Unit IV

4 Security Requirements

Syllabus

IP Security : Introduction, Architecture, IPV6, IPv4, IPSec protocols, and Operations, AH Protocol, ESP Protocol, ISAKMP Protocol, VPN. WEB Security: Introduction, Secure Socket Layer (SSL), SSL Session and Connection, SSL Record Protocol, Change Cipher Spec Protocol, Alert Protocol, Handshake Protocol. Electronic Mail Security: Introduction, Pretty Good Privacy, MIME, S/MIME, Comparison. Secure Electronic Transaction (SET).

Contents

4.1 IPv4

- IP corresponds to the network layer in the OSI reference model and provides a connectionless best effort delivery service to the transport layer. An Internet Protocol (IP) address has a fixed length of 32 bits.
- IPv4 addresses are unique. Two devices on the internet can never have the same address at the same time.
- The address structure was originally defined to have a two level hierarchy : Network ID and host ID.
- The network ID identifies the network the host is connected to. The host ID identifies the network connection to the host rather than the actual host.
- IP addresses are usually written in dotted decimal notation so that they can be communicated conveniently by people.
- The IP address structure is divided into five address classes : Class A, Class B, Class C, Class D and Class E, identified by the most significant bits of the addresses.

4.1.1 IPv4 Header Format

- Packets in the IPv4 layer are called datagrams. A datagram is a variable length packet consisting of two parts : Header and data.
- Fig. 4.1.1 shows IPv4 header format

1. **VER** is the field that contains the IP protocol version. The current version is 4.5 is an experimental version. 6 is the version for IPv6.
2. **HLEN** is the length of the IP header in multiples of 32 bits without the data field.The minimum value for a correct header is 5 (i.e. 20 bytes), the maximum value is 15 (i.e., 60 bytes).
3. **Service type** : The service type is an indication of the quality of service requested for this IP datagram. It contains the following information.

Precedence	Types of service	R

Precedence specifies the nature / priority :

000	Routine
001	Priority
010	Immediate
011	Flash
100	Flash override
101	Critical
110	Internetwork control
111	Internetwork control

TOS specifies the type of service value :

TOS bits	Description
1000	Minimize delay
0100	Maximum throughout
0010	Maximize reliability
0001	Minimize monetary cose
0000	Normal service

The last bit is reserved for future use.

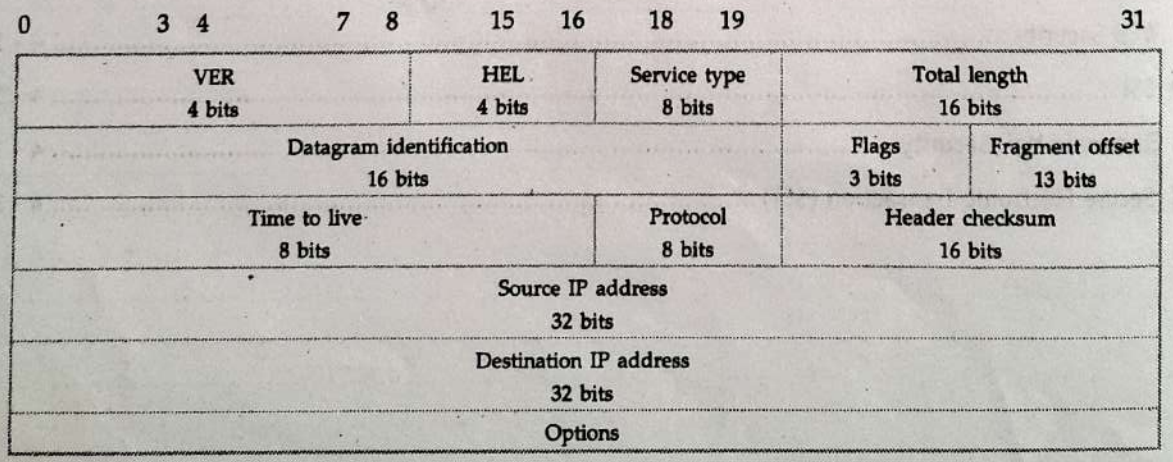


Fig. 4.1.1 IPv4 header format

4. **Total length** specifies the total length of the datagram, header and data, in octets.

5. **Identification** is a unique number assigned by the sender used with fragmentation.

6. **Flags** contain control flags :
   a. The first bit is reserved and must be zero;
   b. The 2nd bit is DF (Do not Fragment), 0 means allow fragmentation;
   c. The third is MF (More Fragments), 0 means that this is the last fragment.

7. **Fragment offset** is used to reassemble the full datagram. The value in this field contains the number of 64-bit segments (header bytes are not counted) contained in earlier fragments. If this is the first (or only) fragment, this field contains a value of zero.

8. **TTL** (Time To Live) specifies the time (in seconds) the datagram is allowed to travel. In practice, this is used as a hop counter to detect routing loops.

9. **Protocol number** indicates the higher level protocol to which IP should deliver the data in this datagram. E.g., ICMP = 1; TCP = 6; UDP = 17.

10. **Header checksum** is a checksum for the information contained in the header. If the header checksum does not match the contents, the datagram is discarded.

11. **Source/Destination IP addresses** are the 32-bit source/destination IP addresses.

12. **IP options** is a variable-length field (there may be zero or more options) used for control or debugging and measurement. For instance :
    a. The **loose source routing** option provide a means for the source of an IP datagram to supply explicit routing information;
    b. The **timestamp** option tell the routers along the route to put timestamps in the option data.

13. **Padding** is used to ensure that the IP header ends on a 32 bit boundary. The padding is zero.

4.2 IPv6

- IPv6 addresses are 128 bits in length. Addresses are assigned to individual interface on nodes, not to the node themselves.
- A single interface may have multiple unique unicast addresses. The first field of any IPv6 address is the variable length format prefix, which identifies various categories of addresses.
- A new notation has been devised for writing 16-byte addresses. They are written as eight groups of four hexadecimal digits with colons between the groups, like this 8000 : 0000 : 0000 : 0000 : 0123 : 4567 : 89AB : CDEF
- IPv6 allows three types of addresses: 1. Unicast 2. Anycast 3. Multicast

4.2.1 Packet Format

- The IPv6 packet is shown in Fig. 4.2.1. Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts : Optional and data.
- Fig. 4.2.2 shows the IPv6 datagram header format.

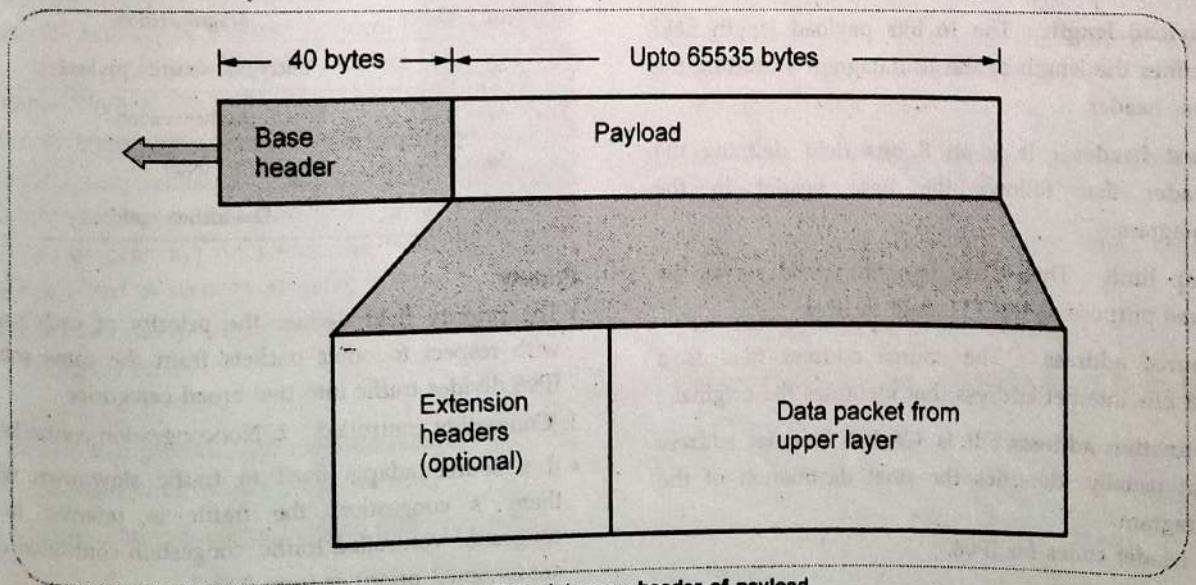


Fig. 4.2.1 IPv6 datagram header of payload

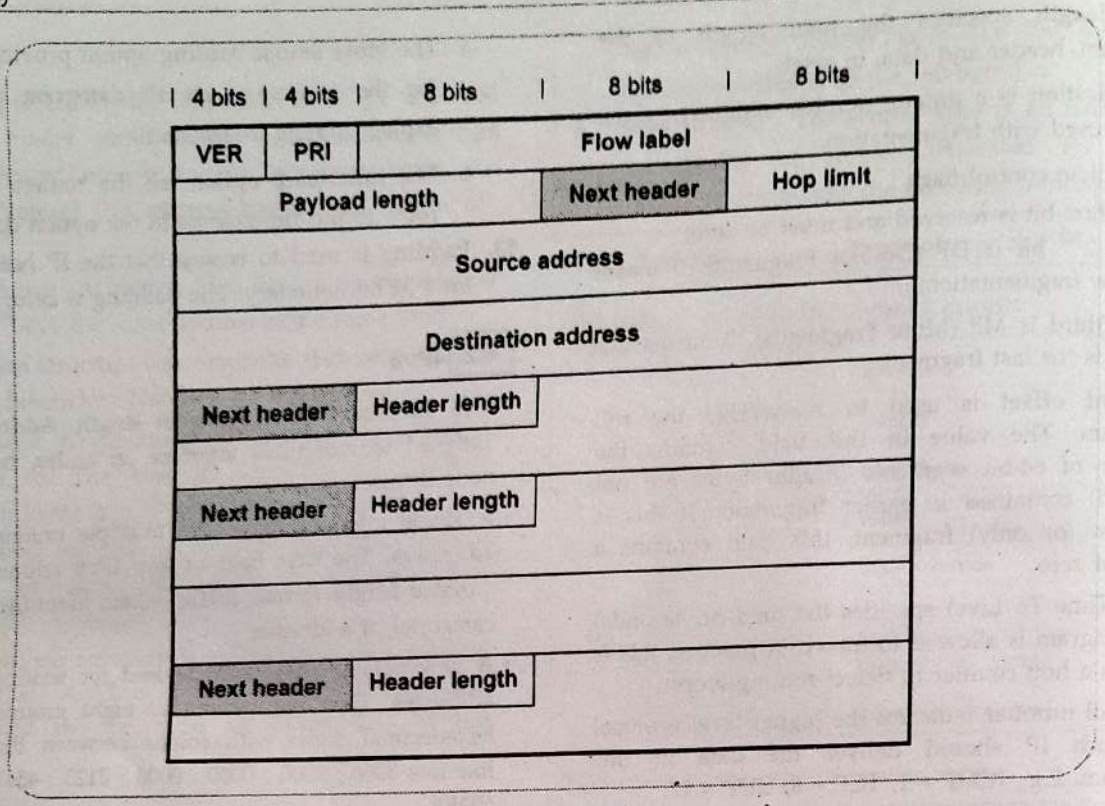


Fig. 4.2.2 IPv6 header

1. **Versions** : This 4 bits field defines the version number of the IP. The value is 6 for IPv6.
2. **Priority** : The 4 bits priority field defines the priority of the packet with respect to traffic congestion.
3. **Flow label** : It is 24 bits field that is designed to provide special handling for a particular flow of data.
4. **Payload length** : The 16 bits payload length field defines the length of the IP datagram excluding the base header.
5. **Next header** : It is an 8 bits field defining the header that follows the base header in the datagram.
6. **Hop limit** : This 8 bits hop limit field serves the same purpose as the TTL field in IPv4.
7. **Source address** : The source address field is a 128 bits internet address that identifies the original.
8. **Destination address** : It is 128 bits Internet address that usually identifies the final destination of the datagram.

Next header codes for IPv6.

Code	Next header
0	Hop by hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null
60	Destination option

Priority

- The priority field defines the priority of each packet with respect to other packets from the same source. IPv6 divides traffic into two broad categories

1. Congestion controlled 2. Noncongestion controlled

- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion controlled traffic. congestion controlled data are assigned priorities from 0 to 7.

Priority	Meaning
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

- A priority of 0 is the lowest; a priority of 7 is the highest.
- Noncongestion controlled traffic refers to a type of traffic that excepts minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real time audio and video are examples of this type of traffic.
- Priority numbers from 8 to 15 are assigned to noncongestion controlled traffic.

4.3 IPSec Protocols

- Different application specific security mechanisms are developed such as electronic mail (PAG, S/MIME), client/server (Kerberos), web access (secure sockets layer). An IP level security can ensure secure networking not only for applications with security mechanisms but also for many security ignorant applications.
- IP Security (IPSec) is the capability that can be added to present versions of Internet Protocol (IPv4 and IPv6) by means of additional headers for secure communication across LAN, WAN and Internet.
- IPSec is a set of protocols and mechanism that provide confidentiality, authentication, message integrity and replay detection at IP layer. The device (firewall or gateway) on which the IPSec mechanism reside is called as **security gateway**.
- IPSec has two modes of operation.
  1. Transport mode
  2. Tunnel mode
- IPSec uses two protocols for message security.
  1. Authentication Header (AH) protocol.
  2. Encapsulating Security Payload (ESP) protocol.

4.3.1 Applications of IPSec

1. Secure connectivity over the Internet : A Virtual Private Network (VPN) can be established over the Internet. This reduces cost of private networks and network management overheads.

2. Secure remote access over the Internet : With IPSec, Secure access to a company network is possible.

3. Extranet and intranet connectivity : With IPSec, secure communication with other organizations, ensures authentication and confidentiality and provide a key exchange mechanism.

4. Enhanced electronic-commerce security : Use of IPSec enhances the security in electronic commerce applications.

4.3.2 IP Security Scenario

Fig. 4.3.1 shows an IP security scenario. (See Fig. 4.3.1 on next page)

- Many organizations have LAN at multiple places. The IPSec protocols are used which operates in networking devices e.g. router or firewall.
- The IPSec networking encrypt and compress the outgoing traffic while it decrypt and decompress all incoming traffic. These processes are transparent to workstations and servers on LAN.

4.3.3 Benefits of IPSec

1. IPSec provides strong security within and across the LANs.
2. IPSec in a firewall avoids bypass if all traffic from the outside must use IP.
3. No need to change software for implementing IPSec.
4. IPSec is below transport layer and hence is transparent to applications.
5. IPSec is transparent to end users also.
6. If required IPSec can provide security to individual users.

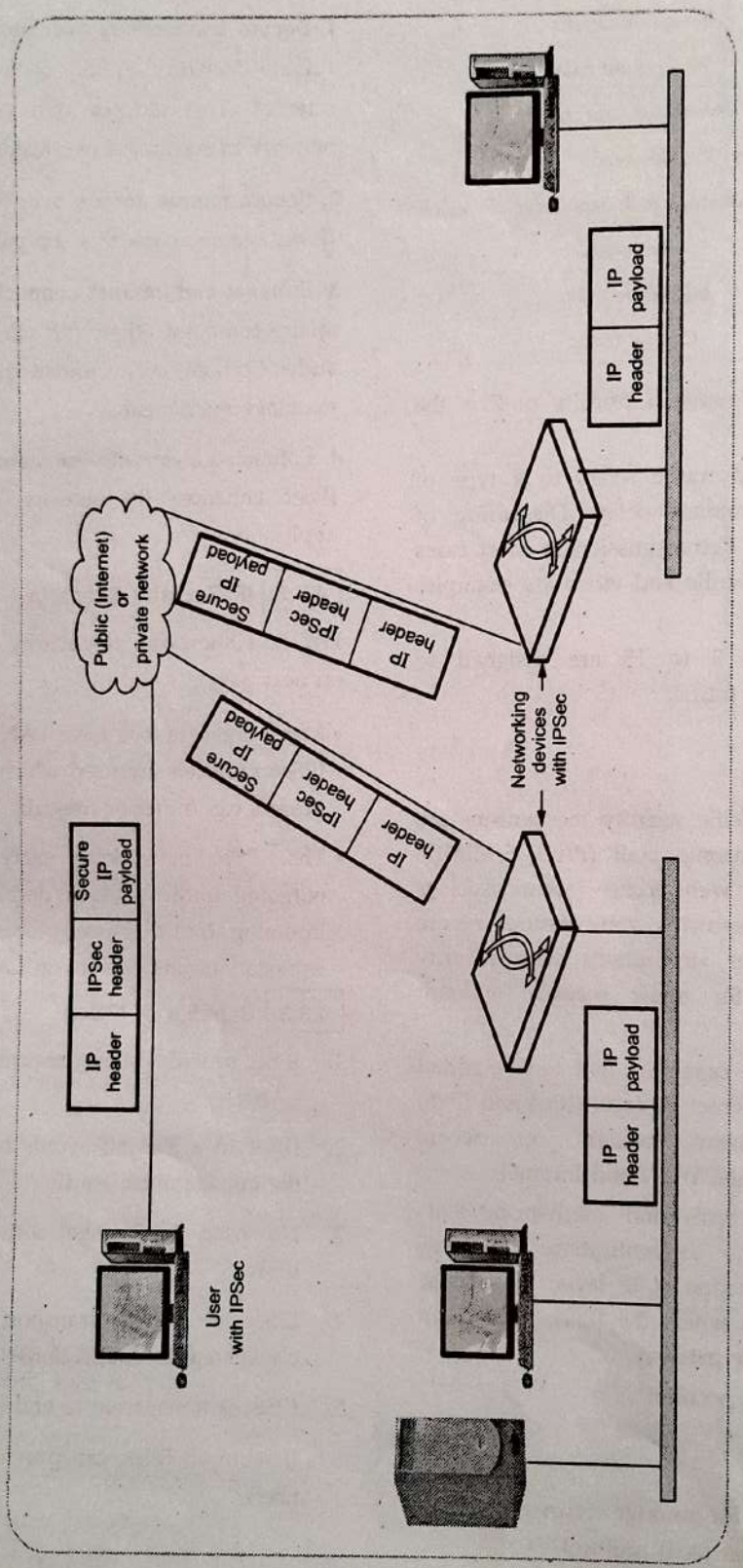


Fig. 4.3.1 IPSec scenario

Review Questions

1. *Describe IPsec protocol with its components and security services.*
2. *List and explain components of IPSec protocol.*

4.4 IP Security Architecture

- IPSec mechanism uses Security Policy Database (SPD) which determines how a messages are to handle also the security services needed and path the packet should take.
- Various documents are used to define complex IPSec specification. The overall architecture of IPSec is constituted by three major components.
  1. IPSec documents
  2. IPSec services
  3. Security Associations (SA)

4.4.1 IPSec Documents

- IPSec specifications are described in various documents. Few important documents and specifications described are as under -

Sr. No.	Documents	Specifications
1.	RFC 2401	Overview of security architecture.
2.	RFC 2402	Packet authentication extension to IPv4 and IPv6.
3.	RFC 2406	Packet encryption extension to IPv4 and IPv6.
4.	RFC 2408	Key management capabilities

- All above specifications are essentially supported by IPv6 and are optional for IPv4. The security features are incorporated as extension header to the main IP header for both IPv4 and IPv6.
- The extension header for authentication is called as Authentication Header (AH) and the extension header for encryption is called as Encapsulating Security Payload (ESP) header.
- Besides RFC various other documents are published by Internet Engineering Task Force (IETF). These documents can be divided into seven groups.
- IPSec protocol consists of seven different groups of document as shown in Fig. 4.4.1.

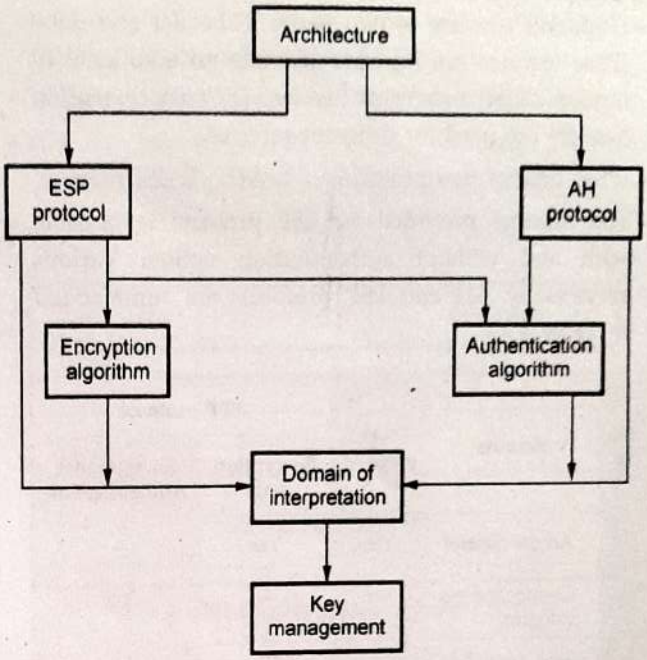


Fig. 4.4.1 IPSec document

1. **Architecture** : Covers security requirements, definitions, IPSec technology.
2. **Encapsulating Security Payload (ESP)** : Covers packet format, packet encryption authentication.
3. **Authentication Header (AH)** : Covers packet format, general issues.
4. **Authentication algorithm** : Encryption algorithms used for ESP.
5. **Key management** : Key management schemes.
6. **Domain of Interpretation (DoI)** : Values to relate documents with each other.

4.4.2 IPSec Services

- IPSec provides security serives at IP layer by selecting required security protocols, algorithms and cryptographic keys as per the services requested.
- Two protocols performs the function of providing security. These are authentication header protocol and protocol for encapsulating security payload. The services provide by these protocols are -
  a. Access control
  b. Connectionless integrity
  c. Data origin authentication
  d. Rejection of replayed packets
  e. Confidentiality
  f. Limited traffic flow confidentiality

IPSec protocol suit

- IP packet consists of two parts; IP header and data. IPSec features are incorporated into an additional IP header called extension header. Different extension headers are used for different services.
- IPSec defines two protocols : 1. AH 2. ESP
- The services provided by ESP protocol is possible with and without authentication option. Various services by AH and ESP protocols are summarized in Table 4.4.1.

Sr. No.	Service	AH protocol	ESP protocol	
			Encryption only	Encryption + Authentication
1.	Access control	Yes	Yes	–
2.	Connectionless integrity	Yes	–	Yes
3.	Data origin authentication	Yes	–	Yes
4.	Rejection of packets	Yes	Yes	Yes
5.	Confidentiality	Yes	Yes	Yes
6.	Limited traffic flow confidentiality	–	Yes	Yes

Table 4.4.1

4.4.3 Security Associations (SA)

- Security Association (SA) is the common between authentication and confidentiality mechanisms. An association is a one-way relationship between transmitter and receiver. For a two-way secure exchange two security associations are required.
- A security association is defined by parameters.
  1. Security Parameters Index (SPI)
  2. IP destination address
  3. Security protocol identifiers

1. Security Parameters Index (SPI) : SPI is a string of bit assigned to this SA and has local significance only. SPI is located in AH and ESP headers. SPI enables the receiving system under which the packet is to process.

2. IP destination address : It is the end point address of SA which can be end user system or a network system (firewall / router).

3. Security protocol identifiers : Security protocol identifier indicates whether the association is an AH or ESP security association.

4.4.4 SA Parameters

- A Security Association (SA) is normally defined by following parameters.

1. Sequence number counter : Sequence number counter is a 32-bit value that indicates the sequence number field in AH or ESP.

2. Sequence counter overflow : Sequence counter overflow is a flag used to indicate whether overflow of the sequence number counter should generate an auditable event and prevent further transmission of packets on SA.

3. Anti-replay window : Anti - replay window determines whether on inbound AH or ESP packet is a replay.

4. AH information : AH information includes authentication algorithm, keys, key life times and related parameters being used with AH.

5. ESP information : ESP information includes encryption and authentication algorithm, keys, initialization values required for ESP implementation.

6. IPSec protocol mode : IPSec protocol mode can be tunnel, transport or wildcard.

7. Path MTU : Path MTU means observed path maximum transmission unit which indicates maximum size of a packet that can be transmitted without fragmentation.

4.4.5 Transport Mode

- AH and ESP can support two modes of operation.
  1. Transport mode
  2. Tunnel mode
- Transport mode mainly provide protection for upper layer protocols. The protection extends to the payload of an IP packet. For example, TCP or UDP segment or ICMP packet.
- The transport mode is suitable for end-to-end communication between two workstations.
- In transport mode, ESP encrypts the IP payload excluding IP header. Authentication of IP payload is optional.
- AH authenticates the IP payload and specific portions of IP header.

4.4.6 Tunnel Mode

- Tunnel mode provides protection to entire IP packets. Security fields are added to IP packets and entire packet (AH or ESP packet + Security packet) is new IP packet with a new IP header.
- Entire new IP packet travels through a tunnel from one point to other over IP network. No router over the network are able to detect inner IP header. Since original packet is encapsulated by new larger packet having different source and destination address.
- Tunnel mode is preferred when one or both ends of an SA a security gateway such as a firewall or router that implements IPSec.
- In tunnel mode, number of hosts on network with firewalls may engage in secure transmission without IPSec. The unsecured packets generated are tunneled through external networks by tunnel mode SAs or IPSec in firewall or router.
- ESP encrypts and optionally authenticates the entire inner IP packet including IP header.
- AH authenticates the entire inner IP packet and selected portion of outer IP header.
- The tunnel mode and transport mode functionality is summarized in Table 4.4.2.

Protocol	Transport mode	Tunnel mode
AH	Authenticates IP paylaod and selected portion of IP header.	Authenticates entire IP packet and selected portion of outer IP header.
ESP	Encrypts IP payload and IPv6 extension headers.	Encrypts entire inner IP packet.
ESP with Authentication	Authenticates IP payload and not IP header. Encrypts IP payload and IPv6 header.	Authenticates inner IP packet. Encrypts entire inner IP packet.

Table 4.4.2

4.5 Authentication Header

- It provides support for data integrity and authentication of IP packets.
- Data integrity service insures that data inside IP packets is not altered during the transit.
- Authentication service enables and end user to authenticate the user at the other end and decides to accept or reject packets accordingly.
- Authentication also prevents the IP spoofing attack.
- AH is based on the MAC protocol, i.e. two communication parties must share a secret key.
- AH header format is shown in Fig. 4.5.1.

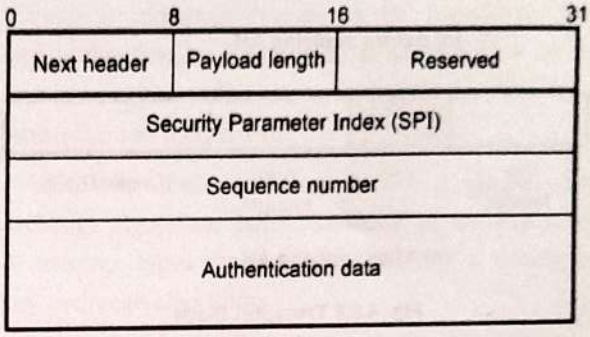


Fig. 4.5.1 IPSec authentication header format

1. **Next header** - This is 8-bits field and identifies the type of header that immediately follows the AH.
2. **Payload length** - Contains the length of the AH in 32-bit words minus 2. Suppose that the length of the authentication data field is 96-bits (or three 32-bit words) with a three word fixed header, then we have a total of 6-words in the header. Therefore this field will contain a value of 4.
3. **Reserved** - Reserved for future use (16-bit).
4. **SPI** - Used in combination with the SA and DA as well as the IPSec protocol used (AH or ESP) to uniquely identify the security association for the traffic to which a datagram belongs.
5. **Sequence number** - To prevent replay attack.

Replay attack

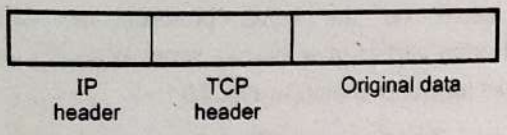
1. Suppose user A wants to transfer some amount to user C's bank account.
2. Both user A and C have the accounts with bank B.
3. User A might send an electronic message to bank B requesting for the funds transfer.
4. User C could capture this message and send a second copy of the message to bank B.
5. Band B have no idea that this is an unauthorized message.
6. User C would get the benefit of the funds transfer twice.

Authentication data

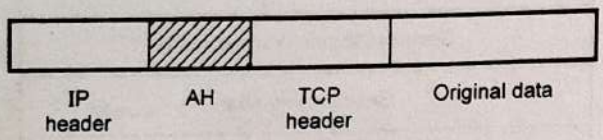
Also called Integrity check value for the datagram. This value is the MAC used for authentication and integrity purposes.

4.5.1 AH Transport Mode

- The position of the AH is between the original IP header and original TCP header of the IP packet.
- Fig. 4.5.2 shows the AH in transport mode.



(a) Before applying AH



(b) After applying AH

Fig. 4.5.2 Transport mode

4.5.2 AH Tunnel Mode

- The entire original IP packet is authenticated.
- AH is inserted between the original IP header and a new outer IP header.
- Fig. 4.5.3 shows AH tunnel mode.

4.6 ESP

- Encapsulating Security Payload (ESP) provides confidentiality services and limited traffic flow confidentiality. An authentication service is optional feature.

4.6.1 ESP Format

- Fig. 4.6.1 shows IPSec ESP format.

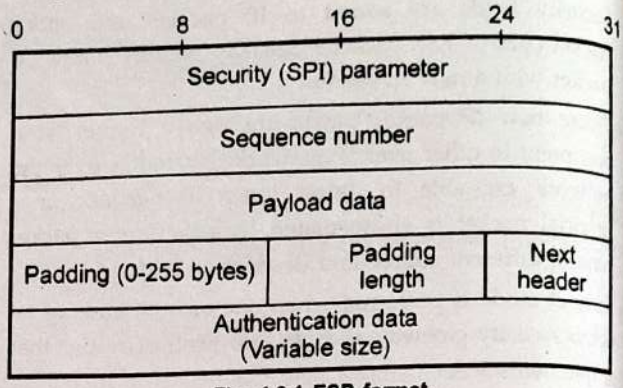
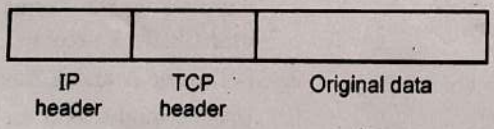
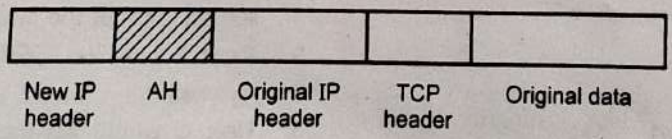


Fig. 4.6.1 ESP format

1. **SPI** - It is 32-bits field used in combination with the source and destination address. It identifies a security association.
2. **Sequence number** - This 32-bit field is used to prevent replay attacks.
3. **Payload data** - This is a transport level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
4. **Padding** - It contains the padding bits.
5. **Padding length** - Indicates the number of pad bytes immediately preceding this field.
6. **Next header** - It identifies the type of encapsulated data in the payload.
7. **Authentication data** - It is variable length field contains the authentication data called as the integrity check value for the datagram.



(a) Before applying AH



(b) After applying AH

Fig. 4.5.3 Tunnel mode

4.6.2 Encryption and Authentication Algorithms

- The payload data, padding, pad length and next header fields are encrypted by ESP.
- Various algorithms used for encryption are -
  1. Three-key triple DES 2. RCS 3. IDEA
  4. Three-key triple IDEA 5. CAST 6. Blowfish

4.6.3 Padding

- Padding field is used for various purposes such as
  1. To expand the plain text if an encryption algorithm requires the plain text to be a multiple of number of bytes.
  2. To assure the alignment of cipher text to make it integer multiple of 32-bits.
  3. To provide partial traffic flow confidentiality by concealing the actual length of payload.

4.6.4 Comparison between AH and ESP

Sr. No.	AH	ESP
1.	Defined in RFC 2402	Defined in RFC 2406
2.	AH mandatory for IPv6 compliance.	Use of ESP with IPv6 is optional.
3.	Provides stronger authentication in transport mode.	Authentication provided is not as strong as AH.
4.	Requires less overhead since it only inserts a header into the IP packet.	Requires more overhead as it inserts a header and trailer.
5.	Provides connectionless integrity and data origin authentication for IPv4 and IPv6	Provides confidentiality, data origin authentication, connectionless integrity, an anti-reply service and limited traffic flow confidentiality.
6.	Protects as much of the IP header as possible as well as upper level protocol data.	It only protects those IP header fields that it encapsulates.
7.	It provides a packet authentication service.	It encrypts and /or authenticates data.

4.7 ISAKMP Protocol

- Internet Security Association and Key Management Protocol (ISAKMP) provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes.
- ISAKMP defines procedures and packet formats to establish, negotiate, modify and delete Security Associations.
- ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.
- ISAKMP by itself does not dictate a specific key exchange algorithm; rather, ISAKMP consists of a set of message types that enable the use of a variety of key exchange algorithms.
- ISAKMP provides a "cookie" or an anti- clogging token (ACT) to make it easier to handle denial of service and prevents connection hijacking by linking the authentication, key exchange and Security Association exchanges.
- Fig. 4.7.1 shows ISAKMP header format.
- **Initiator cookie (8 bytes)** : The cookie of the entity that initiated SA establishment, SA notification, or SA deletion.
- **Responder cookie (8 bytes)** : The cookie of the entity that is responding to an SA establishment request, SA notification, or SA deletion.
- **Next payload (8 bits)** : Indicates the type of the first payload in the message.
- **Mj version (4 bits)** : The major version of the ISAKMP protocol in use.
- **Mn version (4 bits)** : The minor version of the ISAKMP protocol in use.
- **Exchange type (8 bits)** : Indicates the type of exchange being used. This dictates the message and payload orderings in the ISAKMP exchanges.
- **Flags (8 bits)** : Indicates the options that are set for the ISAKMP exchange
- **Message ID (4 bytes)** : A unique value used to identify the protocol state during Phase 2 negotiations. It is randomly generated by the initiator of the Phase 2 negotiation.
- **Length (4 bytes)** : The total length of the ISAKMP header and the encapsulated payloads in bytes. The

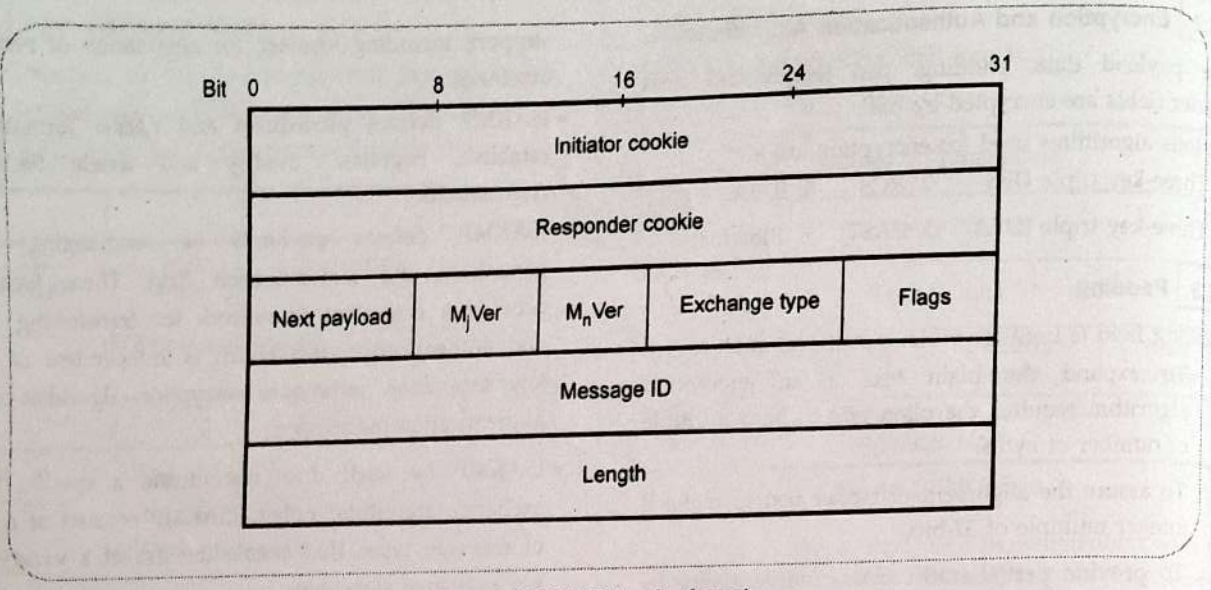


Fig. 4.7.1 ISAKMP header format

Length field of the ISAKMP header shows the total length of the message and the header together in octets.

4.7.1 OAKLEY Determination Protocol

- Key management is related to determination and distribution of secret keys. Four keys for communication between two applications : Transmitter and receiver pairs for both AH and ESP.
- Basically Oakley is a protocol to carry out the key exchange negotiation process for both peers, in which both ends after being authenticated can agree on secure and secret keying material.
- Oakley is based on the Diffie-Hellman key algorithm in which two gateways can agree on a key without the need to encrypt.
- Two users A and B agree on two global parameters : q, a large prime number and a primitive root of q.
- Secret keys created only when needed. Exchange requires no preexisting infrastructure. This algorithm is simple to use and did not require to much computational time.
- Authentication is used as part of the identity protection and since the oakley protocol uses the users public key we see a hash function used to retain the certification of these keys.
- Cookie generation criteria :
  1. must depend on the specific parties
  2. must not be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity
  3. cookie generation function must be fast to thwart attacks intended to sabotage CPU resources
  4. hash over the IP source & destination address, the UDP source and destination ports and a locally generated secret random value

Review Questions

1. *Explain OAKLEY key determination protocol.*
2. *Explain ISAKMP protocol for IP sec.*
3. *What is the role OAKLEY protocol in communication ?*

4.8 VPN

- Generalized architecture of VPN is shown in Fig. 4.8.1.
- Virtual Private Network (VPN) based on IPSec protocol are widely used for providing secure encrypted communication over insecure network, such as the internet.
- VPN can be implemented on the top of ATM.
- Authentication in IPSec is handled by the Internet Key Exchange (IKE) protocol.
- Virtual private network is a restricted to use logical computer network that is constructed from the system resources of a public and physical network such as the Internet, by using encryption.

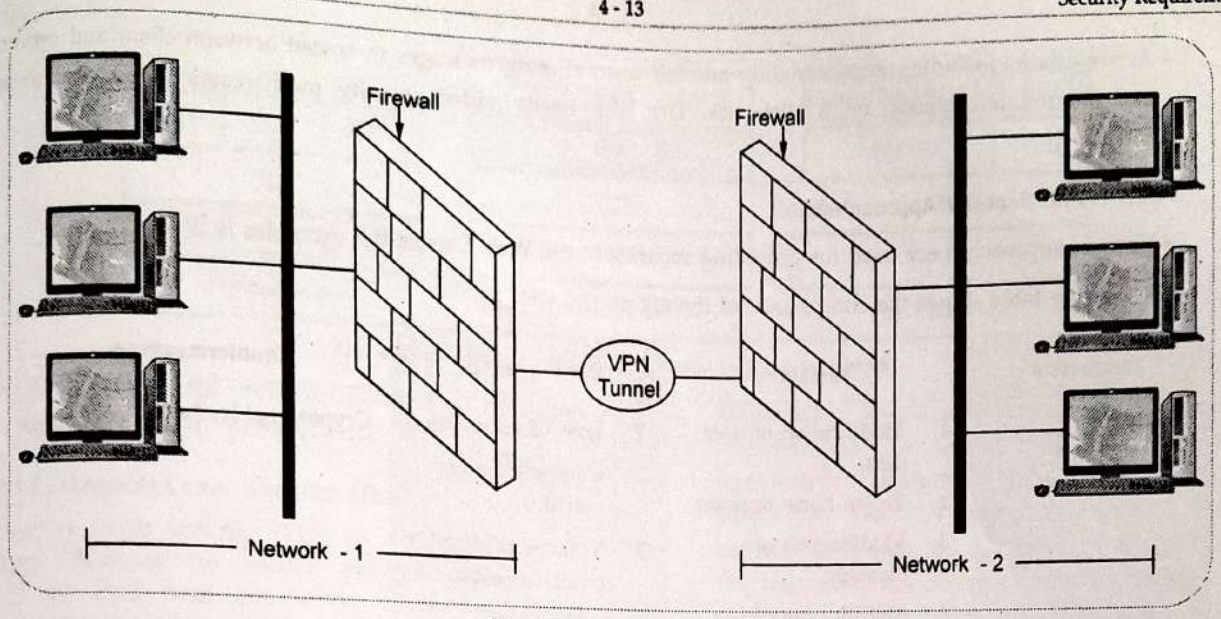


Fig. 4.8.1 VPN architecture

- VPN technology is based on a tunneling strategy. Tunneling involves encapsulating packets constructed in a base protocol format within some other protocol.
- In the case of VPNs running over the Internet, packets in one of several VPN protocol formats are encapsulated within IP packets.
- Following network protocols have become popular as a result of VPN developments : PPTP, L2F, L2TP, IPsec, SOCKS etc.
- Authentication allows VPN clients and servers to correctly establish the identity of people on the network.
- Encryption allows potentially sensitive data to be hidden from the general public.

4.8.1 Components of VPN

- A VPN connection includes the following components :

1. **VPN server :** A computer that accepts VPN connections from VPN clients.
2. **VPN client :** A computer that initiates a VPN connection to a VPN server. A VPN client can be an individual computer or a router.
3. **Tunnel :** The portion of the connection in which your data is encapsulated.
4. **VPN connection :** The portion of the connection in which your data is encrypted. For typical secure VPN connections, the data is encrypted and encapsulated along the same partion of the connection.
5. **Tunneling protocols :** Protocols that are used to manage tunnels and encapsulate private data. Data that is tunneled must also be encrypted to be a VPN connection.
6. **Tunneled data :** Data that is usually sent across a private point-to-point link.
7. **Transit internetwork :** The shared or public network crossed by the encapsulated data. The transit internetwork can be the Internet or a private IP-based intranet.

Review Questions

1. *State security measure applied by VPN for security.*
2. *What is VPN ? Explain types of VPN.*

4.9 WEB Security

- The Web is very visible. The WWW is widely used by businesses, government agencies, and many individuals. But the Internet and the Web are extremely vulnerable to compromises of various sorts, with a range of threats.
- Complex software hides many security flaws. Web servers are easy to configure and manage. Users are not aware of the risks.
- These can be described as passive attacks including eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.

- Active attacks including impersonating another user, altering messages in transit between client and server, and altering information on a Web site. The Web needs added security mechanisms to address these threats.

Web Traffic Security Approaches

- Various approaches are used for providing security to the Web. One of the examples is IP security.
- Following table shows the comparison of threats on the web.

Parameters	Threats	Consequences	Countermeasures
Integrity	1. Modification of user data 2. Trojan horse browser 3. Modification of memory 4. Modification of message traffic in transit	1. Loss of information 2. Compromise of machine 3. Vulnerability to all other threats	Cryptographic checksums
Confidentiality	1. Eavesdropping on the Net 2. Theft of information from server 3. Theft of data from client 4. Information about network configuration 5. Information about which client talks to server	1. Loss of information 2. Loss of privacy	Encryption, Web proxies
Denial of Service	1. Killing of user threads 2. Flooding machine with bogus requests 3. Filling up disk or memory 4. Isolating machine by DNS attacks	1. Disruptive 2. Annoying 3. Prevent user from getting work done	Difficult to prevent
Authentication	1. Impersonation of legitimate users 2. Data forgery	1. Misrepresentation of user 2. Belief that false information is valid	Cryptographic techniques

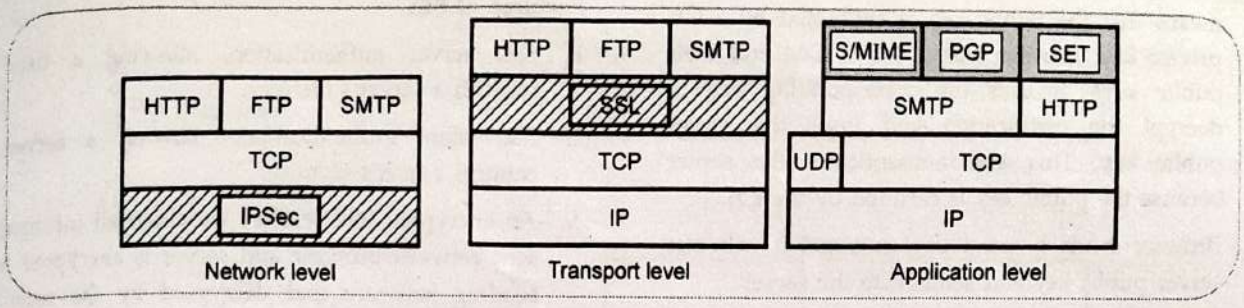


Fig. 4.9.1 Relative locations of security facilities in TCP/IP

- Fig 4.9.1 shows the relative location of security facilities in the TCP/IP protocol stack.

4.9.1 Transport Layer Security (TLS)

- Transport Layer Security (TLS) is a feature of mail servers designed to secure the transmission of electronic mail from one server to another using encryption technology. TLS can reduce the risk of eavesdropping tampering and message forgery mail communications.
- TLS is a security protocol from the Internet Engineering Task Force (IETF) that is based on the Secure Sockets Layer (SSL) 3.0 protocol developed by Netscape.
- TLS was designed to provide security at the transport layer. TLS is a non-proprietary version of SSL. For transactions on Internet, a browser needs :
  1. Make sure that server belongs to the actual vendor.
  2. Contents of message are not modified during transition.
  3. Make sure that the imposter does not interpret sensitive information such as credit card number.
- Fig. 4.9.2 shows the position of TLS in the protocol.

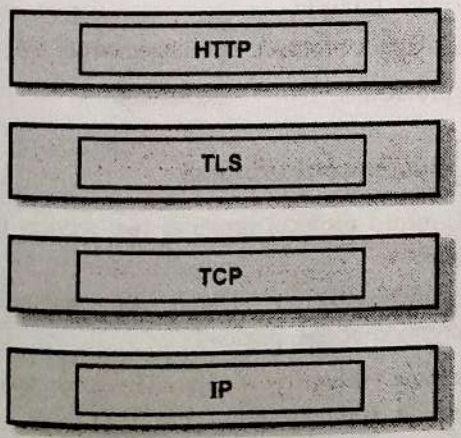


Fig. 4.9.2 TLS

- TLS has two protocols : Handshake and data exchange protocol

1. **Handshake** : Responsible for negotiating security, authenticating the server to the browser and (optionally) defining other communication parameters. The TLS handshake protocol allows authentication between the server and client and the negotiation of an encryption algorithm and cryptographic keys before the application protocol transmits or receives any data.
2. **Data exchange (record) protocol** : Data exchange (record) protocol uses the secret key to encrypt the data for secrecy and to encrypt the message digest for integrity. The TLS record protocol is designed to protect confidentiality by using symmetric data encryption.

Handshake protocol

- Fig. 4.9.3 shows the TLS handshake protocol.

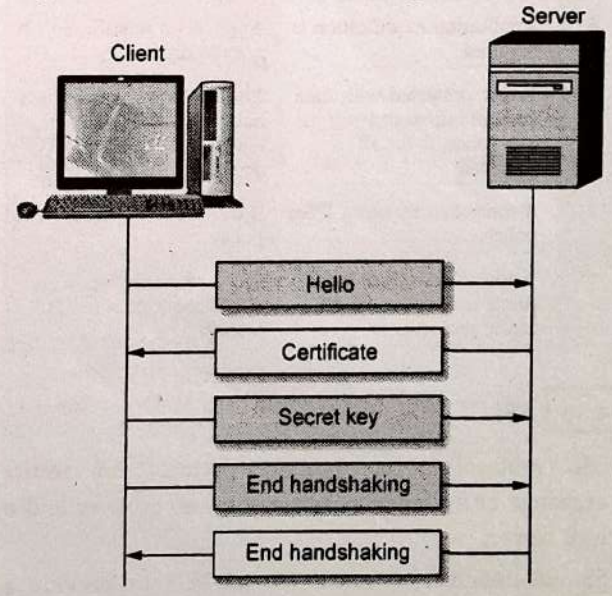


Fig. 4.9.3 TLS handshake protocol

1. Browser sends a hello message that includes TLS version and some preferences.
2. Server sends a certificate message that includes the public key of the server. The public key is certified by some certification authority, which

means that the public key is encrypted by a CA private key. Browser has a list of CAs and their public keys. It uses the corresponding key to decrypt the certification and finds the server public key. This also authenticates the server because the public key is certified by the CA.

3. Browser sends a secret key, encrypts it with the server public key and sends it to the server.
4. Browser sends a message, encrypted by the secret key to inform the server that handshaking is terminating from the browser key.
5. Server decrypts the secret key using it private key and decrypts the message using the secret key. It then sends a message, encrypted by the secret key, to inform the browser that handshaking is terminating from the server side.

4.9.2 Comparison between IPsec and TLS

Sr. No.	IPSec	TLS
1.	Type of security is device to device.	Type of security is application to application.
2.	It provides network segment protection.	It does not provides network segment protection.
3.	Application modification is required.	Application modification is not required.
4.	Traffic protected with data authentication and encryption is for all protocol.	Traffic protected with data authentication and encryption is only for TCP protocol.
5.	It controlled by using IPSec policy.	It controlled by using TLS policy.
6.	Scope of protection is for single connection for all traffic protocol.	Scope of protection is for single connection for TLS session.

4.10 SSL

- SSL protocol is an internet protocol for secure exchange of information between a web browser and a web server.
- SSL is designed to make use of TCP to provide a reliable end to end secure service.
- Secure Socket Layer (SSL) provides security services between TCP and applications that use TCP. The SSL protocol is an internet protocol for secure exchange of information between a web browser and a web server.

Features of SSL

1. SSL server authentication, allowing a user to confirm a server's identity.
2. SSL client authentication, allowing a server to confirm a user's identity.
3. An encrypted SSL session, in which all information sent between browser and server is encrypted by a sending software and decrypted by the receiving software.
4. SSL supports multiple cryptographic algorithms.

4.10.1 SSL Protocol Stack

SL uses TCP to provide reliable end-to-end secure service. SSL consists of two subprotocols, one for establishing a secure connection and other for using it. Fig. 4.10.1 shows SSL protocol stack.

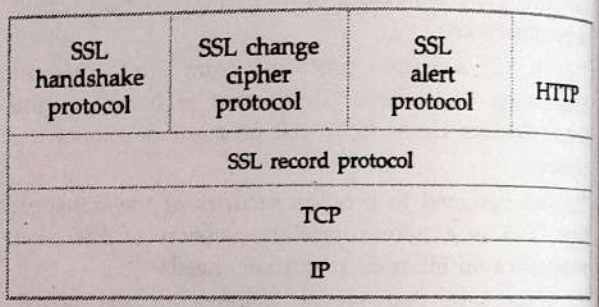


Fig. 4.10.1 SSL protocol stack

SSL record protocol : It provides basic security services to various higher layer protocols

HTTP : Provides the transfer service for web client/server interaction.

SSL handshake protocol,
SSL change cipher protocol, : Management of SSL
SSL alert protocol. exchanges.

4.10.2 SSL Record Protocol

- The SSL record protocol provides two services for SSL connection.
  1. **Confidentiality** - Handshake protocol for encryption of SSL payload.
  2. **Message integrity** - Handshake protocol for Message Authentication Code (MAC).
- SSL record protocol operation is shown in Fig. 4.10.2

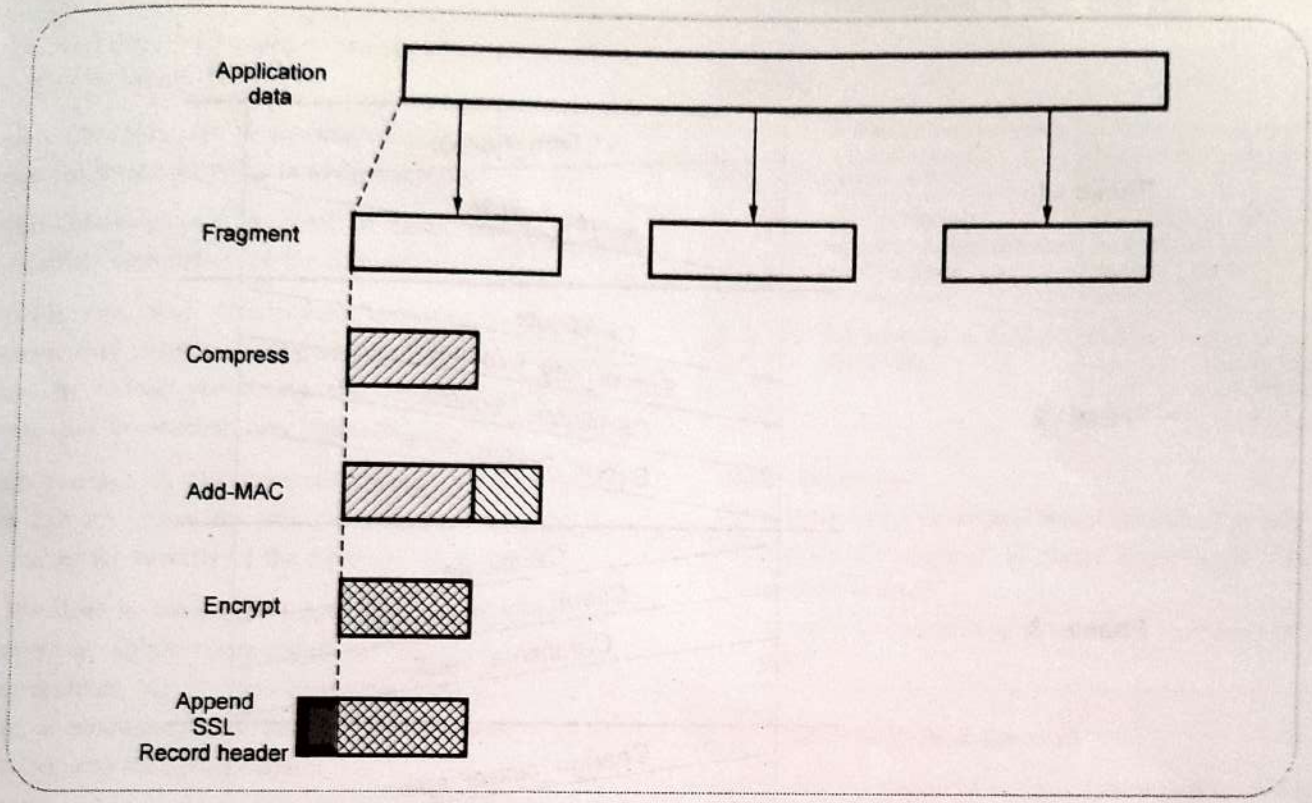


Fig. 4.10.2 SSL record protocol operation

- The record protocol takes application message to transmit, fragments the data, compress, applies MAC, encrypts, adds a header and transmits the TCP segment.

4.10.3 Handshake Protocol

- Handshake protocol allows the server and client to authenticate each other and to negotiate an encryption before transmitting application data various massages are used in protocol. Table 4.10.1 enlist these messages and there associated function.

Phase	Message type	Function
1.	Hello - request Client - hello Server - hellow	Null Version, session id, cipher, compression Version, session id, cipher, compression.
2.	Certificate Server - key - exchange Certificate - request Server - done	Chain of X.509 V3 certificates. Parameters, signature. Type, authorities. Null
3.	Certificate - verify	Signature
4.	Client - key - exchange finished.	Parameters, signature hash value.

Table 4.10.1 SSL handshake protocol message types

Fig. 4.10.3 shows handshake protocol action.

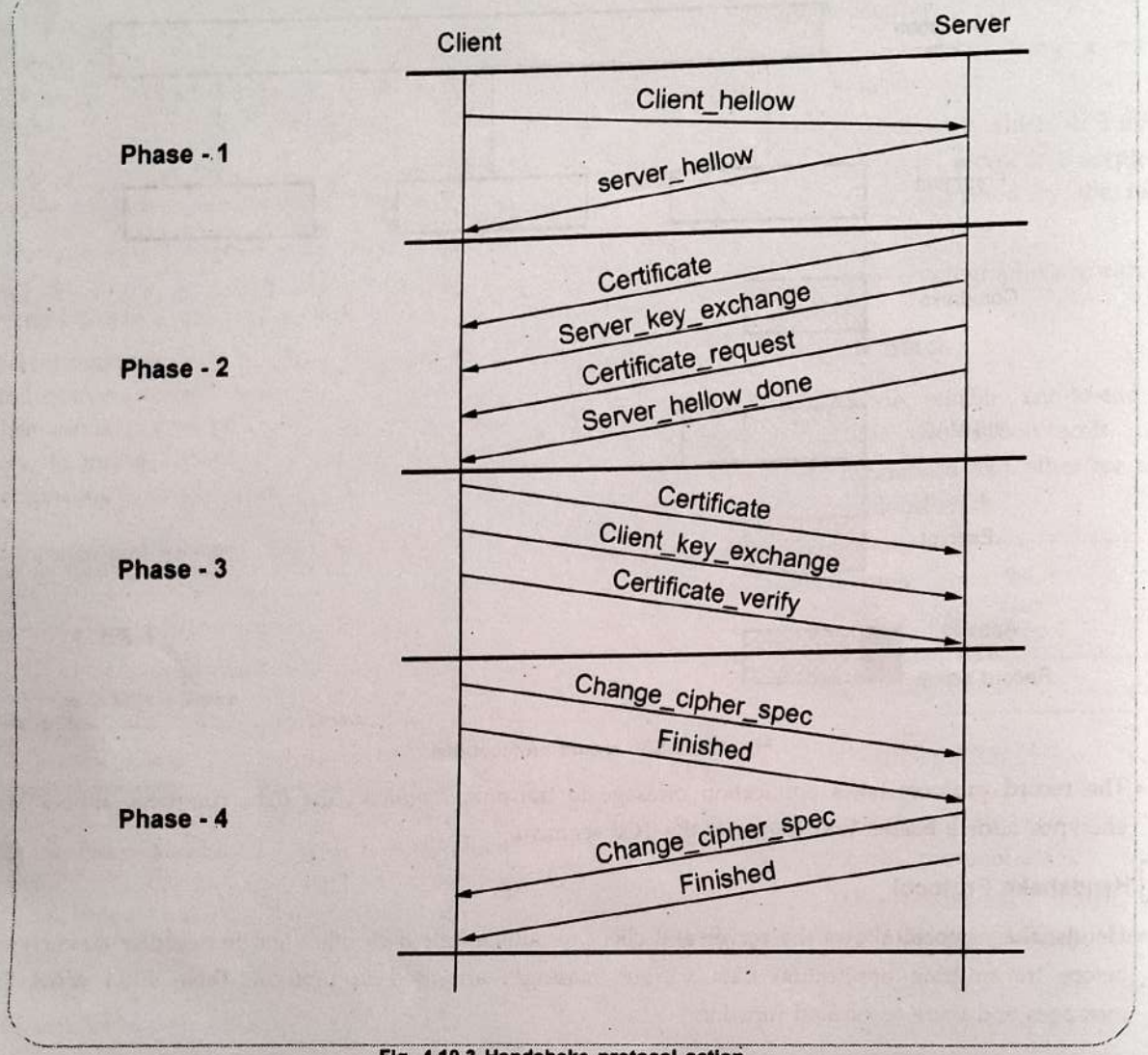


Fig. 4.10.3 Handshake protocol action

4.10.4 Change Cipher Spec Protocol

- The change cipher spec protocol is used to change the encryption being used by the client and server. It is normally used as part of the handshake process to switch to symmetric key encryption.
- This protocol consists of a single message which consists of a single byte with the value 1.
- The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.
- The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed under the current CipherSpec. The message consists of a single byte of value 1.
- The change cipher spec message is sent by both the client and server to notify the receiving party that subsequent records will be protected under the just-negotiated CipherSpec and keys.
- When the client or server receives a change cipher spec message, it copies the pending read state into the current read state.
- When the client or server writes a change cipher spec message, it copies the pending write state into the current write state.
- The client sends a change cipher spec message following handshake key exchange and certificate verify messages (if any), and the server sends one after successfully processing the key exchange message it received from the client.

4.10.5 Alert Protocol

- The Alert Protocol is used to convey SSL-related alerts to the peer entity.
- Alert messages are encrypted and compressed, as specified by the current connection state.
- Alert messages with a level of fatal, result in the immediate termination of the connection.
- In this case, other connections corresponding to the session may continue, however the session identifier must be cancel, preventing the failed session from being used to establish new connections.
- Each message in this protocol consists of two bytes. The first byte takes the value warning (1) or fatal (2) to convey the severity of the message.
- If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert.

4.10.6 Comparison between IPSec and SSL

Sr. No.	Parameters	IPSec	SSL
1.	Position in the OSI model	Internet layer	Between transport and application layers
2.	Configuration	Complex	Simple
3.	NAT	Problematic	No problem
4.	Software location	Kernel area	User area
5.	Firewall	Not friendly	Friendly
6.	Installation	Vender non-specific	Vender specific
7.	Interoperability	Yes	No
8.	Deploy	More expensive to deploy, support and maintain	Less costly to deploy and maintain

4.10.7 Comparison of SSL and TLS

Sr. No.	SSL	TLS
1.	In SSL the minor version is 0 and the major version is 3.	In TLS, the major version is 3 and the minor version is 1.
2.	SSL use HMAC algorithm except that the padding bytes concatenation.	TLS makes use of the same algorithm the padding bytes concatenation.
3.	SSL supports 12 various alert codes.	TLS supports all of the alert codes defined in SSL 3 with the exception of no certificate.

Review Questions

1. *Explain secure socket layer handshake protocol in brief.*
2. *Explain the operation of Secure Socket Layer (SSL) protocol in detail.*
3. *Describe the operation of secure socket layer protocol in detail.*

4.11 Electronic Mail Security

- Email security describes various techniques for keeping sensitive information in email communication and accounts secure against unauthorized access, loss, or compromise.
- Email remains a key productivity tool for today's organizations, as well as a successful attack vector for cyber criminals.

4.11.1 PGP

- PGP stands for Pretty Good Privacy. It was developed originally by Phil Zimmerman. However, in its incarnation as OpenPGP, it has now become an open standard. PGP is open-source. Although PGP can be used for protecting data in long-term storage, it is used primarily for email security.
- PGP is a complete e-mail security package that provides privacy, authentication, digital signatures, and compression all in an easy to use form.
- The complete package, including all the source code, is distributed free of charge via the Internet. Due to its quality, zero price, and easy availability on UNIX, Linux, Windows and Mac OS platforms, it is widely used today.
- PGP encrypts data by using a block cipher called IDEA, which uses 128-bit keys. IDEA is similar to DES and AES. Key management uses RSA and data integrity uses MDS.

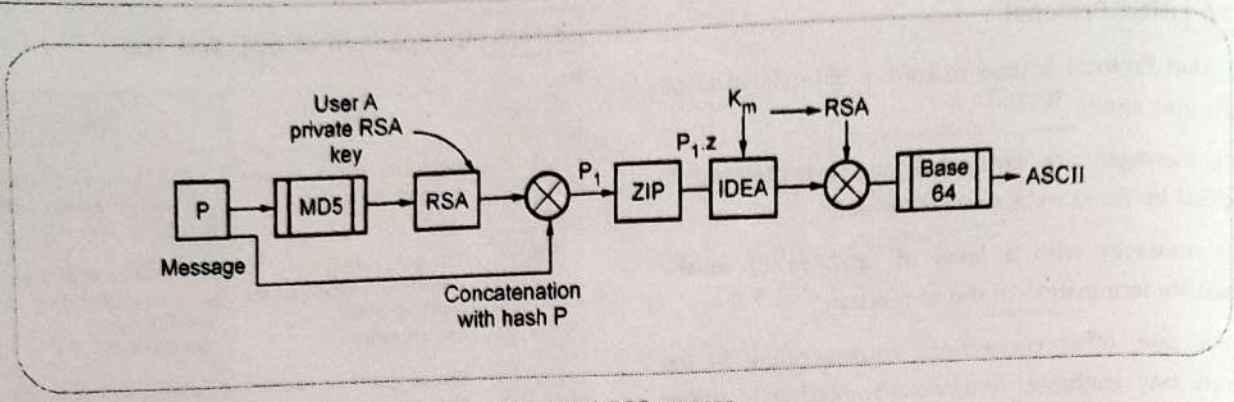


Fig. 4.11.1 PGP process

Characteristics of PGP

1. PGP is available free world wide.
2. PGP can run on various platform windows, UNIX and machintosh.
3. The algorithms used are extremely secure.
4. World wide acceptability.
5. PGP is not developed and controlled by government or standard organization.
6. PGP is on an Internet Standards track.

PGP works as follows

- Suppose user A wants to send a message (P) to user B in a secure way. Both the user have private and public RSA keys. Each user knows the other's user public key. User A uses PGP program for security purpose. At sender side i.e. at user A, PGP apply the hash function to the plain text message using MD5 and that message is encrypted. After encrypting again apply hash function using own private RSA key. Fig. 4.11.1 shows this process.
- When message is received by user B, he decrypts the hash with user A public key and verifies that the hash is correct. MD5 is the difficult to break. The encrypted hash and original message are concatenated into a single message P_1 and compressed using the ZIP program (P_1.Z).
- Using 128-bit IDEA message key (K_m), the ZIP program is encrypted with IDEA. Also K_m is encrypted with user B's public key (B_P). These two components are then concatenated and converted to base64.
- When this is received by user B, he reverses the base64 encoding and decrypts the IDEA key using his private RSA key. Using this key, user B decrypts the message to get P_1.Z. After decompressing P_1.Z, user B gets the plaintext message.
- For getting correct message, user B separates the plaintext from hash and decrypts the hash using user A public key. If the plaintext hash agrees with his own MD5 computation, user B knows that P is the correct message and that message came from user A.

Notation used in PGP

K_S = Session key used in conventional encryption scheme

PR_a = Private key of user A, used in public key encryption scheme

PU_a = Public key of user A, used in public key encryption scheme

EP = Pubilc key encryption

DP = Public key decryption

EC = Conventional encryption

DC = Conventional decryption

H = Hash function

|| = Concatenation

Z = Compression using ZIP algorith

R64 = Conversion to radix 64 ASCII format

4.11.1.1 PGP Operation

- PGP operation involves five different services.
  1. Authentication
  2. Confidentiality
  3. Compression
  4. E-mail compatibility
  5. Segmentation.

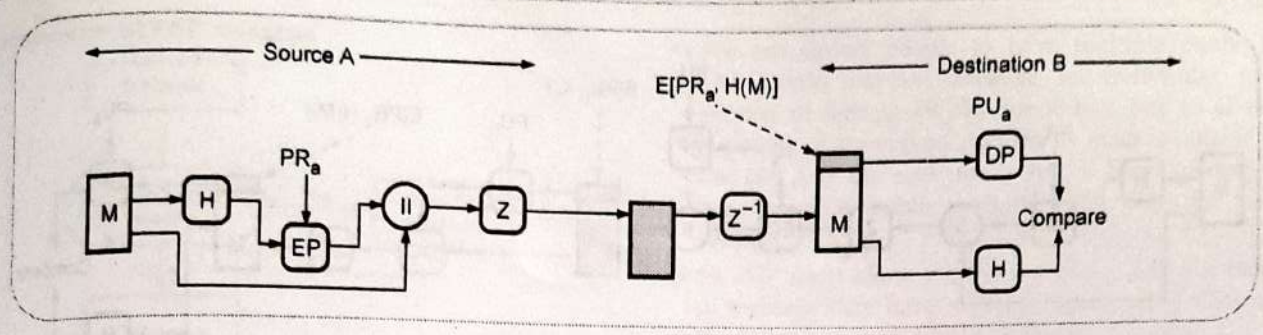


Fig. 4.11.2 Authentication

1. Authentication

- Signatures are attached to the message or file are detached signatures are also supported and are stored and transmitted separately from the message it signs.
- The digital signature is generated by either
  i) SHA-1 and RSA
  ii) DSS/SHA-1
- Sender authentication consists of the sender attaching his/her digital signature to the email and the receiver verifying the signature using public-key cryptography. Here is an example of authentication operations carried out by the sender and the receiver :
  1. At the sender's end, the SHA-1 hash function is used to create a 160-bit message digest of the outgoing email message.
  2. The message digest is encrypted with RSA using the sender's private key and the result prepended to the message. The composite message is transmitted to the recipient.
  3. The receiver uses RSA with the sender's public key to decrypt the message digest.
  4. The receiver compares the locally computed message digest with the received message digest.
- The description was based on using a RSA/SHA based digital signature. PGP also support DSS/SHA based signature. DSS stands for Digital Signature Standard. PGP also supports datached signatures that can be sent separately to the receiver. Detached signatures are also useful when a document must be signed by multiple individuals.
- Fig. 4.11.2 shows an authentication only.

2. Confidentiality

- Confidentiality is provided by encrypting messages to be transmitted. The algorithms used for encrypties are CAST-128, IDEA, 3DES with multiple keys.
- Only a portion of plaintext is encrypted with each key and there is no relationship with keys. Hence, the public key algorithm is secure.
- This service can be used for encrypting disk files. As you'd expect, PGP uses symmetric-key encryption for confidentiality. The user has the choice of three different block-cipher algorithms for this purpose : CAST-128, IDEA, or 3DES, with CAST-128 being the default choice.
  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 pulic 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.
- Fig. 4.11.3 shows a confidentiality operation.

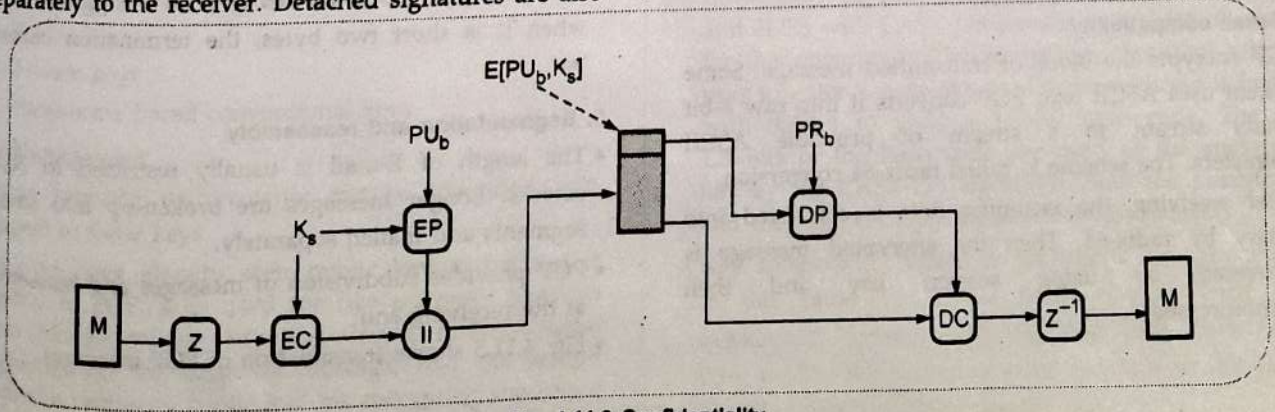


Fig. 4.11.3 Confidentiality

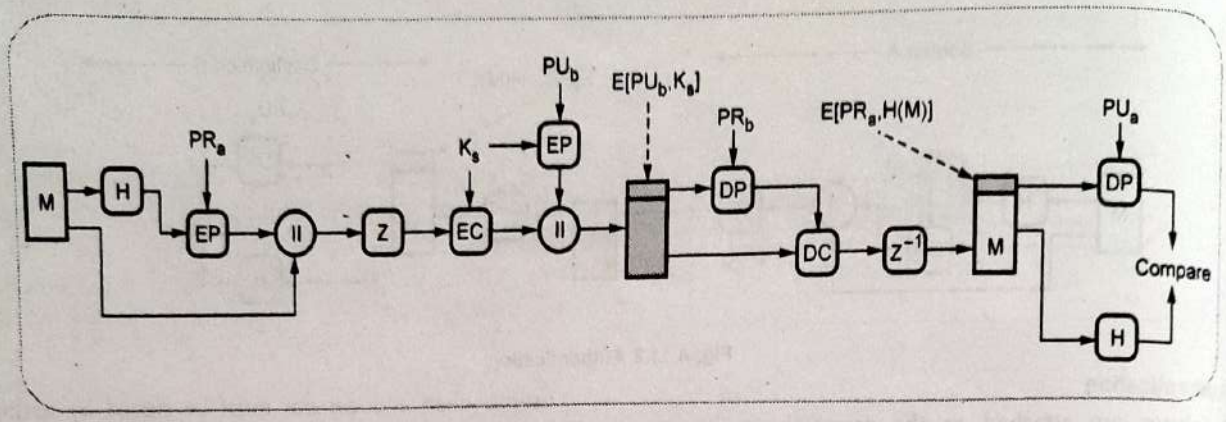


Fig. 4.11.4 Confidentiality and authentication

Confidentiality and Authentication

- May be both services used same message
  a. Create signature for plain text and attach to message
  b. Encrypt both message and signature using CAST - 128 or IDEA or TDEA
  c. Attach RSA encrypted session key
- Fig. 4.11.4 shows confidentiality and authentication.
- When both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and then encrypts the session key with the recipient's public key.

3. Compression

- Before encryption, the message alongwith signature is compressed. Compression of message saves space and ease of transmission. PGP makes use of a compression package called ZIP. Another algorithm lampd-ZIV LZ77 is also used in zip compression scheme.
- By default PGP compresses the email message after applying the signature but before encryption. This is to allow for long-term storage of uncompressed messages along with their signatures. This also decouples the encryption algorithm from the message verification procedures.
- Compression is achieved with the ZIP algorithm.

4. E-mail compatibility

- PGP encrypts the block of transmitted message. Some system uses ASCII text, PGP converts it into raw 8-bit binary stream to a stream of printable ASCII characters. The scheme is called radix-64 conversion.
- After receiving, the incoming data is converted into binary by radix-64. Then the encrypted message is recovered by using session key and then decompressed.
- Since encryption, even when it is limited to the signature, results in arbitrary binary strings, and since many email systems only permit the use of ASCII characters, we have to be able to represent binary data with ASCII strings.
- PGP uses radix-64 encoding for this purpose.
- Radix-64 encoding, also known as Base-64 encoding has emerged as probably the most common way to transmit binary data over a network. It first segments the binary stream of bytes (the same thing as bytes) into 6-bit words.
- The 2^6 = 64 different possible 6-bit words are represented by printable characters as follows : The first 26 are mapped to the uppercase letters A through Z, the next 26 to the lowercase a through z, the next 10 to the digits 0 through 9, and the last two to the characters / and +. This causes each triple of adjoining bytes to be mapped into four ASCII characters.
- The Base-64 character set includes a 65^{th} character, '=', to indicate how many characters the binary string is short of being an exact multiple of 3 bytes. When the binary string is short one byte, that is indicated by terminating the Base-64 string with a single '='. And when it is short two bytes, the termination becomes '=='.

5. Segmentation and reassembly

- The length of E-mail is usually restricted to 50,000 octecks. Longer messages are broken-up into smaller segments and mailed separately.
- PGP provides subdivision of messages and reassembly at the receiving end.
- Fig. 4.11.5 shows transmission of PGP messages.

Transmission of PGP message

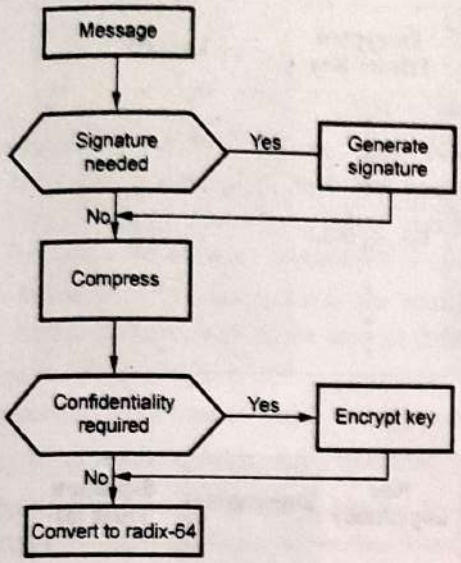


Fig. 4.11.5 Transmission of PGP message

Reception of PGP message

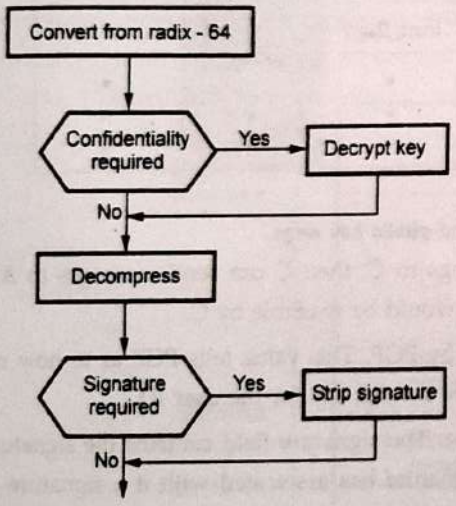


Fig. 4.11.6 Reception of PGP message

4.11.1.2 Cryptographic Keys and Key Rings

- PGP makes use of four types of keys
  1. One time session conventional keys
  2. Public keys
  3. Private keys
  4. Passphrase based conventional keys.

Key Management

- Three separate requirements can be identified with respect to these keys.
- As you have already seen public key encryption is central to PGP. It is used for two purposes : sender uses his/her private key for placing his/her digital signature on the outgoing message, and the sender uses the receiver's public key for encrypting the secret session key.
- We can expect people to have multiple public keys. This could happen because an individual in the process of retiring an old public key, but, to allow for a period of transition, decides to make available both the old and the new for a while. Some people may also choose to public multiple public keys for various reasons.
- So PGP must allow for the possibility that the receiver of message may have multiple public keys. This raises the following procedural questions :
  1. Let's say PGP uses one of the public keys made available keys that the sender has at the recipient know which public key it is.
  2. Let's say that the sender uses one of the multiple private keys that the sender has at his/her disposal for signing the message, how does the recipient know which of the corresponding public keys to use ?
- Both of these problems can be gotten around by the sender also sending along the public key used. The only problem here is that it is wasteful in space because the RSA public keys can be hundreds of decimal digits long.
- The PGP protocol solves this problem by using the notion of a relatively short **key identifiers (key ID)** and requiring that every PGP agent maintain its own list of private / public keys, along with their key identifiers, and a list of public keys, along with their associated key identifiers, for all the email correspondents.
- The former list is known as the private key ring and the latter as the public key right. The keys for a particular user are uniquely identifiable through a comination of the user ID and the key ID. The key ID associated with a public key consists of its least significant 64-bits.
- Going back to private key ring for security reasons, PGP stores the private keys in the table in an encrypted form so that the keys are only accessible to the user who owns them. PGP can use any of the three block ciphers at its disposal, CAST-128, IDEA and 3DES with CAST-128 serving as the default choice, for this encryption. The encryption algorithm asks the user to enter a pass-phrase. The pass-phrase is hashed with SHA-1 to yield a 160-bit hash code. The first 128-bits of the hash code are used as the encryption key by the CAST-128 algorithm. Both the pass-phrase and the hash code are immediately discarded.
- **Key Rings** with regard to the public key ring shown in the Table 4.11.1, the fields Owner Trust, Key Legitimacy, Signature and Signature Trust are to assess how much and siganture trust to place in the public keys belonging to other people.

Private Key IDs

Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(Pi)}[KR_i]$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Private Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust (s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$KU_i \bmod 2^{64}$	KU_i	trust_flag i	User i	trust_flag i		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

*= Field used to index table

Table 4.11.1 General structure of private and public key rings

- If A has B's public key in the ring, but the key really belongs to C, then C can send messages to A and forge B's signature and any encrypted messages from A to B would be readable by C.
- The values in the key legitimacy field column are computed by PGP. This value tells PGP as to how much trust to place in the public key in the corresponding row to be a valid key for the user ID.
- The entry stored in the public key field is actually a certificate. The signature field contains the signature of one or more certifying authorities on the certificate. Each signature has assoicated with it a signature trust field value that indicated how much trust PGP has in the signer of the certificate. The value for the key legitimacy field is derived from the value stored for the signature trust field.
- The entry in the owner trust field of the public-key-ring table indicates the extent to which the owner of a particular pulic key can be trusted to sign other certificates. This value is assigned by the user to whom the public-key-ring belongs.

4.11.1.3 Message Format

- The Fig. 4.11.7 shows the general format of a PGP message. As the figure shows, a PGP message consists of three components :

a) Session key component b) Signature component c) Actual email message

Notation used in message format

$E(PU_b, \bullet)$ = Encyrption with user b's public key

$E(PR_a, \bullet)$ = Encryption with user a's private key

$E(K_s, \bullet)$ = Encryption with session key

ZIP = Zip compression function

R64 = Radix-64 conversion function

- Perhaps the only unexpected entry is the leading two bytes of message digest. This is to enable the recipient to determine that the correct public key was used to decrypt the message digest for authentication. These two octets also server as a 16-bit frame check sequence for the actual email message. The message digest itself is calculated using SHA-1.
- **Message component** includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation.

- Signature component consists of

a. **Timestamp** : The time at which the signature was made.

b. **Key ID of sender's public key** : Identifies the public key that should be used to decrypt the message digest.

c. **Leading two octets of message digest** : To enable the recipient to determine if the correct public key was used to decrypt the message digest for authentication, by comparing this plain text copy of the first two octets with the first two octets of the decrypted digest.

d. **Message digest** : The 160-bit SHA-1 digest encrypted with the sender's private signature key.

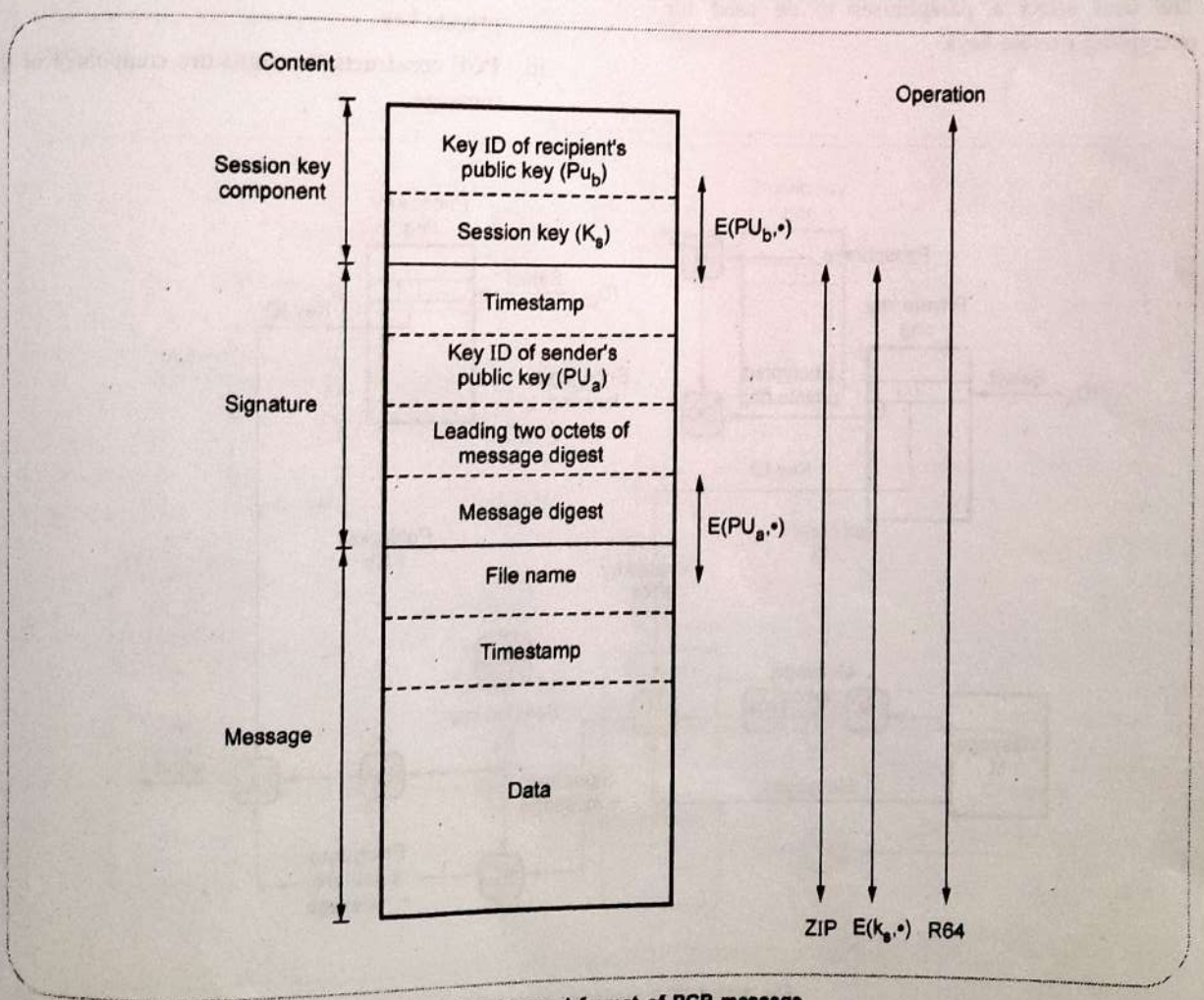


Fig. 4.11.7 General format of PGP message

- Session key component includes the session key and the identifier of the recipient's public key that was used by the sender to encrypt the session key.
- In the table, each row represents one the public / private key pairs owned by this user. Each row contains the following entries :
  1. **Timestamp :** The date or time when this key pair was generated.
  2. **Key ID :** The least significant 64 bits of the public key for this entry.
  3. **Public key :** The public key portion of the pair.
  4. **Private key :** The private key portion of the pair; this field is encrypted.
  5. **User ID :** This will be the user's e-mail address or to reuse the same user ID more than one.
- The private key itself is not stored in the key ring. Rather, this key is encrypted using CAST-128. The procedure is as follows :
  1. The user select a passphrases to be used for encrypting private keys.
  2. When the system generates a new public / private key pair using RSA, it ask the user for the passphrases. A 160-bit hash code is generated from the pass-phrase using SHA-1.
  3. The system encrypts the private key using CAST-128 with the 128-bits of the hash code as the key.
- PGP will retrieve the encrypted private key, generate the hash code of the pass-phrase, and decrypt the encrypted private key using CAST-128 with the hash code.

4.11.1.4 PGP Message Generation

- Fig. 4.11.8 shows PGP message generation.
- The sending PGP entity performs the following steps :
  a) Signs the message :
     i. PGP gets sender's private key from key ring using its user id as an index.
     ii. PGP prompts user for pass-phrase to decrypt private key.
     iii. PGP constructs the signature component of the message.

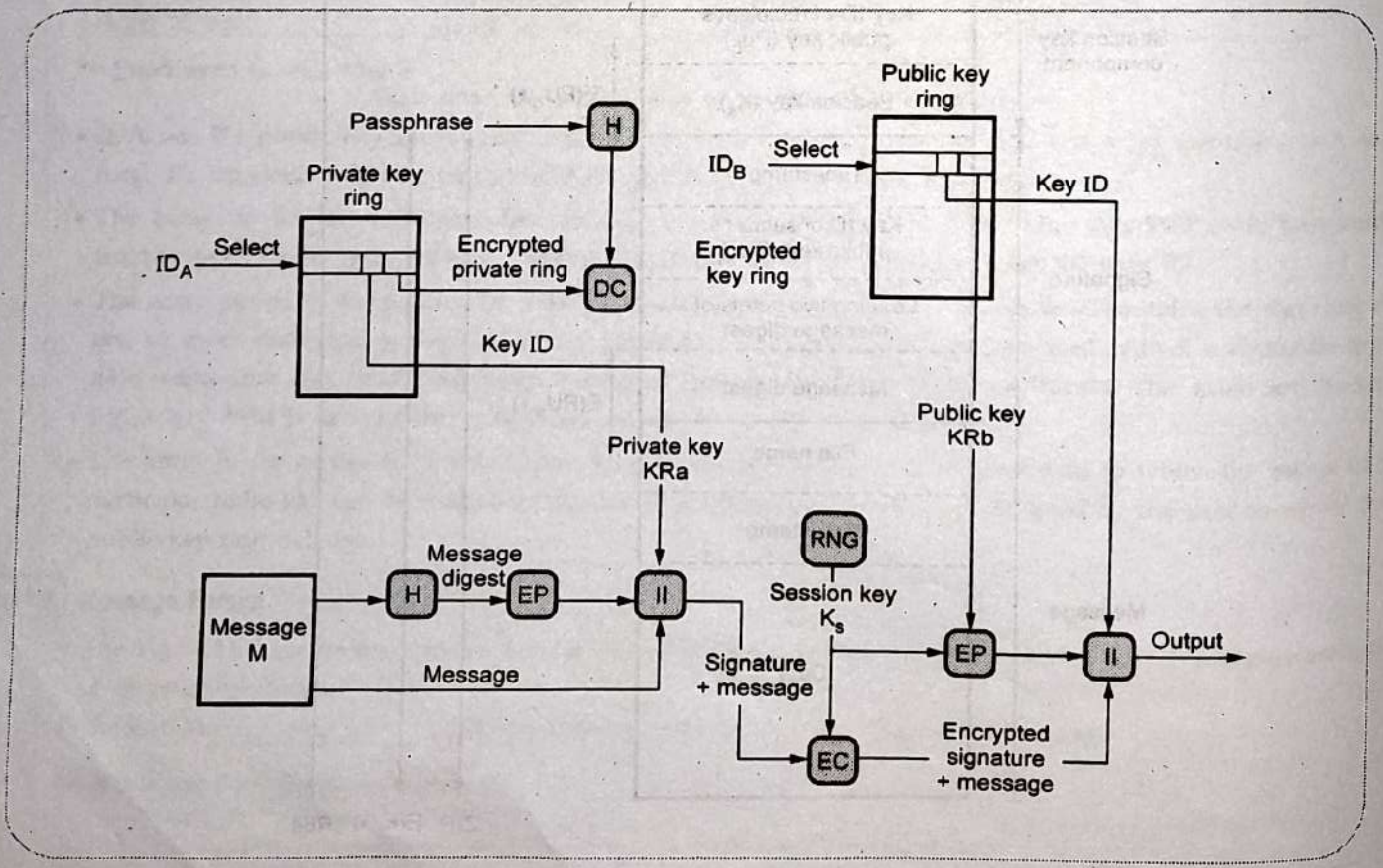


Fig. 4.11.8 PGP message generation

b) Encrypts the message :

i. PGP generates a session key and encrypts the message.

ii. PGP retrieves the receiver public key from the key ring using its user id as an index.

iii. PGP constructs session component of message.

4.11.1.5 PGP Message Reception

• Fig. 4.11.9 shows PGP message reception.

The receiving PGP entity performs the following steps :

a) Decrypting the message :

i. PGP get private key from private-key ring using Key ID field in session key component of message as an index.

ii. PGP prompts user for pass-phrase to decrypt private key.

iii. PGP recovers the session key and decrypts the message.

b) Authenticating the message :

i. PGP retrieves the sender's public key from the public-key ring using the Key ID field in the signature key component as index.

ii. PGP recovers the transmitted message digest.

iii. PGP computes the message for the received message and compares it to the transmitted version for authentication.

4.11.1.6 Concept of Trust

• PGP uses trust field for trust information. These fields are

1. Key legitimate field
2. Signature trust field
3. Owner trust field.

1. Key legitimate field : • Key legitimate field indicates the validity of the public key. i.e. extent to which PGP will trust.

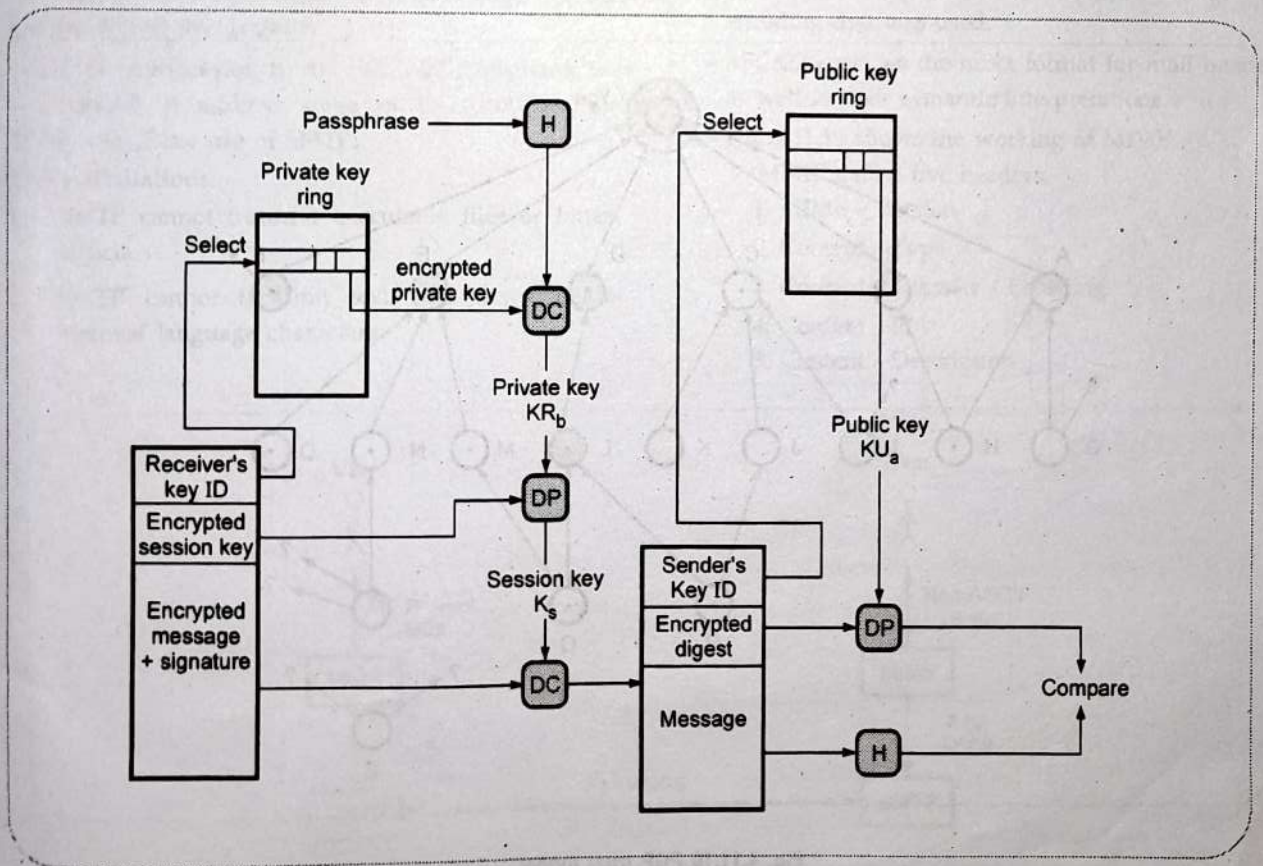


Fig. 4.11.9 PGP message reception

2. Signature trust field : • It indicates the degree to which PGP user trusts the signer to certify public keys.

3. Owner trust field : • It indicates the degree to which the public key is trusted to sign other public key certificates. This level is assigned by user.

4.11.1.7 Trust Processing Operation

- On the public key ring, user A inserts a new public key, then PGP assign a value to the trust flag which is associated with the owner of this public key. If the owner is user A, then this public key also appears in the private key ring and the value of ultimate trust is automatically assigned to the trust field.
- If user A is not the owner, PGP asks user A for his assessment of the trust to be assigned to the owner of this key, and user A must enter the desired level.
- The user can specify that this owner is unknown, untrusted or completely trusted.
- When the new public key is entered, one or more signatures may be attached to it.
- When a signature is inserted into the entry, PGP searches the public key ring to see if the author of this signature is among the known public key owners.
- The value of the key legitimacy field is calculated on the basis of the signature trust fields present in this entry.
- Fig. 4.11.10 shows PGP trust model example. It is an example of the way in which signature trust and key legitimacy are related. The figure shows the structure of a public key ring. The user has acquired a number of public keys.
- The node labeled "You" refers to the entry in the public key ring corresponding to this user. This key is legitimate and the OWNERTRUST value is ultimate trust.
- In this example, this user has specified that it always trusts the following user to sign other keys : D, E, F, L. This user partially trust users A and B to sign other keys.

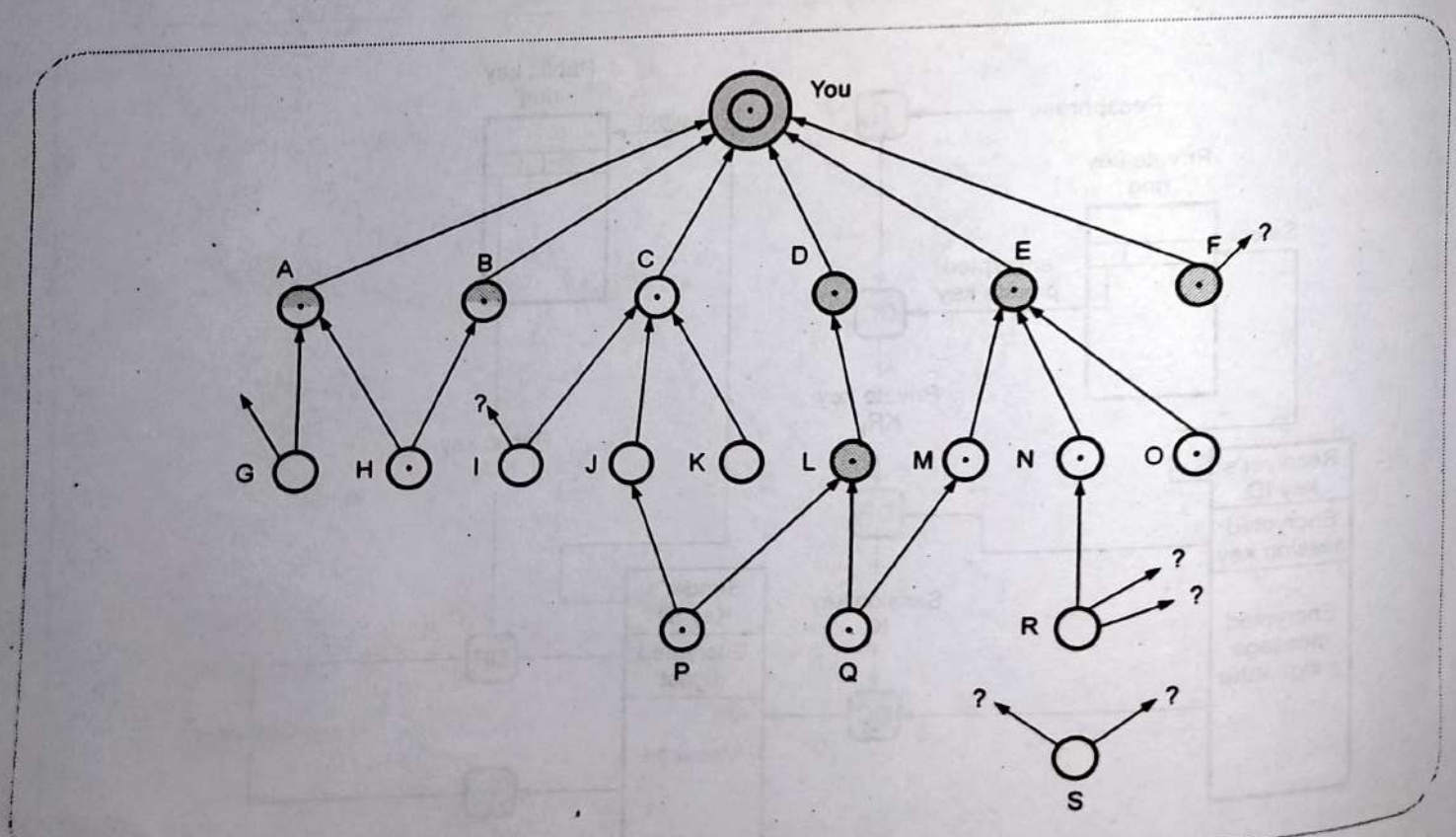
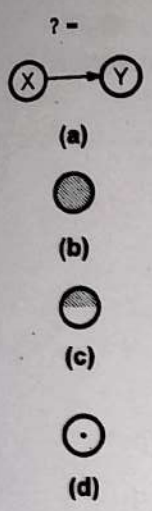


Fig. 4.11.10 PGP trust model

Notation used in above figure



4.11.2 S/MIME

- S/MIME is a Secure / Multipurpose Internet Mail Extension. It is a security enthancement to the MIME Internet e-mail format standard.
- RFC 822 defines a format for text messages that are sent using electronic mail. The RFC 822 standard applies only to the contents.
- MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP.
- **SMTP limitations**
  1. SMTP cannot transmit executable files or binary objects.
  2. SMTP cannot transmit text data that includes national language characters.
  3. SMTP servers may reject mail message over a certain size.
  4. SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages.
  5. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.

4.11.2.1 Multipurpose Internet Mail Extensions

- MIME is a supplementary protocol that allows non-ASCII data to be sent through SMTP.
- MIME defined by IETF to allow transmission of non-ASCII data via e-mail.
- It allows arbitary data to be encoded in ASCII for normal transmission. .
- All media types that are sent or received over the world wide web (www) are encoded using different MIME types.
- Messages sent using MIME encoding include information that describes the type of data and the encoding that was used.
- RFC822 specifies the exact format for mail header lines as well as their semantic interpretations.
- Fig. 4.11.11 shows the working of MIME.
  - MIME define five headers.
    1. MIME - Version
    2. Content - Type
    3. Content - Transfer - Encoding
    4. Content - Id
    5. Content - Description

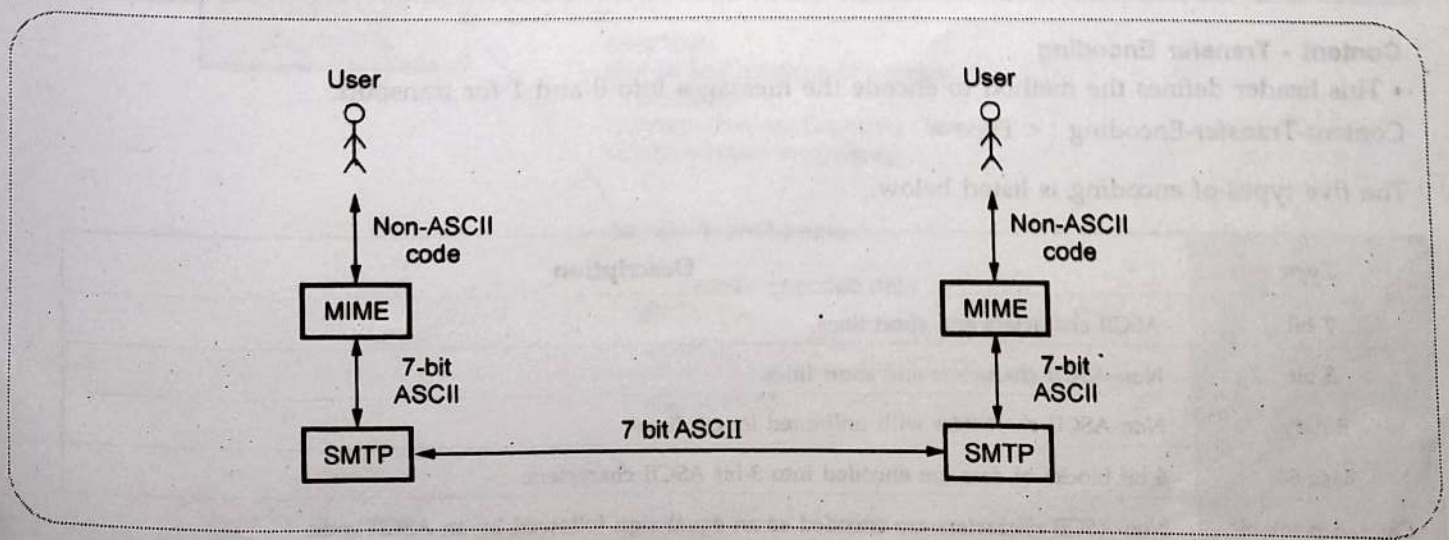


Fig. 4.11.11 MIME

Mail Message Header

- From : iresh@e-mail.com
- To : rupali@sinhgad.edu
- MIME - Version : 1.0
- Content - Type : image/gif
- Content - Transfer - Encoding : base64
- ..... data for the image .....
- ....
- ....

MIME Types and Subtypes

- Each MIME content - type must contain two identifiers :
- - Content type
- - Content subtype
- There are seven standardized content-types that can appear in a MIME content - type declaration.

Type	Subtype	Description
Text	Plain	Unformatted text
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed, but the default is message
	Alternative	Parts are different versions of the same message
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 kHz. (Sound file)
Image	JPEG	Image is in JPGE format
	GIF	Image is in GIF
Message	Partial and external body	An entire e-mail message or an external reference to a message
Application	Postscript	Adobe postscript
	Octet stream	General binary data

Content - Transfer Encoding

- This header defines the method to encode the messages into 0 and 1 for transport.

Content-Transfer-Encoding : < Type >

The five types of encoding is listed below.

Type	Description
7 bit	ASCII characters and short lines.
8 bit	Non-ASCII characters and short lines.
Binary	Non-ASCII characters with unlimited length lines.
Base 64	6 bit blocks of data are encoded into 8 bit ASCII characters.
Quoted printable	Non-ASCII characters are encoded as an equal sign followed by an ASCII code.

Mail Message Format

- SMTP requires all data to be 7-bit ASCII characters and all non-ASCII data must be encoded as ASCII strings.
- Additional lines in the message header declare MIME content type.

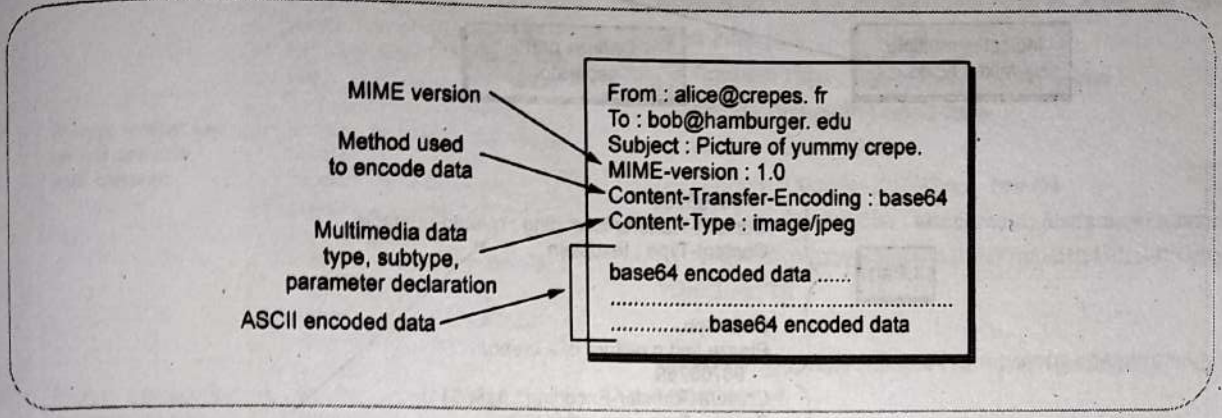


Fig. 4.11.12

4.11.2.2 Message Headers

- The message headers include the addresses of the receiver and the sender. Each header consists of the type of header, a colon, and the content of the header. Following is the sample of the complete header for a message.

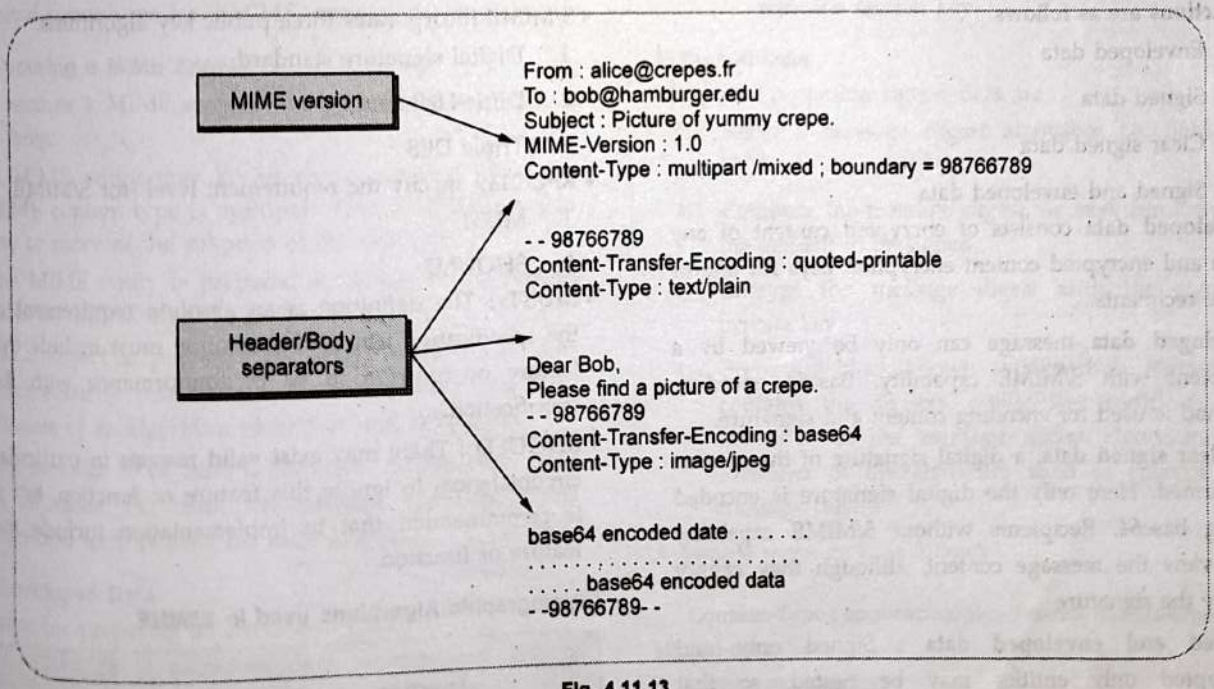


Fig. 4.11.13

Multipart Type

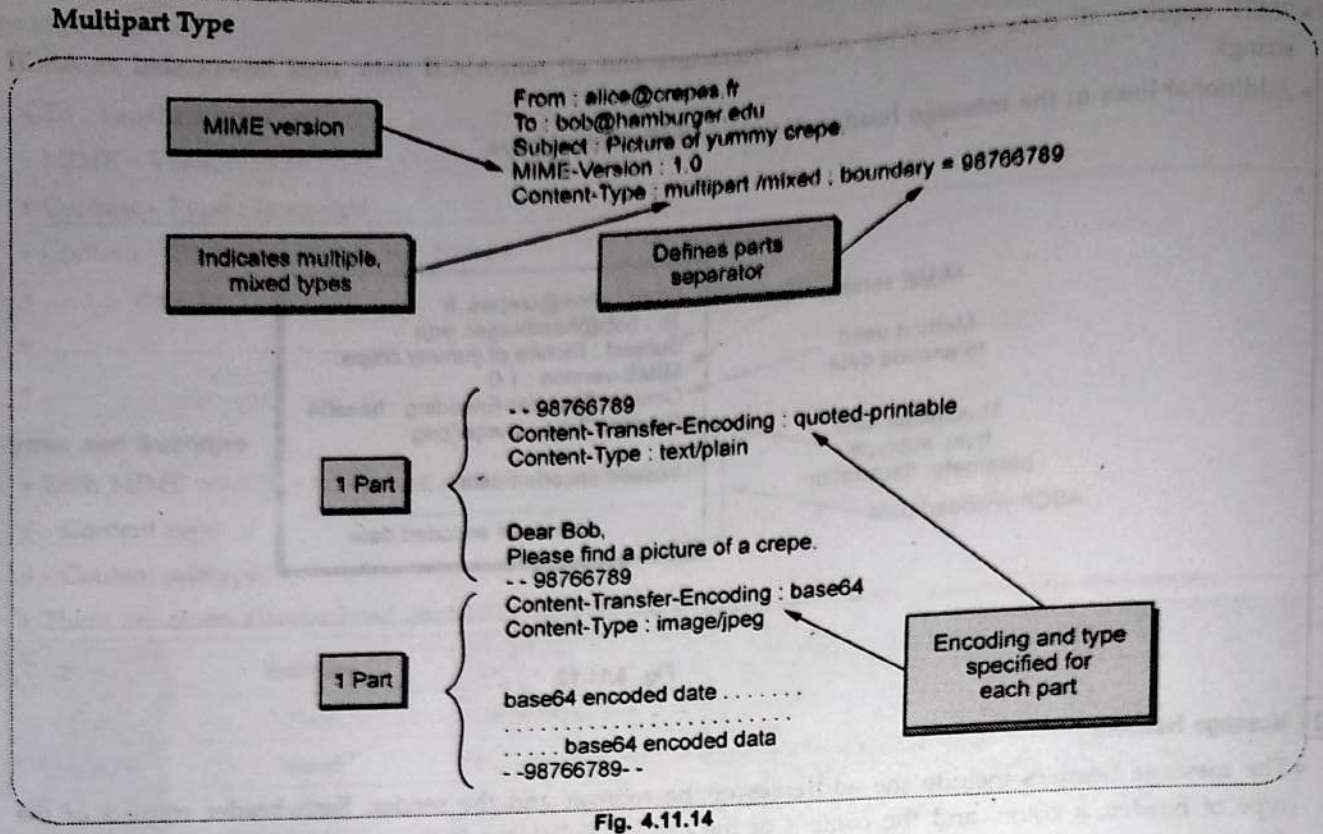


Fig. 4.11.14

4.11.2.3 S/MIME Functionality

- **Functions** are as follows
  1. Enveloped data
  2. Signed data
  3. Clear signed data
  4. Signed and enveloped data
- **Enveloped data** consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
- **A singed data** message can only be viewed by a recipient with S/MIME capability. Base64 encoding method is used for encoding content and signature.
- In **clear signed data**, a digital signature of the content is formed. Here only the digital signature is encoded using base64. Recipients without S/MIME capability can view the message content, although they cannot verity 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.

4.11.2.4 Cryptographic Algorithms in S/MIME

- S/MIME incorporates three public key algorithms.
  1. Digital signature standard
  2. Diffie-Hellman
  3. Triple DES
- RFC 2119 specify the requirement level for S/MIME.
  1. MUST
  2. SHOULD
- **MUST** : The definition is an absolute requirement of the specification. An implementation must include this feature or function to be in comformance with the specification.
- **SHOULD** : There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that in implementation include the feature or function.

Cryptographic Algorithms used in S/MIME

Sr. No.	Function	Requirement
1.	Create message digest to be used in forming a digital signature.	a) MUST support SHA-1 and MD5. b) SHOULD use SHA-A .

2.	Encrypt message digest to form digital signature.	a) Sending and receiving agents MUST support DSS. b) Sending agents SHOULD support RSA encryption. c) Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
3.	Encrypt session key for transmission with message.	a) Sending and receiving agents MUST support Diffie-Hellman. b) Sending agent SHOULD support RSA encryption with key sizes 512 bits to 1024 bits. c) Receiving agent SHOULD support RSA decryption.
4.	Encrypt message for transmission with one time session key.	a) Sending agents SHOULD support encryption with triple DES and RC2/40. b) Receiving agents SHOULD support decryption using 3DES and MUST support decryption with RC2/40.

4.11.2.5 S/MIME Messages

General procedures for S/MIME message preparation.

1. Securing a MIME Entity

- It secures a MIME entity with a signature, encryption, or both.
- A MIME entity may be an entire message, or if the MIME content type is multipart, then a MIME entity is one or more of the subports of the message.
- The MIME entity is prepared according to the normal rules for MIME message preparation.
- PKCS object is prepared by using MIME entity plus some security related data. Security related data item consists of an algorithm identifiers and certificates.
- The message to be send is converted to canonical form in all cases. For multipart message, the appropriate canonical form is used for each subpart.

2) Enveloped Data

- Steps for preparing an enveloped data

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm.
2. For each recipient, encrypt the session key with the recipient's public. RSA key.
3. Prepare a block for each recipient. Block is known as RecipientInfo which contains the sender public key certificate, an identifier of the algorithm used to encrypt the session key and the encrypted session key.
4. Encrypt the message content with the session key.

Enveloped Data is encoded into base64. A sample message is as follows :

Content-Type : application / pkcs7-mime ;
smime-type=enveloped-data;
name=smime.p7m
Content-Transfer-Encoding : base64
Content-Disposition : attachment; filename=smime.p7m
rfvbn756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6

7nHHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V

- To recover the encrypted message, the recipient first strips off the base64 encoding. Then the recipient's private key is used to recover the session key. Finally, the message content is decrypted with the session key.

3) Signed Data

- Steps for preparing signed data are

1. Select a message digest algorithm i.e. SHA or MD5.
2. Compute the message digest, or hash function, of the content to be signed.
3. Encrypt the message digest with the signer's private key.
4. Prepare a block known as signerInfo. SignerInfo contains the signer's public key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest.

A sample message is as follows

Content-Type : application/pkcs7-mime; smime-type = signed-data;
name=smime.p7m
Content-Transfer-Encoding : base64
Content-Disposition : attachment; filename=smime.p7m

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7

77n8HHGT9HG4VQpfyF467GhIGfHfyYT6rfvbnj756tbBghyHhHUujhJhjH

HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGhyHh

6YT64V0GhIGfHfQbnj75

- The recipient first strips off the base64 encoding then the signer's public key is used to decrypt the message digest.

4) Clear Signing

- Clear signing is achieved using the multipart content type with a signed subtype. Message is sent "in the clear" because recipient with MIME capability but not S/MIME capability are able to read the incoming message.
- A multipart/signed message has two parts.
  1. MIME type 2. MIME content type
- If the first part is not 7 bit, then it needs to be encoded using base64 or quoted printable.
- Second part has a MIME content type of application and a subtype of PKCS7 signature.
- Following is a sample message :

```
Content-Type : multipart / signed;
    protocol="application/pkcs7-signature";
    micalg=shal; boundary=boundary42
---- boundary42
Content-Type : text/plain

This is a clear-signed message.
---- boundary42
Content-Type : application/pkcs7-signature;
name=smime.p7s
Content-Transfer-Encoding:base64
Content-Disposition : attachment;filename=sime.p7s
ghyHhHUujhJhjH77n8HHGTrfvbnj756tb9HG4VQpfyF467
GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756t
bB9HGTrfvbnj
n8HHGTrfvhJhH776tbB9HG4VQbnj7567GhIGfHfYT6ghy
HhUuujpfyF4
----boundary42----
```

5) Registation Request

- The certification request includes
  a. Certification Request Info block
  b. Identifier of the public key encryption algorithm
  c. Signature of the certification RequestInfo block
- The certification RequestInfo blocks includes a name of the certificate subject and a bit-string representation of the users public key.

S/MIME content types

Type	Subtype	SMIME Parameter	Description
Multipart	Signed		A clear signed message in two parts : One is the message and the other is the signature
Application	pkcs 7-mime	Signed data	A signed S/MIME entity
	pkcs 7- mime	Enveloped data	An encrypted S/MIME entity
	pkcs 7-mime	Degenerate signed data	An entity containing only public key certificate
	pkcs 7-signature	–	The content type of the signature subpart of a multipart / signed message
	pkcs 10-mime	–	A certificate registration request message

4.11.2.6 S/MIME Certificate Processing

User Agent

- An S/MIME user has several key managed functions to performs :
  1. **Key generation :** A user agent SHOULD generate RSA key pairs with a length in the range of 768 to 1024 bits and MUST NOT generate a length of less than 512 bits.
  2. **Registration :** A user's public key must be registered with a certification authority in order to receive an X.509 public key certificate.
  3. **Certificate storage and retrivel :** A user requires access to a local list of certificate in order to verify incoming signatures and to encrypt outgoing messages.

Verisign Certificates

- Verisign provides a CA service that is intended to be compatible with S/MIME and a variety of other applications. Verisign issues X.509 certificates with the product name verisign Digital ID.

- Each Digital ID contains the following :
  a) Owner's public key
  b) Owner's name
  c) Expiration data of the Digital ID
  d) Serial number of the Digital ID
  e) Name of the certification authority that issued the Digital ID
  f) Digital signature of the certification authority that issued the Digital ID.

4.11.3 PEM

- Primary goal of PEM is to add security services for e-mail users in the internet community. Began in 1985 as an activity of the Privacy and Security Research Group (PSRG) and defined in RFCs 1421/1422/1423/1424.
- It consists of extensions to existing message processing software plus a key management infrastructure.
- Developed by IETF, to add encryption, source authentication and integrity protection to e-mail. Allows both public and secret long-term keys and message key is always symmetric. It also specifies a detailed certification hierarchy.
- Uses symmetric cryptoraphy to provide (optional) encryption of messages.
- The use of X.509 certificates is the base for public key management in PEM.
- This certification hierarchy supports universal authentication of PEM users.
- PEM can be used in a wider range of messaging environments. PEM represents a major effort to provide security for an application that touches a vast number of users within the Internet and beyond.
- PEM was designed to have backward compatibility with existing mail system.
- PEM depends on a successful establishment of the certification hierarchy that underlies asymmetric key management.
- **Problem :** PEM does not support security services to multimedia files (MIME)

PEM Security Services

1. Integrity, which ensures a message recipient that the message has not been modified en route.
2. Authenticity, which ensures a message recipient that a message was sent by the indicated originator.
3. Non-repudiation, which allows a message to be forwarded to a third party, who can verify the identity of the originator.
4. Confidentiality (optional), which ensures a message originator that the message text will be disclosed only to the desingated recipients.

PEM Message Processing

Step 1 :

- Uses the canonicalization specified by SMTP to ensure a uniform presentation syntax among a heterogeneous collection of computer systems.
- The shortcoming is that it restricts the input to 7-bit ASCII.
- The reason is that the Internet e-mail imposes the same restrictions.

Step 2 :

- A MIC is calculated over the canonicalized message to permit uniform verification in the heterogeneous environments.
- The cannonical (padded as required) message text is then (optionally) encrypted using a per-message symmetric key.
- The encryption action is performed only if the message is of type ENCRYPTED.

Step 3 :

- Renders an ENCRYPTED or MIC-ONLY message into a printable form suitable for transmission via SMTP.
- This encoding step transforms the (optionally encrypted) message text into a restricted 6-bit alphabet.
- A MIC-CLEAR messages are not subject to any portion of the third processing step.

PEM Message Types

- ENCRYPTED is a signed, encrypted and encoded (in step 3) message.
- MIC-ONLY is a signed, but not encrypted, encoded message.
- MIC-ONLY is a signed, but not encrypted, and message that is not encoded.
- Specially so it can be sent to a mixed set of recipients, some of whom use PEM and some do not.

PEM Message Delivery Processing (1)

- Recipient receives a PEM message.
- Scans the PEM header for the version and the type (ENCRYPTED, MIC-ONLY, MIC-CLEAR).

- If ENCRYPTED or MIC-ONLY then decode the 6-bit encoding back to ciphertext or canonical plaintext form.
- If ENCRYPTED then decrypt the symmetric message key using the private component of his public key pair and decrypt the message using the symmetric message key.
- Validate the public key of the sender by validating a chain of certificates.
- Validate the digital signature using the public component of the public key of the sender.
- The canonical form is translated into the local representation and presented to the recipient.

Review Questions

1. *What is backdoors and key Esorow in PGP ?*
2. *Explain working of PGP in detail.*
3. *What is S/MIME ? State operation of S/MIME in detail.*
4. *Explain working of S/MIME with secracy and authenticaiton.*
5. *What are the security services provided by PGP ?*

4.12 Secure Electronic Transaction (SET)

- SET is an encryption and security specification develop to protect credit card transactions through Internet SET is not a payment system but a set of security protocols for secured way for payment transactions
- SET is a complex specification defined in -
  1. Business description
  2. Programmer's guide
  3. Formal protocol definition

4.12.1 Services Provided by SET

1. SET provides a secure communication channel among all parties.
2. Provides trust by using X.509V3 digital certificates.
3. Ensures privacy.

4.12.2 Requirements for SET

- For secured payment processing over Internet following are the requirements of SET protocol specifications :
  1. Provide confidentiality of payment and ordering information.
  2. Ensure the integrity of all transmitted data
  3. Provide authentication about card holder
  4. Provide authentication about merchant
  5. Ensure use of best security practices and system design.
  6. Develop a protocol that does not depend on transport security.
  7. Facilitate interoperability between software and network.

4.12.3 Features of SET

1. Confidentiality of information.
2. Integrity of data.
3. Account authentication of card holder.
4. Merchant authentication.

4.12.4 SET Participants

- Following are the participants of SET system.
  a) Card holder b) Merchant
  c) Issuer d) Acquirer
  e) Payment gateway f) Certification authority
- The sequence of event in SET system is as follows :
  1. Customer opens an account
  2. Customer receives a certificate
  3. Merchant's certificate
  4. Customer places an order
  5. Verification of merchant
  6. Order and payment sent
  7. Request for payment authorization by merchant
  8. Merchant confirms order.
  9. Merchant provides goods or service
  10. Merchant requests payment

4.12.5 Key Technologies of SET

- **Confidentiality of information :** DES.
- **Integrity of data :** RSA digital signatures with SHA-1 hash codes.
- **Cardholder account authentication :** X.509v3 digital certificates with RSA signatures.
- **Merchant authentication :** X.509v3 digital certificates with RSA signatures.
- **Privacy :** Separation of order and payment information using dual signatures.

4.12.6 SET Supported Transactions

1. Card holder registration
2. Merchant registration
3. Purchase request
4. Payment authorization
5. Payment capture
6. Certificate query
7. Purchase inquiry
8. Purchase notification
9. Sale transaction
10. Authorization reversal
11. Capture reversal

4.12.7 Dual Signature

- Dual signature is needed for linking two messages that are intended for two different receiver Order Information and Payment Information (OI and PI).
- The operation of dual signature can be summarized as,

$$DS = E(PRc, [H(H(PI) \| (OI)])$$

where,

PRc, is customer's private signature key

PI is payment information

OI is order information

H is Hash function

|| is concatenation

E is encryption (RSA)

- Dual signature limit the information on need to know basis i.e. merchant does not need credit card number and bank does not need details of customer order. This provides extra protection interms of privacy.
- Fig. 4.12.1 shows implementation of dual signatures.

4.12.7.1 Why Dual Signature ?

- Suppose that customer send the merchant two messages :

1. The signed Order Information (OI)
2. The signed Payment Information (PI)

- In addition, the merchant passes the Payment Information (PI) to the bank. If the merchant can capture another Order Information (OI) from this customer, the merchange could claim this order goes with the Payment Information (PI) rather than the original. Dual signature confirms the payment is made for specific order.

A] DS Verification by Merchant

- The merchant has the public key of the customer obtained from the customer's certificate.
- Now, the merchant can compute two values :

  H(PIMD || H(OI)

  DKUC[DS]
- Should be equal.

B] DS Verification by Bank

- The bank is in possession of DS, PI the message digest for OI [OIMD], and the customer's public key, then the bank can compute the following :

  H(H(PI) || OIMD)

  DKUC[DS]

4.12.8 Process of SET

4.12.8.1 Purchase Request

- Browsing, selecting, and ordering is done.
- Purchasing involves four messages :

  i) Initiate request

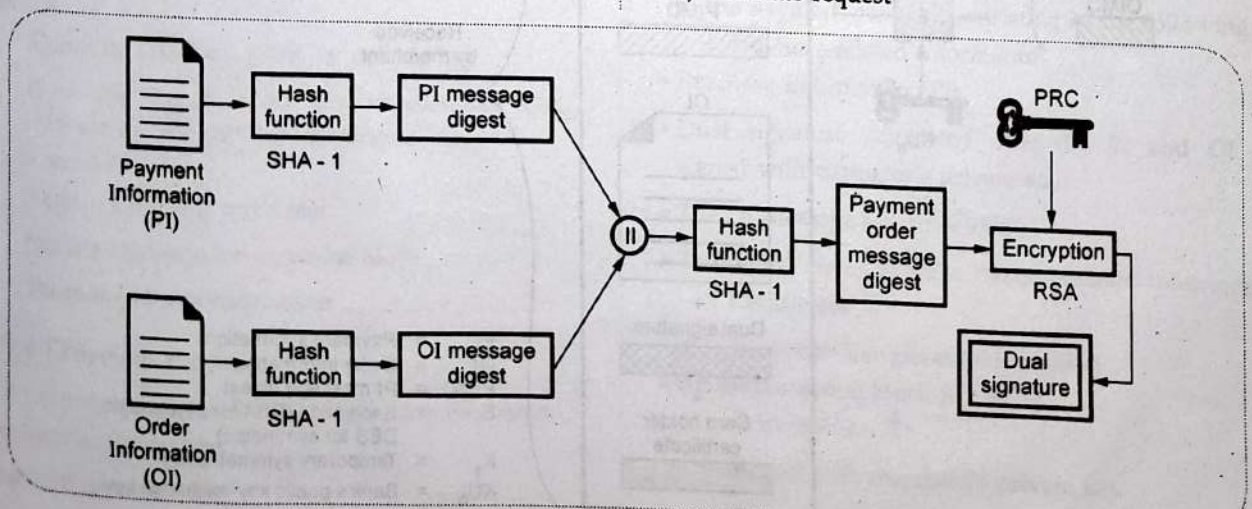


Fig. 4.12.1 Generation of dual signature

ii) Initiate response
iii) Purchase request
iv) Purchase response

I) Initiate Request

- Basic requirements :
  - Cardholder must have copy of certificates for merchant and payment gateway.
- Customer requests the certificates in the initiate request message to merchant.
  - Brand of credit card
  - ID assigned to this request/response pair by customer

II) Initiate Response

- Merchant generates a response
  - Signs with private signature key
  - Include customer nonce
  - Include merchant nonce (returned in next message)
  - Transaction ID for purchase transaction
- In addition ...
  - Merchant's signature certificate
  - Payment gateway's key exchange certificate

iii) Purchase Request

- Cardholder verifies two certificates using their CA_s and creates the OI and PI
- Message includes :
  - Purchase-related information
  - Order-related information
  - Cardholder certificate
- The cardholder generates a one-time symmetric encryption key KS

Merchant Verifies Purchase Request

- When the merchant receives the purchase request message, it performs the following actions :
  - Verify the cardholder certificates by means of its CA signatures.
  - Verifies the dual signature using the customer's public key signature.

Processes the order and forwards the payment information to the payment gateway for authorization.

Sends a purchase response to the cardholder.

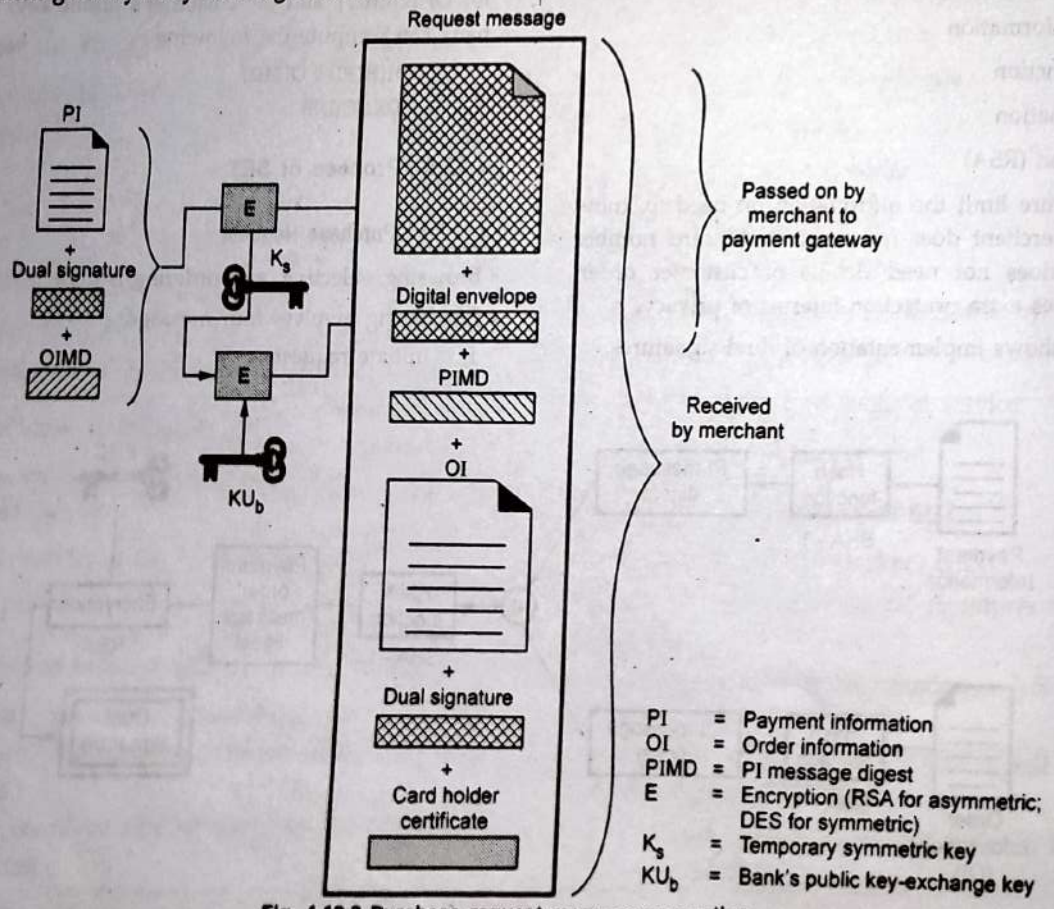


Fig. 4.12.2 Purchase request message generation

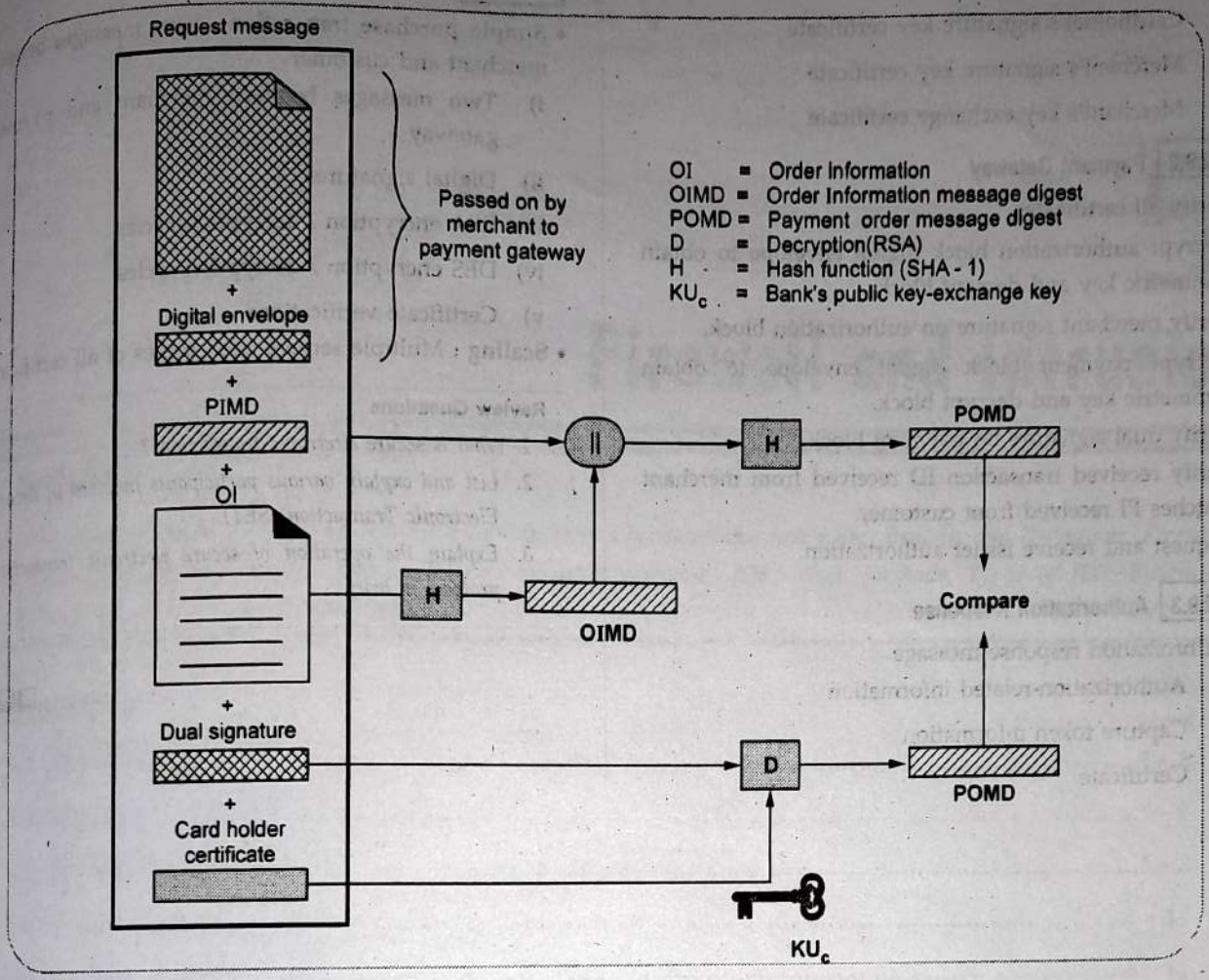


Fig. 4.12.3 Verification of purchase request

iv) Purchase Response Message

- Message that acknowledges the order and references corresponding transaction number.
- Block is,
  - Signed by merchant using its private key
  - Block and signature are sent to customer along with merchant's signature certificate
- Upon reception
  - Verifies merchant certificate
  - Verifies signature on response block
  - Takes the appropriate action

4.12.9 Payment Process

- The payment process is broken down into two steps :

1. Payment authorization
2. Payment capture

4.12.9.1 Payment Authorization

- The merchant sends an authorization request message to the payment gateway consisting of the following :
  - Purchase-related information
- Purchase Information (PI)
- Dual signature calculated over the PI and OI and signed with customer's private key.
- The OI Message Digest (OIMD)
- The digital envelope-authorization-related information
  - Certificates
  - Authorization-related information
- An authorization block including :
  - A transaction ID
  - Signed with merchant's private key
  - Encrypted one-time session key

- Certificates
  1. Cardholder's signature key certificate
  2. Merchant's signature key certificate
  3. Merchant's key exchange certificate

4.12.9.2 Payment Gateway

- Verify all certificates.
- Decrypt authorization block digital envelope to obtain symmetric key and decrypt block.
- Verify merchant signature on authorization block.
- Decrypt payment block digital envelope to obtain symmetric key and decrypt block.
- Verify dual signature on payment block.
- Verify received transaction ID received from merchant matches PI received from customer.
- Request and receive issuer authorization.

4.12.9.3 Authorization Response

- Authroization response message
  - Authorization-related information
  - Capture token information
  - Certificate

4.12.10 SET Overhead

- Simple purchase transaction : Four messages between merchant and customer
  i) Two messages between merchant and payment gateway
  ii) Digital signatures
  iii) RSA encryption / decryption cycles
  iv) DES encryption / decryption cycles
  v) Certificate verifications
- Scaling : Multiple servers need copies of all certificates

Review Questions

1. *What is secure electronic transaction ?*
2. *List and explain various participants involved in Secure Electronic Transaction (SET).*
3. *Explain the operation of secure electronic transaction protocol in brief.*

❑❑❑