IP Security

by William Stallings

In 1994, the *Internet Architecture Board* (IAB) issued a report entitled "Security in the Internet Architecture" (RFC 1636). The report stated the general consensus that the Internet needs more and better security, and it identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

These concerns are fully justified. As confirmation, the 1998 annual report from the *Computer Emergency Response Team* (CERT) lists over 1,300 reported security incidents affecting nearly 20,000 sites. The most serious types of attacks included IP spoofing, in which intruders create packets with false IP addresses and exploit applications that use authentication based on IP address; and various forms of eavesdropping and packet sniffing, in which attackers read transmitted information, including logon information and database contents.

In response to these issues, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IP (IPv4) and IPv6, meaning that vendors can begin offering these features now, and many vendors do now have some *IP Security Protocol* (IPSec) capability in their products.

Applications of IPSec

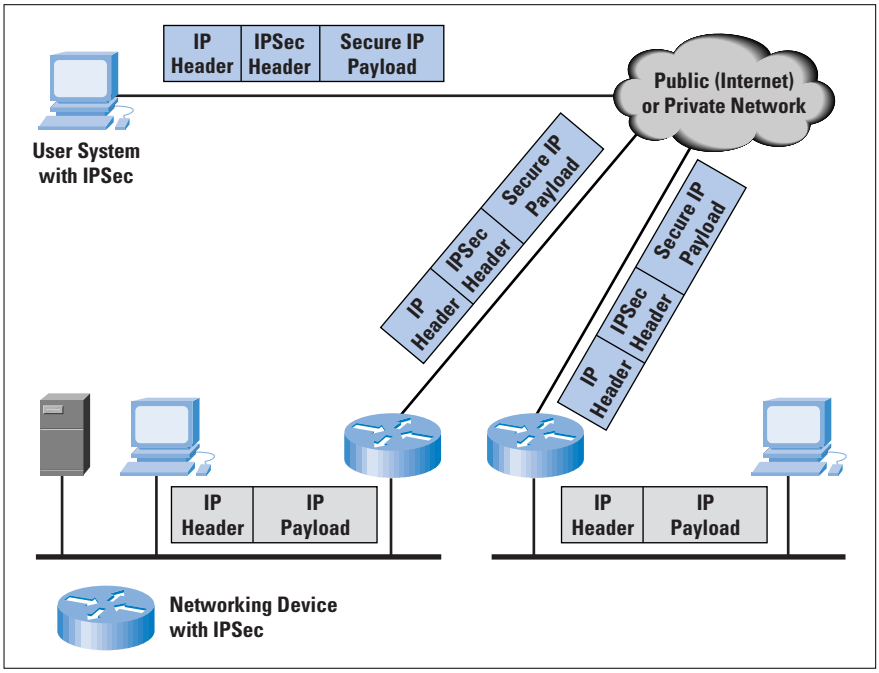
The Internet community has developed application-specific security mechanisms in numerous application areas, including electronic mail (*Privacy Enhanced Mail*, *Pretty Good Privacy* [PGP]), network management (*Simple Network Management Protocol Version 3* [SNMPv3]), Web access (Secure HTTP, *Secure Sockets Layer* [SSL]), and others. However, users have some security concerns that cut across protocol layers. For example, an enterprise can run a secure, private TCP/IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications.

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

- Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
- Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an *Internet Service Provider* (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
- Establishment of extranet and intranet connectivity with partners: IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
- Enhancement of electronic commerce security: Most efforts to date to secure electronic commerce on the Internet have relied upon securing Web traffic with SSL since that is commonly found in Web browsers and is easy to set up and run. There are new proposals that may utilize IPSec for electronic commerce.

The principal feature of IPSec that enables it to support these varied applications is that it can encrypt or authenticate *all* traffic at the IP level. Thus, all distributed applications, including remote logon, client/server, e-mail, file transfer, Web access, and so on, can be secured. Figure 1 shows a typical scenario of IPSec usage. An organization maintains LANs at dispersed locations. Traffic on each LAN does not need any special protection, but the devices on the LAN can be protected from the untrusted network with firewalls. Since we live in a distributed and mobile world, the people who need to access the services on each of the LANs may be at sites across the Internet. These people can use IPSec protocols to protect their access. These protocols can operate in networking devices, such as a router or firewall that connects each LAN to the outside world, or they may operate directly on the workstation or server. In the diagram, the user workstation can establish an IPSec tunnel with the network devices to protect all the subsequent sessions. After this tunnel is established, the workstation can have many different sessions with the devices behind these IPSec gateways. The packets going across the Internet will be protected by IPSec but will be delivered onto each LAN as a normal IP packet.

Figure 1: An IP Security Scenario



Benefits of IPSec

The benefits of IPSec include:

- When IPSec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPSec is below the transport layer (TCP, UDP), so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router. Even if IPSec is implemented in end systems, upper layer software, including applications, is not affected.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPSec can provide security for individual users if needed. This feature is useful for offsite workers and also for setting up a secure virtual subnetwork within an organization for sensitive applications.

Is IPSec the Right Choice?

There are already numerous products that implement IPSec, but it is not necessarily the security solution of choice for a network administrator. Christian Huitema, who at the time of the development of the initial IP-Sec documents was the head of the IAB, reports that the debates over how to provide Internet-based security were among the most heated that he ever observed. One issue concerns whether security is being provided at the right protocol layer. To provide security at the IP level, it is necessary for IPSec to be a part of the network code deployed on all participating platforms, including Windows NT, UNIX, and Macintosh systems. Unless a desired feature is available on all the deployed platforms, a given application may not be able to use that feature.

On the other hand, if the application, such as a Web browser/server combination, incorporates the function, the developer can guarantee that the features are available on all platforms for which the application is available. A related point is that many Internet applications are now being released with embedded security features. For example, Netscape and Internet Explorer support SSL, which protects Web traffic. Also, many vendors are planning to support *Secure Electronic Transaction* (SET), which protects credit-card transactions over the Internet. However, for a virtual private network, a network-level facility is needed, and this is what IPSec provides.

The Scope of IPSec

IPSec provides three main facilities: an authentication-only function, referred to as *Authentication Header* (AH), a combined authentication/encryption function called *Encapsulating Security Payload* (ESP), and a key exchange function. For virtual private networks, both authentication and encryption are generally desired, because it is important both to (1) assure that unauthorized users do not penetrate the virtual private network and (2) assure that eavesdroppers on the Internet cannot read messages sent over the virtual private network. Because both features are generally desirable, most implementations are likely to use ESP rather than AH. The key exchange function allows for manual exchange of keys as well as an automated scheme.

The IPSec specification is quite complex and covers numerous documents. The most important of these, issued in November 1998, are RFCs 2401, 2402, 2406, and 2408.

Security Associations

A key concept that appears in both the authentication and confidentiality mechanisms for IP is the *Security Association* (SA). An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both. A security association is uniquely identified by three parameters:

- *Security Parameters Index* (SPI): The SPI assigns a bit string to this SA that has local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- *IP destination address:* Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- *Security protocol identifier:* This indicates whether the association is an AH or ESP security association.

Hence, in any IP packet, the security association is uniquely identified by the destination address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

An IPSec implementation includes a security association database that defines the parameters associated with each SA. A security association is defined by the following parameters:

- *Sequence number counter:* A 32-bit value used to generate the sequence number field in AH or ESP headers
- *Sequence counter overflow:* A flag indicating whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on this SA
- *Anti-replay window:* Used to determine whether an inbound AH or ESP packet is a replay, by defining a sliding window within which the sequence number must fall
- *AH information:* Authentication algorithm, keys, key lifetimes, and related parameters being used with AH
- *ESP information:* Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP
- *Lifetime of this security association:* A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur
- *IPSec protocol mode:* Tunnel, transport, or wildcard (required for all implementations); these modes are discussed later
- *Path MTU:* Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations)

The key management mechanism that is used to distribute keys is coupled to the authentication and privacy mechanisms only by way of the security parameters index. Hence, authentication and privacy have been specified independent of any specific key management mechanism.

SA Selectors

IPSec provides the user with considerable flexibility in the way in which IPSec services are applied to IP traffic. IPSec provides a high degree of granularity in discriminating between traffic that is afforded IPSec protection and traffic that is allowed to bypass IPSec, in the former case relating IP traffic to specific SAs.

The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPSec) is the nominal *Security Policy Database* (SPD). In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic. In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.

Each SPD entry is defined by a set of IP and upper-layer protocol field values, called *selectors*. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA. Outbound processing obeys the following general sequence for each IP packet:

- Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs.
- Determine the SA (if any) for this packet and its associated SPI.
- Do the required IPSec processing (that is, AH or ESP processing).

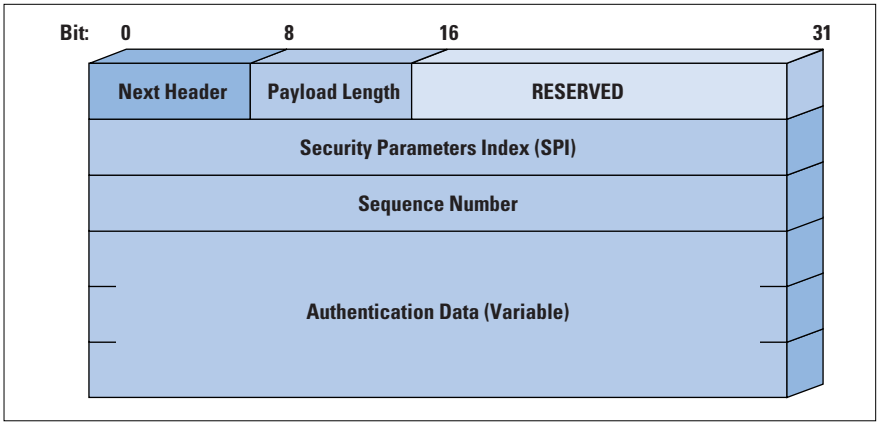
The following selectors determine an SPD entry:

- *Destination IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (for instance, behind a firewall).
- *Source IP address:* This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (for instance, behind a firewall).
- *UserID:* UserID is used to identify a policy tied to a valid user or system name.
- *Data sensitivity level:* The data sensitivity level is used for systems providing information flow security (for instance, "Secret" or "Unclassified").
- *Transport Layer protocol:* This value is obtained from the IPv4 protocol or IPv6 *Next Header* field. This may be an individual protocol number, a list of protocol numbers, or a range of protocol numbers.
- *IPSec protocol* (AH or ESP or AH/ESP): If present, this is obtained from the IPv4 Protocol or IPv6 Next Header field.
- *Source and destination ports:* These may be individual TCP or *User Datagram Protocol* (UDP) port values, an enumerated list of ports, or a wildcard port.
- *IPv6 class:* This class is obtained from the IPv6 header. It may be a specific IPv6 Class value or a wildcard value.
- *IPv6 flow label:* This label is obtained from the IPv6 header. It may be a specific IPv6 flow label value or a wildcard value.
- *IPv4 Type of Service* (TOS): The TOS is obtained from the IPv4 header. It may be a specific IPv4 TOS value or a wildcard value.

Authentication Header

The authentication header provides support for data integrity and authentication of IP packets. The data integrity feature ensures that undetected modification to the content of a packet in transit is not possible. The authentication feature enables an end system or network device to authenticate the user or application and filter traffic accordingly; it also prevents the address spoofing attacks observed in today's Internet. The AH also guards against the replay attack described later.

Figure 2: IPSec Authentication Header



Authentication is based on the use of a *Message Authentication Code* (MAC); hence the two parties must share a secret key. The authentication header consists of the following fields (Figure 2):

- *Next Header* (8 bits): This field identifies the type of header immediately following this header.
- *Payload Length* (8 bits): This field gives the length of the authentication header in 32-bit words, minus 2. For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header, and the Payload Length field has a value of 4.
- *Reserved* (16 bits): This field is reserved for future use.
- *Security Parameters Index* (32 bits): This field identifies a security association.
- *Sequence Number* (32 bits): This field contains a monotonically increasing counter value.
- *Authentication Data* (variable): This variable-length field (must be an integral number of 32-bit words) contains the *Integrity Check Value* (ICV), or MAC, for this packet.

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The *Sequence Number* field is designed to thwart such attacks.

When a new SA is established, the *sender* initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past $2^{32} - 1$ back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of $2^{32} - 1$ is reached, the sender should terminate this SA, and negotiate a new SA with a new key.

Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets will be delivered. Therefore, the IPSec authentication document dictates that the *receiver* should implement a window of size *W,* with a default of $W = 64$. The right edge of the window represents the highest sequence number, *N,* so far received for a valid packet. For any packet with a sequence number in the range from $N - W + 1$ to N that has been correctly received (that is, properly authenticated), the corresponding slot in the window is marked. Inbound processing proceeds as follows when a packet is received:

- If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
- If the received packet is to the right of the window and is new, the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
- If the received packet is to the left of the window, or if authentication fails, the packet is discarded; this is an auditable event.

Message Authentication Code

The message authentication algorithm is used to calculate a message authentication code, using an algorithm known as *HMAC.* HMAC takes as input a portion of the message and a secret key and produces a MAC as output. This MAC value is stored in the Authentication Data field of the AH header. The calculation takes place over the entire enclosed TCP segment plus the authentication header. When this IP packet is received at the destination, the same calculation is performed using the same key. If the calculated MAC equals the value of the received MAC, then the packet is assumed to be authentic. The authentication data field is calculated over:

- IP header fields that either do not change in transit (immutable) or that are predictable in value upon arrival at the endpoint for the AH SA. Fields that may change in transit and whose value on arrival are unpredictable are set to zero for purposes of calculation at both source and destination.
- The AH header other than the Authentication Data field. The Authentication Data field is set to zero for purposes of calculation at both source and destination.
- The entire upper-level protocol data, which is assumed to be immutable in transit (for instance, a TCP segment or an inner IP packet in tunnel mode).

For IPv4, examples of immutable fields are *Internet Header Length* and *Source Address.* An example of a mutable but predictable field is the Destination Address (with loose or strict source routing). Examples of mutable fields that are zeroed prior to ICV calculation are the *Time to Live* (TTL) and *Header Checksum* fields.

Note that both source and destination address fields are protected, so that address spoofing is prevented. For IPv6, examples in the base header are *Version* (immutable), *Destination Address* (mutable but predictable), and *Flow Label* (mutable and zeroed for calculation).

Encapsulating Security Payload

The encapsulating security payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality. As an optional feature, ESP can also provide the same authentication services as AH.

Figure 3: IPSec ESP Format

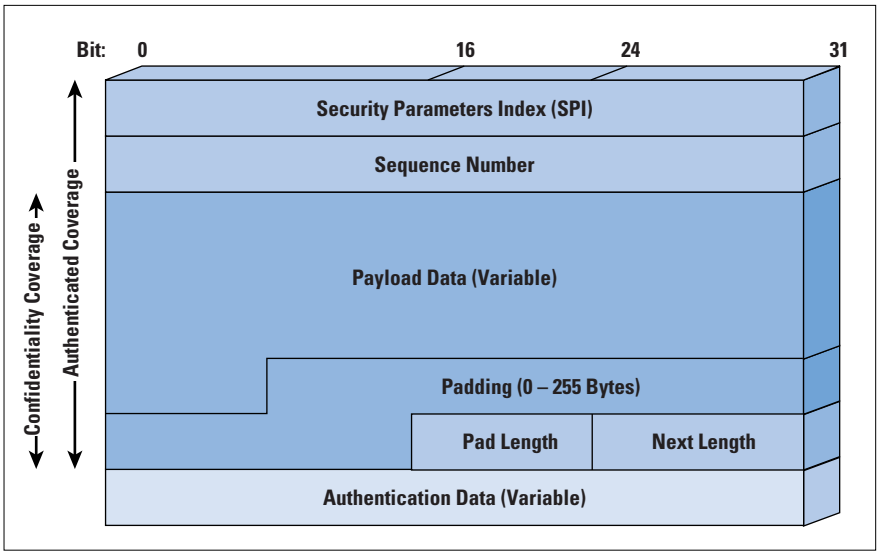


Figure 3 shows the format of an ESP packet. It contains the following fields:

- *Security Parameters Index* (32 bits): Identifies a security association
- *Sequence Number* (32 bits): A monotonically increasing counter value
- *Payload Data* (variable): A transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption
- *Padding* (0–255 bytes): Extra bytes that may be required if the encryption algorithm requires the plaintext to be a multiple of some number of octets
- *Pad Length* (8 bits): Indicates the number of pad bytes immediately preceding this field
- *Next Header* (8 bits): Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP)
- *Authentication Data* (variable): A variable-length field (must be an integral number of 32-bit words) that contains the integrity check value computed over the ESP packet minus the Authentication Data field

Encryption and Authentication Algorithms

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an *Initialization Vector* (IV), then this data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext. The current specification dictates that a compliant implementation must support the *Data Encryption Standard* (DES). A number of other algorithms have been assigned identifiers and could, therefore, be used for encryption; these include:

- Three-key triple DES
- RC5
- International Data Encryption Algorithm (IDEA)
- Three-key triple IDEA
- CAST
- Blowfish

It is now well known that DES is inadequate for secure encryption, so it is likely that many future implementations will use triple DES and eventually the *Advanced Encryption Standard* (AES). As with AH, ESP supports the use of a MAC, using HMAC.

Padding

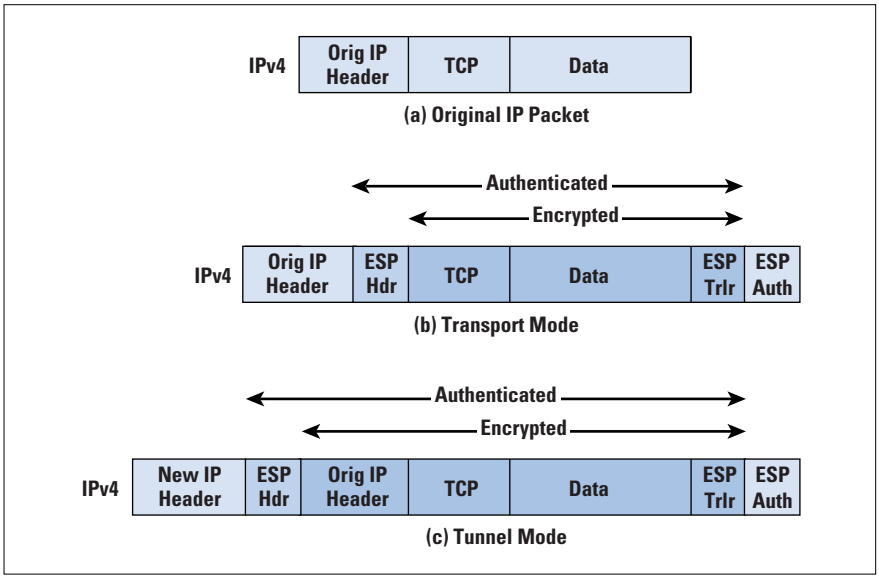
- The Padding field serves several purposes: If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (for instance, the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.
- Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

Figure 4 indicates the scope of ESP encryption and authentication in both transport and tunnel modes.

Transport and Tunnel Modes

Both AH and ESP support two modes of use: *transport* and *tunnel* mode.

Figure 4: Scope of ESP Encryption and Authentication



Transport Mode

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Examples include a TCP or UDP segment, or an *Internet Control Message Protocol* (ICMP) packet, all of which operate directly above IP in a host protocol stack. For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (for instance, TCP, UDP, ICMP) and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet. This setup is shown in Figure 4b. If authentication is selected, the ESP Authentication Data field is added after the ESP trailer. The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the ciphertext plus the ESP header.

Typically, transport mode is used for end-to-end communication between two hosts (for instance, communications between a workstation and a server, or two servers). When a host runs AH or ESP over IPv4, the payload is the data that normally follows the IP header. For IPv6, the payload is the data that normally follows both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.

ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header. All IPv4 packets have a *Next Header* field. This field contains a number for the payload protocol, such as 6 for TCP and 17 for UDP. For transport mode, the IP Next Header field is decimal 51 for AH, or 50 for ESP. This tells the receiving machine to interpret the remainder of the packet after the IP header as either AH or ESP. Both the AH and ESP headers also have a Next Header field.

As an example, let's examine a Telnet session within an ESP packet in transport mode. The IP header would contain 51 in the Next Header field. In the ESP header, the Next Header field would be 6 for TCP. Within the TCP header, Telnet would be identified as port 23.

Transport mode operation may be summarized for ESP as follows:

- At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its ciphertext to form the IP packet for transmission. Authentication is added if this option is selected.
- The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but will not need to examine the ciphertext.
- The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment. This process is similar for AH, however the payload (upper layer protocol) is not encrypted.

Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. This mode of operation is also reasonably efficient, adding little to the total length of the IP packet. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

Tunnel Mode

Tunnel mode encapsulates an entire IP packet within an IP packet to ensure that no part of the original packet is changed as it is moved through a network. The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way need to examine the inner IP header. For ESP, this is shown in Figure 4c. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis. Tunnel mode is used when one or both ends of an SA is a security gateway, such as a firewall or router that implements IPSec. With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPSec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPSec process in the firewall or secure router at the boundary of the local network.

Whereas the transport mode is suitable for protecting connections between hosts that support the ESP feature, the tunnel mode is useful in a configuration that includes a firewall or other sort of security gateway that protects a trusted network from external networks. In this latter case, encryption occurs only between an external host and the security gateway or between two security gateways. This setup relieves hosts on the internal network of the processing burden of encryption and simplifies the key distribution task by reducing the number of needed keys. Further, it thwarts traffic analysis based on ultimate destination.

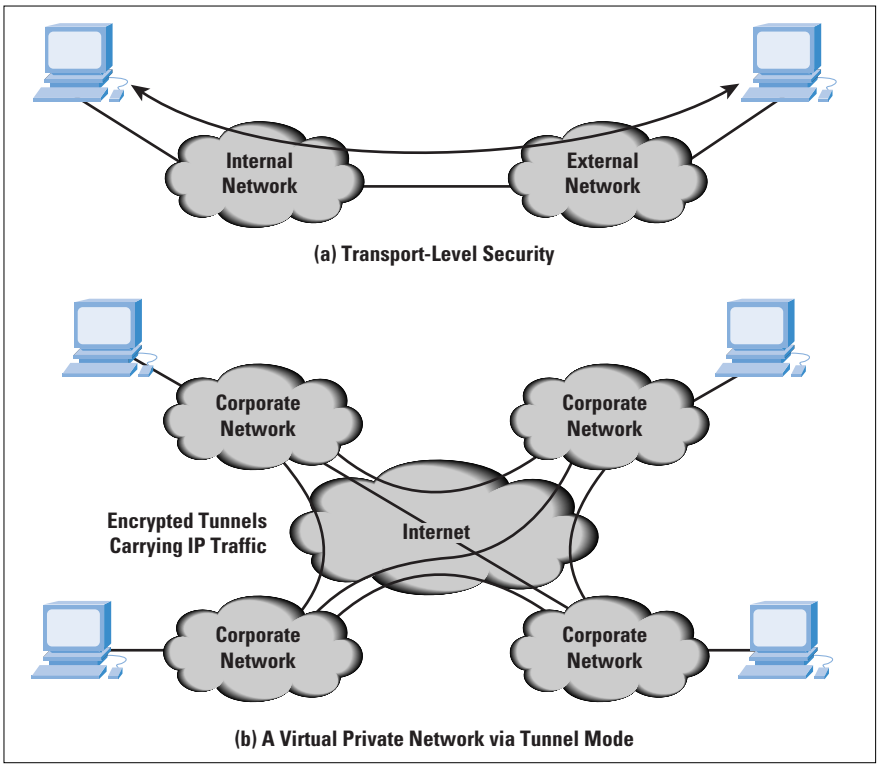
Let's use the diagram in Figure 1 as an example of how tunnel mode IPSec operates. The following steps occur for transfer of a transport-layer segment from the user system to one of the servers on one of the protected LANs.

- The user system prepares an inner IP packet with a destination address of the target host on the internal LAN. For a Telnet session, this packet would be a TCP packet with the original SYN flag set with a destination port set to 23. This entire IP packet is prefixed by an ESP header; then the packet and ESP trailer are encrypted and Authentication Data may be added. The Next Header field of the ESP header would be decimal 4 for IP-in-IP, indicating that the entire original IP packet is contained as the "payload." The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet. The Next Header field for this IP packet is 50 for ESP.
- The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the ciphertext.
- The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the gateway decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network.
- The inner packet is routed through zero or more routers in the internal network to the destination host. The receiver would have no indication that the packet had been encapsulated and protected by the "tunnel" between the user system and the gateway. It would see the packet as a request to start a Telnet session and would respond back with a TCP SYN/ACK, which would go back to the gateway. The gateway would encapsulate that packet into an IPSec packet and transport it back to the user system through this "tunnel." That return packet would be processed to find the original packet, which would contain the SYN/ACK for the Telnet session.

Common Uses of IPSec in Real Networks

Figure 5 shows two ways in which the IPSec ESP service can be used. In the upper part of the figure, encryption (and optionally authentication) is provided directly between two hosts. Figure 5b shows how tunnel mode operation can be used to set up a *Virtual Private Network* (VPN). In this example, an organization has four private networks interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts. By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability. The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA.

Figure 5: Transport-Mode versus Tunnel-Mode Encryption



Key Management

The key management portion of IPSec involves the determination and distribution of secret keys. The IPSec Architecture document mandates support for two types of key management:

- *Manual:* A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
- *Automated:* An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration. An automated system is the most flexible but requires more effort to configure and requires more software, so smaller installations are likely to opt for manual key management.

The default automated key management protocol for IPSec is referred to as *Internet Key Exchange* (IKE). IKE provides a standardized method for dynamically authenticating IPSec peers, negotiating security services, and generating shared keys. IKE has evolved from many different protocols and can be thought of as having two distinct capabilities. One of these capabilities is based on the *Internet Security Association and Key Management Protocol* (ISAKMP). ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes. ISAKMP by itself does not dictate a specific key exchange algorithm; rather, ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms. The actual key exchange mechanism in IKE is derived from Oakley and several other key exchange protocols that had been proposed for IPSec. Key exchange is based on the use of the Diffie-Hellman algorithm, but provides added security. In particular, Diffie-Hellman alone does not authenticate the two users that are exchanging keys, making the protocol vulnerable to impersonation. IKE includes mechanisms to authenticate the users.

Public Key Certificates

An important element of IPSec key management is the use of public key certificates. In essence, a public key certificate is provided by a trusted *Certificate Authority* (CA) to authenticate a user's public key. The essential elements include:

- Client software creates a pair of keys, one public and one private. The client prepares an unsigned certificate that includes a user ID and the user's public key. The client then sends the unsigned certificate to a CA in a secure manner.
- A CA creates a signature by calculating the hash code of the unsigned certificate and encrypting the hash code with the CA's private key; the encrypted hash code is the signature. The CA attaches the signature to the unsigned certificate and returns the now signed certificate to the client.
- The client may send its signed certificate to any other user. That user may verify that the certificate is valid by calculating the hash code of the certificate (not including the signature), decrypting the signature using the CA's public key, and comparing the hash code to the decrypted signature.

If all users subscribe to the same CA, then there is a common trust of that CA. All user certificates can be placed in the directory for access by all users. In addition, a user can transmit his or her certificate directly to other users. In either case, once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable.

If there is a large community of users, it may not be practical for all users to subscribe to the same CA. Because it is the CA that signs certificates, each participating user must have a copy of the CA's own public key to verify signatures. This public key must be provided to each user in an absolutely secure (with respect to integrity and authenticity) way so that the user has confidence in the associated certificates. Thus, with many users, it may be more practical for there to be many CAs, each of which securely provides its public key to some fraction of the users. In practice, there is not a single CA but rather a hierarchy of CAs. This complicates the problems of key distribution and of trust, but the basic principles are the same.

Whither IP Security

The driving force for the acceptance and deployment of secure IP is the need for business and government users to connect their private WAN/LAN infrastructure to the Internet for (1) access to Internet services and (2) use of the Internet as a component of the WAN transport system. Users need to isolate their networks and at the same time send and receive traffic over the Internet. The authentication and privacy mechanisms of secure IP provide the basis for a security strategy.

Because IP security mechanisms have been defined independent of their use with either the current IP or IPv6, deployment of these mechanisms does not depend on deployment of IPv6. Indeed, it is likely that we will see widespread use of secure IP features long before IPv6 becomes popular.

Recommended Web Sites

- The IPSec Working Group of the IETF. Charter for the group and latest RFCs and Internet Drafts for IPSec:
  `http://ietf.org/html.charters/ipsec-charter.html`
- IPSec Resources: List of companies implementing IPSec, implementation survey, and other useful material:
  `http://web.mit.edu/tytso/www/ipsec/index.html`

WILLIAM STALLINGS is a consultant, lecturer, and author of over a dozen books on data communications and computer networking. He has a Ph.D. in computer science from M.I.T. His latest book is *Local and Metropolitan Area Networks, Sixth Edition* (Prentice Hall, 2000). His home in cyberspace is **`WilliamStallings.com`** and he can be reached at **`ws@shore.net`**