. Specifying the 3DES version of the 'Firewall' Feature Pack should ensure that new equipment has full support for IPSec pre-installed. Installing IPSec functionality also requires additional flash and working memory and consideration should be given to providing sufficient processing power. Any router that is required to perform a lot of IPSec should be fitted with a hardware accelerator module so that the main processor does not become overloaded.

Configuring IPSec on a Cisco® device comprises the four stages that are outlined below. In the interests of brevity and pertinence, this document does not discuss any subsidiary Cisco® technologies such as Access Control Lists (ACLs) or the general operation of Cisco®'s command-line IOS.

7.1.1 Configure Crypto Lists

The router must be configured with an ACL that specifies the traffic that should be subject to IPSec processing. This may be of the usual standard or extended types. If the former, then only the source address is matched while extended lists can match the protocol and source and destination addresses. In this context, a 'permit' statement in the ACL means that the matching packets should undergo IPSec processing. It is the author's opinion that crypto lists should not contain any 'deny' statements, although the router will process such lines. The implicit denial at the end of a list should be sufficient for almost all cases.

7.1.2 Configure Transform Sets

The router is configured with the set of protocols and algorithms that it will offer to a peer during negotiation of the IPSec SAs. The protocols will be either AH or ESP and the algorithms refer to the encryption and one-way hash functions to be used in conjunction with the selected protocol(s). The desired IPSec mode must also be selected here, the default being tunnelling.

7.1.3 Apply Crypto Maps

The two preceding tasks define what traffic the IPSec kernel should process and how that processing should be performed. These two strands are drawn together as a named policy called a crypto map that must be bound to the relevant egress interface. Any traffic leaving this interface will be checked against the relevant crypto list to determine whether it should be passed to the IPSec kernel for encapsulation before leaving the interface.

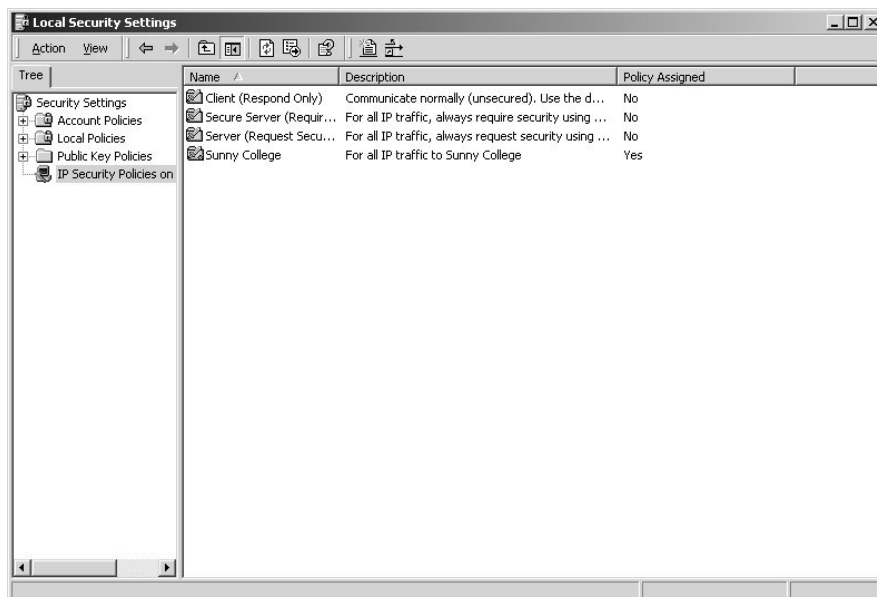
7.1.4 Configure Key Exchange Policies

An IKE policy must be configured so that the router is aware of how to authenticate the remote peer and how to conduct the key exchange. The variables comprise the encryption and hashing algorithms, the authentication method and the Diffie-Hellman group identifier.

This recipe is a general overview of the necessary stages in configuring Cisco®'s implementation of IPSec. There are many subtleties and additional techniques some of which, such as dynamic crypto maps, are demonstrated in the examples in *Sections 7.3* and *7.4*. The interested reader should consult the official Cisco® documentation for a complete and authoritative discussion.

7.2 Configuring IPSec on Windows® 2000

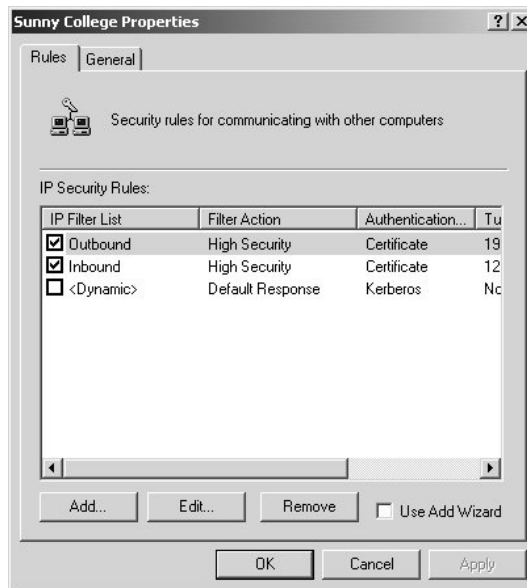
All versions of Microsoft® Windows® 2000 are supplied with a full implementation of IPSec. The Microsoft® Management Console provides a snap-in called 'IP Security Policy Management' through which all aspects of IPSec may be controlled by constructing and applying policies. Access to this interface is most readily obtained by running a program called 'secpol.msc' and selecting the 'IP Security Policies on Local Machine' node.



Screen shot printed by permission from Microsoft Corporation.

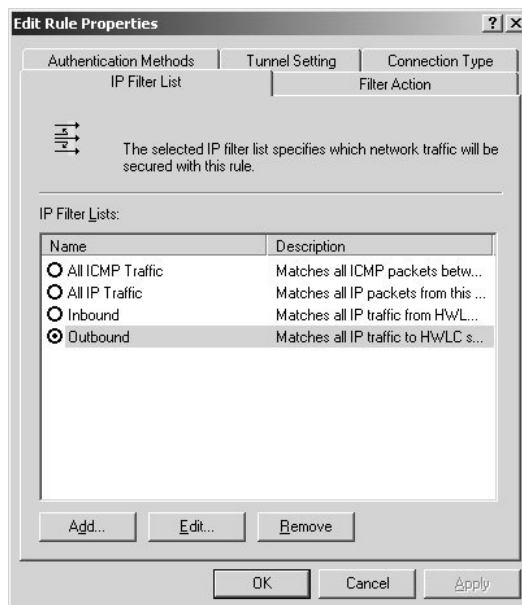
In the screen shot above, there are four IPSec policies, three of which are pre-installed but are not active. The fourth policy, called 'Sunny College' has been assigned and is therefore actively processing traffic.

Double-clicking a policy in the right-hand list view displays a properties page listing the rules that comprise the policy.



Screen shot printed by permission from Microsoft Corporation.

Double-clicking a rule in the policy's properties page displays all of its characteristics. A rule is composed of a filter list, a filter action, an authentication method and an optional tunnel setting, which is not required for transport-mode IPsec.



Screen shot printed by permission from Microsoft Corporation.

7.2.1 Filter Lists

Multiple filters may be combined to create a filter list that specifies which traffic should be protected by the IPsec policy. These filter lists are equivalent to Cisco®'s extended ACLs in that source and destination addresses or subnets may be specified as well as the IP protocol and, where appropriate, port number. There are two restrictions when constructing filters. Firstly, subnets may only be expressed using traditional classful masks and so multiple subnets cannot be summarised with a single classless mask. Secondly, a tunnelling mode policy does not support protocol or port specification in the filters. Only IP addresses or subnets may be used in filter lists for a tunnelling mode IPsec policy.

Some traffic is automatically exempted from IPSec regardless of the filter lists. Firstly, only unicast traffic may be secured. Any packet sent to a broadcast or multicast address will be unprotected. This is to be expected, as IPSec requires SAs to be negotiated between two peer stations before any secured packets may be transmitted. By definition, a sender does not know the identity of receiving stations when transmitting to broadcast or multicast addresses. Any IKE-related traffic on the ISAKMP port (UDP/500) will not be secured because this traffic is concerned with negotiating IPSec SAs.

Kerberos traffic (UDP/88) is also exempted from IPSec processing on Windows®. Unlike the other exemptions, this is not a consequence of the IPSec paradigm. Kerberos is the security protocol used during authentication against a Windows® 2000 user database. If tunnel mode IPSec is required to relay packets between a remote Windows® client and a domain controller, then any authentication traffic generated by the client will never reach the server because the IPSec exemption prevents the packets from proceeding via the tunnel. A modification to the Windows® registry suppresses this behaviour. The registry editor should be used to locate the following key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC
```

The following value should be added to this key:

```
Value Name: NoDefaultExempt  
Data Type: REG_DWORD  
Value: 1
```

Once the machine has rebooted, this value will take effect. Windows® authentication will fail over an IPSec tunnel if this value is absent.

7.2.2 Filter Actions

Having decided what traffic should be protected by means of a filter list, the policy must specify the desired combination of algorithms and IPSec protocols. Either AH or ESP may be selected along with the appropriate hashing algorithm and, in the case of ESP, either the DES or 3DES encryption algorithms.

7.2.3 Authentication Methods

The preferred authentication method is to use a certificate, which is selected by specifying the relevant CA's self-signed 'trusted root' certificate. If the shared secret method is preferred, then a new registry value must be added to suppress the automatic filter that enforces CA authentication. The registry editor should be used to locate the following key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rasman\  
Parameters
```

The following value should be added to this key:

```
Value Name: ProhibitIpSec  
Data Type: REG_DWORD  
Value: 1
```

Once the machine has rebooted, this new value will take effect. Shared secret authentication will fail if this registry value is absent.

7.2.4 Tunnel Setting

For tunnelling mode IPSec, two rules are required, one for the outbound traffic and a second for the inbound. This is because only a single tunnel endpoint may be specified within a rule and so two rules are required, one for each end of the tunnel. From the perspective of one peer, one rule will apply to inbound traffic as the tunnel endpoint will be itself and the other rule will apply to outbound traffic with the tunnel endpoint being the remote peer.

7.3 Configuring an IPSec-Protected GRE Tunnel

Consider two sites, each with networks operating on shared-media Ethernet and a range of private IP addresses (see the diagram in *Section 6.2.1*). A Cisco® router performing Network Address Translation (NAT) provides Internet connectivity via an E1 (2 Mbit/s) leased-line connection to an ISP. Suppose the two sites wish to communicate using their existing external connectivity. One solution would be to create NAT mappings within each router's configuration so that every station that needs to be visible to the remote network can be reached by means of a public IP address.

This arrangement is undesirable on three counts:

- increased consumption of public IP addresses may limit the number of machines that can be made reachable;
- machines that had been secure from ill-intentioned crackers on the Internet by virtue of running on private address space have now been exposed, and although traffic filters on the router might reduce the risks, mistakes in their construction could still allow an intruder some form of access;
- the traffic between the sites may be of a confidential nature and should not be transmitted over an insecure medium such as the Internet without first authenticating and encrypting the packets.

At first glance, IPSec in tunnel mode appears to be the ideal solution, but in this instance it will not work because any traffic entering the router from the internal network will be operated upon by the NAT first, and the source address changed. Therefore, the crypto will never be triggered. By routing any traffic to the remote site through a 'Tunnel' interface, the packets will not be operated upon by NAT. If a crypto map is bound to the tunnel interface, then IPSec protection will also be afforded to any traffic leaving via the tunnel. In this case, the tunnel is established by means of the initial GRE encapsulation, and so it is appropriate for IPSec to run in transport mode. Relevant excerpts from the configuration of one of the routers are given overleaf.

```
!the IKE policy defines the encryption, authentication, Diffie-  
!Hellman group, authentication method and IKE SA lifetime  
crypto isakmp policy 1  
  encryption des  
  hash md5  
  group 1  
  authentication pre-share  
  lifetime 1800  
crypto isakmp key gypsy-rose address 194.83.103.186  
!  
!defines a Transform Set comprising DES encryption and MD5  
!authentication  
crypto ipsec transform-set encryp-auth esp-des esp-md5-hmac  
  mode transport  
!  
!a crypto map defines the trigger, transform set, peer identity  
!and SA lifetime  
crypto map toRemoteNetwork 10 ipsec-isakmp  
  set peer 194.83.103.186  
  set security-association lifetime seconds 900  
  set transform-set encryp-auth  
  match address 101  
!  
!note how the crypto map is bound to the Tunnel  
!the Tunnel interface is deliberately excluded from the NAT  
!process  
interface Tunnel0  
  ip unnumbered FastEthernet0/0  
  ip mtu 1438  
  tunnel source Serial0/0  
  tunnel destination 194.83.103.186  
  crypto map toRemoteNetwork  
!  
interface FastEthernet0/0  
  ip address 192.168.17.254 255.255.255.0  
  ip nat inside  
  duplex auto  
  speed auto  
!  
!the crypto map must also be bound to the physical Serial  
!interface as well as the logical Tunnel interface  
interface Serial0/0  
  bandwidth 2048  
  ip address 193.61.71.246 255.255.255.252  
  ip nat outside  
  crypto map toRemoteNetwork  
!  
ip nat pool sunny-net 212.219.176.174 212.219.176.174 prefix-  
length 28  
ip nat inside source list 1 pool sunny-net overload  
ip classless  
  
ip route 0.0.0.0 0.0.0.0 Serial0/0  
ip route 192.168.100.0 255.255.255.0 Tunnel0  
!  
!for non-crypto packets, this line acts as a NAT trigger  
access-list 1 permit 192.168.17.0 0.0.0.255  
!  
!this crypto trigger ensures IPSec headers are applied after !  
GRE encapsulation  
access-list 101 permit gre host 193.61.71.246 host 194.83.103.186
```

When the router receives a packet with the destination address in the 192.168.100.0/24 range, the following sequence of events takes place:

- the router learns from its routing table that the packet should be sent via the Tunnel0 interface;
- the packet enters the tunnel interface, and a GRE and a new IP header are prepended;
- because a crypto map has been bound to the tunnel interface, the tunnelled packet is diverted to the IPSec kernel and is not immediately dispatched via the serial interface;
- if a suitable IPSec SA is not present, the router commences IKE negotiations with the peer (whose IP address is known from the router configuration);
- once an SA has been established, an ESP header is inserted between the outer IP header and the GRE-header;
- the IPSec kernel has completed its processing and so the heavily encapsulated packet is free to proceed to the IPSec peer router by way of the serial interface.

Once GRE encapsulation has been completed, the packets' source and destination addresses will be 193.61.71.246 and 194.83.103.186 respectively. The NAT is configured only to trigger for packets with a source address in the 192.168.17.0/24 range, and so any packets routed through the tunnel interface are shielded from the NAT process.

One subtle consequence of combining transport-mode IPSec with GRE tunnelling is that large packets may be subjected to two fragmentations within the same router before transmission through the serial interface. This is clearly undesirable as it consumes processor cycles on both the source and destination routers and wastes buffer memory and bandwidth. Assuming that the Maximum Transmission Unit (MTU) of both the internal (Ethernet) and external (serial) interfaces of the router is 1500 bytes, then fragmentation of packets of this size can occur following both the GRE and IPSec encapsulations. By reducing the MTU of the tunnel interface to a specific lower value, (as shown in the example configuration) packets are fragmented only once after GRE encapsulation with sufficient clearance to accommodate the ESP encapsulation without the need for a second fragmentation.

7.4 Configuring Tunnel Mode IPSec

With the advent of affordable home broadband based on ADSL technology, it is worth investigating whether this type of connectivity could be used to provide secure remote access to an organisation's network. By employing IPSec tunnelling with a Windows® 2000 workstation at the home end and a Cisco® router at the office end of the tunnel, a remote user can transparently access the office network. The router is configured with 'dynamic' crypto maps in which it is not necessary to specify the IP addresses of the peers ahead of time in the router's configuration. This means that remote users whose ISP assigns IP addresses dynamically can still establish properly authenticated tunnels with the router.

The router configuration is simpler than in the previous example as much of the complexity has been moved to the remote users' Windows® 2000 workstations. The college servers are all located on the privately numbered 192.168.17.0/24 network and so the remote user must tunnel IPSec-secured packets through to this range of addresses. Relevant excerpts from the router configuration follow overleaf.

```

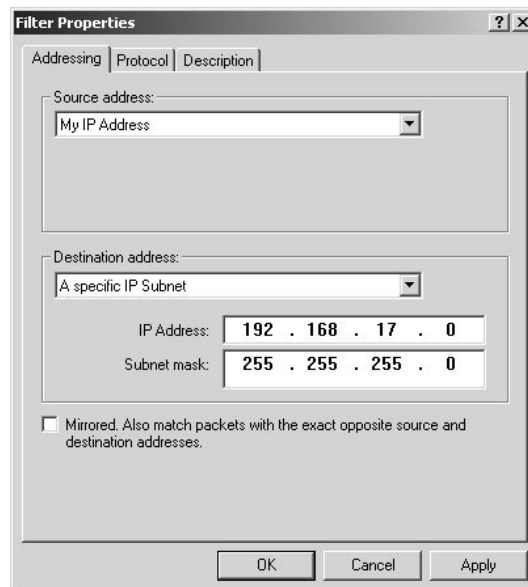
!
crypto ipsec transform-set des-md5-tunnel esp-des esp-md5-hmac
!
crypto dynamic-map toRemoteUsers 10
  set transform-set des-md5-tunnel
!
!
crypto map fromCampus 10 ipsec-isakmp dynamic toRemoteUsers
!
interface FastEthernet0/0
  ip address 192.168.17.254 255.255.255.0
  duplex auto
  speed auto
!
!
interface Serial0/0
  bandwidth 2048
  ip address 193.61.71.246 255.255.255.252

crypto map fromCampus

```

A dynamic crypto map acts as a template for ephemeral crypto maps that are constructed by the router itself when a remote peer initiates a connection. The peer's IP address and a matching crypto list cannot be configured in advance as these contributions to the crypto map can only be known at run-time. Because it is just a template, a dynamic crypto map cannot be directly bound to an interface. Instead, it is associated with a crypto map, which is bound to the interface, as shown in the configuration.

The tunnel will always be initiated from the remote workstation, and so the rules specifying which packets IPSec should process are located on this peer. As discussed previously, the IPSec policy comprises two separate rules, one for outbound and another for inbound traffic. Each rule's filter list comprises just one entry; the filter for the outbound rule is shown below.



Screen shot printed by permission from Microsoft Corporation.

In this example, a specific IP address has not been used as the filter's source. This is necessary if the remote workstation obtains its IP address dynamically. Suppose, however, that college staff have been issued with notebook computers that can be used either from home or directly connected to the college network. When used remotely, traffic to machines on the college network must proceed via a secure tunnel. When the notebook is used at the college, the IPSec policy should be suppressed. Providing the ISP supplies statically assigned IP addresses, by specifying this address as the outbound rule's source, the IPSec policy will not interfere with the correct operation of the notebook when it is connected to the LAN.

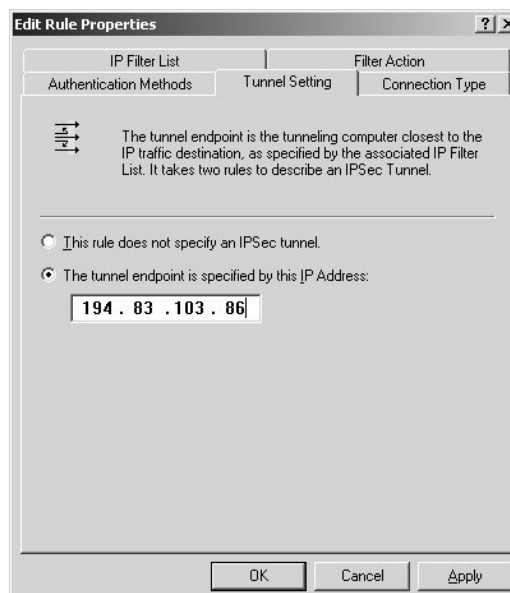
Three security methods (high, medium and custom) are available for the filter action. The first corresponds to ESP with the DES encryption algorithm and the second corresponds to AH. If the 'Custom Security Method' is selected, then all the parameters affecting encryption and authentication may be modified. It is even possible to apply both AH and ESP, but this does not offer any significant benefits.

In the screen shot below, the 'Custom Security Method' has been selected and the more secure 3DES algorithm has been chosen as the encryption algorithm. The filter action may comprise a number of different such methods, in which case the workstation will negotiate one of the configured methods with the peer router.



Screen shot printed by permission from Microsoft Corporation.

For tunnel mode IPsec, the identities of both ends of the tunnel must be specified. Each of the two rules (outbound and inbound) corresponds to opposite ends of the tunnel. The tunnel endpoint for the inbound rule will invariably be the workstation itself, and for the outbound rule it will be the IP address of the remote peer. The tunnel setting for the outbound rule is shown below.



Screen shot printed by permission from Microsoft Corporation.

The IPsec policy is complete once an authentication method has been selected. This may be Kerberos (generally only suitable when both endpoints are Windows® devices), a pre-

configured shared secret or a certificate. If certificates are not to be used, the reader's attention is drawn to the advice in *Section 7.2.3*.

Glossary

8

8. Glossary

3DES	Triple Data Encryption Standard
ACL	Access Control List
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
ATM	Asynchronous Transfer Mode
AVP	Attribute-Value Pairs
CA	Certification Authority
CBC	Cipher Block Chaining
CHAP	Challenge Handshake Authentication Protocol
CRL	Certificate Revocation List
DES	Data Encryption Standard
DOI	Domain of Interpretation
ECB	Electronic Code Book
ESP	Encapsulating Security Payload
FR	Frame Relay
GRE	Generic Routing Encapsulation
HMAC	Hash-Keyed Message Authentication Code
HMAC- SHA	Hash-Keyed Message Authentication Code Secure Hash Algorithm
HMAC- MD5	Hash-Keyed Message Authentication Code Message Digest-5
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
IKE	Internet Key Exchange
IOS	Internetworking Operating System
IP	Internet Protocol
IPSec	Internet Protocol Security
IPX	Internet Packet Exchange
ISAKAMP	Internet Security Association Key Management Protocol
ISP	Internet Service Provider
IV	Initialisation Vector
L2F	Layer II Forwarding
L2TP	Layer II Tunnelling Protocol
LAC	L2TP Access Concentrator
LAN	Local Area Network
LNS	L2TP Network Server

MAC Digest-5	Message Authentication CodesMD5	Message
MPPE	Microsoft® Point-to-Point Encryption	
MS-CHAP	Microsoft® Challenge Handshake Authentication Protocol	
MTU	Maximum Transmission Unit	
NAT	Network Address Translation	
OSI	Open Systems Interconnection	
OSPF	Open Shortest Path First	
PFS	Perfect Forward Security	
PKI	Public Key Infrastructure	
PPP	Point-to-Point Protocol	
PPTP	Point-to-Point Tunnelling Protocol	
RA	Registration Authority	
RC4	Rivest's Cipher 4, also known as Ron's Code 4	
RIP	Routing Information Protocol	
RSA algorithm	Rivest, Shamir and Adleman algorithm	
SA	Security Association	
SADB	Security Association Database	
SHA	Secure Hash Algorithm	
SKEME	Secure Key Exchange Mechanism	
SPD	Security Policy Database	
SPI	Security Parameter Index	
TCP	Transmission Control Protocol	
TCP/IP	Transmission Control Protocol/Internet Protocol	
UDP	User Datagram Protocol	
VPN	Virtual Private Network	
WAN	Wide Area Network	
X.509	The standard for certificates set by the International Telecommunications Union branch concerned with international standardisation of telecommunications.	
X.509v3	X.509 certificates containing or capable of containing extensions.	
XOR	The Boolean operator eXclusive OR	

Tell us what you think

Technical Guides are produced and published by UKERNA for use within the JANET Community. We welcome your comments on all aspects of this document and on any other UKERNA publication. Please direct feedback to JANET Customer Service, at the address below, or e-mail us directly at: service@janet.ac.uk

UKERNA manages the networking programme on behalf of the higher and further education and research community in the United Kingdom. JANET, the United Kingdom's education and research network, is funded by the Joint Information Systems Committee (JISC).

For further information please contact:

JANET Customer Service
UKERNA
Atlas Centre, Chilton, Didcot
Oxfordshire, OX11 0QS

Tel: +44 (0) 1235 822 212
Fax: +44 (0) 1235 822 397
E-mail: service@janet.ac.uk

Copyright:

This document is copyright The JNT Association trading as UKERNA. Parts of it, as appropriate, may be freely copied and incorporated unaltered into another document unless produced for commercial gain, subject to the source being appropriately acknowledged and the copyright preserved. The reproduction of logos without permission is expressly forbidden. Permission should be sought from JANET Customer Service.

Trademarks:

JANET®, SuperJANET® and UKERNA® are registered trademarks of the Higher Education Funding Councils for England, Scotland and Wales. The JNT Association is the registered user of these trademarks.

AppleTalk is a trademark of Apple Computer, Inc., registered in the U.S. and other countries.

Cisco is a registered trademark of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Screen shots are printed by permission from Microsoft Corporation.

Disclaimer:

The information contained herein is believed to be correct at the time of issue, but no liability can be accepted for any inaccuracies.

The reader is reminded that changes may have taken place since issue, particularly in rapidly changing areas such as internet addressing, and consequently URLs and e-mail addresses should be used with caution.

The JNT Association cannot accept any responsibility for any loss or damage resulting from the use of the material contained herein.

Availability:

Further copies of this document may be obtained from JANET Customer Service at the above address.

This document is also available electronically from: http://www.ja.net/documents/tg_svpn.pdf



© The JNT Association 2003

**Joint Information
Systems Committee**