Réseaux

IPsec - Security at the Network Layer

Daniel Wasserrab <dwasserr@ens-lyon.fr>
Andreas Wundsam <andi@wundsam.net>

May 20, 2002

Contents

1 Introduction

1.1 Definition

IPsec stands short for ***IP Security***. It is a system of related protocols designed to support secure exchanges of packets at the IP procotol layer.

Protocols Most of these protcols are located in the Network Layer of the protocol stack, so any IP based communication can take advantage of it - for instance, but not limited to, the most common associated Transport Layer protocols *TCP* and *UDP*. The protocols themelves use IP for their network services, so they work seamlessly with the existing Internet routing mechanims, and no specialized handling has to be installed.

IPsec consists of the following main protocols:

- **IKE** The *Internet Key Exchange* protocol is used to negociate the cryptographic algorithms and keys necessary for communication.
- **AH** The *Authentication Header* protocol protects against alterations of the sent data on its way and provides a way to prove the sender of a message is who he claims to be. It does not, however, encrypt the data.
- **ESP** The *Encrypted Security Payload* protocol provides the same services as AH, with the additional option of encrypting the packages to provide secrecy.

Relation to IPv6 *IPsec* was originally designed as a side-product of the development of *IPv6*, the proposed successor of the standard Network Protocol in use on the Internet today. In the light of ever-increasing use of the Internet for commercial purposes and the severe security problems discovered frequently, it was considered a good idea to provide strong security from the Network Layer up, so a designated IPsec commitee was appointed, under the supervision of the reknowned *Internet Engineering Taskforce*, that has developed most of the standard protocols in widespread use today.

Design Process During the specification phase of the new security standard, it was quickly decided to design it as a generic architecture usable with the "old" *IPv4* as well as *IPv6*[1]. Due to its design process (by commitee) and the the many different environments it must be usable in (machine power, legal situation of "strong" cryptography), it has been designed a very open standard, with many optional features and little dependency on specific algorithms. The resulting complexity of the standard, however, has resulted in massive critique from security experts[2].

[1] Given the slow introduction *IPv6* appears to suffer in practical environments, this turned out to be an intelligent move.

[2] See section 5 for further information

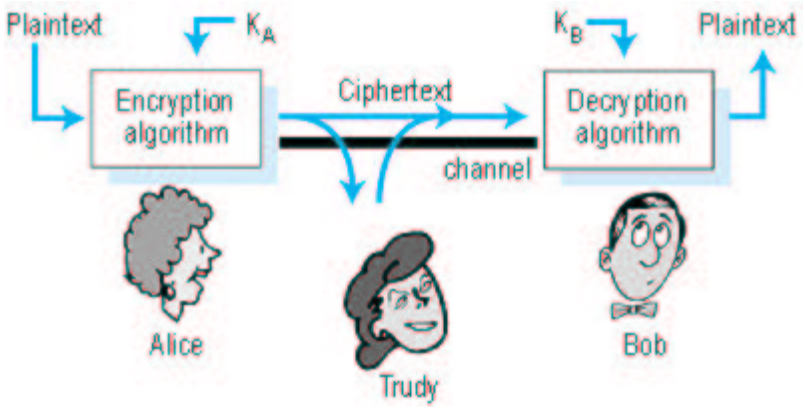

Figure 1: Cryptographic components: Alice, Bob and Trudy.

Standard The *IPsec* standard has been defined in a great number of RFCs, the most important ones including #2401 *(general architecture)*, #2402 *(the AH protcol)* and #2406 *(the ESP protocol)*. For a comprehensive listing, please refer to [3]. In theory, all components implementing this standard should be able to work together flawlessly. Regrettably, this is not always the case in practise, due to a number of reasons, which will be discussed in greater detail in section 5.

1.2 What is Security?

The notion of *security*, as used in the previous section, has remained relativly vague. In contrast to its usage in everyday-life, where most people would offer a "pretty clear" (if not necessarily identical) view of what security means to them, it has to be defined more precisely to be of use in a scientific context.

For the purpose of the secure exchange of messages between a number of partners, security is generally considered as the combination of four importants attributes, *secrecy*, *authentication*, *guarantee of integrity*, and *replay protection*. These will be explained in greater detail in the following sections.

We are using quasi-standard cryptographic notion for this examples: The two "real" partners communicating will be called **A**lice and **B**ob. The unwanted intruder trying to get access to the exchanged messages is called **T**rudy. (See also figure 1 for an overview). Since the channel used to transfer the encrypted messages is generally insecure, Trudy is assumed to have access to all of the *ciphertext* transmitted, as well as to the general algorithm used for the encryption, but not to the specific secret keys held by Alice and Bob - otherwise the whole encryption business would be useless.

1.2.1 Secrecy

Secrecy is the most obvious aspect of secure communication, and it is often falsely believed to be the only one: Even with Trudy having captured the entire encrypted message sent from Alice to Bob, she should not be able to decrypt the message to its plaintext representation without knowing the *secret key* associated with this

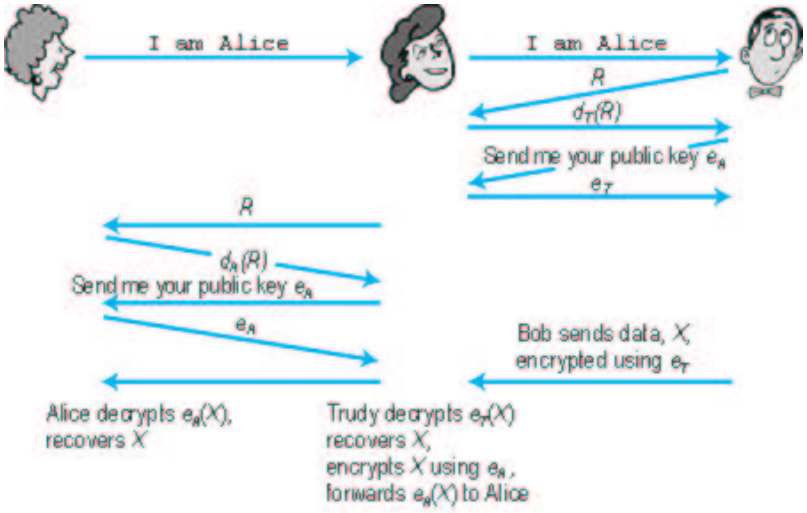

Figure 2: example of a "man-in-the-middle" attack.

communication.

While it is obvious that this an important property of a secure system, it is useless without the Autentication property. The best encryption will not help you if you cannot be sure to talk to the correct partner...

1.2.2 Authentication

Authentication is the process of proving to someone else that you are who you claim to be. This is important in two aspects: You will want to be sure your communication partner is indeed your intended recipient before transmitting any confidential data. On the other hand, you do not want your partner to be able to deny having sent an important message later - a message should be non-repudiable.

In everyday human communication, we are used to "authenticate" each other in implicit ways: We recognize a friend's face or voice, for official contexts, signatures and passports are used as authentication helpers. All of these methods are doomed to fail when communicating digitally over the Internet; new methods are required.

Note that when the two partners are communicating using a shared secret key, authentication is provided implicitly by the encrypting process - since the sender knew the shared secret key of the connection, he is obviously the intendent partner[3]. In public key systems, the situation is not as easy, however: By catching the messages and impersonating the partner, an intruder can break a security system even with the encryption working flawlessly, and this might not even be noticed by the partners. (See figure 2 for an example of a so-called "man-in-the-middle"-attack)

[3]or else, something *very, very* bad has happend and security is compromised anyway.

1.2.3 Guarantee of Integrity

The Guarantee of Integrity is a close relative of the Authentication process, and they are usually provided as a pair. Its purpose is to ensure that though the Intruder may be able to alter messages passing the insecure channel, he should not be able to do so without the partners noticing. Otherwise, security could be easily compromised. Generally, a cryptographic hash algorithm is used to guarantee this property.

1.2.4 Replay Protection

Another important attribute to secure communication is an effective replay protection: If the Intruder captures parts of the encrypted communications and resends them later, they should not be accepted as valid. While this problem would not yield an immidiate disclosure of the plaintext that was transmitted, it could certainly be used to launch *Denial Of Service*-attacks against the partners.

Replay Protection is usually provided by adding a sequence number to the data whose integrity if garanteed; messages with too "old" sequence-numbers are ignored. Thanks to the guarantee of integrity, the Intruder will not be able to modify it without the partners noticing.

Note: While *Secrecy* is generally useless without the other threee attributes, there may be cases where you might want to use *authentication*, *guarantee of integrity*, and *replay protection* without *secrecy*. For example, a published balance sheet of a company is not at all "secret", however, you will want to make sure it was published by the company's management and that nobody messed with the numbers, before buying stock....

Relation to the IPsec protocols The IPsec protocols have been designed to meet these important requirements for secure communication.

The *AH* protocol has been designed to provide protected transmisson of non-secret data and therefore provides *authentication*, *guarantee of integrity*, and *replay protection*.

ESP is able to provide all four characterists, though it can be configured to leave *authentication* and *Guarantee of Integrity* aside, in order to be used with in combination with *AH*.

1.3 IPsec vs. Application Layer Security

There already exists a handsome number of secure protocols on the Application Layer for specific jobs; examples include *ssh*, an encrypted replacement for *rsh* and *telnet*, *https* for secure connections to web servers, and *pgp*, a general encryption toolkit that can be used for encrypting e-mail messages. With this wide-spread solutions already available, is there a real nead for another instance of security in the Network Layer? The answer is a firm "yes", for a number of reasons:

Weak Applications While the cited protocols and solutions generally provide sensible security, the vast majority of applications does not: Applications like the Windows file sharing service SMB(unencrypted data transfers, no replay protection for authentication) or POP3 (completely unencrypted) prove an easy prey even for amateur crackers. In general, application programmers are not (and need not be) specialists in cryptographic matters, so protection mechanisms (particularly the ones provided by closed source products) should be taken with a grain of salt. However, by adding a strong layer of security further down in the hierarchy, weak or unproved applications can securily be used even over an insecure channel (Internet).

Practical considerations Typically, every 'secure' application uses its own method of authentication, and thus requires the protected distribution a set of keys in order to do its job. IPsec, however, requires the keys only to be transferred once for each pair of communicating hosts, which is a considerable avantage in terms of administration time.

Many Application Level security systems place considerable responsability in the hands of the end users. For instance, for the certification mechanism associated with the *https* protocol *(X.509)*, the user of the browser is often required to decide whether a certificate with a (yet) unknown root authority is acceptable or not - a decision you do not necessarily want your secretary[4] to take... The security provided by IPsec however, is - once installed - completely transperant to the user.

Different Scopes IPsec does not, however, completely erase the need for security on the Application layer, since its scope of functionality is fundamentally different from its Application Layer siblings: While those aim in general to authenticate the connection of one User to another, IPsec focusses on protecting the traffic from machine to machine. So, while an email transferred over a channel protected by IPsec will be save from attacks from "the outside", it is still open to anyone who has access to the recipient machine and sufficient permissions to read the corresponding file; in contrast, *PGP* aims to provide real end-to-end security, with the message ideally only decoded when the inended recipient is opening it.

IPsec is by design unable to provide some of the additional functionality offered by high level privacy systems. *PGP*, for example, does not only provide a strong encryption to ensure secrecy, but can also be used to create a non-repudiable signature for a document that has in some countries already been permitted as legal evidence. IPsec, providing only secure *transfer* of the message, is unable to provide such services.

[4] or worse: the vice president of marketing

2 Applications of IPsec

As has been previously stated, the IPsec standard has been designed to be as flexible as possible. We will introduce three possible and quite common applications in this section: Protecting *LANs*[5] *built on insecure media*, Supporting *Road Warriors* and, most frequently in use today, connecting *VPNs*[6].

2.1 Protecting LANs on Insecure Media

On first evidence, it might seem illogical to build a LAN of hosts exchanging confidential data on an insecure medium. However, closer inspection reveals that this is far from uncommon:

WireLess LAN Wireless LANs based on the IEEE standard *802.11b* are in widespread use today. Their big advantage lies in their easy setup and their close ressemblence to current wire-based networking technologies: A network based on 802.11b behaves largely identical to a normal ethernet network. The associated flexibility particularly for notebook users has caused many companies to offer their employees wireless access to their internal networks, while relying on the standard's proposed encryption and access control mechanism called *WEP* for security.

However, a number of studies have revealed that this encryption standard is in fact fatally flawed[7]. In order not to abandon either the flexibility offered by wireless work or the security necessary for the exchange of confidential data, an additional layer of security was necessary.

IPsec considerations IPsec proves an ideal solution to this problem: The Wireless LAN interface on all participating hosts will be configured to encrypt all outgoing traffic and accept incomming traffic only it is has been authenticated as valid. Once the mechanisms are installed, the users can transparently continue to work in their internal networks, with the additional security provided by IPsec. The sole disadvantage of this solution is the performance aspect: Due to the computational complexity of encrypting the all data transfered in the LAN, users of slower systems might notice a considerable drop of network performance.

2.2 Connecting Virtual Private Networks

Until recently, companies used to interconnect only the networks of their most important offices permanently, using loaned permanent connections. Branch offices were either not connected at all or used modem based dialin schemata to exchange the most important informations. However, such an approach is not practical any more

[5] Local Area Network

[6] Virtual Private Networks

[7] The weakness was originally published in [11], an academic paper. However, its theoretical results have been proven to be exploitable in practise and have been already been implemented, for example in the software *AirSnort*. See [12] for an analysis of the situation.

Figure 3: Schematic overview of a VPN setup

in today's business world: The stiff competition on the global markets and innovative production schemes like *Just In Time* require that that all information and knowledge in a company be available in near-realtime.

Principles Long distance permanent telephone links remain immensely expensive, while powerful local internet uplinks are available at moderate costs almost everywhere in the world. This sparked the idea of building *Virtual Private Networks*: Encrypted "tunnel" connections are created through the Internet that seamlessly transfer data from the branch office to the headquarters. Dedicated *security gateways*[8] act as centralized gateways to the internet and ensure any traffic to the internal partner subnets is encrypted before being sent out to the Internet. Figure 3 provides an, albeit slightly simplified, overview of the setup.

IPsec Considerations Despite the considerable number of proprietary solutions available for this job, IPsec, being an open standard of reliable security properties, has become the de-facto standard for building VPNs. Its advantages lie in its power to protect any IP based traffic regardless of application or transport protocol, and the possibility to interconnect with gateways of other manufacturers.

2.3 Supporting Road Warriors

Situation A road warrior is a mobile employee who uses his notebook to link up to the company network from his changing places of work. Traditionally, modem dial-ups to specialized routers in the internal company network were used for this job, but these do not meet the speed requirements posed by many current applications. On the other hand, fast and cheap internet access is available in many hotels, airports or congress centers.

Principles The road warrior setup can be regarded as a special case of the VPN scenario, where one connected network consists just of the road warrior's notebook. In all other aspects, the configuration is similar to the VPN case. One glitch that

[8] Often specialized hardware, due to its greater realiability and encryption performance

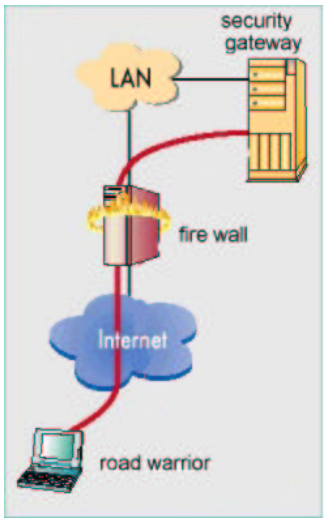

Figure 4: Schematic overview of a Roadwarrior setup

has to be handled by the security gateway in the headquarter is that the IP address of the connecting roadwarrior is *a priori* unknown. See figure 4 for an overview of a road warrior setup.

IPsec Considerations While there are specialized solutions available for this problem[9], several of these have proven vulnerable to different security attacks. IPsec, more secure and open, can be used just as well: An IPsec driver in the road warrior's notebook creates a secure tunnel to a security gateway in the internal company network, and all traffic in the direction of this network is encrypted. Many modern operating systems like Windows 2000 and Linux already contain the necessary drivers, and current computers are doubtlessly fast enough to handle the encryption of the moderate traffic passing these links.

[9]For example, the *Point-To-Point-Tunneling-Protocol* (PPTP) from Microsoft

3 The Different Protocols of IPsec

3.1 Internet Key Exchange (IKE)

3.1.1 IKE, Laying the Ground for AH and ESP

The IKE protocol is used to negotiate certain parameters which are essential for setting up the AH and the ESP protocol: These parameters include the specific algorithms to be used, shared keys, and other data like connection lifetimes. To do so, the two entities wishing to communicate create a Security Association (SA), in which all parameters for AH and ESP are noted down. IKE works by exchanging packets on UDP port 500 between the two gateways or hosts. Details about the SA can be found in the section "The Security Association".

The IKE negotiation mainly consists of two phases:

- *Phase one*: First the two entities set up a two-way SA which is used for the real negotiation in phase two. The SAs used for this are so called ISAKMP SAs. Details about ISAKMP SAs can be found in RFC 2408.
  With one such SA multiple negotiations on different tunnels can be carried out.
- *Phase two*: By using the SA defined in phase one, the two gateways (hosts) can now start to negotiate the required SAs (so-called IPsec SAs, but further only called SAs). As SAs are unidirectional (a different key is used in each direction), for using them in communication a pair of them must be negotiated. Between two gateways (hosts) there can (and sometimes must) be defined several pairs of them.

After having finished these two phases, the communicators have achieved SAs which they can now use to establish the AH and/or ESP protocol.

3.1.2 The Diffie-Hellman Key Exchange

To find the key needed to establish a secure connection, symmetric key algorithms are useless, as you have to agree on the key before using it but since a secure connection is not available, this cannot be done safely. But this basic problem of encryption has been solved by the invention of so-called asymmetric key algorithms, where a secure key can be negotiated even with an intruder listening to the whole communication.

The algorithm usually used in the IPsec protocol is the Diffie-Hellman key exchange, which is based on a discrete logarithm problem in number theory, for whose solution there does not seem to exist a reasonably fast (that is, polynomial) deterministic algorithm. This algorithm uses prime numbers and modulo calculation, but is nevertheless not too hard to understand. (see also Fig. 5):

Alice has to find two large prime numbers, n and g, where $(n-1)/2$ is also a prime and certain conditions hold for g. These numbers can be used publically, they need not be hidden. Next, Alice must choose a large (e.g. 512-bit) number x and keep it secret. Bob chooses a large secret number y as well. Then Alice transmits the

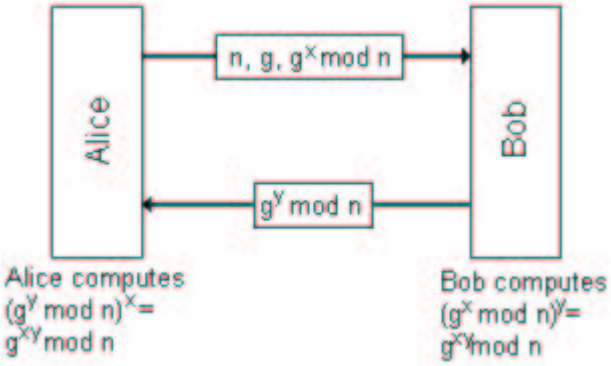

Figure 5: The Diffie-Hellman key exchange.

numbers n, g and $g^x \ mod \ n$ to Bob. Bob calculates the key $a = (g^x \ mod \ n)^y$ and retransmits $g^y \ mod \ n$. Alice now calculates $(g^y \ mod \ n)^x$ and therefore receives the key $a = g^{xy} \ mod \ n$ as well, based on a number theory theorem. The trick of this calculations is that any intruder can listen to this transmissions and therefore receive g, n, $g^x \ mod \ n$ and $g^y \ mod \ n$, and nevertheless he will not be able to calculate the key a, because solving the equations $g^x \ mod \ n$ or $g^y \ mod \ n$ cannot be done easily, even if g, n and the results are known. So the sender and the receiver have agreed on the secret key in spite of possibly being wiretapped.

3.1.3 The Security Association (SA)

The advantage of using SAs is the resulting separation between key management and the security mechanisms. The mechanism which brings this two principles together is the Security Parameter Index (SPI) by identifying the SA by a given number resulting from the negotiated key. To assign a packet to a particular SA, a triple of three fields must be checked: (destination IP address, SPI, security protocol). The security protocol defines whether AH or ESP is used. Since the IP address of the receiver is part of the triple, this is guaranteed to an unique value.

Next, we will list some of the parameters of the SA:

- The Security Parameter Index (SPI) to identificate the SA in the AH and/or ESP protocol
- AH informations like key, authentication algorithm, key lifetime
- ESP informations like key, authentication and encryption algorithm, key lifetime, initial data
- Lifetime of the SA
- IPsec protocol mode (Transport or Tunnel)
- if used: tunnel path MTU

Security Parameter	Example value
SPI (Security Parameter Index)	2916
AH Algorithm	MD5
AH Algorithm Mode	Keyed
AH Transform	RFC 1828
AH Key(s)	a 128 bit MD5 key
AH Mode	Entire Datagram
ESP Algorithm	DES
ESP Algorithm Mode	CBC
ESP Transform	RFC 1829
ESP Key(s)	a 56 bit DES key
ESP Mode	Transport
ESP Synch/Init Vector Size	64
Lifetime	an absolute time in Unix Time format
IPSO/CIPSO Sensitivity Label	Nuclear/Classified

Figure 6: Example of a Security Association Database.

The administrative entity in which all these values are stored is called Security Association Database (SAD). In Fig. 6 you can see an example of a SAD, filled with typical values.

There are two possibilites how the SA can be defined: the SA can be host-oriented, all users on the same host use the same session key, or it can be user-oriented, so that every user has to negotiate a key himself, resulting in having a session key for every user.

Another administrative entity which controls the package handling is the *Security Police Database* (SPD). The SPD is used for outgoing packets - it defines which parameters of the SAD should be used. If there are no suitable SADs defined, SPD is able to create new ones. Outgoing packets use the SPD to get the encoding parameters of the SA whereas incoming packets use the (destination IP address, SPI, security protocol) triple to get to the SA.

The SPD works similarily to a packet filter in choosing the appropriate SA by examining different “selectors”, such as source or destination address, the level of the security sensitivity, etc.

Types of entries a SPD must contain:

- Pointer to the active SAs
- Selector fields to choose the appropriate SA

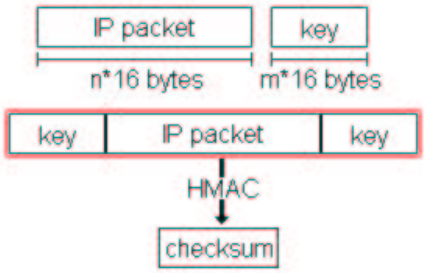

Figure 7: Application of the HMAC algorithm.

3.2 Authentication Header (AH)

3.2.1 The Goal of AH

When we receive an IP package, we generally assume that this package was not read or changed by any other user (except the sender). But with the means of IPv4, we can not be sure: An intruder who wants to read our traffic could catch the packets destined for the receiver, read or change the contents, and fill in the original sender's source address, recalculate the checksum and forward it to the receiver (performing a Man-in-the-middle-Attack). The receiver has no opportunity to realize that something happened to the package because the checksum does not show any errors and the address of the sender is correct as well.

The Authentication Header protocol was introduced to make attacks like this impossible. Furthermore, using this protocol we can also make sure our communication partner is in fact who he claims to be. Contrary to the ESP, however, it does not encrypt the payload, so once an intruder has succeeded in getting access to packets, he can easily extract the contents. Both protocols differ in the scope of their protection as well, the AH authenticates the whole package while ESP only encrypts certain fields (more information in section 3.3).

3.2.2 How is the AH used?

To achieve authentication, AH uses cryptographic *Message Authentication Codes*, so-called MACs. A MAC is an algorithm which takes the message and a cryptographic key to calculate a Message Digest or Fingerprint. The difference between MACs and hash functions is the use of this cryptographic key, which is not used by hash functions. The most frequently used MAC is the so called HMAC, which can be used in combination with every cryptographic Hashing-function, for instance MD5 or SHA-1. To be able to use the MAC, both sender and receiver must have agreed on a key, which they do as shown in the previous subsection Internet Key Exchange.

To authenticate a message, we take the IP-package including all used IP-headers, set all fields that might change on its way (e.g. the hop limit) to zero and pad it with zeros until the length reaches a multiple of 16 bytes. Then pad we the key to a multiple of 16 bytes, as well. Now we compute the cryptographic checksum on the concatenation of the key, the package and again the key (all with lengths multiples

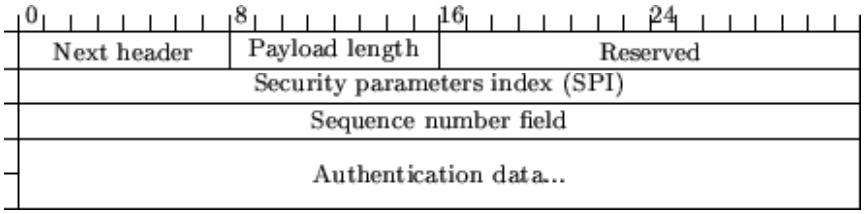

Figure 8: The Authentication Header.

to 16 bytes). The received checksum is then inserted in the header.

The receiver does the same, he takes the package, sets changeable fields to zero, and pads it as well as the key, if necessary. Then he computes the checksum of key-package-key and verifies it against the checksum given in the header. If both checksums match, he can be sure that the package has been sent by the sender, with whom he had agreed on the key, and that the package was not changed on its way.

3.2.3 The AH in detail

The AH consists of five fields of fixed length and an Authentication Data field of variable length. The Header is shown in Fig. 8. Below we will discuss the fields in detail:

- *Next Header*: This 8 bit field identifies the next IP-header or defines the transport protocol handler to pass the packet to.
- *Payload Length*: This field represents the length of the AH in 32 bit words minus 2. This diminution is necessary because the specification for IPv6 Extension Header says that the length of Extension Headers should be computed subtracting a 64 word at first. As the Payload Field returns the length in 32 bit words, we have to subtract 2 from the real header length.
- *Reserved*: A 16 bit field is reserved for future use. Until then it must be set to zero.
- *Security Parameter Index (SPI)*: This 32 bit integer number is used in combination with the destination address to identify the Security Association (SA) of the data transfer. The numbers 1 – 255 are reserved by the *Internet Assigned Number Authority* (IANA) for future use, while 0 is used for special implementations. So the available numbers are 256 – $(2^{32} - 1)$.
- *Sequence Number*: The 32 bit unsigned integer in this field is used as a counter, which is initialized to 0 on the source and the destination side as they begin to establish the SA. Every time a package with this particular SA number is sent, the counter is incremented by one. So the sequence number prevents the replay of data packets, e.g. by an intruder who intercepts the transfer and retransmits the packets. The AH specification states that the sender has to transmit the Sequence Number, whereas the receiver can choose to ignore the Sequence

Number field and thus effectively switch off the anti-replay functionality. However, if anti-replay is used, a sliding windows on the receiver side is used to identify duplicated packets. The sliding window implementation can vary, but in general the size is set to at least 32 bits, where the number on the right outside margin defines the highest verified Sequence Number in the given SA. Packets with counters less than the number on the left side margin should be rejected (a package with this number has already arrived and had been found correct), whereas packets with Sequence Numbers "inside" the window should be compared with a list of received data packets within the window. If the received packets are new or their Sequence Number is bigger than the right margin of the window (and less than 2^{32}), the receiving host computes the authentication data. Sequence Numbers within an SA must not be reused, so if we want to send more than 2^{32} packets, we have to negotiate a new SA.

- *Authentication Data*: This field of variable length contains the authentication data, also referred to as Integrity Check Value (ICV). Using the IPv4 protocol, this field has to be a multiple of 32, whereas by using IPv6, the length must be a multiple of 64; if this requirement is not fulfilled, the ICV has to be padded with zeros to achieve the needed length. The algorithm used to compute the ICV is specified in the SA (e.g. HMAC-MD5 or HMAC-SHA-1, existing in all IPsec implementations).

3.3 Encrypted Security Payload (ESP)

3.3.1 The Goal of ESP

ESP, contrary to AH, provides authentication *and* encryption. This is necessary because without any authentication, the encryption is vulnerable to active attacks which may allow an enemy to break the encryption, rendering it essentially useless. When using encrypted packets, an intruder who succeeds in gaining access to the transmissions has in fact not achieved the possibility to read the contents, which also makes it impossible to transmit modified packets. The security ESP provides is strongly dependent on the encryption algorithm that is employed. Unlike the AH which offers authentication for the whole package, including the different headers, ESP encrypts only the fields following the ESP header respectively authenticates only the ESP header and the following fields (both times excluding the Authentication Data field), so the preceding headers are exempted.

3.3.2 How is the ESP used?

When using ESP, you can chose from numerous possible options. Either MD5, SHA-1 or no authentication at all can be used as authentication algorithm, mirroring the possibilities offered by AH for the same purpose. To encrypt the data, two options are available by default, the DES algorithm or null encryption. However, several implementations of IPsec have chosen to drop DES in favour of 3DES, argueing that DES does not provide enough security against attacks performed with advanced skills and hardware equipment.

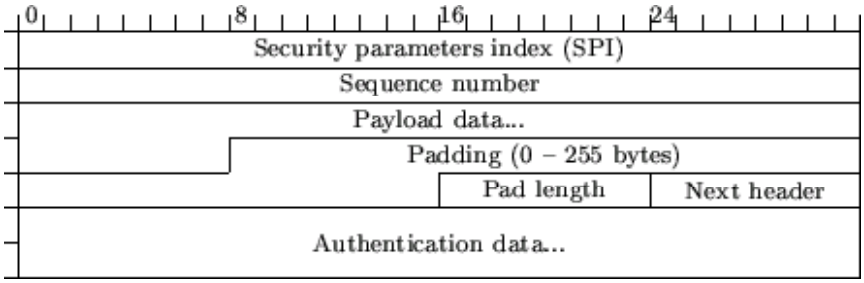

Figure 9: The ESP-Header and the ESP-Trailer.

To encrypt a message, the plaintext data is given to the algorithm defined in the SA, the returned ciphertext is inserted in the Payload Data field. The receiver decrypting the message simply extracts the Payload Data field and uses the decryption algorithm associated to the chosen encryption algorithm. If the matching encryption and decryption algorithms were used, he gains the original data.

3.3.3 The ESP-Header and the ESP-Trailer in detail

Contrary to the AH, the ESP protocol embraces the payload data by two elements, the ESP header and the ESP trailer, shown in Fig. 9. The header consists of two fields with fixed length, the trailer of two fields of fixed length and one of variable length. A description which field serves for what is given in the following:

ESP-Header:

- *Security Parameter Index (SPI)*: This field matches with the field of the same name in the AH. Details can be found in his description above.
- *Sequence Number*: The same holds for this field, which corresponds to the Sequence Number field in the AH.

ESP-Trailer:

- *Padding*: The Padding field can be used in different ways, so its length can vary between 0 and 255 bytes. Mostly it "pads" other data, e.g. if the encryption algorithm needs an input of a precise length, the padding information is added to the package data to achieve the desired length. Using the Padding field could also be necessary simply to adjust the fields in the header's 32 bit pattern. Besides, we could also use the Padding field to cloud the real length of our data.
- *Pad Length*: In this 8 bit field you can find the exact number of the "filling" bytes, between 0 and 255.
- *Next Header*: Similar to the Next Header field in the AH, the *next header* field defines which type of Header will follow (IP or Transport Layer)

- *Authentication Data*: Being an optional field, it can be used to transmit a checksum, an ICV, using the algorithm stated in the SA. That way, the receiver can calculate the checksum when the package arrives and, if the calculated and the transmitted checksum differ, discard the package.

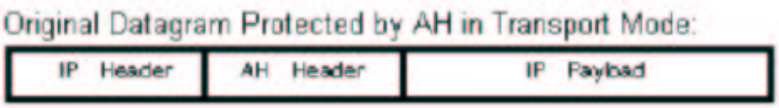

Figure 10: AH in Transport Mode.

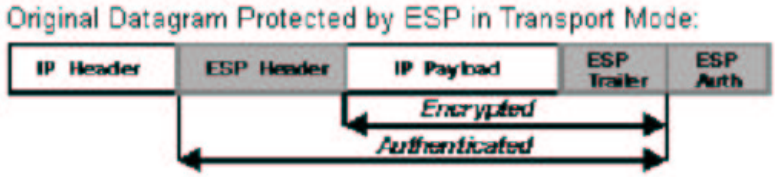

Figure 11: ESP in Transport Mode.

4 Transport vs. Tunnel Mode

The IPsec protocol supports two different modes for AH or ESP which differ in the percentage of the packet they authorize or encrypt. In the following two sections we will take a closer look at these two modes.

4.1 Transport Mode

Transport mode has been designed to support direct host-to-host connections like they are necessary in the *Insecure-LAN* scenario described in section 2.1.

In this mode, the AH and/or ESP headers are placed behind the IP Header and also behind certain extension headers such as Hop-by-Hop, Routing or Fragmentation Header (in special cases the Destination header can be placed there, as well). Any other extension header is placed behind the IPsec headers. If both AH and ESP are used, the AH is placed in front of the ESP header. (see Fig. 10 and 11)
When AH is used, the whole packet, including the IP Header, is authorized.

However, Transport Mode is doomed to fail in several cases: Imagine that the sender is using a private IP address (10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255 or 192.168.0.0 – 192.168.255.255), then his packets cannot be delivered over the Internet as border routers do not (and should not!) forward packets with such addresses. This problem is normally solved by using a Network Address Translation (NAT)-Gateway, which replaces the private IP address in the Source Adress field by a public IP address assigned to iself, recomputes the checksum and forwards the packet. As the NAT-Gateway changes the value of one field authenticated by the used protocol, the receiver will not accept the packet as valid, as his computation of the checksum produces another result than the checksum given in the packet, caused by different source addresses.

Another disadvantage concerns the the amount of traffic hidden from pontential wiretapers: Since source and destination address need to be visible on the whole way, they can be used to account the traffic sent from and to specific hosts - an information

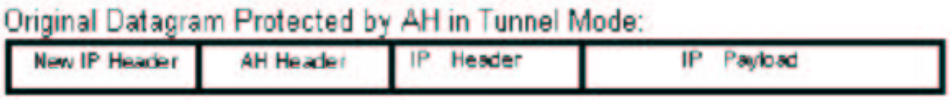

Figure 12: AH in Tunnel Mode.

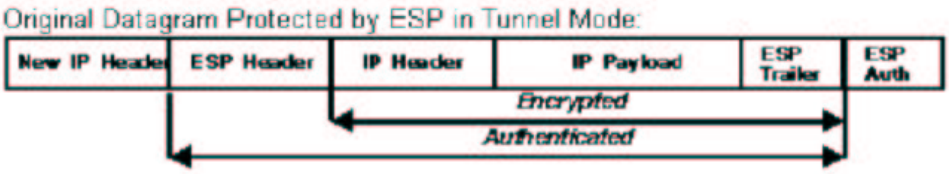

Figure 13: ESP in Tunnel Mode.

one does not necessarily want to reveal in a security-critical environment.

Since in LANs the application of the authentication and encryption algorithms are left to the hosts in Transport Mode, the performance is strongly dependant on the machines used. If the machines used in the networks are slow, the performance in the whole system will suffer from this.

A big advantage of the Transport Mode is the small size of the packets, so that only a small overhead of the data is necessary and the amount of redundant information is small compared to the transmitted data.

4.2 Tunnel Mode

Tunnel Mode has been designed to suite the *VPN* scenario described in section 2.2; it is optimized for connecting two private LANs by the help of dedicated security gateways.

In this mode, the AH and/or the ESP header are added in front of the former IP Header and a new IP Header is placed in front of this whole new packet. Examples are shown in Fig. 12 and 13.

The Tunnel Mode hides the former IP packet entirely by placing it in the payload of a newly created packet. This mode can also prove very useful when ESP is used; as stated in the ESP section, the ESP only encrypts the fields behind the ESP header, letting several header fields unsecured. When Tunnel Mode is used, the ESP encrypts not only the payload, but all headers, as the entire packet is placed behind the ESP header. The new IP Header remains unencrypted, of course, as he must be readable for the routers.

Information hiding is also a major issue of secure communications and can, in a certain amount, be achieved by using Tunnel Mode. As the original source and destination addresses are included in the payload, they cannot be read by routers, so only few informations about the involved hosts are revealed.

The Tunnel Mode is normally used to create secure networks passing a public WAN (e.g. Internet). By encapsulating the informations within a newly created packet, a

secure "tunnel" between the two gateways is built, the packets are "tunneling" the WAN and remain secured by the IPsec protocols. So private networks can communicate securely, although their connection is passing the Internet.

An important drawback of this method is that there is no secure protocol between the host and the gateway. So the use of the whole IPsec protocol could be useless because the intruder is placed between host and gateway, and this is the area where we don't use authentication or encryption.

Also the amount of transportation data compared to the data to be transmitted is much greater than in Transport Mode, as new headers for each packet have to be introduced, resulting in a lower performance.Problem diagnosis is also made difficult by the Tunnel Mode information hiding, so when problems occur underways, they are impossible to diagnose from "inside" the tunnel.

5 Limitations and Problems of IPsec Today

Given the presented flexibility and power of the IPsec standard, one might be tempted to assume that it is the ideal solution for today's growing requirements on data transmission security. However, observing the vast amount of experience reports available on the Web, it is quite the opposite image that prevails: IPsec has received harsh critique both from reknowned security experts like Niels Fergusen and Bruce Schneier, who describe IPsec as a "great disappointment"[10], and by frustated users overwhelmed by the standard's complexity and tormented by buggy and incompatible implementations. What are the primary reasons for these problems?

5.1 Standard

The standard itself is discribed by Fergusen and Schneier as a "typical result of a commitee specification process": It is quite vague, and contains many options and facultative features which turn it into a rather complicated beast. This, however, violates the principle that security standards should be as simple as possible since "Security's worst enemy is complexity" [10]. A complex standard is difficult to implement correctly and bug-free, which is of course essential for a security protocol system.

Some protocols (particularly IKE) contain moderate cryptographic problems that make the standard weaker than it would have to be, and might be used in combination with minor implementation faults to break the security of the system.

Another heavily critized point is largely due to the heterogenous nature of the world-wide legislation regarding strong cryptography: DES, an algorithm with known cryptographic weaknesses and an insufficent key length of 56 bit, has been chosen as the standard encryption cipher. Its only advantage is the fact that its weak security is permitted even by those countries who otherwise fiercely restrict the use of strong cryptography between their borders[11]. However, the usage of improved standards like 3DES or AES is encouraged where it is legally permitted.

5.2 Design Aspects

Some other limitations of IPsec are by design: It is not - and does not try to be - an end-to-end all-in-one security solution for every problem. The very principle of encryption in the Network Layer accounts for some limitations.

IPsec cannot provide all the services provided by higher-level security solutions; specifically, it only authenticates machines and never users, so it does not provide an non-repudiable signature of any use for legal aspects. Another problem concerns multi-user systems like current versions of Windows NT or UNIX: While IPsec greatly reduces the danger of the data being compromised during network transfer, it is of no use for protecting the data once it has arrived on the destination systems, so an

[10] see [10] for details

[11] Notably, Iraq, China and until recent days France

improper set of access priviliges or security holes in the host operating system can pose a serious security threat.

Another aspect concerns the the efficency of the system: An IPsec system takes its encryptions decisions in Layer 3, that is on router level, usually based on the destination address of the host the data is being sent to. This can lead to unnecessary computing overhead, since there might be a lot of traffic going to that network that is not really confidential - for instance, normal webserver requests or DNS requests. On the other hand, packets from secure application protocols like *SSH* will be encrypted one more time, causing double cryptographic overhead. However, these problems are not generally regarded as grave: Given the processing power of modern machines, it is easy to pay the efficency price of IPsec for its additional security. Besides, unnecessary encryption of non-confidential data can be minimized by careful design of the destination subnets.

5.3 Implementations

Largely due to the complex matter and the vague standard, many implementations of IPsec cannot be called mature to this day. A nonexaustive lists of limitations and bugs we encountered during practical testing follows:

Linux FreeS/WAN

- **Incompleteness** *FreeS/WAN* does not yet implement the full scope of the IPsec standard. While some omissions are due to design decisions and no problem for practical use (as, for instance, the choice not to support "simple" DES), others restrict heavily its practical usability in combination with other implementations: The standard distribution of *FreeS/WAN* does not yet support authentification using *X.509* ceritifates, restricting the communication with common Windows implementations to Shared-Secret mode which is inpractical for larger applications[12].
- **Weak integration in the Kernel** *FreeS/WAN* is integrated quite roughly into the linux kernel. It uses its own routing table and requires associated network configuration (for example the standard gateway) to be done a second time in its configuration files. Several common operation modes require awkward fiddling with low-level kernel configuration parameters to work flawlessly.

Windows 2000/XP

- **Complex configuration** Somewhat contrary to Windows' repuation, the configuration of the necessary *security associations* and *rule chains* proved incredibly complex. The help system and the informations located in the MSDN

[12] There exists a patch for integrating *X.509*. However, it is still tagged "beta", which makes it unusable for security-critical environments.

support database did not prove of much use: While positivly verbose, they describe almost exclusivly the setup of purely Windows-based configurations and contain sparse information about setups for heterogenous environments [13].

- **Compatibility problems** During the setup of a VPN between a *CISCO* router and a *Windows NT*-based security gateway, grave compatibility problems were observed, causing a regular breakdown of the encrypted tunnel during key renegotiation. These problems could not even be resolved with the help of *CISCO* and Microsoft supporting personnel. Finally, the Windows-based security gateway was replaced with a second dedicated *CISCO* router.

6 Conclusion

Despite its enormous potential for amending some of the worst security problems that torture the Internet today, *IPsec* remains almost exclusively used for connecting VPNs by dedicated security gateways. In this scenario, the setup complexity is relativly small since IPsec needs only to be installed on the gateways themselves; compatibility issues can be minimized by choosing identical or related hardware.

Many other projects, for example to provide Road Warrior connectivity, have either been suspendend, or have reverted to using propriatary (but less complex and better implemented) technologies.

What does the future hold for IPsec? Hard to tell, especially since strong cryptography in general has faced considerable oposition in the the last year, following the dramatic events of the fall of 2001. This has caused some some vendors like *Network Associates* to drop cryptography support entirely or suspend further development.

However, with the broad introduction of *IPv6* to be expected any century now and the ongoing evolution of many free implementations, IPsec might one day just live up to its promise:

> To Make the Net a Safer Place.

[13] This problem has been ameliorated by several freeware tools that generate the required *rule chains* and *associations* from a simple configuration file

References

[1] Andrew S. Tanenbaum: *Computer Networks*, 3rd ed., Prentice Hall (1996), ISBN: 0-13-349945-6

[2] James F. Kurose, Keith W. Ross: *Computer Networking: A Top-Down-Approach Featuring the Internet*, Addison-Wesley (2001), ISBN: 0-201-47711-4

[3] Barbara Fraser, Theodore Ts'o: *IP security protocol charter*, IETF
http://www.ietf.org/html.charters/ipsec-charter.html

[4] FreeS/WAN Project Homepage, http://www.freeswan.org

[5] NetBSD Homepage, http://www.netbsd.org

[6] OpenBSD Homepage, http://www.openbsd.org

[7] funkschau, funkschau handel and funkschau mobilcom Homepage
http://www.funkschau.de

[8] RFC 1825-1829,1851-1852, 2401-2411, http://www.ietf.org/rfc

[9] Jürgen Schmidt, Peter Siering: *Moderner Tunnelbau - Der Weg zum eigenen Virtual Private Network*, c't Magazin 18/2001, ps. 182ff

[10] Niels Ferguson, Bruce Schneier: *A Cryptographic Evaluation of IPsec*
http://www.counterpane.com/ipsec.html

[11] Scott Fluhrer, Itsik Mantin, Adi Shamir: *Weaknesses in the Key Scheduling Algorithm of RC4*
http://downloads.weblogger.com/gems/80211b/rc4weakness.pdf

[12] 802.11b Networking News: *Weak Defense*
http://80211b.weblogger.com/weak.defense.html