Module 3

IP Security
Overview of IP security(IPsec), IP Security Architecture, Modes of Operation, Security Associations(SA), Authentication Header(AH), Encapsulating Security Payload(ESP), Internet Key Exchange.

IP security (IPSec) is a collection of protocols designed to provide security for a packet at the network level. IP-level security encompasses three functional areas: **authentication, confidentiality,** and **key management.**

- The **authentication** mechanism assures that a received packet was transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit.
- The **confidentiality** facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- The **key management** facility is concerned with the secure exchange of keys.

3.1 IP Security Overview

- In 1994, the Internet Architecture Board (IAB) issued a report titled “Security in the Internet Architecture” (RFC 1636). The report 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.
- To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. These security capabilities were designed to be usable both with the current IPv4 and the future IPv6.

➢ Applications of IPsec

- 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.

- **Establishing 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.
- **Enhancing electronic commerce security**: IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

- The principal feature of IPsec that enables it to support these varied applications is that it can encrypt and/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 3.1 is a scenario of IPsec usage. An organization maintains LANs at dispersed locations. Nonsecure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.
- These protocols operate in networking devices, such as a router or firewall, that connect each LAN to the outside world. The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN.

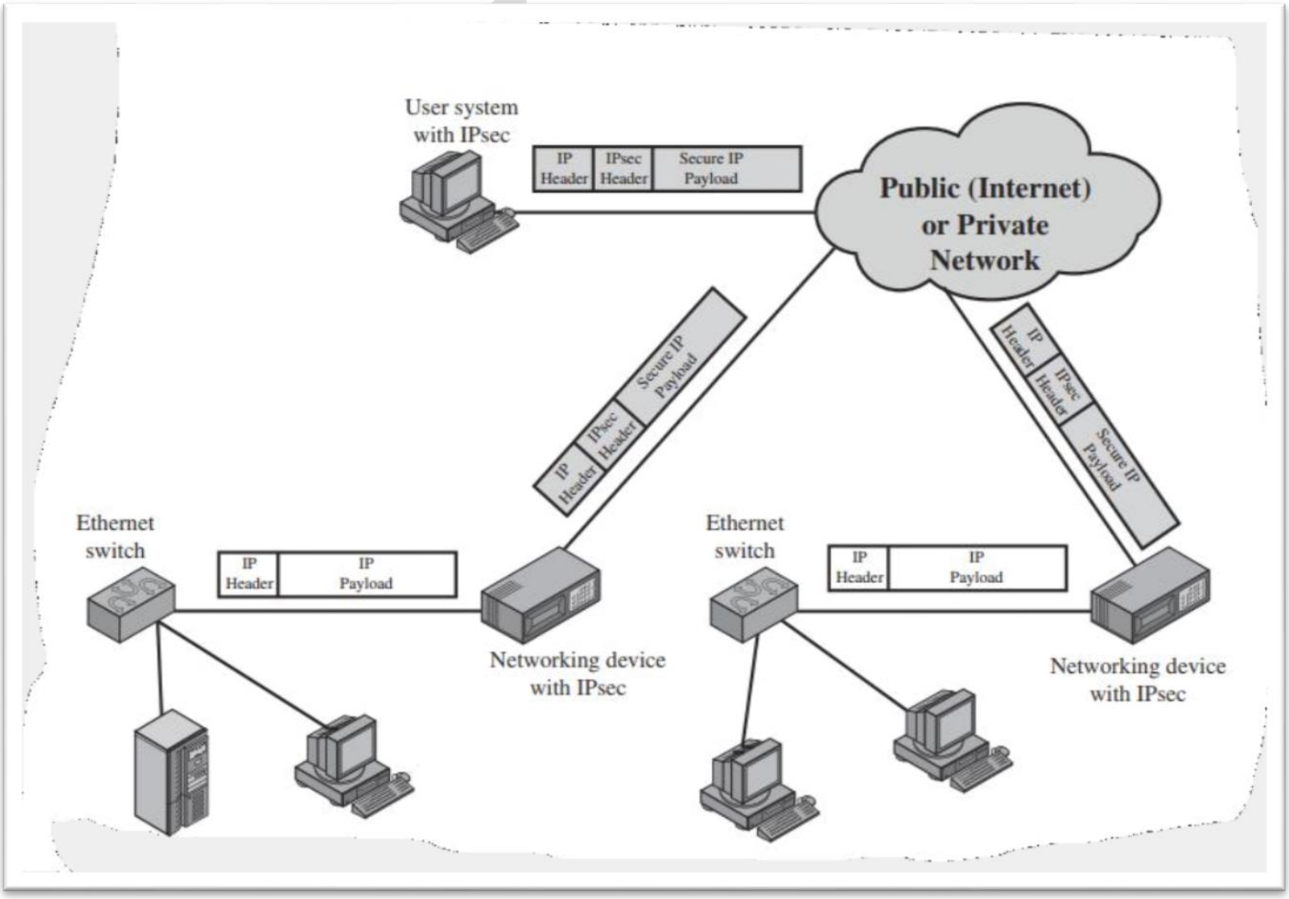


Figure 3.1 : IP Security Scenario

Benefits of IPsec

Some of the benefits of IPsec are:

- When IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and 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.
- 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 is useful for offsite workers and for setting up a secure virtual subnet work within an organization for sensitive applications.

Routing Applications

In addition to supporting end users and protecting systems and networks, IPsec can play a vital role in following examples of the use of IPsec. IPsec can assure that

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbour advertisement (a router seeks to establish or maintain a neighbour relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial IP packet was sent.
- A routing update is not forged.

Without such security measures, an opponent can disrupt communications or divert some traffic.

IPsec Documents

IPsec encompasses three functional areas: authentication, confidentiality, and key management. The documents can be categorized into the following groups.

- **Architecture**: Covers the general concepts, security requirements, definitions, mechanisms defining IPsec technology.
- **Authentication Header (AH)**: AH is an extension header to provide message authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP, the use of AH is deprecated. It is included in IPsecv3 for backward compatibility but should not be used in new applications.

- **Encapsulating Security Payload (ESP)**: ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication.
- **Internet Key Exchange (IKE)**: This is a collection of documents describing the key management schemes for use with IPsec.
- **Cryptographic algorithms**: This category encompasses a large set of documents that define and describe cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange.
- **Other**: There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content.

➢ IPsec Services

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services. Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, **Authentication Header (AH);** and a combined encryption/authentication protocol designated by the format of the packet for that protocol, **Encapsulating Security Payload (ESP).** IPSec services are listed below:

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

3.2 Modes of Operation

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

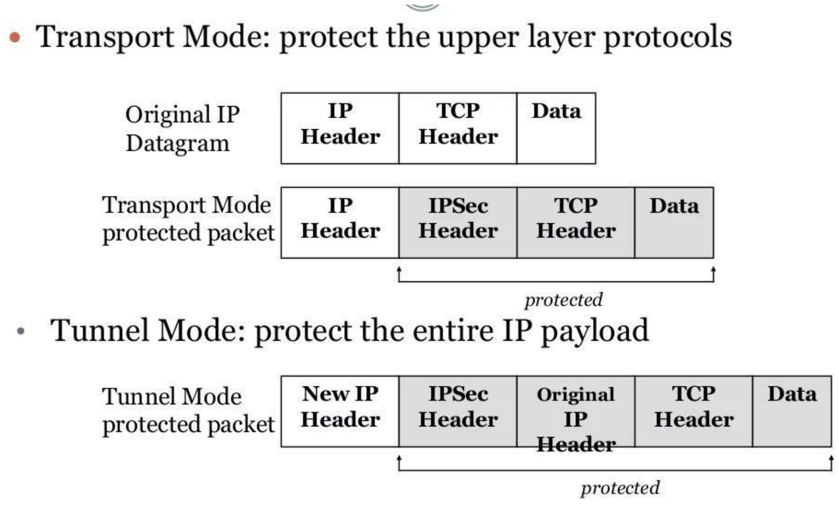
Transport mode

- Provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.
- Typically, transport mode is used for end-to-end communication between two hosts (e.g., a client and a server, or two workstations).
- When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header.
- For IPv6, the payload is the data that normally follow 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.

TUNNEL MODE :

- Tunnel mode provides protection to the entire IP packet.
- To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new outer IP packet with a new outer IP header.
- The entire original, inner, packet travels through a tunnel from one point of an IP network to another; no routers along the way are able to examine the inner IP header.
- Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security.
- Tunnel mode is used when one or both ends of a security association (SA) are 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.
- ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header.
- AH in tunnel mode authenticates the entire inner IP packet and selected portions of the outer IP header.
- Figure 3.2 and Table 3.1 summarizes transport and tunnel mode functionality.



- Transport Mode: protect the upper layer protocols
- Tunnel Mode: protect the entire IP payload

Figure 3.2 : Transport and Tunnel mode

Table 3.1 : Transport and Tunnel mode functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

➢ Transport and Tunnel Mode ESP

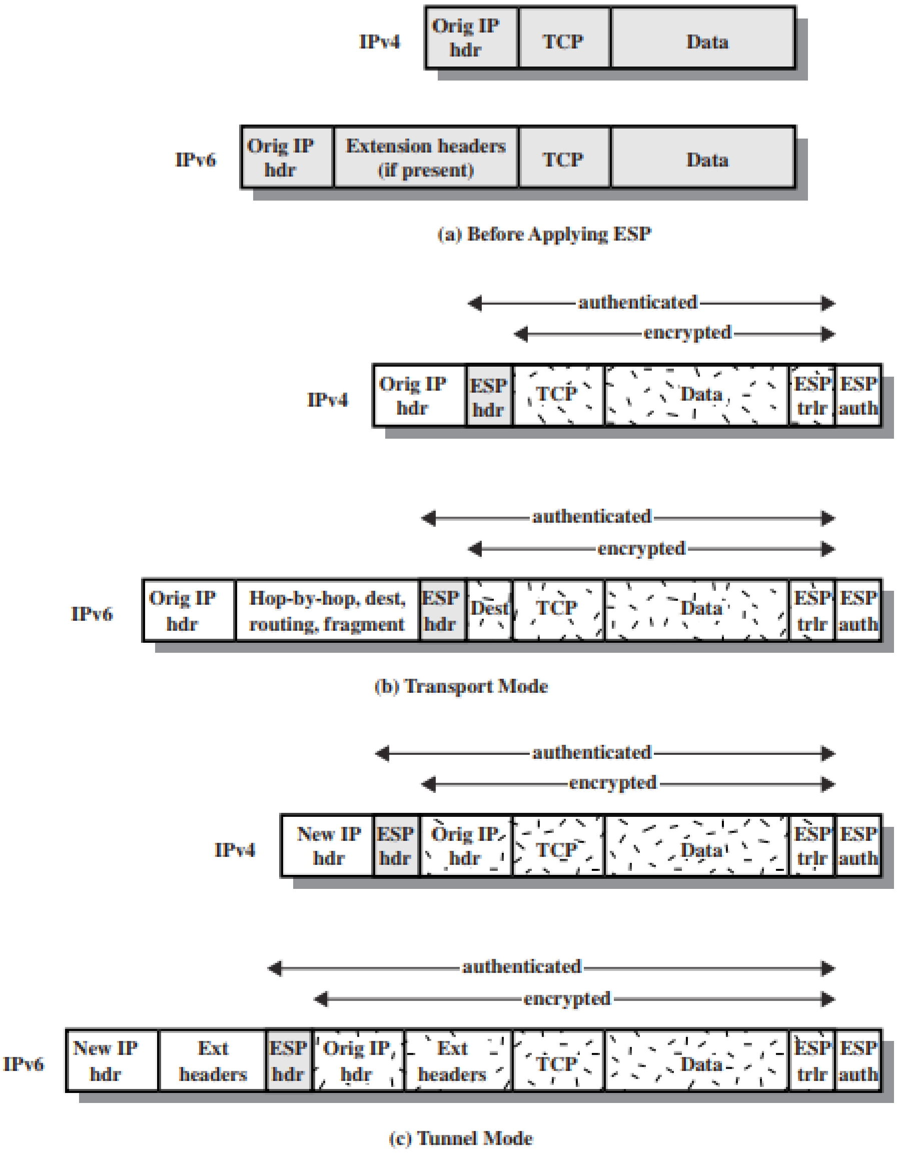


Figure 3.3 Scope of ESP Encryption and Authentication

Transport Mode ESP

- Transport mode ESP is used to encrypt and optionally authenticate the data carried by IP (e.g., a TCP segment), as shown in Figure 3.3 b.
- For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP), and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet.
- 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.
- In the context of IPv6, ESP is viewed as an end-to-end payload; that is, it is not examined or processed by intermediate routers. Therefore, the ESP header appears after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers.
- The destination options extension header could appear before or after the ESP header, depending on the semantics desired. For IPv6, encryption covers the entire transport-level segment plus the ESP trailer plus the destination options extension header if it occurs after the ESP header. Again, authentication covers the ciphertext plus the ESP header.

Tunnel Mode ESP

- Tunnel mode ESP is used to encrypt an entire IP packet (Figure 3.3 c). For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted. This method can be used to counter traffic analysis.
- 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.
- 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.

3.3 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 logical connection 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.

The concept of a security policy is applied to each IP packet that transits from a source to a destination. IPsec policy is determined primarily by the interaction of two databases, the **security association database (SAD)** and the **security policy database (SPD)**.

A security association is uniquely identified by three parameters.

- **Security Parameters Index (SPI):** A bit string assigned to this SA and having 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:** 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 field from the outer IP header indicates whether the association is an AH or ESP security association.

➢ Security Association Database

In each IPsec implementation, there is a Security Association Database that defines the parameters associated with each SA.

A security association is normally defined by the following parameters in an SAD entry.

- **Security Parameter Index**: A 32-bit value selected by the receiving end of an SA to uniquely identify the SA. In an SAD entry for an outbound SA, the SPI is used to construct the packet's AH or ESP header. In an SAD entry for an inbound SA, the SPI is used to map traffic to the appropriate SA.
- **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
- **AH Information**: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations).
- **ESP Information**: Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations).

- **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 (required for all implementations).
- **IPsec Protocol Mode**: Tunnel or transport mode.
- **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).

➢ Security Policy Database

The means by which IP traffic is related to specific SAs is the nominal Security Policy Database (SPD). An SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic.

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.

1. 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.
2. Determine the SA if any for this packet and its associated SPI.
3. Do the required IPsec processing (i.e., AH or ESP processing).

The following selectors determine an SPD entry:

- **Remote 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 (e.g., behind a firewall).
- **Local 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 (e.g., behind a firewall).
- **Next Layer Protocol**: The IP protocol header (IPv4, IPv6, or IPv6 Extension) includes a field (Protocol for IPv4, Next Header for IPv6 or IPv6 Extension) that designates the protocol operating over IP. This is an individual protocol number, ANY, or for IPv6 only, OPAQUE. If AH or ESP is used, then this IP protocol header immediately precedes the AH or ESP header in the packet.
- **Name**: A user identifier from the operating system. This is not a field in the IP or upper-layer headers but is available if IPsec is running on the same operating system as the user.
- **Local and Remote Ports:** These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

IP Traffic Processing

IPsec is executed on a packet-by-packet basis. When IPsec is implemented, each outbound IP packet is processed by the IPsec logic before transmission, and each inbound packet is processed by the IPsec logic after reception and before passing the packet contents on to the next higher layer (e.g., TCP or UDP).

OUTBOUND PACKETS Figure 3.4 highlights the main elements of IPsec processing for outbound traffic. A block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body. Then the following steps occur:

1. IPsec searches the SPD for a match to this packet.

2. If no match is found, then the packet is discarded and an error message is generated.

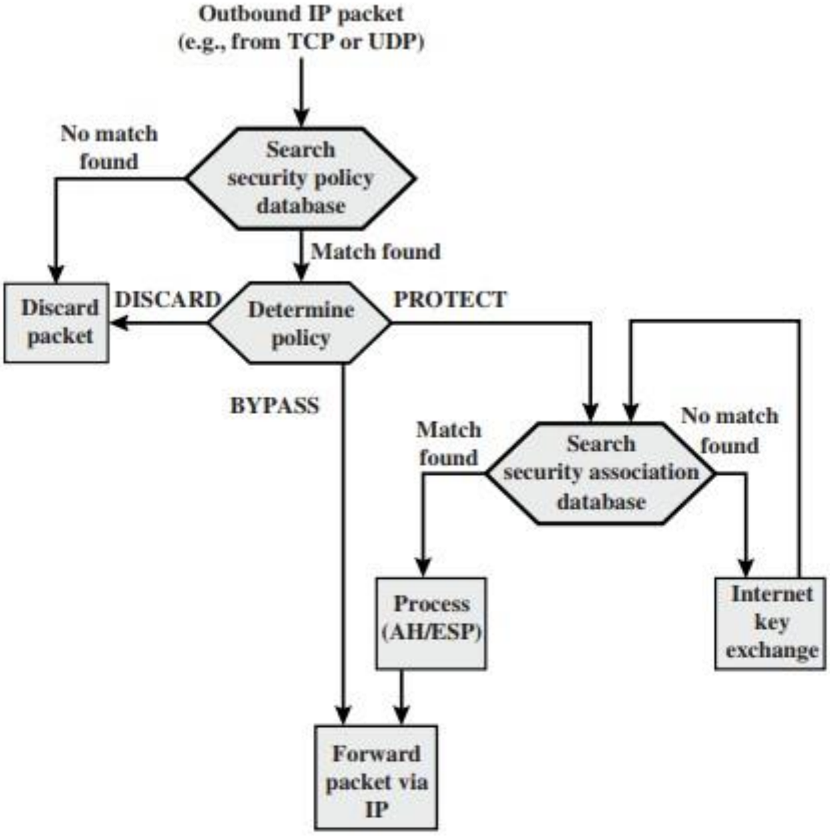


Fig 3.4 Processing model for outbound packets

3. If a match is found, further processing is determined by the first matching entry in the SPD. If the policy for this packet is DISCARD, then the packet is discarded. If the policy is BYPASS, then there is no further IPsec processing; the packet is forwarded to the network for transmission.

4. If the policy is PROTECT, then a search is made of the SAD for a matching entry. If no entry is found, then IKE is invoked to create an SA with the appropriate keys and an entry is made in the SA.

5. The matching entry in the SAD determines the processing for this packet. Either encryption, authentication, or both can be performed, and either transport or tunnel mode can be used. The packet is then forwarded to the network for transmission.

INBOUND PACKETS Figure 3.5 highlights the main elements of IPsec processing for inbound traffic. An incoming IP packet triggers the IPsec processing. The following steps occur:

- IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6)
- If the packet is unsecured, IPsec searches the SPD for a match to this packet. If the first matching entry has a policy of BYPASS, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP. If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.
- For a secured packet, IPsec searches the SAD. If no match is found, the packet is discarded. Otherwise, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP.

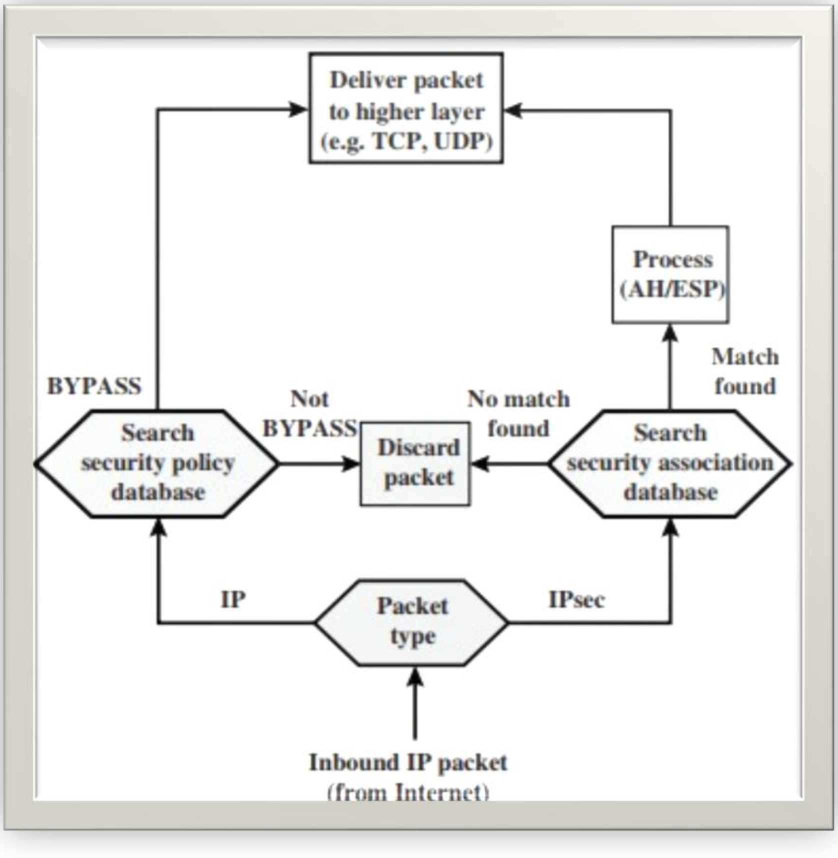
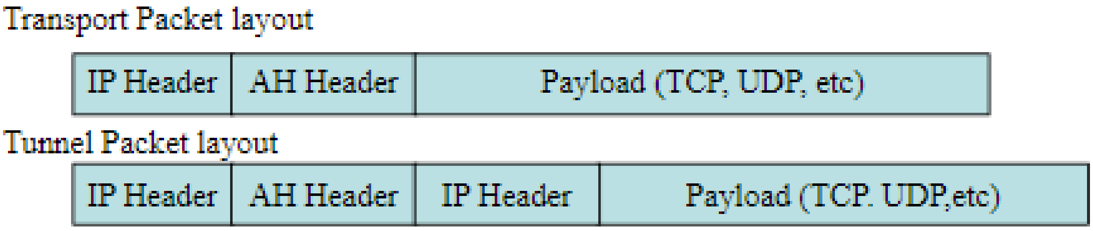

Deliver packet to higher layer (e.g. TCP, UDP)
Process (AH/ESP)
Match found
BYPASS
Not BYPASS
No match found
Search security policy database
Discard packet
Search security association database
IP
Packet type
IPsec
Inbound IP packet (from Internet)

Fig 3.5 Processing model for inbound packets

3.4 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
  - ✓ A symmetric (secret) 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 3.6 shows the fields and the position of the authentication header in transport mode.

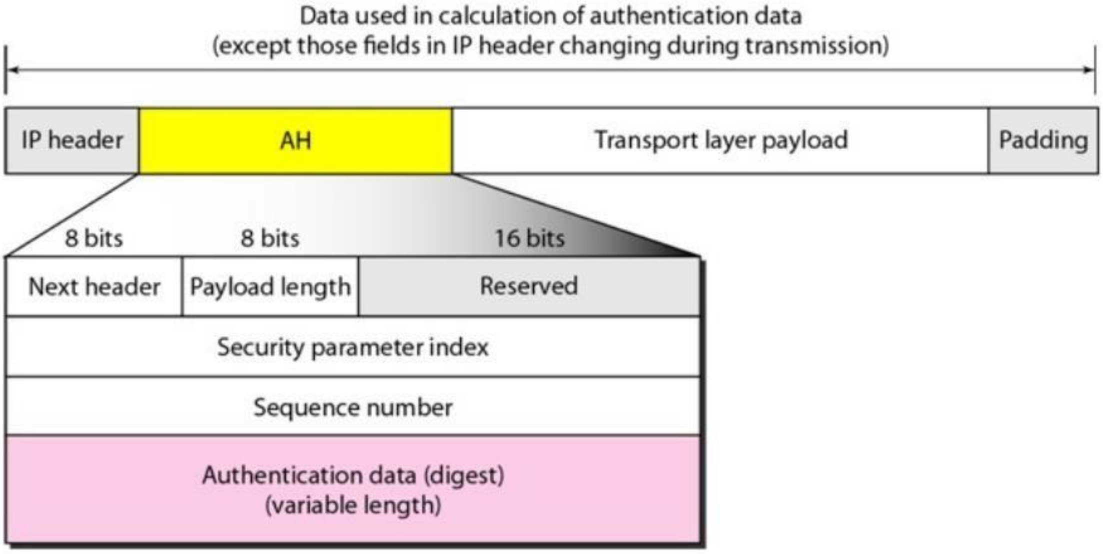


Fig 3.6 Authentication Header (AH) Protocol

- **The addition of an authentication header follows these steps.**
- An Authentication header is added to the payload with the authentication data field set to 0.
- Padding may be added to make the total length even for a particular hashing algorithm.
- Hashing is based on the total packet. Only those fields of the IP header that do not change during transmission are included in the calculation of the message digest (authentication data).
- The authentication data are inserted in the authentication header.
- The IP header is added after changing the protocol field to 51

The authentication header consists of the following fields:

- ✓ **Next header:** The 8-bit next header field defines the type of payload carried by the IP datagram(such as TCP, UDP). It copies the value of the protocol field in the IP datagram to this field.
- ✓ **Payload length:** The length of authentication header in 4-byte multiples.
- ✓ **Security parameter index:** The 32-bit security parameter index(SPI) field plays the role of a virtual circuit identifier and is same for all packets sent during a connection called Security Association. 32-bit number fixed for a session.
- ✓ **Sequence Number:** A 32-bit sequence number provides ordering information for a sequence of datagrams. The sequence number prevent a playback.
- ✓ **Authentication data:** The authentication data field is the result of applying a hash function to the entire IP datagram

3.5 ENCAPSULATION SECURITY PAYLOAD (ESP)

ESP can be used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and (limited) traffic flow confidentiality. The set of services provided depends on options selected at the time of Security Association (SA) establishment and on the location of the implementation in a network topology.

ESP can work with a variety of encryption and authentication algorithms.

ESP Format

Figure 3.7 a shows the top-level 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; this provides an anti-replay function, as discussed for AH.
- ✓ **Payload Data** (variable): This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- ✓ **Padding (0 – 255 bytes):** The purpose of this field is discussed later.
- ✓ **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).
- ✓ **Integrity Check Value (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.

Two additional fields may be present in the payload (Figure 3.7b). An initialization value (IV), or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP.

If tunnel mode is being used, then the IPsec implementation may add traffic flow confidentiality (TFC) padding after the Payload Data and before the Padding field,

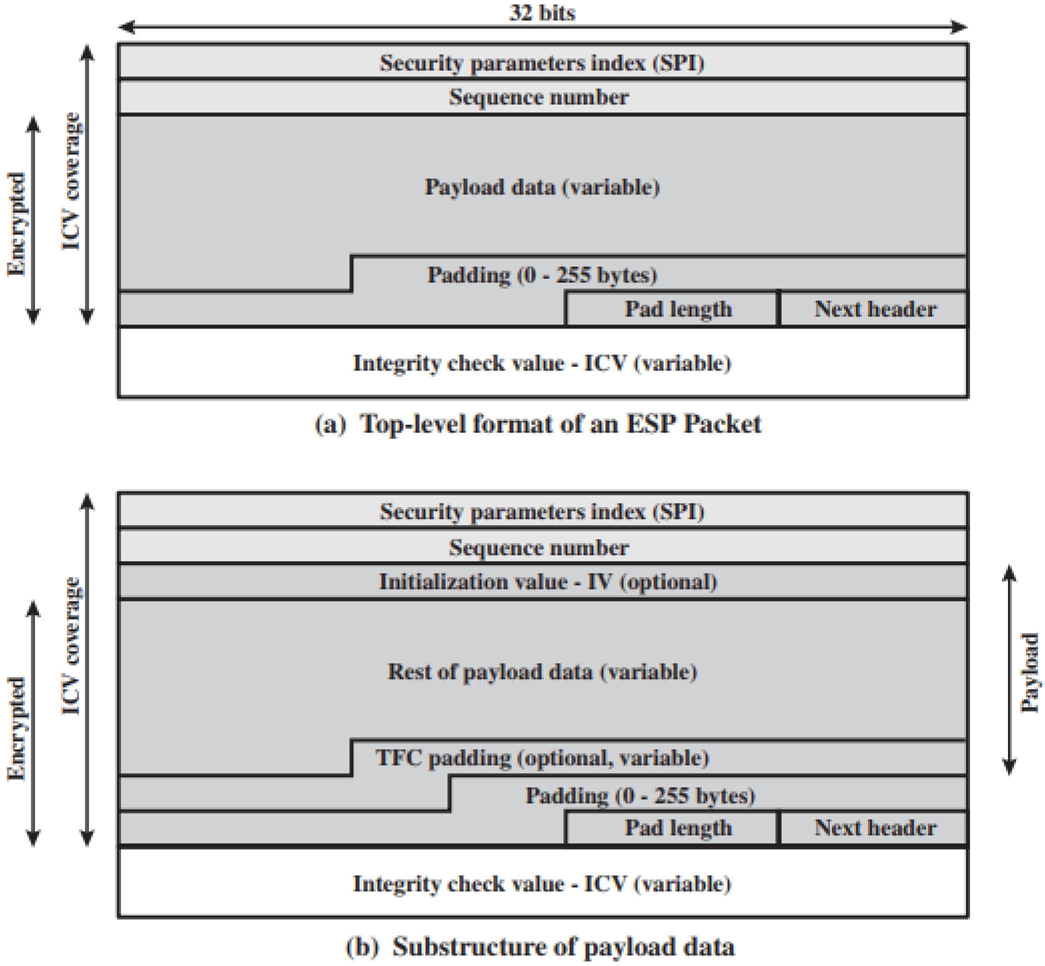

32 bits
Security parameters index (SPI)
Sequence number
Payload data (variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)
Encrypted
ICV coverage
(a) Top-level format of an ESP Packet
Security parameters index (SPI)
Sequence number
Initialization value - IV (optional)
Rest of payload data (variable)
TFC padding (optional, variable)
Padding (0 - 255 bytes)
Pad length
Next header
Integrity check value - ICV (variable)
Encrypted
ICV coverage
Payload
(b) Substructure of payload data

Fig 3.7 ESP packet format

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 these data may be carried explicitly at the beginning of the Payload Data field.
- The ICV field is optional. It is present only if the integrity service is selected and is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV.
- The ICV is computed after the encryption is performed. This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver prior to decrypting the packet, hence potentially reducing the impact of denial of service (DoS) attacks.
- It also allows for the possibility of parallel processing of packets at the receiver, i.e., decryption can take place in parallel with integrity checking.

Padding

The Padding field serves several purposes:

- If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., 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.

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.
- First, we discuss sequence number generation by the sender, and then we look at how it is processed by the recipient.
- 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 , with a default of .
- The right edge of the window represents the highest sequence number N-W+1 to N, , so far received for a valid packet. For any packet with a sequence number in the range from to that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked (Figure 3.8).
- 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.

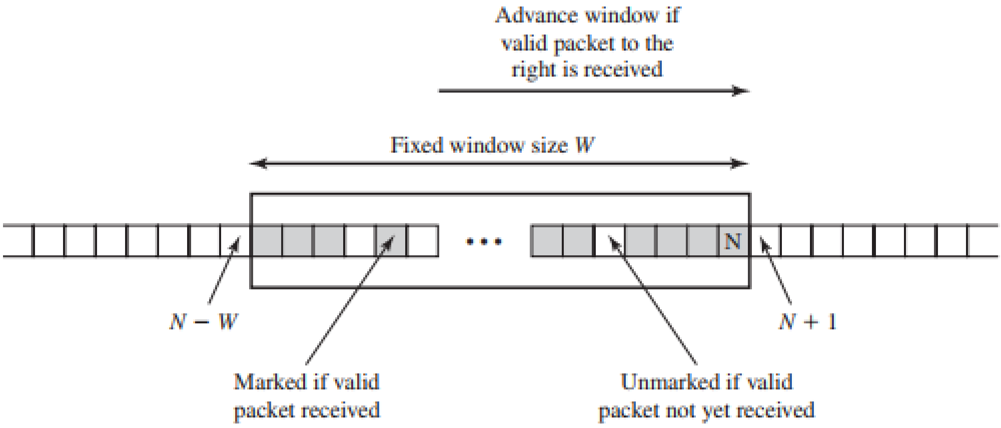

Advance window if valid packet to the right is received
Fixed window size W
N
N − W
N + 1
Marked if valid packet received
Unmarked if valid packet not yet received

Fig 3.8 Anti-Replay Mechanism

3.6 INTERNET KEY EXCHANGE

The key management portion of IPsec involves the determination and distribution of secret keys. A typical requirement is four keys for communication between two applications: transmit and receive pairs for both integrity and confidentiality. 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.

The default automated key management protocol for IPsec is referred to as ISAKMP/Oakley and consists of the following elements:

- **Oakley Key Determination Protocol:** Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats.
- **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 consists of a set

of message types that enable the use of a variety of key exchange algorithms. Oakley is the specific key exchange algorithm mandated for use with the initial version of ISAKMP.

➢ Key Determination Protocol

IKE key determination is a refinement of the Diffie-Hellman key exchange algorithm. The Diffie-Hellman involves the following interaction between users A and B. There is prior agreement on two global parameters: **q** , a large prime number; and **α** , a primitive root of **q**. A selects a random integer X_A as its private key and transmits to B its public key $Y_A = \boldsymbol{\alpha}^{X_A}$ **mod q**.

Similarly, B selects a random integer as its private key and transmits to A its public key $Y_B = \boldsymbol{\alpha}^{X_B}$ **mod q .** Each side can now compute the secret session key:

$$K = (Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q = a^{X_A X_B} \bmod q$$

The Diffie-Hellman algorithm has two attractive features:

- Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability.
- The exchange requires no pre-existing infrastructure other than an agreement on the global parameters.

However, there are a number of weaknesses to Diffie-Hellman, listed below

- It does not provide any information about the identities of the parties.
- It is subject to a man-in-the-middle attack, in which a third party C impersonates B while communicating with A and impersonates A while communicating with B. Both A and B end up negotiating a key with C, which can then listen to and pass on traffic. The man-in-the-middle attack proceeds as

1. B sends his public key in a message addressed to A.
2. The enemy (E) intercepts this message. E saves B’s public key and sends a message to A that has B’s User ID but E’s public key . This message is Y_E sent in such a way that it appears as though it was sent from B’s host system. A receives E’s message and stores E’s public key with B’s User ID. Similarly, E sends a message to B with E’s public key, purporting to come from A.

3. B computes a secret key based on B’s private key and .A computes a secret key based on A’s private key and E computes using E’s secret key and and computers using and .

4. From now on, E is able to relay messages from A to B and from B to A, appropriately changing their encipherment en route in such a way that neither A nor B will know that they share their communication with E.

➢ FEATURES OF IKE KEY DETERMINATION

The IKE key determination algorithm is characterized by five important features:

1. It employs a mechanism known as cookies to thwart clogging attacks.

2. It enables the two parties to negotiate a group; this, in essence, specifies the global parameters of the Diffie-Hellman key exchange.

3. It uses nonces to ensure against replay attacks.

4. It enables the exchange of Diffie-Hellman public key values.

5. It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

- In clogging attack, an opponent forges the source address of a legitimate user and sends a public DiffieHellman key to the victim. The victim then performs a modular exponentiation to compute the secret key.
- Repeated messages of this type can clog the victim's system with useless work. The cookie exchange requires that each side send a pseudorandom number, the cookie, in the initial message, which the other side acknowledges.
- This acknowledgment must be repeated in the first message of the Diffie-Hellman key exchange. If the source address was forged, the opponent gets no answer. Thus, an opponent can only force a user to generate acknowledgments and not to perform the Diffie-Hellman calculation.

IKE mandates that cookie generation satisfy three basic requirements:

1. The cookie must depend on the specific parties. This prevents an attacker from obtaining a cookie using a real IP address and UDP port and then using it to swamp the victim with requests from randomly chosen IP addresses or ports.

2. It must not be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity. This implies that the issuing entity will use local secret information in the generation and subsequent verification of a cookie. It must not be possible to deduce this secret information from any particular cookie.

3. The cookie generation and verification methods must be fast to thwart attacks intended to sabotage processor resources.

Three different authentication methods can be used with IKE key determination:

• **Digital signatures**: The exchange is authenticated by signing a mutually obtainable hash; each party encrypts the hash with its private key. The hash is generated over important parameters, such as user IDs and nonces.

• **Public-key encryption**: The exchange is authenticated by encrypting parameters such as IDs and nonces with the sender's private key.

• **Symmetric-key encryption**: A key derived by some out-of-band mechanism can be used to authenticate the exchange by symmetric encryption of exchange parameters.

➢ Header and Payload Formats

IKE defines procedures and packet formats to establish, negotiate, modify, and delete security associations. As part of SA establishment, IKE defines payloads for exchanging key generation and authentication data. These payload formats provide a consistent framework independent of the specific key exchange protocol, encryption algorithm, and authentication mechanism.

IKE HEADER FORMAT An IKE message consists of an IKE header followed by one or more payloads. All of this is carried in a transport protocol. The specification dictates that implementations must support the use of UDP for the transport protocol.

Figure 3.9 shows the header format for an IKE message. It consists of the following fields.

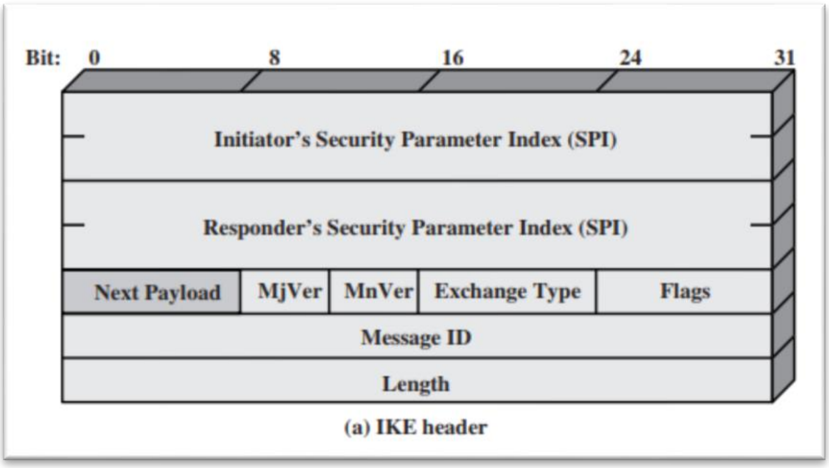

Bit: 0
8
16
24
31
Initiator's Security Parameter Index (SPI)
Responder's Security Parameter Index (SPI)
Next Payload
MjVer
MnVer
Exchange Type
Flags
Message ID
Length
(a) IKE header

Fig 3.9 IKE header format

• **Initiator SPI (64 bits)**: A value chosen by the initiator to identify a unique IKE security association (SA).

• **Responder SPI (64 bits)**: A value chosen by the responder to identify a unique IKE SA.
• **Next Payload (8 bits):** Indicates the type of the first payload in the message;
• **Major Version (4 bits):** Indicates major version of IKE in use.
• **Minor Version (4 bits):** Indicates minor version in use.

- **Exchange Type (8 bits)**: Indicates the type of exchange;

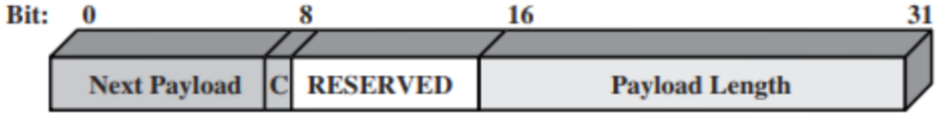
- **Flags (8 bits):** Indicates specific options set for this IKE exchange. Three bits are defined so far. The initiator bit indicates whether this packet is sent by the SA initiator. The version bit indicates whether the transmitter is capable of using a higher major version number than the one currently indicated. The response bit indicates whether this is a response to a message containing the same message ID.

- **Message ID (32 bits):** Used to control retransmission of lost packets and matching of requests and responses.

- **Length (32 bits):** Length of total message (header plus all payloads) in octets.

➢ IKE PAYLOAD TYPES

All IKE payloads begin with the same generic payload header shown in Figure below.



(b) Generic Pavload header

- The Next Payload field has a value of 0 if this is the last payload in the message; otherwise its value is the type of the next payload.
- The Payload Length field indicates the length in octets of this payload, including the generic payload header.
- The critical bit is 0 if the sender wants the recipient to skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload. It is set to 1 if the sender wants the recipient to reject this entire message if it does not understand the payload type.

The payload types defined for IKE are listed below:

- ➢ **SA payload** is used to begin the establishment of an SA. The payload has a complex, hierarchical structure. The payload may contain multiple proposals. Each proposal may contain multiple protocols. Each protocol may contain multiple transforms. And each transform may contain multiple attributes. These elements are formatted as substructures within the payload as follows.
  - ✓ **Proposal:** This substructure includes a proposal number, a protocol ID (AH, ESP, or IKE), an indicator of the number of transforms, and then a transform substructure. If more than one protocol is to be included in a proposal, then there is a subsequent proposal substructure with the same proposal number.

- ✓ **Transform:** Different protocols support different transform types. The transforms are used primarily to define cryptographic algorithms to be used with a particular protocol.
- ✓ **Attribute:** Each transform may include attributes that modify or complete the specification of the transform. An example is key length.

- **Key Exchange payload** can be used for a variety of key exchange techniques, including Oakley, Diffie-Hellman, and the RSA-based key exchange used by PGP. The Key Exchange data field contains the data required to generate a session key and is dependent on the key exchange algorithm used.
- **Identification payload** is used to determine the identity of communicating peers and may be used for determining authenticity of information. Typically the ID Data field will contain an IPv4 or IPv6 address.
- **Certificate payload** transfers a public-key certificate. The Certificate Encoding field indicates the type of certificate or certificate-related information.
- **Certificate Request payload** At any point in an IKE exchange, the sender may include a to request the certificate of the other communicating entity. The payload may list more than one certificate type that is acceptable and more than one certificate authority that is acceptable.
- **Authentication payload** contains data used for message authentication purposes. The authentication method types defined are RSA digital signature, shared-key message integrity code, and DSS digital signature.
- **Nonce payload** contains random data used to guarantee liveness during an exchange and to protect against replay attacks.
- **Notify payload** contains either error or status information associated with this SA or this SA negotiation.
- **Delete payload** indicates one or more SAs that the sender has deleted from its database and that therefore are no longer valid.
- **Vendor ID payload** contains a vendor-defined constant. The constant is used by vendors to identify and recognize remote instances of their implementations. This mechanism allows a vendor to experiment with new features while maintaining backward compatibility.
- **Traffic Selector payload** allows peers to identify packet flows for processing by IPsec services.
- **Encrypted payload** contains other payloads in encrypted form. The encrypted payload format is similar to that of ESP. It may include an IV if the encryption algorithm requires it and an ICV if authentication is selected.
- **Configuration payload** is used to exchange configuration information between IKE peers.
- **Extensible Authentication Protocol (EAP)** payload allows IKE SAs to be authenticated using EAP.