Published in the SWITCH Journal nr. 1/96, SWITCH Zürich, May 1996.

Proposed Security Mechanisms in the "New" Internet

Germano Caronni
Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich

Dr. Hannes P. Lubich
SWITCH Computer Emergency Response Team (SWITCH-CERT)

Introduction

Currently, the Internet Protocol is undergoing a major re-design under the supervision of the Internet Engineering Task Force. Although one of the main goals for this re-design was the enlargement of the address space (from 32 to 128 bits) and the possibility to differentiate between different types of service classes, the integration of security service elements into the existing version 4 as well as into the new version 6 of IP has become one of the most important design issues. Currently, a multitude of security solutions (such as "ad hoc" packet filtering and firewall methods) is being deployed in the Internet, mainly because of the increasing commercialisation of the net. However, these solutions are mostly isolated, and no interoperability is foreseen for the conceptually different approaches. On the other hand, offering encryption and authentication on the network layer has the great advantage that security may be engaged transparently, and thus, existing applications need not be security-aware. The IETF forum is pursuing the goal to provide secure networking infrastructure in the working group 'IPSEC' that has been founded in late '94.

A few months ago, IPSEC has turned out proposed standards as to what the security framework in IP v4 and IP v6 will look like (RFC 1825) and how authentication and encryption are to be done (RFC 1826 – 1829). There are first implementations offering this service, both under IP v4 and IP v6. Currently IPSEC is pursuing the standardisation of one or more key management protocols, the most prominent approaches being Photuris and SKIP (Simple Key Management in the Internet Protocol). This document will try to highlighten some principal issues of IP v6 security, and then elaborate on SKIP.

The Proposed Security Mechanisms

At the moment, two different security mechanisms are being developed for IP v6 as extensions to the base protocol.

Authentication and integrity of an IP packet (RFC 1826) through a keyed message digest as part of an *Authentication Header.* According to RFC 1828, this service element is implemented through an initial exchange of a secret key using a method where such a "secret" can be passed along between sender and receiver even though an insecure transmission channel is used (Diffie-Hellman key exchange). Once this secret key is known to both parties, the sender of a packet concatenates the text to be signed and the secret key and computes a MD5 message hash function. The receiver of the packet may then apply the same hash function to the concatenated text and secret key, and verify authentication and integrity of the IP packet by comparing the result with the sender's signature.

Confidentiality and integrity through the *Encapsulating Security Payload* (ESP, RFC 1827) mechanism, which allows a "transport mode", in which only the transport protocol data unit is encrypted (see figure 1), as well as a "tunnel mode", in which the original IP packet is encrypted and encapsulated in another IP packet (see figure 2).

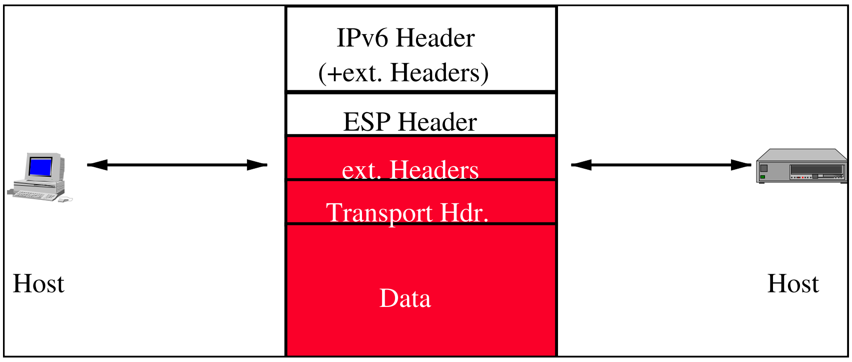

Figure 1:Transport mode ESP

While the first method is sufficient to secure the content of an IP packet on an end-to-end basis, the second method may serve two different, but related purposes:

1)Since the original IP packet is encrypted and encapsulated in another IP packet , the tunnel mode could be used to hide the originator and recipient of an IP packet, since the "outer", readable IP packet will only contain the IP addresses of the two routers between which the IP packet is currently in transit. This however will require encryption capabilities from the routers at each end of the tunnel.

2)Similarly, such a tunnel could be used between an end system and a router, e.g. to securely connect a mobile computer to a "home" network through an IP v6 capable firewall and router. In this scenario, the mobile end system would serve as one end of the "IP in IP" tunnel, encapsulating the original IP packet in an IP packet addressed to the firewall system. The firewall would act as the other end of the tunnel, decrypt the packet, determine the "real" recipient and forward the "inner" packet to the end system in the protected network.

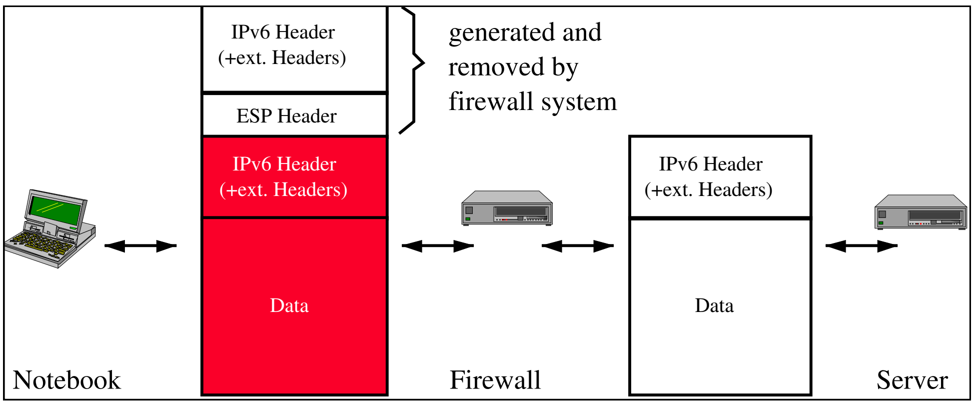

Figure 2:Tunnel mode ESP

The two security mechanisms described above are currently being refined and prototyped. It is too early to say when these mechanisms will be broadly available, but one can expect first implementations to be available in the beginning of 1996. It will then depend on the router and workstation operating system vendors, how fast secure IP v6 will be commonly available, and it will depend on the "accompanying measures" for IP v6 transition (such as public key management and distribution), how fast the user communities will migrate to IP v6.

4.Key Management Issues

Bulk data encryption and authentication is normally done using symmetric algorithms for efficiency reasons. This implies that sender and receiver own the same key, or – more generally – the same shared secret. This, however implies that non-repudiation, e.g. being able to prove the receipt of data sent by the other party can not be achieved this way. It is feasible to use the already defined IP security mechanisms with asymmetric algorithms to achieve this functionality, but only required under certain circumstances, and computationally expensive. This is the prime reason why efforts concentrate on mechanisms that offer an acceptable performance/computation ratio. In this context, it is the purpose of the key management to create the shared secret for the communicating peers. The simplest form is manual keying, where a human operator provides the secret on all communication endpoints. It is obvious that this solution does not scale well, and that it is quite clumsy to employ. Additionally, it is not favourable to use the shared secret (which should be valid for a longer period of time, such as a month or a few years) directly for bulk data encryption. Derived traffic encryption keys are introduced instead, which can then be changed periodically by the key management.

4.1.Requirements for IP Layer Key Management

The primary goal of security on the IP layer is to achieve secure communication between end systems, or alternatively between complete networks via firewalls. The employed system should be structured such that the existing (and appreciated) properties of IP like re-routing, load balancing, crash recovery are maintained, and only a minimal overhead is being introduced in order to secure the network traffic. At the same time the system should be flexible and extensible enough to allow for future extensions like per-user or per-port keying, support of smartcards, secure multicasting, migration to IP v6, addition of new algorithms, etc.

4.2.Key Management Protocols: Photuris and SKIP

Currently, two different approaches in key management are being pursued by the IETF working group on IP security (IPSEC). Although they offer a common subset of functionality, the fundamental approach of solving the key management problem is vastly different. In the *Photuris* approach, the communicating parties hold the capability to dynamically sign randomly generated keys. During a connection setup phase, a shared secret is negotiated and authenticated by signing it with the private part of the authentication key, and half a dozen messages go back and forth until the actual communication can start. Communicating partners have to know the initiator's public part of the authentication key, and have to keep context throughout the lifetime of the security association that has been established through the negotiation of the shared secret. This has the advantage, that – if the shared secret is dropped on both sides (which happens when the security association is dissolved, e.g. every few hours or days) – nobody will have easy access to the transmitted (and possibly intercepted) data, as the keys do not exist anymore, but it fixes one association to one site. If the context is lost somehow, or part of the traffic should go to a different destination, a new association needs to be established with that site.

SKIP (Simple Key Management in the Internet Protocol) on the other hand relies on the fact that a public certificate (signed e.g. by a certificate authority, or by oneself using the PGP infrastructure), is somehow provided to the communicating peer. Just by combining the public certificate with the secret part of one's own certificate, both parties can calculate an *implicit shared secret*. No context is needed during the lifetime of such a security association (other than for efficiency reasons), thus switching from one host to another (as long as both hold the same secret part of the certificate) can be done easily. The drawback is, that possession of the secret part of ones certificate reveals all past traffic from or to that side. This drawback can be mitigated by either changing the certificate frequently, or employ an (optional and planned) scheme to generate shared secrets 'on the fly'.

At the moment, versions of Photuris are being implemented for various operating systems including NetBSD, KA9Q and Linux. The draft is (except for a few issues under discussion) almost complete, although the process is slightly slowed down by the higher complexity of the Photuris approach, and more variable parameters.

4.3.Inner Workings of SKIP

Being a key management protocol, SKIP has the duty to provide a shared secret for communicating peer entities. As SKIP does not try to re-introduce the notion of states below of the state-less IP layer, this happens implicitly. Each participant accesses a certificate, which contains the public value of the peer, by fetching it from a local database, a NFS fileserver, secure DNS, a X.509 infrastructure, NIS+, a built-in certificate discovery protocol, or other means. 'Certificate' denotes the fact that the public value has been signed by a trustworthy party. This may be an official certification authority (CA), or the involved end system itself, also depending on the form of certification one prefers, e.g. X.509 , PGP 3.0 (the new PGP API will foreseeably be available in its first versions at about end of March 1996) or any custom solution. The built-in certificate discovery protocol works automatically, and actually allows for independence from servers of any kind, as long as the certificate one receives from the remote peer contains a signature from a trusted entity.

The acquired public value of the communication peer is combined with the system's own secret value using the Diffie-Hellman scheme, thus resulting in the same, shared secret on both sides. As the shared secret should be usable for a long time (it might be expensive to create a new correctly signed certificate) it is not used to encrypt data directly. Instead, a random traffic key is generated and used to encrypt data, and this random traffic key is encrypted with the long-lived shared secret, using a symmetric algorithm.

When an outgoing packet is processed using the IPSEC mechanisms, newly defined headers will be added after the original IP header, namely 'Authentication Header (AH)' and 'Encapsulated Secure Payload (ESP)'. In the case of SKIP, a SKIP header, which contains per-packet security association information, is inserted additionally. It might consist of the above-mentioned encrypted traffic encryption key (usually 8 – 16 bytes long), information about which algorithms have been used to provide privacy and authentication, and the names of the public keys used on the sending and receiving side, together with additional information. The minimal (useful) size of a SKIP header is 20 bytes, but can grow to up to 68 bytes if one wants to employ all (optional) features. Under certain circumstances, header compression might be useful, and is currently being evaluated. Experience shows, that for normal modes of employment, header sizes of 24 bytes are a reasonable expectation.

SKIP: Source NSID	none (IPv4)
SKIP: Destination NSID	none (IPv4)
SKIP: Next Protocol Field	AH
SKIP: Counter n Field	Fri Jan 19 15:00:00 1996
SKIP: Kij alg (key encryption)	IDEA-CBC
SKIP: Crypt Alg	none
SKIP: Auth Alg	keyed MD5
SKIP: Encrypted Kp	9c4ff70e5ab9d3a7489afe6273ca6d8a
SKIP-AH: Next Protocol Field	TCP
SKIP-AH: length	16
SKIP-AH: SPI	SKIP association
SKIP-AH: Authentication Data	e3994872dbc3885c49235ae46c27585b
TELNET: SYN 2201335296 : 2201335296 (0)	
TELNET: win 9268 <mss 1324>	

Figure 3:Authentication only (start of a telnet session)

SKIP: Source NSID	MD5 DH Public Key
SKIP: Destination NSID	MD5 DH Public Key
SKIP: Next Protocol Field	AH
SKIP: Counter n Field	Fri Jan 19 15:00:00 1996
SKIP: Kij alg (key encryption)	3DES3-EDE-CBC
SKIP: Crypt Alg	RC4-128
SKIP: Auth Alg	keyed MD5
SKIP: Encrypted Kp	6132f68b975ee384932209a13fa69409
SKIP: Source Master Key-ID	5e3d37dbdb3b325a971bea1a5c3246b0
SKIP: Dest. Master Key-ID	67c593c41e09280b73a6522bb9051f58
SKIP-AH: Next Protocol Field	ESP
SKIP-AH: length	16
SKIP-AH: SPI	SKIP association
SKIP-AH: Authentication Data	fe2c701ea2b9091e26743905073ac8a0
SKIP-ESP: SPI	SKIP association
SKIP-ESP: Initalization Vector	00000000000002cb
SKIP-ESP: Opaque Data	32 bytes

Figure 4:Authenticated and encrypted data (ping)

To be prepared for future developments, SKIP introduces namespaces, and the concept of different possible key names. This means, that one does not have to depend on IP addresses to identify the sender and receiver of authenticated and encrypted data, but that one can directly address (the key of) a specific user. This can for instance be very useful in the context of mobile computing, or when IP addresses are assigned dynamically, and allows features like authentication and address rewriting of already encrypted data, e.g. in a virtual private network. Additionally, this allows for a selection of employed keys and algorithms on a per-user (or per-application) basis.

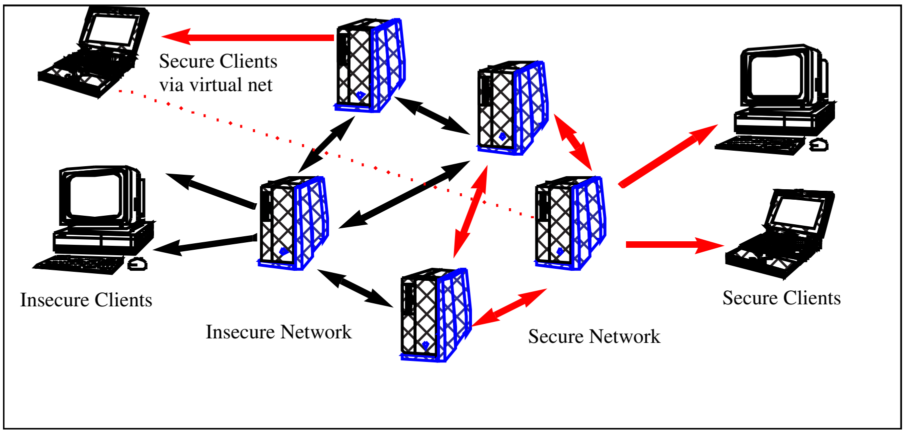

Figure 5:Example of a virtual private network (VPN)

4.4.State of the Art and Availability

The draft of the Internet standard SKIP has passed through its last decisive revision a short while ago, and will hopefully (if the required acceptance can be rallied) be on its way to proposed standard in Spring 1996. Commercial SKIP for the end user is only available in the USA (and Russia) due to the difficult export legislation in the USA, and cryptography-related legislation in Russia, but considerable effort is currently invested to change the situation.

Even without source code availability, installed systems can already be expanded to use AH/ESP, SKIP or Photuris. For SKIP, there currently exist multiple implementations which can be integrated into a binary UNIX Kernel (Solaris 2.4+, SunOS 4.1.3, NetBSD, Nextstep, IRIX, SCO Unix), and at least one Windows NT version is in work. For non-commercial use, there are multiple SKIP versions available on-line. One version from SUN Microsystems, domestic use only, "`http://skip.incog.com`", one version from Elvis+ in Russia (currently being rewritten), "`http://elvis.ru/skip`", and an internationally available source code release from the Computer Engineering and Networks Laboratory at ETH Zurich at "`http://www.tik.ee.ethz.ch/~skip`".

5.Further Reading

More information about SKIP and IPSEC can be found on the above mentioned Web pages, and in the various related RFCs and Internet drafts (RFC 1825 – 1829, 1851 – 1852, draft-ietf-ipsec-skip-*, draft-ietf-ipsec-photuris-*).