Unit IV

CHAPTER 4

Security Requirements

Syllabus

IP Security : Introduction, Architecture, IPV6, IPv4, IPSec protocols, and Operations, AH Protocol, ESP Protocol, ISAKMP Protocol, Oakley determination 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).

Syllabus Topic : Security Introduction

4.1 IP Security - IP Security Overview

→ (SPPU - May 15)

Q. 4.1.1 Define IP sec. (Ref. Sec. 4.1) May 15. 2 Marks
Q. 4.1.2 What is IPSec? (Ref. Sec. 4.1)

- The network connectivity not only gives us authority to access world from computer but at the same time the network lets the outer world access to us in the way that we may not be desire. Any loop hole in our network can make our systems vulnerable and we can be the victim of cybercrime.
- To secure information and to send data through network are both linked to each other. One formal way that network engineer's uses for data communication is the OSI seven layer model for networking.
- The OSI model describes the seven layers of interaction for a system communication in the network.
- Starting from the top most layer data is sent to layer by layer, each layer adding its own information to the original information until the original data and the added content of each layer reach to the physical medium.
- All the layers "communicate" with each other at the sender and receiver side, they need to send pure data and data should be intact with no change at the receiver's side. Reviewing the flow of information through the layers, we observe that all layers depend upon each other so security is important in each layer.
- The OSI security architecture reference model (ISO 7498-2) is designed around the seven layers of OSI reference model (ISO-7498-1), reflecting the different requirements of security in each layer for secure data transfer.
- In TCP/IP protocol suite there are various protocols and techniques used to secure data and ongoing traffic at each layer called as layer wise security concerns. At the network (internet) layer TCP/ IP supports the most significant protocol called Internetworking Protocol/ Internet Protocol (IP).
- **Internet Protocol security (IPSec)** is a framework of open standards for protecting communications over Internet Protocol (IP) networks through the use of cryptographic security services. The most important feature of IPsec is that it can authenticate and encrypt all on-going traffic at the IP level.
- In terms of cryptography Internet Protocol support for sending and receiving encrypted information of any kind without any modification. IPSec provides different

kinds of cryptographic services like confidentiality integrity and authentication.

Table 4.1.1 shows the layers and the perspective security in each layer.

Table 4.1.1

Layers (ISO 7498-1)	ISO 7498-2 Security Model
Application	Authentication
Presentation	Access Control
Session	Non-Repudiation
Transport	Data Integrity
Network	Confidentiality
Data Link	Assurance / Availability
Physical	Notarization / Signature

Syllabus Topic : Architecture

4.2 IP Security Architecture

Q. 4.2.1 Distinguish between tunnel and transport mode in IPSec. Describe briefly how IPSec work. (Ref. Sec. 4.2)

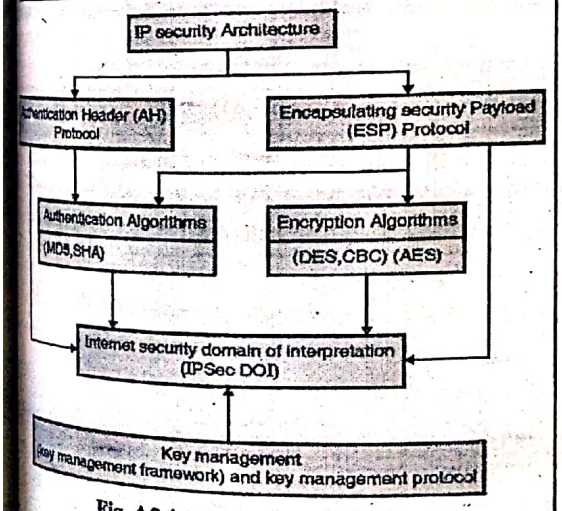

Fig. 4.2.1 : IPSec architecture

IPSec was designed by the Internet Engineering task force IETF. It is a collection of protocols which provides security for a packet at network level. IPSec creates authenticated and confidential packets for network layer also known as IP (Internet protocol layer).

- IPSec provides node to node communication in routing protocols; it provides security to other protocols also which are used for client-server communication in transport layer.
- IPSec defines two protocols as they are backbone of IPSec, are Authentication Header (AH) and Encapsulating Security Payload (ESP) protocol. Architecture of IPSec is shown in Fig. 4.2.1 and following sections defines details on each fields.

1. Authentication Header (AH)

It defines the AH packet format for processing incoming and outgoing packets. AH helps to ensures that authentication and integrity of the data/packets is protected.

2. Encapsulating Security Payload (ESP)

It defines the ESP packet header, which transmits packets in encrypted and unreadable format. ESP helps to ensure that confidentiality, authenticity and integrity of the data is protected.

3. Authentication Algorithms

Use of MD-5 and SHA with Encapsulating Security Payload and Authentication to achieve integrity and protection of data. Hash is attached to the IP header as an integrity checksum.

4. Encryption Algorithms

Few standard encryption algorithms are implemented in IPSec are DES, AES and CBC because of large key size to secure data.

5. Internet security Domain of Interpretation (DOI)

It contains the supporting database of all IP Security protocols, their parameters, all defined algorithms, key size with lifetime and identity of all approved encryption and decryption algorithms.

6. Key Management

As defined earlier key management is used to generate and distribute the keys required for IPSec protocols.

4.2.1 IPSec Modes

IPSec operates in two different modes :

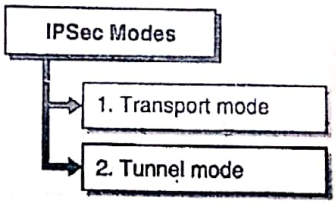

Fig. 4.2.2 : IPSec Modes

1. Transport Mode

- In Transport mode IPSec protects the data that is delivered from transport layer to network layer or in other words we can say that, transport mode protects the payload(a packet consist of controlled information and user data) of network layer.
- It encapsulates the transport layer payload by adding IPSec header and IPSec trailer and sends this encapsulated packet to network layer.
- After that the IP header of network layer is added to that encapsulated payload. IPSec transport mode is responsible for complete delivery of packet (traffic) from one host to another host or from host to gateways called as end-to-end communications.
- End-to-End communications means communications between client machine and a server machine, communications between two routes and from router to gateway is also considered as end-to-end communication. IPSec transport mode is responsible for secure communications between all these devices.
- Transport mode helps to protect user data, also known as IP payload through an AH or ESP header. In transport mode payload of IP packet is encrypted by the IPSec headers and trailers but the IP header information, which is remain unchanged. The payload of an IP packet is protected before it is handled by the network layer as shown in as in Fig. 4.2.3.
- Fig. 4.2.4 shows how the data exchange (end to end security) take place after encrypting payload.

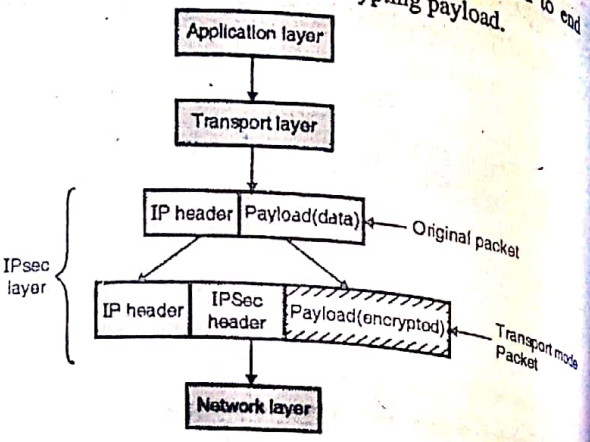

Fig. 4.2.3 : Transport mode

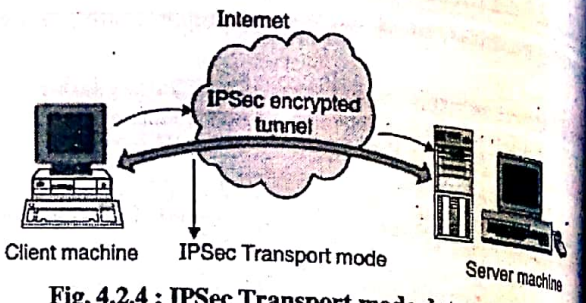

Fig. 4.2.4 : IPSec Transport mode data exchange

2. Tunnel Mode

- In tunnel mode the IPSec protects the entire IP Packet of Network Layer.
- It takes whole IP packet including the header of that IP Packet and applies the IPSec method to the whole packet and adds new IP header.
- IPSec tunnel mode is responsible for network-to-network communications, it encrypt the traffic between routers, gateways or host-to-network and host-to-host communications over the Internet and creates a secure tunnel. IPSec tunnel mode encrypts complete IP packet including IP header and transfer it over network layer (entire original IP packet is encrypted).
- Tunnel mode binds the original IP packet, encrypts it, adds a new IP header and IPSec header and sends it to the other end of IPSec shown in Fig. 4.2.5.
- Fig. 4.2.6 shows IPSec tunnel mode during data exchange process.

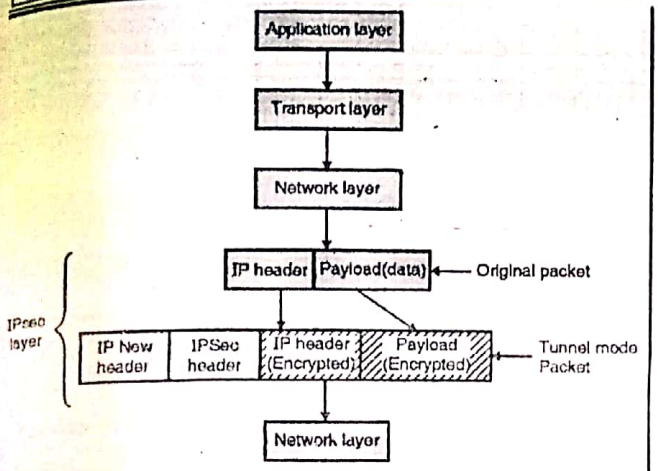

Fig. 4.2.5 : Tunnel Mode

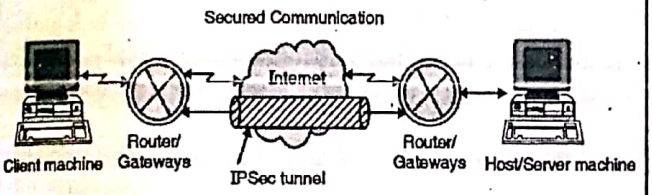

Fig. 4.2.6 : IPSec tunnel mode data exchange

- Tunnel mode is used on most of the IPSec gateway devices such as firewalls, routers, and connecting remote locations such as branch offices, organizations, and universities securely through a network called Virtual Private Network (VPN).
- The entire original, inner, packet travels through a tunnel from one point of an IP network to another;
- Tunnel mode is generally used for secure communication between two routers, a host and a router or vice versa.

Syllabus Topic : IPv6 and IPv4

4.3 IPv6 and IPv4

➔ (SPPU - May 15)

Q. 4.3.1 Discuss IP sec protocols in detail. (Ref. Sec. 4.3) May 15. 6 Marks

Before discussing IPv6 first we need to understand basics of IPv4.

4.3.1 IPv4

IPv4 is a Network Layer Protocol. IPv4 Stands for Internet Protocol Version 4. An IPv4 address is a 32-bit :

- Address that uniquely identifies host or a computer to the Internet.
- Length of IPv4 address is a 32-bit IPv4 Address are unique and all nodes connecting Internet must have IPv4. The address space of IPv4 is 2^{32} or 4,294,967,296.
- Dotted decimal notation of IPv4 is representing as 192.168.0.1 (each single byte separated. (dot) symbol)
- Packets in the network layer is called as datagram's, which consist of two parts : header and data. The header is 20 to 60 bytes in length and contains necessary information about delivery of the packet from one router to another as shown in Fig. 4.3.1.

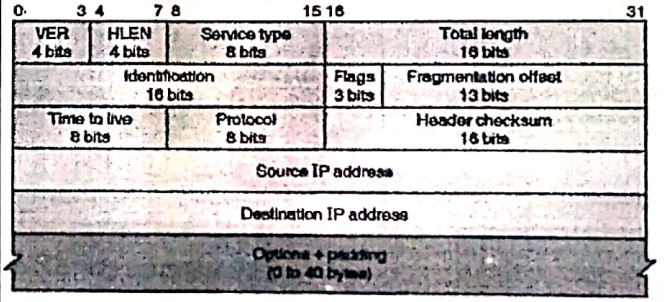

Fig. 4.3.1 : IPv4 Packet Format

- Detail explanation about of each field is given below.

Version (4 bits)

Indicates the version of IP and is set to 4.

Internet Header Length (4 bits)

- Indicates the number of 4-byte blocks in the IPv4 header (Length of entire IP header).
- Because an IPv4 header is a minimum of 20 bytes in size, the smallest value of the Internet Header Length (IHL) field is 5.

Type of Service (8 bits)

Indicates the desired service expected by this packet for delivery through routers across the IPv4 internetwork.

Total Length (16 bits)

Indicates the total length of the IPv4 packet (IPv4 header + IPv4 payload) that is up to 65,535 bytes long

Identification (16 bits)

- Identifies this specific IPv4 packet.
- The Identification field is selected by the originating source of the IPv4 packet. If the IPv4 packet is fragmented, all of the fragments retain the
- Identification field value so that the destination node can group the fragments for reassembly.

Flags (3 bits)

- Identifies flags for the fragmentation process.
- There are two flags one to indicate whether the IPv4 packet might be fragmented and another to indicate whether more fragments follow the current fragment.

Fragment Offset (13 bits)

Indicates the position of the fragment relative to the original IPv4 payload.

Time to Live (8 bits)

- Indicate the maximum number of links on which an IPv4 packet can travel before being discarded.
- Therefore, the TTL becomes a maximum link count with the value set by the sending node.

Protocol (8 bits)

- Identifies the upper layer protocol.
- For example, TCP uses a Protocol of 6, UDP uses a Protocol of 17, and ICMP uses a Protocol of 1.

Header Checksum (16 Bits)

- Provides a checksum on the IPv4 header only.
- Each IPv4 node that receives IPv4 packets verifies the IPv4 header checksum and silently discards the IPv4 packet if checksum verification fails. Therefore, the Header Checksum is recomputed at each hop between source and destination.

Source Address (32 bits)

Stores the IPv4 address of the originating host.

Destination Address (32 bits)

Stores the IPv4 address of the destination host.

Options (multiple of 32 bits)

Stores one or more IPv4 options.

4.3.2 IPv6

IPv6 stands for Internet Protocol version 6. As discussed earlier the Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses. IPv6 is a new design which has 128-bit address that give much greater flexibility in address allocation.

☞ **Why IPv6 ?**

- Larger address space
- Better header format
- New options
- Allowance for extension
- Support for resource allocation
- Support for more security
- Support for mobility

☞ **IPv6 address format (128 bits)**

2031:0000:130F:0000:0000:09C0:876A:130B

- 8 groups of 4 hexadecimal digits
- Each group represents 16 bits
- Separator is ":"
- Case-independent
- Leading zeros in a field are optional:
  2031:0:130F:0:0:9C0:876A:130B

☞ **IPv6 Header Format**

Fig. 4.3.2 shows format of IPv6.

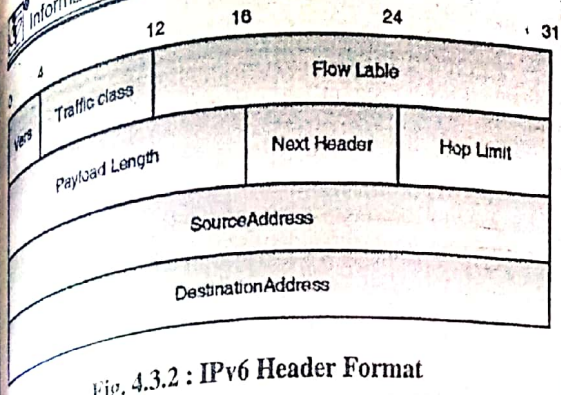

Fig. 4.3.2 : IPv6 Header Format

Version (4 bits)

4 bits are used to indicate the version of IP and is set to 6.

Traffic Class (8 bits)

The 8-bit field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets.

Flow Label (20 bits)

- Identifies a flow and it is intended to enable the router to identify packets that should be treated in a similar way without the need for deep lookups within those packets.
- This label is used to maintain the sequential flow of the packets belonging to a communication.
- Set by the source and should not be changed by routers along the path to destination.
- The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets.
- Unique and powerful tool to IPv6.

Payload Length (16 bits)

With the header length fixed at 40 bytes, it is enough to indicate the length of the payload to determine the length of the entire packet.

Next Header (8 bits)

- Indicates either the first extension header (if present) or the protocol in the upper layer (such as TCP, UDP, or ICMPv6).
- When indicating an upper layer protocol above the Internet layer, the same values used in the IPv4 Protocol field are used here.

Hop Limit (8 bits)

In IPv6, the IPv4 TTL was appropriately renamed Hop Limit because it is a variable that is decremented at each hop, and it does not have a temporal dimension.

Source IPv6 Address (128 bits)

Stores the IPv6 address of the originating host.

Destination IPv6 Address (128 bits)

Stores the IPv6 address of the current destination host.

Syllabus Topic : IPSec Protocols and Operations, AH Protocol, ESP Protocol

4.4 IP Security Protocols

→ (SPPU - May 16, Dec. 16)

Q. 4.4.1 How AH and ESP are differs while working under transport and tunnel mode ? (Ref. Sec. 4.4)

Q. 4.4.2 Describe IPSec protocol with its components and security services. (Ref. Sec. 4.4) May 16, Dec. 16, 8 Marks

Encryption of data and its authenticity is prime concern for secure communication, to avail this two features, IPSec provides two protocols at network layer :

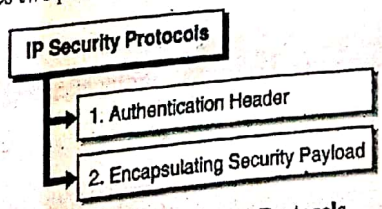

Fig. 4.4.1: IP Security Protocols

4.4.1 Authentication Header

Q. 4.4.3 Explain Authentication Header. (Ref. Sec. 4.4.1)

- It is designed for authentication, integrity of payload which is carried in IP Packet. It is first protocol of IPSec called Authentication Header (AH) protocol designed to provide data authentication (to identify source host), data integrity (if data get modified while in transit) and non-repudiation but doesn't provide data confidentiality (if attacker able to access the contents of a message) because Authentication Header does not encrypt the data/ IP packet.
- The main functionality of this protocol is protection against replay attacks (sending same data to receiver again and again) and protection against tampering of data over a network.
- Authentication Header is also used to protect the upper-layer or the entire IP packet, with the help of message authentication code (MAC - used to generate fixed length value from message and secret key to provide authentication) using well known hashing algorithms like MD5 or SHA1.
- By using Hash function and symmetric key algorithm, message digest is calculated and inserted in authentication data as shown in Fig.4.4.2 because of this AH protocol provides data authentication and data integrity, but not confidentiality or privacy.
- The internal fields of authentication header format are shown in Fig. 4.4.2.
- This protocol uses cryptographic checksum which is similar to hash function or message digest, the checksum is inserted in authentication header and placed in location depends on which mode it is using (tunnel mode or transport mode).

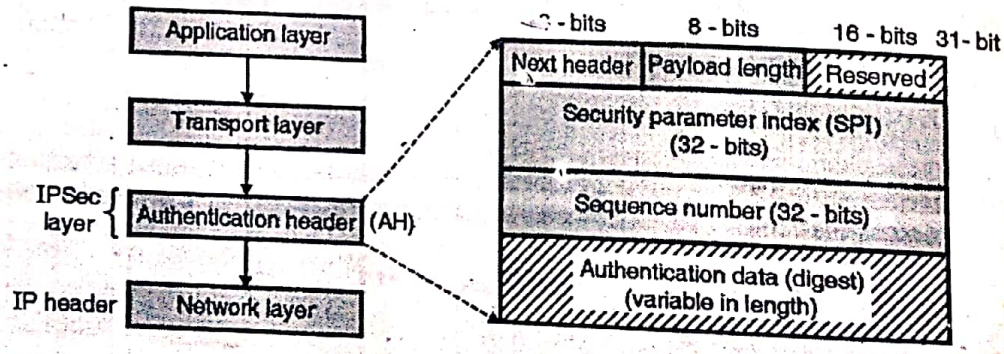

Fig. 4.4.2 : Authentication Header

☞ **A brief description of each field**

- **Next header (8 bits) :** The next header is an 8 - bit filed which is used to identify the type of payload/ data carried by IP packet.
- Identifies the type of header immediately following this header.
- **Payload length (8 bits) :** The payload header is also an 8 - bit filed which defines length of the authentication header.
- Length of the AH in 32-bit words minus 2.
- **Reserved (16 bits) :** AH contains 16 - bit field which is reserved for future use and always set to zero.
- **Security Parameter Index (SPI) (32 bits) :** SPI is a 32-bit field used in combination with source IP address, destination IP address and AH security protocol to uniquely identify a security association (SA) for the traffic to which IP packet belongs, we will discuss SA in next bit. This field also defining which different security algorithms and keys were used to calculate the message authentication code (MAC).

- **Sequence number (32 bits) :** It is also a 32 bit field. It prevents the retransmission of datagram which is also known as **Replay attack.**
- A monotonically increasing counter value.
- **Authentication Data :** This is variable length field whose length depends upon encryption algorithm used. Authentication data field of AH protocol is the output of hashing algorithm or message digest algorithm. AH protocol performs the Integrity Check Value (ICV) on packet header or MAC is computed over the complete IP packet including the outer IP header to ensure that the data has not been changed during transmission process. As mentioned earlier AH doesn't encrypt the data the reason it doesn't provide confidentiality during transmission.

Modes of Operation

AH can work in two modes :

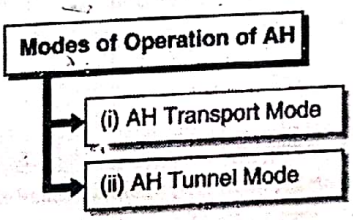

Fig. 4.4.3 : Modes of Operation of AH

(i) AH Transport Mode

In Transport mode the authentication header is placed between original IP Header and original TCP header as shown in Fig. 4.4.4.

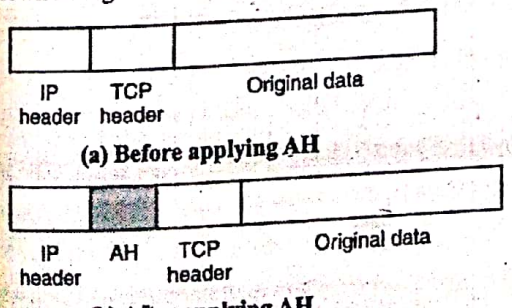

Fig. 4.4.4 : AH transport mode

(ii) AH Tunnel Mode

- In Tunnel Mode the AH is inserted between the new IP header and original IP header.
- The inner IP address contain source and destination address of sender and receiver and the out IP address contain the address of security gateway or firewall as shown in Fig. 4.4.5.

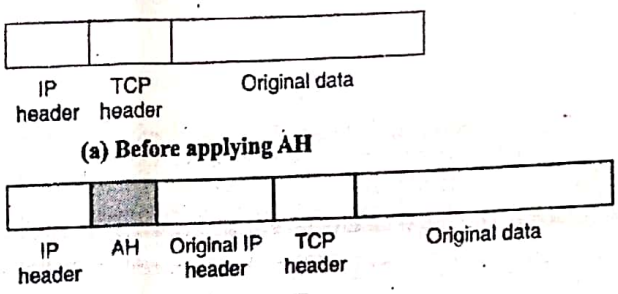

Fig. 4.4.5 : AH tunnel mode

4.4.2 Encapsulating Security Payload

Q. 4.4.5 Explain in detail IPSec ESP format. (Ref. Sec. 4.4.2)

- One of the most important feature that Authentication Header was unable to provide called data confidentiality (if attacker able to access the contents of a message).
- An Encapsulating Security Payload is primarily designed to provide encryption, authentication and confidentiality for the data or payload that is being transferred in an IP network
- As defined earlier ESP is used to encrypt the entire payload of an IPSec packet the reason ESP alone can provide data authentication, protection against replay attacks and data integrity by adding ESP header, ESP trailer and MAC to the packet.
- ESP has the same fields as defined in AH, but it integrates these fields in a different way instead of having just a header, it divides these fields into three components: An ESP header, ESP trailer and ESP authentication block as shown in Fig. 4.4.6.
- It is designed for confidentiality and integrity of messages. ESP can be used alone or with combination

of AH.ESP adds a header and a trailer to the payload. Following are the steps for adding ESP header and trailer.

Step 1 : In the initial step, ESP trailer is added to IP payload.

Step 2 : Payload and trailer or encrypted

Step 3 : After the encryption ESP header is added to the encrypted packet.

Step 4 : ESP header, payload and ESP trailer are used to create authenticated data.

Step 5 : This authentication data is added at the End of Trailer.

Step 6 : Lastly the IP header is added.

- The main functionality of ESP is to provide the confidentiality to IP packet by encrypting them. Encryption algorithms (Triple DES, Blowfish, and IDEA etc.)used to combines the data in the packet with a key and transform it into an encrypted form. The encrypted packet now then transmitted to the destination, and decrypts it using the same algorithm.

The detail description of Encapsulating Security Payload (ESP) fields is given below :

- **ESP Header :** This contains two fields, Security Parameter Index (SPI) of 32 bits and Sequence Number of 32 bits, as defined in AH protocol SPI is a 32-bit field used in combination with source IP address, destination IP address and ESP security protocol to identify a security association (SA) for the traffic to which IP packet belongs.
- **Sequence number :** It is also a 32 bit field. It prevents the retransmission of data gram which is also known as **Replay attack as defined earlier.** This field is not encrypted but it's authenticated to perform anti-replay checking before decryption.
- **Encrypted data :** This is variable length filed contains transport layer segment or IP packet which is protected by performing ESP encryption.
- **ESP Trailer :** ESP trailer field contains padding (0-255 bytes), pad length 8- bits and next header 8 - bits.
- **Padding (0-255 bytes) :** Padding filed used to expand plain text message to required size or to align the encrypted data by adding padding bits to the actual data which provides confidentiality to traffic flow.
- If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- **Pad Length (8 bits) :** This is mandatory field in ESP protocol which used to indicate the number of pad (protection) bytes added into the packet.
- Indicates the number of pad bytes immediately preceding this field.
- **Next Header (8 bits) :** The same bit length as of pad length used to identifies the type of encrypted data in the Payload Data field.
- Identifies the type of data contained in the Payload Data field (an upper-layer protocol - TCP, UDP, or an IPv6 extension header).
- **ESP Authentication Data :** This is variable length field whose length depends upon encryption algorithm used.
- A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.
- As mentioned earlier ESP encrypts the data the reason it provide data confidentiality during transmission.

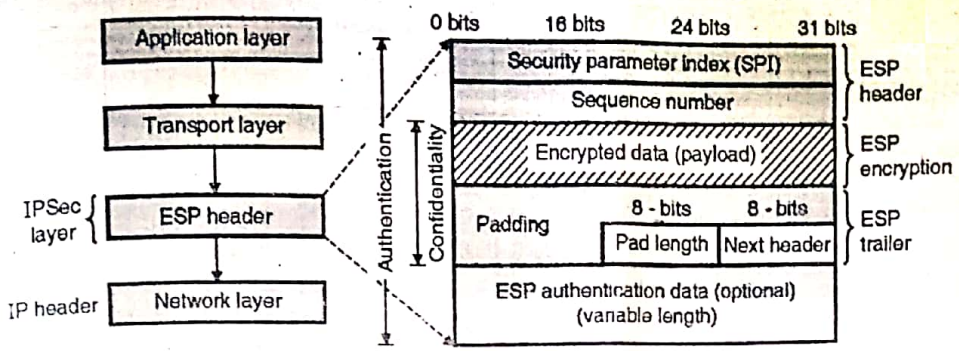

Fig. 4.4.6 : ESP header, trailer and encryption

Modes of Operation

ESP can work in both modes namely :

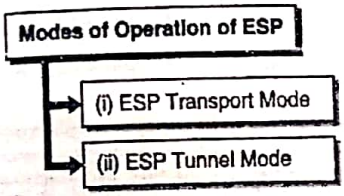

Fig. 4.4.7 : Modes of Operation

(i) ESP Transport mode

In this case ESP header is added before the transport layer header (like TCP,UDP) and trailer is added after the IP Packet whereas if authentication is required then authentication data is added after the ESP trailer. Fig. 4.4.8 shows transport mode in ESP.

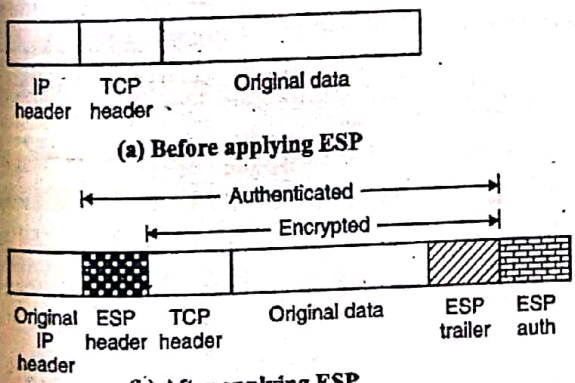

Fig. 4.4.8 : Transport mode in ESP

(ii) ESP tunnel mode

In this case ESP header is added before original IP header and ESP trailer after the original data.

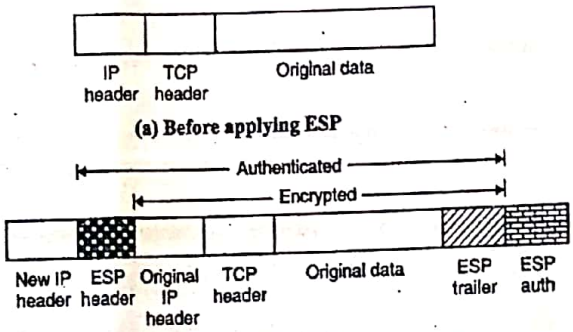

Fig. 4.4.9 : Tunnel mode in ESP

The whole packet is encrypted. As whole packet is encrypted so New IP header will be added, which will contain information for routing from on network to another, so that packet can be transmitted.

Fig. 4.4.9 shows tunnel mode in ESP.

4.4.3 Security Association

- It is an important aspect of IPSec, Security Association (SA) is a contract between the communication parties about factors like IPSec protocol version, mode of operation (tunnel or transport), cryptographic algorithm, key etc. Security Association creates a secure channel between two communicating parties.
- If both AH and ESP are used SA for actual operation then they will need two sets of SA one for AH and one for ESP.
- For communication each party needs two set of SA one for incoming transmission and one for outgoing transmission because SA is simplex unidirectional.

Security Association Database

- Security Association can be very complex.
- Each participating parties need to have inbound and outbound SAs to allow bidirectional communication. It is a two directional table.
- Each row in table defines Security Association which is collectively called as Security Association Database. Each requires party requires maintaining its own database.
- For one way communication (called unidirectional) single SA is required whereas for two way communication (bidirectional) two security association are required. SA uses different parameters to perform data handling between sender and receiver like security parameter index (SPI). IP address of the host (usually destination IP address of end user); encryption algorithms; protocol format (AH or ESP); and security protocol identifier (SPI).
- Almost all these parameters we have discussed in pervious bit still let us have small look out on these parameters.
  - **Security Parameter Index (SPI) :** A 32-bit number used to uniquely identify a particular security association between any connected devices. The SPI is placed in AH or ESP packet for linking the each secure packet to the security association.
  - **Destination IP Address :** Destination IP address of a host, router or firewall who involved in communication or the address of devices for which security associations are established.
  - **Security Protocol Identifier (SPI) :** To identify which protocol (AH or ESP) is used for security associations. If both are used then they have separate security associations.

Transport vs. Tunnel Mode

Protocols	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet plus selected portions of outer IP header.
ESP	Encrypts IP payload and any iPv6 extension header	Encrypts inner IP packet
ESP with authentication	Encrypts IP payload and any IPv6 extension header. Authenticates IP payload but no IP header	Encrypts inner IP packet. Authenticates inner IP packet.

Syllabus Topic : ISAKMP Protocol, Oakley Determination Protocol

4.5 ISAKMP Protocol, Oakley Determination Protocol

Q. 4.5.1 Explain Internet key exchange protocol. (Ref. Sec. 4.5)

- ISAKMP stands for Internet Security Association and Key Management Protocol.
- It consists of the security concepts like key management, authorization, and authentication. It also combines the different higher level associations those are established in order to provide the security for government, private organizations, and commercials on the network.
- The Internet Security Association and Key Management Protocol is very essential in order to define procedures and fixing the structure of packets we can say packet formatting which is used to build, negotiate, change/update and delete security associations.
- It also decides the actual properties in terms of payloads for key generation exchange and to do authentication of data.

The framework involved in this for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is totally different from Key exchange protocol. Hence it works as common framework and it can be implemented over any transport protocol.

4.5.1 ISAKMP

→ (SPPU - Dec. 16, May 17)

Q. 4.5.2 Explain ISAKMP protocol for IP sec. (Ref. Sec. 4.5.1) Dec. 16, May 17, 6 Marks

The internet security association and key management protocol is a framework that defines the formats of payload, the mechanics of implementation of a key exchange protocol, and the exchange of a security association between the parties.

ISAKMP protocol defines the mechanics of implementing a key exchange protocol, and agreement between communicating parties i.e. which are the different features of IPSec protocol has to use etc. and all (simply its negotiation of security association).

ISAKMP features

- It is used to authenticate of remote entity.
- It manages the secure session between communicating parties by applying different cryptographic techniques.
- Exchanging required information about key sharing.
- Negotiation over all data transmission by applying security policies.
- The reasons ISAKMP establish secure communicating channel between two parties and authenticate them for secure key exchange and negotiation on certain security terms and condition.

ISAKMP header

1. **Initiator cookie (64 bit) :** The cookie of the entity that initiated SA establishment, SA notification, or SA deletion.
2. **Responder cookie (64 bit) :** The cookie of the entity that is responding to an SA establishment request, SA notification, or SA deletion.
3. **Next payload (8 bits) :** Indicates the type of the first payload in the message.
4. **Major version (4 bits) :** The major version of the ISAKMP protocol in use.
5. **Minor version (4 bits) :** The minor version of the ISAKMP protocol in use.
6. **Exchange type (8 bits) :** Indicates the type of exchange being used. This dictates the message and payload orderings in the ISAKMP exchanges.
7. **Flags (8 bits) :** Indicates the options that are set for the ISAKMP exchange.
8. **Message ID (32 bit) :** 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.
9. **Length (32 bit) :** The total length of the ISAKMP header and the encapsulated payloads in bytes.

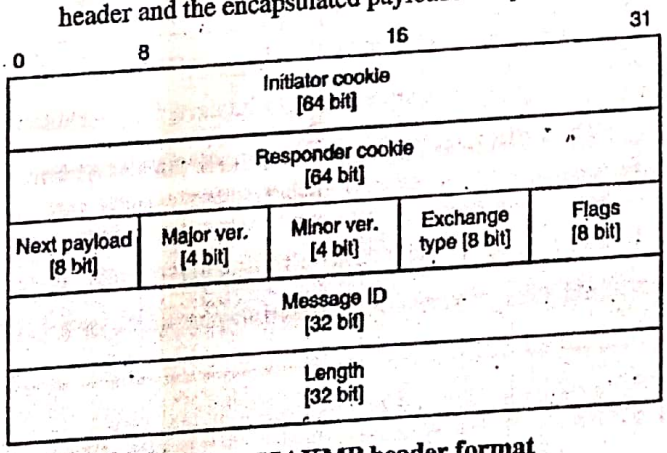

Fig. 4.5.1 : ISAKMP header format

4.5.2 Oakley

→ (SPPU - May 16)

Q. 4.5.3 Explain OAKLEY key determination protocol. (Ref. Sec. 4.5.2) May 16, 6 Marks

- Oakley protocol defines the mechanism of key exchange or key agreement protocols in which two parties must agree on key generated before data transmission.

- **IKE** uses different cryptographic techniques and security policies for securely exchanging information between two entities such as Diffie-Hellman key exchange, DES, MD5, SHA, RSA algorithm etc.

☞ **OAKLEY key determination protocol**

- It is a key Determination Protocol proposed by Hilarie K. Orman in 1998, which is a firm base for broadly used Internet Key Exchange protocol.
- Diffie-Hellman key exchange algorithm is used in between authenticated parties', so on insecure connection also both the parties can exchange the keying material over the network. Hence it is also known as Key agreement protocol.
- This protocol was proposed to increase the cryptographic strength where it allows two organizations to agree on a shared value without requiring encryption. The shared value is immediately available for use in encrypting subsequent conversation, e.g. data transmission and/or authentication.
- OAKLEY is a generic key exchange protocol, because the keys that it generates might be used for encrypting data with a long privacy lifetime, 20 years or more.
- For distribution of keys there are some options laid down. Along with Diffie-Hellman key exchange, OAKLEY protocol can be used to generate a new key from an existing key and before distributing the newly (externally) derived key encryption is performed.
- As per the security requirements and performance requirement of two parties protocol allows using some of the forward secrecy features and anti-clogging features. It also gives an authority to use encryption and non-encryption algorithms.

☞ **Advantages of OAKLEY key determination protocol**

(1) It uses the mechanism of cookie exchange to avoid clogging attacks (DoS type of attack).

(2) It uses nonce (arbitrary number) to detect the replay attacks.

(3) It authenticates the Diffie-Hellman using digital signature, public key encryption or symmetric key encryption to overcome man in the middle attack.

☞ **SKEME**

It is another protocol for exchanging authenticated key between the parties. It uses public key encryption for authentication in key exchange protocol.

4.5.3 IPSec and IKE (Internet Key Exchange) Relationship

- To protect network from traffic, the SAs are needed to be established in IPSec. If there is no SAs present, the IPSec to protect network from traffic, the SAs are needed to be established in IPSec.
- If there is no SAs present, the IPSec which security parameter will be used for IKE negotiation and protection.
- In the protected session, IPsec SAs are negotiated and established. With a Protection of traffic (IPsec SAs) policy is established and keys are exchanged using the Diffie-Hellman method hence IPsec can start to protect the network traffic. When IPsec SAs' lifetime expires, IKE is invoked again, and new IPsec SAs are created and established.

☞ **IKE protocol**

When negotiations of IKE begin, it looks for the IKE policy that is same for both the parties. A match is made when both the parties contains same policies for encryption, authentication, hashing and Diffie-Hellman parameter values. If it does not match then IKE refuses negotiation and IPSec SA's will not be established and negotiated for the parties.

4.5.4 IKE Phases and Modes

Q. 4.5.4 Explain different phases of IKE protocol. (Ref. Sec. 4.5.4)

IKE has two phases of operations :

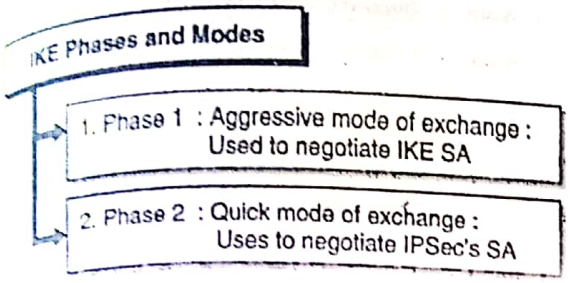

Fig. 4.5.2 : IKE Phases and Modes

1) **IKE phase 1 : Aggressive mode of exchange : Uses to negotiate IKE SA**

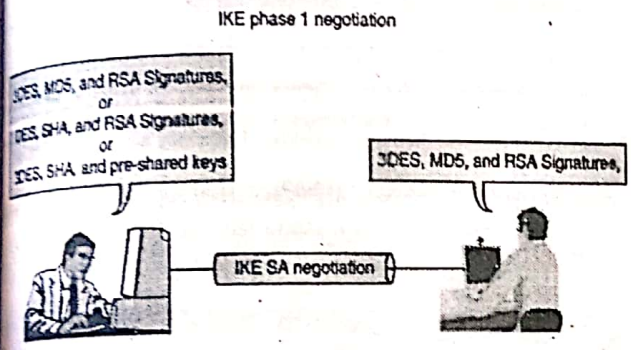

Fig. 4.5.3

For Instance

- As shown in Fig. 4.5.3 user A and user B want to talk IKE, They must agree on a common IKE protection suite.
- The initiator (user A) proposes several protection suites and the responder (user B) chooses one of the offered protection suite.
- The selection is made according to the priorities and the configuration of the responder.
- In the Fig. 4.5.4, user A proposes three protection suites out of which user B chooses the second protection suite. Both must agree on the same protection suite.
- If they do not, no common policies may exist and the IKE session may be terminated.

2) **IKE Phase 2 : Quick mode of exchange : Uses to negotiate IPSec's SA**

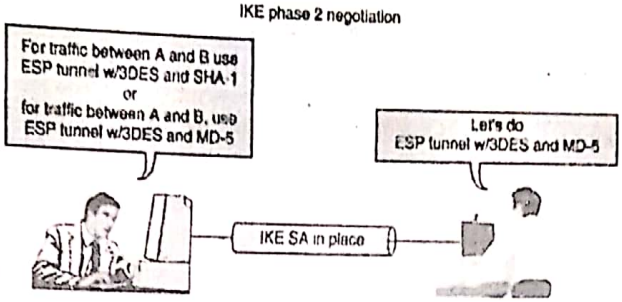

Fig. 4.5.4

For Instance

- As shown in Fig. 4.5.4 user A and user B wish to protect the traffic with IPSec and the IKE SA is already established between them.
- User A proposes various IPSec security policies and user B chooses one of them (with highest priority according to configuration).
- After successful negotiation, keying material is exchanged and IPSec SAs are established to protect network traffic

4.5.5 IP Security Benefits / Applications

(SPPU - Dec. 14)

Q. 4.5.5 Enlist the applications and benefits of IPSec. (Ref. Sec. 4.5.5)

Q. 4.5.6 What are the benefits of IPSec? (Ref. Sec. 4.5.5) **Dec. 14, 4 Marks**

- As mentioned earlier IP Sec operates at the network layer where secure data transmission take place. For secure access of remote computer over Internet IPSec is used.
- For securely connecting all branches of bank sectors over internet IPSec protocol is used. For secure communication between same organization which are located at different places.
- For connecting to college server any time from any location IPSec protocol is used.
- Most of the corporate sector allowing employees to performed their task from home and update it to

company server at any time from any location or secure access of company server at any time.

- IPSec now-a-days called as one of the standard of Virtual Private Networks that allow low cost connectivity, secure data transmission between various locations over insecure communication channel.
- If IPSec is implemented in a *firewall or router*, can provide strong security to the ongoing traffic crossing the network.

Syllabus Topic : VPN

4.5.6 VPN

→ (SPPU - Dec. 16)

Q. 4.5 What is VPN? Explain types of VPN. (Ref. sec. 4.5.6) Dec. 16, 6 Marks

- **VPN stands for Virtual Private Network.** The network technology VPN extends the private network (LAN) over a public network (Internet).
- The computer (or network) can be connected securely using VPN if they are physically connected. The companies use VPN, which allows remote workers to connect securely to their private network over public network.
- Also it is used to interconnect remote offices with a head office. VPN is creating secure tunnel between two more devices. It also helps to protect the web traffic from snooping and interference.

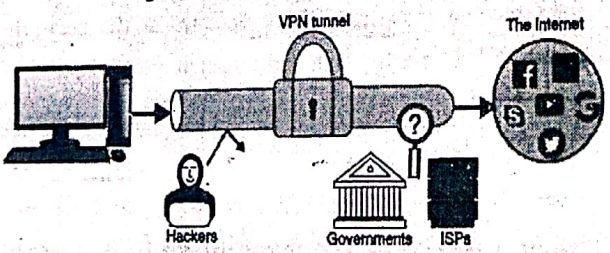

Fig. 4.5.5 : Virtual Private Network

☞ **Advantages**

1. Making client to server connection may not be practical for individual users and also creating WAN connectivity is very costly. VPN is created a secure connection between two endpoints and the information exchanged between the two VPN endpoints is encrypted. So when information is transmitted over the internet no eavesdropping occurs.
2. A VPN can also be used to hide your privacy by disguising true IP address of the user's computer. The company owners protect their identity by using VPN to change their IP address.

☞ **Disadvantages**

1. To encrypt/decrypt and additional data transmission additional processing power required which has have negligible impact on overall usage of the network.
2. The VPN device from one vendor may not work well from a device from another vendor, all VPN devices is not interoperating always. Verify compatibility between the two endpoints by network engineers implementing VPN technology. If VPN is not setup properly then client server connection gets slowdown.

4.5.6(A) VPN Protocols

To create a virtual private network, a virtual tunnel is established between two endpoints via a virtual tunneling protocol or by data encryption. Most popular VPN protocols include IPsec, SSL/TLS, PPTP and L2TP.

1. **PPTP** - The oldest Point-to-point protocol developed by a consortium found by Microsoft, which is supported by vast majority of operating systems. The encryption based on 128-bit key has been cracked, and it is no longer considered very secure
2. **L2TP/IPsec** - VPN based on Layer 2 Tunnel Protocol with IPsec encryption provides more secure service with more features than PPTP.
3. **Open VPN** - Open source technology Open VPN developed on OpenSSL provides secure connection and strong encryption. It has become the default VPN connection type, and is widely supported by 3rd-party software including iOS and Android.

Syllabus Topic : Web Security - Introduction

4.6 Web Security Considerations

➔ (SPPU - Dec. 13)

Q. 4.6.1 Draw SSL Protocol Stack and explain same. (Ref. Sec. 4.6.1) **Dec. 13, 8 Marks**

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets. There are multiple ways or Approaches are used to provides web security.

Most of the ways are similar or provides similar services for the web security but they can be differentiating with the capability and their scope within the TCP/IP stack.

Location of Security facilities in TCP/IP Protocol Stack is shown in Fig. 4.6.1.

- IP security (IPSec) is the one of the important way to provide the web security.
- Advantage of IPSec is that it is transparent to application as well as end user.
- IPSec is used for filtering traffic and it is a general solution for web security.
- Another solution is to implement security just above TCP called as Secure Socket Layer & Transport Layer Security.
- SSL can be embedded in specific packages. For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol.

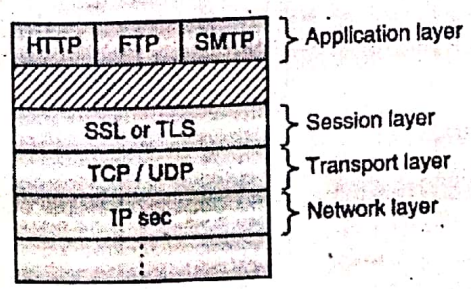

Fig. 4.6.1 : Location of Security facilities in TCP/IP Protocol Stack

☞ **Possible attacks on web security are as follows**

1. **Deceptive phishing**

   Sending bulk of email messages, which make user to click any one of the bulk email such type of attack called as deceptive phishing.

2. **Malware based phishing**

   Running malicious software on target's or users machine. There malware comes from the email attachments.

3. **Key loggers and screen loggers**

   These malware track input from keyboard and send information of target through target's keyboard to hacker (attacker) via internet.

4. **Session hijacking**

   User activities are monitored to get login into the user's system.

5. **Web trojans**

   They are a kind of pop-ups, when logging into some website. These pop ups usually ask for user's credentials.

6. **System reconfiguration attacks**

   It is kind of phishing attack where user's PC setting are modified or changed.

7. **DNS based phishing**

   In this type of phishing the URL requested return to some bogus or fake site which is actually sent by hacker by changing the URL of the requested site of the user.

8. **Content injection phishing**

   It is an act of inserting some malicious content in the websites which can redirect to some other website or may install malware.

9. **Bandwidth attack**

   - Every website is given particular amount of bandwidth to host (e.g. 50 GB) loading of any

websites takes certain amount of time to display whole webpage.

- If more visitors load particular websites page or consumes whole 50 GB bandwidth than particular websites can be ban.
- The attacker does the same by opening 100 pages of site and keeps on loading and refreshing, consuming all bandwidths to make the site out of services.

10. Logic attack

Attack on the network software to make it vulnerable.

For example : in TCP/IP stack.

11. Protocol attacks

This attack, consumes more amount of resources in victims system. It is an attack on the particular features of some protocol that are been installed in the victims systems.

12. Unintentional Dos attack

Sometimes because of huge popularity among users the particular wets suddenly end up.

Solution to achieve web security

- Websites are always to prone to security risks. Website is the main target for installing malicious software or malware on computer. Hackers may also steal important data such as credit card information, destroy the business and can propagate illegal content on user's system.
- **Updated software :** Software updating is mandatory for security consideration.
- **SQL injection :** In SQL injection change in database is done by alerting table data in database.
- **Cross Site Scripting (XSS) :**It allows the attackers to inject client side script into web pages. Therefore, while creating a form it is good to ensure that check the data being submitted and encode or strip out any HTML.
- **Error messages :** While sending error message, it should be prescribed how much information should be send in error message, for example if user is failed while doing logging then user should not come to know which part of login has error username or password.
- **Validation of data :** Both client side and server side data should be valid.
- **Passwords :** It is good to enforce password requirements such as of minimum of eight characters, including upper case, lower case and special character. It will help to protect user's information in long run.
- **Upload files :** The file uploaded by the user may contain a script that when executed on the server opens up your website.
- **SSL :** It is good practice to use SSL protocol while passing personal information between website and web server or database.

Syllabus Topic : SSL, SSL Session and Connection

4.6.1 Secure Socket Layer (SSL)

→ (SPPU - Dec. 14, May 17)

Q. 4.6.2 Explain SSL Protocol interaction sequence diagram between client and server. (Ref. Sec. 4.6.1)

Q. 4.6.3 Discuss SSL with respect to 4 phases.
(a) Establish Security capabilities.
(b) Server authentication and key exchange.
(c) Client authentication and key exchange.
(d) Finish.
(Ref. Sec. 4.6.1)

Q. 4.6.4 Explain architecture of Secure Socket Layer (SSL). (Ref. Sec. 4.6.1) Dec. 14. 8 Marks

Q. 4.6.5 Explain the operation of Secure Socket Layer (SSL) protocol in detail. (Ref. Sec. 4.6.1) May 17. 8 Marks

- Secure Socket layer invented by Netscape communications in 1994. Secure Socket layer is an internet protocol used for securely exchanging the information between client's web browser and the web server.
- Secure socket layer ensure the authentication, data integrity and data confidentiality between web browser and web server i.e. it creates a secure tunnel between

client and server. The main role of SSL is to provide the security to web traffic in all the way.

The current version of SSL is 3.0. The position of SSL in TCP/ IP protocol suite is shown in Fig. 4.6.2.

SSL is works in between application layer and transport layer the reason SSL is also called as **Transport Layer Security (TLS).**

Transport Layer Security (TLS) protocol is used to ensure security between communicating applications and their users on the Internet.

Main function of transport layer protocol is to protect attacker when a server and client communicate, it ensures that attacker or third party should not modify or tamper with any message.

TLS is the successor to the Secure Sockets Layer (SSL). Will discuss TLS in section 4.6.3.

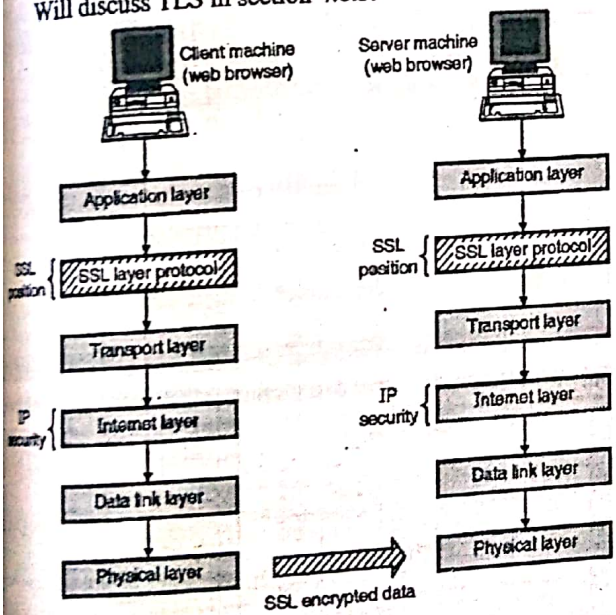

Fig. 4.6.2 : Position of SSL in TCP/ IP protocol suite

- The data will not be passed directly to transport layer instead it will passed to secure socket layer.
- Secure Socket Layer will perform encryption to the data received by application layer and add its own encryption information header called SSH i.e. Secure Socket Layer Header. In the receiver's end SSL will remove the SSH header and then pass data to application layer.
- The Fig. 4.6.2 shows position of SSL protocol in TCP/ IP protocol suite. SSL protocol uses digital certificate and digital signature for securely communication between client machine and server machine.
- SSL encrypt the data received from application layer of client machine and add its own header (SSL Header) into the encrypted data and send encrypted data to the server side.
- Upon receiving encrypted data, server removes the SSL header and decrypts the data and sends the decrypted data to application layer.
- SSL is composed of four protocols in two layers, which support SSL as shown in Fig. 4.6.3. Out of the four, the two most important protocols that are at the heart of SSL are the SSL Handshake Protocol and the SSL Record Protocol, The other two protocols such as SSL Change Cipher Specification and the SSL Alert Protocol play a minor role relatively to previous two protocols.
- The role of these higher-level protocols is the connection establishment, use of required cipher techniques for data encryption and alert (warning, error if any) generation before starting actual data transmission process between client and server.

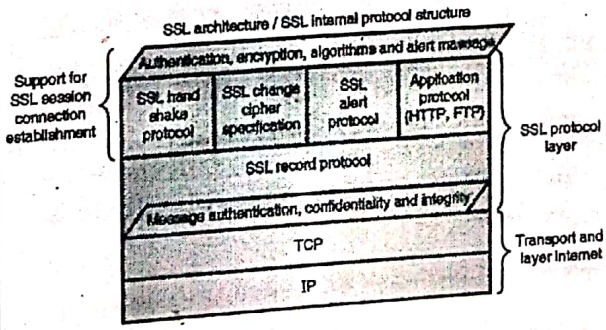

Fig. 4.6.3 : SSL protocols internal architecture

- The SSL Record Protocol is responsible for encrypted data transmission and encapsulation of the data sent by the higher layer protocols (handshake, alert, HTTP) also to provide basic security services to higher layer protocols.

- SSL was designed to make use of TCP protocol to provide a reliable secure process-to-process delivery of entire message/packets.
- We will discuss how client machine securely communicate with the server machine by using underlying network architecture.

4.6.2 Working of SSL

We will discuss SSL Handshake Protocol and the SSL Record Protocol in details.

Syllabus Topic : Handshake Protocol, Change Cipher Spec Protocol

4.6.2(A) Handshake Protocol

➔ **(SPPU - May 15, May 16, Dec. 16)**

Q. 4.6.5 Explain the handshake protocol actions in SSL. (Ref. Sec. 4.6.2(A))

Q. 4.6.6 Explain steps of SSL handshaking protocols. (Ref. Sec. 4.6.2(A)) **May 15, 8 Marks**

Q. 4.6.7 Explain Secure Socket Layer handshake protocol in brief. (Ref. Sec. 4.6.2(A)) **May 16, Dec. 16, 5 Marks**

- As the name suggests when we meet to our friend/colleagues, we have habit to say hi/hello and do the *shake-hands* with each other before starting our actual communication. SSL handshake protocol uses somewhat same ideology but in terms of client and server.
- The first sub-protocol of SSL called *handshake protocol* used for secure communication between client and the server using an SSL enabled connections.
- In this protocol client authentication to the server is more important that server authentication because server has different options available for client authentication.
- The details **steps of SSL handshake protocol** are shown in Fig. 4.6.4.

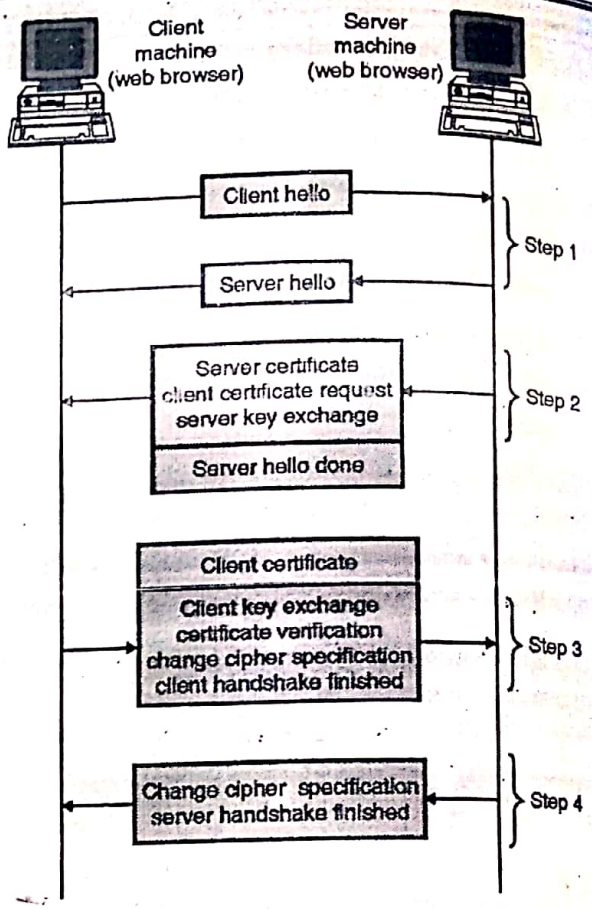

Fig. 4.6.4 : SSL handshake protocol

1. It is used by client and server to start communication using SSL enabled connection.
2. The handshaking is done 4 phases :

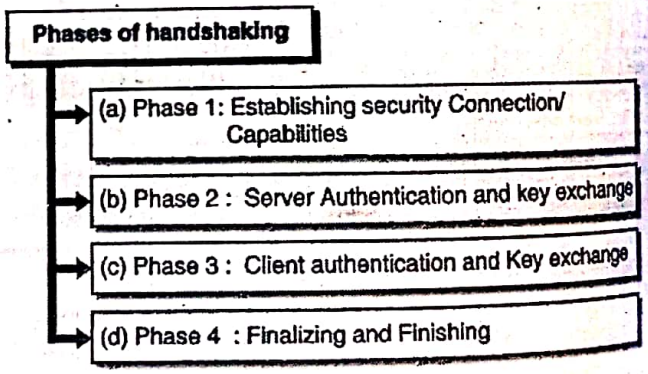

Fig. 4.6.5 : Phases of handshaking

➔ **(a) Phase 1 : Establishing Security Connection/ Capabilities**

In this phase logical connections is established between client and server and establish security capabilities associated with that connections. It consists of two messages, the client hello and the server hello.

Client hello

The client hello message contains the following parameters.

(i) The highest SSL version number which the client can support.

(ii) A 32-bit timestamp and a 28-byte random field that together serve as nonce during key exchange to prevent re- play attacks

(iii) A session id that defines the session (a variable length session identifier).

(iv) There is a cipher suite parameter that contains the entire list of cryptographic algorithms which supports client's system.

(v) A list of compression methods that can be supported by client.

Server

(i) The SSL version number, the highest among both SSL number of client and server, will be supported by client and other will be supported by server.

(ii) A 32 byte random number that will be used for master secret generation, however this random number is totally independent from the random number of client.

(iii) A session id that defines the session.

(iv) A cipher suite contains the list of all cryptographic algorithms that is sent by the client from which the server will select the algorithm

(v) A list of compression methods sent by the client from which the server will select the method.

→ (b) Phase 2 : Server Authentication and Key Exchange

- In this phase, the server authenticates itself if it is needed. The server sends its certificate, its public key, and also request certificate (digital certificate) from the client.
- **Certificate :** The server sends a certificate message to authenticate itself to the client. If the key exchange algorithm is Diffie-Hellman than no need of authentication.
- **Server key Exchange :** This is optional. It is used only if the server doesn't sends its digital certificate to client.
- **Certificate Request :** The server can request for the digital certificate of client. The client's authentication is optional.
- **Server Hello done :** The server message hello done is the last message in phase 2. This indicates to the client that the client can now verify all the certificates received by the server. After this hello message done, the server waits for the client's side response in phase 3.

→ (c) Phase 3 : Client Authentication and Key Exchange

- In this phase, the client authenticates itself if it is needed. The client sends its certificate, client key exchange and certificate verify to the server.
- **Certificate :** Client certificate is optional, it is only required if the server had requested for the client's digital certificate. If client doesn't have client's digital certificate it can send no certificate message or an Alert message to the server. Then it is upto server's decision whether to continue with the session or to abort the session.
- **Client key Exchange :** The client sends a client key exchange, the contents in this message are based on key exchange algorithms between both the parties.
- **Certificate verify :** It is necessary only if the server had asked for client authentication. The client has already sent its certificate to the server. But additionally if server wants then the client has to prove that it is authorized holder of the private key .The server can verify the message with its public key which was already sent to ensure that the certificate belongs to client.

→ (d) Phase 4 : Finish

The client and server send messages to finish the handshaking protocol. It contains 4 steps. The first two messages are from the client i.e. change cipher specs, finished. The server responds back with change cipher spec and finished.

- **Change cipher spec :** It is a client side message telling about the current status of cipher protocols and parameters which has been made active from pending state.
- **Finished :** This message announces the finish of the handshaking protocol from client side.
- **Change Cipher spec :** This message is sent by server to show that it has made all the pending state of cipher protocols and parameters to active state.
- **Finished :** This message announces the finish of the handshaking protocol from server and finally handshaking is totally completed.

Syllabus Topic : Alert Protocol

4.6.2(B) Alert Protocol

- SSL uses the Alert protocol for reporting error that is detected by client or server, the party which detects error sends an alert message to other party. If error is serious then both parties terminate the session.
- Table 4.6.1 shows the types of alert messages. SSL alert protocol is the last protocol of SSL used transmit alerts (warnings, errors, fatal etc.) if any via SSL record protocol to the client or server.
- The SSL alert protocol format is shown in Fig. 4.6.6. Alert protocol uses two bytes to generate alert. First 1 byte indicates two values either 1 or 2. "1" value indicate warning and "2" value indicate a fatal error (if fatal error terminate the session/ connection).
- Whereas second 1 byte indicates predefined error code either the server or client detects any error it sends an *alert* containing the error (error occurred during handshaking, error occurred during data processing at server or client side, certificate defeats, etc.)

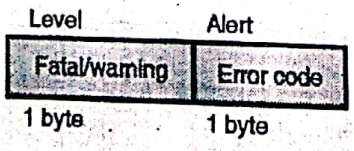

Fig. 4.6.6 : SSL Alert protocol

Table 4.6.1 : Types of alert messages

Alert Code	Alert Message	Description
0	close_notify	No more message from sender
10	unexpected_message	An incorrect message received
20	bad_record_mac	A wrong MAC received
30	decompression_failure	Unable to decompress.
40	handshake_failure	Unable to finalize handshake by the sender.
42	bad_certificate	Received a corrupted certificate.
42	Nocertificate	Client has no certificate to send to server.
42	Certificate expired	Certificate has expired.

Syllabus Topic : Record Protocol

4.6.2(C) Record Protocol

- After completion of successful SSL handshaking the keen role of SSL record protocol starts now.
- SSL record protocol is second sub-protocol of SSL also called lower level protocol.
- As defined earlier the SSL Record Protocol is responsible for encrypted data transmission and encapsulation of the data sent by the higher layer protocols (handshake, alert, HTTP) also to provide basic security services to higher layer protocols.
- SSL record protocol is basics for data transfer and specially used to build a data path between client and server and encrypt the data path before communication.
- SSL record protocol provides different service like data authentication; data confidentiality though encryption

algorithms and data integrity through message authentication (MAC) to SSL enabled connections.

The details steps involved in SSL record protocol and SSL record header format as shown in Fig. 4.6.7.

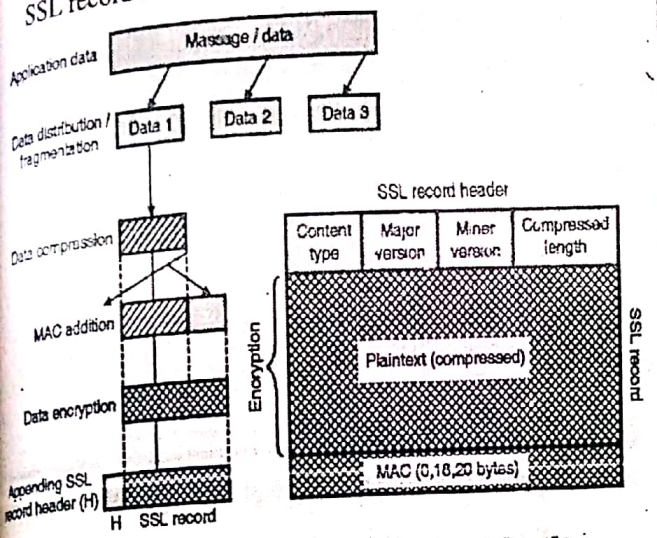

Fig. 4.6.7 : Record protocol and record header

- At this stage all necessary authentication and cryptographic parameters are exchanged between client and server now it's time of secure SSL data transmission through SSL record protocol.
- SSL record protocol takes application data i.e. actual data that client wants to send over server. Divide this data into the different blocks for each length should not exceed 16384 bytes this process is called as *data distribution* or *data fragmentation*.
- Next step is Data compression using lossless compression techniques; compression size of data should not exceed 1024 bytes.
- After the data fragmentation and compression step the MAC ((Message Authentication Code) is computed over the data and MAC is then appended to the compressed data (the data is now encapsulated) to form a new encrypted data / payload.
- The compressed data and MAC again goes through data encryption process. SSL record protocol uses symmetric key cryptographic techniques like DES, triple DES, AES, and IDEA because these techniques are specially designed to operate on block cipher.
- Finally SSL record header is prepended onto each encrypted blocks obtained from encryption process.
- Each output block produced by the SSL Record Protocol is referred to as an SSL record. The length of a record is not to exceed 32, 767 bytes.
- **SSL record header** (Refer Fig 4.6.7) consists of 8-bit content type to which identify nature of the message whether any application data or connection termination or any error message.
- Next field is Major Version which is 8-bit field used to indicate latest version of SSL is in use (e.g., 3). Minor Version which is 8-bit field indicates the lowest version of SSL is in use (e.g., 0).
- Plaintext (compressed) / compressed length which is 16-bit field indicates the length of the plaintext being compressed.
- Finally sends SSL layer encrypted data to TCP and IP (Transport and Internet layer) for necessary transmission over network
- At the receiver end, the encrypted blocks are decrypted and then checked for data authentication, data confidentiality and data integrity, reassemble these data into single unit, and delivered to the application-layer protocol.
- The Record Protocol provides **two services** in SSL connection :
  - **a) Confidentiality :** This can be achieved by using secret key, which is already defined by handshake protocol.
  - **b) Integrity :** The handshake protocol defines a shared secret key that is used to assure the message integrity.

Following are the **operations** performed in Record protocol after connection is established and authentication is done of both client and server.

1. **Fragmentation :** The original message that is to be sent is broken into blocks. The size of each block is less than or equal to 2^{14} (16384) bytes.
2. **Compression :** The fragmented blocks are compressed which is optional. It should be noted

that the compression process must not result into loss of original data.

3. **Addition of MAC :** The Message authentication code (a short piece of information used to authenticate a message for integrity and assurance of message) for each block is to be calculated using shared secret key.

4. **Encryption :** The overall steps including message is encrypted using symmetric key but the encryption should not increase the overall block size.

5. **Prepend Header :** After all the above operations, header is prepended in the encrypted block which contains following fields :

- Content type (8 bits) specifies which protocol is used for processing.
- Major Version (8 bits) specifies the major version of SSL used, for example if SSL version 3.1 is in use than this field contains 3.
- Minor Version (8 bits) specifies the minor version of SSL used, for example if SSL version 3.0 is in use than this field contains 0.
- Compressed length (16 bit) specifies the length in bytes of the original plain text block.

4.6.3 Transport Layer Security (TLS)

Q. 4.6.9 Explain TLS. (Ref. Sec. 4.6.3)

It is an extension of secure socket layer. The main aim of TLS is to provide security and data at the transport layer between two web applications. Almost all web browsers and web servers support TLS. It ensures no eavesdropping and tampering of the message.

- **The TLS protocol consists of two main components :** Handshake protocol, to start session and share private key, and Record protocol, to transmit data securely using the shared keys.
- **Handshake protocol :** In the Handshake protocol, both sending and receiving parties acknowledge their protocol versions, agree on cryptographic and compression algorithms, optionally authenticate each other through certificates, and use public-key encryption techniques to generate shared private keys.

Following are the steps

Step 1 : Clients sends message publicly to containing version of TLS,32-byte random number r_A consisting of a 4-byte timestamp and a 28-byte random number.

A Cipher Suite list in decreasing order of preference for each of the following algorithm families : Public-Key Algorithm (PKA), encryption algorithm used in the Cipher Block Chaining, and compression algorithm (COMPRESS).

Step 2 : Server informs the client about the decided algorithms (after examining the Cipher Suite list sent by the client) along with a 32-byte random number r_B constructed similarly as r_A.

Step 3 : Client replies with a number called pre-master secret s_{pm} using the public key algorithm PKA with public keys retrieved from the server's certificate signed by a Certifying Authority (CA).

Step 4 : Both parties independently calculate the 48-byte long master secret, s_m, to further obtain the keys to exchange data. The master secret is calculated using Pseudorandom Function

$PRF: s_m = PRF(s_{pm}, \text{"master secret"}, r_A || r_B)$

It is worth mentioning that in the previous version of TLS the master secret was computed as follows, before MD5 proven to be insecure :

$MD5(s_{pm} || SHA\text{-}1(A || s_{pm} || r_A || r_B)) \quad || MD5(s_{pm} || SHA\text{-}1(BB || spm || rA || rB)) \quad || MD5(s_{pm} || SHA\text{-}1(CCC || s_{pm} || r_A || r_B))$

Where A, BB, and CCC are strings added for padding.

Step 5 : At this stage, both parties know s_m, s_{pm}, r_A, and r_B, they independently compute the Key Block (KB)

that contains all needed private shared keys for this session : KB = P RF(s_m, "key expansion", $r_A||r_B$) KB is then broken into six pieces and labeled as K1, K2, ..., and K6, before terminating the Handshake phase

Record protocol : Now the client and the server are ready to communicate securely using the key block as a set of security parameters obtained by the Handshake protocol. The Record protocol takes data to be transmitted in one endpoint, fragments the data into manageable blocks, compresses the data, applies a MAC, encrypts by block cipher, and transmits the result. Received data is then decrypted, verified, decompressed, reassembled, and then delivered to higher-level application on the other endpoint. In short, Record protocol ensures that the connection is private via symmetric encryption by session unique keys and reliable via integrity check. Suppose the client wants to send data chunk, d.

The client

1. Compresses the data using the agreed algorithm :
   $d' = COMPRESS(d)$.
2. Hashes the compressed data for data integrity using K_2:
   $d'' \{d', HMAC_{K2}(d')\}$.
3. Encrypts the data along with its MAC using CBC mode block cipher BCA where the secret key is K_1 and the initialization vector is K_3:
   $d''' = BCA_{K1}(d'' K_3)$.
4. Sends d''' over the public channel.

And the server retrieves d from d'''

1. Decrypts the data along with its MAC using BCA_{K1}.
2. Verifies data integrity by computing HMAC of data using K_2 and comparing it with the HMAC computed on the client side.
3. Decompresses to retrieve d.

The process is reversed when server wants to send data to the client while the last three pieces of the key block is used instead.

Syllabus Topic : Electronic Mail Security - Pretty Good Privacy

4.7 Electronic Mail Security : Pretty Good Privacy

→ (SPPU - Dec. 13, Dec. 14, May 15)

Q. 4.7.1	Write short note on PGP. (Ref. Sec. 4.7)	Dec. 13. 6 Marks
Q. 4.7.2	Write short note on Email security (Ref. Sec. 4.7)	Dec. 14. 4 Marks
Q. 4.7.3	What is PGP? Explain operations of PGP. (Ref. Sec. 4.7)	May 15. 8 Marks

- We all are aware that most popular use of Internet is to send the email and chatting with the friend's, partner etc. But have you ever think that if we are sending mail to intended person is going in his inbox only?
- Security concerns have estimated that only about one in every 100 messages is secured against interception and modification attacks. Are we aware that sending an email to business partner or friends in clear text message is going through thousands of machines (between sender and receiver before it reaches to intended recipients?) these machines might read and saved the contents of email for future use?
- Many people think that name given in sender of the mail identifies who actually sends it.
- When you send a message through email, we cannot guarantee that it will be deliver to correct destination or received exactly what you sent. And even there is a no way of knowing that the message is received read and forwarded by attacker.
- Because of wide spared problem of email modifications, sending it to wrong destination by intermediate parties, email spoofing, we need a competing solution to overcome and address the issues of authentication, integrity and reliability of the messages between sender and receiver.

- The public key cryptography play an important role because of two keys used, only intended sender can decrypt the message using his public key as message encrypted using private key of the sender.
- The solution is called as Pretty Good Privacy (PGP) program/ software which provide the secrecy and non-repudiation of data sent over Internet especially by email.
- Pretty Good Privacy (PGP) is a popular open-source freely available software package/ techniques used to encrypt and decrypt email messages over the Internet.
- PGP is an e-mail security program written by Phil Zimmermann in 1991, PGP program become a de facto standard for e-mail security used to store the encrypted files so that it can be non-readable by other users or intruders.
- This program also be used to send an encrypted digital signature, let the receiver verify the sender's identity and know that the message was not changed or modified while transmission.
- Once the file is encrypted using PGP program only the intended recipient can decrypt it. Once message content digitally singed by sender, the sender guarantee to the recipients that message or file comes from valid sender and not by attacker.
- Digital signature functionality of PGP guarantees that the message or file come from the sender and not from an intruder.

4.7.1 Working of Pretty Good Privacy

➔ (SPPU - Dec. 14, Dec. 16, May 17)

Q. 4.7.4 What are the different principal services provided by PGP? Discuss each service in detail. (Ref. Sec. 4.7.1) **Dec. 14. 8 Marks**

Q. 4.7.5 Explain working of PGP in detail. (Ref. Sec. 4.7.1) **Dec. 16. 9 Marks**

Q. 4.7.6 Explain working of PGP algorithm in detail. (Ref. Sec. 4.7.1) **May 17. 9 Marks**

- As mentioned earlier PGP uses the concept of public key cryptography, if user encrypts the plain text message using PGP, it first compress the plain text message.
- BGP uses data compression technique to encrypt the plain text message which saves the transmission time and disk space and more important it strengthen the security.
- After data compression PGP generate the session key. Table 4.7.1 shows how PGP encrypt the message in order to achieve the confidentiality, integrity and non-repudiation.

Table 4.7.1 : Encryption and Decryption of Pretty Good Privacy

Sr. No.	Parameter	Algorithm	Description
1.	Digital signature	SHA or RSA	A hash code of a message is created using SHA-1. This message digest is encrypted using RSA with the sender's private key and added with the message.
2.	Message encryption	IDEA or Triple DES with Diffie-Hellman or RSA	A message is encrypted using IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and Rdded with the message.
3.	Compression	ZIP	A message is compressed, which saves the transmission time and disk space.

Sr. No.	Parameter	Algorithm	Description
4.	Email compatibility	Radix 64 conversion	For email applications transparency, an encrypted message converted to an ASCII string using Radix 64 conversion.
5.	Segmentation		To resemble the segments before decryption process.

Following are the detail encryption and decryption steps of PGP :

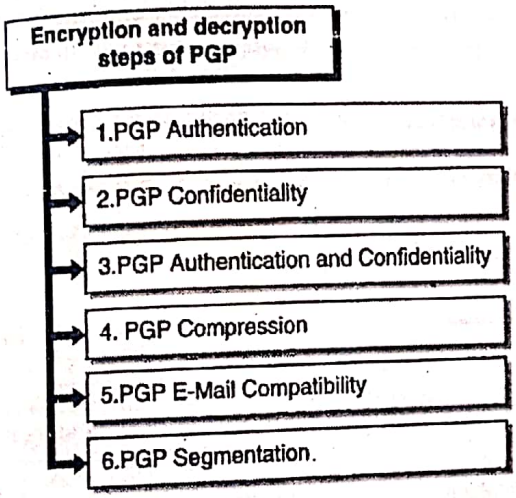

Fig. 4.7.1 : Encryption and decryption steps of PGP

➜ 1. PGP Authentication

1. Ramesh has (private/public) key pair *(Rd/Re)* and he wants to send a digitally signed message *m* to Suresh.
2. Ramesh hashes the message using SHA-1 to obtain *SHA(m)*.
3. Ramesh encrypts the hash using his private key *Rd* to obtain ciphertext *c* given by

$$c = \text{encrypt}_{Rd}(\text{SHA}(m))$$

4. Ramesh sends the pair *(m,c)to Suresh*
5. Suresh receives *(m,c)* and decrypts *c* using Ramesh's public key *Re* to obtain signature *S*

$$S = \text{decrypt}_{Re}(c)$$

6. He computes the hash of *m* using SHA-1 and if this hash value is equal to S then the message is authenticated.

Suresh is sure that the message is correct and that came from Ramesh. Furthermore Ramesh cannot later deny sending the message since only Ramesh has access to his private key *Rd* which works with respective public key *Rd*.

➜ 2. PGP Confidentiality

1. Ramesh wishes to send Suresh a confidential message *m*.
2. Ramesh generates a random session key *k* for a symmetric cryptosystem.
3. Ramesh encrypts *k* using Suresh's public key *Be* to get

$$k' = \text{encrypt}_{Be}(k)$$

4. Ramesh encrypts the message *m* with the session key *k* to get ciphertext*c*

$$c = \text{encrypt}_{k}(m)$$

5. Ramesh sends Suresh the values *(k',c)*
6. Suresh receives the values *(k',c)* and decrypts *k'* using his private key B_d to obtain *k*.

$$k = \text{decrypt}_{Bd}(k')$$

7. Suresh uses the session key *k* to decrypt the ciphertext *c* and recover the message *m*

$$m = \text{decrypt}_{k}(c)$$

Public and symmetric key cryptosystems are combined in this way to provide security for key exchange and then efficiency for encryption. The session key *k* is used only to encrypt message *m* and is not stored for any length of time.

➜ 3. PGP Authentication and Confidentiality

The schemes for authentication and confidentiality can be combined so that Ramesh can sign a confidential message which is encrypted before transmission. The steps required are as follows :

1. Ramesh generates a signature *c* for his message *m* as in the Authentication scheme

$$c = \text{encrypt}_{Rd}(\text{SHA}(m))$$

2. Ramesh generates a random session key k and encrypts the message m and the signature c using a symmetric cryptosystem to obtain ciphertext C

$$C = encrypt_k(m,c)$$

3. He encrypts the session key k using Suresh public key

$$k' = encrypt_{Be}(k)$$

4. Ramesh sends Suresh the values (k', C)

5. Suresh receives k' and C and decrypts k' using his private key Bd to obtain the session key k

$$k = decrypt_{Bd}(k')$$

6. Suresh decrypts the ciphertext C using the session key k to obtain m and c

$$(m, c) = decrypt_k(C)$$

7. Suresh now has the message m. In order to authenticate it he uses Ramesh public key Re to decrypt the signature c and hashes the message m using SHA-1.

If $SHA(m) = decrypt_{Re}(c)$

Then the message is authenticated.

➜ 4. PGP Compression

PGP can also compress the message if desired. The compression algorithm is ZIP and the decompression algorithm is UNZIP.

1. The original message m is signed as before to obtain

$$c = encrypt_{Rd}(SHA(m))$$

2. Now the original message m is compressed to obtain

$$M = ZIP(m)$$

3. Ramesh generates a session key k and encrypts the compressed message and the signature using the session key

$$C = encrypt_k(M,c)$$

4. The session key is encrypted using Suresh's public key as before.

5. Ramesh sends Suresh the encrypted session key and ciphertext C.

6. Suresh decrypts the session key using his private key and then uses the session key to decrypt the ciphertext C to obtain M and c

$$(M,c) = decrypt_k(C)$$

7. Suresh decompresses the message M to obtain the original message m

$$m = UNZIP(M)$$

8. Now Suresh has the original message m and signature c. He verifies the signature using SHA-1 and Ramesh's public key as before.

⇒ 5. PGP E-Mail Compatibility

- Many electronic mail systems can only transmit blocks of ASCII text. This creates a problem when sending encrypted data which is in cipher text form might not correspond to ASCII characters that can be transmitted.
- PGP overcomes this problem by using **Radix-64 conversion.**
- Suppose the text to be encrypted has been converted into binary using ASCII coding and encrypted to give a cipher text stream of binary. Radix-64 conversion maps arbitrary binary into printable characters.

1. The binary input is split into blocks of 24 bits (3 bytes).
2. Each 24 block is then split into four sets each of 6-bits.
3. Each 6-bit set will then have a value between 0 and $2^6 - 1$ (= 63).
4. This value is encoded into a printable character.

➜ 6. PGP Segmentation

- Another constraint of e-mail is that there is usually a maximum message length.
- PGP automatically blocks an encrypted message into segments of an appropriate length. On receipt, the segments must be re-assembled before the decryption process.
- Following are the service offered by the PGP:

1. Authentication
2. Confidentiality
3. Non-repudiation
4. Integrity
5. Compression
6. E-mail compatibility
7. Segmentation

4.7.2 Backdoors and Key Escrow in PGP

→ (SPPU - May 16)

Q. 4.7.7 What is Backdoors and Key Escrow in PGP ? (Ref. Sec. 4.7.2) May 16, 9 Marks

- Suppose, we have saved your password in laptop. So, anyone who has access the laptop, can get unauthorized access to your account. And that is a simple way of saying what a Backdoor is.
- A Backdoor is a method for bypassing normal authentication in a system and thus, provide unauthorized remote access to the system to malicious users.
- A backdoor is a "feature" in the software of PGP like an utility functions but not in the encryption algorithm that allows an outside party to decrypt which is encrypted by PGP.
- A Backdoor may be implemented as a hidden part of a program or a separate program or even be implemented by hardware.
- Just to give an example, in 2003 a Backdoor was planted in Linux Kernel. In a conditional statement for checking root access permission, '==' was replaced with '='. As a result, it gave unauthorizd access to malicious callers. Even very recently, in 2015, **Juniper Networks** have warned about a malicious Backdoor in their firewalls that automatically decrypts VPN traffic.
- There are two types of Backdoors – **Object Code Backdoors** and **Asymmetric Backdoors.**
- In **Object Code Backdoors**, software source code remains unchanged, but the object code gets modified maliciously. As the object code is designed to be machine readable, it becomes much more difficult to detect. These type of Backdoors are inserted in the on-disk object code or inserted at some point during compilation, linking or loading.
- Recompiling the software source code may get rid of the Backdoors. So, malicious users sometimes change the compiler source code in such a way that, whenever it compiles, links and loads the source code, the Backdoor is inserted. These Backdoors can be fixed by recompiling the compiler and removing the Backdoor inserting codes.
- Normally, Backdoors are symmetric. Anyone who finds the Backdoor, can in turn use it. But, **Asymmetric Backdoors** can be exploited only by the attacker who plants it, even if the Backdoor implementation becomes public. This type of attacks are termed as **Kleptography** and they can be carried out in software, hardware or in combination of both. The theory of Asymmetric Backdoors is a part of a larger field named **Cryptovirology.**

Counter measures

- Once Backdoors are detected, rebuild a clean system and transfer data.
- Another method is to use Diverse Double Compiling or DDC. It requires a different compiler and the source code of the compiler to be tested. That source code, while compiled with two different compilers, would result in two different stage-1 compilers showing same behaviour.
- Thus, the same source code compiled in two different stage-1 compilers, must result in two identical stage-2 compilers. This method was applied to verify that C compiler of GCC Suite contained no Trojan, using the icc as the other compiler. Normally, Operating Systems vendors implement these type of methods to make sure they are not distributing a compromised system.

Key Escrow

- Key escrow is a cryptographic key exchange process in which a key is held in escrow, or stored, by a third party. A key that is lost or compromised by its original user(s) may be used to decrypt encrypted material, allowing restoration of the original material to its unencrypted state.
- Key escrow systems provide a backup source for cryptographic keys. Escrow systems are somewhat risky because a third party is involved.

- The Clipper Chip was a U.S. government encryption chipset introduced in 1993. The chipset was promoted as an encryption device with a government-held (escrow) master key to facilitate encryption in the face of security threats. The controversial Clipper Chip was defunct by 1996, but the concept evolved into the Pretty Good Privacy (PGP) encryption tool, which is used worldwide.
- Key escrow (also known as a "fair" cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys.

Syllabus Topic : S/MIME

4.8 S/MIME

→ **(SPPU - Dec. 13, May 15)**

Q. 4.8.1 Describes the functions of S/MIME. Also describes the functions of Cryptographic Algorithms used in S/MIME. (Ref. Sec. 4.8) **Dec. 13. 10 Marks**

Q. 4.8.2 Write a short note on S/MIME. (Ref. Sec. 4.8) **May 15. 4 Marks**

- S/MIME stands for Secure/Multipurpose Internet Mail Extensions provides a de facto standard to send and receive secure Multipurpose Internet Mail Extensions (MIME) data.
- S/MIME developed by RSA Data Security, S/MIME version 3 is now being maintained by the S/MIME Working Group of the Internet Engineering Task Force (IETF).
- As explained in PGP, because of large number of increasing Internet usage for sending and receiving emails over an insecure communication channel increases the risk of data modification, confidentiality, authentication and non-repudiation.
- We need our mail should be safe from unauthorized users. It should reach to intended recipients only. Based on the popular Internet usage, S/MIME provides the different cryptographic security services to secure electronic messaging applications: authentication, message integrity and non-repudiation and data security.
- Traditional mail user agents (MUAs) uses S/MIME to add cryptographic security services to mail that is sent, and to interpret cryptographic security services in mail that are received.
- S/MIME uses both symmetric as well as asymmetric key cryptographic techniques to sign and encrypt e-mail. Sender as well as receiver of the mail has two keys: A private key, which is kept secret and a public key, which is available to everyone. Mails encrypted using senders private key can only be decrypted using his public key and vice versa.

Mail signing concepts

- Generally when sender sends a message he could just encrypt the message using his private key at the receiver side. If the message decrypted using senders public key then we can say that the message came from authenticated user and its content cannot be modified, because a message, that can be decrypted using senders public key must have been encrypted using senders private key only. To increase the performance S/MIME uses the following concepts.
- The message is not encrypted using receivers public key instead of that it is encrypted using a randomly created symmetric session key because symmetric key is faster than asymmetric key cryptography.
- The generated session key is encrypted using receiver's public key so that only receivers can retrieve the session key and thus decrypt the original message.
- The following steps are taken in order to create an encrypted message :
  1. The user writes the message in plain text.
  2. Triple DES algorithm is used to create random session key
  3. The message is being encrypted using the random session key.
  4. For every recipient, the session key is being encrypted using the recipient's public key.

- S/MIME-e-mail software must support Secure Hash Algorithms (SHA-1) standard and Message Digest Algorithm (MD5) in order to provide backward-compatibility with MD5-digested S/MIME v2 messages.

Table 4.8.1 : Contents of encrypted mail

Encrypted Mail	
Message encryption	Message encrypted with the session key.
Session key encryption	Session encrypted with the recipient's public key can only be decrypted using his private key only.
Identify algorithm	To tell receiver which decryption algorithm used..
Sender's public key	To enable the recipient to encrypt his response.

- **Digital signatures :** To encrypt the message digest, the S/MIME client must support DSS (Digital Signature Standard) and should support RSA (Rivest, Shamir, Adleman)
- **Content Encryption:** To encrypt the message content (symmetrically, using a random session key), Triple DES must be supported
- **Key encryption and management :** To encrypt the session key, Diffie-Hellmann algorithm must be supported. Key S/MIME uses X.509v3 certificates to determine whether a public key used to verify a signature is trustworthy. The certificate signed by Certificate Authority (CA) for claiming that public key belongs to the person.
- S/MIME (Secure/Multipurpose Internet Mail Extensions) is a specification for secure electronic mail providing authentication and confidentiality. It is not a particular software product but a standard designed to be implemented by various e-mail vendors, so that any two S/MIME-supporting mail clients can communicate securely.

Syllabus Topic : Secure Electronic Transaction

4.9 Secure Electronic Transaction (SET)

➔ **(SPPU - Dec. 13, May 15, May 16)**

Q. 4.9.1 Explain key features of SET. (Ref. Sec. 4.9)

Q. 4.9.2 With help of diagram explain SET Participants. (Ref. Sec. 4.9) **Dec. 13, 8 Marks**

Q. 4.9.3 Explain working principles of SET with suitable diagram. (Ref. Sec. 4.9) **May 15, 8 Marks**

Q. 4.9.4 What is secure Electronic Transaction ? (Ref. Sec. 4.9) **May 16, 6 Marks**

- SET stands for Secure Electronic Transaction. It is an encryption and security specification protocol designed to protect credit card transactions over an insecure channel such as Internet.
- SET is not a payment system it is set of rules and regulations designed to protect credit card payments of users, employee over an open network such as Internet in a secure way.
- SET was developed in 1996 by VISA and MasterCard, with participation from different leading technology companies, which includes Microsoft, IBM, Netscape, RSA, Teresa Systems and VeriSign.

After testing of the SET in 1998 it declares as a standard for safeguarding credit card purchases made over open networks and made it available to users with following requirements.

- SET create a secure communications channel among all parties involved in a transaction.
- SET provides privacy because the information is only available to sender, receiver and bank or the communication parties' involved in transaction.
- SET provides confidentially, only sender and his intended receiver should be able to access the contents of a message. It assures to card holder is that is safe and accessible only to the intended recipient.
- SET provides integrity of the message. Integrity gives assurance that data received exactly as sent by an authorized entity. (No alteration, no modification, no deletion and no intersection etc.).

4.9.1 SET Participants

→ (SPPU - May 17)

Q. 4.9.5 List and explain various participants involved in Secure Electronic Transaction (SET). (Ref. sec. 4.9.1) **May 17, 5 Marks**

Following are the components of the Secure Electronic Transaction (SET) which involves in the electronic payment as shown in Fig. 4.9.1.

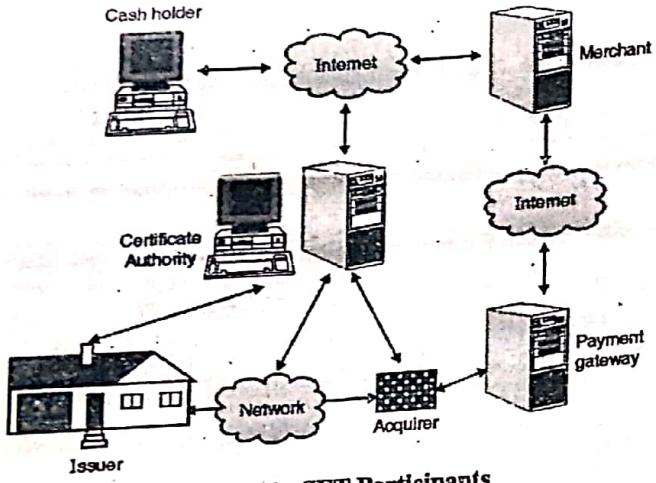

Fig. 4.9.1 : SET Participants

- **The cardholder :** called as buyer in the transaction who initiates the transaction.
- **The merchant :** Also called seller of goods and services which maintains an account with a bank or acquirer.
- **The acquirer :** Also known as bank or financial institution. The financial institution that establishes an account with a merchant and processes payment card authorizations and payments.
- **The issuing bank :** Bank that maintains the account of the buyer and issues a credit card to the buyer and also sets limit on the amount of purchases.
- **Certification Authority (CA) :** Certification Authority (CA) is a trusted unit that helps to issue certificates.
- A CA takes the certificate request from owner, verifies the requested information according to the terms and conditions of the CA, and uses its private key to apply digital signature to the certificate.
- Responsibility of the CA is to identify the correct identity of the person who asks for a certificate to be issued, and make sure that the information contained within the certificate is legal and later digitally sign on certificate.
- This is an entity that is trusted to issue X509v3 public-key certificates for cardholders, merchants, and payment gateways.
- **Payment gateways :** It is designated third party that processes merchant payment messages. The merchant exchanges Secure Electronic Transaction messages with the payment gateway over the Internet, while the payment gateway has some direct or network connection to the acquirer's financial processing system.
- Following are the steps of interactions used in SET protocol :

1. **The customer opens the account :** Once the customer obtains a credit card account, such as MasterCard or Visa, from the bank which supports electronic payment and Secure Electronic Transaction then customer may proceed for future communication over network.
2. **The customer receives a certificate :** After suitable verification of identity, the customer receives an X.509v3 digital certificate, which is signed by the bank which verifies the customers RSA public key and its expiration date.
3. **Merchants have their own certificates :** A merchant have two public keys one for signing message and another for key exchange The merchant also needs a copy of the payment gateway's public-key certificate.
4. **The customer places an order :** Here customer first browsing through the merchant's Web site to select items and determine the price. The customer now sends its list of items to be purchased to the merchant. Upon receiving list of items from customer merchant returns an order from containing the list of items, their price, a total price, and an order number to the customer.

5. **The merchant is verified :** Along with order number, the merchant sends a copy of its certificate, so that the customer can verify that he or she is dealing with a valid merchant store.

6. **The order and payment is verified :** The customer sends both order and payment information to the merchant, along with the customer's certificate (approved by CA). Customer also confirms the purchase of the items in the order form. The payment contains credit card details. The payment information is encrypted in such a way that it cannot be read by the merchant. The customer's certificate enables the merchant to verify the customer.

7. **The merchant requests payment authorization:** The merchant sends the payment information to the payment gateway for authentication as well as to check whether customer's available credit is sufficient for this purchase.

8. **The merchant confirm the order :** Upon receiving payment confirmation from customers credit, the merchant sends confirmation of the order to the customer.

9. **The merchant provides the goods or service :** After all verification the merchant provides the goods or service to the customer.

10. **The merchant request payment :** This request is sent to the payment gateway, which handles all of the payment processing.

MIME

S/MIME (for Secure MIME, or Secure Multipurpose Mail Extension) is a security process used for e-mail exchanges that makes it possible to guarantee the confidentiality and non-repudiation of electronic messages

- S/MIME is based on the MIME standard, the goal of which is to let users attach files other than ASCII text files to electronic messages.
- The MIME standard therefore makes it possible to attach all types of files to e-mails. S/MIME was originally developed by the company RSA Data Security.
- Ratified in July 1999 by the IETF, S/MIME has become a standard, whose specifications are contained in RFCs 2630 to 2633.
- The S/MIME standard is based on the principle of public-key encryption. S/MIME therefore makes it possible to encrypt the content of messages but does not encrypt the communication.
- The various sections of an electronic message, encoded according to the MIME standard, are each encrypted using a session key.
- The session key is inserted in each section's header, and is encrypted using the recipient's public key. Only the recipient can open the message's body, using his private key, which guarantees the confidentiality and integrity of the received message.
- In addition, the message's signature is encrypted with the sender's private key. Anyone intercepting the communication can read the content of the message's signature, but this ensures the recipient of the sender's identity, since only the sender is capable of encrypting a message (with his private key) that can be decrypted with his public key.

Chapter Ends...

□□□