Internet Protocol, version 6, was conceived with two main goals: increase address space and improve security, relative to IPv4.1 The community achieved the first goal by increasing the IP address length from 32 bits to 128 bits. To meet the second goal, the Internet Engineering Task Force chartered the IP Security Working Group to design a security architecture and corresponding protocols that would provide cryptographically based security for IPv6 (http://www.ietf.org/html.charters/ipsec-charter.html). As work progressed, however, the IP community realized that the security architecture proposed for IPv6 could also be used for IPv4. Consequently, they extended this charter to retrofitting the security protocols, or IPsec protocols, into IPv4 implementations.2 Many IPv4 software vendors have announced they will support the IPsec protocols in future releases. This retrofitting is an important part of the working group’s charter because IPv6 deployment is turning out to be slow. As the sidebar “Should Security Be at the Internet Layer?” describes, some vendors are using other security technologies and protocols to retrofit security at some higher layer in the TCP/IP protocol stack. In this article, I overview the proposed security architecture and the two main protocols—the IP Security Protocol and the Internet Key Management Protocol—describe the risks they address, and touch on some implementation requirements. IPsec’s major advantage is that it can provide security services transparently to both applications and users. Also, the application programs using IPsec need not be modified in any way. This is particularly important when securing application programs that are not available in source code, which is common today. This transparency sets IPsec apart from security protocols that operate above the Internet layer. At present, IPsec is likely to be used in conjunction with and complemented by other security technologies, mechanisms, and protocols. Examples include firewalls and strong authentication mechanisms for access control, and higher layer security protocols for end-to-end communication security.3,4 In the near future, however, as virtual private networking and corporate intranets and extranets mature, IPsec is likely to be deployed on a larger scale.

RISKS ADDRESSED
During the past decade, reports of network-based attacks and exploitations of bugs and design limitations have grown dramatically.5 More recently, the use and proliferation of downloadable, executable content, such as that provided by Java applets and ActiveX controls, have opened new possibilities to attack networked computer systems and Internet sites. A 1996 survey (http://www.trouble.org/survey) of approximately 2,200 computer systems connected to the Internet found that almost two-thirds of the more interesting sites had serious security problems at one time. The IPsec protocols aim to address several of these problems:

- **Password sniffing.** An intruder eavesdrops on a communications line to capture passwords being transmitted unencrypted. Having captured a valid password, the intruder can use it to masquerade as a legitimate user. Using the IPsec protocols, the data traffic (including passwords) may be encrypted so that an intruder will not be able to capture passwords in the clear.
- **IP spoofing.** An intruder fakes his IP address to masquerade as a trusted host when address-based authentication is used. The IPsec protocols use cryptographically strong authentication techniques, instead of address-based authentication, to protect against IP spoofing attacks.
- **Session hijacking.** An intruder takes over a connection after the original source has been strongly authenticated. Again, the encryption facilities of the IPsec protocols protect a connection from an intruder takeover because the intruder does not know the session keys required to encrypt or decrypt the data stream.
- **Denial-of-service.** Examples include e-mail bomb-
Should Security Be at the Internet Layer?

Although the IETF IPsec Working Group was chartered to develop a security architecture for the Internet layer, some believe that the disadvantages of security at this level may outweigh the advantages. The main advantage is that applications need not be changed to use the IPsec protocols. Also an increasing number of applications—especially in real-time and multicast communications—are based on the connectionless User Datagram Protocol (UDP) that is generally hard to secure at the transport layer.

The main disadvantage of security at the Internet layer is that IP stacks must either be changed or extended. Because of the inherent complexity of the IP security architecture and protocol specifications, these changes or extensions are not at all trivial. In the long term, high-speed networking may also provide a performance problem.

As of this writing, it is not clear whether encryption rates will always be sufficiently fast to compete with data throughputs of future high-speed networks. Because of these disadvantages, alternative approaches have been proposed. The Secure Shell (SSH), Secure Sockets Layer (SSL), and Transport Layer Security (TLS) protocols have been proposed for the transport layer, and several application-specific enhancements and authentication and key distribution systems have been proposed for the application layer. All these technologies are in widespread use today, which is considerably slowing the deployment of IPsec protocols.

ARCHITECTURAL OVERVIEW

Figure 1 is an overview of the proposed IPsec architecture. In general, the IPsec protocols operate in a router or security gateway (firewall system). They can, in principle, be implemented in hosts and end systems, but at this time, that type of implementation is rare. Each IPsec module contains implementations of the IP Security and Internet Key Management Protocols. A common thread among modules is the security association—a “connection” that provides security services to the traffic it carries. As the figure shows, the IP Security Protocol contains the Authentication Header and Encapsulating Security Payload Protocols, which either alone or in combination provide the security association with corresponding services.

Figure 1 shows a single security association. However, if protection from both the Authentication Header and Encapsulating Security Payload Protocols is needed, the IPsec modules must establish and maintain two security associations. Similarly, to secure bidirectional communications between two hosts or security gateways, it must establish and maintain two associations, one in each direction.

A security policy database, which a user or system administrator establishes and maintains within the IPsec module, contains requirements for the specific level of protection. Each application has its IP packets selected for a particular processing mode by matching IP and transport layer header information (IP source and destination addresses, port numbers, and so on) against the entries in the database. A security association either affords each packet IPsec security services, discards it, or allows it to bypass IPsec protocols entirely.

Each security association is uniquely identified by a triple that consists of a security parameters index, an IP destination address, and a security protocol identifier, which refers to the Authentication Header or Encapsulating Security Payload protocol. The security association database contains an entry for each association that defines its security parameters.

As this overview shows, the proposed IP security architecture does not address all security aspects at the Internet layer. It is still unclear how to best use the protocols in conjunction with network address translation and how to more completely support IP multicast. However, IPsec is relatively young. As IPsec implementation data grows, the community should be able to better address these open questions.

IP Security Protocol

The IP Security Protocol services are provided through the Authentication Header and Encapsulating Security Payload Protocols.

Authentication Header Protocol. The Authentication Header Protocol is used when the integrity and authenticity of the IP packet or its payload must be protected but not necessarily the confidentiality of the packet itself. In a money transfer, for example, you want to be sure that the amount being transferred is not compromised but you don’t care that someone eventually learns what that amount is.

The Authentication Header Protocol provides an additional header between the IP and the transport layer headers that includes some authentication data, which the receiver then verifies to ascertain that the source of data is as claimed. A keyed one-way hash function, such as keyed MD5 or keyed SHA, is used to compute and verify the authentication header data. Computing and verifying authentication data in this way is much more efficient than encrypting and decrypting the entire IP packet.

The precision of the authentication depends on how the security association was established. If it was established with host-oriented keying, only two hosts can be distinguished, not the individual users within a host. If it was established with user-oriented keying, individual users can be distinguished. Establishing the
Encapsulating Security Payload Protocol. The Encapsulating Security Payload Protocol is used to encrypt and encapsulate either the transport layer payload or the entire IP packet (depending on the mode of use, as described later). The IP module must include an IP header and encrypt parts of the IP packet accordingly. Encryption is done on the sender side and decryption on the receiver side. The precise format of the payload data depends on the particular encryption algorithm and transformation in use. Unfortunately, export, import, and use of specific encryption algorithms may be regulated in some countries, which means the protocol’s services may be limited; in some cases the protocol cannot be used.

Modes of use. Both the Authentication Header and Encapsulating Security Payload Protocol support two modes of use: In transport mode, they encapsulate and protect primarily upper layer protocols. This mode is simpler and more commonly used between end systems. In tunnel mode, they protect tunneled IP packets, using IP encapsulation as an enabling technique. In encapsulation, the IP module authenticates and encrypts outgoing plaintext packets and encloses them in outer network layer headers, which are used to route the packets through the network. The receiving network layer protocol module decapsulates the incoming packets, strips off the outer network layer headers, and authenticates and decrypts the inner packets and forwards them to their final destination.

IP encapsulation requires no changes to the existing Internet routing infrastructure. Because authenticated and encrypted IP packets have an unencrypted, normal-looking outer IP header, they can be routed as usual and processed at their final destination. When one end of a security association is a security gateway, the association must be in tunnel mode. The exception is when traffic is destined for a security gateway, and the security gateway is acting as a host; in that case, the transport mode is allowed.

In tunnel mode, an entire IP packet is encapsulated and a new IP header is used to route the packet through the Internet. Consequently, the original source and destination IP addresses can be hidden to provide some services that ensure the confidentiality of traffic flows and protect against some traffic analysis attacks.

Protection provided. The Authentication Header Protocol provides data origin authentication services and connectionless data integrity services. Depending on the cryptographic algorithm used and method of keying, it may also support digital signatures to provide nonrepudiation services; that is, the sender of an IP packet won’t be able to deny having sent it, since it carries a digital signature. Finally, the Authentication Header Protocol may offer antireplay at the discretion of the receiver, which aids in keeping an eavesdropper from replaying old data.

The Encapsulating Security Payload Protocol provides data confidentiality services (protects data from unauthorized disclosure), and partial traffic flow confidentiality services (provide protection against traffic analysis), if it is used in tunnel mode and with padding data to hide the size of an IP packet. Like the Authentication Header Protocol, it provides authentication (data origin authentication, connectionless data integrity, and antireplay), but its authentication scope is narrower because the IP headers below the Encapsulating Security Payload header are not protected. If an implementer needs only the upper layer protocols authenticated, authentication using only the Encapsulating Security Payload Protocol is more space efficient than using both protocols.

Figure 1. Overview of the IPsec architecture. Two IPsec modules establish a security association. The security parameters for that association are stored in the security association database on both sides. Similarly, a security policy database is set up on both sides to define how the IPsec protocols are going to be used. Key establishment can be done manually or automatically, using the Internet Key Management Protocol. Communications between the two IKMP implementations (dotted line) is generally not connection-oriented. The IKMP establishes and maintains the security associations used by the IP Security Protocol and feeds the security association database accordingly.
In its current form, the Encapsulating Security Payload Protocol does not provide data compression. This presents a problem because data compression and encryption are interdependent. When encryption is employed at the Internet layer, lower layer protocols cannot compress the data. An alternative is to have higher layer protocols provide that function or, in the future, the IP itself can provide that function. Another IETF working group has been chartered to develop protocol specifications that will allow data compression (with no loss of data) on individual payloads before the payload is processed by a protocol that encrypts it (http://www.ietf.org/html.charters/ippcp-charter.html).

Both protocols provide access control services that protect against the unauthorized use of a resource or using a resource in an unauthorized manner. Note that a prerequisite for access control is proper authentication.

Internet Key Management Protocol

The IP Security Protocol assumes that security associations exist between entities that want to use IPsec. The aim of the Internet Key Management Protocol is to negotiate both parties’ cryptographic capabilities so that they agree on algorithms and parameters and to perform a key exchange. In other words, the protocol establishes and maintains the security associations that the Authentication Header and Encapsulating Security Protocols are to use. The Internet Key Management Protocol has a long history. The current protocol version combines the Internet Security Association Key Management Protocol developed by the US National Security Agency and the Oakley key determination protocol developed at the University of Arizona. The ISAKMP is used to negotiate mutually supported algorithms and mathematical structures for the Diffie-Hellman key exchange and the subsequent authentication step. The Oakley protocol is used to actually exchange keys. More recently, ISAKMP P/Oakley has been renamed the Internet Key Exchange and will probably replace the Internet Key Management Protocol at some point. The RFC documents that specify the Internet Key Exchange will ultimately result in a protocol that is elective for IPv4 implementations and mandatory for IPv6 implementations.

The ISAKMP P/Oakley (and IKE) proposal combines a Diffie-Hellman key exchange with a subsequent authentication of the Diffie-Hellman parameters. A key exchange occurs in three phases. In the first phase, the initiator and responder exchange cookies to protect against resource-clogging attacks (a special form of denial-of-service attacks, in which the attacker floods the victim with some computationally intensive tasks, such as doing several Diffie-Hellman key exchanges simultaneously). In the second phase, they perform a Diffie-Hellman key exchange to mutually compute a session key. This key can then be used within the IPsec protocols to protect subsequent communications. To authenticate each other and protect against the man-in-the-middle attack, the initiator and responder conclude with an exchange of digital signatures for authentication.

Both the mathematical structure (multiplicative group in a finite field) in which the Diffie-Hellman key exchange is to take place and the method for subsequent authentication are negotiable.

IMPLEMENTATION ISSUES

An October 1997 survey of IPsec implementers revealed that IPsec architecture is being implemented in hosts and/or security gateways in three ways.

- Integration into a native IP implementation. This is the most simple and straightforward way and applies to hosts and security gateways. However, it requires access to the corresponding source code.
- Bump-in-the-stack. IPsec protocols are implemented under an existing IP stack, between the native IP implementation and the local network drivers. Source code access for the IP stack is not required in this case, making it appropriate for use with legacy systems. This approach is usually employed with hosts.
- Bump-in-the-wire. As in bump-in-the-stack implementations, source code access for the IP stack is not required. Bump-in-the-wire implementations do, however, require outboard cryptographic processors. Fortunately, these are common in most military network security systems and in some commercial systems. Bump-in-the-wire implementations may be designed to serve both hosts and security gateways.

At present, most IPsec implementations are either bump-in-the-stack or bump-in-the-wire. The working group expects this to change, since more networking software vendors, such as Cisco Systems and Microsoft, are integrating the IPsec protocols into their products.

In the first two implementations, implementers must add an interface because of how key management interacts with the remaining parts of the IP and IPsec implementations. As Figure 2 illustrates, the IP Security Protocol must run in (protected) kernel space, while key management generally runs in user space. Consequently, there must be an interface that allows a key management module to feed the security associations table and key engine, either manually or automatically. The PF_KEY management sockets interface,
specified by the working group, provides a set of messages that the kernel can use to indicate the need for a new or updated security association, and that can be used to add, remove, or update security associations.

The suite of IPsec protocols is mandatory for IPv6 and therefore likely to become widespread during the next few years. However, there are advantages and disadvantages related to providing security at the Internet layer, as the sidebar "Should Security Be at the Internet Layer?" describes. The current trend in industry suggests that the IPsec protocols will be used primarily for virtual private networking and connecting mobile users to corporate intranets. A virtual private network consists of a collection of hosts and security gateways that implement either the IP Security Protocol alone with manual key management or both IPsec protocols. Virtual private networking is an attractive technology for securing intranet and extranet connections because it has little effect on applications. However, as a technology, it is still immature and requires significant work and expertise to deploy. For example, creating a multi-vendor virtual private network seems almost impossible. Hopefully, this will change.

The industry also needs to study interoperability and benchmark testing in more depth. The evolving nature of the architecture for IP security and the complexity of the IPsec protocols themselves make true interoperability hard to achieve. The throughput of IPsec implementations usually decreases with the number of security associations established and maintained simultaneously. However, the community needs adequate methods to determine that through benchmark testing.

The IPsec architecture in its current form is a solid first step toward exploring the ramifications of security at the Internet layer. As the community gathers more information from IPsec implementations, we will be in a better position to judge its overall effectiveness.

---

### References