4 IP Security, Transport Level Security and Email Security

Syllabus

At the end of this unit, you should be able to understand and comprehend the following syllabus topics :

- IP level Security
  - Introduction to IPSec
  - IPSec Architecture
  - Protection Mechanism
    - AH
    - ESP
- Transport level security
  - VPN
- Need Web Security considerations
  - Secure Sockets Layer (SSL) Architecture
  - Transport Layer Security (TLS)
  - HTTPS
  - Secure Shell (SSH) Protocol Stack
- Email Security
  - Secure Email S/MIME

4.1 IP Security

- IP stands for Internet Protocol. IP defines a set of protocols that can be used for communication between any two devices on the network. A network protocol is a standard set of rules that determines how systems will communicate across networks.
- Two different systems that use the same protocol can communicate and understand each other very similar to how two people can communicate and understand each other by using the same language.
- IP provides addressing and routing mechanisms for each packet of data that needs to move across the network. Each device on the network must have a unique IP address to communicate with any other device on the network.

Note : It is assumed that you have a general understanding of computer networking. While this section does not dive deeper into computer networks, it focuses on specific security topics around networking.

4.1.1 IPv4

- IPv4 is IP version 4. This is the most common IP addressing scheme used today despite certain challenges. It is 32-bit long and thus has an address space of 2^{32} = 4,294,967,296. This means you can maximally have 4,294,967,296 (approximately 4.3 billion) IPv4 addresses. There are many more devices than the number 4,294,967,296. Fig. 4.1.1 shows an outline of the IPv4 header.

Offsets Octet	Octet	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Fig. 4.1.1 : IPv4 header

- An example of IPv4 address looks like 121.56.78.214.

4.1.2 IPv6

- IPv6 was created to address the limitation of IPv4 to have only 4,294,967,296 IP addresses due to 32-bit length. IPv6 more or less provides the similar addressing and routing capabilities but one core difference between IPv4 and IPv6 is the address space. IPv6 address is 128-bit long and thus you can have 2^{128} IPv6 addresses!
- Fig. 4.1.2 shows an outline of the IPv6 header.

Offsets Octet	Octet	0								1								2								3							
	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class												Flow Label															
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source IP Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination IP Address																															
28	224																																
32	256																																
36	288																																

Fig. 4.1.2 : IPv6 header

- An example of IPv6 address looks like 2001:0:9d38:6abd:2c37:10da:8554:4234.

4.1.3 Internet Protocol Security (IPSec)

✍ ***Definition :*** *IPSec is a suite of protocols that protects IP traffic.*

- IP does not have any integrated security mechanisms by itself and hence IPSec (short form for IP Security) is additionally used to provide security for IP traffic.
- The IPSec suite consists of security protocols shown in Table 4.1.1.

Table 4.1.1 : Security protocols provided by IPSec

Sr. No.	Protocol Name	Functionality Provided
1.	Authentication Header (AH)	Data Integrity Data Origin Authentication Protection from replay attacks
2.	Encapsulating Security Payload (ESP)	Confidentiality Data Origin Authentication Data Integrity
3.	Internet Security Association and Key Management Protocol (ISAKMP)	Framework for Authentication and Key Exchange
4.	Internet Key Exchange (IKE)	Authenticated keying material for use with ISAKMP

- Here IPSec is a framework. It does not mandate which hashing and encryption algorithms should be used or how keys should be exchanged between the communicating devices. Key management can be handled manually or automated by a key management protocol such as ISAKMP.

1. Security Association

- Security Association (SA) is a fundamental concept with respect to IPSec.

✍ ***Definition :*** *Security Association holds several information that determines how security services would be consumed by the communicating devices.*

- IPSec provides many options for performing security services such as encryption, integrity and authentication. The network devices that wish to establish an IPSec connection, must negotiate and arrive at exactly which algorithms and parameters to use for the chosen IPSec security services. The security association is a mechanism to hold all the agreed terms (algorithms, parameters, etc.) for a given IPSec communication session.

2. Modes of operation

IPSec can work in two modes.

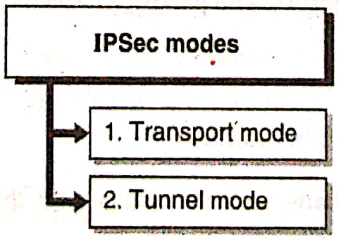


Fig. 4.1.3 : IPSec Modes

A. Transport Mode

In this mode, only the payload (data) part of the information is protected. The addressing and routing information is not protected. It is like a sealed envelope with address on it. The message inside the envelop is protected whereas the source and destination addresses are not.

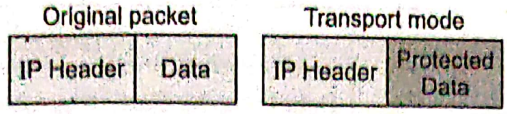


Fig. 4.1.4 : Transport Mode

B. Tunnel Mode

In this mode, both the payload (data) as well as the addressing information is protected. In this mode, the entire packet is protected, and a new IP header is added by IPSec. The original IP header information along with the payload information is protected. Tunnel mode provides more security than the transport mode.

Fig. 4.1.5 : Tunnel Mode

3. Applications / Benefits / Usage of IPSec

i. Establish Virtual Private Network (VPN)

IPSec is predominantly used to establish VPN connection. VPN connections are generally used to access private networks over the internet. For example, you can access your college or your organization's network from home over the internet.

ii. Connecting two or more branch networks

IPSec can be used to extend or connect branch networks. For example, if you have two branches of office each using its own network, the branches can be connected using IPSec. The network traffic then can securely move between the branches.

iii. General security benefits

IPSec adds general security benefits to the core IP protocol. It provides benefits such as data confidentiality, data integrity, data origin authentication and protection from several attacks on the core IP protocol.

4. How does IPSec work?

Overall, communication over IPSec has 5 broad steps.

1. **Initiate IPSec process :** IPSec communication begins with the identification of traffic that requires IPSec security.
2. **IKE Phase 1 :** In this phase, the IKE SAs are negotiated and agreed.
3. **IKE Phase 2 :** In this phase, next set of SAs for actual data transfer are negotiated and agreed.
4. **Data Transfer :** Data is transferred between the communicating entities.
5. **Termination :** The IPSec connection is terminated once the data transfer is complete.

4.1.4 Authentication Header (AH)

Definition : *The Authentication Header (AH) protocol provides data integrity and data original (source address) authentication over the network communication.*

- It also provides replay protection. It does not provide encryption.
- AH calculates the Integrity Check Value (ICV) over non-changing fields of the IP header:
  - Next Header
  - Payload Len
  - Reserved
  - Security Parameter Index (SPI)
  - Sequence number
  - Padding bytes
- ICV is a hash value which is often computed using SHA-1 or other hashing algorithms.
- In the transport mode, Fig. 4.1.6 shows the simplistic diagram of before and after applying AH.

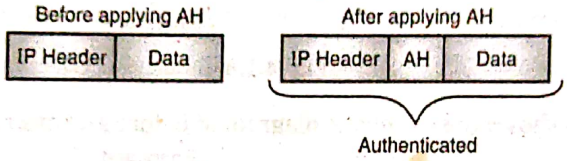


Fig. 4.1.6

- In the tunnel mode, Fig. 4.1.7 shows the simplistic diagram of before and after applying AH.

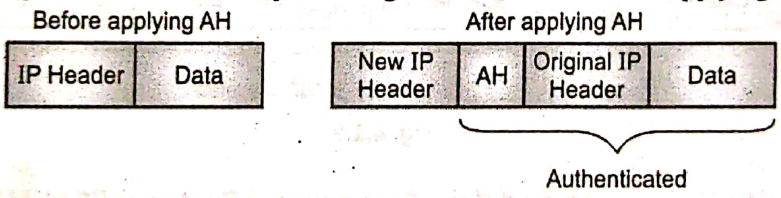


Fig. 4.1.7

- As we discussed earlier on hash values, hash values provide integrity. AH uses hashing algorithms to find out hash value of the IP header and attaches it with the IP header. This way it not only provides data integrity (since hash is calculated on payload as well) but also data origin integrity or authentication (since source address is part of the IP header as well).

4.1.5 Encapsulating Security Payload (ESP)

Definition : *The ESP protocol is designed to provide confidentiality (through encryption), data integrity and data origin authentication over network communication.*

- ESP can be applied with AH or without AH. Here ESP can itself provide integrity. It does not need AH for integrity. You have an option to additionally calculate integrity using AH. ESP in transport mode encrypts the actual payload (data) so that it cannot be read by an unauthorized entity. In tunnel mode, the IP header information is encrypted as well.

- If you choose integrity service, the Integrity Check Value (ICV) is calculated on the following fields in the IP header.
  - Security Parameter Index (SPI)
  - Sequence Number
  - Payload Data
  - ESP trailer
- If you choose confidentiality service, the ciphertext consists of the following fields in the IP header.
  - Payload (Data)
  - ESP trailer
- In the transport mode, Fig. 4.1.8 shows the simplistic diagram of before and after applying ESP.

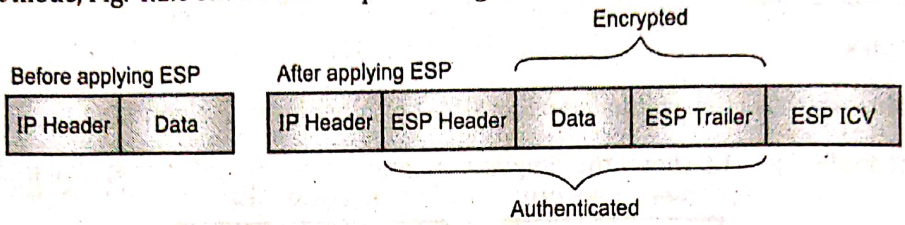


Fig. 4.1.8

- In the tunnel mode, Fig. 4.1.9 shows the simplistic diagram of before and after applying ESP.

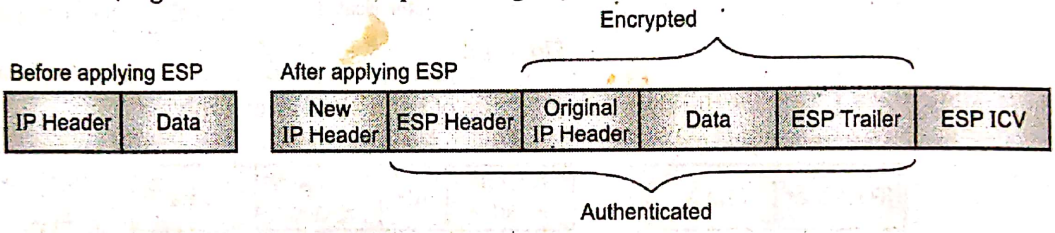


Fig. 4.1.9

4.1.6 Internet Security Association and Key Management Protocol (ISAKMP)

✍ ***Definition :*** *ISAKMP provides a framework for authentication and key exchange.*

- ISAKMP does not define the exact algorithms to be used. It is just a framework within which various exchange protocols can work. ISAKMP defines the procedures for
  - Authenticating communication devices
  - Creation and management of Security Associations (SA)
  - Key generation techniques
  - Threat mitigation
- ISAKMP is distinct from key exchange protocols in order to cleanly separate the details of security association management (and key management) from the details of key exchange.
- There may be many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. ISAKMP serves as this common framework.

- Fig. 4.1.10 shows a simplistic diagram of ISAKMP header.

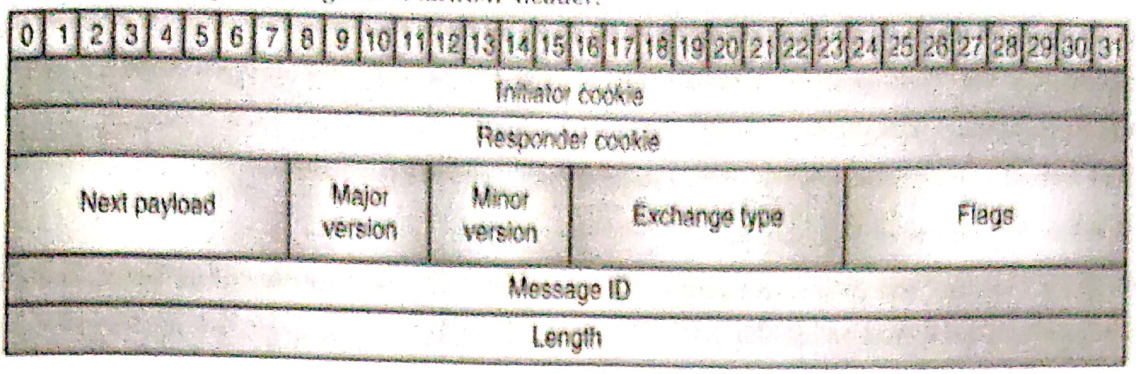


Fig. 4.1.10 : ISAKMP header

1. **Initiator cookie (8 bytes)** : The cookie of entity that initiated Security Association (SA) establishment, SA notification, or SA deletion.
2. **Responder cookie (8 bytes)** : The cookie of entity that is responding to a SA establishment request, SA notification, or SA deletion.
3. **Next Payload (1 byte)** : Indicates the type of first payload in the message. ISAKMP supports the following payload types:
   - None
   - Security Association
   - Proposal
   - Transform
   - KeyExchange
   - Identification
   - Certificate
   - Certificate Request
   - Hash
   - Signature
   - Nonce
   - Notification
   - Delete
   - VendorID
   - NAT Discovery Payload
   - NAT Original Address Payload
   - Reserved
   - PrivateUse
4. **Major version (4-bits)** : Major version of the ISAKMP protocol in use.

5. **Minor version (4-bits) :** Minor version of the ISAKMP protocol in use.
6. **Exchange Type (1 byte) :** The type of exchange in a given ISAKMP session. The primary difference between exchange types is the ordering of the messages and the payload ordering within each message.
7. **Message ID (4 bytes) :** The unique message identifier.
8. **Length (4 bytes) :** The length, in bytes, of the total message (header + payloads).

- ISAKMP offers two phases of negotiation.
  - **Phase 1 :** In the first phase, two entities agree on how to protect further negotiation traffic between themselves, establishing an ISAKMP SA.
  - **Phase 2 :** The second phase of negotiation is used to establish security associations for other security protocols. This second phase can be used to establish many security associations. The security associations established by ISAKMP during this phase can be used by a security protocol to protect many message/data exchanges.

4.1.7 Internet Key Exchange (IKE)

✍ ***Definition :*** *Internet Key Exchange (IKE) is the protocol used to set up a Security Association (SA) in the IPsec protocol suite.*

- IPSec currently uses IKE version 2.
- Recall from our earlier discussion on security association – A security association is a set of negotiated terms (algorithms, parameters, etc.) between two communicating entities such that these terms can be used in successive communication.
- The following attributes are used by IKE and are negotiated as part of the ISAKMP Security Association:
  - Encryption algorithm
  - Hashing algorithm
  - Authentication method
  - Information about a group over which to do Diffie-Hellman exchange
- All of these attributes are mandatory and MUST be negotiated between the communicating entities. IKE supports the following attributes for negotiation.

Attribute For	Supported Attributes
Encryption algorithm	DES, IDEA, Blowfish, 3DES, CAST, RC5, AES
Hashing algorithm	MD5, SHA, TIGER
Authentication method	Pre-shared key, DSS Signature, RSA Signature, Encryption with RSA, Revised encryption with RSA
Group information	MODP (modular exponentiation group), ECP (elliptic curve group over GF[P]), EC2N (elliptic curve group over GF[2^N])

- IKE works in two phases.

- **In Phase 1**, following functions are carried out:
  - Mutual authentication of the communicating entities.
  - Negotiating cryptographic parameters.
  - Creating session keys.

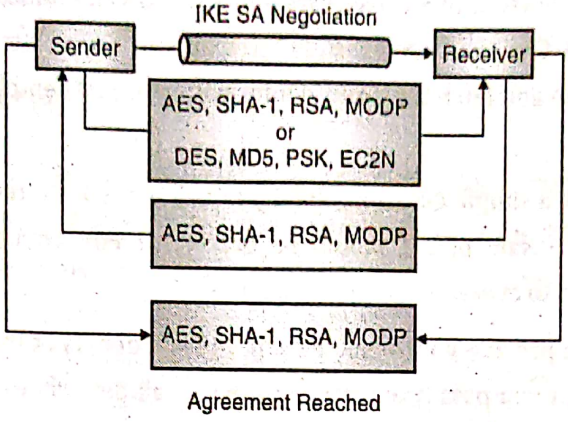


Fig. 4.1.11

- **In Phase 2**, an IPSec tunnel is negotiated by creating keying material for the IPSec tunnel to use (either by using the IKE phase one keys as a base or by performing a new key exchange).

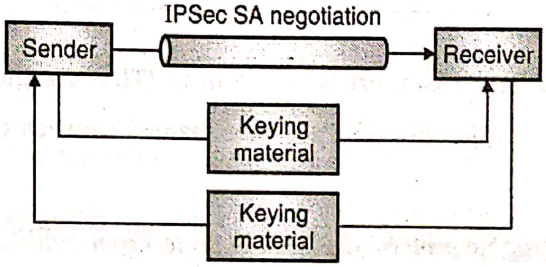


Fig. 4.1.12

4.1.8 OAKLEY Key Determination Protocol

Definition : *OAKLEY is a key determination protocol using which two authenticated parties can agree on secure and secret keying material.*

- It is based on the Diffie-Hellman key exchange algorithm. The OAKLEY protocol has compatibility with the ISAKMP protocol for managing security associations, user-defined abstract group structures for use with the Diffie-Hellman algorithm, key updates, and incorporation of keys distributed via out-of-band mechanisms.
- The OAKLEY protocol is used to establish a shared key with an assigned identifier and associated authenticated identities for the two parties. The name of the key can be used later to derive security associations for AH and ESP. At a high-level, there are three components of the key determination protocol.
  1. Cookie exchange
  2. Diffie-Hellman key exchange
  3. Authentication

Note : Current IPsec implementations actually use IKEv2. But its predecessor, IKEv1, was based on the OAKLEY and Security Key Exchange Mechanism (SKEME) protocols.

4.2 Web Security

- World Wide Web or just web is a collection of web servers that run several websites that hold the desired information.
- The Internet as a whole is a collection of such servers and various communication devices and protocols. You mostly use browsers (such as Chrome, Firefox, Internet Explorer, Safari, etc.) or applications (for example, Mobile Apps) to browse the web and fetch the desired information or just complete a desired interaction such as making a purchase.
- Let me pause you here and ask a simple question. Don't you feel that your interaction with the Internet (which generally is an insecure and unsafe place) should be protected? For example, if you type your Facebook password, should it be available to everyone on the network?
- To make a purchase when you provide your bank account information, isn't that information very confidential and requires secure handling as you pass it through your device all the way to the website? Yes, I am sure you understand that your interaction with the web requires security. There is one protocol that we all need - SSL (obsolete now) followed by TLS (currently used). Let's learn about it.

4.3 Secure Socket Layer (SSL)

- The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are the most widely use web security protocol. It is essentially a protocol that provides a secure channel between two machines operating over the Internet or an internal network.

✍ ***Definition :*** *SSL is a cryptographic protocol designed to protect communication between two entities.*

- SSL underwent several revisions and is now followed by a more secure protocol called TLS. Table 4.3.1 shows a quick version history of SSL/TLS.

Table 4.3.1 : History of SSL/TLS

Protocol	Published	Status
SSL 1.0	Unpublished	Unpublished
SSL 2.0	1995	Obsoleted in 2011
SSL 3.0	1996	Obsoleted in 2015
TLS 1.0	1999	To be obsolete in 2020
TLS 1.1	2006	To be obsolete in 2020
TLS 1.2	2008	Currently good
TLS 1.3	2018	Currently good

Goals of SSL

1. **Cryptographic Security :** Establish and provide a secure connection between two parties.
2. **Interoperability :** Two unrelated applications should be able to establish SSL connection.
3. **Extensibility :** Provides a framework for using various algorithms and methods without changing the protocol.
4. **Efficiency :** Performance enhancement mechanism to avoid overloading the system when protocol is in use.

4.3.1 Overview of SSL Protocol

- SSL protocol works in layers. At each layer, messages may include fields for length, description, and content. SSL takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. On the other side, received data is decrypted, verified, decompressed, and reassembled and then delivered to higher level clients.

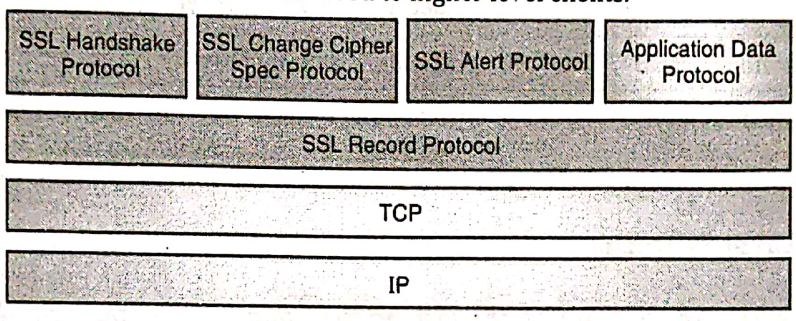


Fig. 4.3.1

- Let's learn about each one of them in detail.

4.3.1(A) Session and Connection States

- An SSL session is stateful which means that parameters negotiated during the session establishment persist (stay the same) until the session is terminated. The SSL handshake protocol coordinates the states of the client and server. It is thus important to preserve Session and Connection States.
- Table 4.3.2 summarizes a Session State.

Table 4.3.2 : Session state

Sr. No.	Fields	Purpose
1.	Session Identifier	A session ID chosen by the server to identify an active or resumable session state.
2.	Peer Certificate	X.509 Certificate of the other party in the communication.
3.	Compression method	The algorithm used to compress data prior to encryption.
4.	Cipher Specification	Chosen encryption algorithm such as AES and a hash algorithm such as SHA.
5.	Master Secret	48-byte secret shared between the client and server.
6.	Is resumable	A Boolean flag indicating whether a session ID can be used to initiate new connections.

- Table 4.3.3 summarizes a Connection State.

Table 4.3.3 : Connection state

Sr. No.	Fields	Purpose
1.	Server and client random	Byte sequences for establishing connection
2.	Server write MAC secret	The secret used in MAC operations on data written by the server
3.	Client write MAC secret	The secret used in MAC operations on data written by the client
4.	Server write key	The bulk cipher key for data encrypted by the server and decrypted by the client
5.	Client write key	The bulk cipher key for data encrypted by the client and decrypted by the server
6.	Initialization vectors	Random number to initialize encryption operation
7.	Sequence numbers	Sequence numbers for transmitted and received messages for each connection

4.3.1(B) SSL Record Layer Protocol

✍ ***Definition*** *: The SSL Record Layer is the last protocol that receives the raw data from the higher application layers and other SSL protocols such as handshake.*

- Its core function is to facilitate (perform) data transfer. The basic unit of data in SSL is a record. Each record consists of a five-byte record header, followed by data.
- There are four types of records in SSL.

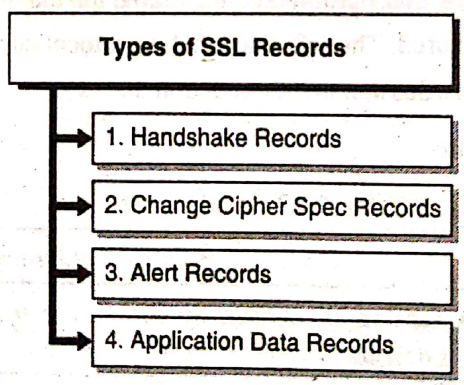


Fig. 4.3.2 : Types of SSL records

- The five-byte format of an SSL Record Header is as follows:

SSL record type (1-byte)	SSL Major Version (1-byte)	SSL Minor Version (1-byte)	Length of data in the record (2-bytes)

Fig. 4.3.3 : SSL record header format

- Fig. 4.3.4 shows a simplistic block diagram of steps involved in the SSL Record Protocol.

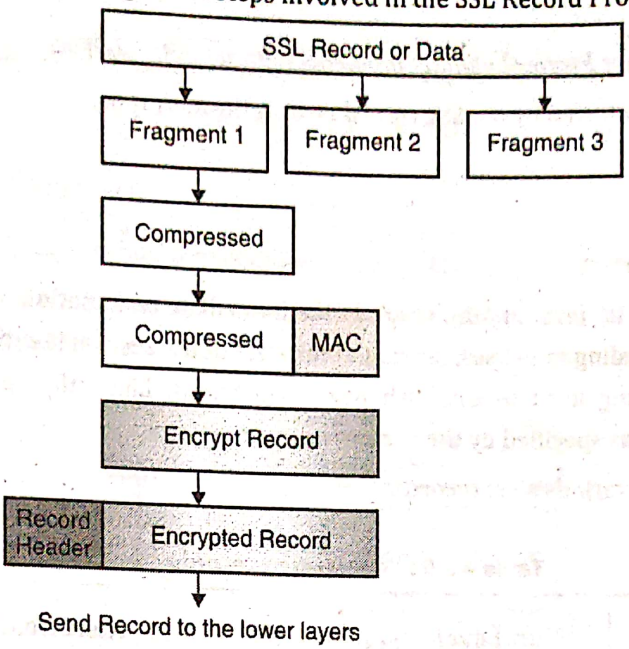


Fig. 4.3.4 : Block diagram of SSL record protocol

- At a high level the SSL Record Protocol performs three operations as shown in Table 4.3.4.

Table 4.3.4 : Operations of SSL record protocol

Sr. No.	Operation Performed	Purpose
1.	Fragmentation	Break original data into SSL Plaintext records of 2^{14} bytes or less
2.	Compression and Decompression	All SSLPlaintext records are compressed using the compression algorithm defined in the current session state. The compression algorithm translates an SSLPlaintext structure into an SSLCompressed structure
3.	Payload Protection	All SSLCompressed records are protected using the encryption and MAC algorithms defined in the current CipherSpec. Once the handshake is complete, the two parties have shared secrets that are used to encrypt records and compute keyed Message Authentication Codes (MACs) on their contents. The techniques used to perform the encryption and MAC operations are defined by the CipherSpec.

4.3.1(C) SSL Change Cipher Spec Protocol

Definition : *The Change Cipher Spec protocol notifies about the changes in cipher parameters.*

- The protocol consists of a single message, which is encrypted and compressed. The Change Cipher Spec Protocol notifies the communicating parties about any change in the previously negotiated Cipher Specifications or Keys.
- The keys or the algorithms need to be changed at times for reasons such as renewing the session or resuming the session. The change cipher spec message is sent by both the client and server to notify the receiving party that subsequent records will be protected under the just-negotiated CipherSpec and keys.

4.3.1(D) SSL Alert Protocol

Definition : *The SSL Alert Protocol signals problems with an SSL session.*

- One of the content types supported by the SSL record layer is the alert type.
- Alert messages notify the
  (i) Severity of the alert and
  (ii) A description of the alert
- Alert messages with a severity level of *fatal* result in the immediate termination of the connection. In this case, other connections corresponding to the session may continue, but the session identifier is invalidated, preventing the failed session from being used to establish new connections. Like other messages, alert messages are encrypted and compressed, as specified by the current connection state.
- Table 4.3.5 summarizes the various alert records.

Table 4.3.5 : Various alert records

Alert Code	Alert Message	Alert Level	Alert Description
0	close_notify	1 (Warning)	notifies the recipient that the sender will not send any more messages on this connection
10	unexpected_message	2 (Fatal)	An inappropriate message was received
20	bad_record_mac	2 (Fatal)	A record is received with an incorrect MAC
30	decompression_failure	2 (Fatal)	The decompression function received improper input
40	handshake_failure	2 (Fatal)	The sender was unable to negotiate an acceptable set of security parameters given the options available
41	no_certificate	1 (Warning)	Sent in response to a certification request if no appropriate certificate is available
42	bad_certificate	1 (Warning)	A certificate was corrupt
43	unsupported_certificate	1 (Warning)	A certificate was of an unsupported type
44	certificate_revoked	1 (Warning)	A certificate was revoked by its signer
45	certificate_expired	1 (Warning)	A certificate has expired or is not currently valid
46	certificate_unknown	1 (Warning)	Some other (unspecified) issues
47	illegal_parameter	2 (Fatal)	A field in the handshake was out of range or inconsistent with other fields

- The alert record consists of 2 bytes of information from Fig. 4.3.5.

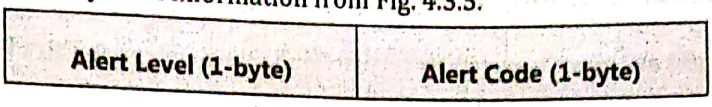


Fig. 4.3.5 : Alert record

4.3.1(E) SSL Handshake Protocols

Definition : *The cryptographic parameters of the session state are produced by the SSL handshake protocol.*

- When an SSL client and the server first start communicating, they need to agree upon certain parameters. There are also several steps that need to be carried out to establish a secure session. At a high level, the following four steps are carried out.

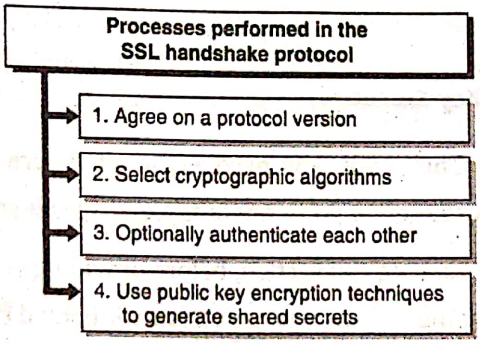


Fig. 4.3.6

- Fig. 4.3.7 illustrates the detail steps of handshake process diagram.

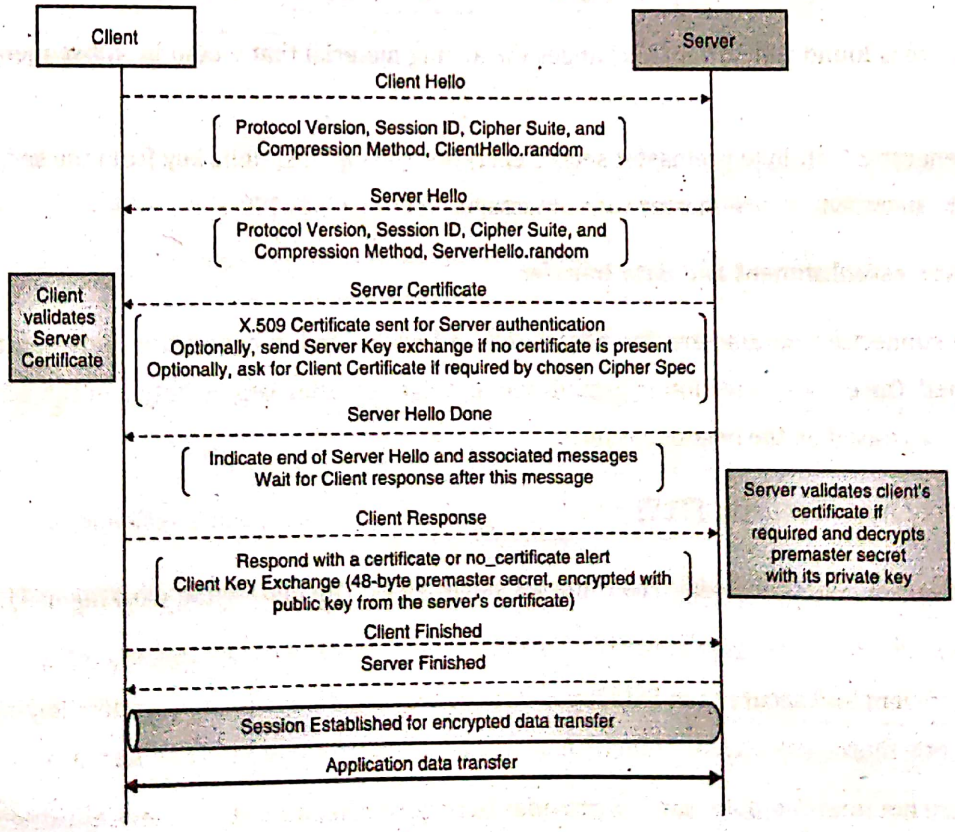


Fig. 4.3.7 : Handshake process diagram

Step 1 : Hello messages (Establish security capabilities)

The hello phase messages are used to exchange security enhancement capabilities between the client and server.

1. **Client Hello**

When a client first connects to a server it is required to send the client hello as its first message. The client can also send a client hello in response to a hello request or on its own initiative in order to renegotiate the security parameters in an existing connection. The list of parameters sent is in Fig. 4.3.7.

2. **Server Hello**

The server processes the client hello message and responds with server hello message. The list of parameters sent is in Fig. 4.3.7.

Step 2 : Server Authentication and Key Exchange

- This is the most important step. This is why you need SSL at all. Before proceeding to interact with the server you should find out "Is this really the server you want to talk to?" This is crucial.
- For example, if you want to do a banking transaction, before providing your account information, username and password, you MUST validate that the website you are on (server behind the website) is legitimate.
- In this step, the client validates the server certificate. Any certificate related errors are highlighted.

Step 3 : Client authentication and Key exchange

If the certificate is found valid, client exchanges the keying material that would be subsequently used to encrypt the messages.

The client generates a 48-byte premaster secret, encrypts it using the public key from the server's certificate and sends the result in an encrypted pre-master secret message.

Step 4 : Connection establishment and data transfer

Once all the connection parameters are negotiated and exchanged, a connection between the server and the client is established. Once the connection is established, the data transfer begins between the server and the client. The data is encrypted based on the negotiated terms.

4.3.2 Transport Layer Security (TLS)

- As you learnt earlier, SSL is obsolete. TLS replaced SSL in 1999. The underlying working of TLS is very similar to SSL.
- TLS is more efficient and secure than SSL. It provides stronger message authentication, key-material generation and supports pre-shared keys, secure remote passwords, elliptical-curve keys and Kerberos.
- TLS and SSL are not interoperable, but TLS provides backward compatibility for devices using SSL.

4.4 HTTPS

- Now that you learnt how SSL works, let's learn about one of its implementations – HTTPS application protocol.

Definition : HTTPS establishes a secure SSL/TLS tunnel before beginning data transfer.

- The Hypertext Transfer Protocol (HTTP) is an application protocol used to transfer data on distributed and connected systems. HTTPS is the secure version of HTTP.
- The 'S' at the end of HTTPS stands for 'Secure'. It means that all the communications between your client (browser, mobile apps) and the server (website, web application) is encrypted.
- HTTPS is often used to protect confidential online interactions such as online banking.
- Conceptually, HTTPS is very simple. Simply use HTTP over TLS (previously SSL) instead of HTTP. The use of TLS (previously SSL) ensures that the adequate protection mechanisms such as encryption, server authentication, hashing, and optionally client authentication are effectively applied, and the communication is adequately protected.

4.4.1 Comparison between HTTP and HTTPS

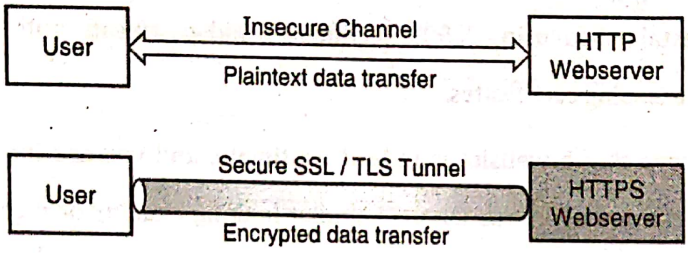


Fig. 4.4.1

Table 4.4.1

Sr. No.	HTTP	HTTPS
1.	Data transfer in plaintext	Data transfer in ciphertext
2.	Default port is 80	Default port is 443
3.	Does not require SSL/TLS or Certificates	Requires SSL/TLS implementation with Certificates
4.	URL has http://	URL has https://
5.	Should be avoided	Should be preferred
6.	Search engines do not favour the insecure websites	Improved reputation of the website in search engine
7.	Users worried about their data	Users confident about the security of their data

4.4.2 Motivation / Benefits of using HTTPS

1. Increasing sensitivity of data

- With the proliferation (widespread use) of internet, a lot of sensitive communication such as online banking, ticketing, shopping, etc. is taking place over the Internet.
- There is an ever increasing need to ensure that the communication is secure (confidentiality and integrity of the information is enforced).
- Information such as your password or credit card number is not transferred in plaintext that can potentially be captured and then misused.

2. Authentication

One of the critical use cases that HTTPS serves is that using it you can potentially authenticate a website or a business.

- HTTPS connection is established using X.509 certificates and certificate authorities do proper business or website validation before issuing certificates.
- Certificates help you prove that a website is indeed legitimate, and you are indeed interacting with the right website. This avoids several online frauds where a similar looking banking or e-commerce site can capture your confidential details.

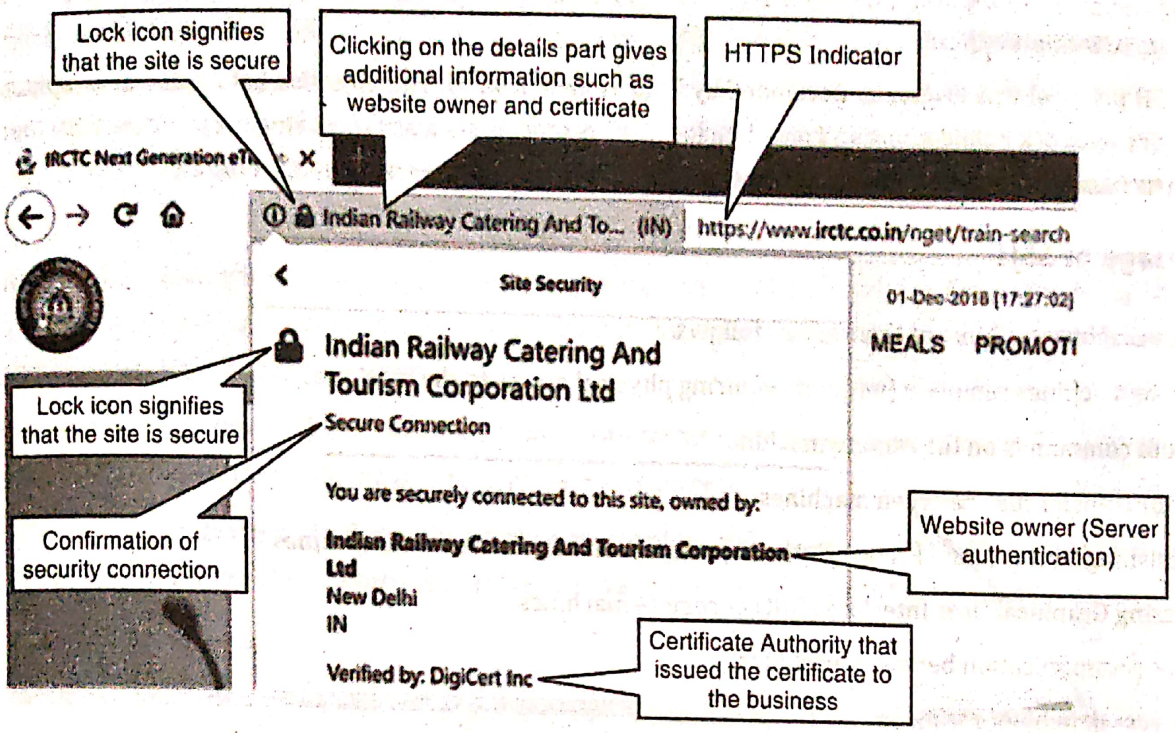
3. Privacy requirements

- Often times, the nature of communication is private even if it is not confidential. For example, your health reports, your chats, your location details, etc. require that they are adequately protected when transferred over the network.
- Use of HTTPS ensures that encryption is applied to all data seamlessly and the private information is adequately protected during transfer.

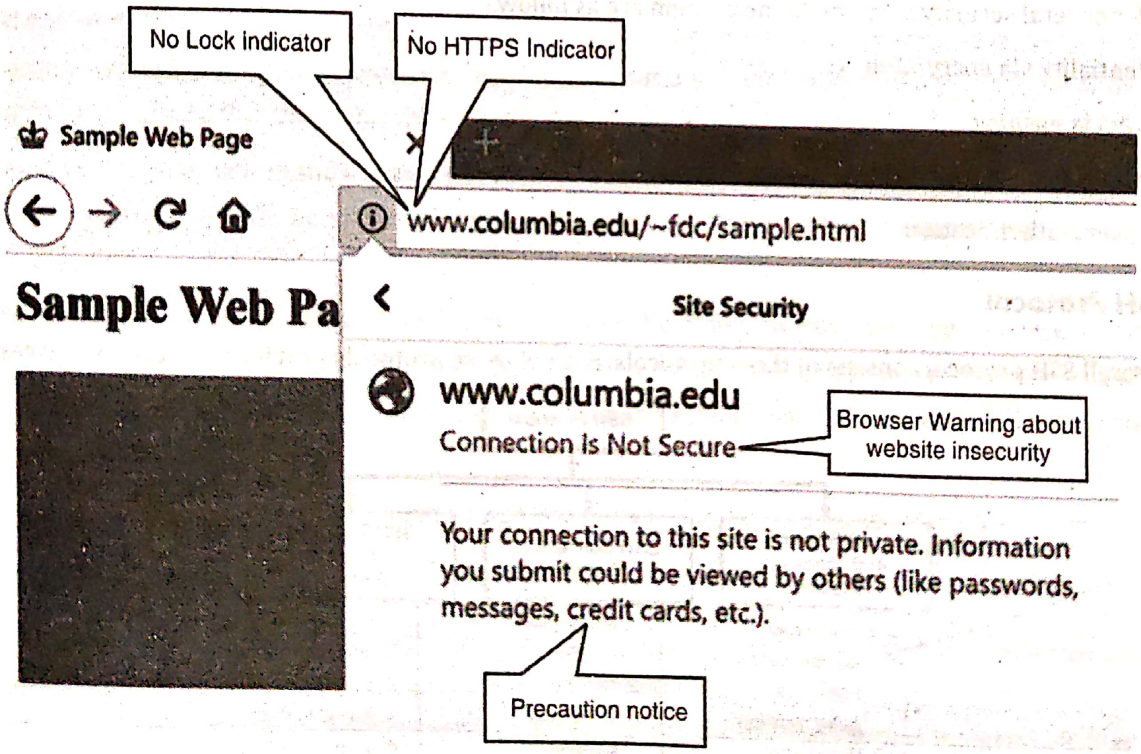
4.4.3 Format, Port Number and Representation

- Typical format of HTTPS is https://www.example.com. It works over port 443 by default. You would have seen various websites with HTTPS enabled.
- These days browsers show green colour in the URL for HTTPS protected websites and warning for non-HTTPS websites.

- Following is an example of a HTTPS protected website.



- Following is an example of a non-HTTPS website.



4.5 Secure Shell (SSH)

Definition : *Secure Shell (SSH) is a protocol for secure remote login and other secure network services over an insecure network.*

The SSH protocol was originally developed by Tatu Ylonen in 1995. The objective behind the development of SSH was to secure network connections and obsolete insecure protocols such as telnet, rlogin, rsh, rexec with their secure counterparts implemented over the SSH protocol. SSH servers typically work over TCP Port 22.

4.5.1 Usage of SSH

SSH has several usages. Some of them are as follows.

1. Login to machines remotely (without requiring physical access to the machine)
2. Execute commands on the remote machines
3. Copy or transfer files between machines
4. Establishing VPN (Virtual Private Network) connection between a set of machines
5. Accessing Graphical User Interface (GUI) of remote machines
6. Secure communication between machines
7. Compress data before transfer

Security services provided by SSH

SSH provides several security services. Some of them are as follows.

1. Confidentiality via encryption
2. Integrity via hashing
3. Server Authentication
4. Client User Authentication

4.5.2 SSH Protocol

The overall SSH protocol consists of three protocols. Each of these protocols provides respective services.

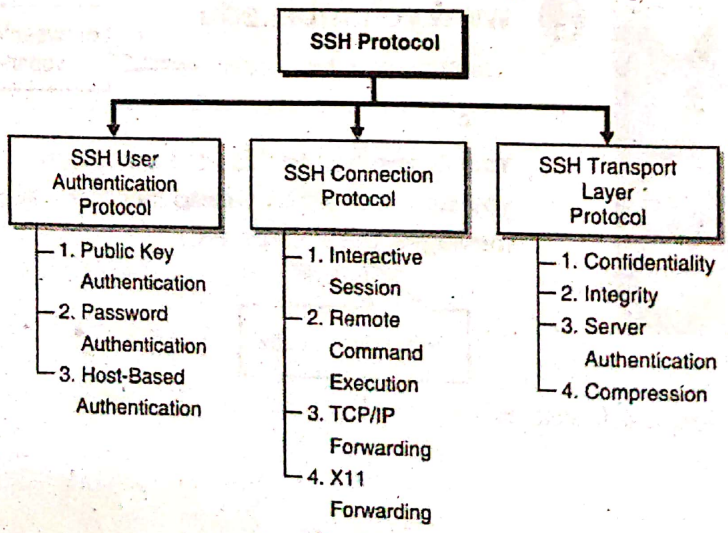


Fig. 4.5.1

1. SSH Authentication Protocol

Definition : *SSH Authentication Protocol performs client user authentication.*

Whenever an SSH client tries to establish a connection with the SSH server, the server requires that the client authenticates with the SSH user that must be present on the SSH server. A client cannot establish an SSH connection without an SSH user already available on the SSH server. The authentication protocol ensures that the user can successfully authenticate with the SSH server. The protocol provides three mechanisms for carrying out the user authentication.

1. **Public Key Authentication :** The client signs the message (using client's private key) containing the client's public key and sends it to the server. The server validates the signature and the underlying public key received and allows access if the signature is correct.
2. **Password Authentication :** The client sends a password to the server for authentication. The server validates the password and allows access if the provided password is correct.
3. **Host-Based Authentication :** In this authentication mechanism, the authentication process is carried out by the host (machine) where the client is installed. The server validates the host instead of the client.

2. SSH Connection Protocol

Definition : *The SSH Connection Protocol deals with managing SSH connections between the client and the server.*

It provides interactive login sessions, remote execution of commands, forwarding TCP/IP connections, and forwarding X11 connections.

1. **Interactive session :** It opens a working session. In a session, you could execute commands, or carry out administrative tasks on the remote machines.
2. **Remote execution of commands :** Once the session has been set up, any program (shell, file transfer, email, document editor, etc.) could be started at the remote machine. You could then execute various commands or carry out activities as desired.
3. **Forwarding TCP/IP connections :** Forwarding provides the ability to carry out any insecure TCP connection over a secure SSH connection. The insecure TCP connection passes through the secure SSH tunnel and hence the insecure connection automatically gets the security from the established SSH tunnel. This is also referred to as SSH tunneling.
4. **Forwarding X11 connections :** X11 forwarding allows you to run any GUI application on a remote machine.

3. SSH Transport Layer Protocol

Definition : *The SSH transport layer is a secure, low level transport protocol. It provides strong encryption, cryptographic host authentication, and integrity protection.*

The Transport Layer Protocol provides server authentication, confidentiality, and integrity. It may optionally also provide compression. The transport layer typically runs over a TCP/IP connection, but might also be used on top of any other reliable data stream.

- Unlike the traditional Public Key Cryptography system that depends upon a CA to establish trust amongst the users, the earlier implementations of PGP did not use the regular CAs for issuing certificates. It used "Web of Trust" where each user generates and distributes his or her public key, and users sign each other's public keys, which creates a community of users who trust each other. This is different from the CA approach, where no one trusts each other; they only trust the CA.
- So, basically, PGP is a system of "I don't know you, but my friend Alice says that you can be trusted, so I will trust you on her words". In the figure, as you understand, there is a trust relationship (User 1, User 2) and (User 2, User 3). Now, when User 3 needs to communicate with User 1, it establishes a trust inherited from its prior trust on User 2.
- There is no third-party involved in this scenario. Each user keeps in a file, referred to as a key ring, a collection of public keys he has received from other users. Each key in that ring has a parameter that indicates the level of trust assigned to that user and the validity of that particular key.

4.6.1(B) PGP Services

PGP provides the following services. You can use one or more services at a time. For example, if you intend to use only encryption service, you can do so. If you intend to use only the digital signature service, you can do so. Let's learn a brief about these services.

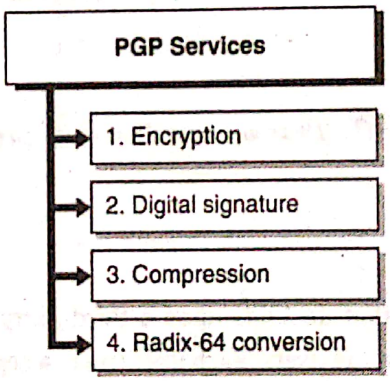


Fig. 4.6.3

1. Encryption

a. The sender creates a message.
b. PGP generates a random number that is used as symmetric key to encrypt it.
c. The symmetric key is encrypted using receiver's public key.
d. Encrypted message and the encrypted symmetric key are sent to the receiver.
e. The receiver decrypts the encrypted symmetric key using her private key.
f. Once the receiver gets the symmetric key after decryption, the key can be used to decrypt the message.

Fig. 4.6.4 is a simplistic diagram of encryption steps followed by PGP.

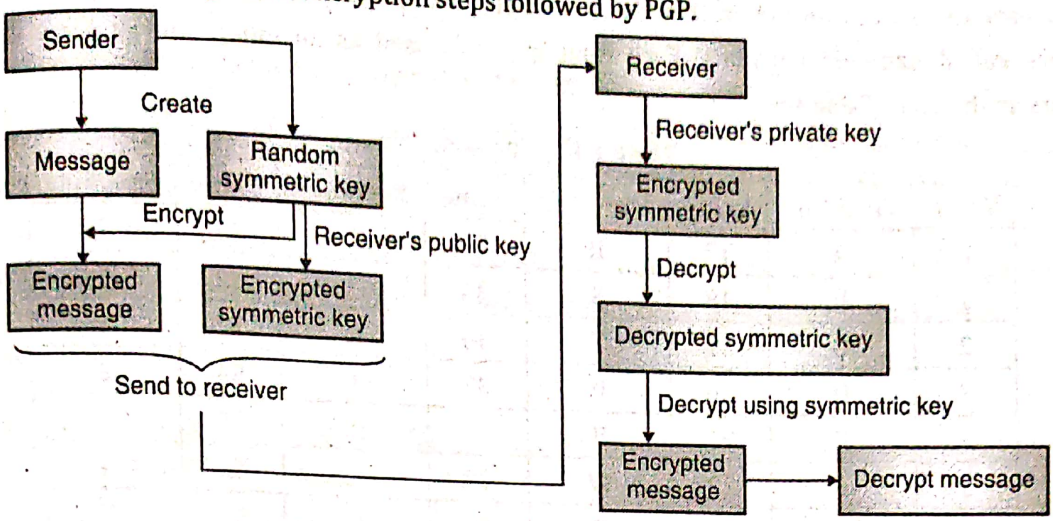


Fig. 4.6.4

2. **Digital Signature**

The digital signature uses a hash code or message digest algorithm, and a public-key signature algorithm. You have already learnt digital signature in detail in Unit 2. Refer it for a quick refresher.

3. **Compression**

PGP compresses the message after applying the signature but before encryption. Compression has the added side effect that some types of attacks can be avoided by the fact that even the slightly altered, compressed data does not decompress without errors. This side security benefit is operationally useful.

4. **Radix-64 conversion**

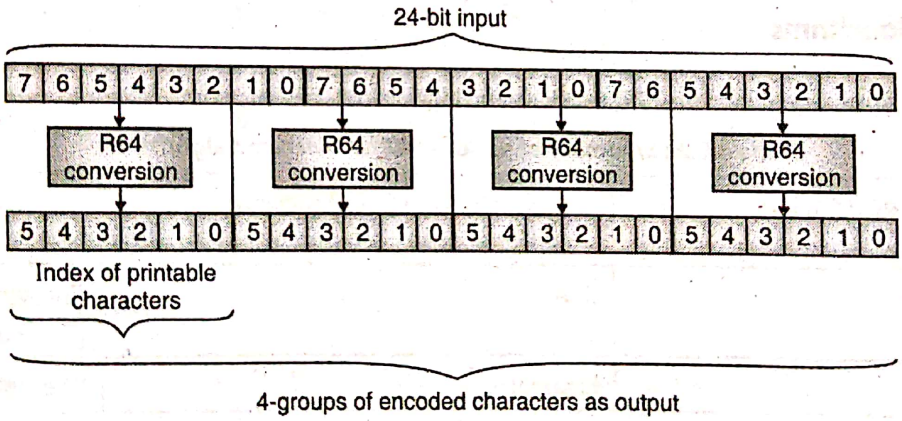


Fig. 4.6.5

R64 conversion is useful for compatibility of emails across varied systems. PGP's underlying native representation for encrypted messages, signature certificates, and keys is a stream of arbitrary octets. Some systems only permit the use of blocks consisting of seven-bit, printable text. So, for transporting PGP's native raw binary octets through channels that are not safe to raw binary data, a printable encoding of these binary octets is needed.

PGP provides the service of converting the raw 8-bit binary octet stream to a stream of printable ASCII characters, called Radix-64 encoding. Each 6-bit group is used as an index into an array of 64 printable characters as shown in Table 4.6.1.

Table 4.6.1 : Encoding map

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	Z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
13	N	30	e	47	v	(pad)	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

4.6.1(C) PGP Algorithms

Table 4.6.2 is a summary of various PGP services and algorithms they support.

Table 4.6.2 : Summary of various PGP services and algorithms

Sr. No.	PGP Service	Supported Algorithm	Purpose
1.	Public Key	RSA	Encrypt or Sign (Symmetric Key)
2.	Public Key	Elgamal	Encrypt (Symmetric Key)
3.	DSA (Digital Signature Algorithm)	DSS	Sign (Message)
4.	Symmetric Key	IDEA, TripleDES, CAST5, Blowfish, AES	Bulk encryption (message)
5.	Hash	MD5, RIPEMD160, SHA1, SHA256, SHA384, SHA224, SHA512	Hashing
6.	Compression	ZIP, ZLIB, BZip2	Compress messages

4.6.2 MIME

- MIME is an acronym that stands for Multipurpose Internet Mail Extensions.

Definition : *MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet e-mail protocol that lets users to use the protocol to exchange different kinds of data files (audio, video, images, application programs, and others) via email.*

- Today, you can, thus, use email for attaching various kind of files and send it across. MIME does not provide security specifications for sending and receiving emails securely. Hence, S/MIME protocol is used.

4.6.3 S/MIME

Definition : *S/MIME (Secure/Multipurpose Internet Mail Extensions) provides a consistent way to send and receive secure MIME data (emails).*

It is based on certificates (Public Key Cryptography) and works as you have learnt in previous sections and units.

4.6.3(A) S/MIME Services

- S/MIME provides the following cryptographic security services for electronic messaging applications as shown in Fig. 4.6.6.

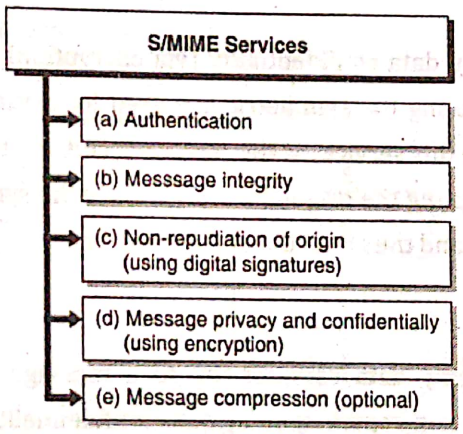


Fig. 4.6.6 : S/MIME Services

- These services are provided using same techniques as you have learnt in the previous sections and units.

4.6.3(B) S/MIME Algorithms

Table 4.6.3 is a summary of various S/MIME services and algorithms they support.

Table 4.6.3 : S/MIME supported services and algorithms

Sr. No.	S/MIME Service	Supported Algorithms	Purpose
1.	Message Integrity	SHA-256, SHA-1, MD5	Hashing
2.	Non-repudiation, Authentication	RSA and DSA with Hashing algorithms	Digital Signature
3.	Key Encryption	RSA, RSAES-OAEP, Diffie-Hellman	Encrypting Symmetric key
4.	Privacy and Confidentiality	AES, DES, Triple DES	Message Encryption

4.6.3(C) S/MIME Cryptographic Message Syntax (CMS)

- S/MIME standard describes a protocol for adding cryptographic signature and encryption services to MIME (email) data. The MIME standard provides a general structure for the content of Internet messages and allows extensions for new content-type-based applications.
- The S/MIME specification defines how to create a MIME (email) body part that has been cryptographically enhanced according to the Cryptographic Message Syntax (CMS).
- There are 4 types of CMS used in S/MIME:

1. Data Content Type

This is the original plaintext form of the email message. It has an identifier that is referred whenever it is compressed, encrypted or digitally signed.

2. SignedData Content Type

SignedData content type is used when a sender needs to apply a digital signature to a message. Applying a signature to a message provides authentication, message integrity, and non-repudiation of origin.

3. EnvelopedData Content Type

This content type is used to apply data confidentiality (via encryption) to a message. A sender needs to have access to a public key (for encrypting the symmetric key used for actual encryption of the message) for each intended message recipient to use this service. At the receiver's end, the receiver uses her private key to decrypt the symmetric key used for encrypting the original message. Once the symmetric key is available, it can be used to decrypt the encrypted message and thus read the email message.

4. CompressedData Content Type

This content type is used to apply data compression to a message. This content type does not provide authentication, message integrity, non-repudiation, or data confidentiality. It is only used to reduce the size of the message.

4.6.3(D) Comparison between PGP and S/MIME

Sr. No.	Comparison Attribute	PGP	S/MIME
1.	Trust established using	Web of Trust	Public Key Infrastructure
2.	Provides Authentication	No	Yes
3.	Used for	Securing text messages only	Securing Messages and attachments
4.	Industry use	Less Common	Widely used
5.	Administrative overhead	High	Low
6.	Cost	High	Low
7.	Convenience	Low	High

4.7 VPN (Transport Level Security)

- Organisations setup internal and private network for use by its authorised users. Its resources are not accessible from public network such as the Internet. Increasingly the task force is becoming global and remote. Physical presence to access the organisation resources is no more efficient. At the same time, the organisation cannot risk exposing its internal resources over the public network.

A Virtual Private Network (VPN) provides a solution for this scenario. It allows to establish a secure channel between the communicating parties over the public network, such as the Internet and, facilitate secure connection between them.

- A Virtual Private Network (VPN) is a data network that enables two or more parties to communicate securely across a public network by creating a private connection, or "tunnel," between them.
- The authorised users can then securely access the private network over the public network. Physical presence within organisation premises is not required to access the private network. The entire traffic between the remote user and the private network is encrypted. IPSec can be one of the mechanisms for establishing a VPN connection.

4.7.1 Types of VPN

At a broad level, there are two types of VPN.

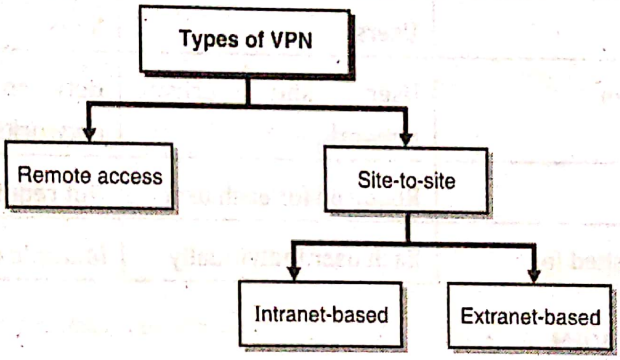


Fig. 4.7.1

1. Remote Access VPN

Remote Access VPN is setup for remote users. They can access the private network securely over the public network.

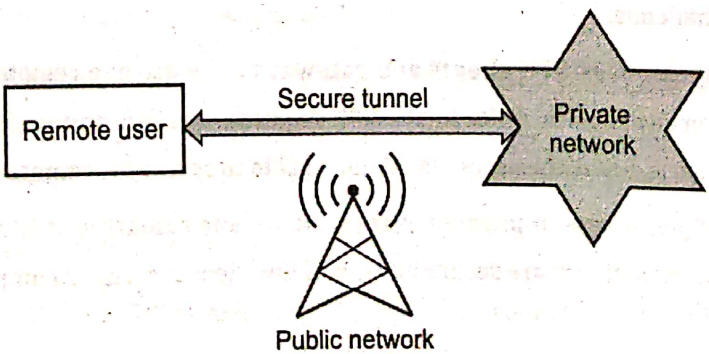


Fig. 4.7.2

2. Site-to-Site VPN

This is established by the organisation for connecting its multiple sites or branch offices so that the users can access the resources across the sites.

a. **Intranet-based site-to-site VPN** is used for organization's own sites.

b. **Extranet-based site-to-site VPN** is used for organization and its partners.

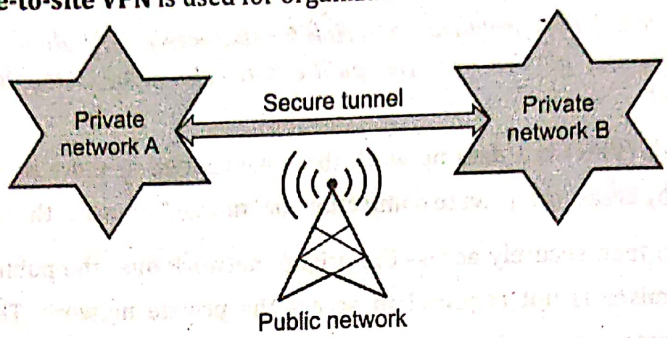


Fig. 4.7.3

Comparison between Remote Access and Site-to-Site VPN

Sr. No.	Comparison Attribute	Remote Access VPN	Site-to-Site VPN
1.	Setup For	Users	Sites
2.	Tunnel between	User and private network	Between two or more private networks
3.	VPN client	Required for each user	Not required for each user
4.	Tunnel established for	Each user individually	Multiple users use the same tunnel

4.7.2 Challenges of using VPN

VPN technologies have the following challenges.

1. **Interoperability :** There are various types of clients these days – Laptop, Desktop, Tablets, Mobile Phones, each running a variety of OS and applications. Ensuring that the VPN technology is compatible with all the possible client types if a major challenge.
2. **Installation and management of VPN clients and gateways :** The use of a remote access VPN requires that a VPN client be installed on each device. A site-to-site VPN requires installing and managing several VPN gateways. If any of the VPN gateways is down, the users would not be able to access the remote resources.
3. **Client security :** VPN allows access to private network and remote resources. It is important to ensure that the clients accessing the private network are secure enough. If the client security is compromised, it might infect the private network as well.
4. **Requires strong authentication :** Since VPNs provide access to the private network, the user must be strongly authenticated to ensure that she is a legitimate user. You must use two-factor authentication to ensure that the user is authorized to use the private network.

OTP

1. **Confidentiality :** Confidentiality is provided via the negotiated encryption algorithms.
2. **Integrity :** Data integrity is protected by including with each packet a MAC that is computed from a shared secret, packet sequence number, and the contents of the packet.
3. **Server authentication :** The SSH server host key is used during key exchange [when client tries to connect to the server] to authenticate the identity of the host.
4. **Compression :** If compression is required, the 'payload' field is compressed using the negotiated algorithm. The 'packet_length' field and the 'mac' is computed from the compressed payload. Encryption is done after compression.

Service	Algorithm
Confidentiality (Encryption)	• 3des-cbc • blowfish-cbc • twofish256-cbc • twofish-cbc • twofish192-cbc • twofish128-cbc • aes256-cbc • aes192-cbc • aes128-cbc • serpent256-cbc • serpent192-cbc • serpent128-cbc • arcfour • idea-cbc • cast128-cbc • none (no encryption)
Integrity (Hashing)	• hmac-sha1 • hmac-sha1-96 • hmac-md5 • hmac-md5-96 • none (no hashing)
Compression	• zlib • none (no compression)

The various algorithms supported at the SSH transport layer are as follows.

4.5.3 Establishing SSH connection

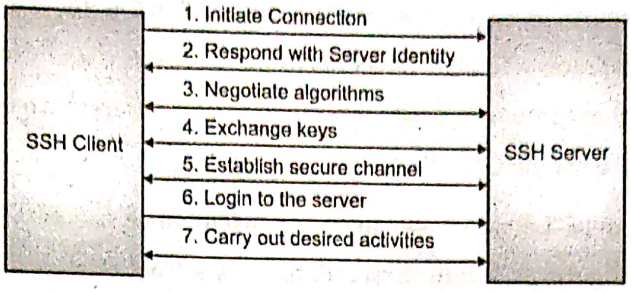


Fig. 4.5.2

This is a highly simplistic SSH connection diagram. The client initiates the connection and a secure channel is established after verifying server identity, negotiating algorithm parameters and exchanging keys. Once the channel is established, the client provides the user authentication after which the remote activities can begin.

4.6 Email Security

- Around billions of emails are sent across the globe every day. Emails have become the primary source of official communication. With such a wide use of emails, attackers are inclined and motivated to intercept emails and get the message and at times modifying the messages before the recipient gets it.
- It is important that you secure the email communication as any other form of communication. In this section, you will learn about a couple of email security standards that you could use.

4.6.1 Pretty Good Privacy (PGP)

✍ ***Definition :*** *Pretty Good Privacy (PGP) is an email security program that was developed in 1991. It is based on public key cryptography.*

4.6.1(A) Web of Trust

- In Public Key Cryptography system that depends upon a third-party Certificate Authorities (CAs) to establish trust, there is no mutual trust amongst the users. Each user trusts a reputed CA and thus CA plays a predominant role in establishing the trust so that communication can happen between users. If there is no CA, the trust relationship is not established and thus the communication may not happen.

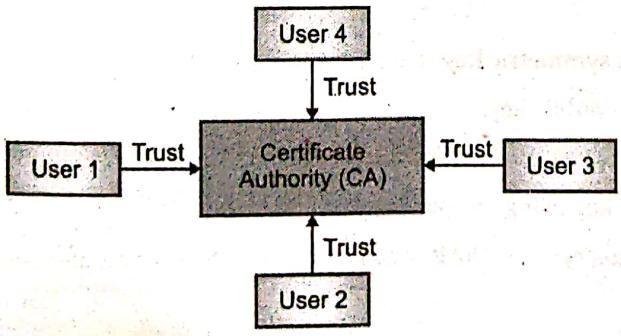


Fig. 4.6.1

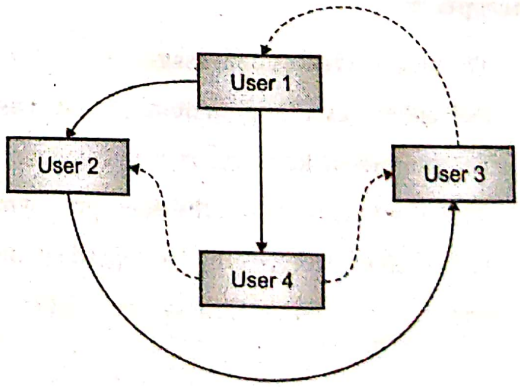


Fig. 4.6.2