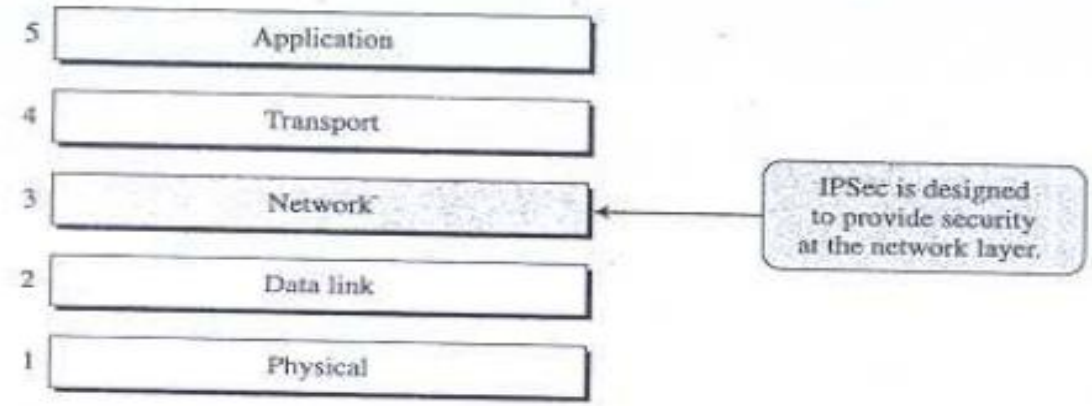
IPSec (Internet Protocol Security) (Security at Network Layer)

G.P. Biswas

Prof/CSE, IIT, Dhanbad

Introduction

Figure 18.1 *TCP/IP protocol suite and IPSec*



IPSec can be useful in several areas. First, it can enhance the security of those client/server programs, such as electronic mail, that use their own security protocols. Second, it can enhance the security of those client/server programs, such as HTTP, that use the security services provided at the transport layer. It can provide security for those client/server programs that do not use the security services provided at the transport layer. It can provide security for node-to-node communication programs such as routing protocols.

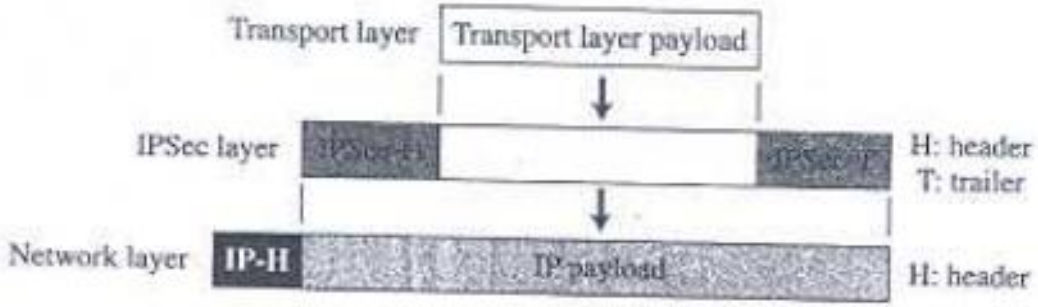
Transport Mode IPSec

IPSec operates in one of two different modes: transport mode or tunnel mode.

Transport Mode

In **transport mode,** IPSec protects what is delivered from the transport layer to the network layer. In other words, transport mode protects the network layer payload, the payload to be encapsulated in the network layer, as shown in Figure 18.2.

Figure 18.2 *IPSec in transport mode*



Note that transport mode does not protect the IP header. In other words, transport mode does not protect the whole IP packet; it protects only the packet from the transport layer (the IP layer payload). In this mode, the IPSec header (and trailer) are added to the information coming from the transport layer. The IP header is added later.

IPSec in transport mode does not protect the IP header; it only protects the information coming from the transport layer.

Transport mode is normally used when we need host-to-host (end-to-end) protection of data. The sending host uses IPSec to authenticate and/or encrypt the payload delivered from the transport layer. The receiving host uses IPSec to check the authentication and/or decrypt the IP packet and deliver it to the transport layer. Figure 18.3 shows this concept.

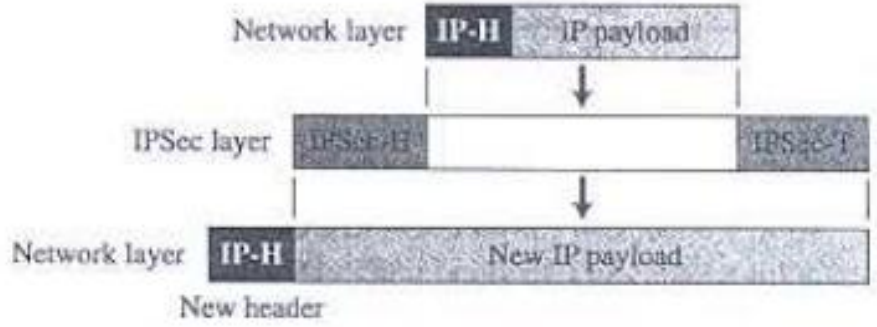
Figure 18.3 *Transport mode in action*

Tunnel Mode IPSec

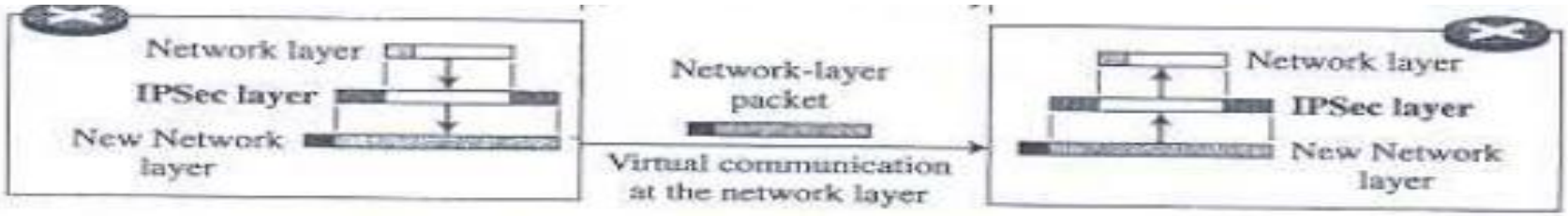
Tunnel Mode

In **tunnel mode,** IPSec protects the entire IP packet. It takes an IP packet, including the header, applies IPSec security methods to the entire packet, and then adds a new IP header, as shown in Figure 18.4.

Figure 18.4 *IPSec in tunnel mode*



The new IP header, as we will see shortly, has different information than the original IP header. Tunnel mode is normally used between two routers, between a host and a router, or between a router and a host, as shown in Figure 18.5. In other words, tunnel mode is used when either the sender or the receiver is not a host. The entire original



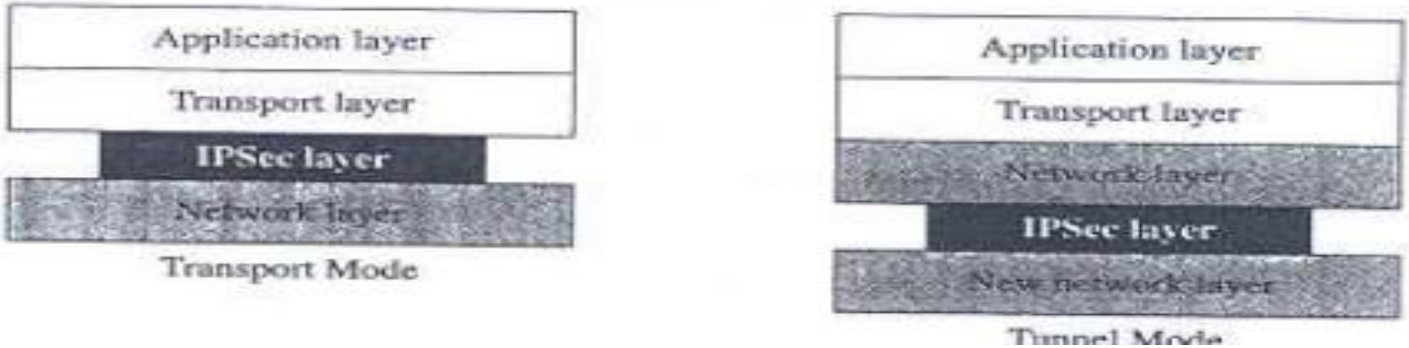
packet is protected from intrusion between the sender and the receiver, as if the whole packet goes through an imaginary tunnel.

IPSec in tunnel mode protects the original IP header.

Comparison

In transport mode, the IPSec layer comes between the transport layer and the network layer. In tunnel mode, the flow is from the network layer to the IPSec layer and then back to the network layer again. Figure 18.6 compares the two modes.

Figure 18.6 *Transport mode versus tunnel mode*



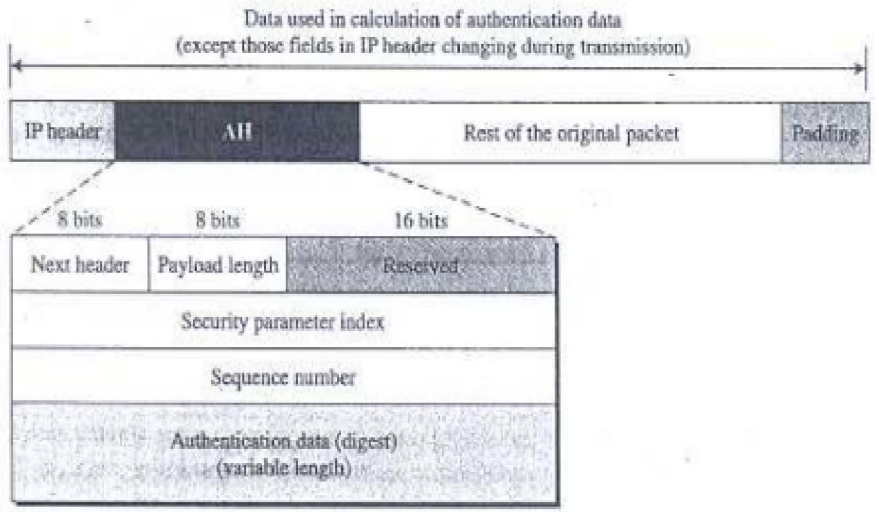
Two Security Protocols

IPSec defines two protocols—the Authentication Header (AH) Protocol and the Encapsulating Security Payload (ESP) Protocol—to provide authentication and/or encryption for packets at the IP level.

Authentication Header (AH)

The **Authentication Header (AH) Protocol** is designed to authenticate the source host and to ensure the integrity of the payload carried in the IP packet. The protocol uses a hash function and a symmetric key to create a message digest; the digest is inserted in the authentication header. The AH is then placed in the appropriate location, based on the mode (transport or tunnel). Figure 18.7 shows the fields and the position of the authentication header in transport mode.

Figure 18.7 *Authentication Header (AH) protocol*



When an IP datagram carries an authentication header, the original value in the protocol field of the IP header is replaced by the value 51. A field inside the authentication header (the next header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram). The addition of an authentication header follows these steps:

1. An authentication header is added to the payload with the authentication data field set to 0.
2. Padding may be added to make the total length even for a particular hashing algorithm.
3. Hashing is based on the total packet. However, only those fields of the IP header that do not change during transmission are included in the calculation of the message digest (authentication data).
4. The authentication data are inserted in the authentication header.
5. The IP header is added after changing the value of the protocol field to 51.

A brief description of each field follows:

- **Next header.** The 8-bit next header field defines the type of payload carried by the IP datagram (such as TCP, UDP, ICMP, or OSPF). It has the same function as the protocol field in the IP header before encapsulation. In other words, the process copies the value of the protocol field in the IP datagram to this field. The value of the protocol field in the new IP datagram is now set to 51 to show that the packet carries an authentication header.

- **Payload length.** The name of this 8-bit field is misleading. It does not define the length of the payload; it defines the length of the authentication header in 4-byte multiples, but it does not include the first 8 bytes.
- **Security parameter index.** The 32-bit security parameter index (SPI) field plays the role of a virtual circuit identifier and is the same for all packets sent during a connection called a Security Association (discussed later).
- **Sequence number.** A 32-bit sequence number provides ordering information for a sequence of datagrams. The sequence numbers prevent a playback. Note that the sequence number is not repeated even if a packet is retransmitted. A sequence number does not wrap around after it reaches 2^{32}; a new connection must be established.
- **Authentication data.** Finally, the authentication data field is the result of applying a hash function to the entire IP datagram except for the fields that are changed during transit (e.g., time-to-live).

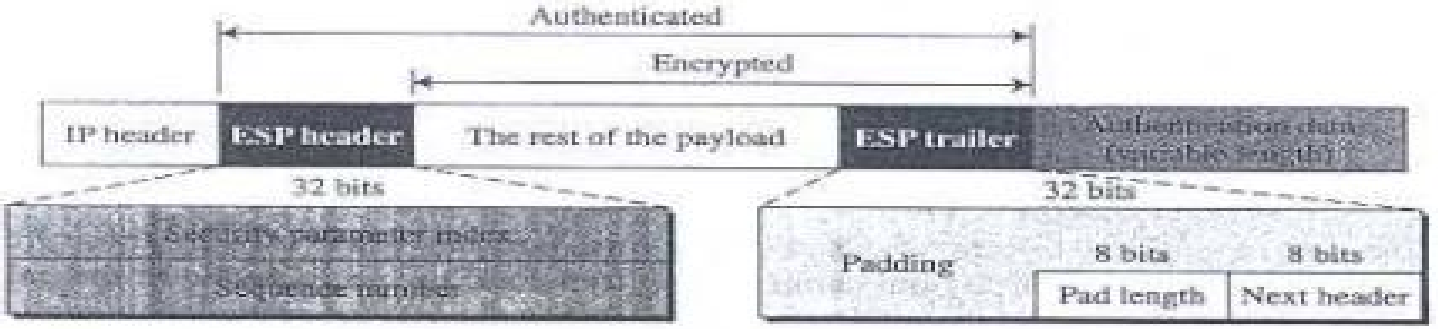
The AH protocol provides source authentication and data integrity, but not privacy.

ESP

Encapsulating Security Payload (ESP)

The AH protocol does not provide privacy, only source authentication and data integrity. IPSec later defined an alternative protocol, **Encapsulating Security Payload (ESP)**, that provides source authentication, integrity, and privacy. ESP adds a header and trailer. Note that ESP's authentication data are added at the end of the packet, which makes its calculation easier. Figure 18.8 shows the location of the ESP header and trailer.

Figure 18.8 *ESP*



When an IP datagram carries an ESP header and trailer, the value of the protocol field in the IP header is 50. A field inside the ESP trailer (the next-header field) holds the original value of the protocol field (the type of payload being carried by the IP datagram, such as TCP or UDP). The ESP procedure follows these steps:

1. An ESP trailer is added to the payload.
2. The payload and the trailer are encrypted.
3. The ESP header is added.
4. The ESP header, payload, and ESP trailer are used to create the authentication data.

5. The authentication data are added to the end of the ESP trailer.
6. The IP header is added after changing the protocol value to 50.

The fields for the header and trailer are as follows:

- **Security parameter index.** The 32-bit security parameter index field is similar to that defined for the AH protocol.
- **Sequence number.** The 32-bit sequence number field is similar to that defined for the AH protocol.
- **Padding.** This variable-length field (0 to 255 bytes) of 0s serves as padding.
- **Pad length.** The 8-bit pad-length field defines the number of padding bytes. The value is between 0 and 255; the maximum value is rare.
- **Next header.** The 8-bit next-header field is similar to that defined in the AH protocol. It serves the same purpose as the protocol field in the IP header before encapsulation.
- **Authentication data.** Finally, the authentication data field is the result of applying an authentication scheme to parts of the datagram. Note the difference between the authentication data in AH and ESP. In AH, part of the IP header is included in the calculation of the authentication data; in ESP, it is not.

ESP provides source authentication, data integrity, and privacy.

IPv4 and IPv6

IPSec supports both IPv4 and IPv6. In IPv6, however, AH and ESP are part of the extension header.

AH versus ESP

The ESP protocol was designed after the AH protocol was already in use. ESP does whatever AH does with additional functionality (privacy). The question is, Why do we need AH? The answer is that we don't. However, the implementation of AH is already included in some commercial products, which means that AH will remain part of the Internet until these products are phased out.

Services Provided by IPSec

The two protocols, AH and ESP, can provide several security services for packets at the network layer. Table 18.1 shows the list of services available for each protocol.

Table 18.1 *IPSec services*

Services	*AH*	*ESP*
Access control	yes	yes
Message authentication (message integrity)	yes	yes
Entity authentication (data source authentication)	yes	yes
Confidentiality	no	yes
Replay attack protection	yes	yes

Access Control

IPSec provides access control indirectly using a Security Association Database (SAD), as we will see in the next section. When a packet arrives at a destination, and there is no Security Association already established for this packet, the packet is discarded.

Message Integrity

Message integrity is preserved in both AH and ESP. A digest of data is created and sent by the sender to be checked by the receiver.

Entity Authentication

The Security Association and the keyed-hash digest of the data sent by the sender authenticate the sender of the data in both AH and ESP.

Confidentiality

The encryption of the message in ESP provides confidentiality. AH, however, does not provide confidentiality. If confidentiality is needed, one should use ESP instead of AH.

Replay Attack Protection

In both protocols, the replay attack is prevented by using sequence numbers and a sliding receiver window. Each IPSec header contains a unique sequence number when the Security Association is established. The number starts from 0 and increases until the value reaches $2^{32} - 1$ (the size of the sequence number field is 32 bits). When the sequence number reaches the maximum, it is reset to 0 and, at the same time, the old Security Association (see the next section) is deleted and a new one is established. To prevent processing duplicate packets, IPSec mandates the use of a fixed-size window at the receiver. The size of the window is determined by the receiver with a default value of 64. Figure 18.9 shows a replay window. The window is of a fixed size, W. The shaded packets signify received packets that have been checked and authenticated.

Figure 18.9 *Replay window*

When a packet arrives at the receiver, one of three things can happen, depending on the value of the sequence number.

1. The sequence number of the packet is less than N. This puts the packet to the left of the window. In this case, the packet is discarded. It is either a duplicate or its arrival time has expired.
2. The sequence number of the packet is between N and $(N + W - 1)$, inclusive. This puts the packet inside the window. In this case, if the packet is new (not marked) and it passes the authentication test, the sequence number is marked and the packet is accepted. Otherwise, it is discarded.
3. The sequence number of the packet is greater than $(N + W - 1)$. This puts the packet to the right of the window. In this case, if the packet is authenticated, the corresponding sequence number is marked and the window slides to the right to cover the newly marked sequence number. Otherwise, the packet is discarded. Note that it may happen that a packet arrives with a sequence number much larger than $(N + W)$ (very far from the right edge of the window). In this case, the sliding of the window may cause many unmarked numbers to fall to the left of the window. These packets, when they arrive, will never be accepted; their time has expired. For example, in Figure 18.9, if a packet arrives with sequence number $(N + W + 3)$, the window slides and the left edge will be at the beginning of $(N + 3)$. This means the sequence number $(N + 2)$ is now out of the window. If a packet arrives with this sequence number, it will be discarded.

Security Association (SA) in IPSec

Security Association is a very important aspect of IPSec. IPSec requires a logical relationship, called a **Security Association (SA)**, between two hosts. This section first discusses the idea and then shows how it is used in IPSec.

Idea of Security Association

A Security Association is a contract between two parties; it creates a secure channel between them. Let us assume that Alice needs to unidirectionally communicate with Bob. If Alice and Bob are interested only in the confidentiality aspect of security, they can get a shared secret key between themselves. We can say that there are two Security Associations (SAs) between Alice and Bob; one outbound SA and one inbound SA. Each of them stores the value of the key in a variable and the name of the encryption/decryption algorithm in another. Alice uses the algorithm and the key to encrypt a message to Bob; Bob uses the algorithm and the key when he needs to decrypt the message received from Alice. Figure 18.10 shows a simple SA.

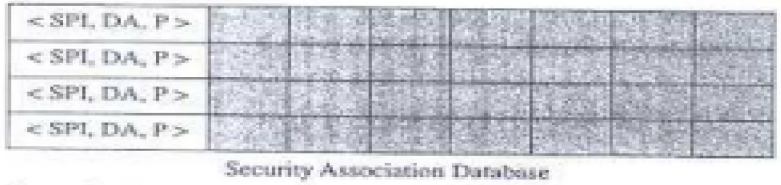
The Security Associations can be more involved if the two parties need message integrity and authentication. Each association needs other data such as the algorithm for message integrity, the key, and other parameters. It can be much more complex if the parties need to use specific algorithms and specific parameters for different protocols, such as IPSec AH or IPSec ESP.

SA

Security Association Database (SAD)

A Security Association can be very complex. This is particularly true if Alice wants to send messages to many people and Bob needs to receive messages from many people. In addition, each site needs to have both inbound and outbound SAs to allow bidirectional communication. In other words, we need a set of SAs that can be collected into a database. This database is called the **Security Association Database (SAD).** The database can be thought of as a two-dimensional table with each row defining a single SA. Normally, there are two SADs, one inbound and one outbound. Figure 18.11 shows the concept of outbound and inbound SADs for one entity.

Figure 18.11 *SAD*



Security Association Database

Legend:

SPI: Security Parameter Index
DA: Destination Address
AH/ESP: Information for either one
P: Protocol
Mode: IPSec Mode Flag
SN: Sequence Number
OF: Overflow Flag
ARW: Anti-Replay Window
LT: Lifetime
MTU: Path MTU (Maximum Transfer Unit)

When a host needs to send a packet that must carry an IPSec header, the host needs to find the corresponding entry in the outbound SAD to find the information for applying security to the packet. Similarly, when a host receives a packet that

carries an IPSec header, the host needs to find the corresponding entry in the inbound SAD to find the information for checking the security of the packet. This searching must be specific in the sense that the receiving host needs to be sure that correct information is used for processing the packet. Each entry in an inbound SAD is selected using a triple index: security parameter index, destination address, and protocol.

- **Security Parameter Index.** The security parameter index (SPI) is a 32-bit number that defines the SA at the destination. As we will see later, the SPI is determined during the SA negotiation. The same SPI is included in all IPSec packets belonging to the same inbound SA.
- **Destination Address.** The second index is the destination address of the host. We need to remember that a host in the Internet normally has one unicast destination address, but it may have several multicast addresses. IPSec requires that the SAs be unique for each destination address.
- **Protocol.** IPSec has two different security protocols: AH and ESP. To separate the parameters and information used for each protocol, IPSec requires that a destination define a different SA for each protocol.

The entries for each row are called the SA parameters. Typical parameters are shown in Table 18.2.

Table 18.2 *Typical SA Parameters*

Sequence Number Counter	This is a 32-bit value that is used to generate sequence numbers for the AH or ESP header.
Sequence Number Overflow	This is a flag that defines a station's options in the event of a sequence number overflow.
Anti-Replay Window	This detects an inbound replayed AH or ESP packet.
AH Information	This section contains information for the AH protocol: 1. Authentication algorithm 2. Keys 3. Key lifetime 4. Other related parameters
ESP Information	This section contains information for the ESP protocol: 1. Encryption algorithm 2. Authentication algorithm 3. Keys 4. Key lifetime 5. Initiator vectors 6. Other related parameters
SA Lifetime	This defines the lifetime for the SA.
IPSec Mode	This defines the mode, transport or tunnel.
Path MTU	This defines the path MTU (fragmentation).

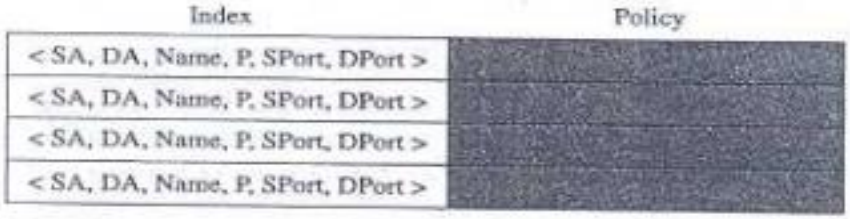
Security Policy

Another import aspect of IPSec is the **Security Policy (SP),** which defines the type of security applied to a packet when it is to be sent or when it has arrived. Before using the SAD, discussed in the previous section, a host must determine the predefined policy for the packet.

Security Policy Database

Each host that is using the IPSec protocol needs to keep a **Security Policy Database (SPD).** Again, there is a need for an inbound SPD and an outbound SPD. Each entry in the SPD can be accessed using a sextuple index: source address, destination address, name, protocol, source port, and destination port, as shown in Figure 18.12.

Figure 18.12 *SPD*



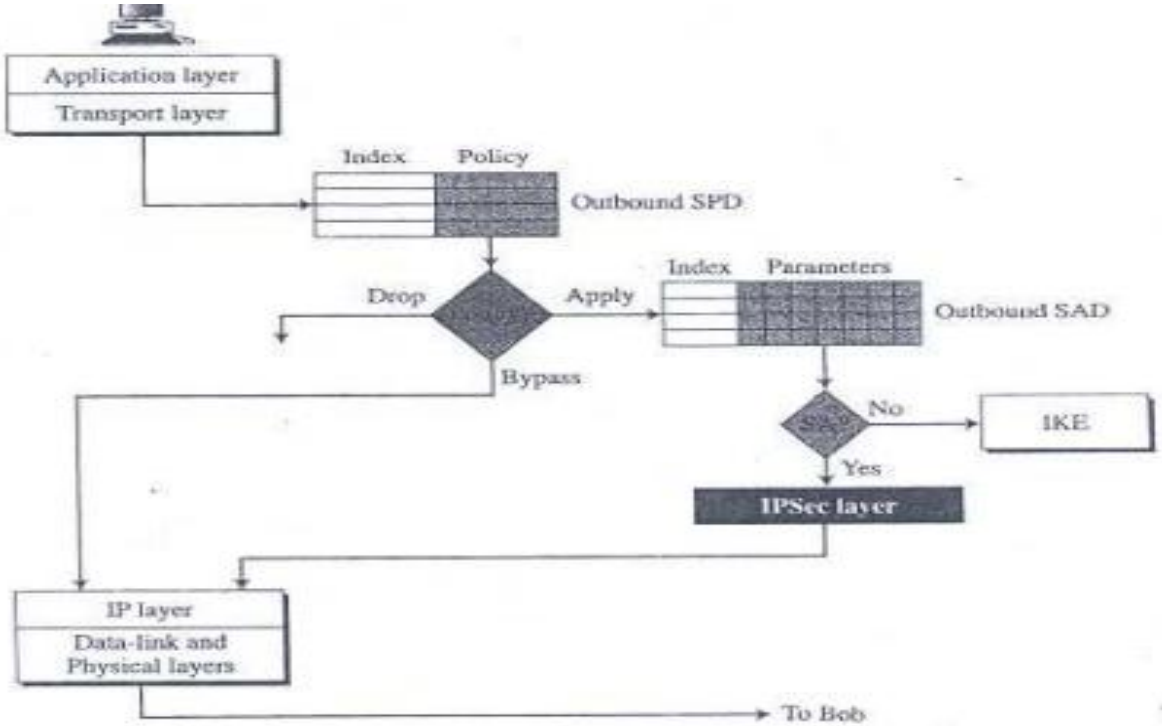
Source and destination addresses can be unicast, multicast, or wildcard addresses. The name usually defines a DNS entity. The protocol is either AH or ESP. The source and destination ports are the port addresses for the process running at the source and destination hosts.

Outbound SPD

When a packet is to be sent out, the outbound SPD is consulted. Figure 18.13 shows the processing of a packet by a sender.

The input to the outbound SPD is the sextuple index; the output is one of the three following cases:

1. **Drop.** This means that the packet defined by the index cannot be sent; it is dropped.
2. **Bypass.** This means that there is no policy for the packet with this policy index; the packet is sent, bypassing the security header application.

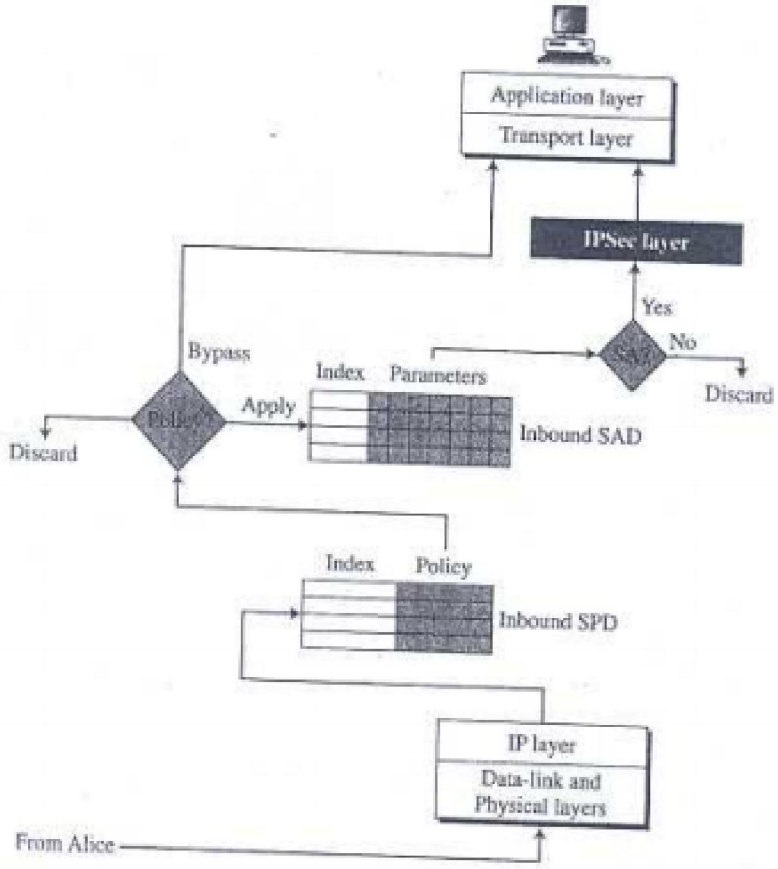


3. **Apply.** In this case, the security header is applied. Two situations may occur.
   a. If an outbound SA is already established, the triple SA index is returned that selects the corresponding SA from the outbound SAD. The AH or ESP header is formed; encryption, authentication, or both are applied based on the SA selected. The packet is transmitted.
   b. If an outbound SA is not established yet, the Internet Key Exchange (IKE) protocol (see the next section) is called to create an outbound and inbound SA for this traffic. The outbound SA is added to the outbound SAD by the source; the inbound SA is added to the inbound SAD by the destination.

Inbound SPD

When a packet arrives, the inbound SPD is consulted. Each entry in the inbound SPD is also accessed using the same sextuple index. Figure 18.14 shows the processing of a packet by a receiver.

3. **Apply.** In this case, the security header must be processed. Two situations may occur:
   a. If an inbound SA is already established, the triple SA index is returned that selects the corresponding inbound SA from the inbound SAD. Decryption, authentication, or both are applied. If the packet passes the security criteria, the AH or ESP header is discarded and the packet is delivered to the transport layer.
   b. If an SA is not yet established, the packet must be discarded.



The input to the inbound SPD is the sextuple index; the output is one of the three following cases:

1. **Discard.** This means that the packet defined by that policy must be dropped.
2. **Bypass.** This means that there is no policy for a packet with this policy index; the packet is processed, ignoring the information from AH or ESP header. The packet is delivered to the transport layer.