CRYPTOGRAPHY AND NETWORK SECURITY
UNIT-IV

USER AUTHENTICATION

REMOTE USER AUTHENTICATION PRINCIPLES:

The process of verifying an identity claimed by or for a system entity. An authentication process consists of two steps:

- *Identification step:* Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
- *Verification step*: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

There are four general means of authenticating a user's identity, which can be used alone or incombination:

- **Something the individual knows:** Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions.
- **Something the individual possesses:** Examples include cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- **Something the individual is (static biometrics):** Examples include recognition by fingerprint, retina, and face.
- **Something the individual does (dynamic biometrics):** Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

For network-based user authentication, the most important methods involve cryptographic keys and something the individual knows, such as a password.

Principle 1: Mutual Authentication:

Mutual authentication protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

Principle 2: One-Way Authentication:

Typically, the recipient wants some assurance that the message is from the alleged sender.

REMOTE USER AUTHENTICATION USING SYMMETRIC ENCRYPTION:

Mutual Authentication

Protocol of Needham and Schroeder for secret key distribution using a KDC can be summarized as follows.

Protocol-1

1. $A \rightarrow KDC: \quad ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A])$
4. $B \rightarrow A: \quad E(K_s, N_2)$
5. $A \rightarrow B: \quad E(K_s, f(N_2))$

- Secret keys Ka and Kb are shared between A and the KDC and B and the KDC, respectively.
- The purpose of the protocol is to distribute securely a session key Ks to A and B. A securely acquires a new session key in step 2.
- The message in step 3 can be decrypted, and hence understood, only by B.
- Step 4 reflects B's knowledge of Ks, and step 5 assures B of A's knowledge of ks and assures B that this is a fresh message because of the use of the nonce N2 .

Steps 4 and 5, the protocol is still vulnerable to a form of replay attack. Improved version onprotocol-1 is as follows:

Protocol-2

Protocol 2 Includes the addition of a timestamp to steps 2 and 3. T is a timestamp that assures Aand B that the session key has only just been generated.

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A: \quad E(K_s, N_1)$
5. $A \rightarrow B: \quad E(K_s, f(N_1))$

One-Way Authentication

Using symmetric encryption, the decentralized key distribution scenario illustrated for a message with content, the sequence is as follows:

1. $A \rightarrow KDC: \quad ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A]) \| E(K_s, M)$

KERBEROS:

Kerberos is an authentication service developed as part of Project Athena at MIT. The problem that Kerberos addresses is this:

Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to identify its users correctly to network services.

In particular, the following three threats exist:

- A user may gain access to a particular workstation and pretend to be another user operating

from that workstation.

- A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
- A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations

In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access. Rather than building in elaborate authentication protocols at each server, Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

Requirements of Kerberos

Kerberos listed the following requirements.

- *Secure*: A network eavesdropper should not be able to obtain the necessary information to impersonate a user.
- *Reliable*: Kerberos should be highly reliable and should employ a distributed server architecture with one system able to back up another.
- *Transparent*: Ideally, the user should not be aware that authentication is taking place beyond the requirement to enter a password.
- *Scalable*: The system should be capable of supporting large numbers of clients and servers.

Kerberos realm

A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers collectively called as Kerberos realm which requires the following:

1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.

2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.

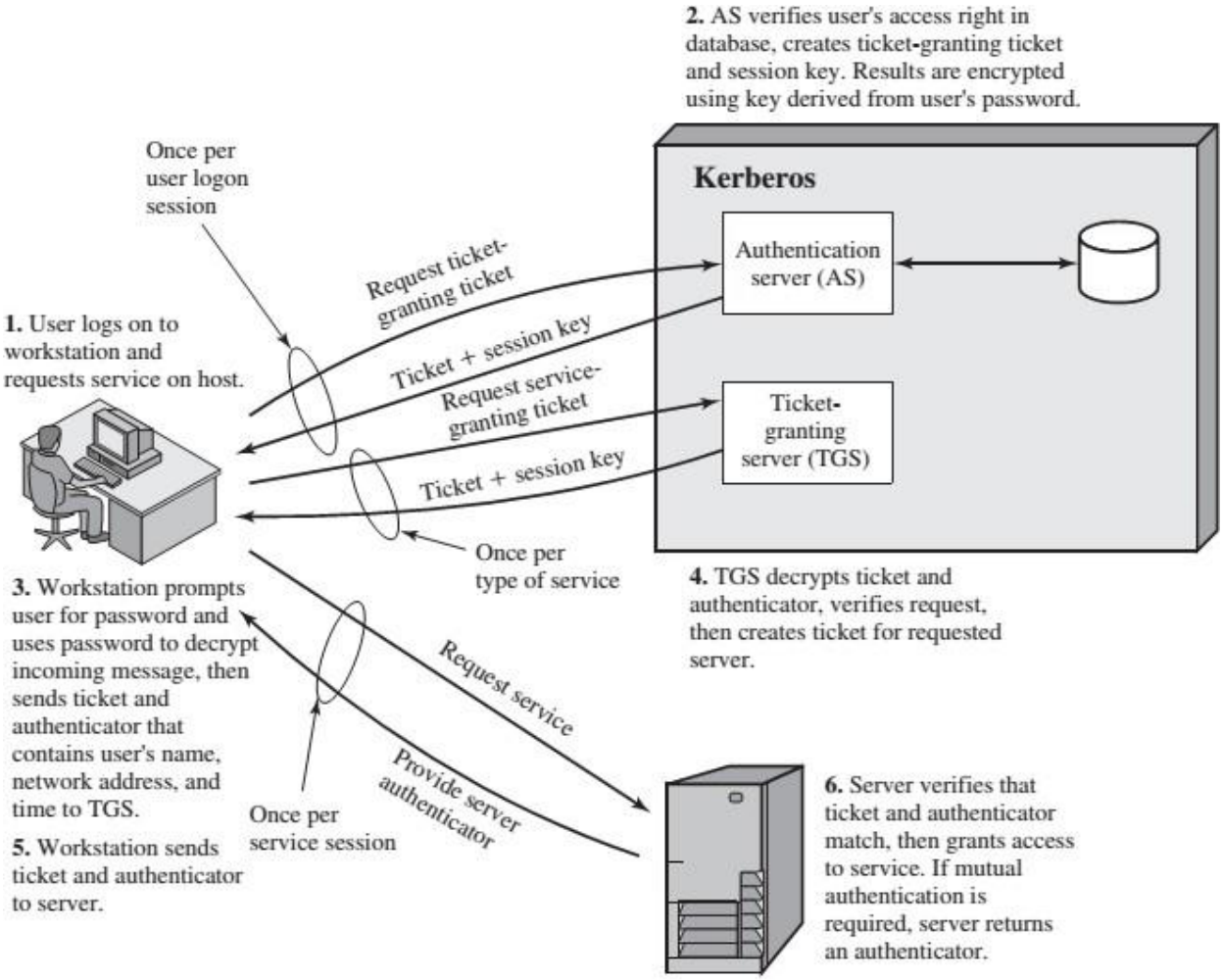


Figure 15.1 Overview of Kerberos

Details of Kerberos working

Once per user logon session:

(1) $C \rightarrow AS: \quad ID_C \| ID_{tgs}$

(2) $AS \rightarrow C: \quad E(K_c, Ticket_{tgs})$

Once per type of service:

(3) $C \rightarrow TGS: \quad ID_C \| ID_V \| Ticket_{tgs}$

(4) $TGS \rightarrow C: \quad Ticket_v$

Once per service session:

(5) $C \rightarrow V: \quad ID_C \| Ticket_v$

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \| AD_C \| ID_{tgs} \| TS_1 \| Lifetime_1])$$
$$Ticket_v = E(K_v, [ID_C \| AD_C \| ID_v \| TS_2 \| Lifetime_2])$$

Notations used are as follows:

Message (1)	Client requests ticket-granting ticket.
ID_C	Tells AS identity of user from this client.
ID_{tgs}	Tells AS that user requests access to TGS.
TS_1	Allows AS to verify that client's clock is synchronized with that of AS.
Message (2)	AS returns ticket-granting ticket.
K_c	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2).
$K_{c,tgs}$	Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}	Confirms that this ticket is for the TGS.
TS_2	Informs client of time this ticket was issued.
$Lifetime_2$	Informs client of the lifetime of this ticket.
$Ticket_{tgs}$	Ticket to be used by client to access TGS.

Communication between two Realm

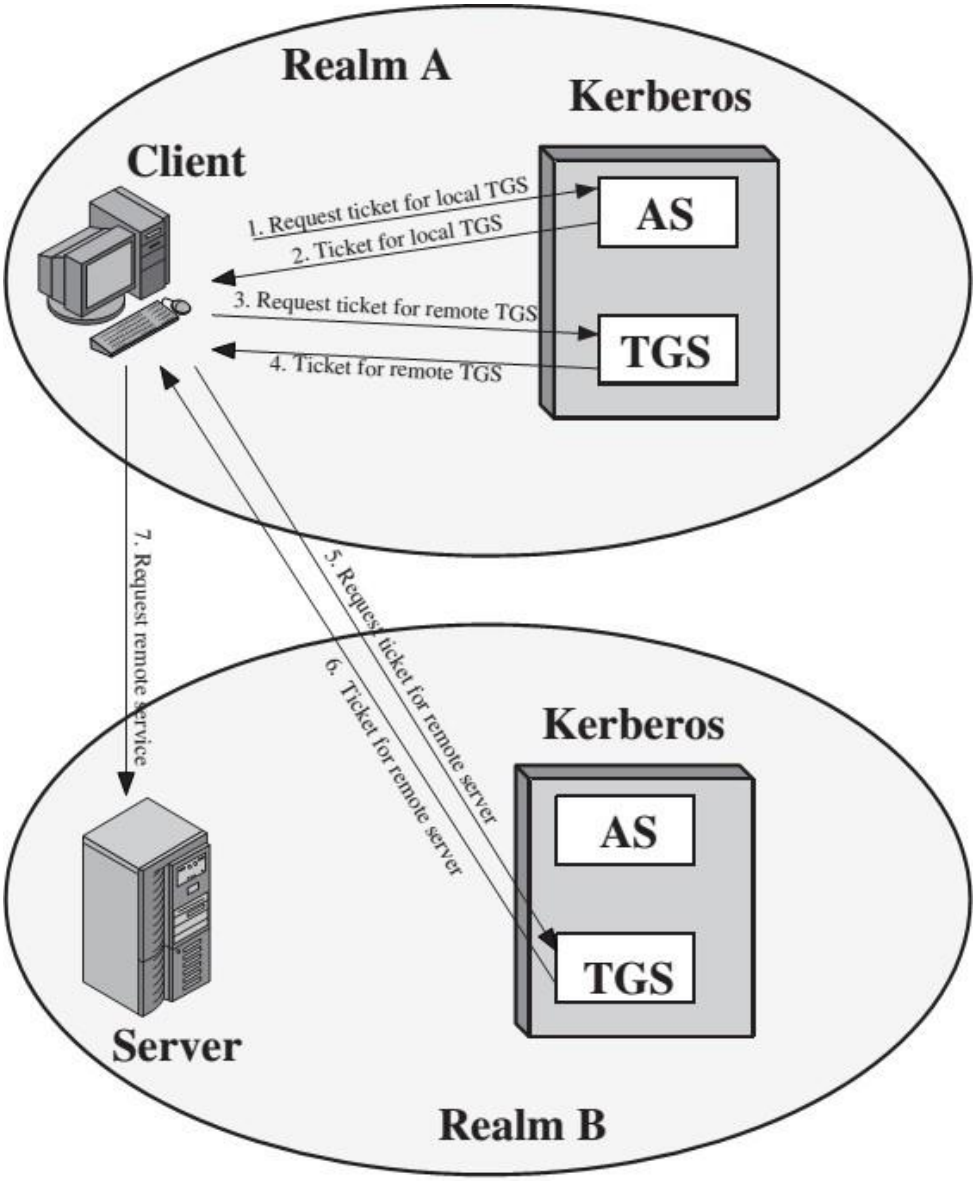


Figure 15.2 Request for Service in Another Realm

Electronic Mail Security:

Email is one of the most widely used and regarded network services. Currently message contents are not secure, may be inspected either in transit or by suitably privileged users on destination system.

Email Security Enhancements:

- confidentiality
    - protection from disclosure
- authentication
    - of sender of message
- message integrity
    - protection from modification
- non-repudiation of origin
    - protection from denial by sender

Pretty Good Privacy (PGP):

- Provides a confidentiality and authentication service that can be used for electronic mail and file storage applications
- Developed by Phil Zimmermann
    - Selected the best available cryptographic algorithms as building blocks
    - Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands
    - Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks
    - Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP

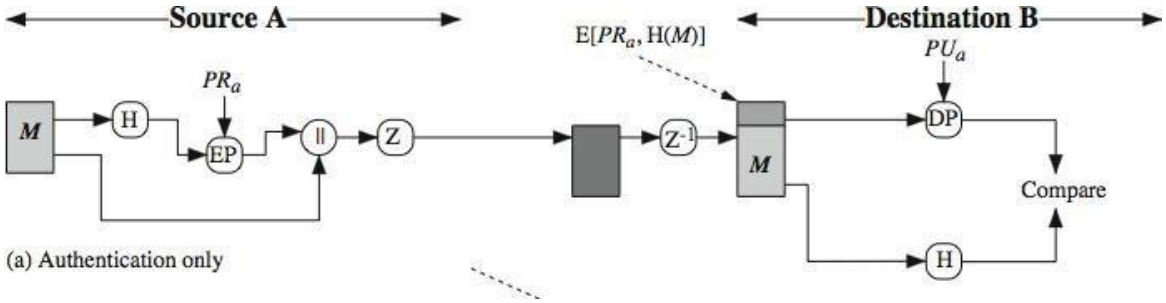
PGP Growth:

- It is available free worldwide in versions that run on a variety of platforms
- The commercial version satisfies users who want a product that comes with vendor support
- It is based on algorithms that have survived extensive public review and are considered extremely secure
- It has a wide range of applicability
- It was not developed by, nor is it controlled by, any governmental or standards organization
- Is now on an Internet standards track, however it still has an aura of an antiestablishment endeavor

PGP Operation – Authentication:

1. sender creates a message
2. SHA-1 used to generate 160-bit hash code of message
3. hash code is encrypted with RSA using the sender's private key, and result is attached to message
4. receiver uses RSA or DSS with sender's public key to decrypt and recover hash code

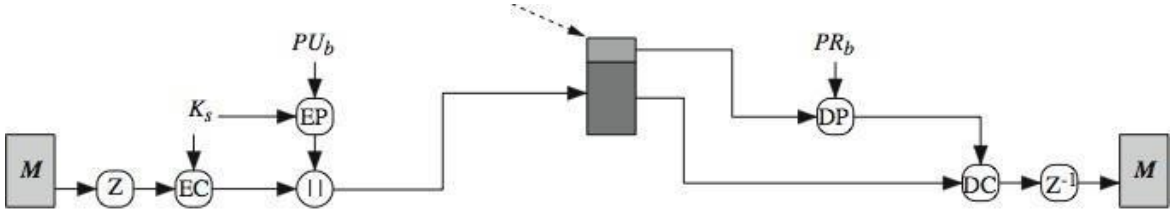
5. receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic



(a) Authentication only

PGP Operation – Confidentiality:

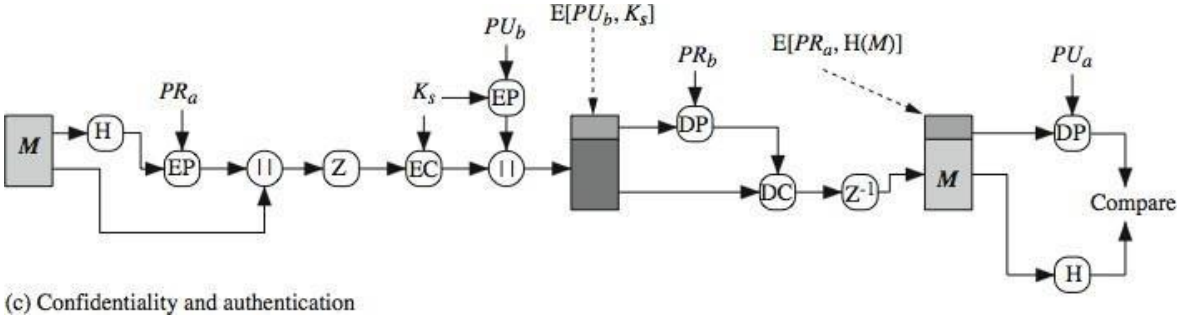
1. sender generates message and random 128-bit number to be used as session key for this message only
2. message is encrypted, using CAST-128 / IDEA/3DES with session key
3. session key is encrypted using RSA with recipient's public key, then attached to message
4. receiver uses RSA with its private key to decrypt and recover session key
5. session key is used to decrypt message



(b) Confidentiality only

PGP Operation – Confidentiality & Authentication:

- uses both services on same message
  - create signature & attach to message
  - encrypt both message & signature
  - attach RSA encrypted session key



(c) Confidentiality and authentication

PGP Operation – Compression:

- by default, PGP compresses message after signing but before encrypting
    - so can store uncompressed message & signature for later verification
    - & because compression is non deterministic
- uses ZIP compression algorithm

PGP Operation – Email Compatibility:

- when using PGP will have binary data to send (encrypted message etc)
- however, email was designed only for text
- hence PGP must encode raw binary data into printable ASCII characters
- uses radix-64 algorithm
    - maps 3 bytes to 4 printable chars
    - also appends a CRC
- PGP also segments messages if too big

S/MIME (Secure/Multipurpose Internet Mail Extensions):

- It is a security enhancement to MIME Internet e-mail format standard based on technology from RSA Data Security
- Defined in:
    - RFCs 3370, 3850, 3851, 3852
- S/MIME support in many mail agents
    - eg MS Outlook, Mozilla, Mac Mail etc
- To understand S/MIME, we need first to have a general understanding of the underlying e-mail format that it uses, namely MIME. We have to learn about RFC5322(internet Message Format)

RFC 5322:

- Defines a format for text messages that are sent using electronic mail
- Messages are viewed as having an envelope and contents
    - The envelope contains whatever information is needed to accomplish transmission and delivery
    - The contents compose the object to be delivered to the recipient
    - RFC 5322 standard applies only to the contents
- The content standard includes a set of header fieldsthat may be used by the mail system to create the envelope

Multipurpose Internet Mail Extensions (MIME):

- An extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP)
- Is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations
- The specification is provided in RFCs 2045 through 2049

The MIME specification includes the following elements.

1. **Five new message header fields** are defined, which may be included in an RFC 5322 header. These fields provide information about the body of the message.
2. A number of content formats are defined, thus standardizing representations that support multimediaelectronicmail.
3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

The Five Header Fields Defined in MIME:

The five header fields defined in MIME are

- **MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- **Content-Type:** Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.
- **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to representthe body of the message in a way that is acceptable for mail transport.
- **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description:** A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

S/MIME Functionality:

S/MIME provides the following functions.

- **Enveloped data:** This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
- **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signatureare then encoded using base64 encoding. A signed data message can only be viewed by a recipientwith S/MIME capability.
- **Clear-signed data:** As with signed data, a digital signature of the content is formed. However,in this case, only the digital signature is encoded using base64. As a result, recipients withoutS/MIME capability can view the message content, although they cannot verify the signature.
- **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

S/MIME Messages:

- S/MIME secures a MIME entity with a signature, encryption, or both.
- forming a MIME wrapped **Public-Key Cryptography Standards(**PKCS) object
- have a range of content-types:
    - enveloped data
    - signed data
    - clear-signed data
    - registration request
    - certificate only message

S/MIME Certificate Processing:

- S/MIME uses public-key certificates that conform to version 3 of X.509
- The key-management scheme used by S/MIME is in some ways a hybrid betweena strict X.509 certification hierarchy and PGP's web of trust
- S/MIME managers and/or users must configure each client with a list of trusted keys and with certificate revocation lists
    - The responsibility is local for maintaining the certificates needed to verify incoming signatures and to encrypt outgoing messages
- The certificates are signed by certification authorities

IP SECURITY OVERVIEW:

IPSec is an Internet Engineering Task Force (IETF) standard suite of protocols that provides data authentication, integrity, and confidentiality as data is transferred between communication points across IP networks. IPSec provides data security at the IP packet level. A packet is a data bundle that is organized for transmission across a network, and it includes a header and payload (the data in the packet). IPSec emerged as a viable network security standard because enterprises wanted to ensure that data could be securely transmitted over the Internet. IPSec protects against possible security exposures by protecting data whilein transit.

IPSEC SECURITY FEATURES:

IPSec is the most secure method commercially available for connecting network sites. IPSec wasdesigned to provide the following security features when transferring packets across networks:

- Authentication: Verifies that the packet received is actually from the claimed sender.
- Integrity: Ensures that the contents of the packet did not change in transit.
- Confidentiality: Conceals the message content through encryption.

IPSEC ELEMENTS:

IPSec contains the following elements:

- **Encapsulating Security Payload (ESP):** Provides confidentiality, authentication, and integrity.
- **Authentication Header (AH):** Provides authentication and integrity.
- **Internet Key Exchange (IKE):** Provides key management and Security Association (SA) management.

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 the following:

► Secure branch office connectivity over the Internet

► Secure remote access over the Internet

► **Establishing extranet and intranet connectivity with partners**: IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing akey exchange mechanism.

► **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security.

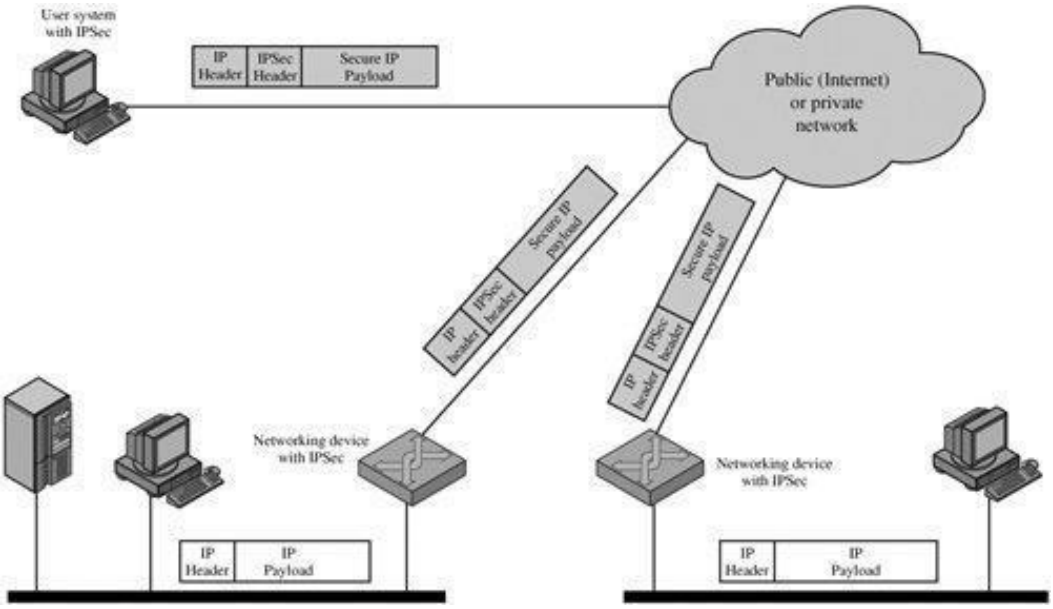


Figure. An IP Security Scenario

BENEFITS OF IPSEC:

- IPSec provides strong security within and across the LANs.
- Firewall uses IPSec to restrict all those incoming packets which are not using IP. Since
- firewall is the only way to enter into an organization, restricted packets cannot enter.
- 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. Even if IPSec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPSec can be transparent to end users.
- IPSec can provide security for individual users if needed.

IP SECURITY ARCHITECTURE:

Mainly the IPSec is constituted by three major components.

- ► IPSec Documents
- ► IPSec Services
- ► Security Associations(SA)

IPSec Documents:

The IPSec specification consists of numerous documents. The most important of these, issued in November of 1998, are RFCs 2401, 2402, 2406, and 2408:

- ► RFC 2401: An overview of a security architecture
- ► RFC 2402: Description of a packet authentication extension to IPv4 and IPv6
- ► RFC 2406: Description of a packet encryption extension to IPv4 and IPv6
- ► RFC 2408: Specification of key management capabilities

The header for authentication is known as the Authentication header(AH); that for encryption is known as the **Encapsulating Security Payload (ESP)** header.

The documents are divided into seven groups, as depicted in Figure

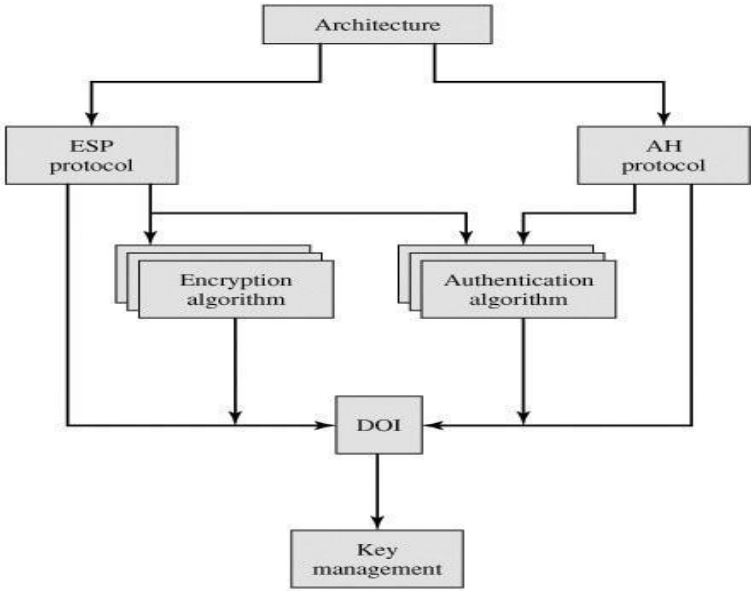


Figure. IPSec Document Overview

- **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPSectechnology
- .**Encapsulating Security Payload (ESP):** Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.
- **Authentication Header (AH):** Covers the packet format and general issues related to theuse of AH for packet authentication.
- **Encryption Algorithm:** A set of documents that describe how various encryption algorithms are used for ESP.
- **Authentication Algorithm:** A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.
- **Key Management:** Documents that describe key management schemes.
- **Domain of Interpretation (DOI):** Contains values needed for the other documents to relate to each other. These include identifiers for approvedencryptionand authenticationalgorithms, as well as operational parameters such as key lifetime.

IPSec Services:

IPSec provides security services at the IP layer by selecting requiredsecurityprotocols, algorithms and cryptographickeys as per the services requested.

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).**

The services are

- ► Access control
- ► Connectionless integrity
- ► Data origin authentication
- ► Rejection of replayed packets
- ► Confidentiality
- ► Limited traffic flow confidentiality

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

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 notboth.

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. SPI is located in AH and ESP headers. SPI enables the receiving system under which the packet is to process.
- **IP Destination Address:** It is the end point address of SA which can be end user system or a network system.
- **Security Protocol Identifier:** security protocol identifier indicates whether the associationsis an AH or ESP.

SA Parameters:

The implementation of IPSec contain SA database which identifies the parameters related to SA.

- **Sequence Number Counter**: A 32-bit value usedto generate the Sequence Number field in AHor 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.
- **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 or terminated.

- **IPSec Protocol Mode:** This parameter represents the type of mode used for IPSec implementation. The mode may be a Tunnel or transport.

- **Path MTU:** Any observed path maximum transmission unit (maximum size of a packet thatcan be transmitted).

SA Selectors:

- IPSec provides flexibility in providing services to the users according to their needs. For this purpose, SA's are used. Different combinations of SA's can give different user configurations.IPSec is also capable of differentiating traffic i.e., which traffic is allowed to pass and which traffic should be forwarded the IPSec protection. The property of IPSec requires the traffic tobe associated with a security association. To associate a particular SA to IP traffic IPSec maintains a database called Security Policy Database(SPD).
- SPD is table entries which maps a set of IP traffic to a single or more SAs.
- Selectors are basically used to define policy that specifies which packet should be forwardedand which packet should be rejected to filter outgoing traffic.

A sequence of steps is performed on the outgoing traffic,

- Compare the values of the appropriate fields in the packet (the selector fields) against the SPDto 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. **Security Parameter Index(SPI**) is one of the fields of IPSec header which is a unique identifier to identify a security association.
- Do the required IPSec processing (i.e., 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.
- **Source IP Address:** This may be a single IP address, an enumerated list or range of addresses.
- **UserID:** 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.
- **Data Sensitivity Level:** Used for systems providing information flow security (e.g., Secret or Unclassified).
- **Transport Layer Protocol:** 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.
- **Source and Destination Ports:** These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

Transport and Tunnel Modes:

- Both AH and ESP support two modes of use: **transport and tunnel mode**.
- The operation of these two modes is best understood in the context of a descri ption of AH and ESP.

Transport Mode:

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Transport mode is used for end-to-end communication between two hosts. ESP in transport mode encrypts and optionallyauthenticates 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, or 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 an SA are a security gateway, such as a firewallor router that implements IPSec. ESP in tunnel mode encrypts and optionally authenticates the entireinner 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.

AUTHENTICATION HEADER(AH):

The Authentication Header provides support for data integrity and authentication of IP packets. Dataintegrity service insures that data inside IP packets is not altered during the transit. The authentication feature enables an end system to authenticate the user or application and filter traffic accordingly. It also prevents the **address spoofing attacks** (A technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP **address** indicating that the message is coming from a trusted host).Authentication is based on the use of a message authentication code (MAC)i.e.; two communication parties must share a secret key.

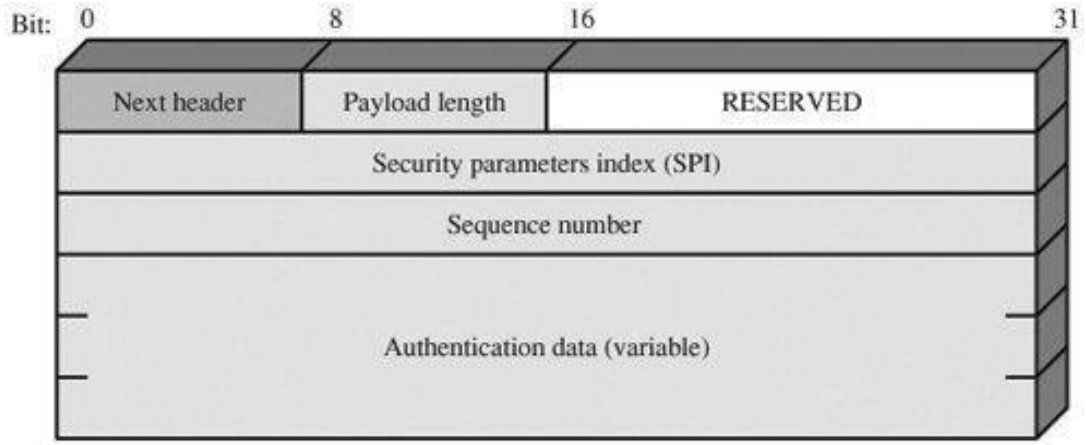


Figure. IPSec Authentication Header

The Authentication Header consists of the following fields

1. **Next Header (8 bits):** Identifies the type of header that immediately following the AH.
2. **Payload Length:** Length of Authentication Header in 32-bit words.
3. **Reserved (16 bits):** For future use.
4. **Security Parameters Index (32 bits):** Identifies a security association.
5. **Sequence Number (32 bits):** A monotonically increasing counter value.
6. **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that 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 intendeddestination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The **Sequence Number** fieldis designed to stop such attacks.

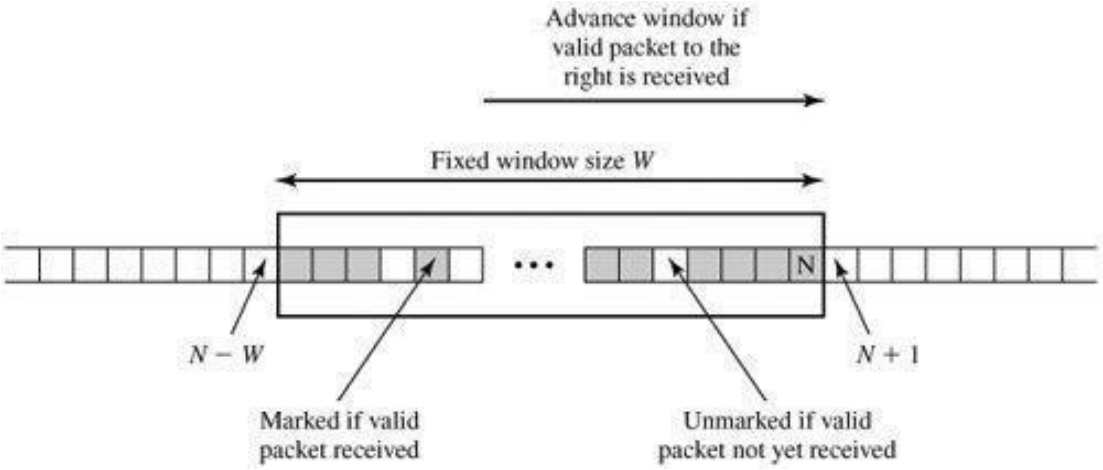


Figure. Antireplay Mechanism

When a new SA is established, the senderinitializes 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. 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 (i.e., properly authenticated), the corresponding slot in the windowis marked (Figure).

Inbound processing proceeds as follows when a packet is received:

1. If the received packet falls within the windowand is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
2. 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.
3. If the received packet is to the left of the window, or if authentication fails, the packet is discarded; this is an auditable event.

Integrity Check Value:

The Authentication Data field holds a value referred to as the Integrity Check Value. The ICV is amessage authentication code or a truncated version of a code produced by a MAC algorithm.

Transport and Tunnel Modes:

There are two ways in which the IPSec authenticationservice can be used. In one case, **authenticationis provided directly** between a server and client workstations; the workstation can be either on the same network as the server or on an external network. As long as the workstation and the server sharea protected secret key, the authentication process is secure. This case uses a **transport mode SA**. In the other case, a **remote workstation authenticates itself to the corporate firewall**, either for accessto the entire internal network or because the requested server does not support the authentication feature. This case uses a **tunnel mode SA**.

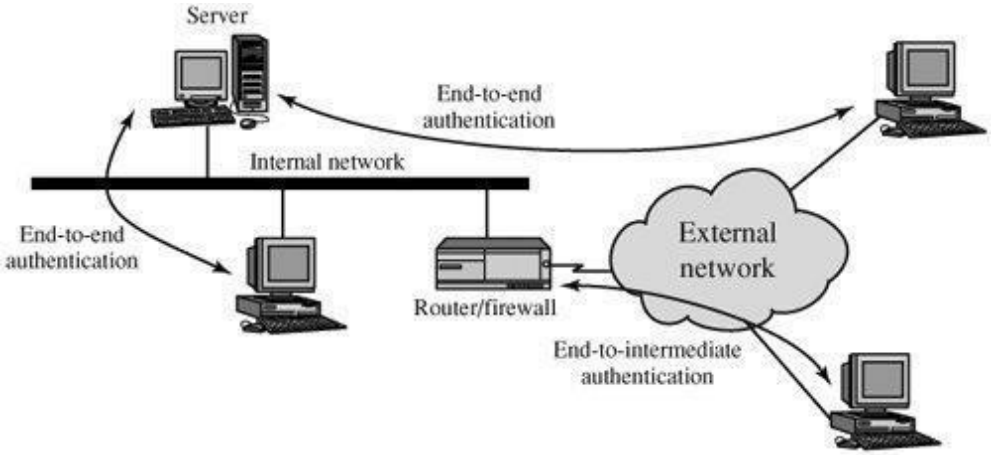


Figure. End-to-End versus End-to-Intermediate Authentication

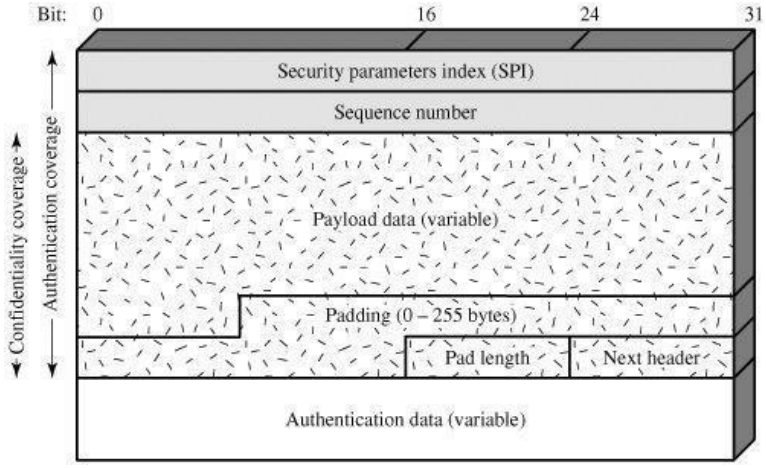
Encapsulating Security Payload(ESP):

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 an authentication service.

ESP Format:

It contains the following fields:

1. **Security Parameters Index (32 bits):** Identifies a security association.
2. **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
3. **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
4. **Padding (0-255 bytes):** The purpose of this field is discussed later.
5. **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.
6. **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that.
7. **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.

Various algorithms used for encryption are: Three-key triple DES, RC5, IDEA, Three-key triple IDEA,CAST, Blowfish

Padding:

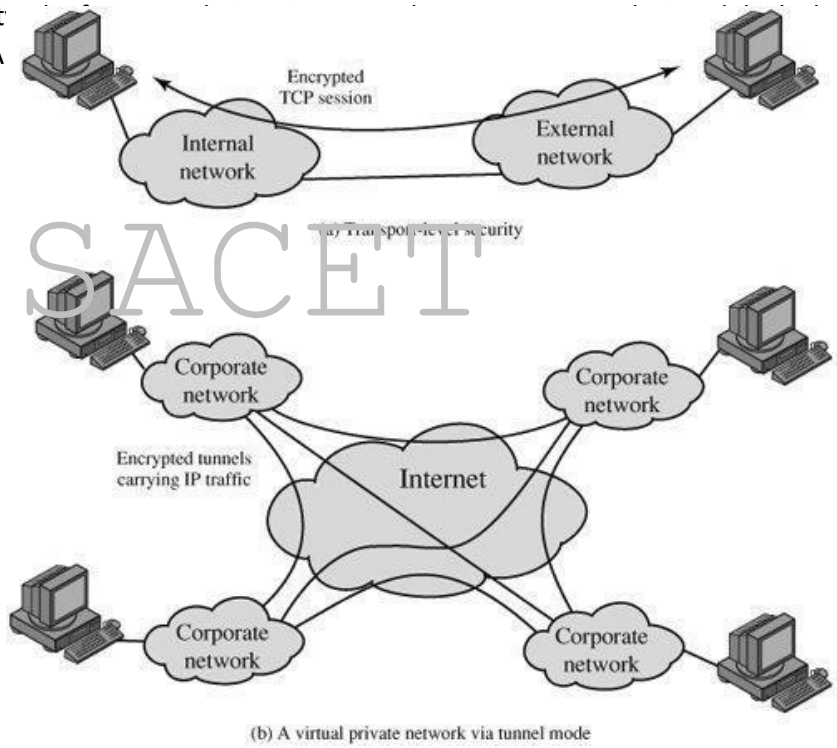
The Padding field serves several purposes:

1. If an encryption algorithm requires the plaintext to be a multiple of some number of bytes. The Padding field is used to expand the plaintext to the required length.
2. 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.
3. Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

Transport and Tunnel Modes:

Figure 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(b) shows how tunnel mode operation can be used to set up a *virtual private network*. 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 allowsthe hosts to avoid implementing the security capabilit tter technique uses atunnel mode SA



(b) A virtual private network via tunnel mode

COMBINING SECURITY ASSOCIATIONS:

An individual SA can implement either the AH or ESP protocol but not both. Sometimes a particular traffic flow will call for the services providedby both AH and ESP. Further, a particular traffic flow mayrequire IPSec

services between hosts and, for that same flow, separate services between security gateways, such as firewalls. In all of these cases, multiple SAs must be employed for the same traffic flow to achieve the desired IPSec services. The term ***security association bundle*** refers to a sequenceof SAs through which trafficmust be processed to provide a desired set of IPSec services.

Security associations may be combined into bundles in two ways:

- **Transport adjacency:** Refers to applying more than one security protocol to the same IPpacket, withoutinvoking tunneling.
- **Iterated tunneling:** Refers to the application of multiple layers of security protocols effectedthrough IP tunneling.

KEY MANAGEMENT:

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 AH and ESP.

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 withthe keys of other communicating systems. This is suitable 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.

The default automated key management protocol for IPSec is referred to as ISAKMP/Oakley andconsists of the following elements:

1. **Oakley Key Determination Protocol**
2. **Internet Security Association and Key Management Protocol (ISAKMP)**

Oakley Key Determination Protocol:

Oakley is a key exchange protocol basedon the Diffie-Hellman algorithmbut providingadded security.Oakley is genericin that it does notdictate specific formats.

The Diffie-Hellman algorithm has **two** attractive features:

1. Secret keys are created only when needed.
2. The exchange requires no preexisting infrastructure other than an agreement on the global parameters.

However, there are a number of weaknesses to Diffie-Hellman, as pointed out in

3. It does not provide any information about the identities of the parties.
4. It is subject to a man-in-the-middle attack

It is computationally intensive. As a result, it is vulnerable to a clogging attack, in which an opponent requests ahigh number of keys.

Oakley is designed to retain the advantages of Diffie-Hellman while countering its weaknesses.

Features of Oakley:

The Oakley algorithm is characterized by five important features:

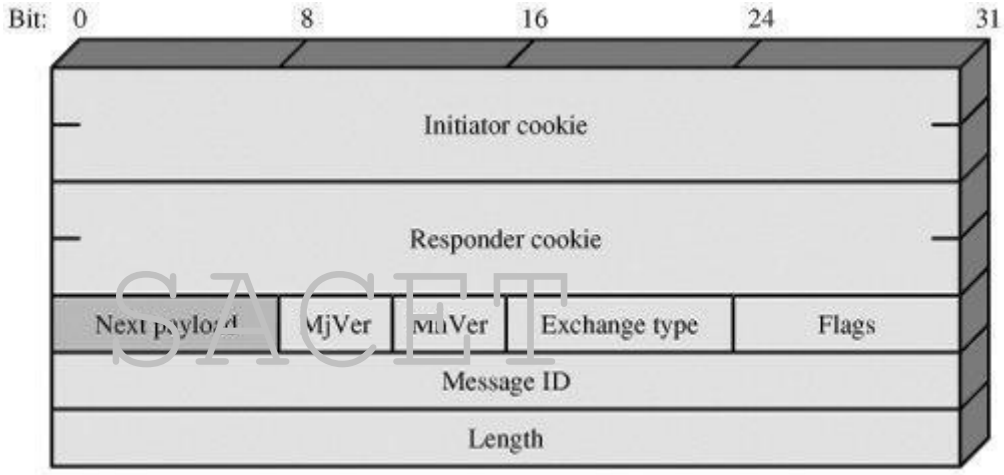
- ► It employs a mechanism known as cookies to thwart clogging attacks.
  - ► It enables the two parties to negotiate a group; this, in essence, specifies the global parameters of the Diffie-Hellman key exchange.
- ► It uses nonces to ensure against replay attacks.
- ► It enables the exchange of Diffie-Hellman publickey values.
- ► It authenticates the Diffie-Hellman exchange to thwart man-in-the-middle attacks.

Internet Security Association and Key Management Protocol (ISAKMP):

ISAKMP provides a framework for Internet key management and provides the specific protocolsupport, including formats, for negotiation of security attributes.

ISAKMP Header Format:

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



(a) ISAKMP header

It consists of the following fields:

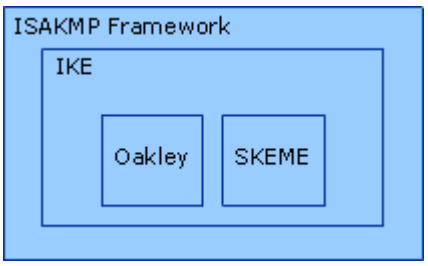
1. **Initiator Cookie (64 bits):** Cookie of entity that initiated SA establishment, SA notification, orSA deletion.
2. **Responder Cookie (64 bits):** Cookie of responding entity; null in first message from initiator.
3. **Next Payload (8 bits):** Indicates the type of the first payload in the message
4. **Major Version (4 bits):** Indicates major version of ISAKMP in use.
5. **Minor Version (4 bits):** Indicates minor version in use.
6. **Exchange Type (8 bits):** Indicates the type of exchange.
7. **Flags (8 bits):** Indicates specific options set for this ISAKMP exchange.
8. **Message ID (32 bits):** Unique ID for this message.
9. **Length (32 bits):** Length of total message (header plus all payloads) in octets.

INTERNET KEY EXCHANGE:

Internet Key Exchange (IKE) is a key management protocol standard used in conjunction

with the Internet Protocol Security (IPSec) standard protocol. It can also be described as a method for exchanging keys for encryption and authentication over an unsecured medium, such as the Internet.

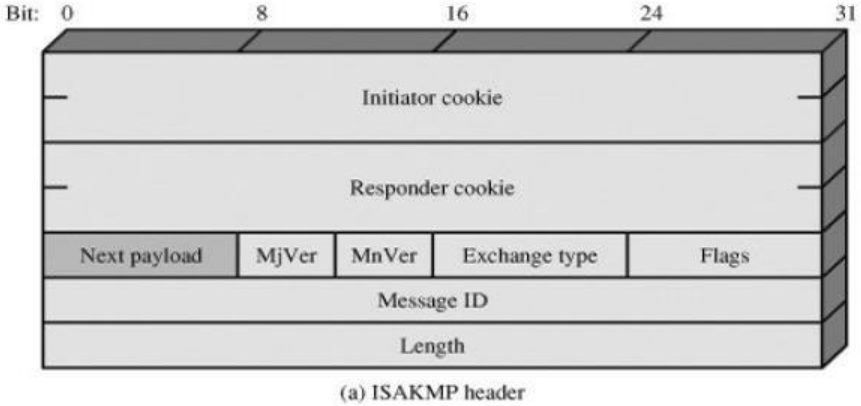
IKE is a hybrid protocol based on:



- ISAKMP
  - **Oakley**:
- SKEME
  -

1. **ISAKMP** : Internet Security Association and Key Management Protocols(ISAKMP) are used for negotiation and establishment of security associations. It's not a key exchange protocol , it's a framework on which key exchange protocols operate.

   ISAKMP Header Format
   - **Initiator Cookie (64 bits):** Cookie of entity that initiated SA establishment, SA notification, or SA deletion.
   - **Responder Cookie (64 bits):** Cookie of responding entity; null in first message from initiator.
   - **Next Payload (8 bits):** Indicates the type of the first payload in the message
   - **Major Version (4 bits):** Indicates major version of ISAKMP 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 ISAKMP exchange.
   - **Message ID (32 bits):** Unique ID for this message.
   - **Length (32 bits):** Length of total message



(a) ISAKMP header

2. **Oakley**: This protocol is used for key agreement or key exchange. Oakley defines the mechanism that is used for key exchange over an IKE session. The default algorithm forkey exchange used by this protocol is the Diffie-Hellman algorithm.

Features of Oakley

The Oakley 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 globalparameters 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.

Consider the problem of **clogging attacks**. In this attack, an opponent **forges** the source address of a legitimate user and sends a public Diffie-Hellman 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.

- **SKEME**: This protocol is another version for key exchange. Provides support for public- key-based key exchange, key distribution centres, and manual installation, it also outlines methods of secure and fast key refreshment. A secure and versatile key exchange protocol for key management over Internet is presented. SKEME constitutes a compact protocolthat supports a variety of realistic scenarios and security models over Internet.