KEY POINTS
(MODULE-1)

NETWORK :-

→ A Network is two (or) more joined together via a Switch, Communicating via a Routing protocol.

(or)

→ One System is talk to other System to Exchange the Some Information is Called Network

→ The biggest World Network is INTERNET

Data Network (or) Computer network :-

→ Computer Network is Called Data Network.

→ The Inter Connection b/w Computer and other devices

Cyber Security :-

Cyber Security is the protection of Internet Connection Systems, Including Hardware, Software and data from Cyber Attacks.

Cryptography :-

→ The word Cryptography was derived from Combining 2 greek words - "Krypto" it means "hidden" and "graphee" means "writting"

→ Cryptography is the art of Secrete Information - hiding (or) Secrete data hiding.

→ The main goal of Cryptography is data Secure from Unauthorized person (or) hackers.

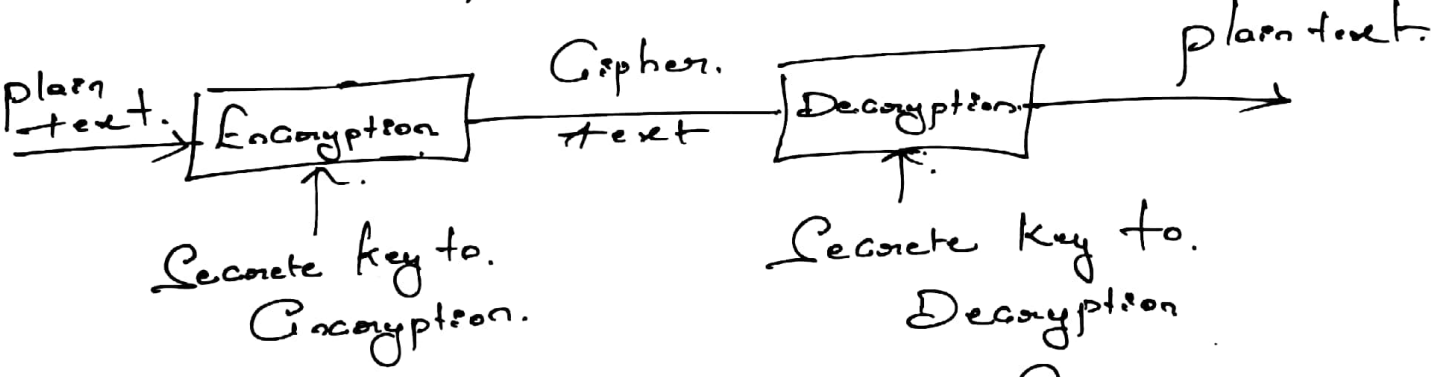


Fig :- Cryptographic flow.

→ Encryption is a Technique for Transforming plaintext into an Unreadable Cipher text.

→ Decryption is a Technique for Transforming Cipher text into to plaintext (or) original data.

→ The key is also a group of bits which as a Major Role in the process of Encryption and Decryption.

→ Types of Cryptography.

* Symmetric key cryptography :- is. Called Secrete key Cryptography (or) private. key Cryptography.
It Uses a Same key (or) Single key for both of Encryption and Decryption method.

* Asymmetric key Cryptography :-
Where a different key used for Encryption. and decryption.

* Hash function :-
Uses no key for Encryption and decryptio

→ Application.
* online banking
* online Transaction
* media Application. E.t.c.

Steganography :

→ The steganography comes from Greek words.

→ Stegano is Greek meaning "hidden." (or) "Covered" and graphic is Greek meaning "written".

→ Steganography is the Invisible Communication.

→ The main Idea of Steganography is to Hide Secret messages in the other. Cover digital media such as text, video and Audio, Image etc.

S.T. Someone (or) hacker (or) other person cannot know the presence of the Secret Info.

→ There are Three basic types of Steganography

1) pure Steganography
2) Secret key Steganography
3) public key steganography.

Dual Steganography :

→ is the process of using Steganography Combined with Cryptography.

→ Dual Steganography is the process of hiding Confidential data's in the media files. Such as Audio, Images, video. etc //

Web Security :

→ Web Security means providing the security for the data which is transmitted to the network.

Ex: Client and Server.

→ The client will send the Request to the Server. The Server will provide to the Server to the client so for. This purpose we will use a one protocol i.e called SSL protocol.

SSL :

→ i.e means Secure Socket layer by Implementing This SSL we can provide the security for the data which is transferred b/w the web browser and Server.

→ This SSL is Implemented by using a different protocols. (or) It includes a different protocol

1) SSL Record protocol.
2) Hand Shake protocol.
3) Change Cipher Spec protocols
4) Alert protocol.

→ So all this protocols will be included SSL protocol.

→ So. This we will call it as SSL protocol Stack.

→ SSL will be Implemented at just above the TCP/IP and just below the HTTP.

→ SSL 2 main Concepts. → Connection.
→ Session.

Connection :- → Transport to provide the Service b/w Client and Server..

[Eg :- First Connection should be established. b/w the Client and Server S.T client will Communicate with Server and Server will Communicate with the Client]

Session :- → Association b/w a Client and Server.

[based upon the Session, the client and Server will be Communicated]

→ The Session will call it as a Temporary Time period.

→ The Session Consists of a multiple Connection

→ The Session Created by Handshake protocol //

Session State parameters :-

1) Session Identifier :- An arbitrary byte sequence chosen by the server to identify an active (or) resumable session state.

2) peer Certificate :-
   → peer is nothing but a client
   → An X509.v3 Certificate of the peer.

3) Compression method :- The algorithm used to compress data prior to Encryption.

4) Cipher Spec :-
   → means Specification.
   → Cipher Specifies the bulk data Encryption Algorithm (such as null, DES, etc) and a hash algorithm (such as MD5 (or) SHA-1) used for MAC Calculation.

   Note :-
   Bulk → means big
   MAC → message Authentication Code.

5) master Secrete :-
   → 48 byte Secrete shared b/w the client & server (which is the Secrete key shared b/w client and server)

master Secrete.

6) Is Resumable :-

A flag indicating whether the session can be used to initiate new connections.

Connection State parameters :-

1) Server and Client Random :

Byte sequences that are chosen by the server and client for each connection.

2) Server Write MAC Secret :

The secrete key used in MAC operations on data sent by the server.

3) Client Write MAC Secret :

The secrete key used in MAC operations on data sent by the client.

4) Server Write key :

The conventional encryption key for data encrypted by the server and decrypted by the client

5) Client Write key :-

The conventional encryption key for data encrypted by the client and decrypted by the server.

6) Initialization vector : when a block cipher in CBC mode is used, an initialization vector is maintained for each key.

7) Sequence numbers : Each party maintains separate sequence numbers for transmitted and received msg for each connection

Secure Socket Layer. (SSL)

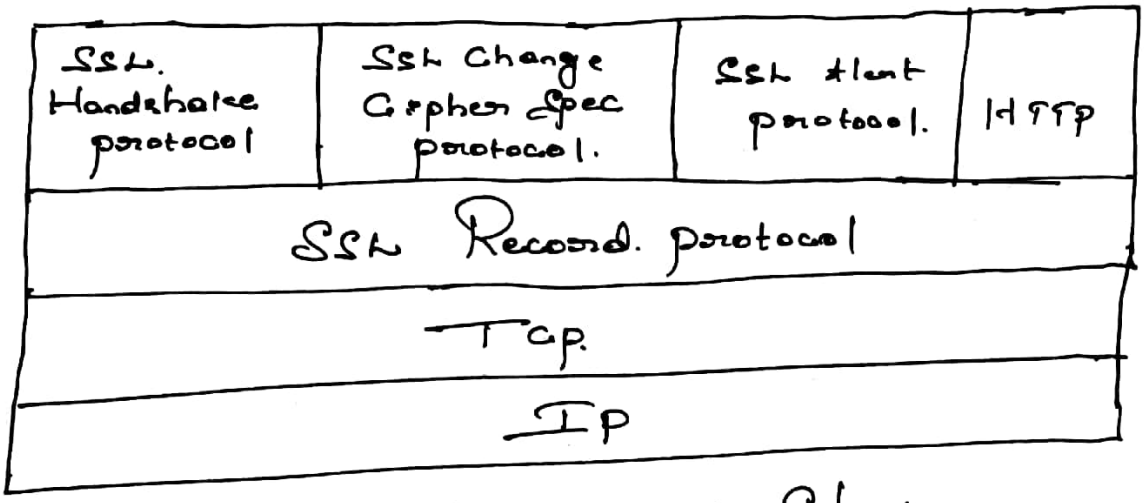


Fig:- SSL protocol Stack.

SSL Record. protocol. :-

→ The data fragmented and the Calculation of MAC (or) the Encryption Algorithm (or) the Compression Algorithm will be Implemented in the SSL. Record protocol.

Hand Shake protocol :- is used to established the Session and this Handshake protocol is used. Authenticate the client with the Server and Server with the client.

Change Cipher-Spec. protocol :- is for Changing the State i.e means the pending State to the Current State.

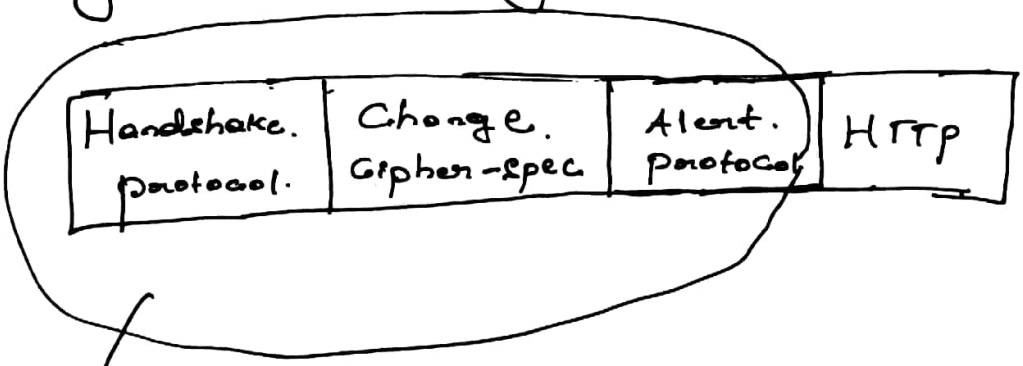
NOTE:

MAC → media Access Control address

→ 48-bit hexadecimal address.

→ For example. 70-54-D2-AB-EF-85

Alert protocol :- This Alert protocol for giving the Alerts by Implementing the SSL.



So all this protocols involved in the SSL.

SSL Record protocol :-

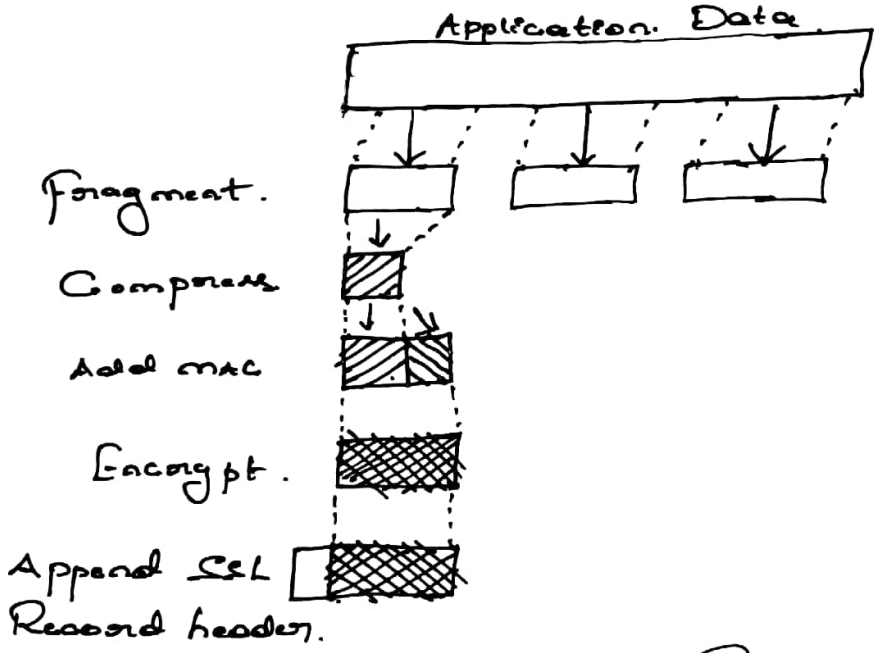


Fig :- SSL Record protocol operation.

Step 1 :-

⇒ The Application Data will be divided into Different fragments. i.e means fragmentation will be done. According to the bandwidth.

Step 2 :-

⇒ Consider the 1st fragment and now apply the Compression function.

(i.e fragment will be Compression format)

(compression — optional frag)

Step 3 :-

→ Next Calculate the MAC by using one of the Authentication Algorithms.

Ex:- Ice Con. use. MD5 Algorithm (or) SHA-1 Algorithm.

Step 4 :-

→ The MAC will be Calculated, So. That MAC will be appended to the Compression function.

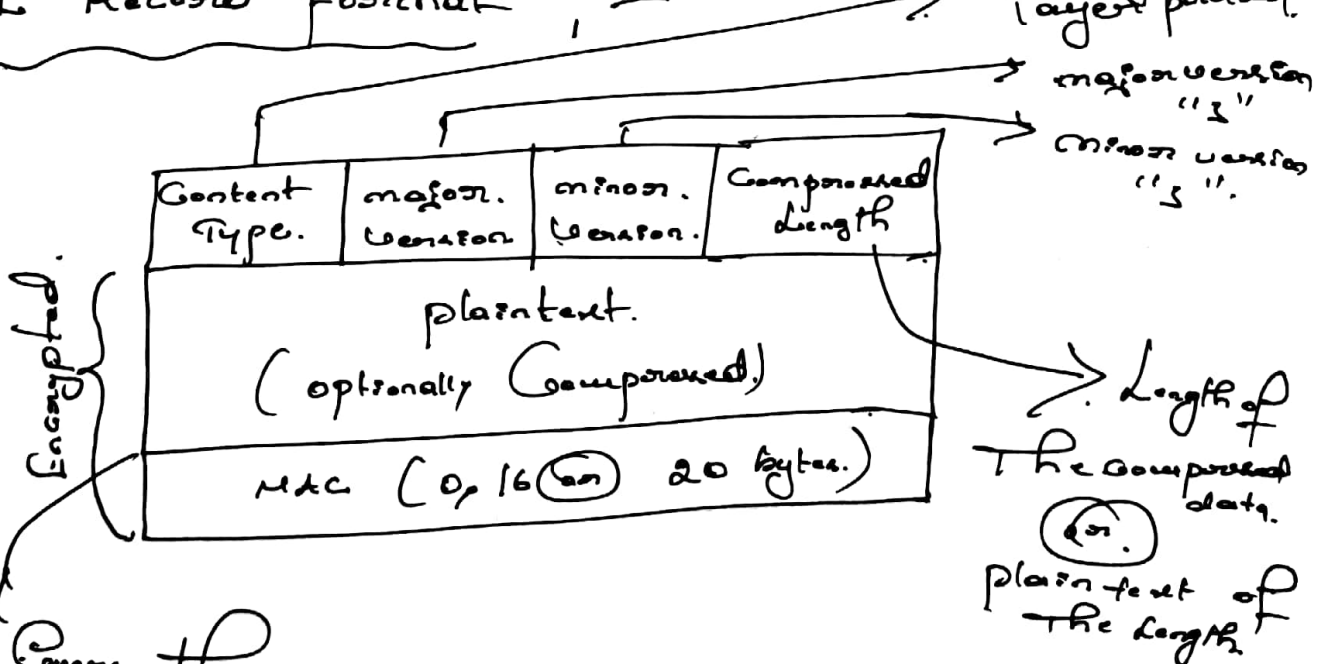
Step 5 :-

→ Next we Encrypt the MAC and Compressed data, So. Encryption done on the fragment.

Step 6 :-

→ After Encryption done, now append the SSL Record header to this fragment i.e means. In order to processing to the higher level protocol.

SSL Record Header (or) SSL Record format :-



There are mainly 4 fields. Involved in the SSL Record Header.

1) Content type. (8bits)
   -> The Higher layer protocol used to process the Enclosed fragment.

2) major Version (8bits)
   -> Indicates major version of SSL in use. For SSLv3 the value is 3

3) Minor Version (8bits)
   -> Indicates minor version of SSL in use. For SSLv3 the value is "0"

4) Compressed Length (16bits).
   -> The Length in bytes of the plaintext fragment. (or Compressed fragment if Compression is used)

Now Let us See the Equation by Calculating the MAC.

hash (MAC-write-Secret || pad-2 ||
hash (MAC-write-Secret || pad-1 || Seq-num ||
SSL Compressed Type || SSL Compressed Length ||
SSL Compressed fragment))

Where.

|| → Concatenation.

MAC_write_Secret → Shared Secret key
↳ This is the Shared Secret key b/w the Client and Server.

hash → Cryptographic hash Algorithm ; either MD5 (or) SHA-1.

pad_1. → The byte 0X36 (0011 0110) Repeated 48 times for MD5 and 40 times (320 bits) for SHA-1.

↳ The pad-1 means 0011 0110 Repeated 48 times for MD5 and 40 times for SHA-1

pad_2 → The byte 0X5C (0101 1100) Repeated 48 times for MD5 and 40 times for SHA-1

↳ If you are applying the MD5 algorithm. The byte Sequence (0101 1100) will be Repeated 48 times S.T Hash to generate the MAC.

↳ Ily the SHA-1 Algorithm the byte Sequence (0101 1100) will be Repeated 40 times. S.T Hash to generate the SHA-1

Seq_num → The Sequence number for this message.

SSL Compressed.type → The higher-level protocol used to process this fragment.

SSL Compressed.length → The length of the Compressed fragment.

SSL Compressed.fragment → The Compressed fragment. (if Compression is not used, This is the plaintext fragment).

Handshake protocol :

→ The main purpose of this Handshake protocol is to established the Session

→ Handshake protocol, means the Session will be generated by this Handshake protocol.

→ Handshake protocol can be Represented by three fields.

first → Type

second → Length.

3rd → Content

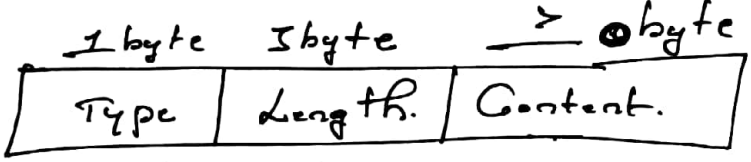


fig:- Handshake protocol.

Where. Type. → Represents the higher layer protocol

Length → Represents the Length of message

Content → means parameters associated with the particular msgs.

→ The Different messages Involved in the Handshake protocol

1) Client Hello.
2) Server Hello.
3) Certificate as X.509
4) Server key exchang
5) Certificate Request
6) Server done
7) Client_Key_Exchange.
8) Certificate_Verify
9) Finished.

So this are the different messages Involved in the Handshake protocol

HANDSHAKE PROTOCOL : XXX IM. OR 8 OR 10M.

→ The main purpose of this Handshake protocol is to Established the Session.

OR

The Session will be generated by this Handshake protocol

→ Handshake protocol is used to Authenticate the Client with Server and Server with Client.

→ The Handshake protocol is used before any Application data is Transmitted.

→ The Handshake protocol Consists of a Series of messages Exchanged by Client and Server. (as shown in below Handshake protocol Action - figure 2).

→ Handshake protocol Can be Represented by three fields

First → Type
Second → Length.
3rd → Content

1 byte.	3 byte.	≥ 0 byte.
Type	Length.	Content

Fig:- Handshake protocol

TYPE (1 byte) :- Represents the Higher Layer protocol

OR

Indicates one of 10 messages of Handshake protocol. (as shown in below Table)

LENGTH (3 bytes) :- Represents the length of message

OR

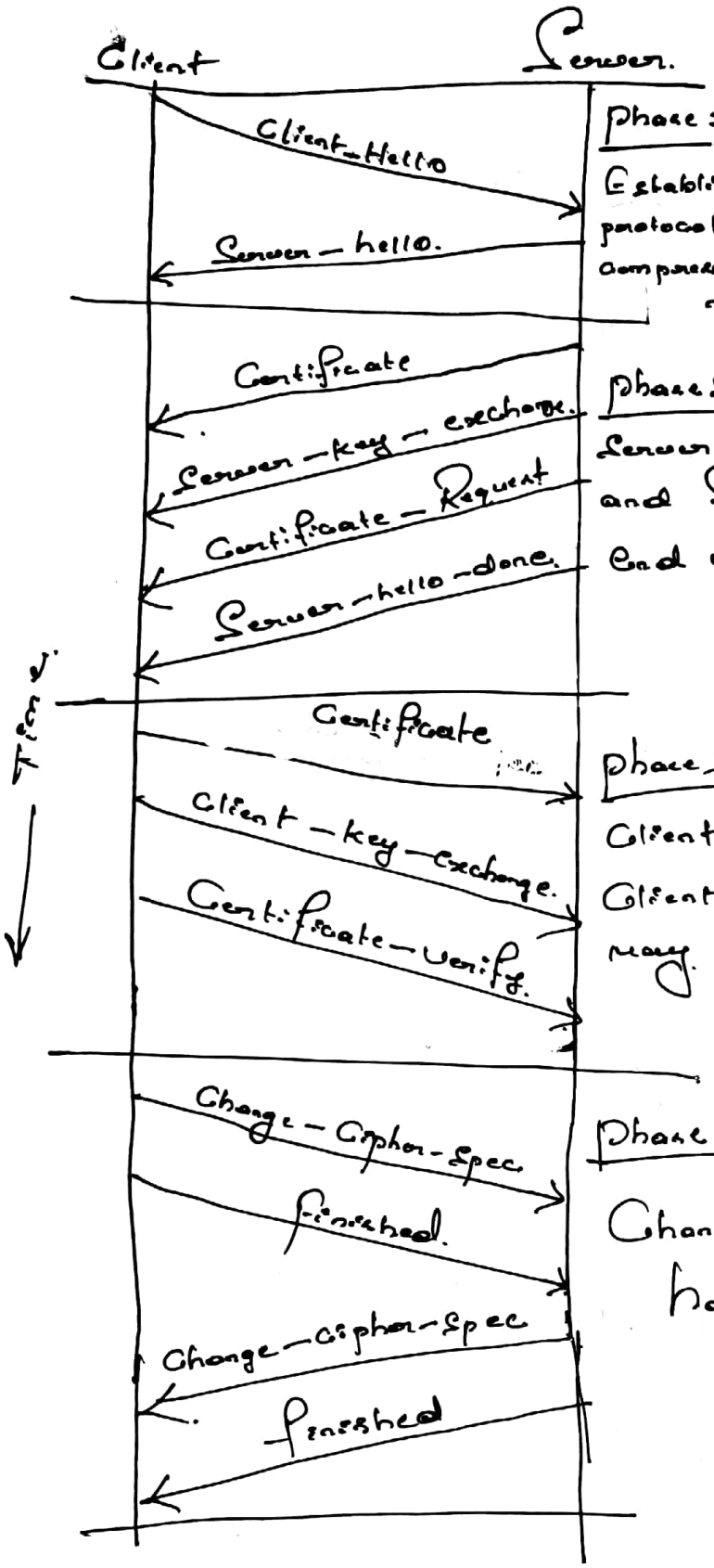
The length of the message in bytes.

CONTENT (byTes) :- Means parameters Associated with this message.

→ The Different Messages Involved in The Handshake Protocol.

Message Type.	Parameters
1) Hello — request.	null
2) Client — hello	version, Random, Session id, Cipher Suite. Compression method.
3) Server — hello	version, Random, Session id, Cipher Suite Compression method
4) Certificate	Chain of X.509V3 Certificates.
5) Server — key — Exchange	parameter, Signature
6) Certificate — Request	Type, Authorities.
7) Server — done.	null
8) Certificate — verify	Signature.
9) Client — key — Exchange.	Parameters, Signature.
10) Finished.	hash value.

So this are the different Messages Involved in the Handshake protocol



Phase 1 :-
Established Security Capabilities. Including protocol version, Session ID, Cipher Suite, compression method, and initial Random numbers.

Phase 2. :-
Server may send Certificate, key exchange, and Request Certificate. Server signals end of hello message phase.

Phase 3 :-
Client sends Certificate if requested. Client sends key exchange. Client may send Certificate verification.

Phase 4 :-
Change Cipher Suite and finish handshake protocol.

Fig :- Handshake protocol Action

CHANGE CIPHER SPEC PROTOCOL :-

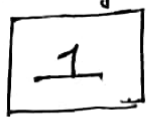


Fig:- Change Cipher Spec protocol.

* The Change Cipher Spec protocol is one of the three SSL-Specific protocols that use the SSL Record protocol and it is the simplest
* This protocol consists of a single message (as shown in above fig) which consists of a single byte (1 byte) with the value 1.
* The sole (one only) purpose of this message is to cause the pending state to be copied into current state, which updates the Cipher Suite to be used on this Connection.

NOTE :-

CipherSuite is basically a Complete Set of methods. (or) is a Set of Cryptographic Algorithms (or) it's a Complete Set of Instruction. Needed to Secure a network Connection through SSL

ALERT PROTOCOL : 3 (or) 4 M.

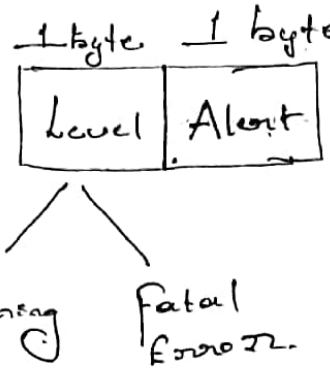


Fig:- Alert protocol

* The Alert protocol used to convey SSL Related Alerts to peer entity
* Each message in this protocol consists of two byte.
* The first byte takes the value Warning (1) (or) fatal (2) to convey the severity of msg.

* Level Represents either Warning (or) fatal error
* If it's a Warning there is no Impact on our Connection.
* If it's a fatal error Automatically the Connection b/w The client and Server will be disconnected, So we have to Re-establish the Connection.

(or)

* If the level is fatal, SSL Immediately terminates the Connection
* other Connection on the Same Session may Continue. but no New Connections Established.
* The Second byte Contains a Code that Indicates the Specific Alert.
* We list those Alerts that are always fatal

1) Unexpected Messages

An inappropriate message was Received.

(or)

Receiving the inappropriate msg.

2) Bad_Record_MAC :- An Incorrect MAC Was Received.

3) Decompression_failure :- The decompression function Received Improper Input.
(Or. Decompression is failure in the Receiver Side)

4) HandShake_failure :- HandShake failure means - Authentication failure.
So. Handshake Mainly for Authenticated to Client to Server (or) Server to Client. If Any one Authentication is failed. means Handshake failure

5) Illegal_parameter :-
Field in a Handshake message was out of Range (or) Inconsistent with other fields.

* The Remaining Alerts are the following

1) Close_notify :- Notifies the Recipient that the Sender will not Send any more Message on this Connection

2) No_Certificate :- may be Sent in Response to a Certificate Request if no Appropriate Certificate is Available

3) Bad — Certificate :- A Received Certificate has Corrupt.

4) Unsupported — Certificate :- The Type of the Received Certificate is not Supported.

5) Certificate — Revoked :- A Certificate has been Revoked by its Signer.

6) Certificate — Expired :- A Certificate has Expired.

7) Certificate — Unknown :- Some other unspecified issue arose in processing the Certificate, Rendering it unacceptable.

TRANSPORT LAYER SECURITY (TLS)

→ TLS is an Internet Engineering Task force (IETF) Standard protocol that provides Authentication, privacy and data Integrity b/w two Communicating Computer Applications

→ It's mostly widely deployed Security protocol in use today and is best Suited for web browsers and other Applications that Require data to be Securely Exchanged over a network.

→ TLS is an IETF Standardization Initiative whose goal is to produce an Internet Standard Version of SSL

→ TLS is defined as a proposed Internet Standard in RFC 5246.

[NOTE:- * TLS 1.2 was defined in RFC 5246

*. The IETF is a Large open International Community of network designers, operators, vendors, and Researchers Concerned with the Evolution of the Internet Architecture and the Smooth operation of the Internet]

→ The RFC 5246 is very Similar to SSLV3.

1) VERSION NUMBER:-

*. The TLS Record format is the Same as that of the SSL Record format and the fields in the Header have the Same Meanings.

*: The one difference is in Version values. For the Current Version of TLS the major Version is 3 and the Minor Version is 3.

2) MESSAGE AUTHENTICATION CODE:-

* There are two differences between the SSLV3 and TLS MAC Schemes.

1) The Actual Algorithm and the Scope of the MAC Calculation.

2) TLS Makes use of the HMAC algorithm defined in RFC 2104

[NOTE

RFC 2104 means : HMAC : Keyed - Hashing for Message Authentication.]

* HMAC is defined as

$$HMAC_K(M) = H[(K^+ \oplus opad) \,||\, H[(K^+ \oplus ipad)||M]]$$

where

H = Embedded Hash function [for TLS, either MD5 or SHA-1]

M = Message I/p to HMAC

K^+ = Secret key padded with zeros on the left so that the result is equal to the block length of the Hash Code (for MD5 and SHA-1, block length = 512 bits)

ipad = 0011 0110 (36 in hexadecimal) repeated 64 times (512 bits)

opad = 0101 1100 (5C in hexadecimal) repeated 64 times (512 bits)

* For. TLS the MAC Calculation Encompasses the fields indicated in the following Expression

MAC (MAC — Write — Secrete, Seq — Num || TLS Compressed. Type || TLS Compressed. Version || TLS Compressed. Length || TLS Compressed. fragment)

* The MAC Calculation. Covers all of the fields Covered by the SSLV3 Calculation, plus the field. TLS Compressed. version which is the version of the protocol being employed.

*** PSEUDORANDOM FUNCTION :- [IOM].

→ TLS makes use of a pseudorandom function. Referred to as. PRF. to Expand Secrets into. blocks of data. for. purpose of key Generation. (or) Validation

NOTE

PRF (pseudorandom function) is mainly based on the data. Expansion function.

→ The objective is to. Make use of a relatively small shared. value but. to generate longer blocks of data in a way that is Secure. from the kinds of attacks made on Hash functions and MACs.

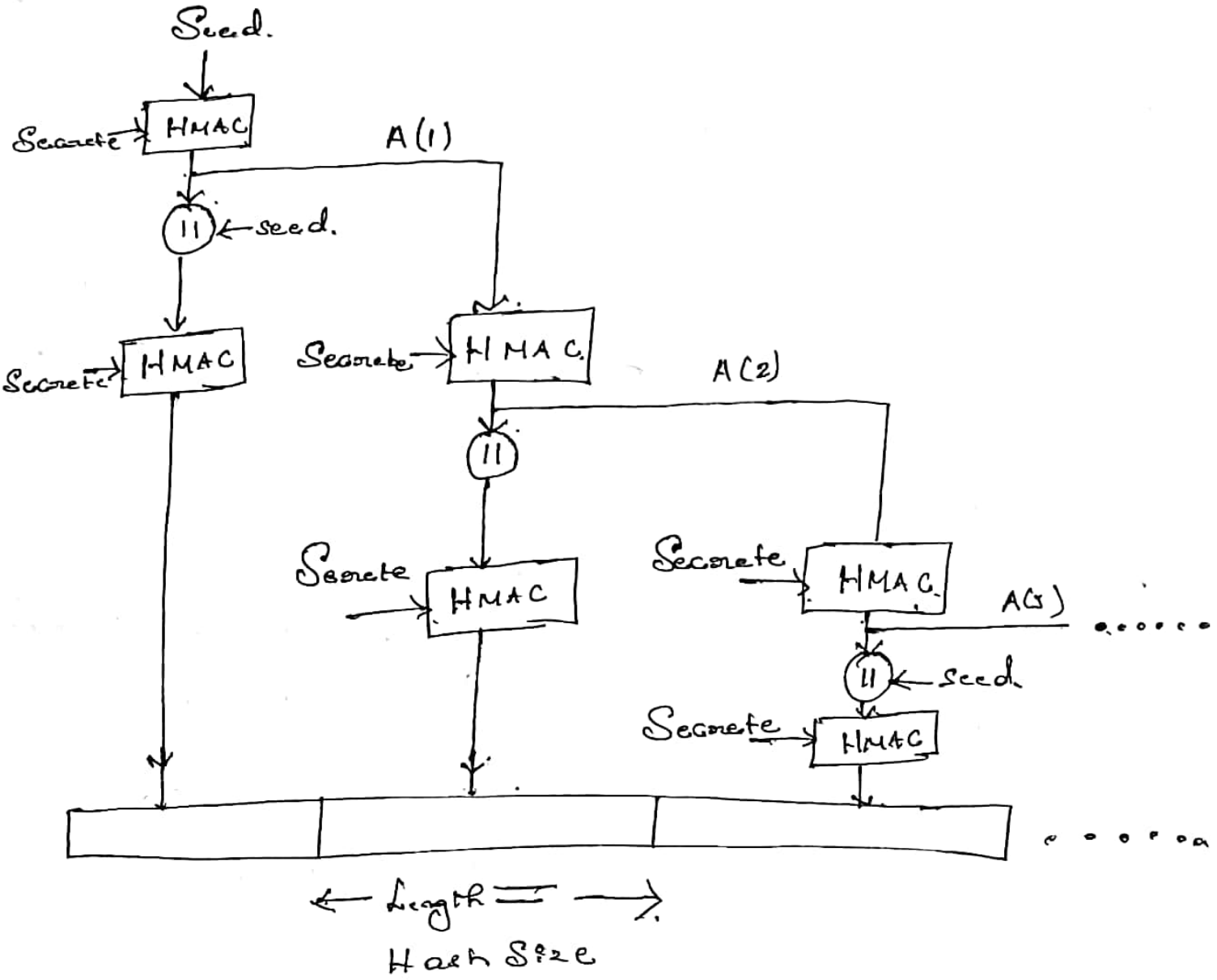


Figure : TLS function. P-hash (secret, Seed)

→ The data expansion function is given as.

P_hash (secret, seed) = HMAC-hash (secrete, A(1) || seed) ||
HMAC-hash (secrete, A(2) || seed) ||
HMAC-hash (secrete, A(3) || seed) || ....

Where A( ) is defined as.

A(0) = seed.

A(i) = HMAC-hash (secrete, A(i-1))

[NOTE :-

Seed (or) Random Seed :- A Random Seed (or seed state (or) just seed) is a Number (or vector) used to Initialize a pseudorandom number generator.]

→ the data Expansion function makes use of the HMAC Algorithm both Either MD5 (or) SHA-1 as the underlying Hash function.

→ As Can be seen (above equation) P-hash Can be iterated as many times as necessary to produce the required quantity of data.

(Repeatedly).

→ For Example :-

* If P-SHA-1 was used to generate 64 bytes of data, it would have to be iterated four times, producing 80 bytes of data of which the last 16 would be discarded.

[NOTE :- If SHA-1 produce 20 bytes → 20 × 4 = 80 bytes
(1st time) (4 times)

We need only 64 bytes of data.

∴ SHA-1 (4 times) = 80 − 16 (discarded bytes (last bytes))

= 64 bytes

* If P_MD5 would also have to be iterated four times, producing exactly 64 bytes of data.

[NOTE:- MD5 produce 16 bytes. = 16 X 4 = 64 bytes
(1st time). (4 times) (Exactly 64 bytes produced)]

→ PRF is defined as.

PRF (secrete, Label, Seed) = P_hash (S1, label || seed)

PRF takes as input a secrete value, an identifying Label, and a seed value, and produces an o/p of arbitrary length.

NOTE (Additional Information)

HTTP :- HTTP (Hyper Text Transfer protocol) is the set of Rules for Transferring files - such as Text, Image, Sound, video and other Multimedia files over the Web. As soon as a User open their web browser they are Indirectly using HTTP. HTTP is an Application protocol that runs on top of the TCP/IP Suite of protocols which forms the foundation of Internet. Through the HTTP protocol, Resources are Exchanged b/w client devices and Servers over the Internet.

NOTE (Additional Info)

[HTTP :- * The Standard protocol for Transferring Hypertext documents on the World Wide Web.

* Communication b/w client Computers and Web Servers is done by Sending HTTP Request and Receiving HTTP Responses.]

HTTPS : *** 10M.

→ Hyperlext Transfer protocol Secure (HTTPS) is an Extension of the HTTP.

[Or]

It is highly Advanced & Secure Version of HTTP.

→ HTTPS (HTTP over SSL) refers to the Combination of HTTP and SSL to implement Secure Communication b/w a Web browser and Web Server.

→ The HTTPS Capability is built into all modern Web browsers. It uses dependent on the Web Server Supporting HTTPS Communication.

→ The principal difference Seen by a User of a Web browser is that URL (Uniform Resource Locator) addresses begin with Https:// rather than Http://

→ A normal HTTP Connection uses port 80.

→ If HTTPS is specified, port 443 is used. Which invokes SSL or TLS.

NOTE
[PORT :- A port is a logical Construct that Identifies a specific process or Type of network Service]

→ When HTTPS is used, the following elements of the Communication are Encrypted.

* URL of the Requested document
* Contents of the documents
* Contents of browser forms (filled in by browser User)
* Cookies Sent from browser to Server and from Server to browser
* Contents of HTTP header.

NOTE
[Cookies :- are text files with small pieces of data like User name and password.
HTTP Cookies are used to Identify Specific users]

→ HTTPS is documented in RFC 2818. HTTP over TLS. There is no fundamental change in using HTTP over either SSL or TLS and both implementations are referred to as HTTPS

Connection Initiation :-

→ Initiation steps :-

STEP 1 :- The client Initiates a Connection to the Server on the appropriate port and then Sends the TLS Client Hello to begin the TLS Handshake.

STEP 2 :- When TLS Handshake has finished the Client may then initiate the first HTTP request.

STEP 3 :- All HTTP data is to be Sent as TLS Application data.

→ There are three level of awareness of a Connection in HTTPS

Level 1 :- At the HTTP level, an HTTP Client requests a Connection to an HTTP Server by Sending a Connection Request to the next lower layer.

Level 2 :- Typically the next lowest layer is TCP. but it also may be TLS/SSL.

Level 3 :- At the Level of TLS, a session is established b/w a TLS client and TLS Server. This Session Can Support one or more Connections at any time. As we have seen a TLS Request to establish a Connection begins with the establishment of a TCP Connection b/w the TCP entity on the client Side and the TCP entity on the Server Side.

Connection Closure :-

An HTTP Client or Server can Indicate the closing of a Connection by including the following line in an HTTP Record : Connection : Close . This Indicates that the Connection will be closed after this record is delivered.

NOTE :

Difference b/w SSH and SSL.

→ SSH is a Cryptographic network protocol for operating network services securely over an Unsecured network

→ SSH Command provides a Secure Encrypted Connection b/w two hosts over an Insecure network. (practically every Unix & Linux System includes the SSH Command)

→ SSH is used for Creating a Secure tunnel to another Computer.

→ It works on the port number 22.

→ Create a Secure remote link to Server.

Continued,

NOTE :-

Difference b/w SSH and SSL

SSL :-

→ It is a networking protocol which gives Secure Transmission in a non Secure network

→ SSL provide the Security for the data which is Transfered b/w the Web browsers and Server

→ Creates a Secure Connection b/w a website and Users.

→ SSL is used for Securely Transfering data b/w two parties.

→ It works on the port number 443

→ Creates a Secure Connection b/w a website and users //

NOTE :-

Tunnel

Tunneling protocol is a Communication protocol that allows for the movements of data from one network to another //

SECURE SHELLS (SSH) 8m (SEP-2020)

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include **remote command line**, **login**, and **remote command execution**, but any **network service** can be secured with SSH.

SSH provides a secure channel over an unsecured network by using a client–server architecture, connecting an SSH client application with an SSH server. The protocol specification distinguishes between **two major versions**, referred to as **SSH-1** and **SSH-2**. The standard TCP port for SSH is 22. SSH is generally used to access **UNIX-like operating systems**, but it can also be used on **Microsoft Windows10**.

SSH was designed as a replacement for **Telnet** (Telnet is an application protocol used on the Internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.) and for **unsecured remote shell protocols**.

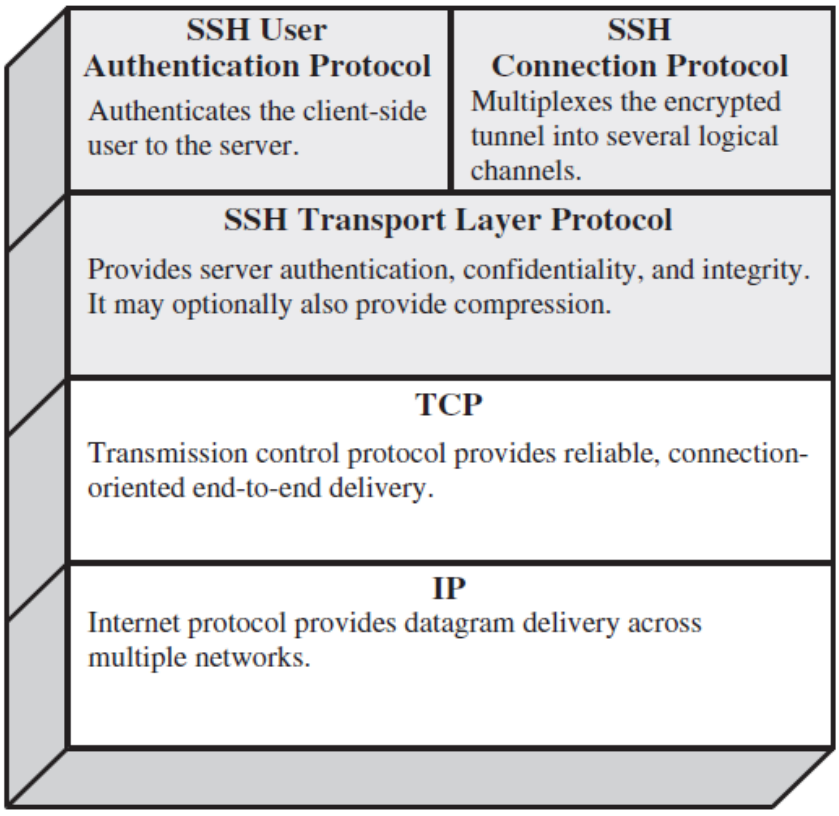


Fig: SSH Protocol Stack

Transport Layer Protocol: which typically runs on top of TCP/IP. This layer handles initial key exchange as well as server authentication, and sets up encryption, compression and integrity verification.

Or

Provides server authentication, data confidentiality, and data integrity with forward secrecy. The transport layer may optionally provide compression.

User Authentication Protocol: This layer handles client authentication and provides a number of authentication methods

Or

Authenticates the user to the server.

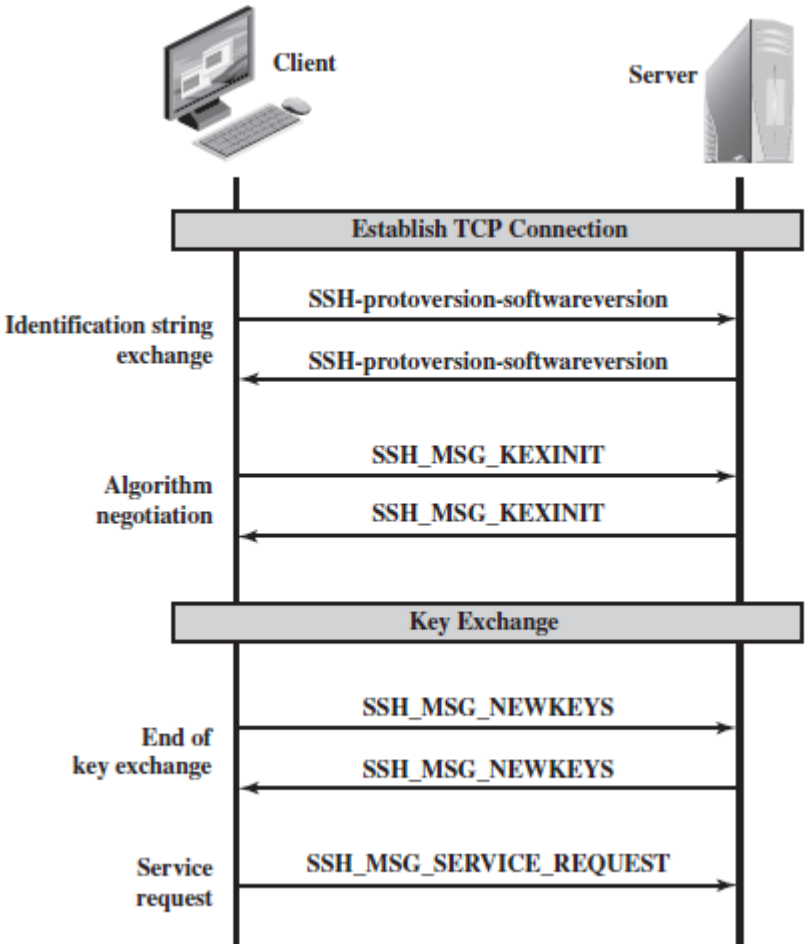
Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.

TCP: Transmission control protocol provides reliable, connection oriented end-to-end delivery.

IP: Internet protocol provides datagram (datagram is a basic transfer unit, associated with a packet-switched network) delivery across multiple networks.

SSH TRANSPORT LAYER PROTOCOL PACKET EXCHANGES 8m

Below Figure illustrates the sequence of events in the SSH Transport Layer Protocol. First, the client establishes a TCP connection to the server. This is done via the TCP protocol and is not part of the Transport Layer Protocol. Once the connection is established, the client and server exchange data, Referred to as packets, in the data field of a TCP segment.



DEC-2019|8M

Figure: SSH Transport Layer Protocol Packet Exchanges

The SSH Transport Layer packet exchange consists of a sequence of steps (Above Figure).

1. **The first step**, the identification string exchange, begins with the client sending a packet with an identification string of the form:

   SSH-protoversion-software version SP comments CR LF

   Where SP, CR, and LF are space character, carriage return, and line feed, respectively

2. **Next comes algorithm negotiatio**n. Each side sends an **SSH_MSG_KEXINIT** containing lists of supported algorithms in the order of preference to the sender. There

is one list for each type of cryptographic algorithm. The algorithms include key exchange, encryption, MAC algorithm, and compression algorithm.

3. The next step is **key exchange**. As a result of these steps, the two sides now share a master key *K*. In addition, he server has been authenticated to the client, because the server has used its private key to sign its half of the Diffie-Hellman exchange. Finally, the hash value *H* serves as a session identifier for this connection. Once computed, the session identifier is not changed, even if the key exchange is performed again for this connection to obtain fresh keys.
4. The **end of key exchange** is signalled by the exchange of **SSH_MSG_NEWKEYS** packets. At this point, both sides may start using the keys generated from *K*, as discussed subsequently.
5. The final step is **service request**. The client sends an **SSH_MSG_SERVICE_REQUEST** packet to request either the User Authentication or the Connection Protocol. Subsequent to this, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

SSH TRANSPORT LAYER PROTOCOL PACKET FORMATION

8m (JUNE/JULY-2019)

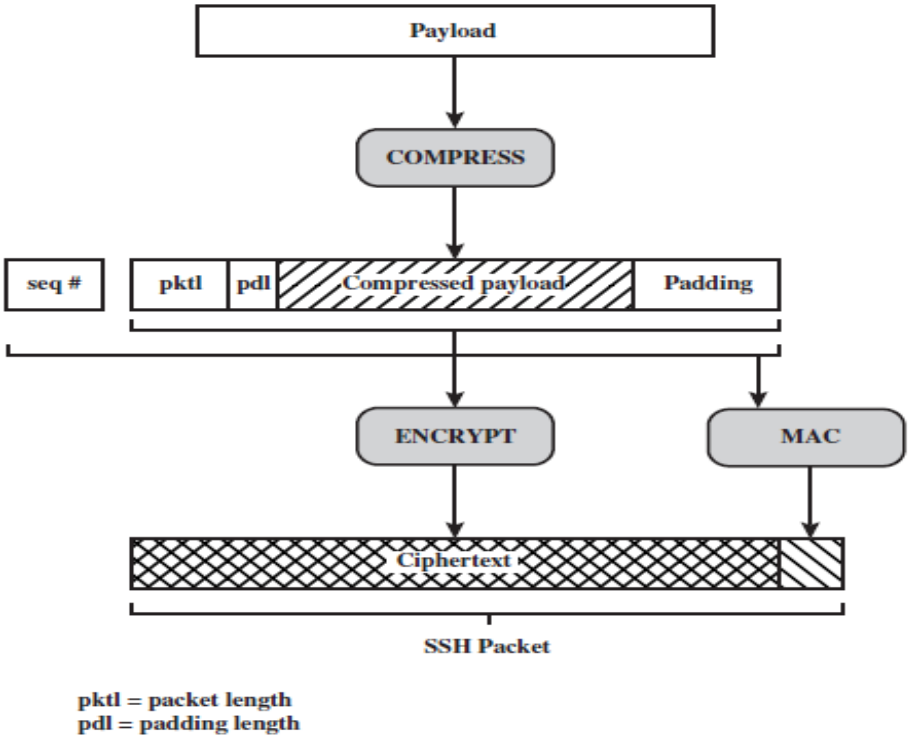


Figure: SSH Transport Layer Protocol Packet Formation

Each packet is in the following Format (Below Figure).

- **Packet length:** Length of the packet in bytes, not including the packet length and MAC fields.
- **Padding length:** Length of the random padding field.
- **Payload:** Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.
- **Random padding:** Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or 8 bytes for a stream cipher.
- **Message authentication code (MAC):**
  - If message authentication has been negotiated, this field contains the MAC value.
  - The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field.
  - The sequence number is an implicit 32-bit packet sequence that is initialized to zero for the first packet and incremented for every packet.

CHANNEL MECHANISM or SSH CONNECTION PROTOCOL MESSAGE EXCHANGE 6m(JAN-2020)

1. Communication using SSH, such as a terminal session, are supported using separate channels. Either side may open a channel.
2. For each channel, each side associates a unique channel number, which need not be the Same on both ends.
3. Channels are flow controlled using a window mechanism.
4. No data may be sent to a channel until a message is received to indicate that window space is available.
5. When either side wishes to **open a new channel,** it allocates a local number for the channel and then sends a message of the form:

Byte	**SSH_MSG_CHANNEL_OPEN**
String	channel type
Uint32	sender channel
Uint32	initial window size
Uint32	maximum packet size

…. channel type specific data follows

Where uint32 means unsigned 32-bit integer

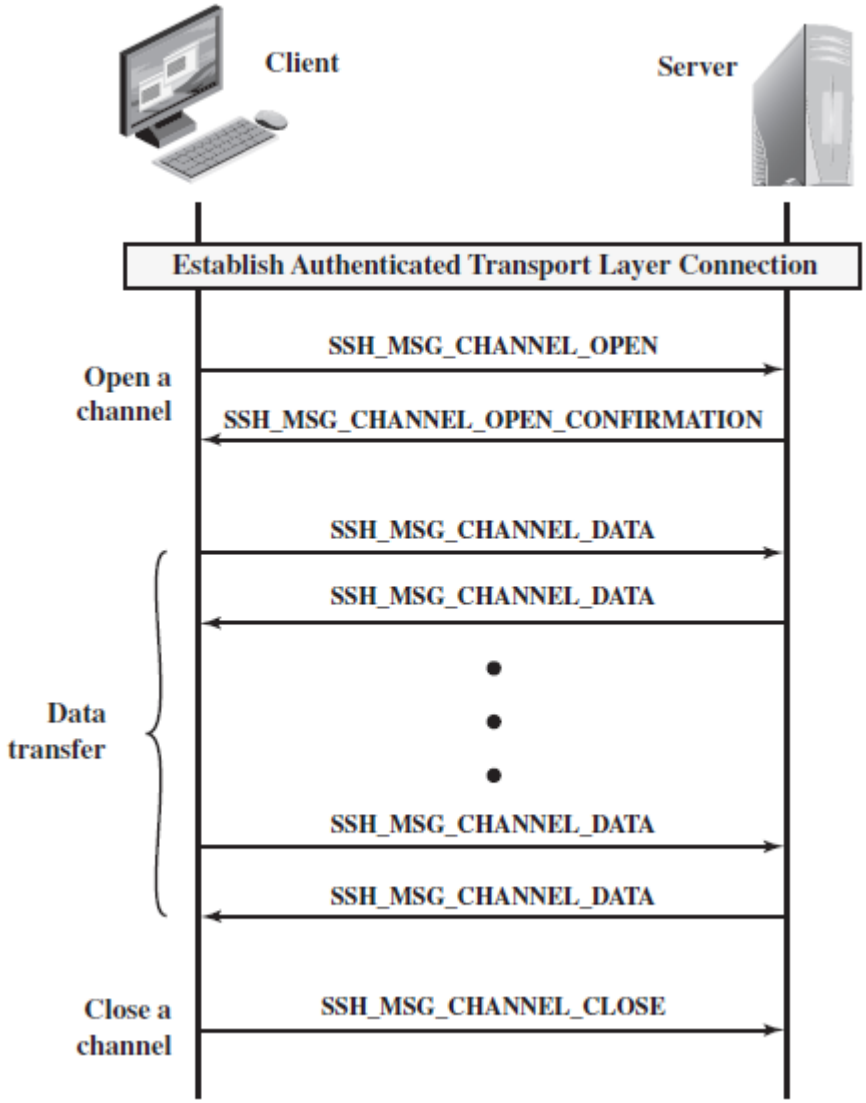


Figure Example of SSH Connection Protocol Message Exchange

6. If the remote side is able to open the channel, it returns a **SSH_MSG_CHANNEL_OPEN_CONFIRMATION** message, which includes the sender channel number, the recipient channel number, and window and packet size values for incoming traffic. Otherwise, the remote side returns a **SSH_MSG_CHANNEL_OPEN_ FAILURE** message with a reason code indicating the reason for failure.
7. Once a channel is open, **data transfer** is performed using a **SSH_MSG_CHANNEL_DATA** message, which includes the recipient channel number and a block of data. These messages, in both directions, may continue as long as the channel is open.

8. When either side wishes to **close a channel,** it sends a SSH_MSG_CHANNEL_ CLOSE message, which includes the recipient channel number.

PORT FORWARDING or SSH TRANSPORT LAYER PACKET EXCHANGES 8m (NOV-2020, SEP-2020)

- One of the most useful features of SSH is port forwarding. In essence, port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. This is also referred to as SSH tunnelling.
- A **port** is an identifier of a user of TCP. So, any application that runs on top of TCP has a port number.
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number. An application may employ multiple port numbers.

Figure (Below) illustrates the basic concept behind port forwarding.

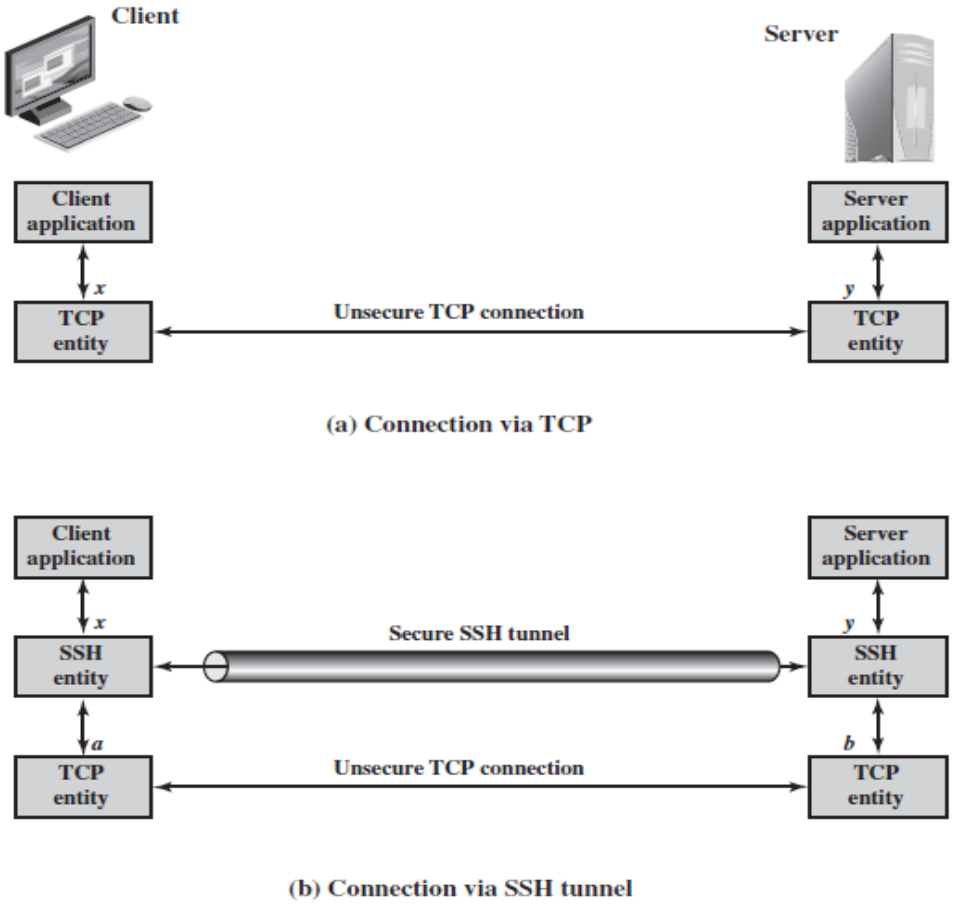


Figure: SSH Transport Layer Packet Exchanges

- We have a client application that is identified by port number *x* and a server application identified by port number *y*.

- At some point, the client application invokes the local TCP entity and requests a connection to the remote server on port *y*.
- The local TCP entity negotiates a TCP connection with the remote TCP entity, such that the connection links local port *x* to remote port *y*.
- To secure this connection, SSH is configured so that the SSH Transport Layer Protocol establishes a TCP connection between the SSH client and server entities, with TCP port numbers *a* and *b*, respectively.
- A secure SSH tunnel is established over this TCP connection.
- Traffic from the client at port *x* is redirected to the local SSH entity and travels through the tunnel where the remote SSH entity delivers the data to the server application on port *y*.
- Traffic in the other direction is similarly redirected.
- SSH supports two types of port forwarding: **local forwarding and remote forwarding**.

LOCAL FORWARDING:

Allows the client to set up a "hijacker" process. This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.

Or

Local forwarding is used to forward a port from the client machine to the server machine. Basically, the SSH client listens for connections on a configured port, and when it receives a connection, it tunnels the connection to an SSH server.

Or

Local port forwarding is the most common type of port forwarding. It is used to let a user connect from the local computer to another server, i.e. forward data securely from another client application running on the same computer as a Secure Shell (SSH) client.

REMOTE FORWARDING

The user's SSH client acts on the server's behalf. The client receives traffic with a given destination port number, places the traffic on the correct port and sends it to the destination the user chooses.

Or

Remote port forwarding is the exact opposite of local port forwarding. It forwards traffic coming to a port on your server to your local computer, and then it is sent to a destination.

ALERT CODES: TRANSPORT LAYER SECURITY 5m (NOV-2020, DEC-2019)

- TLS supports all of the alert codes defined in SSLv3 with the exception of no _ certificate.
- A number of additional codes defined in TLS; of these, the following are always fatal.
  - **Record _ overflow:** A TLS record was received with a payload (cipher text) whose length exceeds 214 + 2048 bytes, or the cipher text decrypted to a length of greater than 214 + 1024 bytes.
  - **Unknown _ ca:** A valid certificate chain or partial chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.
  - **Access _ denied:** A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.
  - **Decode _ error:** A message could not be decoded, because either a field was out of its specified range or the length of the message was incorrect.
  - **Protocol _version:** The protocol version the client attempted to negotiate is recognized but not supported.
  - **Insufficient _ security:** Returned instead of handshake _ failure when a negotiation has failed specifically because the server requires ciphers more secure than those supported by the client.
  - **Unsupported _ extension:** Sent by clients that receives an extended server hello containing an extension not in the corresponding client hello.
  - **Internal _ error:** An internal error unrelated to the peer or the correctness of the protocol makes it impossible to continue.
  - **Decrypt _ error:** A handshake cryptographic operation failed, including being unable to verify a signature, decrypt a key exchange, or validate a finished message.

The remaining alerts include the following.

- **User _ cancelled:** This handshake is being cancelled for some reason unrelated to a protocol failure.

- **No _ renegotiation:** Sent by a client in response to a hello request or by the server in response to a client hello after initial handshaking. Either of these messages would normally result in renegotiation, but this alert indicates that the sender is not able to renegotiate. This message is always a warning.

COMPARISON OF THREATS ON THE WEB 8m (NOV-2020)

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

MODULE-2 (KEY POINTS)

E - Mail security

PGP: - **5m**

- Stands for Pretty good privacy.
- Proposed by PHIL Zimmermann, early 2000s. or the Father of is PHIL Zimmermann.
- The main purpose of PGP is Email Security.
- The best one which provides the EMAIL Security is PGP.
- PGP is an Encryption System or Pretty Good Privacy (**PGP**) is an e-mail encryption scheme.
- PGP is an encryption Programme OR Software that provides cryptographic privacy, Authentication and Integrity for Data Communications.
- PGP uses a combination of secret key encryption and public key encryption to provide privacy. It provides confidentiality through the use of symmetric block encryption. It provides compression by using the ZIP algorithm, and EMAIL compatibility using the radix-64 encoding scheme.
- It is a user-end-to-user-end secure communication.
- It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, and many more.
- PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.
- It is based on algorithms that have survived extensive public review and are considered extremely secure. Specifically, the package includes RSA, DSS, and Diffie-Hellman for public-key encryption; CAST-128, IDEA, and 3DES for symmetric encryption; and SHA-1 for hash coding.
- PGP provides a some services like
    1. Authentication and Digital signature
    2. Confidentiality
    3. Compression(ZIP)
    4. E-mail compatibility.

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or 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 included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applica-tions, an encrypted message may be converted to an ASCII string using radix-64 conversion.

Table: Summary of PGP Services

- The Main Disadvantages of PGP is
  - At the sender and receiver side we have to maintain the same versions of PGP.
  - PGP is very difficult process because PGP uses a Combination of Symmetric and asymmetric keys that is it uses Hybrid keys its somewhat difficult process.
- The following symbols are used in the PGP operations

  Ks = session key used in symmetric encryption scheme

  PRa = private key of user A, used in public-key encryption scheme

  PUa = public key of user A, used in public-key encryption scheme

  EP = public-key encryption

  DP = public-key decryption

  EC = symmetric encryption

  DC = symmetric decryption

  H = hash function

  ||= concatenation

  Z = compression using ZIP algorithm

  R64 = conversion to radix 64 ASCII format

PGP Cryptographic Functions or PGP operations **12 or 14 Marks**

1PGP Operation- Authentication IMP QS (question)-06M

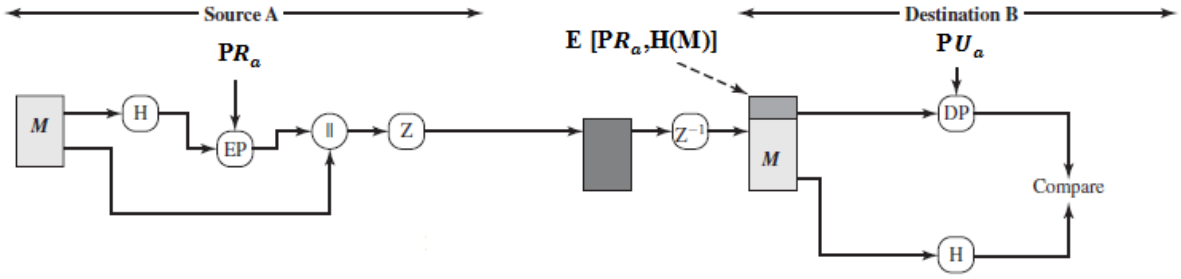


Figure 1 Authentication and Digital signature only

Above Figure 1:-

SENDER:

1. The sender creates a message.
2. SHA-1 is used to generate a 160-bit hash code of the message.
3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended or appended to the message.

RECEIVER:

1. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
2. The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

Note

- In the Encryption side, we are using private key like, digital signature is achieved.
- In the decryption side, if the two messages is match, the message is accepted authentic.

2. PGP Operation- Confidentiality

IMP QS (question)-06M

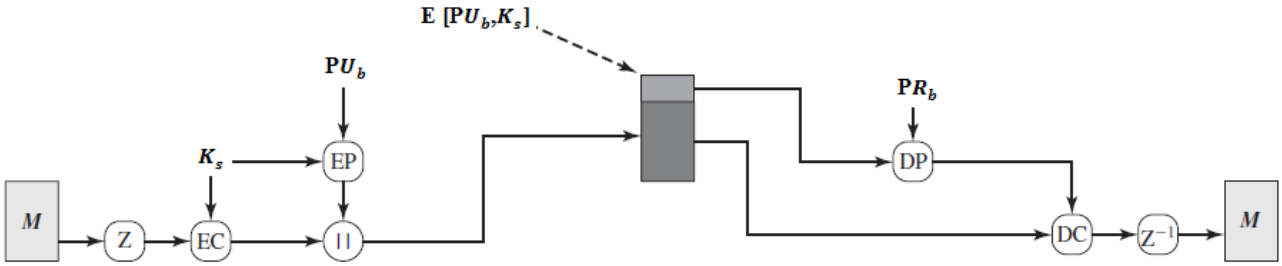


Figure 2 Confidentiality only

Above Figure 2:-

SENDER:

1. The sender generates a message and a random 128-bit number to be used as a session key for this message only.
2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.
3. The session key is encrypted with RSA using the recipient's public key and is prepended to the message.

RECEIVER:

1. The receiver uses RSA with its private key to decrypt and recover the session key.
2. The session key is used to decrypt the message.

Note

Confidentiality achieved like

- Before the encryption we applying a zip function.
- Encryption side we are applying symmetric encryption key.

3 PGP Operation- Confidentiality and Authentication IMP QS (question)-06M

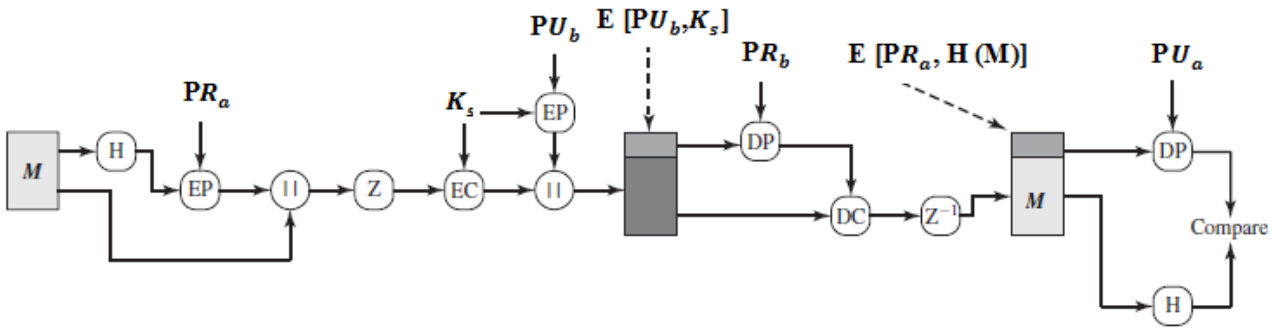


Figure 3 Confidentiality and authentication

Above Figure 3:-

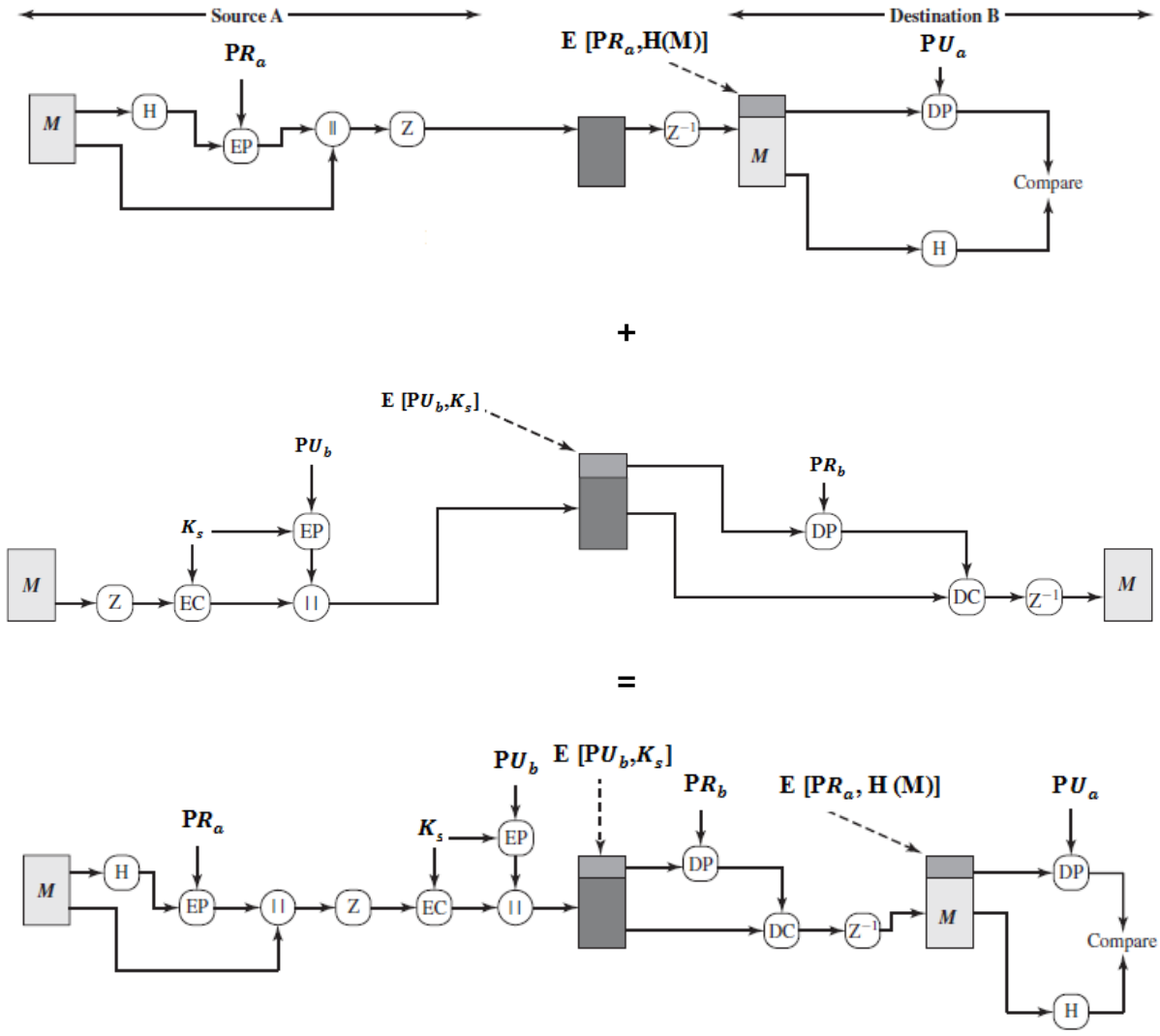
1. Both services may be used for the same message. First, a signature is generated for the plaintext message and prepended to the message.

2. Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA .This sequence is preferable to the opposite: encrypting the message and then generating a signature for the encrypted message.

3. It is generally more convenient to store a signature with a plaintext version of a message. Furthermore, for purposes of third-party verification, if the signature is performed first, a third party need not be concerned with the symmetric key when verifying the signature.

In summary, when both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and finally encrypts the session key with the recipient's public key.

NOTE



4 Compression 04M

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage. The placement of the compression algorithm, indicated by Z for compression and $Z-1$for decompression in figure (1, 2 &3) critical. The compression algorithm used is ZIP.

1. The signature is generated before compression for two reasons:

a) It is preferable to sign an uncompressed message so it is free of the need for a compression algorithm for later verification.

Or

So that one can store only the uncompressed message together with signature for later verification

b) Different version of PGP produces different compressed forms. Applying the hash function and signature after compression would constrain all PGP implementation to the same version of the compression algorithm.

Or

Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm as the PGP compression algorithm is not deterministic.

2. Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.

KEY IDENTIFIERS OR PGP MESSAGE FORMAT IMP QS (question)-08M

The concept of key ID has been introduced; we can take a more detailed look at the format of a transmitted message, which is shown in Below Figure

- A message consists of three components: **the message component, a signature (Optional), and a session key component (optional).**
- The **message component** includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation.
- The **signature component** includes the following:
  - **Timestamp:** The time at which the signature was made.
  - **Message digest:** The 160-bit SHA-1 digest, encrypted with the sender's private signature key.
  - **Leading two octets of message digest:** To enable the recipient to determine if the correct public key was used to decrypt the message digest for authentication, by comparing this plaintext copy of the first two octets (The octet is a unit of digital information in computing and telecommunications that consists of eight bits) with the first two octets of the decrypted digest. These octets also serve as a 16-bit frame check sequence (refers to the extra bits and characters added to data packets for error detection and control) for the message.

- **Key ID of sender's public key:** Identifies the public key that should be used to decrypt the message digest and, hence, identifies the private key that was used to encrypt the message digest.

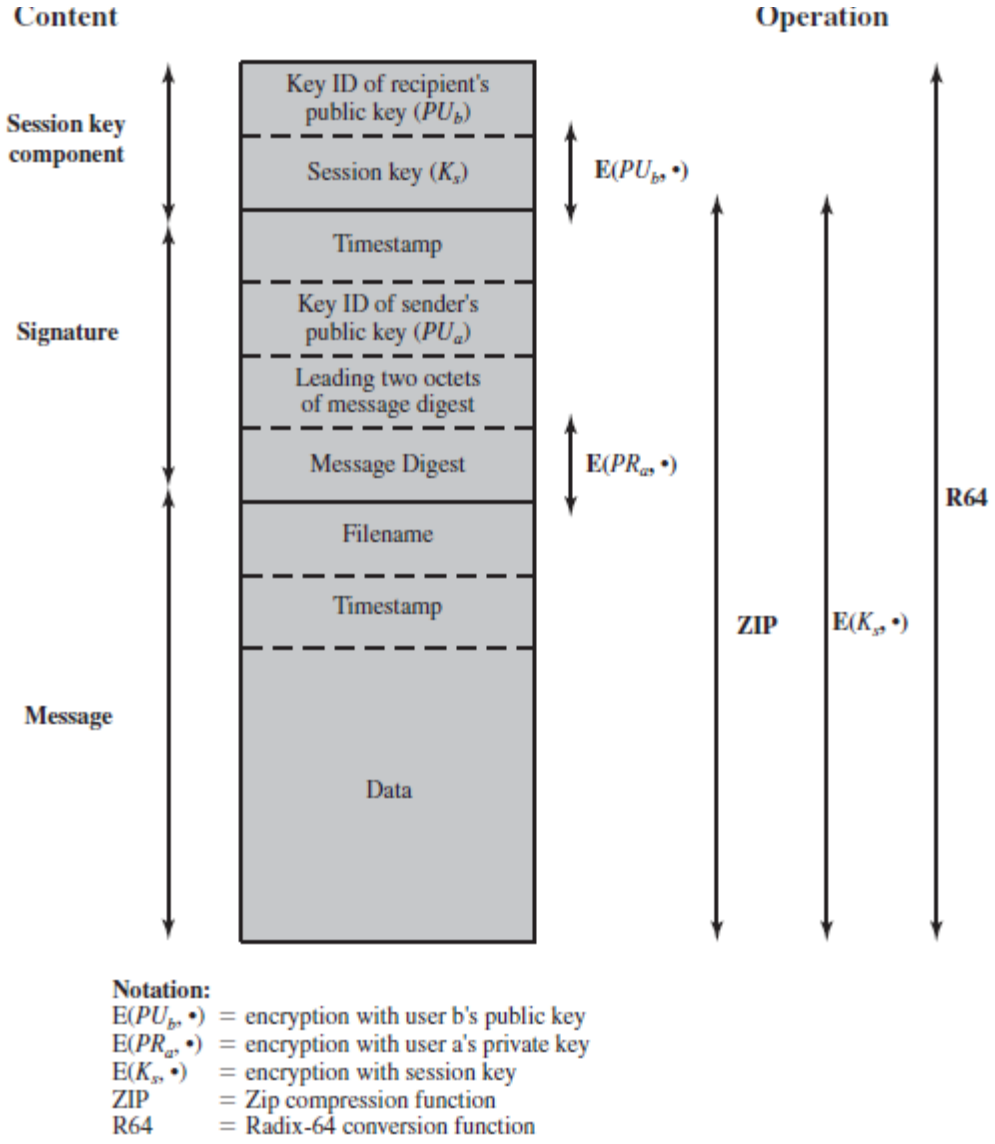


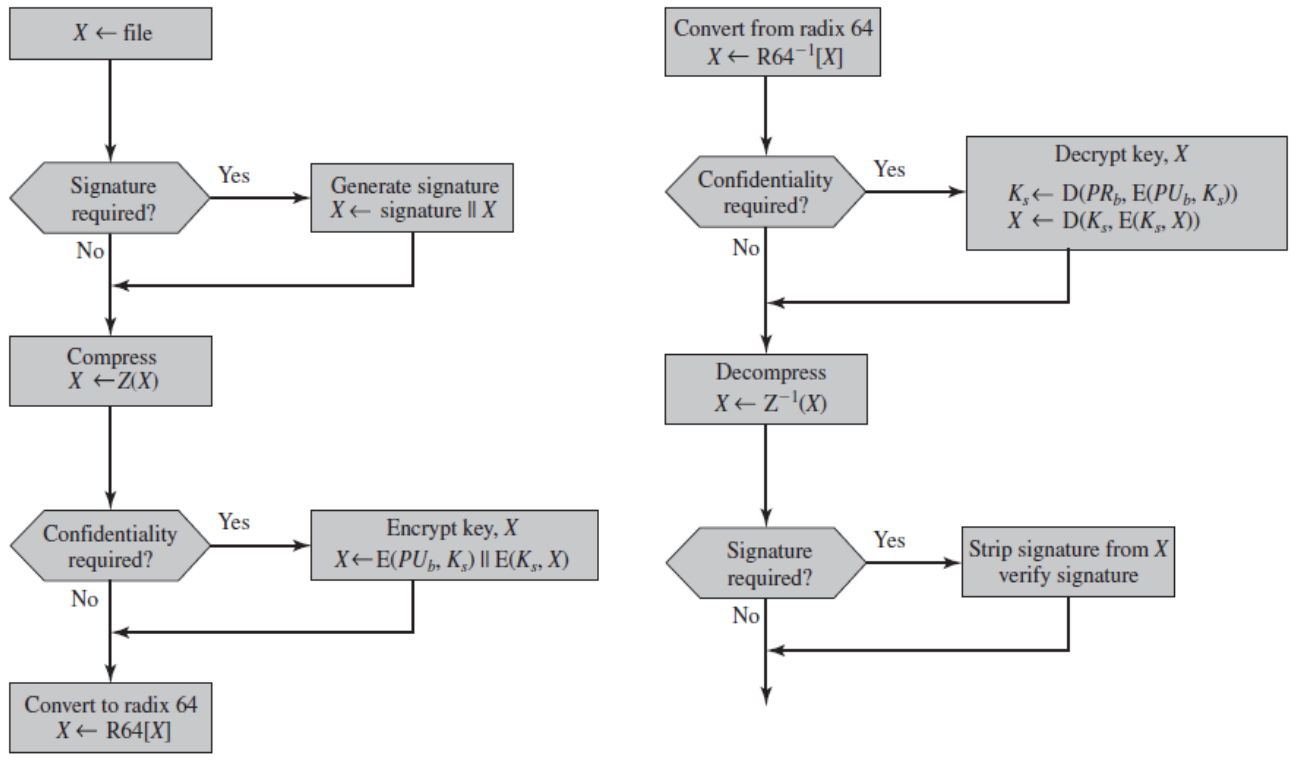
Figure 5 General Format PGP Message (from A to B)

- A **timestamp** is the current time of an event that is recorded by a computer.
- The **session key component** includes the session key and the identifier of the Recipient's public key that was used by the sender to encrypt the session key.
- The entire block is usually encoded with radix-64 encoding.

PGP Operation – Email Compatibility or Transmissions and Reception of PGP Messages DEC-2019(10M), SEPT-2020(10M), JULY-2019[10M]

When PGP is used, at least part of the block to be transmitted is encrypted, and thus consists of a stream of arbitrary 8-bit octets (The **octet** is a unit of digital information in computing and telecommunications that consists of eight bits).

- However many electronic mail systems only permit the use of ASCII text. To accommodate this restriction, PGP provides the service of converting the raw (Raw data is Uncompressed Computer data) 8-bit binary stream to a stream of printable ASCII characters.
- It uses radix-64 conversion, in which each group of three octets of binary data is mapped into four ASCII characters. This format also appends a CRC (Cyclic Redundancy Check in computer networks is an error **detection** method) to detect transmission errors. The use of radix 64 expands a message by 33%, but still an overall compression of about one- third can be achieved.



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

Figure 4 Transmissions and Reception of PGP Messages

Transmissions

1. On transmission (if it is required), a signature is generated using a hash code of the uncompressed plaintext.
2. Then the plaintext (plus signature if present) is compressed. Next, if confidentiality is required, the block (compressed plaintext or compressed signature plus plaintext) is encrypted and prepended with the public key-encrypted symmetric encryption key.
3. Finally, the entire block is converted to radix-64 format.

Reception

4. On reception, the incoming block is first converted back from radix-64 format to binary.
5. If the message is encrypted, the recipient recovers the session key and decrypts the message. The resulting block is then decompressed.
6. If the message is signed, the recipient recovers the transmitted hash code and compares it to its own calculation of the hash code.

PGP MESSAGE GENERATIONS AND RECEPTION OR KEY RINGS

IMP QS (question)-10M

1. Message transmission

The following figure 6 shows the steps during message transmission assuming That the Message is to be both signed and encrypted.

The sending PGP entity performs the following steps

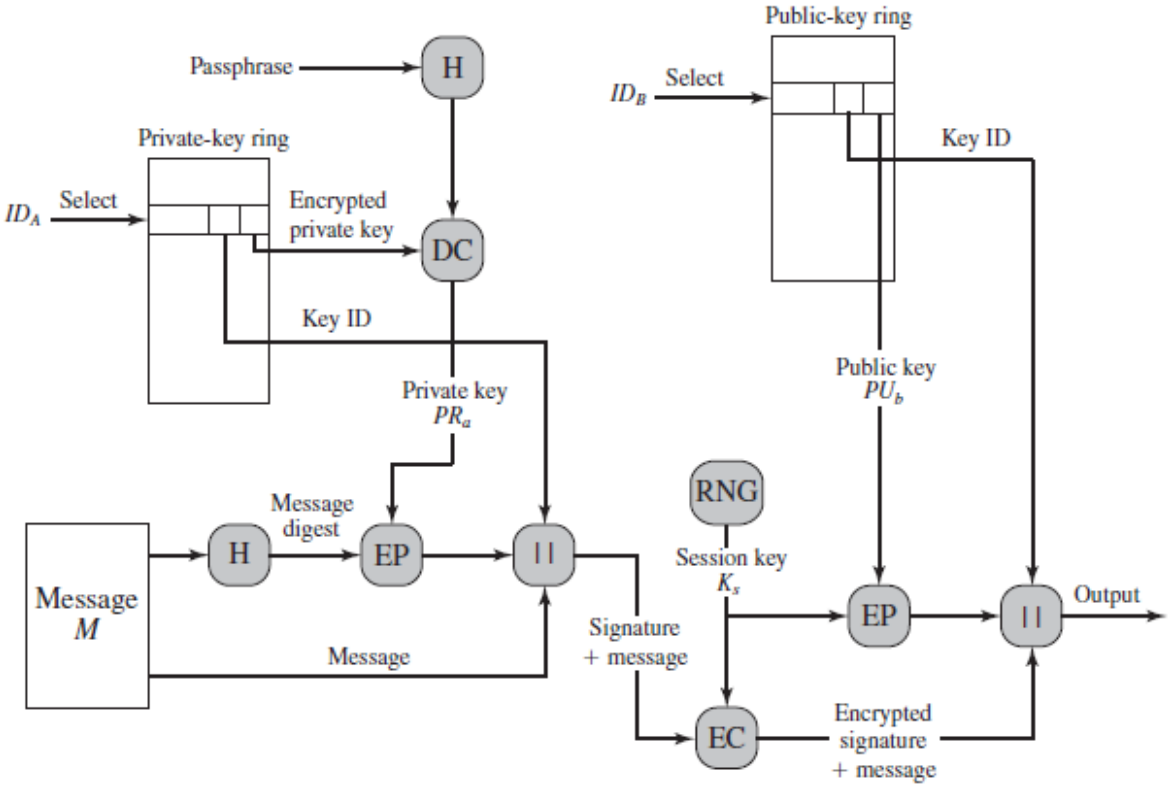


Figure: - PGP Message Generation (from User A to User B: no compression or radix-64 Conversion)

SIGNING THE MESSAGE

a. PGP retrieves the sender's private key from the private-key ring using your _ user id as an index. If your_ user id was not provided in the command, the first private key on the ring is retrieved.
b. PGP prompts the user for the passphrase to recover the unencrypted private key.
c. The signature component of the message is constructed.

ENCRYPTING THE MESSAGE

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

b. PGP retrieves the recipient's public key from the public-key ring using her _ user id as an index.

c. The session key component of the message is constructed.

2 Message Receptions

The receiving PGP entity performs the following steps (Below figure)

a. PGP retrieves the receiver's private key from the private-key ring using the Key ID field in the session key component of the message as an index.

b. PGP prompts the user for the passphrase to recover the unencrypted private key.

c. PGP then recovers the session key and decrypts the message.

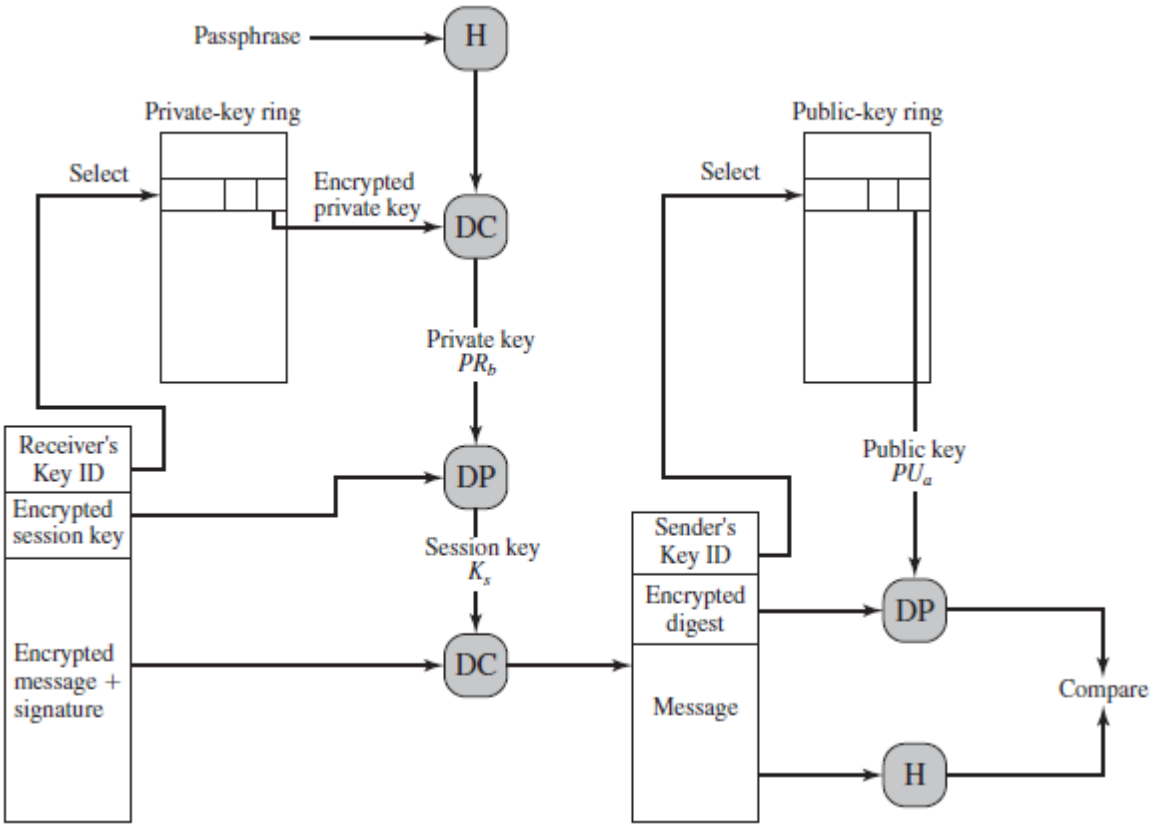


Figure: - PGP Message Reception (from User A to User B; no compression or radix-64 conversion)

AUTHENTICATING THE MESSAGE

a. PGP retrieves the sender's public key from the public-key ring, using the Key ID field in the signature key component of the message as an index.

b. PGP recovers the transmitted message digest.

c. PGP computes the message digest for the received message and compares it to the transmitted message digest to authenticate.

RADIX-64 CONVERSION

IMP QS (question)-06M

- Both PGP and S/MIME (Secure/Multipurpose Internet Mail Extension (S/MIME)) make use of an encoding technique referred to as radix-64 Conversion. This technique maps arbitrary binary input into printable character output.
- Many electronic mail systems can only transmit blocks of ASCII text. This can cause a problem when sending encrypted data since cipher text blocks might not correspond to ASCII characters which 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 as follows:

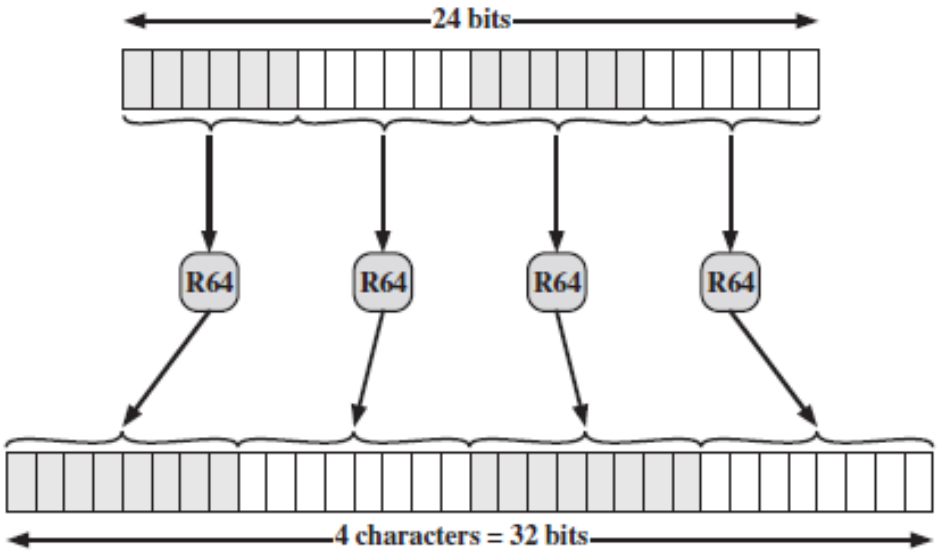


Figure:-Printable Encoding of Binary Data into Radix-64 Format

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 63 (2^6 = 64, 64-1=63)
4. This value is encoded (as 8-bit quantities) into a printable character.

For example (Old radix 64 technique):-

Consider the 24-bit raw text sequence 00100011 01011100 10010001, Which can be expressed in hexadecimal as 235C91. We arrange this input in blocks of 6 Bits: 001000 110101 110010 010001

6-bit Value	Character Encoding	6-bit Value	Character Encoding	6-bit Value	Character Encoding	6-bit Value	Character Encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

Table: Radix-64 Encoding

The extracted 6-bit decimal values are 8, 53, 50, and 17. Looking these up in Below Table Yields the radix-64 encoding as the following characters: I1yR. If these characters are Stored in 8-bit ASCII format with parity bit set to zero, we have 01001001 00110001 01111001 01010010

In hexadecimal, this is 49317952. To summarize:

Input Data	
Binary representation	00100011 01011100 10010001
Hexadecimal representation	235C91
Radix-64 Encoding of Input Data	
Character representation	I1yR
ASCII code (8 bit, zero parity)	01001001 00110001 01111001 01010010
Hexadecimal representation	49317952

Advance Radix-64 Encryption and Decryption Technique :-

Eg :- Consider Mail Text Data. → Hey.

Encryption :-

Input → H e y.

ASCII → 72 101 121

Binary (Bits) → 01001000 01100101 01111001

Consider 6 Bits → 18 6 21 57

Using Radix 64 Table → S Gr V 5

↳ Encrypted printable character o/p.

NOTE:
In the ASCII Table.
H → 72
e → 101
y → 121.

NOTE:
In the Radix 64 Table
18 → S
6 → Gr
21 → V
57 → 5

Decryption : is the Reverse function of Encryption

Printable Character → S Gr V 5

Represented Numbers → 18 6 21 57

Convert Binary (6 Bits) → 010010 000110 010101 111001

Convert ASCII (Consider 8 bits) → 72 101 121

o/p (using ASCII) → H e y //

S/MIME

IMP QS (question)-06M

- When email was 1st developed, people could only send plain text messages or earlier days peoples could only send plain text messages to email.
- MIME (Multipurpose Internet Mail Extension) was developed in early 90s to allow people to send picture , sounds, programmes and general attachments.
- MIME has no security features, can be read along its route or forged (easily).
- S/MIME is a secure version of MIME.
- S/MIME - Secure/Multipurpose Internet Mail Extensions.
- S/MIME (Secure/Multipurpose Internet Mail Extensions) is a standard for public key encryption and signing of MIME data.
- S/MIME provides similar services to PGP.
- Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.
- Both PGP and S/MIME are on an IETF(Internet Engineering Task Force (IETF) is an open standards organization, which develops and promotes voluntary Internet standards,) standards track, it appears likely that S/MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many users.
- This S/MIME is industry standard for Commercial and organization use but PGP use for only personal e-mail security.
- S/MIME is defined in a number of documents—most importantly RFCs 3370, 3850, 3851, and 3852.

RFC 5322

IMP QS (question)-06M

- RFC 5322 defines a format for text messages that are sent using electronic mail.
- It has been the standard for Internet-based text mail messages and remains in common use.
- In the RFC 5322 context, messages are viewed as having an envelope and contents.
- The envelope contains whatever information is needed to accomplish Transmission and delivery.

- The contents compose the object to be delivered to the recipient.
- The RFC 5322 standard applies only to the contents.
- The content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.
- The overall structure of a message that conforms to RFC 5322 is very simple.
- A message consists of some number of header lines (the header) followed by Unrestricted text (the body).
- The header is separated from the body by a blank line. Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system.

MULTIPURPOSE INTERNET MAIL EXTENSIONS **IMP QS (question)-10M**

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail.

OR

Multipurpose Internet Mail Extensions (MIME) is an Internet standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of audio, video, images, and application programs.

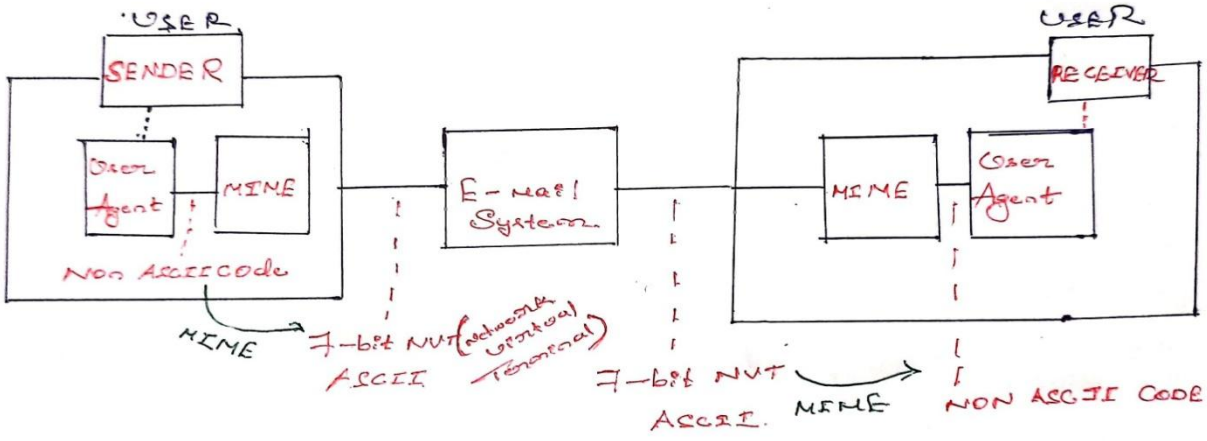


Fig :- MIME Transformation.

SENDER:

User (Sender):-User by using some User agent, user going to send some non ASCII code to MIME.

MIME: - is going to convert the Message in the form of 7-bit ASCII. OR it converts Non ASCII to ASCII Code.

E-Mail system: - MIME sends a 7-bit ASCII data through email system to the receiver by using an internet.

RECEIVER:

MIME:-is convert the 7bit ASCII data into Non ASCII Message.

User (Receive):- receiver (user) receives the non ASCII data from the user agent .

NOTE: -

SMTP

- The Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol for electronic mail transmission. Mail servers and other message transfer agents use SMTP to send and receive mail messages. ... Since SMTP's introduction in 1981, it has been updated, modified and extended multiple times.

RFC5322 and RFC5321

- The email message standard is defined in RFC 5322 also known as IMF or Internet Message Format. The email addressing standard is defined in RFC 5321(updated) also known as SMTP or Simple Mail Transport Protocol.

HTTP and SMTP

- **SMTP** and **HTTP** are both network layer protocols that are used to transfer information between hosts. **SMTP** is used to transfer emails between mail servers, while **HTTP** is used to transfer data from a web server to a web client.

VTP

- A network virtual terminal is a communications concept describing a variety of data terminal equipment (DTE), with different data rates, protocols, codes and formats, accommodated in the same network.

X.400

- **X.400** is a suite of protocols defining standards for **email messaging** system.

Lists the following limitations of the SMTP/5322 scheme.

1. SMTP cannot transmit executable files or other binary objects.
2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle non textual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821.

The MIME specification includes the following elements.

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

The five header fields defined in MIME are

1. **MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.

```
MIME-Version: 1.0
```

2. **Content-Type:** Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner.

```
Content-Type: text/plain
```

3. **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.

```
Content-Disposition: attachment; filename=genome.jpeg;
 modification-date="Wed, 12 Feb 1997 16:29:51 -0500";
```

4. **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
5. **Content-Description:** A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

INTERNET MAIL ARCHITECTURE IMP QS (question)-10M

The Internet mail architecture consists of a user world in the form of Message User Agents (MUA), and the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTA).

Figure illustrates the key components of the Internet mail architecture, which include the following.

1. **Message User Agent (MUA):** Operates on behalf of user actors and user applications. It is their representative within the e-mail service. Typically, this function is housed in the user's computer and is referred to as a client e-mail program or a local network e-mail server. The author MUA formats a message and performs initial submission into the MHS via a MSA. The recipient MUA processes received mail for storage and/or display to the recipient user.

Or

Message User Agent (MUA):

- Is an email client application.
- MUA is a programme that at the very least allows a user to ready and compose email message.

2. **Mail Submission Agent (MSA):** Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards. This function may be located together with the MUA or as a separate functional model. In the latter case, the Simple Mail Transfer Protocol (SMTP) is used between the MUA and the MSA.

3. **Message Transfer Agent (MTA):** Relays mail for one application-level hop. It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the recipients. Relaying is performed by a sequence of MTAs until the message reaches a destination MDA. An MTA also adds trace information to the message header. SMTP is used between MTAs and between an MTA and an MSA or MDA.

Or

Message Transfer Agent (MTA):

- MTA transfer email message b/w hosts using SMTP.
- A Message many Involves several MTAs as it moves to its intended destination.

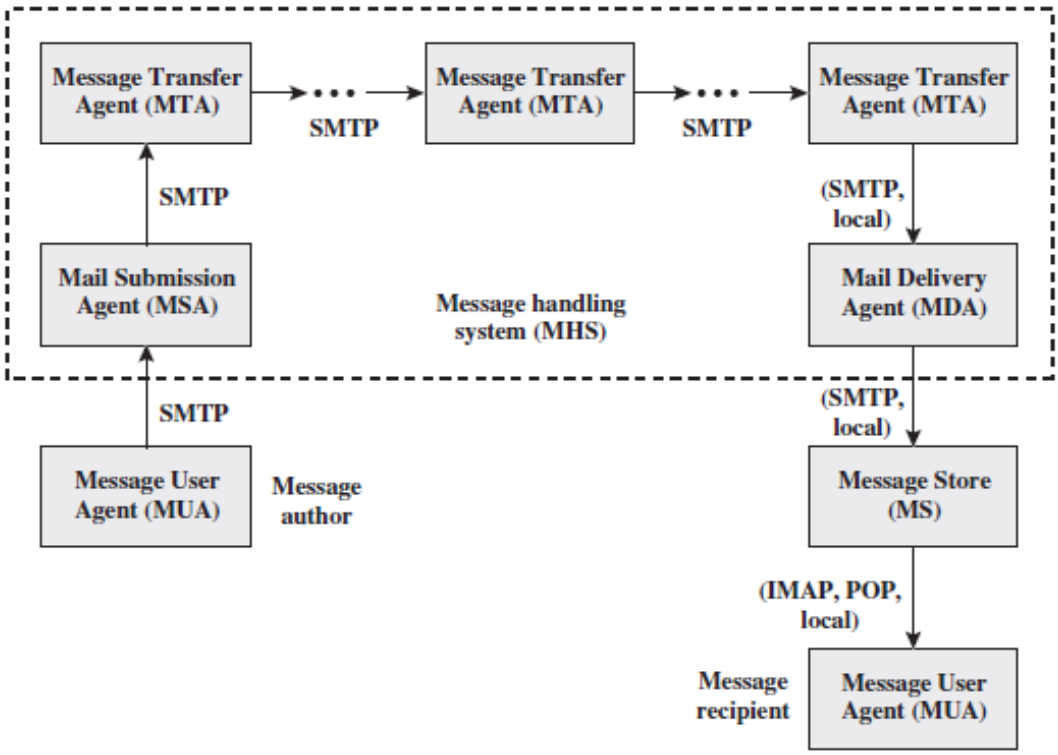


Figure : Function Modules and Standardized Protocols Used Between Them or Internet mail architecture

4. **Mail Delivery Agent (MDA):** Responsible for transferring the message from the MHS to the MS.

Or

Mail Delivery Agent (MDA):

- A mail delivery agent is utilizing by MTA to delivery email to a particular users mail box.
- In many users MDA is actually a logical delivery agent.
- Many users do not directly utilize MDAs because only MTA and MUA are necessary to send and receive email.

5. **Message Store (MS):** An MUA can employ a long-term MS. An MS can be located on a remote server or on the same machine as the MUA. Typically, an MUA retrieves messages from a remote server using POP (Post Office Protocol) or IMAP (Internet Message Access Protocol).

Two other concepts need to be defined

1. An **administrative management domain (ADMD)** is an Internet e-mail provider.
2. The **Domain Name System (DNS)** is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address.

DOMAIN KEYS IDENTIFIED MAIL

IMP QS (question)-06M

- DKIM is designed to provide an e-mail authentication technique that is transparent to the end user.
- In essence, a user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates. The signature covers all of the content of the message and some of the RFC 5322 message headers.
- At the receiving end, the MDA can access the corresponding public key via a DNS(The Domain Name System (DNS) is the Internet's system for mapping alphabetic names to numeric Internet Protocol (IP) addresses like a phone book maps a person's name to a phone number.)and verify the signature, thus authenticating that the message comes from the claimed administrative domain.

- Domain Keys Identified Mail (DKIM) is a specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream.
- Message recipients (or agents acting in their behalf) can verify the signature by querying the signer's domain directly to retrieve the appropriate public key and thereby can confirm that the message was attested to by a party in possession of the private key for the signing domain.
- DKIM is a proposed Internet Standard (RFC 4871: Domain Keys Identified Mail (DKIM) Signatures). DKIM has been widely adopted by a range of e-mail providers, including corporations, government agencies, Gmail, yahoo, and many Internet Service Providers (ISPs).
- The motivation for DKIM is based on the following reasoning.
    1. S/MIME depends on both the sending and receiving users employing S/MIME. For almost all users, the bulk of incoming mail does not use S/MIME, and the bulk of the mail the user wants to send is to recipients not using S/MIME.
    2. S/MIME signs only the message content. Thus, RFC 5322 header information concerning origin can be compromised.
    3. DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need take no action.
    4. DKIM applies to all mail from cooperating domains.
    5. DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders.

SIMPLE EXAMPLE OF DKIM DEPLOYMENT

Figure: - is a simple example of the operation of DKIM.

- We begin with a Message generated by a user and transmitted into the MHS to an MSA that is within the user's administrative domain.
- An e-mail message is generated by an e-mail client program. The content of the message, plus selected RFC 5322 headers, is signed by the e-mail provider using the provider's private key.

- The signer is associated with a domain (A network domain is an administrative grouping of multiple private computer networks or hosts within the same infrastructure.), which could be a corporate local network, an ISP (Internet service provider), or a public e-mail facility such as Gmail.

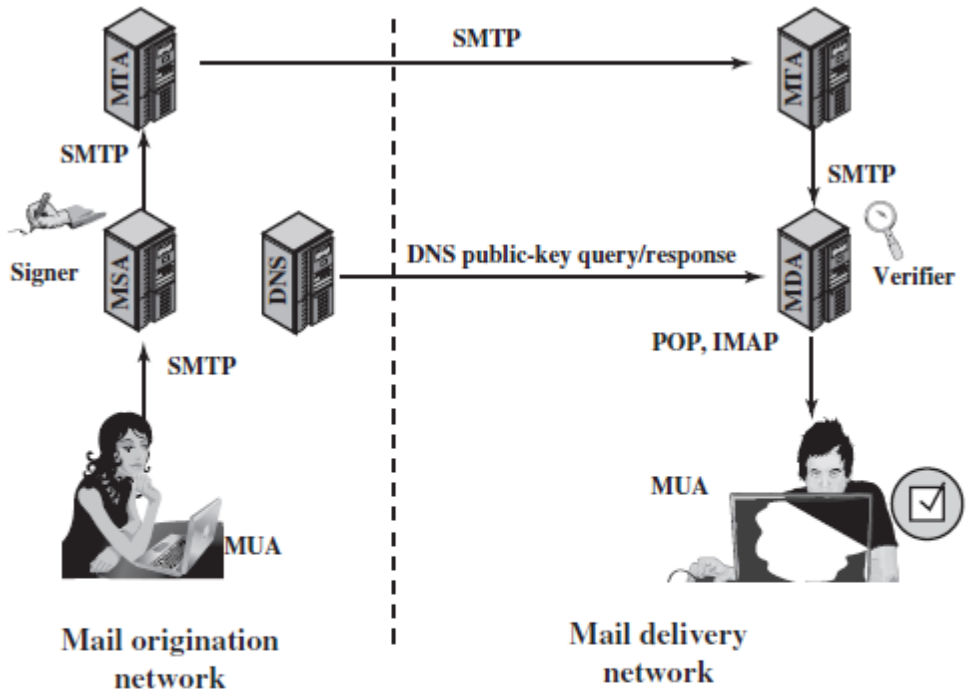


Figure: Simple Example of DKIM Deployment

- The signed message then passes through the Internet via a sequence of MTAs. At the destination, the MDA retrieves the public key for the incoming signature and verifies the signature before passing the message on to the destination e-mail client.
- The default signing algorithm is RSA with SHA-256. RSA with SHA-1 also may be used.

DKIM FUNCTIONAL FLOW IMP QS (question)-06M

- **Figure 11** provides a more detailed look at the elements of DKIM operation. Basic message processing is divided between a signing Administrative Management Domain (ADMD) and a verifying ADMD.
- At its simplest, this is between the originating ADMD and the delivering ADMD, but it can involve other ADMDs in the handling path.

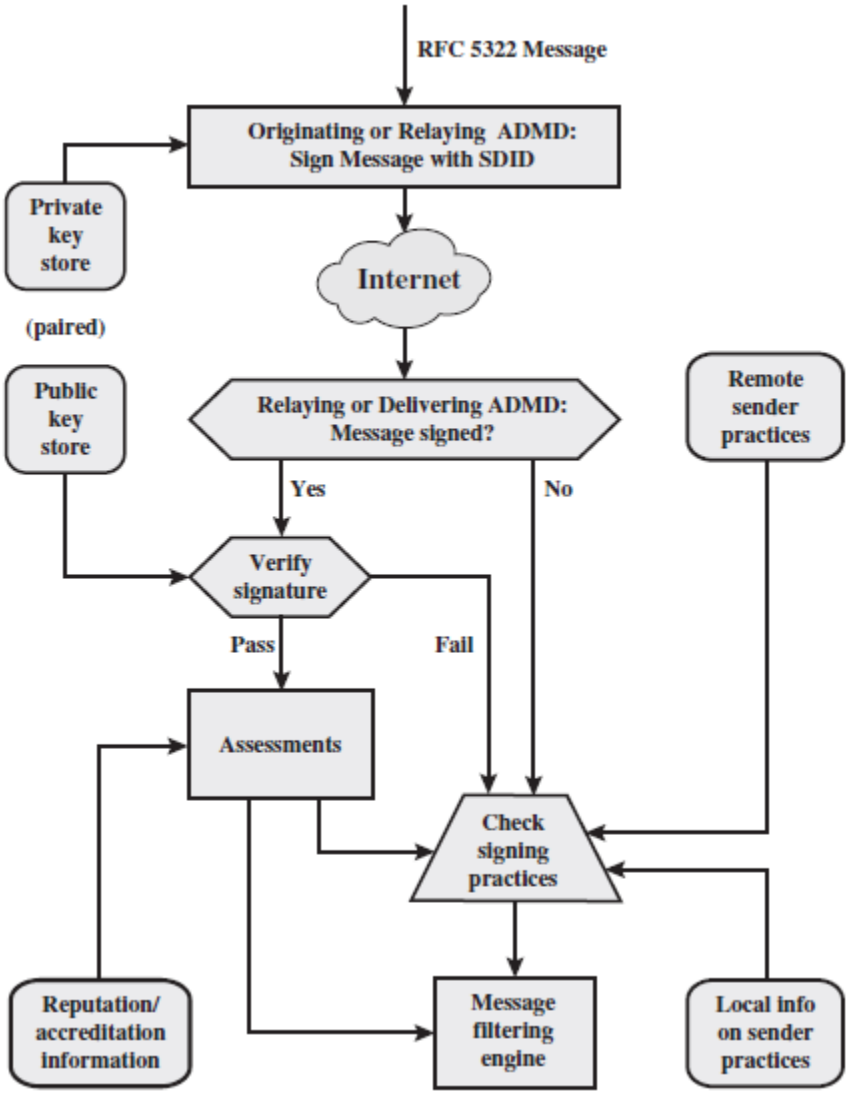


Figure : DKIM Functional Flow

- Verifying is performed by an authorized module within the verifying ADMD. Within a delivering ADMD, verifying might be performed by an MTA, MDA, or MUA.

- The module verifies the signature or determines whether a particular signature was required.
- Verifying the signature uses public information from the Key Store.
- If the signature passes, reputation information is used to assess the signer and that information is passed to the message filtering system.
- If the signature fails or there is no signature using the author's domain, information about signing practices related to the author can be retrieved remotely and/or locally, and that information is passed to the message filtering system.(For example, if the sender (e.g., Gmail) uses DKIM but no DKIM signature is present, then the message may be considered fraudulent)
- The signature includes a number of fields. Each field begins with a tag consisting of a tag code followed by an equals sign and ends with a semicolon. The fields include the following:

- **v** = DKIM version.
- **a** = Algorithm used to generate the signature; must be either rsa-sha1 or rsasha256.
- **c** = Canonicalization method used on the header and the body.
- **d** = A domain name used as an identifier to refer to the identity of a responsible person or organization. In DKIM, this identifier is called the Signing Domain Identifier (SDID). In our example, this field indicates that the sender is using a Gmail address.
- **s** = In order that different keys may be used in different circumstances for the same signing domain (allowing expiration of old keys, separate departmental signing, or the like), DKIM defines a selector (a name associated with a key), which is used by the verifier to retrieve the proper key during signature verification.
- **h** = Signed Header fields. A colon-separated list of header field names that identify the header fields presented to the signing algorithm. Note that in our example above, the signature covers the domain key-signature field. This refers to an older algorithm (since replaced by DKIM) that is still in use.
- **bh** = The hash of the canonicalized body part of the message. This provides additional information for diagnosing signature verification failures.
- **b** = the signature data in base64 format; this is the encrypted hash code.

MODULE-3

1. IP SECURITY OVERVIEW

The **IP security (IPSec)** is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted and authenticated packets. The protocols needed for secure key exchange and key management are defined in it.

Uses of IP security: - IPsec can be used to do the following things:

1. To encrypt application layer data.
2. To provide security for routers sending routing data across the public internet.
3. To provide authentication without encryption, like to authenticate that the data originates from a known sender.
4. To protect network data by setting up circuits using IPsec tunnelling in which all data is being sent between the two endpoints is encrypted, as with a Virtual Private Network (VPN) connection.

2 APPLICATIONS OF IPsec

Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for travelling employees and telecommuters.

Establishing extranet and intranet connectivity with partners: IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.

Enhancing electronic commerce security: Even though some Web and electronic Commerce applications have built-in security protocols; the use of IPsec enhances that security. IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

3. IP SECURITY SCENARIO

IMP QS (question)-08M

Figure 1 is a typical scenario of IPsec usage.

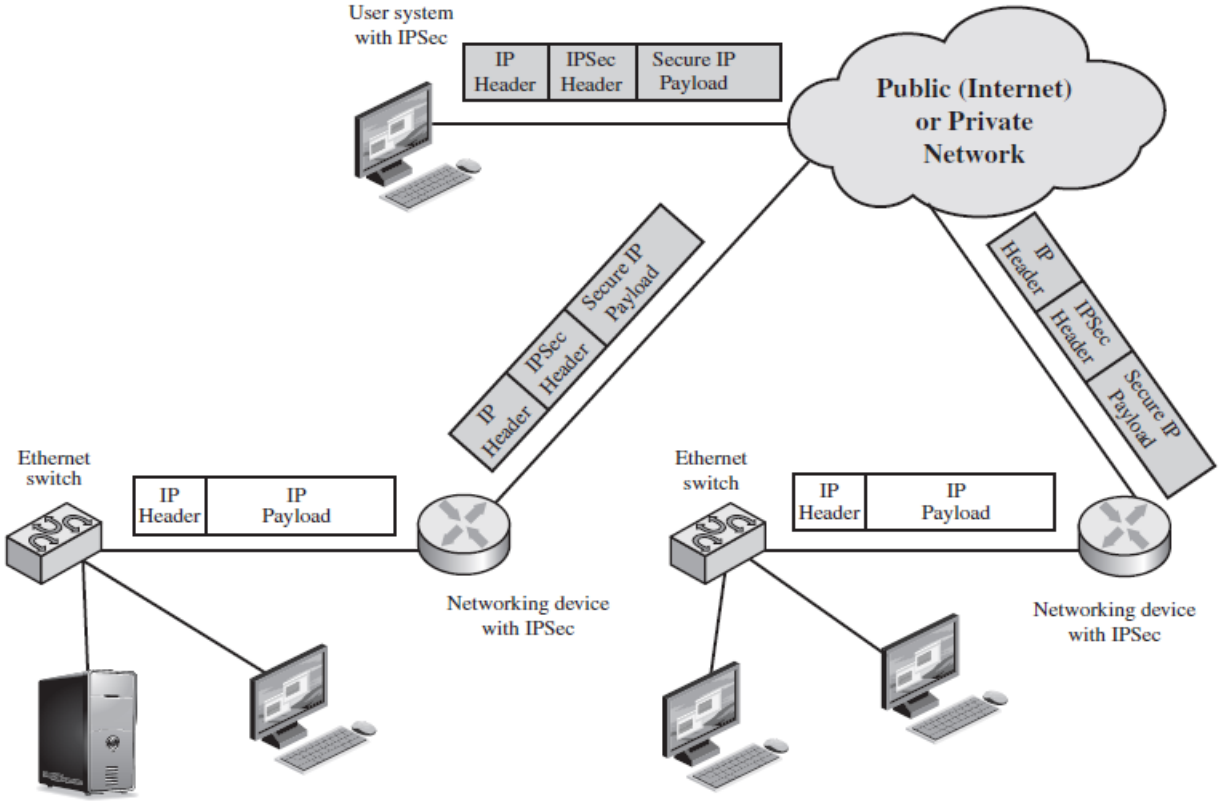


Figure 1: IP Security Scenario

- An organization maintains LANs at dispersed locations. Non secure IP traffic is conducted on each LAN.
- For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.
- These protocols operate in networking devices, such as a router or firewall, that Connect each LAN to the outside world.

- The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN.
- Secure transmission is also possible with individual users who dial into the WAN. Such user workstations must implement the IPsec protocols to provide security.

4 BENEFITS OF IPSEC

IMP QS (question)-04M

Some of the benefits of IPsec:

- Then IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications.
- There is no need to change software on a user or server system when IPsec is implemented in the firewall or router.
- Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users.
- IPsec can provide security for individual users if needed.

5 IPsec DOCUMENTS

IMP QS (question)-05M

IPsec encompasses three functional areas: authentication, confidentiality, and key management.

The documents can be categorized into the following groups.

- **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology. The current specification is RFC 4301, Security Architecture for the Internet Protocol.
- **Authentication Header (AH):** AH is an extension header to provide message Authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP (An Encapsulating Security

Payload (ESP) is a protocol within the IPSec for providing authentication, integrity and confidentially of network packets data/payload in IPv4 and IPv6 networks), the use of AH is deprecated.

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

6 IPsec SERVICES

- IPsec provides security services at the IP layer by enabling a system to select required Security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.
- **Two protocols** are used to provide security: **an authentication protocol designated by** the header of the protocol. **Authentication Header (AH); and a combined encryption/ authentication protocol designated by** the format of the packet for that protocol, Encapsulating Security Payload (ESP).
- RFC 4301 lists the following services:
  - Access control.
  - Connectionless integrity.
  - Data origin authentication.
  - Rejection of replayed packets (a form of partial sequence integrity).
  - Confidentiality (encryption).
  - Limited traffic flow confidentiality.

NOTE:-

RFC 4301 – Security architecture for the internet protocol

RFC 4302-IP authentication header

RFC 4303- ESP Encapsulating payload

7 TRANSPORT AND TUNNEL MODES IMP QS (question)-9M

Both Authentication Headers (AH) and Encapsulating Security Payload (ESP). support two modes of use: transport and tunnel mode.

7.1 Transport Mode

- Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.
- **ESP in transport mode** encrypts and optionally authenticates the IP payload but not the IP header.
- **AH in transport mode** authenticates the IP payload and selected portions of the IP header.

TABLE 1 SUMMARIZES TRANSPORT AND TUNNEL MODE FUNCTIONALITY.

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

Table 1: Tunnel Mode and Transport Mode Functionality

7.2 Tunnel Mode

- Tunnel mode provides protection to the entire IP packet.
- Tunnel mode is used when one or both ends of a security association (SA) are a security gateway, such as a firewall or router that implements IPsec.
- **ESP in tunnel mode** encrypts and optionally authenticates the entire inner IP packet, including the inner IP header.
- **AH in tunnel mode** authenticates the entire inner IP packet and selected portions of the outer IP header.

8 IP SECURITY POLICY

IMP QS (question)-04M

Fundamental to the operation of IPsec is the concept of a security policy applied to each IP packet that transits from a source to a destination.

IPsec policy is determined primarily by the interaction of two databases, the **security association database (SAD)** and the **security policy database (SPD)**.

This section provides an overview of these two databases and then summarizes their use during IPsec operation. **Figure 2** illustrates the relevant relationships.

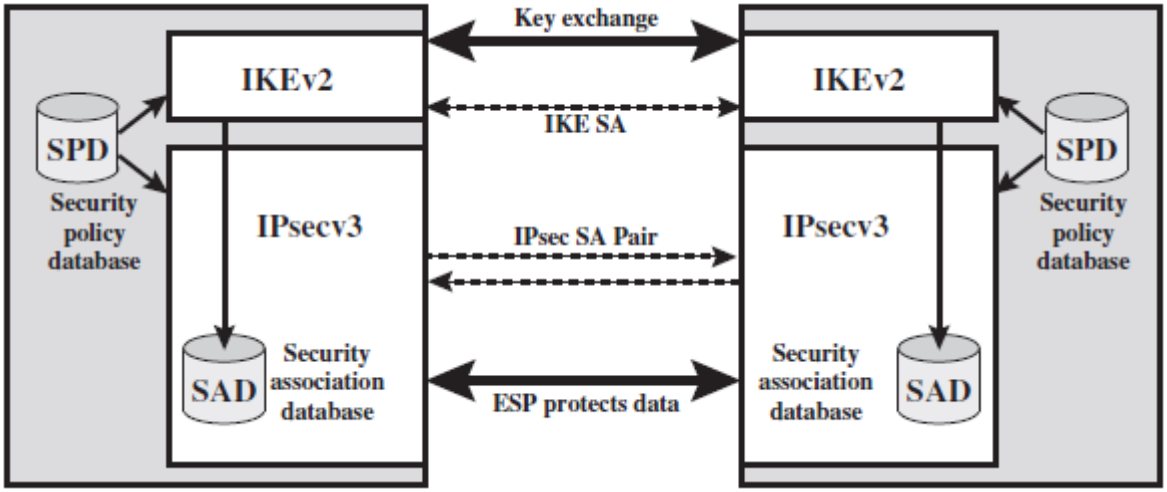


Figure IPsec Architecture

8.1 SECURITY ASSOCIATIONS

IMP QS (question)-03M

- A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA).
- An association is a one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it.
- If a peer relationship is needed for two-way secure exchange, then two security associations are required.
- A security association is uniquely identified by three parameters.
  - **Security Parameters Index (SPI):** A 32-bit unsigned integer assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.

- **IP Destination Address:** This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
- **Security Protocol Identifier:** This field from the outer IP header indicates whether the association is an AH or ESP security association.

➢ Hence, in any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

8.2 SECURITY ASSOCIATION DATABASE

IMP QS (question)-05M

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

1. **Security Parameter Index:** A 32-bit value selected by the receiving end of an SA to uniquely identify the SA.
2. **Sequence Number Counter:** A 32-bit value used to generate the Sequence Number field in AH or ESP headers.
3. **Sequence Counter Overflow:** A flag indicating whether overflow of the Sequence Number
4. **Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is a replay. (Required for all implementations).
5. **AH Information:** Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations).
6. **ESP Information:** Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations).
7. **Lifetime of this Security Association:** A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur (required for all implementations).
8. **IPsec Protocol Mode:** Tunnel, transport, or wildcard.
9. **Path MTU:** Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations).

8.3 SECURITY POLICY DATABASE IMP QS (question)-06M

- The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPsec) is the nominal Security Policy Database (SPD).
- In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic.
- In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.
- Each SPD entry is defined by a set of IP and upper-layer protocol field values, called selectors.
- The following selectors determine an SPD entry:
  - **Remote IP Address:** This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (e.g., behind a firewall).
  - **Local IP Address:** This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (e.g., behind a firewall).
  - **Next Layer Protocol:** The IP protocol header (IPv4, IPv6, or IPv6 Extension) includes a field (Protocol for IPv4, Next Header for IPv6 or IPv6 Extension) that designates the protocol operating over IP..
  - **Name:** A user identifier from the operating system. This is not a field in the IP or upper-layer headers but is available if IPsec is running on the same operating system as the user.
  - **Local and Remote Ports:** These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

9 IP TRAFFIC PROCESSING

IMP QS (question)-10M

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

OUTBOUND PACKETS

- Figure 3 highlights the main elements of IPsec processing for outbound traffic.
- A block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body. Then the following steps occur:-

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

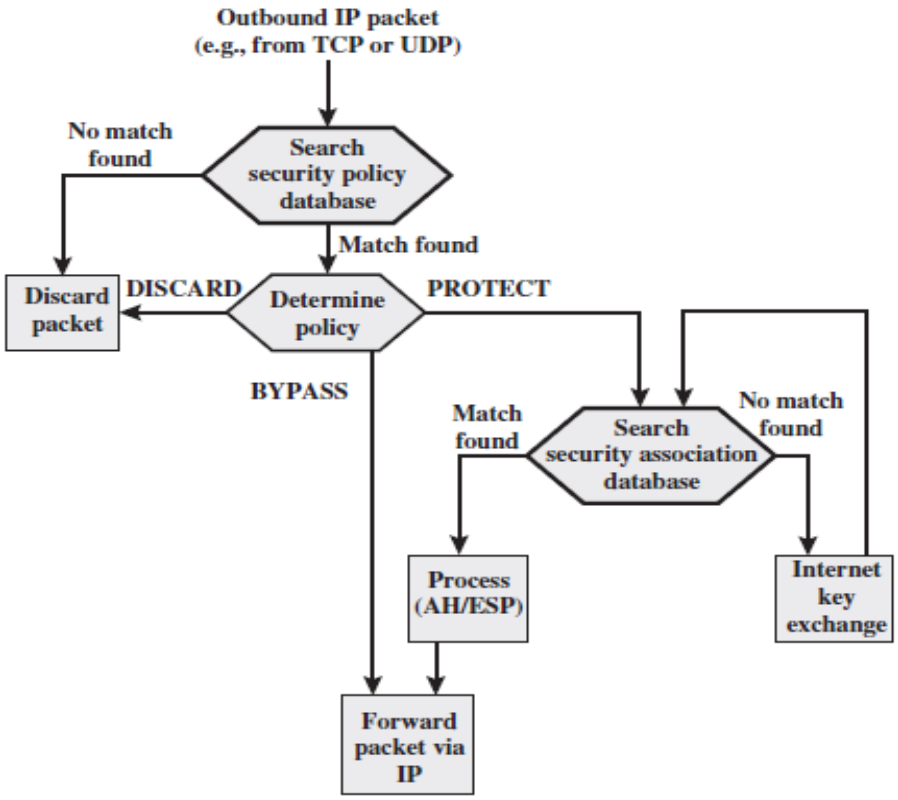


Figure 3 Processing Model for Outbound Packets

4. If the policy is PROTECT, then a search is made of the SAD for a matching entry. If no entry is found, then IKE is invoked to create an SA with the appropriate keys and an entry is made in the SA.
5. The matching entry in the SAD determines the processing for this packet. Encryption, authentication, or both can be performed, and either transport or tunnel mode can be used. The packet is then forwarded to the network for transmission.

INBOUND PACKETS

Figure 4 highlights the main elements of IPsec processing for inbound traffic. An incoming IP packet triggers the IPsec processing. The following steps occur:

1. IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6).

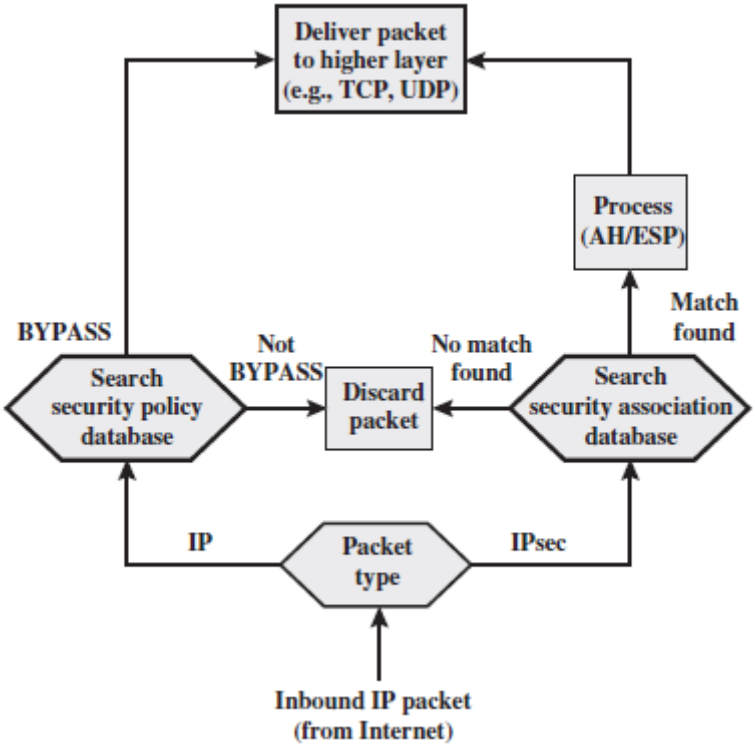


Figure 4 Processing Model for Inbound Packets

2. If the packet is unsecured, IPsec searches the SPD for a match to this packet. If the first matching entry has a policy of BYPASS, the IP header is processed and

stripped off and the packet body is delivered to the next higher layer, such as TCP. If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.

3. For a secured packet, IPsec searches the SAD. If no match is found, the packet is discarded. Otherwise, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP.

10 ENCAPSULATING SECURITY PAYLOAD IMP QS (question)-08M

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

10.1 ESP Format IMP QS (question)-08M

Figure 5(a) shows the top-level format of an ESP packet. It contains the following Fields.

1. **Security Parameters Index (32 bits):** Identifies a security association.
2. **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
3. **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
4. **Padding (0–255 bytes):** the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
5. **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.

6. **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6, or an upper-layer protocol such as TCP).
7. **Integrity Check Value (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

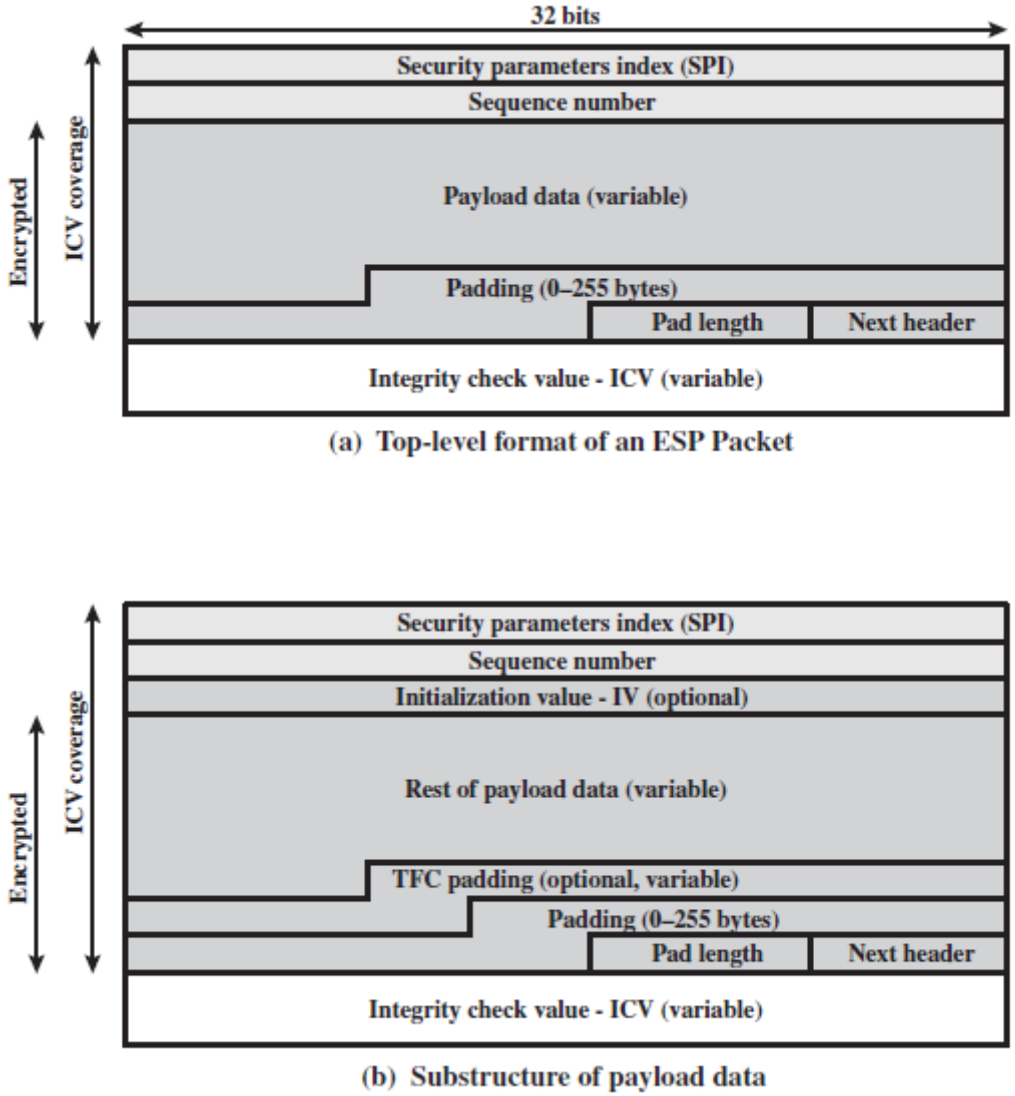


Figure 5 ESP Packet Format

Note: **Anti-replay** is a sub-protocol of IPsec that is part of Internet Engineering Task Force (IETF). The main goal of **anti-replay** is to avoid hackers injecting or making changes in packets that travel from a source to a destination.

Figure 5(b).

Two additional fields may be present in the payload **figure 5(b).** an **initialization value (iv)**, or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for esp. if tunnel mode is being used, then the IPsec implementation may add **traffic flow confidentiality (TFC)** padding after the payload data and before the padding field, as explained subsequently.

11 ANTI – REPLY SERVICE

IMP QS (question)-06M

- A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence.
- The sequence number field is used to thwart the reply attack.

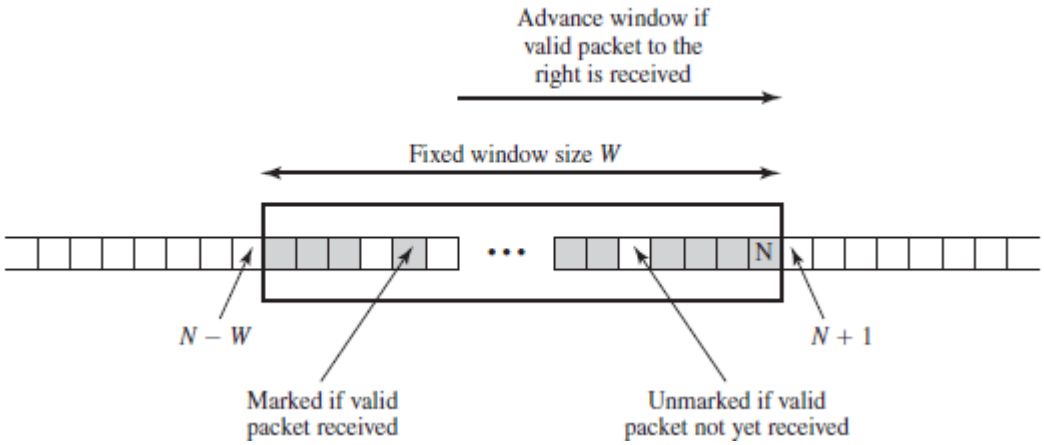


Figure 6 Anti-replay Mechanism

FIGURE 6:–

- The sequence number is set to zero with a new SA (Security Associations) established
- The number is incremented by 1 for each packet sent on the SA.
- The SA is terminated or negotiated with a new key is $N=2^{32}-1$
- A window of size W is implemented in order for IP packed to be delivered in reliable manner (with a default of w=64).
- The right edge of the window represents the highest sequence number, *N*, so far received for a valid packet.

- For any packet with a sequence number in the range from $N - W + 1$ to N that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked (Figure 6). Inbound processing proceeds as follows when a packet is received:
  1. If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
  2. If the received packet is to the right of the window and is new, the MAC is Checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
  3. If the received packet is to the left of the window or if authentication fails, the packet is discarded; this is an auditable event.

12 TRANSPORT AND TUNNEL MODES

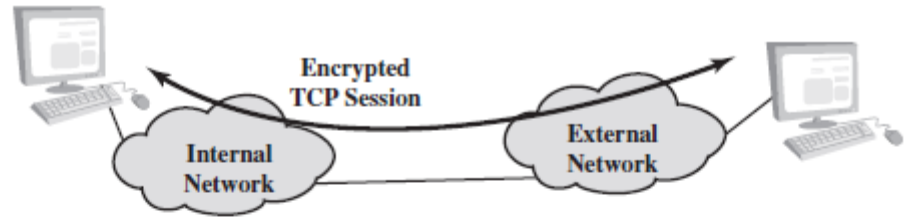
IMP QS (question)-08M

FIGURE 7 SHOWS

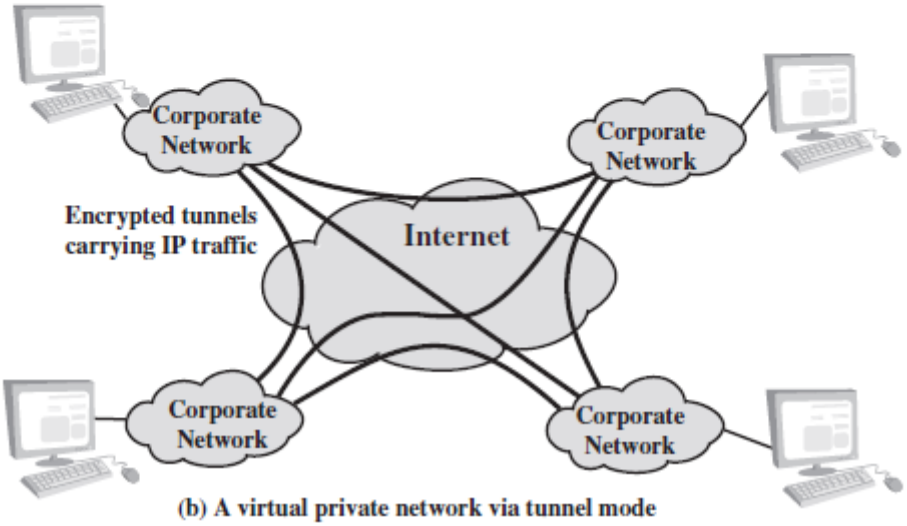
- Two ways in which the IPsec ESP service can be used.
- In the **upper part of the figure**, encryption (and optionally authentication) is provided directly between two hosts.
- **Figure 7(b)** shows how tunnel mode operation can be used to set up a **virtual private network**.

In this example

- An organization has four private networks interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts.
- By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability.



(a) Transport-level security



(b) A virtual private network via tunnel mode

Figure 7 Transport-Modes versus Tunnel-Mode Encryption

- The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA.

➢ The scope of ESP for the two modes. The considerations are somewhat different for IPv4 and IPv6. We use the packet formats of **Figure 8(a)** as a starting point.

13 Transport Mode ESP

IMP QS (question)-08M

Transport mode ESP is used to encrypt and optionally authenticate the data carried by IP (e.g., a TCP segment), as shown in Figure 8(b.)

AS SHOWN IN FIGURE 8(B.)

IPv4:-

- For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP), and

an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet.

- If authentication is selected, the ESP Authentication Data field is added after the ESP trailer.
- The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the cipher text plus the ESP header.

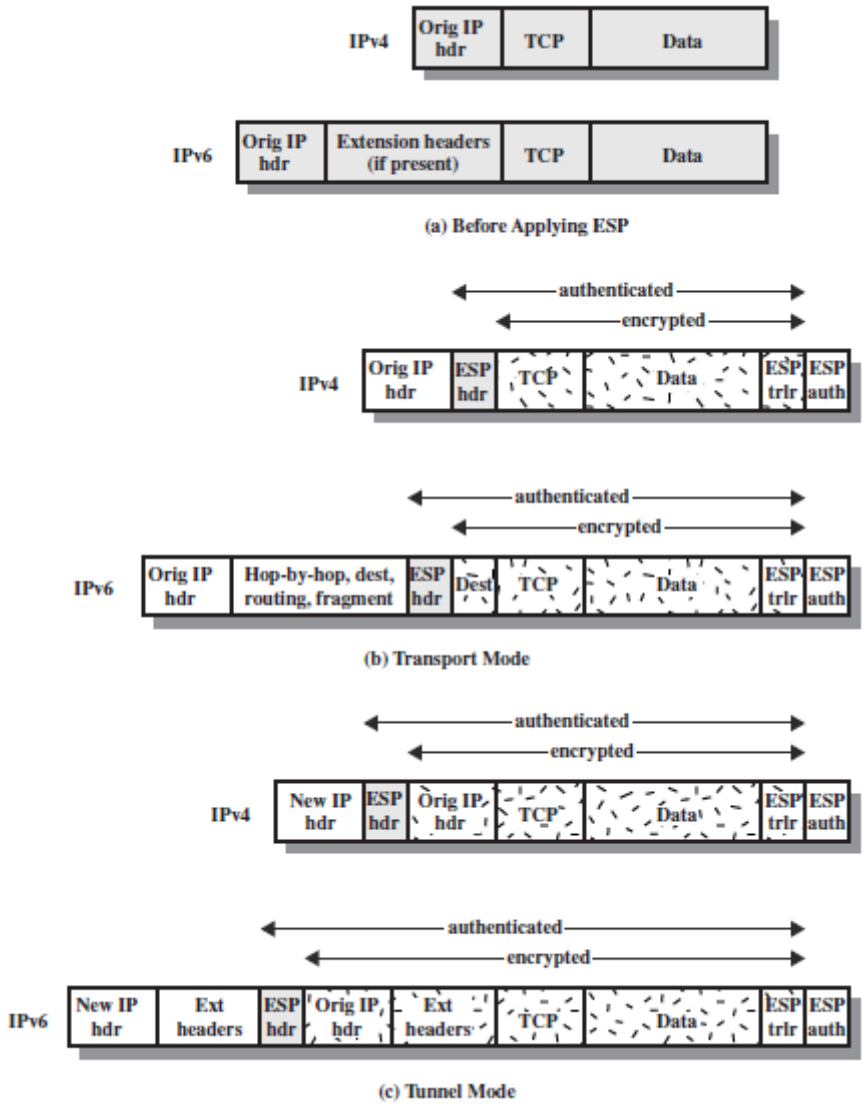


Figure 8 Scopes of ESP Encryption and Authentication

IPv6:-

- In the context of IPv6, ESP is viewed as an end-to-end payload; that is, it is not examined or processed by intermediate routers.
- Therefore, the ESP header appears after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers.

- The destination options extension header could appear before or after the ESP header, depending on the semantics desired.
- For IPv6, encryption covers the entire transport-level segment plus the ESP trailer plus the destination options extension header if it occurs after the ESP header.
- Again, authentication covers the cipher text plus the ESP header.

TRANSPORT MODE OPERATION MAY BE SUMMARIZED AS FOLLOWS.

1. At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its cipher text to form the IP packet for transmission. Authentication is added if this option is selected.
2. The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but does not need to examine the cipher text.
3. The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment.

ADVANTAGES AND DRAWBACKS

- Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application.
- One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

14 Tunnel Mode ESP

IMP QS (question)-06M

- Tunnel mode ESP is used to encrypt an entire IP packet **(FIGURE 8C).**
- For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted.
- This method can be used to counter traffic analysis. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header.

- Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus cipher text plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis.

FIGURE 9 SHOWS THE PROTOCOL ARCHITECTURE FOR THE TWO MODES.

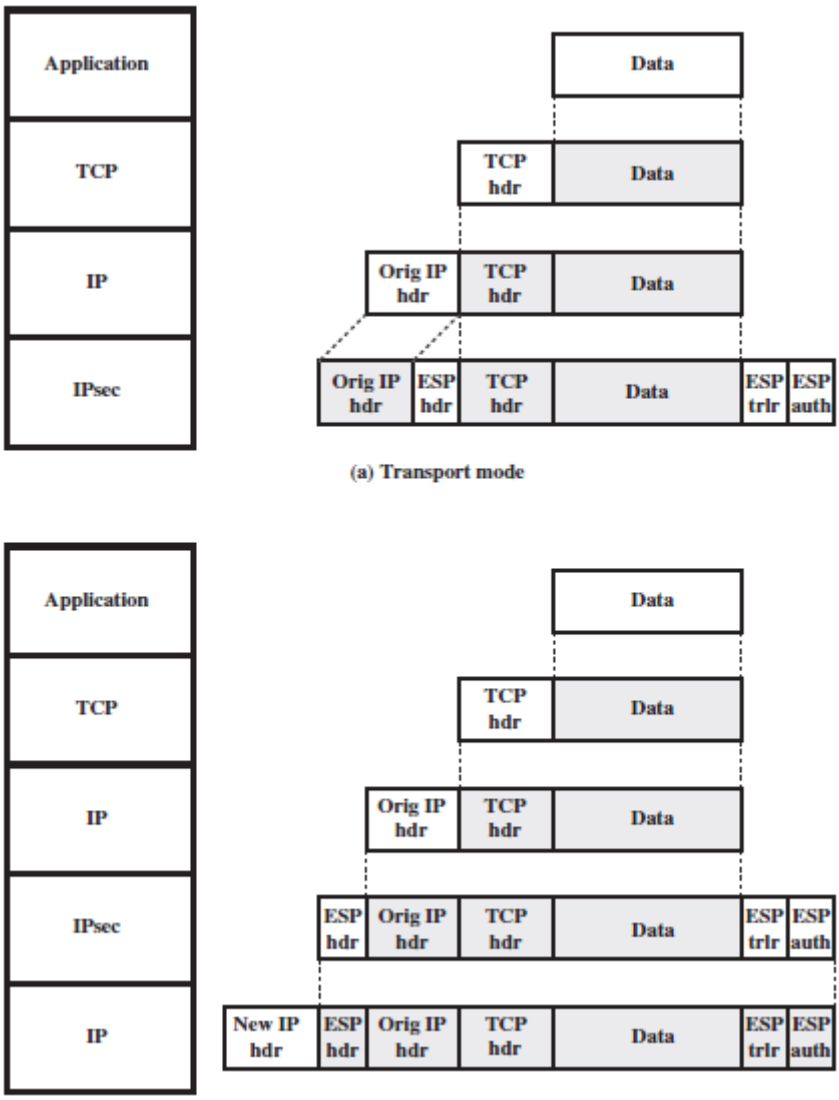


Figure 9 Protocol Operations for ESP

- Consider a case in which an external host wishes to communicate with a host on an internal network protected by a firewall, and in which ESP is implemented in the external host and the firewalls.
- The following steps occur for transfer of a transport-layer segment from the external host to the internal host.

1. The source prepares an inner IP packet with a destination address of the target internal host. This packet is prefixed by an ESP header; then the packet and ESP

trailer are encrypted and Authentication Data may be added. The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet.

2. The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the cipher text.
3. The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network.
4. The inner packet is routed through zero or more routers in the internal network to the destination host.

15 BASIC COMBINATIONS OF SECURITY ASSOCIATIONS IMP QS (question)-08M

- The IPsec Architecture document lists four examples of combinations of SAs that must be supported by compliant IPsec hosts (e.g., workstation, server) or security gateways (e.g., firewall, router).

THESE ARE ILLUSTRATED IN FIGURE 10.

- The lower part of each case in the figure represents the physical connectivity of the elements; the upper part represents logical connectivity via one or more nested SAs.
- Each SA can be either AH or ESP.
- For host-to-host SAs, the mode may be either transport or tunnel; otherwise it must be tunnel mode.

CASES

- **Case 1**. All security is provided between end systems that implement IPsec.

For any two end systems to communicate via an SA, they must share the appropriate secret keys. Among the possible combinations are

1. AH in transport mode
2. ESP in transport mode

3. ESP followed by AH in transport mode (an ESP SA inside an AH SA)
4. Any one of a, b, or c inside an AH or ESP in tunnel mode

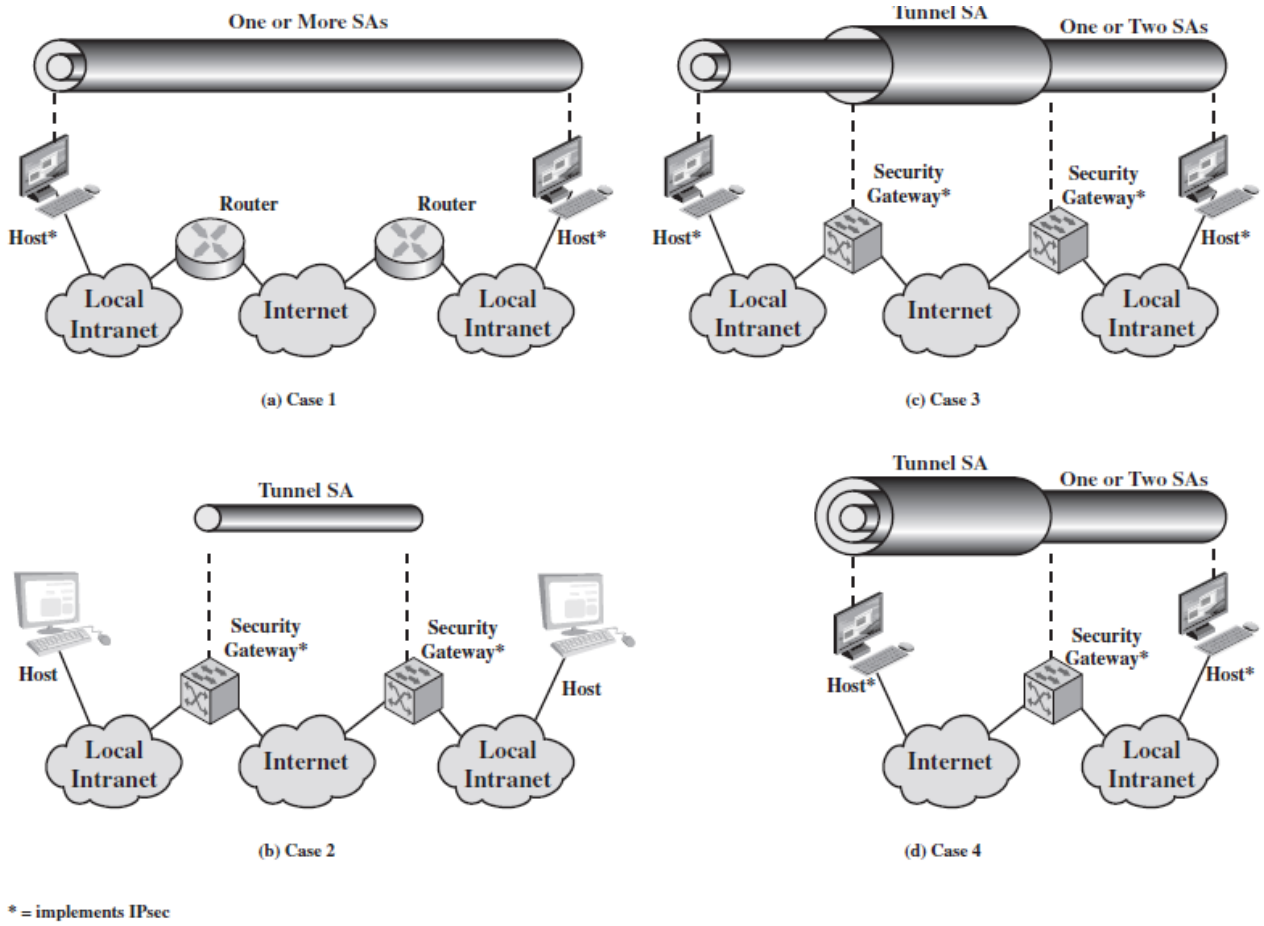


Figure 10 Basic Combinations of Security Associations

- **Case 2.** Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document specifies that only a single tunnel SA is needed for this case. The tunnel could support AH, ESP, or ESP with the authentication option. Nested tunnels are not required, because the IPsec services apply to the entire inner packet.
- **Case 3**. This builds on case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems. When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality. Individual hosts can implement

any additional IPsec services required for given applications or given users by means of end-to end SAs.

- **Case 4.** This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Only tunnel mode is required between the remote host and the firewall. As in case 1, one or two SAs may be used between the remote host and the local host.

16 IKE V2 EXCHANGES

IMP QS (question)-06M

- The IKEv2 protocol involves the exchange of messages in pairs.

THE FIRST TWO PAIRS OF EXCHANGES ARE REFERRED TO AS THE INITIAL EXCHANGES FIGURE 12(A)

- In the first exchange, the two peers exchange information concerning cryptographic algorithms and other security parameters they are willing to use along with nonce's and Diffie-Hellman (DH) values. The result of this exchange is to set up a special SA called the IKE SA.
- In the second exchange, the two parties authenticate one another and set up a first IPsec SA to be placed in the SADB and used for protecting ordinary (i.e. non-IKE) communications between the peers. Thus, four messages are needed to establish the first SA for general use.

FIGURE 12(B) AND FIGURE 12(C)

- The CREATE_CHILD_SA exchange can be used to establish further SAs for protecting traffic. The informational exchange is used to exchange management information, IKEv2 error messages, and other notifications.

NOTE

PAYLOAD:-

When data is sent over the Internet, each unit transmitted includes both header information and the actual data being sent. The header identifies the source and destination of the packet, while the actual data is referred to as the **payload**. Therefore, the **payload** is the only data received by the destination system.

Nonce's:- a **nonce** is an arbitrary number that can be used just once in a cryptographic communication.

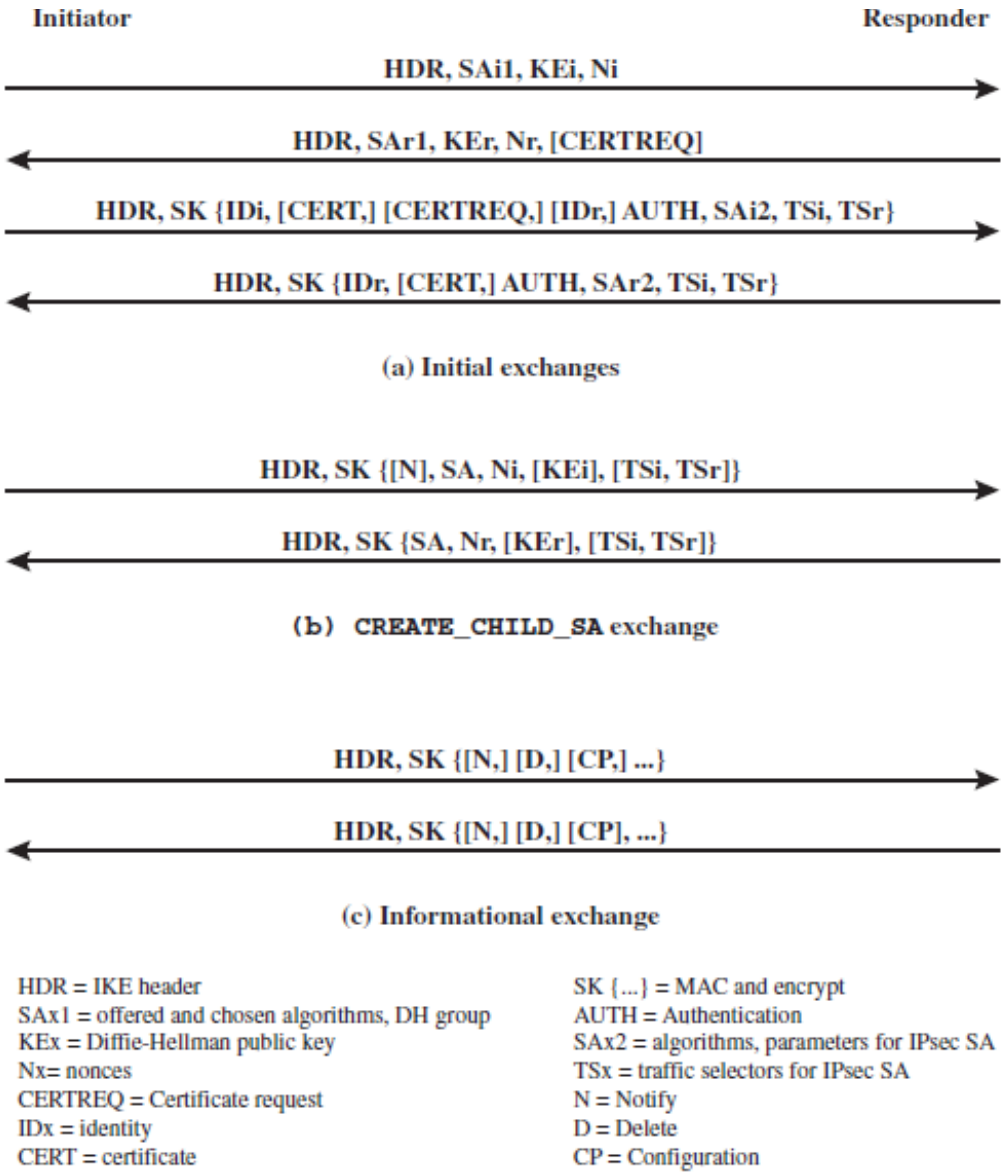


Figure 12 IKEv2 Exchanges

17 IKE HEADER FORMAT

IMP QS (question)-06M

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

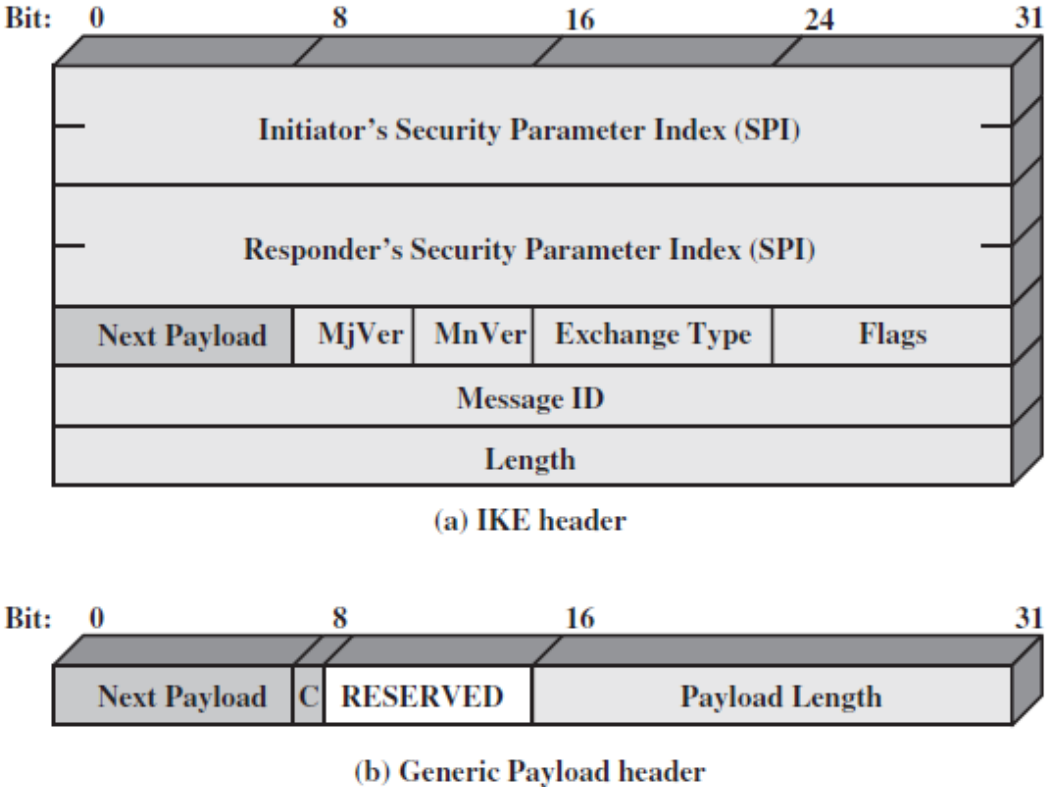


Figure 13 IKE Formats

Figure 13(A) shows the header format for an IKE message. It consists of the following fields.

1. **Initiator SPI (64 bits):** A value chosen by the initiator to identify a unique IKE security association (SA).
2. **Responder SPI (64 bits):** A value chosen by the responder to identify a unique IKE SA.
3. **Next Payload (8 bits):** Indicates the type of the first payload in the message.
4. **Major Version (4 bits):** Indicates major version of IKE in use.
5. **Minor Version (4 bits):** Indicates minor version in use.
6. **Exchange Type (8 bits):** Indicates the type of exchange.
7. **Flags (8 bits):** Indicates specific options set for this IKE exchange. Three bits are defined so far. The initiator bit indicates whether this packet is sent by the SA initiator. The version bit indicates whether the transmitter is capable of using a higher major version number than the one currently indicated. The response bit indicates whether this is a response to a message containing the same message ID.

8. **Message ID (32 bits):** Used to control retransmission of lost packets and matching of requests and responses.
9. **Length (32 bits):** Length of total message (header plus all payloads) in octets.

17.1 IKE PAYLOAD TYPES

IMP QS (question)-06M

- All IKE payloads begin with the same generic payload header shown in Figure **Figure 13(B).**
- The Next Payload field has a value of 0 if this is the last payload in the message; otherwise its value is the type of the next payload.
- The Payload Length field indicates the length in octets of this payload, including the generic payload header.
- These elements are formatted as substructures within the payload as follows.
    1. **Proposal:** This substructure includes a proposal number, a protocol ID (AH, ESP, or IKE), an indicator of the number of transforms, and then a transform substructure. If more than one protocol is to be included in a proposal, then there is a subsequent proposal substructure with the same proposal number.
    2. **Transform:** Different protocols support different transform types. The transforms are used primarily to define cryptographic algorithms to be used with a particular protocol.
    3. **Attribute:** Each transform may include attributes that modify or complete the specification of the transform. An example is key length.

IKE PAYLOAD TYPES

- The **Key Exchange payload** can be used for a variety of key exchange techniques, including Oakley, Diffie-Hellman, and the RSA-based key exchange used by PGP.
- The **Identification payload** is used to determine the identity of communicating peers and may be used for determining authenticity of information. Typically the ID Data field will contain an IPv4 or IPv6 address.
- The **Certificate payload** transfers a public-key certificate.

Type	Parameters
Security Association	Proposals
Key Exchange	DH Group #, Key Exchange Data
Identification	ID Type, ID Data
Certificate	Cert Encoding, Certificate Data
Certificate Request	Cert Encoding, Certification Authority
Authentication	Auth Method, Authentication Data
Nonce	Nonce Data
Notify	Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data
Delete	Protocol-ID, SPI Size, # of SPIs, SPI (one or more)
Vendor ID	Vendor ID
Traffic Selector	Number of TSs, Traffic Selectors
Encrypted	IV, Encrypted IKE payloads, Padding, Pad Length, ICV
Configuration	CFG Type, Configuration Attributes
Extensible Authentication Protocol	EAP Message

Table 3 IKE Payload Types

- At any point in an IKE exchange, the sender may include a **Certificate Request** payload to request the certificate of the other communicating entity.
- The **Authentication** payload contains data used for message authentication purposes.
- The **Nonce** payload contains random data used to guarantee liveness during an exchange and to protect against replay attacks.
- The **Notify** payload contains either error or status information associated with this SA or this SA negotiation.
- The **Delete** payload indicates one or more SAs that the sender has deleted from its database and that therefore are no longer valid.
- The **Vendor ID** payload contains a vendor-defined constant.
- The **Traffic Selector** payload allows peers to identify packet flows for processing by IPsec services.
- The **Encrypted** payload contains other payloads in encrypted form.
- The **Configuration** payload is used to exchange configuration information between IKE peers.
- The **Extensible Authentication Protocol (EAP)** payload allows IKE SAs to be authenticated using EAP,

MODULE-4

Cyber network security concepts: Security Architecture, Antipattern: signature based malware detection versus polymorphic threads, document driven certification and accreditation, policy driven security certifications. Refactored solution: reputational, behavioural and entropy based malware detection.

The problems: cyber antipatterns concept, forces in cyber antipatterns, cyber anti pattern templates, cyber security Antipattern catalog.

CHAPTER -1 CYBER NETWORK SECURITY CONCEPTS

1. SECURITY ARCHITECTURE

IMP QS (question)-04M

- The cyber security crisis is a fundamental failure of architecture. Many of the Networked technologies we depend upon daily have no effective security whatsoever.
- The architecture of the Internet and the vast majority of deployed software create significant opportunities for malicious exploitation.
- It is worth stating that if infrastructure and software technologies were engineered properly, they would be built to withstand known and manage unknown risks, and they would be significantly more secure than current-day technologies.
- the Zachman Framework for Enterprise Architecture and applies it to securing enterprises.
- The Zachman Framework is a powerful intellectual tool that enables complex organizations to describe themselves, including their mission, business, and information technology (IT) assets. With this self-knowledge comes awareness of risks and mitigations, and ways of engineering security into solutions from inception.
- The Zachman Framework serves as an overarching structure that organizes the problem-solving patterns catalog.

2. ANTIPATTERN: SIGNATURE-BASED MALWARE DETECTION VERSUS POLYMORPHIC THREATS

IMP QS (question)-06M

- The conventional wisdom is that all systems with up-to-date antivirus signatures will be safe.
- However, many popular antivirus solutions are nearly obsolete, with many missing the majority of new malware.
- Current signature-based antivirus engines miss 30 percent to 70 percent of malicious code, and nearly 100 percent of zero day infections, which, by definition, are unreported exploits.
- Malicious signature growth is exploding from 5 new ones per day in 2000 to 1,500 per day in 2007 and more than 15,000 per day in 2009, according to Symantec (from a 2010 conference briefing on reputational anti-malware), which is an average of 200 percent to 300 percent cumulative growth per year.
- Malware variability has grown so rapidly that signature-based detection is rapidly becoming obsolete.
- The proliferation of malware signatures is exploding primarily due to polymorphic malware techniques.
    - For example, hash functions used by signature-based detectors yield very different values with only slight changes to a malicious file. Changing a string literal in the file is sufficient to trigger a false negative.
    - Other polymorphic techniques include varying character encodings, encryption, and random values in the files.
- One interesting online application from VirusTotal.com runs more than 30 antivirus programs on each file that any Internet user can submit. You can witness just how haphazard antivirus tests are.

3. ANTIPATTERN: DOCUMENT-DRIVEN CERTIFICATION AND ACCREDITATION

IMP QS (question)-06M

- Some of the most flagrant antipatterns involve the IT security industry itself.
- Assessment and Authorization (A&A), formerly called Certification and Accreditation (C&A), has attracted much public criticism because it has a

reputation as a paper-driven process that does not secure systems from real threats.

- A&A is the process of assuring the information security of systems before they are deployed.
- Certification is an assessment and testing phase that identifies and confirms vulnerabilities.
- Accreditation is an executive approval process that accepts risks discovered during certification.
- Although A&A is formalized in government organizations, it is also widely practiced in industry. For example, payment card industry (PCI) standards require businesses that process credit cards (in other words, virtually all retail companies), to conduct penetration tests and other formal assessments.
- Refactored solutions for this Antipattern can be derived from the practical security testing and investigation techniques.

4. ANTIPATTERN: POLICY-DRIVEN SECURITY CERTIFICATIONS DO NOT ADDRESS THE THREAT.

IMP QS (question)-06M

- The gold standard of professional security certifications is the Certified Information System Security Professional (CISSP). It is an entirely paper-based qualification, requiring a great deal of memorization in 10 diverse security domains, such as physical security, communications security, and systems security.
- CISSP is required by the U.S. Department of Defense (DoD) for both management and technical security workers, and demanded in the job market.
- The one report states clearly that "the current professional certification regime is not merely inadequate; it creates a dangerously false sense of security" with an overemphasis on security compliance on paper versus combating threats.
- Many people in the cyber security community view this finding as controversial because their careers, reputations, and credentials are invested in security compliance policies and procedures. This is the industry that drives A&A, risk management, security controls compliance, and other labor-intensive security activities.

- Unfortunately, for most professionals, it is much easier to turn a highly technical person into a policy person, whereas it is very difficult (or impossible) to turn a policy person into a highly technical one. It is a one-way street.

5. REFACTORED SOLUTION: REPUTATIONAL, BEHAVIOURAL AND ENTROPY BASED MALWARE DETECTION.

IMP QS (question)-06M

Vendors are developing innovative techniques that can detect zero day and polymorphic malware. Several promising approaches for the future include:

- Symantec is harnessing a 100M+ global customer base to identify potential malware signatures. The technique, called reputation-based signatures, is able to identify 240 million new malware signatures by comparing binaries across millions of systems for anomalous variations.
- Fire Eye has created a behavioral intrusion detection system (IDS) that uses elements of honey pots and forensics to automatically identify malicious content as it flows across corporate networks. Behavioral IDS techniques simulate the execution of sniffed content in a virtual machine, which then observes resulting configuration changes, such as changes in registry settings, services, and the file system. There are other emerging behavioural antivirus products, for example, from ThreatFire.com.
- An emerging field of research called entropy-based malware detection looks for mathematical similarity to known malware signatures. Hash functions that are used by most antivirus programs detect subtle differences between a file and its known hash. Minor changes to a file, such as modification of strings or encodings can cause a hash match to fail. Entropy-based matching uses mathematical functions that measure similarity rather than differences. If a suspicious file nearly matches the same entropy measure as malware, there is a high likelihood that the malware is present.

1. ANTIPATTERNS CONCEPT

IMP QS (question)-06M

Design forces are the competing concerns, priorities, and technical factors that influence the choice of solutions. In antipatterns, there are two solutions: the Antipattern solution and the Refactored solution.

1. An **Antipattern solution** represents a commonplace dysfunctional situation or configuration. The Antipattern solution may be the result of multiple choices over an extended system lifecycle, or it may have evolved inadvertently. Every solution or design choice yields benefits and consequences.

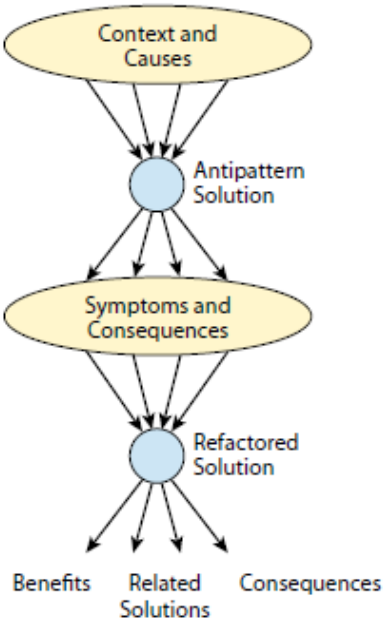


Figure 1: Antipattern concept

2. The **Refactored solution** results from reconsideration of the design forces and the selection of a more effective solution (see Figure 1). The Refactored solution yields benefits that outweigh its consequences. There may also be related solutions or variations that also resolve the design forces beneficially.

2. FORCES IN CYBER ANTIPATTERNS

IMP QS (question)-08M

The major types of forces in antipatterns include primal, horizontal, and vertical forces.

1. Primal forces are pervasive design forces present in almost every design decision.
2. Horizontal forces are forces that can apply in all domains.
3. Vertical forces are domain or system specific design forces.

The primal design forces in the cyber security domain include:

- Management of functionality
- Management of confidentiality
- Management of integrity
- Management of availability

You probably recognize this formulation as the famous **Confidentiality, Integrity, and Availability (CIA)** from IT security.

The **functionality** design force is added because it drives the other forces. Systems are granted accreditation with respect to a defined level of **functionality**. **Functionality** is tested and verified by the developers prior to security testing.

- **Confidentiality**
  - Is the protection of information on the system.
  - In most current systems, the information is the primary resource being secured and the sensitivity of the information defines the level of risk and security priority for each system or database element.
- **Integrity**
  - Is protection of the coherence of the data and system metadata (for example, configuration).
  - The significant threat of damage to data can be very costly to remediate.
  - This threat affects even the most sensitive systems that have very limited connectivity to external networks because data, e-mail, and removable media with malware can migrate to those systems through normal and erroneous operations.
- **Availability**
  - is the continuous readiness of the system to execute its functionality in response to users and other systems' requests, and the ability to continually access its data.
  - Availability is an aspect of the more general concept of Quality of Service (QOS).

- QOS is a service-level requirement for the system, such as guarantees of throughput bandwidth or user request response time.

To assure the security of a system, testing is a necessary evil. A test is a comparison of two things, such as a security specification and a system implementation. As a result, requirements for functionality, confidentiality, integrity, and availability should be clearly identified in the system documentation.

3. CYBER ANTIPATTERN TEMPLATES

IMP QS (question)-10M

The **two templates** include the **micro-Antipattern template and the full cyber Antipattern template**. You use **the micro-template** for simpler patterns in which detailed explanation is not necessary.

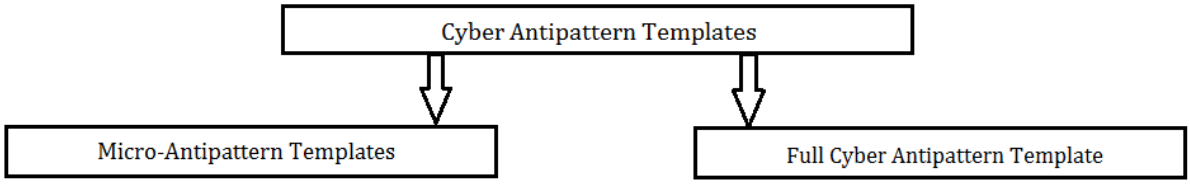


Fig2: Cyber Antipattern templates types

You use the **full template** for the more complex and more important antipatterns, providing a much more complete coverage of the problem and the solution.

3.1 MICRO-ANTIPATTERN TEMPLATES

IMP QS (question)-04M

The micro-Antipattern It is a flexible and informal way to present antipatterns. The components of a micro-Antipattern template are:

1. **Name:** The name of the micro-Antipattern is usually a pejorative term, suggesting the negative consequences of the antipatterns presence.
2. **Antipattern Problem:** The problem section summarizes the micro-antipatterns symptoms, consequences, and characterization.
3. **Refactored Solution:** The solution section summarizes alternative ways To resolve the Antipattern design forces with improved benefits.

Because the micro-Antipattern template is so simple, it can be presented without the formality of templates at all.

3.2 FULL CYBER ANTIPATTERN TEMPLATE IMP QS (question)-08M

- The full cyber Antipattern template has two main parts: a header and a body.
- The header gives a quick sense of the Antipattern and the solution, inviting the reader to dive deeper.
- The body sections contain the pattern details.
- The full cyber Antipattern template It allows for a more structured and comprehensive definition with additional Antipattern attributes.
- Many of the attributes are considered optional, depending on the particulars of the Antipattern concerned.
- The heading fields in the full cyber Antipattern template are
  - **Antipattern Name:** The name is a unique pejorative noun phrase. The intent is to make this Antipattern a well-known phenomenon, easily recognizable, with an organizational reputation as an important security gap.
  - **Also Known As:** Many antipatterns are known by various names across different organizations. Some known names or analogous names from different domains are listed here. A given organization might want to adopt a name from this list if their members find it suitable.
  - **Refactored Solution Names:** One or more names of alternative solutions are listed here. The purpose is to give the reader a sense of the direction that this pattern write-up is heading toward and to promote a common terminology for the solution identity associated with the Antipattern.
  - **Unbalanced Primal Forces:** This field lists the primal design forces that are poorly resolved by this Antipattern.
  - **Anecdotal Evidence:** These are some quips that characterize this Antipattern. These phrases are sometimes heard when the Antipattern is present and in the early recognition of it.
- The body fields in the full cyber Antipattern template are
  - **Background:** This optional field provides contextual explanations that are potentially useful or of general interest but are not central to the Antipattern and its Refactored solution.
  - **Antipattern Solution:** This field defines the Antipattern solution through diagrams, explanations, examples, and discussions of design forces. The

Antipattern solution is a commonly occurring situation or configuration with significant security implications, such as risks, threats, and vulnerabilities.

- **Causes, Symptoms, and Consequences:** This bulleted section lists the typical causes, common symptoms, and resulting consequences of the Antipattern solution. The intent is to make it easier to recognize the Antipattern and understand how and why its replacement is necessary.
- **Known Exceptions:** If there are some situations where the Antipattern solution might be desirable, this section identifies them. For example, if the consequences are acceptable in a context or if replacement is not worthwhile.
- **Refactored Solution and Examples:** This field defines the Refactored solution. The Refactored solution is proposed as an alternative to the Antipattern solution. Refactoring is a process of replacing or reworking a given solution into an alternative solution. The new solution resolves the design forces differently, particularly providing a more effective solution that resolves design forces more satisfactorily.
- **Related Solutions:** If there are other potential solutions to the Antipattern, they are identified in this section. Often there are different approaches to resolving the same problem that don't conveniently fall under the umbrella of the chosen Refactored solution.

4. CYBER SECURITY ANTIPATTERN CATALOG

IMP QS (question)-06M

- The concept of antipatterns as a way to motivate organizational and behavioral changes.
- the following general antipatterns informally, along with potential solutions:
  - Signature-Based Malware Detection Versus Polymorphic Threats
  - Document-Driven Certification and Accreditation
  - Proliferating IA Standards with No Proven Benefits
  - Policy-Driven Security Certifications Do Not Address the Threat
- Cyber mistakes and bad security habits with these prevalent antipatterns:
  - Can't Patch Dumb
  - Unpatched Applications
  - Never Read the Logs
  - Networks Always Play by the Rules

- Hard on the Outside, Gooey in the Middle
- Webify Everything
- No Time for Security

➢ The antipatterns are intended to be light reading to raise awareness of major security gaps created by how current practitioners develop and manage systems and networks.

4.1 CAN'T PATCH DUMB

IMP QS (question)-07M

Antipattern Name: Can't Patch Dumb

Also Known As: Social Engineering, Phishing, Spam, Spyware, Drive-by Malware, Ransom-Ware, Auto play Attacks

Refactored Solution Names: Security Awareness

Unbalanced Primal Forces: Confidentiality (for example, divulging private information), integrity (for example, rootkits)

Anecdotal Evidence: "Technology is not the problem; people are the problem," and "Technology is easy; people are difficult."

4.2 UNPATCHED APPLICATIONS

IMP QS (question)-07M

Antipattern Name: Unpatched Applications

Also Known As: Vendor-Specific Updates, Default Configuration

Refactored Solution Names: Patch Management

Unbalanced Primal Forces: Management of integrity

Anecdotal Evidence: "Most new attacks are going after the applications, not the operating systems."

4.3 NEVER READ THE LOGS

IMP QS (question)-07M

Antipattern Name: Never Read the Logs

Also Known As: Guys Watching Big Network Displays Miss Everything, Insider Threat, Advanced Persistent Threat (APT), Network Operations Center (NOC)

Refactored Solution Names: Advanced Log Analysis

Unbalanced Primal Forces: Management of confidentiality

Anecdotal Evidence: Nick Leeson at Barings Bank, Wikileaks, Aurora Cyber Intrusions

4.4 NETWORKS ALWAYS PLAY BY THE RULES IMP QS (question)-07M

Antipattern Name: Networks Always Play by the Rules

Also Known As: Trust All Servers, Trust All Clients, Do You Believe in Magic?

Refactored Solution Names: System Hardening, State-of-the-Art Wireless Security Protocols

Unbalanced Primal Forces: Management of confidentiality and integrity

Anecdotal Evidence: In wireless, the access point with the strongest signal is the one that user devices will trust, even if it's malicious.

4.5 HARD ON THE OUTSIDE, GOOEY IN THE MIDDLE

IMP QS (question)-06M

Antipattern Name: Hard on the Outside, Gooey in the Middle

Also Known As: Tootsie Pop, Defense in Depth, Perimeter Security, Protect Everything from All Threats

Refactored Solution Names: HBSS, Network Enclaves

Unbalanced Primal Forces: Management of confidentiality

Anecdotal Evidence: "Each user's browser is sending thousands of spyware beacons every day!"; Advanced Persistent Threat; "Our networks are totally secure; we have a firewall."

4.6 WEBIFY EVERYTHING

IMP QS (question)-07M

Antipattern Name: Webify Everything

Also Known As: Cross-site scripting, Cross-site Request Forgery, US Power Grid on Internet, Global Financial System on Internet

Refactored Solution Names: Physical Separation, Out of Band Separation

Unbalanced Primal Forces: Management of integrity and availability

Anecdotal Evidence: "Why the hell would they put the electrical power grid on the Internet?"

4.7 NO TIME FOR SECURITY

IMP QS (question)-07M

Antipattern Name: No Time for Security

Also Known As: Add Security Last, Blame Security for Schedule Slippage, Deliver It Now!

Refactored Solution Names: Security Requirements Are Real Requirements, Cyber Risk Management

Unbalanced Primal Forces: Management of confidentiality, integrity, and availability

Anecdotal Evidence: “Wait until it’s time to test the system, and then worry about security.”

NETWORK AND CYBER SECURITY
(15EC835, 17EC835)

8TH SEM E&C

JAYANTH DWIJESH H P BE (ECE), M.tech (DECS).

Assistant Professor – Dept of E&CE, BGSIT.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

B.G.S INSTITUTE OF TECHNOLOGY (B.G.S.I.T)

B.G Nagara, Nagamangala Tq, Mandya District- 571448

NETWORK AND CYBER SECURITY
B.E., 8TH Semester, Electronics & Communication Engineering/ Telecommunication Engineering
[As per Choice Based credit System (CBCS) Scheme

Course Code	**17EC835**	**CIE Marks**	**40**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**60**
Total Number of Lecture Hours	**40 (8 Hours per Module)**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives: This course will enable students to:

- Know about security concerns in Email and Internet Protocol.
- Understand cyber security concepts.
- List the problems that can arise in cyber security.
- Discuss the various cyber security frame work.

Module-1

Transport Level Security: Web Security Considerations, Secure Sockets Layer, Transport Layer Security, HTTPS, Secure Shell (SSH) (Text 1: Chapter 15)

Module-2

E-mail Security: Pretty Good Privacy, S/MIME, Domain keys identified mail (Text 1: Chapter 17)

Module-3

IP Security: IP Security Overview, IP Security Policy, Encapsulation Security Payload (ESP), Combining security Associations Internet Key Exchange. Cryptographic Suites(Text 1: Chapter 18)

Module-4

Cyber network security concepts: Security Architecture, Antipattern: signature based malware detection versus polymorphic threads, document driven certification and accreditation, policy driven security certifications. Refactored solution: reputational, behavioural and entropy based malware detection.

The problems: cyber antipatterns concept, forces in cyber antipatterns, cyber anti pattern templates, cyber security Antipattern catalog (Text-2: Chapter1 & 2)

Module-5

Cyber network security concepts contd. :

Enterprise security using Zachman framework

Zachman framework for enterprise architecture, primitive models versus composite models, architectural problem solving patterns, enterprise workshop, matrix mining, mini patterns for problem solving meetings.

Case study: cyber security hands on – managing administrations and root accounts, installing hardware, reimaging OS, installing system protection/ antimalware, configuring firewalls (Text-2: Chapter 3 & 4).

Course Outcomes: After studying this course, students will be able to:

- Explain network web security protocols of SSL, TLS, HTTPS, SSH.
- Outline the basic cyber security concepts - Pretty Good Privacy, S/MIME, and Domain keys identified mail.

- Discuss the IP Security, Cyber network security concepts and cyber security problems.
- Explain Enterprise Security using Zachman Framework.
- Apply concept of cyber security framework to computer system administration.

Text Books:

1. William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325- 1877-3.
2. Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.

Reference Books:

1. Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.
2. Cryptography and Network Security, Atul Kahate, TMH, 2003.

NETWORK AND CYBER SECURITY

MODULE-1

MODULE-1

Transport level security: Web Security Consideration, Security socket layer (SSL), Transport layer security, HTTPS, Secure Shell (SSH).

TEXT BOOK:

1. **William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325-1877-3.**
2. **Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.**

REFERENCE BOOKS:

1. Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.
2. Cryptography and Network Security, Atul Kahate, TMH, 2003.

MODULE - 1

Web Security Consideration, Security socket layer (SSL), Transport layer security, HTTPS, Secure Shell (SSH).

1. WEB SECURITY CONSIDERATION

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

The web provides the following Web Security Threats which make web security a must:

- The Internet is two way. Even unimportant systems like electronic publishing systems, voice response, or fax-back are vulnerable to attacks on the Web servers over the Internet.
- The Web is increasingly serving as a platform for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.
- Although Web browsers, web servers are very easy to use and manage and web content is easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws and hence is more vulnerable to a variety of security attacks.
- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site.
- Casual and untrained users' common clients for Web-based services. Such users are not always aware of the security risks.

1.1 WEB SECURITY THREATS JUNE/JULY-2013[8M], NOV-2020[8M]

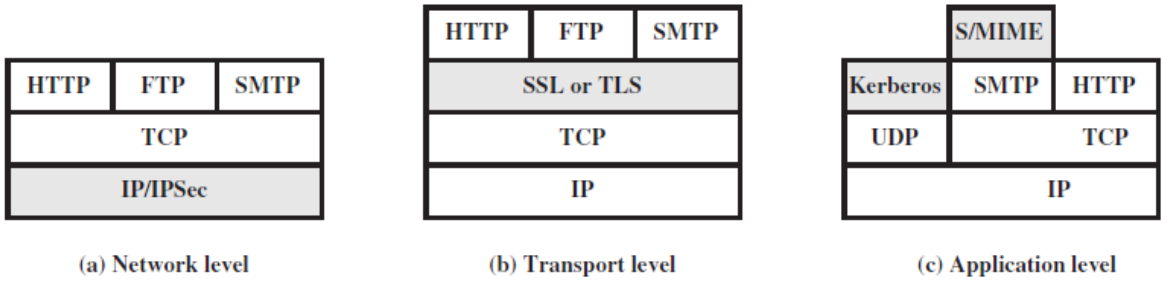
Table 1 provides a summary of the types of security threats faced when using the Web. One way to group these threats is in terms of passive and active attacks. Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.

Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site. Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server.

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

Table 1 A Comparison of Threats on the Web

1.2 WEB TRAFFIC SECURITY APPROACHES



(a) Network level (b) Transport level (c) Application level

Figure 1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

- Figure illustrates that one way to provide Web security is to use IP security (IPsec) (Figure 1 (a)).
- The advantage of using IPsec is that it is transparent to end users and applications and provides a general purpose solution.

- Furthermore, IPsec includes a filtering capability so that only selected traffic need incur the overhead of IPsec processing.
- Another relatively general-purpose solution is to implement security just above TCP (Figure 1 (B)).
- The foremost example of this approaches the Secure Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS).
- At this level, there are two implementation choices.
- For full generality, SSL (or TLS) could provide as part of the underlying protocol suite and therefore be transparent to applications.
- Alternatively, SSL can embed in specific packages.
- For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol.
- Application-specific security services embedded within the particular application.
- The figure shows (Figure 1 (C)). Examples of this architecture.
- The advantage of this approach is that the service can tailor to the specific needs of a given application

2. SECURE SOCKET LAYER

2.1 SSL ARCHITECTURE

MAY/JUNE-2010 [10M], JUN/JULY-2011 [10M], DEC-2011[4M], DEC-2012[10M], JUNE/JULY-2013[8M], DEC/JAN-2016[10M]

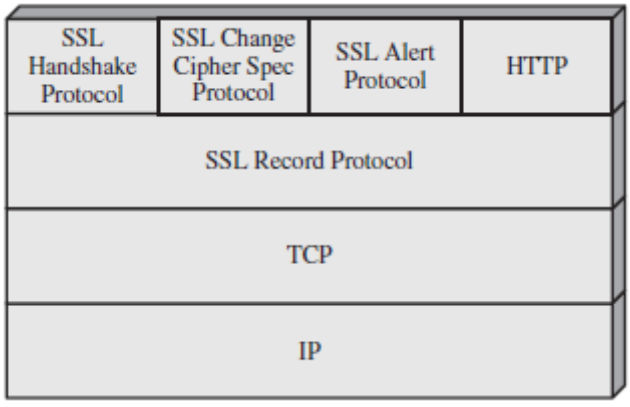


Figure 2 SSL Protocol Stack

- Secure Socket Layer is designed to make use of TCP to provide a reliable end-to-end secure service.

- Moreover, Secure Socket Layer is not a single protocol but rather two layers of protocols, as illustrated in Figure (2) below.
- The SSL Record Protocol provides basic security services to various higher layer protocols.
- In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL.
- Three higher-layer protocols are defined as part of SSL: the **Handshake Protocol**, the **Change Cipher Spec Protocol**, and the **Alert Protocol**.

TWO IMPORTANT SSL CONCEPTS ARE THE SSL SESSION AND THE SSL CONNECTION, WHICH ARE DEFINED IN THE SPECIFICATION AS FOLLOWS. DEC-2012[4M]

- **Connection:** A connection is a transport that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection associated with one session.

- **Session:** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

- There a number of states associated with each session. Once a session established, there is a current operating state for both read and write (i.e., receive and send)
- In addition, during the Handshake Protocol, pending read and writes states created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states

2.2 A SESSION STATE IS DEFINED BY THE FOLLOWING PARAMETERS MAY/JUNE-2010 [10M], JUN/JULY-2011[10M], JUNE/JULY-2019[8M], DEC-2019[8M], SEPT-2020[4M].

- **Session identifier:** An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
- **Peer certificate:** An X509.v3 certificate of the peer. This element of the state may be null.

- **Compression method:** The algorithm used to compress data prior to encryption.
- **Cipher spec:** Specifies the bulk data encryption algorithm (such as null, AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash size.
- **Master secret:** 48-byte secret shared between the client and the server.
- **Is resumable:** A flag indicating whether the session can be used to initiate new connections.

2.3 A CONNECTION STATE IS DEFINED BY THE FOLLOWING PARAMETERS. MAY/JUNE-2010 [10M], JUN/JULY-2011[10M], JUNE/JULY-2019[8M], DEC-2019[4M].SEPT-2020[4M].

- **Server and client random:** Byte sequences that are chosen by the server and client for each connection.
- **Server write MAC secret:** The secret key used in MAC operations on data sent by the server.
- **Client write MAC secret:** The secret key used in MAC operations on data sent by the client.
- **Server write key:** The secret encryption key for data encrypted by the server and decrypted by the client.
- **Client write key:** The symmetric encryption key for data encrypted by the client and decrypted by the server.
- **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol.
- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64} - 1$.

2.4 SSL RECORD PROTOCOL: SSL PROTOCOL JUN/JULY-2017[10M], JUNE-2012[10M], DEC/JAN-2019[10M], JUNE/JULY-2019[8M], SEPT-2020[6M]

- The SSL Record Protocol provides two services for SSL connections: **Confidentiality** and **Message Integrity**.
  - **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
  - **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).
- Moreover, the overall operation of Record Protocol is:
  - **Fragmentation**: Each upper-layer message fragmented into blocks of 2^{14} bytes (16384 bytes) or less.
  - **Compression**: Compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes.

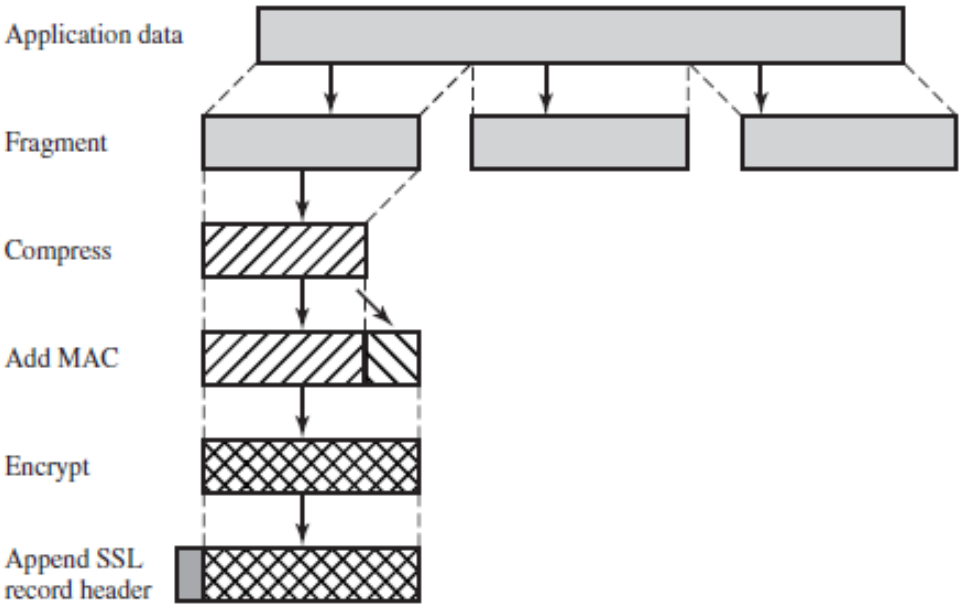


Figure 3 SSL Record Protocol Operation

- **Add message authentication code(MAC)**: MAC calculated over the compressed data by the following expression.

Hash (MAC_write_secret || pad_2 || hash (MAC_write_secret || pad_1 || seq _ num || SSL Compressed. type || SSL Compressed. length || SSL Compressed. fragment))

Where

|| = concatenation.

MAC_write_secret = shared secret key.

Hash = cryptographic hash algorithm.

pad_1 = the byte 0x36 (0011 0110) repeated 48 times (384 bits) for MD5 and 40 times (320 Bits) for SHA-1.

pad_2 = the byte 0x5C (0101 1100) repeated 48 Times for MD5 and 40 times for SHA-1.

Seq _ num = the sequence number for this message.

SSL Compressed .type = the higher-level protocol used to process this fragment.

SSL Compressed. Length = the length of the compressed fragment.

SSL Compressed. Fragment = the compressed fragment or plain text (if compression =not used).

- **Encryption:** The compressed message plus the MAC encrypted using symmetric encryption. Algorithms supported are AES, RC4-40, IDEA, RC2, DES, 3DES and Fortezza.

Figure 4 illustrates the SSL record format.

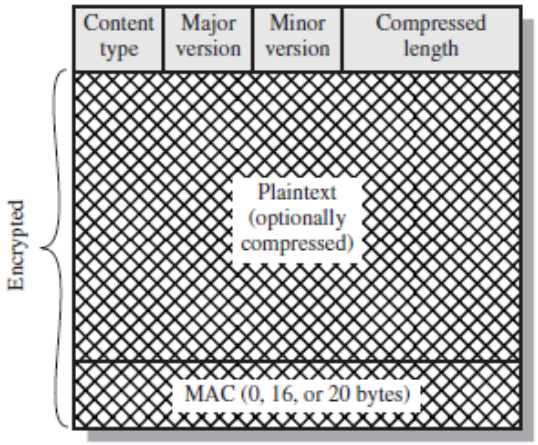


Figure 4 SSL Record Formats

- **Final step:** The **final step** of SSL Record Protocol processing is to prepare a header consisting of the following fields:
  - **Content Type (8 bits)**: The higher-layer protocol used to process the fragment.
  - **Major Version (8 bits)**: Indicates major version of SSL in use. For SSLv3, the value is 3.

- **Minor Version (8 bits)**: Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits)**: The length in bytes of the fragment.

2.5 CHANGE CIPHER SPEC PROTOCOL

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest.This protocol consists of a single message (Figure 5(a)) of a single byte with the value 1.

The purpose of this message to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

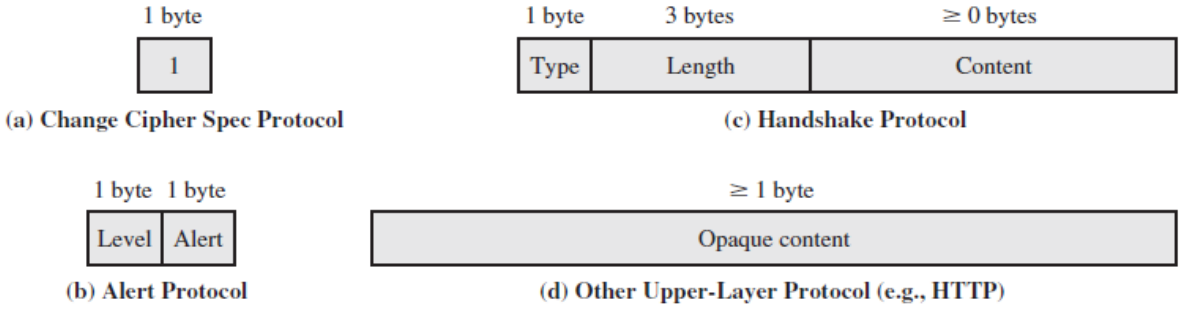


Figure 5 SSL Record Protocol Payload

2.6 ALERT PROTOCOL

- The Alert Protocol used to convey SSL-related alerts to the peer entity.
- Moreover, each message in this protocol consists of two bytes (Figure 5(b)).
- The first byte takes the value warning (1) or fatal (2) to convey the severity of the message.
- If the level is fatal, SSL immediately terminates the connection.
- Other connections on the same session may continue, but no new connections established.
- The second byte contains a code that indicates the specific alert.
- First, we list those alerts that are always fatal (definitions from the SSL specification):
  - **Unexpected message:** An inappropriate message was received.
  - **Bad _ record _ mac:** An incorrect MAC was received.

- **Decompression _ failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).
- **Handshake _ failure:** Sender was unable to negotiate an acceptable set of security parameters given the options available.
- **Illegal _ parameter:** A field in a handshake message was out of range or inconsistent with other fields.

➢ The remaining alerts are the following.

- **Close _ notify:** Notifies the recipient that the sender will not send any more messages on this connection. Each party is required to send a **close _ notify** alert before closing the write side of a connection.
- **No_ certificate:** May be sent in response to a certificate request if no appropriate certificate is available.
- **Bad _ certificate:** A received certificate was corrupt (e.g., contained a signature that did not verify).
- **Unsupported _ certificate:** The type of the received certificate is not supported.
- **Certificate _ revoked:** A certificate has been revoked by its signer.
- **Certificate _ expired:** A certificate has expired.
- **Certificate _ unknown:** Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

2.7 Handshake Protocol DEC-2010 [12M], DEC-2011[8M], JAN 2015[10M], NOV-2020[10M], SEPT-2020[8M],JULY-2019[8M]

➢ This Handshake Protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.

➢ Moreover, The Handshake Protocol is used before any application data is transmitted.

➢ The Handshake Protocol consists of a series of messages exchanged by client And server. All of these have the format shown in (Figure 5(c)).

➢ A handshake message has the following format:

1. **Type (1 byte)**: Indicates one of 10 messages of handshake protocol. Table 2 lists the defined message types
2. **Length (3 bytes)**: The length of the message in bytes.
3. **Content (bytes)**: The parameters associated with this message.This are Listed in Below Table

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Table 2 SSL Handshake Protocol Message Types

- (Figure 6) shows the initial exchange needed to establish a logical connection between client and server. The exchange can be viewed as having four phases.

PHASE 1. ESTABLISH SECURITY CAPABILITIES:

- This phase is used to initiate a logical connection and to establish the security capabilities that will be associated with it.
- The exchange is initiated by the client, which sends a **client _hello message** with the following parameters:
  - **Version**: The highest SSL version understood by the client.
  - **Random**: A client-generated random number which serves as the nonce.
  - **Session ID**: A variable-length session identifier. A nonzero value indicates that the client wishes to update the parameters of an existing session. A zero value indicates that the client wishes to establish a new connection on a new session.
  - **Cipher Suite**: This is a list that contains the cryptographic algorithms (key exchange, encryption, and MAC) supported by the client, in decreasing order of preference.

- **Compression Method**: This is a list of the compression methods the client supports.
- After sending the client _ hello message, the client waits for the server _ hello message, which contains the same parameters as the client _ hello message. The parameters contain the values which client had sent to the server and the server has chosen to use.

PHASE 2: SERVER AUTHENTICATION AND KEY EXCHANGE:

This phase provides authentication of the server to the client. o The server sends its certificate (one or more) if it needs to be authenticated the message contains one or a chain of X.509 certificates.

- The server sends a **server _ key _ exchange message** which contains the list of secret keys to be used for the subsequent data. The **certificate _ request message** is sent next which includes two parameters: certificate _ type and certificate _ authorities.
- Moreover, the final message in phase 2, and one that always required is the **server _ done** message, which sent by the server to indicate the end of the server hello and associated messages.
- After sending this message, the server will wait for a client response. This message has no parameters.

PHASE 3. CLIENT AUTHENTICATION AND KEY EXCHANGE:

This phase provides client authentication to the server**.**

- The client verifies the server certificates and checks whether the server _ hello parameters are acceptable.
- Moreover, if all is satisfactory, the client sends a **certificate** message if the server has requested a certificate. If no suitable certificate is available, the client sends a no _ certificate alert.
- Next is the **client _ key _ exchange** message which has the same parameters as the server _key _ exchange message.
- Similarly, the client may send a **certificate _ verify** message to provide explicit verification of a client certificate.
- The client encrypts all the previous messages and master secret with its private key.

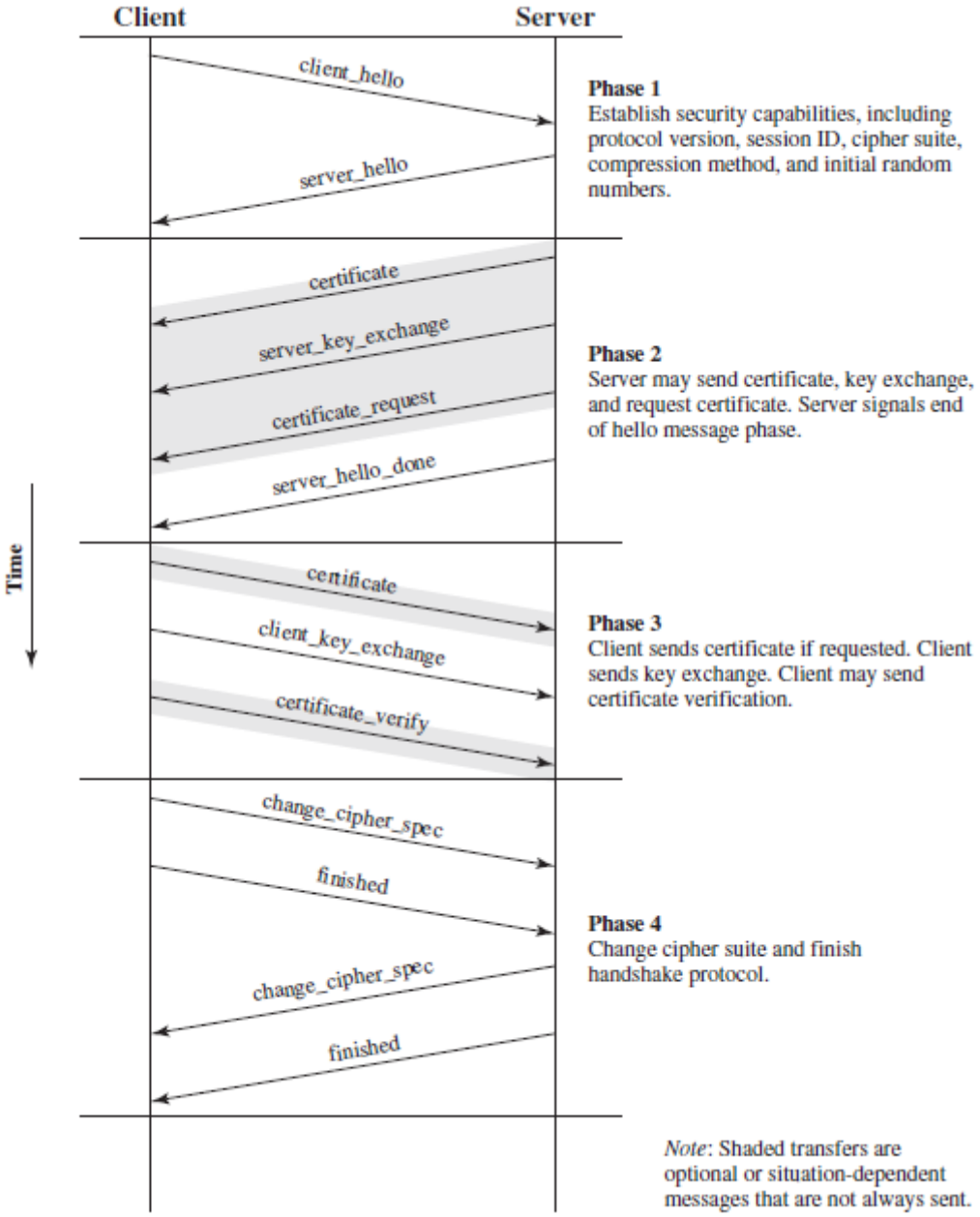


Figure 6 Handshake Protocol Actions

PHASE 4. FINISH

This phase completes the setting up of a secure connection.

- The client sends a **change _ Cipher _ spec** message and copies the pending **Cipher Spec** into the current **Cipher Spec**.
- Moreover, the client then immediately sends the **finished message** under the new algorithms, keys, and secrets.
- The content of the finished message is the concatenation of two hash values:

MD5 (master _secret || pad2 || MD5 (handshake _messages || Sender || master _secret || pad1))

SHA (master _secret || pad2 || SHA (handshake _messages || Sender || master _secret || pad1))

- The server sends its own **change _ cipher _ spec** message, transfers the pending to the current **Cipher Spec**, and sends it **finished**
- At this point, the handshake is complete and the client and server may begin to exchange application-layer data.

3. TRANSPORT LAYER SECURITY (TLS)

- TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL.
- Moreover, TLS is defined as a Proposed Internet Standard in RFC 5246. Is very similar to SSLv3.
- We highlight the differences.

VERSION NUMBER

- The TLS Record Format is the same as that of the SSL Record Format and the fields in the header have the same meanings.
- The one difference is in version values. For the current version of TLS, the major version is 3 and the minor version is 3.

MESSAGE AUTHENTICATION CODE: TRANSPORT LAYER SECURITY

- There are two differences between the SSLv3 and TLS MAC schemes:
- The actual algorithm and the scope of the MAC calculation.
- TLS makes use of the HMAC algorithm defined in RFC 2104.
- HMAC is defined as

$$\text{HMAC}_K(M) = \text{H}[(K^+ \oplus \text{opad}) \parallel \text{H}[(K^+ \oplus \text{ipad}) \parallel M]]$$

where

H = embedded hash function (for TLS, either MD5 or SHA-1)

M = message input to HMAC

K^+ = secret key padded with zeros on the left so that the result is equal to the block length of the hash code (for MD5 and SHA-1, block length = 512 bits)

Ipad = 00110110 (36 in hexadecimal) repeated 64 times (512 bits)

opad = 01011100 (5C in hexadecimal) repeated 64 times (512 bits)

- SSLv3 uses the same algorithm, except that the padding bytes are concatenated with the secret key rather than being XORed with the secret key padded to the block length. Moreover, The level of security should be about the same in both cases.
- For TLS, the MAC calculation encompasses the fields indicated in the following expression:

MAC(MAC_write_secret,seq_num || TLSCompressed.type ||

TLSCompressed.version || TLSCompressed.length ||

TLSCompressed.fragment)

- The MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLSCompressed.version, , which is the version of the protocol being employed.

3.1 PSEUDORANDOM FUNCTION NOV-2020[6M]

- TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation.
- Moreover, the objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash functions and MACs.
- The PRF is based on the data expansion function (Figure 7) given as

P _ hash (secret, seed) = HMAC _ hash (secret, A(1) || seed) ||

HMAC _hash (secret, A (2) || seed) ||

HMAC _ hash (secret, A (3) || seed) ||...

Where A () is defined as

A (0) = seed

A (i) = HMAC _ hash (secret, A (i – 1))

- PRF defined as

PRF (secret, label, seed) = P _ hash (S1, label || seed)

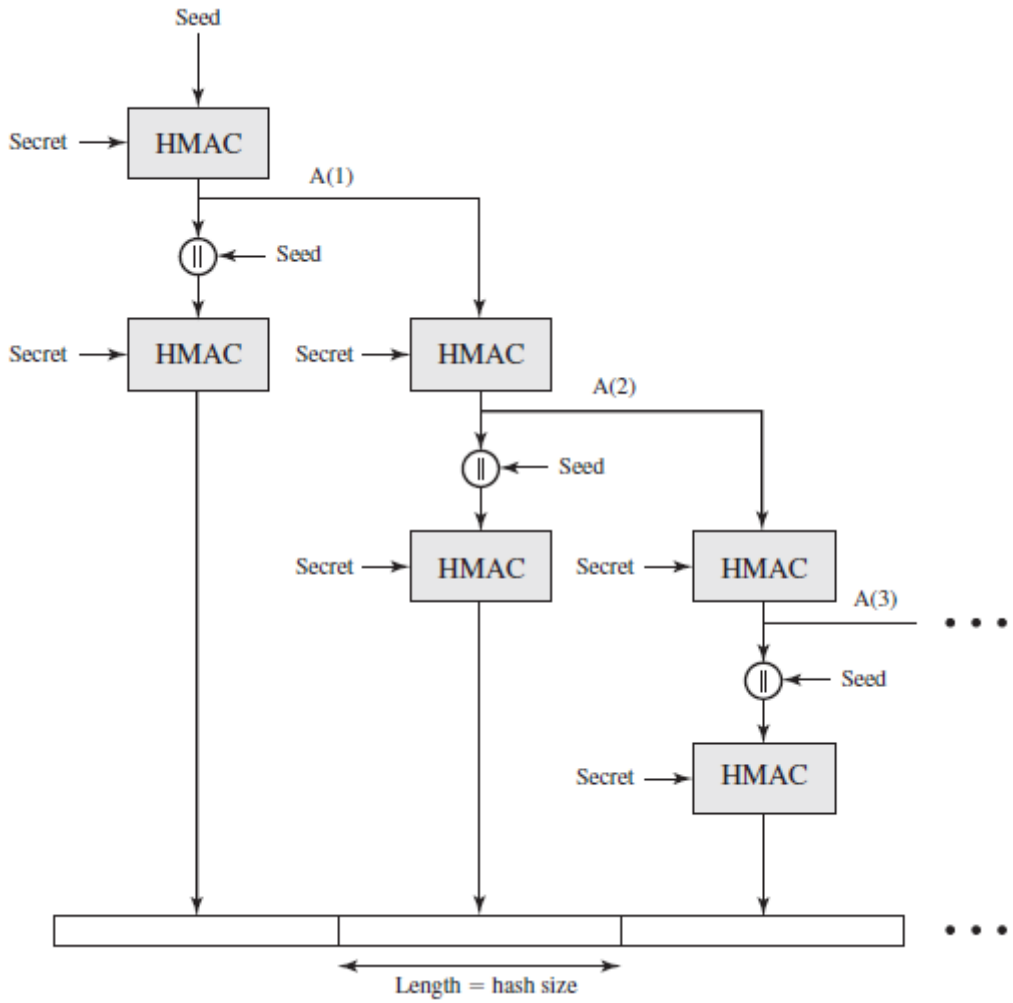


Figure 7 TLS Function P _ hash (secret, seed)

PRF takes as input a secret value, an identifying label, and a seed value and produces an output of arbitrary length.

3.2 Alert Codes: Transport Layer Security DEC-2019[8M], NOV-2020[6M]

- TLS supports all of the alert codes defined in SSLv3 with the exception of no _ certificate.
- A number of additional codes defined in TLS; of these, the following are always fatal.
  - **Record _ overflow:** A TLS record was received with a payload (cipher text) whose length exceeds 214 + 2048 bytes, or the cipher text decrypted to a length of greater than 214 + 1024 bytes.
  - **Unknown _ ca:** A valid certificate chain or partial chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.

- **Access _ denied:** A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.
- **Decode _ error:** A message could not be decoded, because either a field was out of its specified range or the length of the message was incorrect.
- **Protocol _version:** The protocol version the client attempted to negotiate is recognized but not supported.
- **Insufficient _ security:** Returned instead of handshake _ failure when a negotiation has failed specifically because the server requires ciphers more secure than those supported by the client.
- **Unsupported _ extension:** Sent by clients that receives an extended server hello containing an extension not in the corresponding client hello.
- **Internal _ error:** An internal error unrelated to the peer or the correctness of the protocol makes it impossible to continue.
- **Decrypt _ error:** A handshake cryptographic operation failed, including being unable to verify a signature, decrypt a key exchange, or validate a finished message.

➢ The remaining alerts include the following.

- **User _ cancelled:** This handshake is being cancelled for some reason unrelated to a protocol failure.
- **No _ renegotiation:** Sent by a client in response to a hello request or by the server in response to a client hello after initial handshaking. Either of these messages would normally result in renegotiation, but this alert indicates that the sender is not able to renegotiate. This message is always a warning.

3.3 Cipher Suites:

Moreover, there are several small differences between the cipher suites available under SSLv3 and under TLS:

➢ **Key Exchange:** TLS supports all of the key exchange techniques of SSLv3 with the exception of Fortezza.

➢ **Symmetric Encryption Algorithms:** TLS includes all of the symmetric encryption algorithms found in SSLv3, with the exception of Fortezza.

3.4 Client Certificate Types

- TLS defines the following certificate types to be requested in **a certificate_ request** message: **rsa _sign, dss _sign, rsa _fixed _dh,** and **dss _fixed _dh.**
- These are all defined in SSLv3. In addition, SSLv3 includes **rsa _ ephemeral _dh, dss _ ephemeral _dh,** and **fortezza _kea.**
- Ephemeral Diffie-Hellman involves signing the Diffie-Hellman parameters with either RSA or DSS. For TLS, the **rsa _ sign** and **dss _sign** types are used for that function; a separate signing type is not needed to sign Diffie-Hellman parameters.
- TLS does not include the Fortezza scheme.

3.5 Certificate _verify and Finished Messages

- In the TLS certificate _verify message, the MD5 and SHA-1 hashes are calculated only over handshake _messages.
- Recall that for SSLv3, the hash calculation also included the master secret and pads. These extra fields were felt to add no additional security.
- As with the finished message in SSLv3, the finished message in TLS is a hash based on the shared master _secret, the previous handshake messages, and a label that identifies client or server. The calculation is somewhat different. For TLS, we have

  PRF (master secret, finished label, MD5 (handshake _ messages) || SHA-1 (handshake _messages))

  Where finished label is the string "client finished" for the client and "server finished" for the server.

3.6 Cryptographic Computations

The pre _ master _secret for TLS is calculated in the same way as in SSLv3. As in SSLv3, the master _secret in TLS is calculated as a hash function of the pre_ master _secret and the two hello random numbers. The form of the TLS calculation is different from that of SSLv3 and is defined as

Master _secret = PRF (pre _master _secret, "master secret", Client Hello .random || Server Hello .random)

The algorithm is performed until 48 bytes of pseudorandom output are produced. The calculation of the key block material (MAC secret keys, session encryption keys, and IVs) is defined as

Key _block = PRF (master _secret, "key expansion", Security Parameters. Server _ random|| Security Parameters. Client _random)

Until enough output has been generated. As with SSLv3, the key _block is a function of the master _secret and the client and server random numbers, but for TLS, the actual algorithm is different.

3.7Padding

In SSL, the padding added prior to encryption of user data is the minimum amount required so that the total size of the data to be encrypted is a multiple of the cipher's block length. In TLS, the padding can be any amount that results in a total that is a multiple of the cipher's block length, up to a maximum of 255 bytes.

For example,

- if the plaintext (or compressed text if compression is used) plus MAC plus padding.
- Length byte is 79 bytes long, then the padding length (in bytes) can be 1, 9, 17, and so on, up to 249.
- A variable padding length may be used to frustrate attacks based on an analysis of the lengths of exchanged messages.

4. HTTPS MQP[8M]

- HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browserand a Web server.
- The HTTPS capability is built into all modern Web browsers.
- Its use depends on the Web server supporting HTTPS communication. For example, search engines do not support HTTPS.
- The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://.
- A normal HTTP connection uses port 80.
- If HTTPS is specified, port 443 is used, which invokes SSL.
- When HTTPS is used, the following elements of the communication are encrypted:
  - URL of the requested document

- Contents of the document
- Contents of browser forms (filled in by browser user)
- Cookies sent from browser to server and from server to browser
- Contents of HTTP header

- HTTPS is documented in RFC 2818, HTTP over TLS. There is no fundamental change in using HTTP over either SSL or TLS, and both implementations are referred to as HTTPS.

4.1 Connection Initiation

- For HTTPS, the agent acting as the HTTP client also acts as the TLS client.
- The client initiates a connection to the server on the appropriate port and then sends the TLS Client Hello to begin the TLS handshake.
- When the TLS handshake has finished, the client may then initiate the first HTTP request. All HTTP data is to be sent as TLS application data. Normal HTTP behavior, including retained connections, should be followed.
- There are three levels of awareness of a connection in HTTPS.
  - At the HTTP level, an HTTP client requests a connection to an HTTP server by sending a connection request to the next lowest layer.
  - Typically, the next lowest layer is TCP, but it also may be TLS/SSL.
  - At the level of TLS, a session is established between a TLS client and a TLS server. This session can support one or more connections at any time. As we have seen, a TLS request to establish a connection begins with the establishment of a TCP connection between the TCP entity on the client side and the TCP entity on the server side.

4.2 Connection Closure

- An HTTP client or server can indicate the closing of a connection by including the following line in an HTTP record: ***Connection: close.*** This indicates that the connection will be closed after this record is delivered.
- The closure of an HTTPS connection requires that TLS close the connection With the peer TLS entity on the remote side, which will involve closing the underlying TCP connection.

- At the TLS level, the proper way to close a connection is for each side to use the TLS alert protocol to send a ***close notify*** alert.
- TLS implementations must initiate an exchange of closure alerts before closing a connection.
- A TLS implementation may, after sending a closure alert, close the connection without waiting for the peer to send its closure alert, generating an "incomplete close".
- HTTP clients also must be able to cope with a situation in which the underlying TCP connection is terminated without a prior ***close notify*** alert and without a *Connection: close* indicator. Such a situation could be due to a programming error on the server or a communication error that causes the TCP connection to drop.
- However, the unannounced TCP closure could be evidence of some sort of attack. So the HTTPS client should issue some sort of security warning when this occurs.

5 SECURE SHELLS (SSH) DECAUG/SEPT-2020[8M]

- Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.
- The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and Other remote logon schemes that provided no security.
- SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail.
- A new version, SSH2, fixes a number of security flaws in the original scheme.
- SSH2 is documented as a proposed standard in IETF RFCs 4250 through 4256.
- SSH client and server applications are widely available for most operating systems.
- It has become the method of choice for remote login and X tunnelling and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems.
- SSH is organized as three protocols that typically run on top of TCP (above Figure 8):
  - **Transport Layer Protocol:** Provides server authentication, data confidentiality, and data integrity with forward secrecy. The transport layer may optionally provide compression.

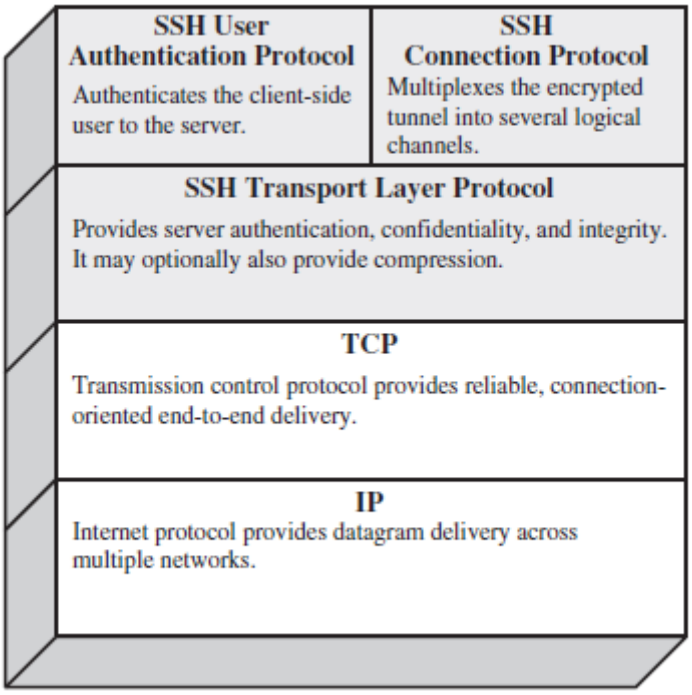


Figure 8: SSH Protocol Stack

- **User Authentication Protocol:** Authenticates the user to the server.
- **Connection Protocol:** Multiplexes multiple logical communications channels over a single, underlying SSH connection.

5.1 TRANSPORT LAYER PROTOCOL

5.1.1 Host Keys

- Server authentication occurs at the transport layer, based on the server possessing a public/private key pair.
- A server may have multiple host keys using multiple different asymmetric encryption algorithms. Multiple hosts may share the same host key.
- In any case, the server host key is used during key exchange to authenticate the identity of the host. For this to be possible, the client must have a priori knowledge of the server's public host key. RFC 4251 dictates two alternatives Trust models that can be used:
  1. The client has a local database that associates each host name (as typed by the user) with the corresponding public host key. This method requires no centrally administered infrastructure and no third-party coordination. The

downside is that the database of name-to-key associations may become burdensome to maintain.

2. The host name-to-key association is certified by a trusted certification authority (CA). The client only knows the CA root key and can verify the validity of all host keys certified by accepted CAs. This alternative eases the maintenance problem, since ideally, only a single CA key needs to be securely stored on the client. On the other hand, each host key must be appropriately certified by a central authority before authorization is possible.

5.1.2 Packet Exchange

- **SSH Transport Layer Protocol Packet Exchanges** **MQP [8M]**

Figure (9) illustrates the sequence of events in the SSH Transport Layer Protocol. First, the client establishes a TCP connection to the server. This is done via the TCP protocol and is not part of the Transport Layer Protocol. Once the connection is established, the client and server exchange data, Referred to as packets, in the data field of a TCP segment.

The SSH Transport Layer packet exchange consists of a sequence of steps (**Figure 9**).

1. **The first step**, the identification string exchange, begins with the client sending a packet with an identification string of the form:

   SSH-protoversion-software version SP comments CR LF

   Where SP, CR, and LF are space character, carriage return, and line feed, respectively

2. **Next comes algorithm negotiatio**n. Each side sends an **SSH_MSG_KEXINIT** containing lists of supported algorithms in the order of preference to the sender. There is one list for each type of cryptographic algorithm. The algorithms include key exchange, encryption, MAC algorithm, and compression algorithm.

   Below Table (2) shows the allowable options for encryption, MAC, and compression.

3. The next step is **key exchange**._As a result of these steps, the two sides now share a master key *K*. In addition, the server has been authenticated to the client, because the server has used its private key to sign its half of the Diffie-Hellman

exchange. Finally, the hash value *H* serves as a session identifier for this connection. Once computed, the session identifier is not changed, even if the key exchange is performed again for this connection to obtain fresh keys.

Cipher	
3des-cbc*	Three-key 3DES in CBC mode
blowfish-cbc	Blowfish in CBC mode
twofish256-cbc	Twofish in CBC mode with a 256-bit key
twofish192-cbc	Twofish with a 192-bit key
twofish128-cbc	Twofish with a 128-bit key
aes256-cbc	AES in CBC mode with a 256-bit key
aes192-cbc	AES with a 192-bit key
aes128-cbc**	AES with a 128-bit key
Serpent256-cbc	Serpent in CBC mode with a 256-bit key
Serpent192-cbc	Serpent with a 192-bit key
Serpent128-cbc	Serpent with a 128-bit key
arcfour	RC4 with a 128-bit key
cast128-cbc	CAST-128 in CBC mode

MAC algorithm	
hmac-sha1*	HMAC-SHA1; digest length = key length = 20
hmac-sha1-96**	First 96 bits of HMAC-SHA1; digest length = 12; key length = 20
hmac-md5	HMAC-MD5; digest length = key length = 16
hmac-md5-96	First 96 bits of HMAC-MD5; digest length = 12; key length = 16

Compression algorithm	
none*	No compression
zlib	Defined in RFC 1950 and RFC 1951

* = Required
** = Recommended

Table1 2: SSH Transport Layer Cryptographic Algorithms

4. The **end of key exchange** is signalled by the exchange of **SSH_MSG_NEWKEYS** packets. At this point, both sides may start using the keys generated from *K*, as discussed subsequently.

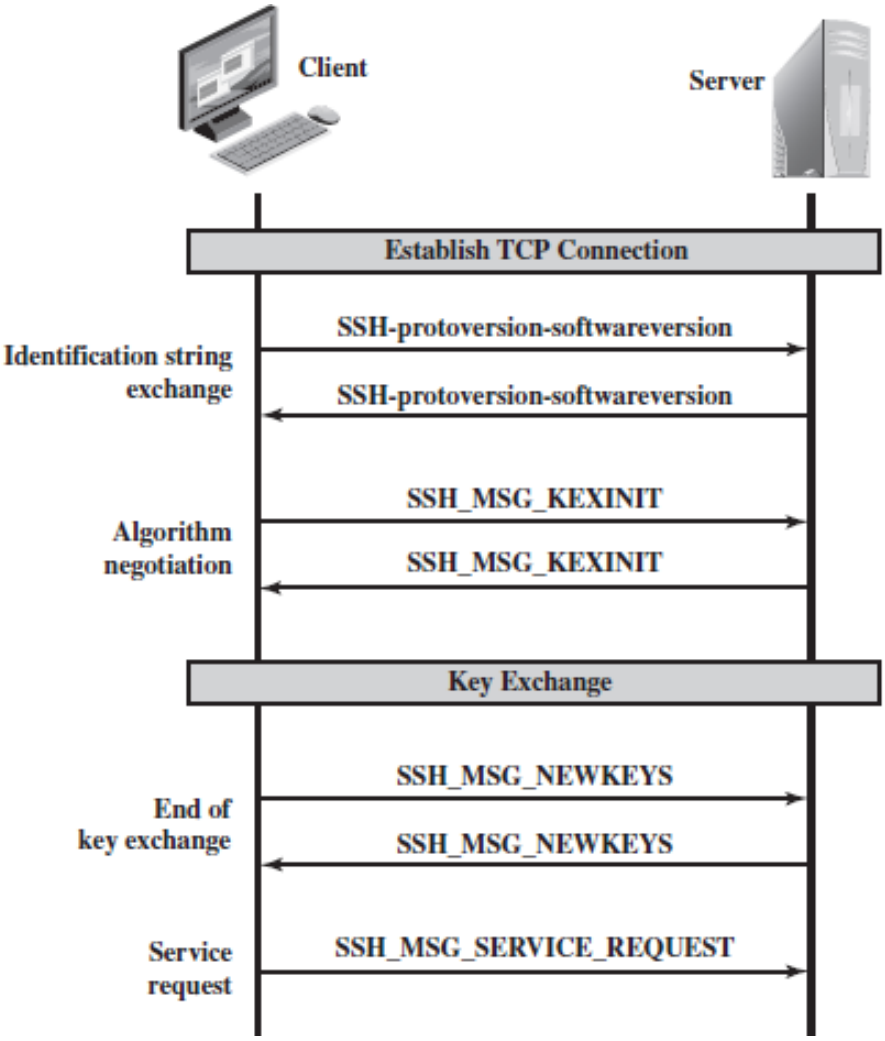


Figure 9: SSH Transport Layer Protocol Packet Exchanges

5. The final step is **service request**. The client sends an **SSH_MSG_SERVICE_ REQUEST** packet to request either the User Authentication or the Connection Protocol. Subsequent to this, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

- **SSH Transport Layer Protocol Packet Formation**

JUNE/JULY-2019 [8M]

Each packet is in the following Format (Figure 10).

- **Packet length:** Length of the packet in bytes, not including the packet length and MAC fields.
- **Padding length:** Length of the random padding field.

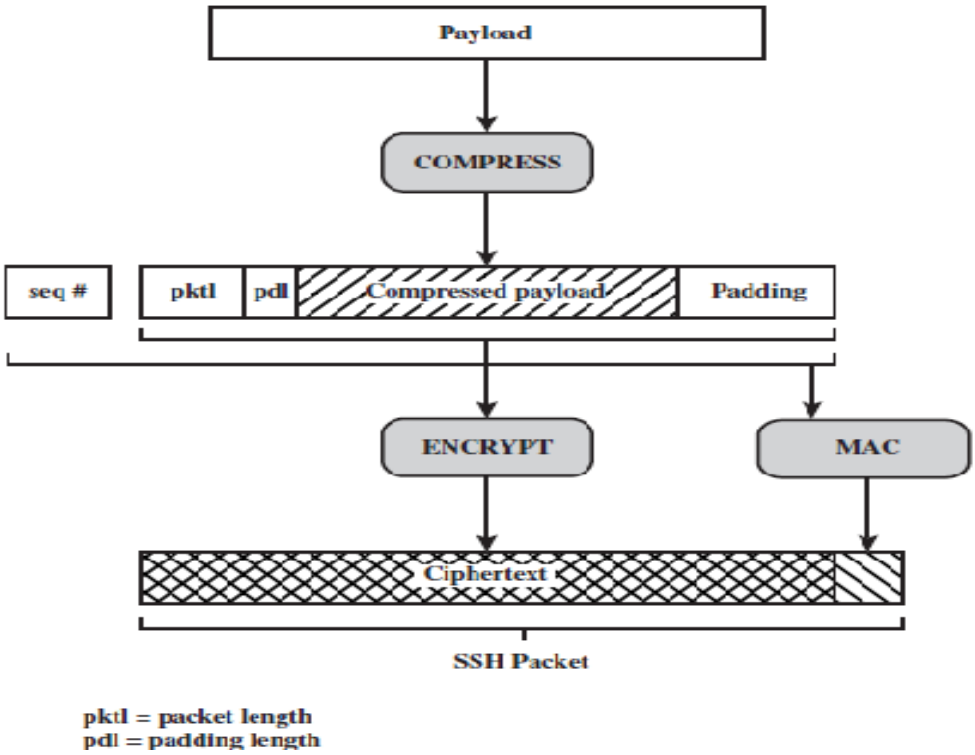


Figure 10: SSH Transport Layer Protocol Packet Formation

- **Payload:** Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.
- **Random padding:** Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or 8 bytes for a stream cipher.
- **Message authentication code (MAC):**
  - If message authentication has been negotiated, this field contains the MAC value.
  - The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field.
  - The sequence number is an implicit 32-bit packet sequence that is initialized to zero for the first packet and incremented for every packet.

Once an encryption algorithm has been negotiated, the entire packet (excluding the MAC field) is encrypted after the MAC value is calculated.

5.1.3 Key generation

the keys used for encryption and MAC (and any needed IVs) are generated from the shared secret key K, the hash value from the key exchange H, and the session identifier, which is equal to H unless there has been a subsequent key exchange after the initial key exchange.

The values are computed as follows.

- **Initial IV client to server: HASH (K‖H‖ "A"‖ session _id)**
- **Initial IV server to client: HASH (K‖H‖ "B" ‖ session _id)**
- **Encryption key client to server: HASH (K‖H‖ "C" ‖ session _id)**
- **Encryption key server to client: HASH(K‖H‖ "D" ‖ session _id)**
- **Integrity key client to server: HASH(K‖H‖ "E" ‖ session _id)**
- **Integrity key server to client: HASH(K‖H‖"F" ‖ session _id)**

Where HASH () is the hash function determined during algorithm negotiation.

5.2 USER AUTHENTICATION PROTOCOL

The User Authentication Protocol provides the means by which the client is authenticated to the server.

5.2.1 Message Types and Formats

Three types of messages are always used in the User Authentication Protocol. Authentication requests from the client have the format:

Byte SSH_MSG_USERAUTH_REQUEST (50)

String user name

String service name

String method name

... Method specific fields

Where user name is the authorization identity the client is claiming, service name is the facility to which the client is requesting access (typically the SSH Connection Protocol), and method name is the authentication method being used in this request. The first byte has decimal value 50, which is interpreted as **SSH_MSG_USERAUTH_REQUEST.**

If the server either (1) rejects the authentication request or (2) accepts the request but requires one or more additional authentication methods, the server sends a message with the format:

Byte **SSH_MSG_USERAUTH_FAILURE (51)**

Name-list **authentications that can continue**

Boolean **partial success**

Where the name-list is a list of methods that may productively continue the dialog.

If the server accepts authentication, it sends a single byte message: SSH_MSG_ USERAUTH_SUCCESS (52).

5.2.2 Message Exchange

The message exchange involves the following steps.

1. The client sends a **SSH_MSG_USERAUTH_REQUEST** with a requested method Of none.
2. The server checks to determine if the user name is valid. If not, the server returns **SSH_MSG_USERAUTH_FAILURE** with the partial success value of false. If the user name is valid, the server proceeds to step 3.
3. The server returns **SSH_MSG_USERAUTH_FAILURE** with a list of one or more authentication methods to be used.
4. The client selects one of the acceptable authentication methods and sends a **SSH_MSG_USERAUTH_REQUEST** with that method name and the required method-specific fields. At this point, there may be a sequence of exchanges to perform the method.
5. If the authentication succeeds and more authentication methods are required, the server proceeds to step 3, using a partial success value of true. If the authentication fails, the server proceeds to step 3, using a partial success value of false.
6. When all required authentication methods succeed, the server sends a **SSH_MSG_USERAUTH_SUCCESS** message, and the Authentication Protocol is over.

5.2.3 Authentication Methods

The server may require one or more of the following authentication methods.

1. **Public key:** The details of this method depend on the public-key algorithm chosen. In essence, the client sends a message to the server that contains the client's public key, with the message signed by the client's private key. When the

server receives this message, it checks whether the supplied key is acceptable for authentication and, if so, it checks whether the signature is correct.

2. **Password:** The client sends a message containing a plaintext password, which is protected by encryption by the Transport Layer Protocol.
3. **Host based:** Authentication is performed on the client's host rather than the client itself. Thus, a host that supports multiple clients would provide authentication for all its clients. This method works by having the client send a signature created with the private key of the client host. Thus, rather than directly verifying the user's identity, the SSH server verifies the identity of the client host—and then believes the host when it says the user has already authenticated on the client side.

5.3 Connection Protocol

The SSH Connection Protocol runs on top of the SSH Transport Layer Protocol and assumes that a secure authentication connection is in use. That secure authentication connection, referred to as a tunnel, is used by the Connection Protocol to multiplex a number of logical channels.

5.3.1 Channel Mechanism

- **SSH Connection Protocol Message Exchange** **JAN-2020 [6M]**

Figure 11 provides an example of Connection Protocol Message Exchange

1. All types of communication using SSH, such as a terminal session, are supported using separate channels.
2. Either side may open a channel. For each channel, each side associates a unique channel number, which need not be the same on both ends.
3. Channels are flow controlled using a window mechanism.
4. No data may be sent to a channel until a message is received to indicate that window space is available. The life of a channel progresses through three stages: opening a channel, data transfer, and closing a channel.
5. When either side wishes to **open a new channel,** it allocates a local number for the channel and then sends a message of the form:

   Byte SSH_MSG_CHANNEL_OPEN

   String channel type

   Uint32 sender channel

Uint32 initial window size

Uint32 maximum packet size

.... channel type specific data follows

Where uint32 means unsigned 32-bit integer

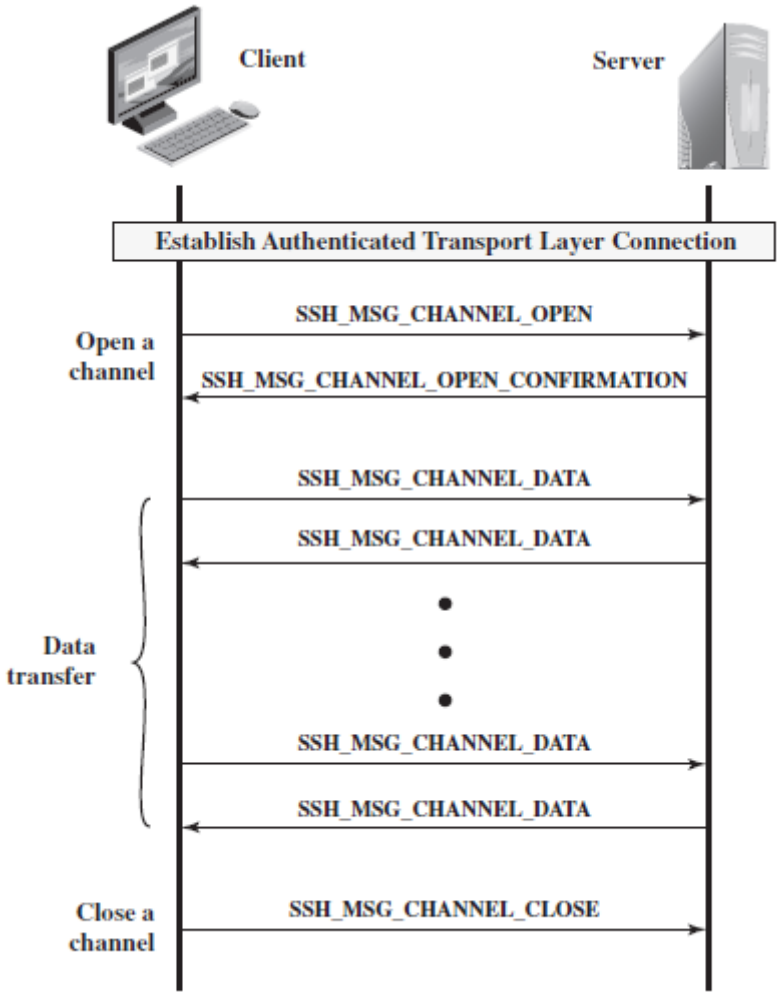


Figure 11 Example of SSH Connection Protocol Message Exchange

6. If the remote side is able to open the channel, it returns a SSH_MSG_ **CHANNEL_OPEN_CONFIRMATION** message, which includes the sender channel number, the recipient channel number, and window and packet size values for incoming traffic. Otherwise, the remote side returns a **SSH_MSG_CHANNEL_OPEN_ FAILURE** message with a reason code indicating the reason for failure.
7. Once a channel is open, **data transfer** is performed using a SSH_MSG_ CHANNEL_DATA message, which includes the recipient channel number and a

block of data. These messages, in both directions, may continue as long as the channel is open.

8. When either side wishes to **close a channel,** it sends a SSH_MSG_CHANNEL_ CLOSE message, which includes the recipient channel number.

5.3.2 Channel Types

Four channel types are recognized in the SSH Connection Protocol specification.

- **Session:** The remote execution of a program. The program may be a shell, an application such as file transfer or e-mail, a system command, or some built-in subsystem. Once a session channel is opened, subsequent requests are used to start the remote program.
- **X11:** This refers to the X Window System, a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers. X allows applications to run on a network server but to be displayed on a desktop machine.
- **Forwarded-tcpip:** This is remote port forwarding, as explained in the next subsection.
- **Direct-tcpip:** This is local port forwarding, as explained in the next subsection.

5.3.3 Port Forwarding NOV-2020[8M],SEP-2020[M]

- One of the most useful features of SSH is port forwarding. In essence, port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. This is also referred to as SSH tunnelling. t.
- A **port** is an identifier of a user of TCP.
- So, any application that runs on top of TCP has a port number.
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number. An application may employ multiple port numbers.

FIGURE 12 ILLUSTRATES THE BASIC CONCEPT BEHIND PORT FORWARDING.

- We have a client application that is identified by port number *x* and a server application identified by port number *y*.
- At some point, the client application invokes the local TCP entity and requests a connection to the remote server on port *y*.

- The local TCP entity negotiates a TCP connection with the remote TCP entity, such that the connection links local port *x* to remote port *y*.
- To secure this connection, SSH is configured so that the SSH Transport Layer Protocol establishes a TCP connection between the SSH client and server entities, with TCP port numbers *a* and *b*, respectively.

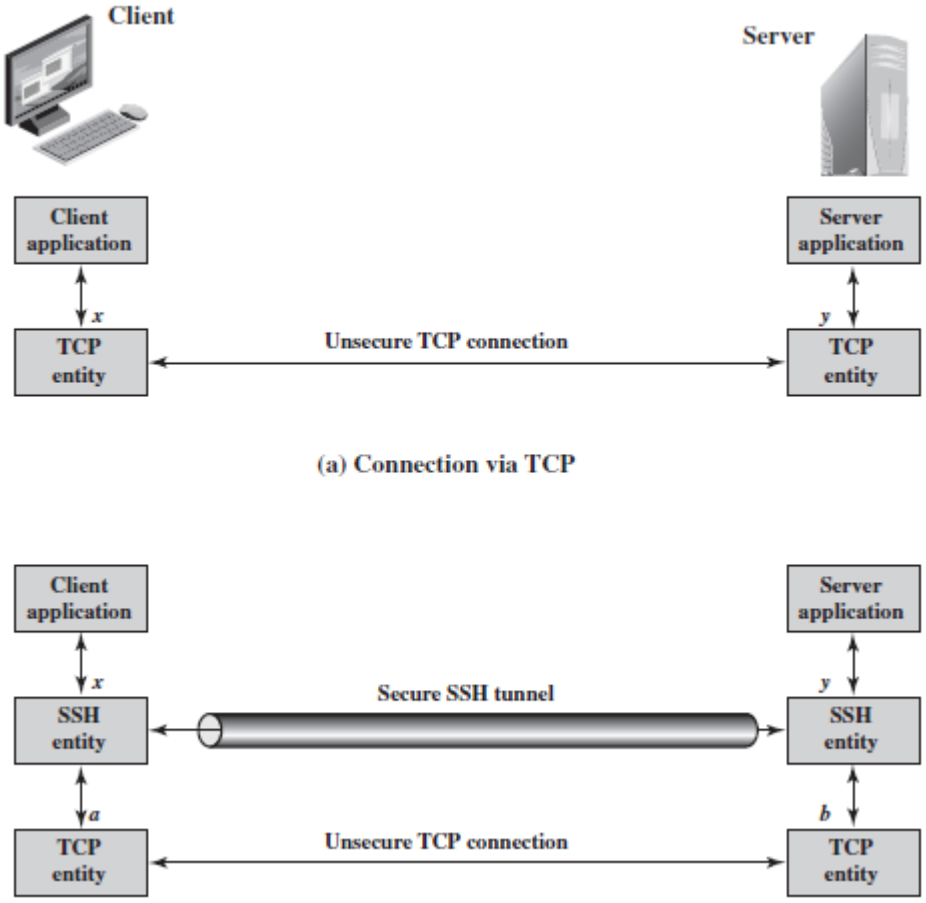


Figure 12 SSH Transport Layer Packet Exchanges

- A secure SSH tunnel is established over this TCP connection.
- Traffic from the client at port *x* is redirected to the local SSH entity and travels through the tunnel where the remote SSH entity delivers the data to the server application on port *y*.
- Traffic in the other direction is similarly redirected.
- SSH supports two types of port forwarding: **local forwarding and remote forwarding**.

- **LOCAL FORWARDING :**
  - allows the client to set up a "hijacker" process. This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.
  - The following example should help clarify local forwarding. Suppose you have an e-mail client on your desktop and use it to get e-mail from your mail server via the Post Office Protocol (POP). The assigned port number for POP3 is port 110.
  - We can secure this traffic in the following way:

1. The SSH client sets up a connection to the remote server.
2. Select an unused local port number, say 9999, and configure SSH to accept traffic from this port destined for port 110 on the server.
3. The SSH client informs the SSH server to create a connection to the destination, in this case mail server port 110.
4. The client takes any bits sent to local port 9999 and sends them to the server inside the encrypted SSH session. The SSH server decrypts the incoming bits and sends the plaintext to port 110.
5. In the other direction, the SSH server takes any bits received on port 110 and sends them inside the SSH session back to the client, who decrypts and sends them to the process connected to port 9999.

- **REMOTE FORWARDING**
  - The user's SSH client acts on the server's behalf. The client receives traffic with a given destination port number, places the traffic on the correct port and sends it to the destination the user chooses.
  - A typical example of remote forwarding is the following. You wish to access a server at work from your home computer. Because the work server is behind a firewall, it will not accept an SSH request from your home computer. However, from work you can set up an SSH tunnel using remote forwarding. This involves the following steps.

1. From the work computer, set up an SSH connection to your home computer. The firewall will allow this, because it is a protected outgoing connection.
2. Configure the SSH server to listen on a local port, say 22, and to deliver data across the SSH connection addressed to remote port, say 2222.

3. You can now go to your home computer, and configure SSH to accept traffic on port 2222.
4. You now have an SSH tunnel that can be used for remote logon to the work server.

QUESTION BANK – NETWORK AND CYBER SECURITY

MODULE-1

MAY/JUNE-2010

1. Explain secure socket layer (SSL) protocol stack with a neat diagram and define the different parameters used in session and connection states. (10M)

DEC-2010

1. Explain the various phases of SSL handshake protocol**.** (12M)

JUN/JULY-2017

1. Discuss security socket layer (SSL) record protocol in terms of fragmentation, compression and encryption. (10M)

JUNE/JULY-2011

1. Explain the two SSL concepts with their parameters. (10M)

DEC-2011

1. With a diagram, explain handshake protocol action.(8M)
2. Explain SSL protocol stack. (4M)

JUNE-2012

1. Discuss SSL record protocol in terms of fragmentation, compression and encryption.(10M)

DEC-2012

1. Explain SSL architecture with neat diagram. (10M)
2. What is the difference b/w SSL connection and SSL session? (04M)

JUNE/JULY-2013

1. List different types of threats and consequence when using the web. Also countermeasures to be taken. (08M)
2. Elucidate SSL architecture. (08M)

JAN-2015

1. Explain the various phases of SSL handshake protocol. (10M)

DEC/JAN-2016

1. Explain the SSL architecture. (10M)

JUNE/JULY-2019

1. Explain the operation of SSL record protocol with a neat sketch.
2. Explain SSH transport layer protocol packet formation with Neat Sketch
3. Explain the 4 Phases of Handshake Protocol with a diagram
4. Describe SSL connection and SSL session detail.

DEC-2019/JAN-2020

1. Define various parameters that are associated with session state and connection state of SSL Protocol.
2. Explain the Additional alert codes in TLS over SSLVs. Describe SSL record protocol
3. With relevant diagram explain the various phases of handshake protocol.
4. Discuss sequence of steps involved during message exchange in user authentication protocol of SSH.

AUG/SEP-2020

1. Differentiate b/w SSL connection SSL session.
2. Discuss the overall operation of SSL Record Protocol.
3. What is port forwarding? Discuss the two types of port forwarding supported by SSH Protocol.
4. Explain the SSL Handshake Protocol Action.
5. Discuss the SSH protocol stack in Details.

NOV-2020

1. Write the comparison of threats on the web
2. What is port forwarding? Explain local and remote forwarding.
3. Explain different phases in a SSL Handshake Protocol
4. Explain the following with respect to transport layer security:
   a) Pseudorandom function
   b) Alert codes

KEY POINTS
(MODULE-1)

NETWORK :-

→ A Network is two (or) more joined together via a Switch, Communicating via a Routing protocol.

(or)

→ One System is talk to other System to Exchange the Some Information is Called Network

→ The biggest World Network is INTERNET

Data Network (or) Computer network :-

→ Computer Network is Called Data Network.

→ The Inter Connection b/w Computer and other devices

Cyber Security :-

Cyber Security is the protection of Internet Connection Systems, Including Hardware, Software and data from Cyber Attacks.

Cryptography :-

→ The word Cryptography was derived from Combining 2 greek words - "Krypto" it means "hidden" and "graphee" means "writting"

→ Cryptography is the art of Secrete Information - hiding (or) Secrete data hiding.

→ The main goal of Cryptography is data Secure from Unauthorized person (or) hackers.

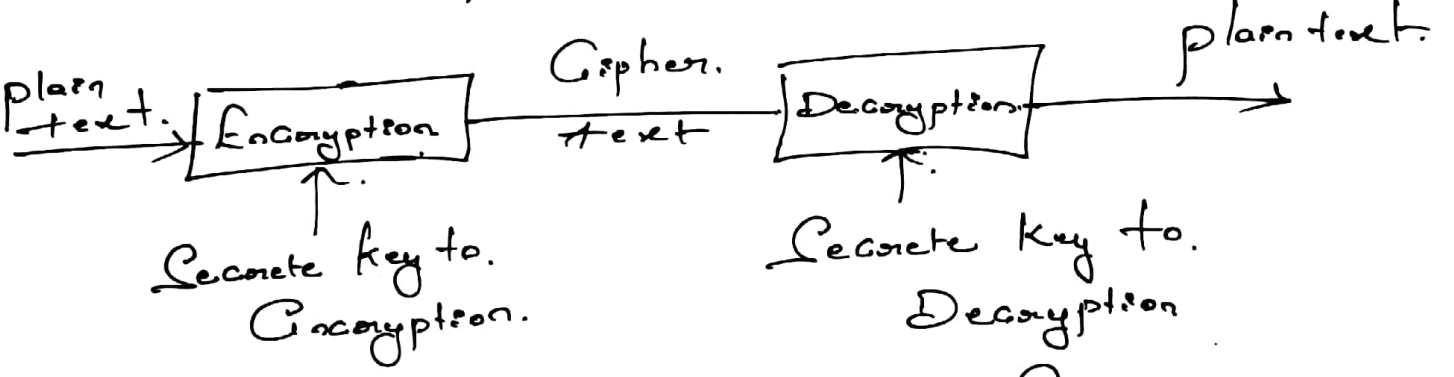


Fig :- Cryptographic flow.

→ Encryption is a Technique for Transforming plaintext into an Unreadable Cipher text.

→ Decryption is a Technique for Transforming Cipher text into to plaintext (or) original data.

→ The key is also a group of bits which as a Major Role in the process of Encryption and Decryption.

→ Types of Cryptography.

* Symmetric key cryptography :- is. Called Secrete key Cryptography (or) private. key Cryptography.
It Uses a Same key (or) Single key for both of Encryption and Decryption method.

* Asymmetric key Cryptography :-
Where a different key used for Encryption. and decryption.

* Hash function :-
Uses no key for Encryption and decryptio

→ Application.
* online banking
* online Transaction
* media Application. E.t.c.

Steganography :

→ The steganography comes from Greek words.

→ Stegano is Greek meaning "hidden." (or) "Covered" and graphic is Greek meaning "written".

→ Steganography is the Invisible Communication.

→ The main Idea of Steganography is to Hide Secret messages in the other. Cover digital media such as text, video and Audio, Image etc.

S.T. Someone (or) hacker (or) other person cannot know the presence of the Secret Info.

→ There are Three basic types of Steganography

1) pure Steganography
2) Secret key Steganography
3) public key steganography.

Dual Steganography :

→ is the process of using Steganography Combined with Cryptography.

→ Dual Steganography is the process of hiding Confidential data's in the media files. Such as Audio, Images, video. etc //

Web Security :

→ Web Security means providing the security for the data which is transmitted to the network.

Ex: Client and Server.

→ The client will send the Request to the Server. The Server will provide to the Server to the client so for. This purpose we will use a one protocol i.e called SSL protocol.

SSL :

→ i.e means Secure Socket layer by Implementing This SSL we can provide the security for the data which is transferred b/w the web browser and Server.

→ This SSL is Implemented by using a different protocols. (or) It includes a different protocol

1) SSL Record protocol.
2) Hand Shake protocol.
3) Change Cipher Spec protocols
4) Alert protocol.

→ So all this protocols will be included SSL protocol.

→ So. This we will call it as SSL protocol Stack.

→ SSL will be Implemented at just above the TCP/IP and just below the HTTP.

→ SSL 2 main Concepts. → Connection.
→ Session.

Connection :- → Transport to provide the Service b/w Client and Server..

[Eg :- First Connection should be established. b/w the Client and Server S.T client will Communicate with Server and Server will Communicate with the Client]

Session :- → Association b/w a Client and Server.

[based upon the Session, the client and Server will be Communicated]

→ The Session will call it as a Temporary Time period.

→ The Session Consists of a multiple Connection

→ The Session Created by Handshake protocol //

Session State parameters :-

1) Session Identifier :- An arbitrary byte sequence chosen by the server to identify an active (or) resumable session state.

2) peer Certificate :-
   → peer is nothing but a client
   → An X509.v3 Certificate of the peer.

3) Compression method :- The algorithm used to compress data prior to Encryption.

4) Cipher Spec :-
   → means Specification.
   → Cipher Specifies the bulk data Encryption Algorithm (such as null, DES, etc) and a hash algorithm (such as MD5 (or) SHA-1) used for MAC Calculation.

   Note :-
   Bulk → means big
   MAC → message Authentication Code.

5) master Secrete :-
   → 48 byte Secrete shared b/w the client & server (which is the Secrete key shared b/w client and server)

master Secrete.

6) Is Resumable :-

A flag indicating whether the session can be used to initiate new connections.

Connection State parameters :-

1) Server and Client Random :

Byte sequences that are chosen by the server and client for each connection.

2) Server Write MAC Secret :

The secrete key used in MAC operations on data sent by the server.

3) Client Write MAC Secret :

The secrete key used in MAC operations on data sent by the client.

4) Server Write key :

The conventional encryption key for data encrypted by the server and decrypted by the client

5) Client Write key :-

The conventional encryption key for data encrypted by the client and decrypted by the server.

6) Initialization vector : when a block cipher in CBC mode is used, an initialization vector is maintained for each key.

7) Sequence numbers : Each party maintains separate sequence numbers for transmitted and received msg for each connection

Secure Socket Layer. (SSL)

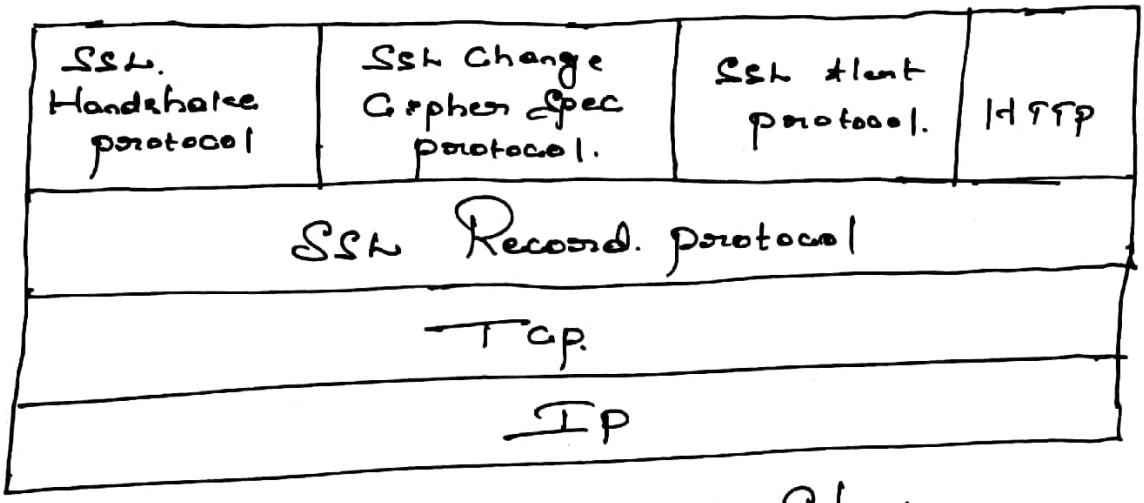


Fig:- SSL protocol Stack.

SSL Record. protocol. :-

→ The data fragmented and the Calculation of MAC (or) the Encryption Algorithm (or) the Compression Algorithm will be Implemented in the SSL. Record protocol.

Hand Shake protocol :- is used to established the Session and this Handshake protocol is used. Authenticate the client with the Server and Server with the client.

Change Cipher-Spec. protocol :- is for Changing the State i.e means the pending State to the Current State.

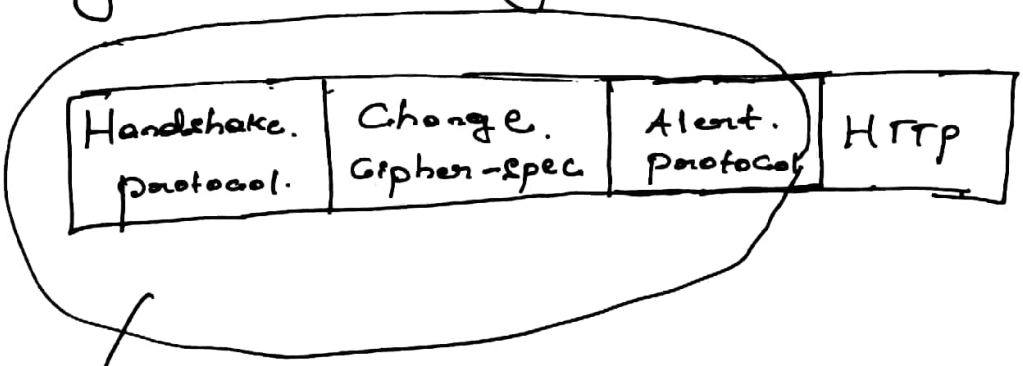
NOTE:

MAC → media Access Control address

→ 48-bit hexadecimal address.

→ For example. 70-54-D2-AB-EF-85

Alert protocol :- This Alert protocol for giving the Alerts by Implementing the SSL.



So all this protocols involved in the SSL.

SSL Record protocol :-

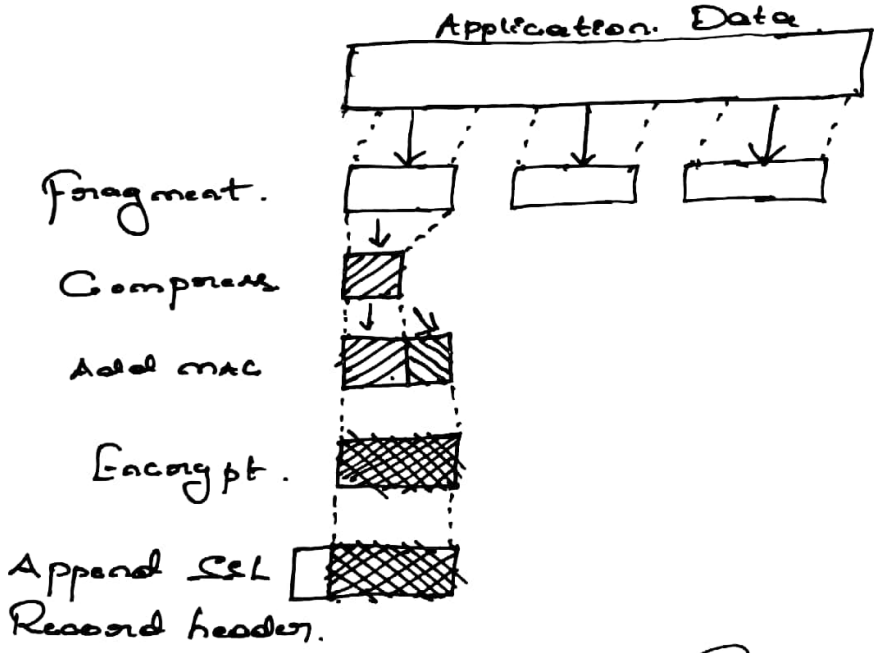


Fig :- SSL Record protocol operation.

Step 1 :-

⇒ The Application Data will be divided into Different fragments. i.e means fragmentation will be done. According to the bandwidth.

Step 2 :-

⇒ Consider the 1st fragment and now apply the Compression function.

(i.e fragment will be Compression format)

(compression — optional frag)

Step 3 :-

→ Next Calculate the MAC by using one of the Authentication Algorithms.

Ex:- Ice Con. use. MD5 Algorithm (or) SHA-1 Algorithm.

Step 4 :-

→ The MAC will be Calculated, So. That MAC will be appended to the Compression function.

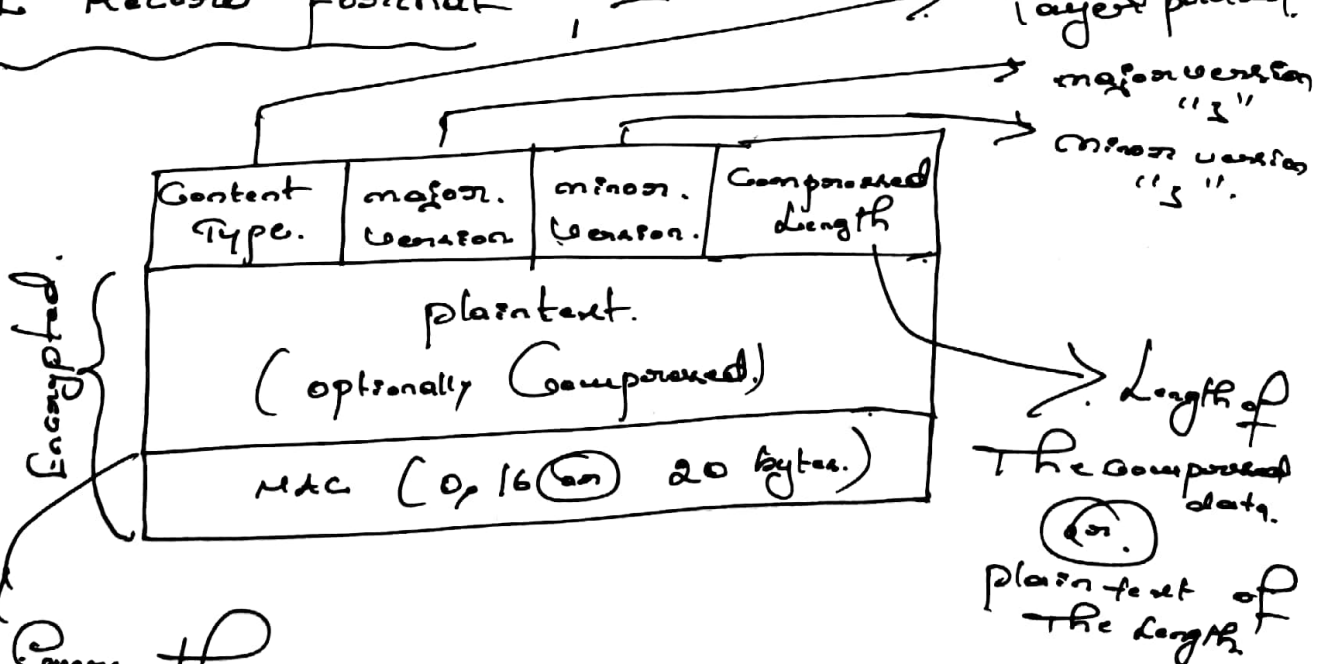
Step 5 :-

→ Next we Encrypt the MAC and Compressed data, So. Encryption done on the fragment.

Step 6 :-

→ After Encryption done, now append the SSL Record header to this fragment i.e means. In order to processing to the higher level protocol.

SSL Record Header (or) SSL Record format :-



There are mainly 4 fields. Involved in the SSL Record Header.

1) Content type. (8bits)
   -> The Higher layer protocol used to process the Enclosed fragment.

2) major Version (8bits)
   -> Indicates major version of SSL in use. For SSLv3 the value is 3

3) Minor Version (8bits)
   -> Indicates minor version of SSL in use. For SSLv3 the value is "0"

4) Compressed Length (16bits).
   -> The Length in bytes of the plaintext fragment. (or Compressed fragment if Compression is used)

Now Let us See the Equation by Calculating the MAC.

hash (MAC-write-Secret || pad-2 ||
hash (MAC-write-Secret || pad-1 || Seq-num ||
SSL Compressed Type || SSL Compressed Length ||
SSL Compressed fragment))

Where.

|| → Concatenation.

MAC_write_Secret → Shared Secret key
↳ This is the Shared Secret key b/w the Client and Server.

hash → Cryptographic hash Algorithm ; either MD5 (or) SHA-1.

pad_1. → The byte 0X36 (0011 0110) Repeated 48 times for MD5 and 40 times (320 bits) for SHA-1.

↳ The pad-1 means 0011 0110 Repeated 48 times for MD5 and 40 times for SHA-1

pad_2 → The byte 0X5C (0101 1100) Repeated 48 times for MD5 and 40 times for SHA-1

↳ If you are applying the MD5 algorithm. The byte Sequence (0101 1100) will be Repeated 48 times S.T Hash to generate the MAC.

↳ Ily the SHA-1 Algorithm the byte Sequence (0101 1100) will be Repeated 40 times. S.T Hash to generate the SHA-1

Seq_num → The Sequence number for this message.

SSL Compressed.type → The higher-level protocol used to process this fragment.

SSL Compressed.length → The length of the Compressed fragment.

SSL Compressed.fragment → The Compressed fragment. (if Compression is not used, This is the plaintext fragment).

Handshake protocol :

→ The main purpose of this Handshake protocol is to established the Session

→ Handshake protocol, means the Session will be generated by this Handshake protocol.

→ Handshake protocol can be Represented by three fields.

first → Type

second → Length.

3rd → Content

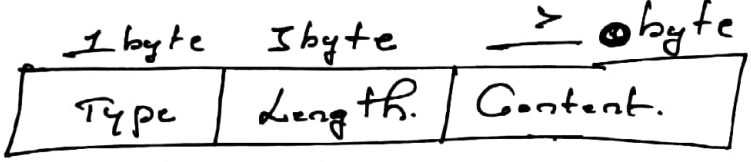


fig:- Handshake protocol.

Where. Type. → Represents the higher layer protocol

Length → Represents the Length of message

Content → means parameters associated with the particular msgs.

→ The Different messages Involved in the Handshake protocol

1) Client Hello.
2) Server Hello.
3) Certificate as X.509
4) Server key exchang
5) Certificate Request
6) Server done
7) Client_Key_Exchange.
8) Certificate_Verify
9) Finished.

So this are the different messages Involved in the Handshake protocol

HANDSHAKE PROTOCOL : XXX IM. OR 8 OR 10M.

→ The main purpose of this Handshake protocol is to Established the Session.

OR

The Session will be generated by this Handshake protocol

→ Handshake protocol is used to Authenticate the Client with Server and Server with Client.

→ The Handshake protocol is used before any Application data is Transmitted.

→ The Handshake protocol Consists of a Series of messages Exchanged by Client and Server. (as shown in below Handshake protocol Action - figure 2).

→ Handshake protocol Can be Represented by three fields

First → Type
Second → Length.
3rd → Content

1 byte.	3 byte.	≥ 0 byte.
Type	Length.	Content

Fig:- Handshake protocol

TYPE (1 byte) :- Represents the Higher Layer protocol

OR

Indicates one of 10 messages of Handshake protocol. (as shown in below Table)

LENGTH (3 bytes) :- Represents the length of message

OR

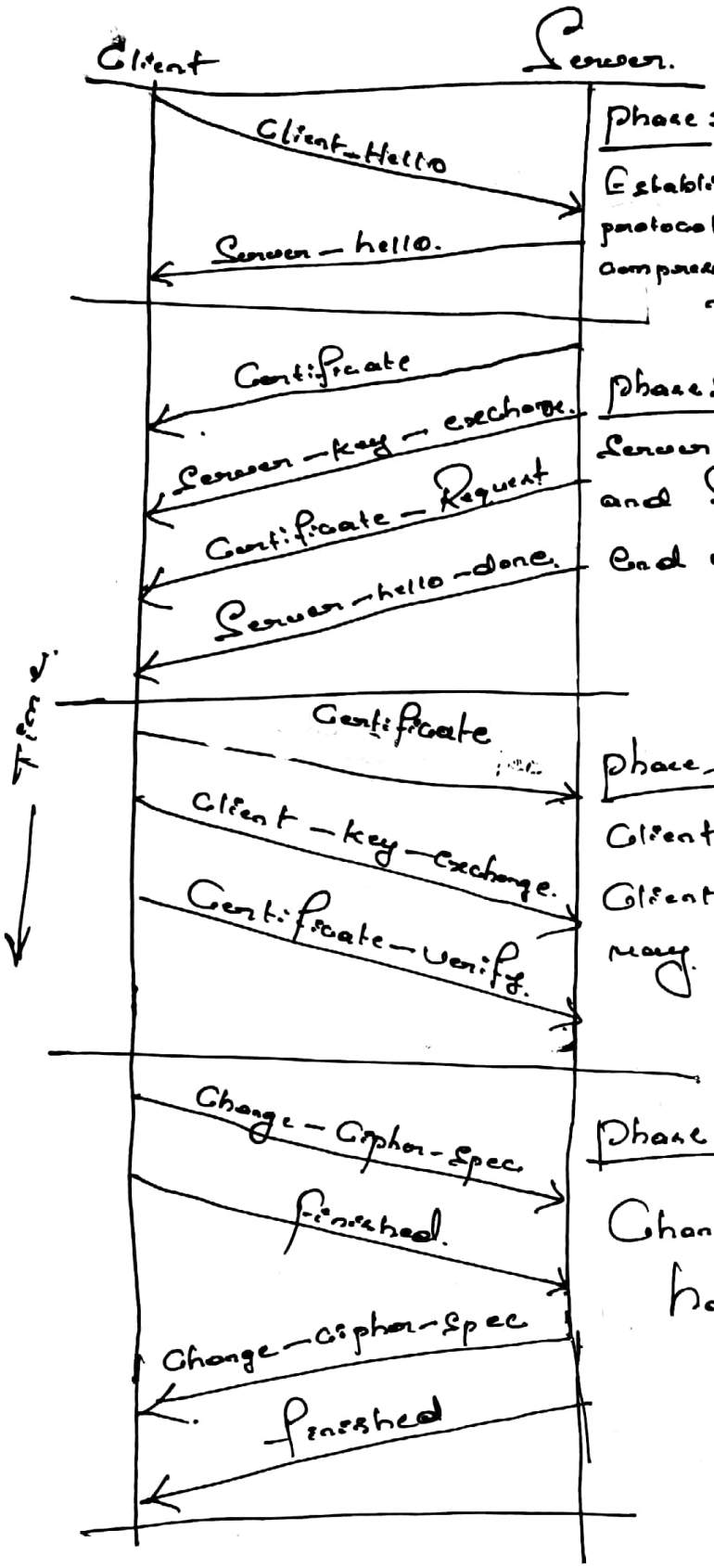
The length of the message in bytes.

CONTENT (byTes) :- Means parameters Associated with this message.

→ The Different Messages Involved in The Handshake Protocol.

Message Type.	Parameters
1) Hello — request.	null
2) Client — hello	version, Random, Session id, Cipher Suite. Compression method.
3) Server — hello	version, Random, Session id, Cipher Suite Compression method
4) Certificate	Chain of X.509V3 Certificates.
5) Server — key — Exchange	parameter, Signature
6) Certificate — Request	Type, Authorities.
7) Server — done.	null
8) Certificate — verify	Signature.
9) Client — key — Exchange.	Parameters, Signature.
10) Finished.	hash value.

So this are the different Messages Involved in the Handshake protocol



Phase 1 :-
Established Security Capabilities. Including protocol version, Session ID, Cipher Suite, compression method, and initial Random numbers.

Phase 2. :-
Server may send Certificate, key exchange, and Request Certificate. Server signals end of hello message phase.

Phase 3 :-
Client sends Certificate if requested. Client sends key exchange. Client may send Certificate verification.

Phase 4 :-
Change Cipher Suite and finish handshake protocol.

Fig :- Handshake protocol Action

CHANGE CIPHER SPEC PROTOCOL :-

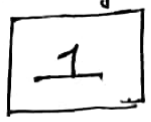


Fig:- Change Cipher Spec protocol.

* The Change Cipher Spec protocol is one of the three SSL-Specific protocols that use the SSL Record protocol and it is the simplest
* This protocol consists of a single message (as shown in above fig) which consists of a single byte (1 byte) with the value 1.
* The sole (one only) purpose of this message is to cause the pending state to be copied into current state, which updates the Cipher Suite to be used on this Connection.

NOTE :-

CipherSuite is basically a Complete Set of methods. (or) is a Set of Cryptographic Algorithms (or) it's a Complete Set of Instruction. Needed to Secure a network Connection through SSL

ALERT PROTOCOL : 3 (or) 4 M.

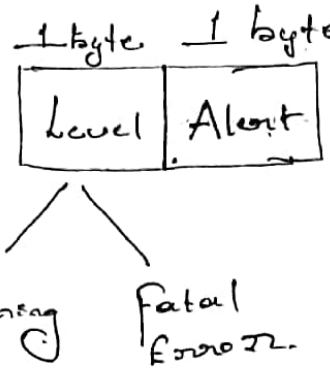


Fig:- Alert protocol

* The Alert protocol used to convey SSL Related Alerts to peer entity
* Each message in this protocol consists of two byte.
* The first byte takes the value Warning (1) (or) fatal (2) to convey the severity of msg.

* Level Represents either Warning (or) fatal error
* If it's a Warning there is no Impact on our Connection.
* If it's a fatal error Automatically the Connection b/w The client and Server will be disconnected, So we have to Re-establish the Connection.

(or)

* If the level is fatal, SSL Immediately terminates the Connection
* other Connection on the Same Session may Continue. but no New Connections Established.
* The Second byte Contains a Code that Indicates the Specific Alert.
* We list those Alerts that are always fatal

1) Unexpected Messages

An inappropriate message was Received.

(or)

Receiving the inappropriate msg.

2) Bad_Record_MAC :- An Incorrect MAC Was Received.

3) Decompression_failure :- The decompression function Received Improper Input.
(Or. Decompression is failure in the Receiver Side)

4) HandShake_failure :- HandShake failure means - Authentication failure.
So. Handshake Mainly for Authenticated to Client to Server (or) Server to Client. If Any one Authentication is failed. means Handshake failure

5) Illegal_parameter :-
Field in a Handshake message was out of Range (or) Inconsistent with other fields.

* The Remaining Alerts are the following

1) Close_notify :- Notifies the Recipient that the Sender will not Send any more Message on this Connection

2) No_Certificate :- may be Sent in Response to a Certificate Request if no Appropriate Certificate is Available

3) Bad — Certificate :- A Received Certificate has Corrupt.

4) Unsupported — Certificate :- The Type of the Received Certificate is not Supported.

5) Certificate — Revoked :- A Certificate has been Revoked by its Signer.

6) Certificate — Expired :- A Certificate has Expired.

7) Certificate — Unknown :- Some other unspecified issue arose in processing the Certificate, Rendering it unacceptable.

TRANSPORT LAYER SECURITY (TLS)

→ TLS is an Internet Engineering Task force (IETF) Standard protocol that provides Authentication, privacy and data Integrity b/w two Communicating Computer Applications

→ It's mostly widely deployed Security protocol in use today and is best Suited for web browsers and other Applications that Require data to be Securely Exchanged over a network.

→ TLS is an IETF Standardization Initiative whose goal is to produce an Internet Standard Version of SSL

→ TLS is defined as a proposed Internet Standard in RFC 5246.

[NOTE:- * TLS 1.2 was defined in RFC 5246

*. The IETF is a Large open International Community of network designers, operators, vendors, and Researchers Concerned with the Evolution of the Internet Architecture and the Smooth operation of the Internet]

→ The RFC 5246 is very Similar to SSLV3.

1) VERSION NUMBER:-

*. The TLS Record format is the Same as that of the SSL Record format and the fields in the Header have the Same Meanings.

*: The one difference is in Version values. For the Current Version of TLS the major Version is 3 and the Minor Version is 3.

2) MESSAGE AUTHENTICATION CODE:-

* There are two differences between the SSLV3 and TLS MAC Schemes.

1) The Actual Algorithm and the Scope of the MAC Calculation.

2) TLS Makes use of the HMAC algorithm defined in RFC 2104

[NOTE

RFC 2104 means : HMAC : Keyed - Hashing for Message Authentication.]

* HMAC is defined as

$$HMAC_K(M) = H[(K^+ \oplus opad) \,||\, H[(K^+ \oplus ipad)||M]]$$

where

H = Embedded Hash function [for TLS, either MD5 or SHA-1]

M = Message I/p to HMAC

K^+ = Secret key padded with zeros on the left so that the result is equal to the block length of the Hash Code (for MD5 and SHA-1, block length = 512 bits)

ipad = 0011 0110 (36 in hexadecimal) repeated 64 times (512 bits)

opad = 0101 1100 (5C in hexadecimal) repeated 64 times (512 bits)

* For. TLS the MAC Calculation Encompasses the fields indicated in the following Expression

MAC (MAC — Write — Secrete, Seq — Num || TLS Compressed. Type || TLS Compressed. Version || TLS Compressed. Length || TLS Compressed. fragment)

* The MAC Calculation. Covers all of the fields Covered by the SSLV3 Calculation, plus the field. TLS Compressed. version which is the version of the protocol being employed.

*** PSEUDORANDOM FUNCTION :- [IOM].

→ TLS makes use of a pseudorandom function. Referred to as. PRF. to Expand Secrets into. blocks of data. for. purpose of key Generation. (or) Validation

NOTE

PRF (pseudorandom function) is mainly based on the data. Expansion function.

→ The objective is to. Make use of a relatively small shared. value but. to generate longer blocks of data in a way that is Secure. from the kinds of attacks made on Hash functions and MACs.

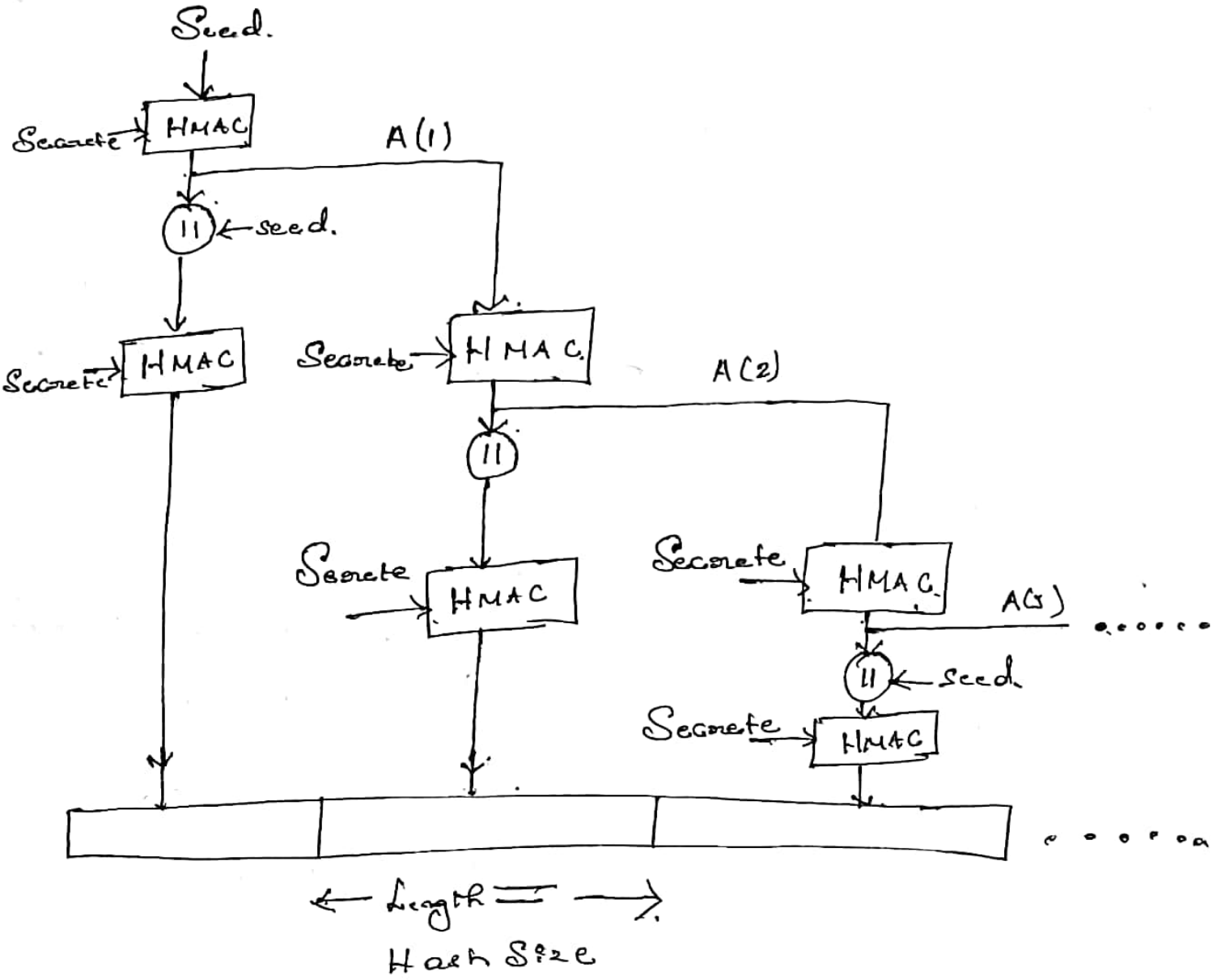


Figure : TLS function. P-hash (secret, Seed)

→ The data expansion function is given as.

P_hash (secret, seed) = HMAC-hash (secrete, A(1) || seed) ||
HMAC-hash (secrete, A(2) || seed) ||
HMAC-hash (secrete, A(3) || seed) || ....

Where A( ) is defined as.

A(0) = seed.

A(i) = HMAC-hash (secrete, A(i-1))

[NOTE :-

Seed (or) Random Seed :- A Random Seed (or seed state (or) just seed) is a Number (or vector) used to Initialize a pseudorandom number generator.]

→ the data Expansion function makes use of the HMAC Algorithm both Either MD5 (or) SHA-1 as the underlying Hash function.

→ As Can be seen (above equation) P-hash Can be iterated as many times as necessary to produce the required quantity of data.

(Repeatedly).

→ For Example :-

* If P-SHA-1 was used to generate 64 bytes of data, it would have to be iterated four times, producing 80 bytes of data of which the last 16 would be discarded.

[NOTE :- If SHA-1 produce 20 bytes → 20 × 4 = 80 bytes
(1st time) (4 times)

We need only 64 bytes of data.

∴ SHA-1 (4 times) = 80 − 16 (discarded bytes (last bytes))

= 64 bytes

* If P_MD5 would also have to be iterated four times, producing exactly 64 bytes of data.

[NOTE:- MD5 produce 16 bytes. = 16 X 4 = 64 bytes
(1st time). (4 times) (Exactly 64 bytes produced)]

→ PRF is defined as.

PRF (secrete, Label, Seed) = P_hash (S1, label || seed)

PRF takes as input a secrete value, an identifying Label, and a seed value, and produces an o/p of arbitrary length.

NOTE (Additional Information)

HTTP :- HTTP (Hyper Text Transfer protocol) is the set of Rules for Transferring files - such as Text, Image, Sound, video and other Multimedia files over the Web. As soon as a User open their web browser they are Indirectly using HTTP. HTTP is an Application protocol that runs on top of the TCP/IP Suite of protocols which forms the foundation of Internet. Through the HTTP protocol, Resources are Exchanged b/w client devices and Servers over the Internet.

NOTE (Additional Info)

[HTTP :- * The Standard protocol for Transferring Hypertext documents on the World Wide Web.

* Communication b/w client Computers and Web Servers is done by Sending HTTP Request and Receiving HTTP Responses.]

HTTPS : *** 10M.

→ Hyperlext Transfer protocol Secure (HTTPS) is an Extension of the HTTP.

[Or]

It is highly Advanced & Secure Version of HTTP.

→ HTTPS (HTTP over SSL) refers to the Combination of HTTP and SSL to implement Secure Communication b/w a Web browser and Web Server.

→ The HTTPS Capability is built into all modern Web browsers. It uses dependent on the Web Server Supporting HTTPS Communication.

→ The principal difference Seen by a User of a Web browser is that URL (Uniform Resource Locator) addresses begin with Https:// rather than Http://

→ A normal HTTP Connection uses port 80.

→ If HTTPS is specified, port 443 is used. Which invokes SSL or TLS.

NOTE
[PORT :- A port is a logical Construct that Identifies a specific process or Type of network Service]

→ When HTTPS is used, the following elements of the Communication are Encrypted.

* URL of the Requested document
* Contents of the documents
* Contents of browser forms (filled in by browser User)
* Cookies Sent from browser to Server and from Server to browser
* Contents of HTTP header.

NOTE
[Cookies :- are text files with small pieces of data like User name and password.
HTTP Cookies are used to Identify Specific users]

→ HTTPS is documented in RFC 2818. HTTP over TLS. There is no fundamental change in using HTTP over either SSL or TLS and both implementations are referred to as HTTPS

Connection Initiation :-

→ Initiation steps :-

STEP 1 :- The client Initiates a Connection to the Server on the appropriate port and then Sends the TLS Client Hello to begin the TLS Handshake.

STEP 2 :- When TLS Handshake has finished the Client may then initiate the first HTTP request.

STEP 3 :- All HTTP data is to be Sent as TLS Application data.

→ There are three level of awareness of a Connection in HTTPS

Level 1 :- At the HTTP level, an HTTP Client requests a Connection to an HTTP Server by Sending a Connection Request to the next lower layer.

Level 2 :- Typically the next lowest layer is TCP. but it also may be TLS/SSL.

Level 3 :- At the Level of TLS, a session is established b/w a TLS client and TLS Server. This Session Can Support one or more Connections at any time. As we have seen a TLS Request to establish a Connection begins with the establishment of a TCP Connection b/w the TCP entity on the client Side and the TCP entity on the Server Side.

Connection Closure :-

An HTTP Client or Server can Indicate the closing of a Connection by including the following line in an HTTP Record : Connection : Close . This Indicates that the Connection will be closed after this record is delivered.

NOTE :

Difference b/w SSH and SSL.

→ SSH is a Cryptographic network protocol for operating network services securely over an Unsecured network

→ SSH Command provides a Secure Encrypted Connection b/w two hosts over an Insecure network. (practically every Unix & Linux System includes the SSH Command)

→ SSH is used for Creating a Secure tunnel to another Computer.

→ It works on the port number 22.

→ Create a Secure remote link to Server.

Continued,

NOTE :-

Difference b/w SSH and SSL

SSL :-

→ It is a networking protocol which gives Secure Transmission in a non Secure network

→ SSL provide the Security for the data which is Transfered b/w the Web browsers and Server

→ Creates a Secure Connection b/w a website and Users.

→ SSL is used for Securely Transfering data b/w two parties.

→ It works on the port number 443

→ Creates a Secure Connection b/w a website and users //

NOTE :-

Tunnel

Tunneling protocol is a Communication protocol that allows for the movements of data from one network to another //

SECURE SHELLS (SSH) 8m (SEP-2020)

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include **remote command line**, **login**, and **remote command execution**, but any **network service** can be secured with SSH.

SSH provides a secure channel over an unsecured network by using a client–server architecture, connecting an SSH client application with an SSH server. The protocol specification distinguishes between **two major versions**, referred to as **SSH-1** and **SSH-2**. The standard TCP port for SSH is 22. SSH is generally used to access **UNIX-like operating systems**, but it can also be used on **Microsoft Windows10**.

SSH was designed as a replacement for **Telnet** (Telnet is an application protocol used on the Internet or local area network to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.) and for **unsecured remote shell protocols**.

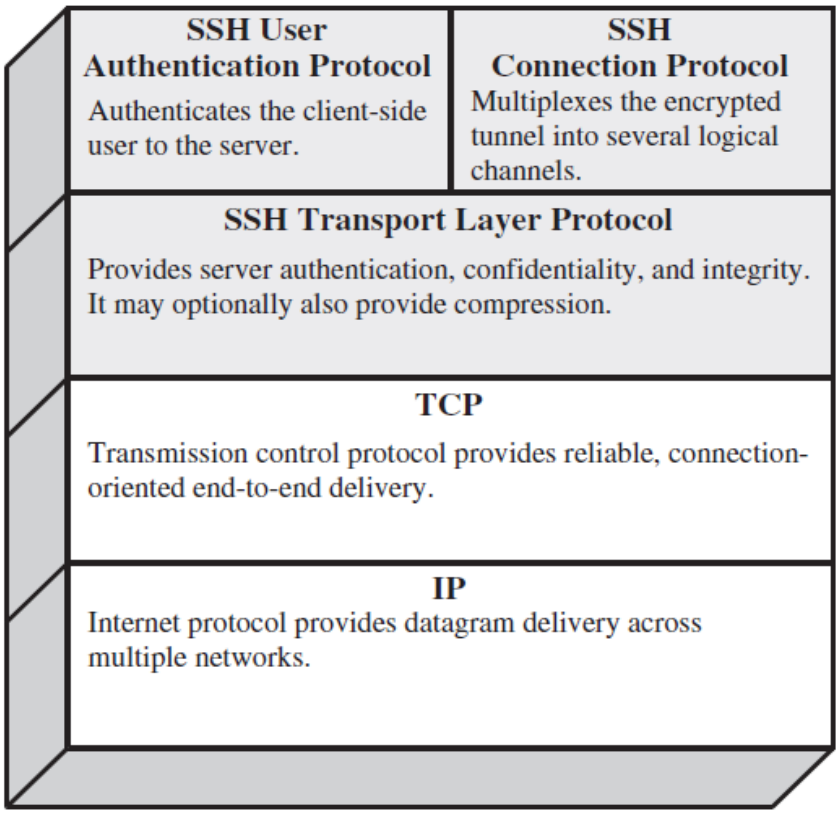


Fig: SSH Protocol Stack

Transport Layer Protocol: which typically runs on top of TCP/IP. This layer handles initial key exchange as well as server authentication, and sets up encryption, compression and integrity verification.

Or

Provides server authentication, data confidentiality, and data integrity with forward secrecy. The transport layer may optionally provide compression.

User Authentication Protocol: This layer handles client authentication and provides a number of authentication methods

Or

Authenticates the user to the server.

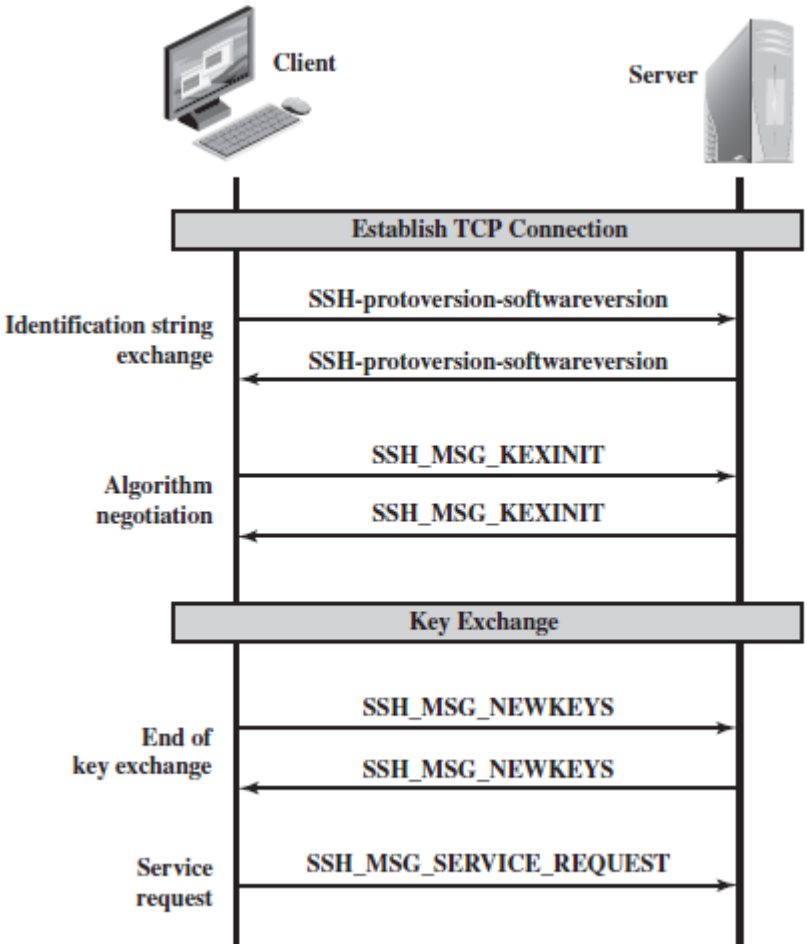
Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.

TCP: Transmission control protocol provides reliable, connection oriented end-to-end delivery.

IP: Internet protocol provides datagram (datagram is a basic transfer unit, associated with a packet-switched network) delivery across multiple networks.

SSH TRANSPORT LAYER PROTOCOL PACKET EXCHANGES 8m

Below Figure illustrates the sequence of events in the SSH Transport Layer Protocol. First, the client establishes a TCP connection to the server. This is done via the TCP protocol and is not part of the Transport Layer Protocol. Once the connection is established, the client and server exchange data, Referred to as packets, in the data field of a TCP segment.



DEC-2019|8M

Figure: SSH Transport Layer Protocol Packet Exchanges

The SSH Transport Layer packet exchange consists of a sequence of steps (Above Figure).

1. **The first step**, the identification string exchange, begins with the client sending a packet with an identification string of the form:

   SSH-protoversion-software version SP comments CR LF

   Where SP, CR, and LF are space character, carriage return, and line feed, respectively

2. **Next comes algorithm negotiatio**n. Each side sends an **SSH_MSG_KEXINIT** containing lists of supported algorithms in the order of preference to the sender. There

is one list for each type of cryptographic algorithm. The algorithms include key exchange, encryption, MAC algorithm, and compression algorithm.

3. The next step is **key exchange**. As a result of these steps, the two sides now share a master key *K*. In addition, he server has been authenticated to the client, because the server has used its private key to sign its half of the Diffie-Hellman exchange. Finally, the hash value *H* serves as a session identifier for this connection. Once computed, the session identifier is not changed, even if the key exchange is performed again for this connection to obtain fresh keys.
4. The **end of key exchange** is signalled by the exchange of **SSH_MSG_NEWKEYS** packets. At this point, both sides may start using the keys generated from *K*, as discussed subsequently.
5. The final step is **service request**. The client sends an **SSH_MSG_SERVICE_REQUEST** packet to request either the User Authentication or the Connection Protocol. Subsequent to this, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

SSH TRANSPORT LAYER PROTOCOL PACKET FORMATION

8m (JUNE/JULY-2019)

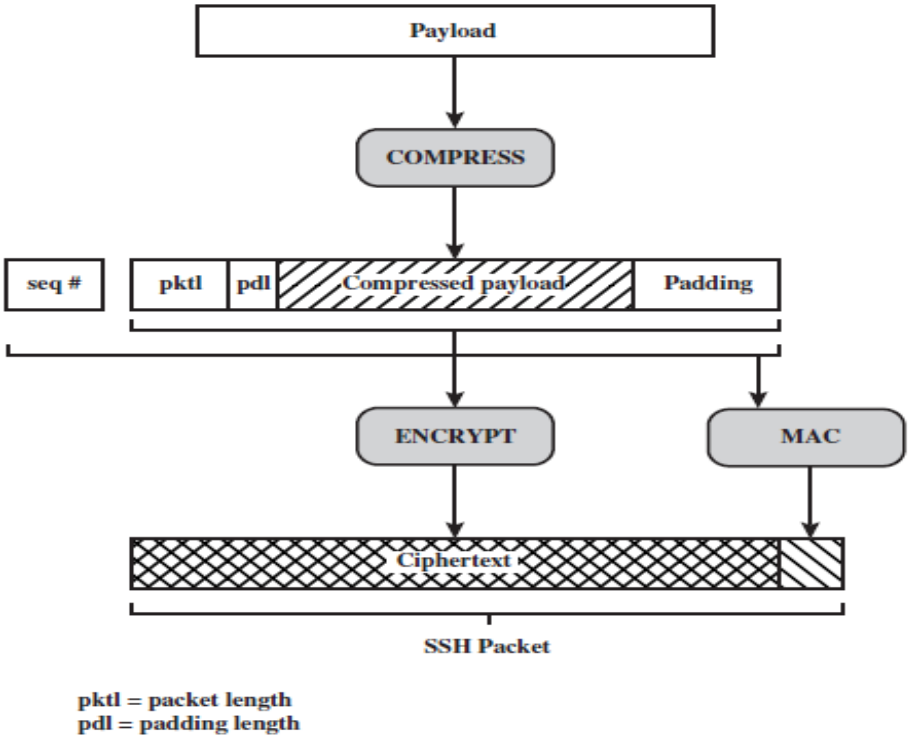


Figure: SSH Transport Layer Protocol Packet Formation

Each packet is in the following Format (Below Figure).

- **Packet length:** Length of the packet in bytes, not including the packet length and MAC fields.
- **Padding length:** Length of the random padding field.
- **Payload:** Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.
- **Random padding:** Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or 8 bytes for a stream cipher.
- **Message authentication code (MAC):**
  - If message authentication has been negotiated, this field contains the MAC value.
  - The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field.
  - The sequence number is an implicit 32-bit packet sequence that is initialized to zero for the first packet and incremented for every packet.

CHANNEL MECHANISM or SSH CONNECTION PROTOCOL MESSAGE EXCHANGE 6m(JAN-2020)

1. Communication using SSH, such as a terminal session, are supported using separate channels. Either side may open a channel.
2. For each channel, each side associates a unique channel number, which need not be the Same on both ends.
3. Channels are flow controlled using a window mechanism.
4. No data may be sent to a channel until a message is received to indicate that window space is available.
5. When either side wishes to **open a new channel,** it allocates a local number for the channel and then sends a message of the form:

Byte	**SSH_MSG_CHANNEL_OPEN**
String	channel type
Uint32	sender channel
Uint32	initial window size
Uint32	maximum packet size

…. channel type specific data follows

Where uint32 means unsigned 32-bit integer

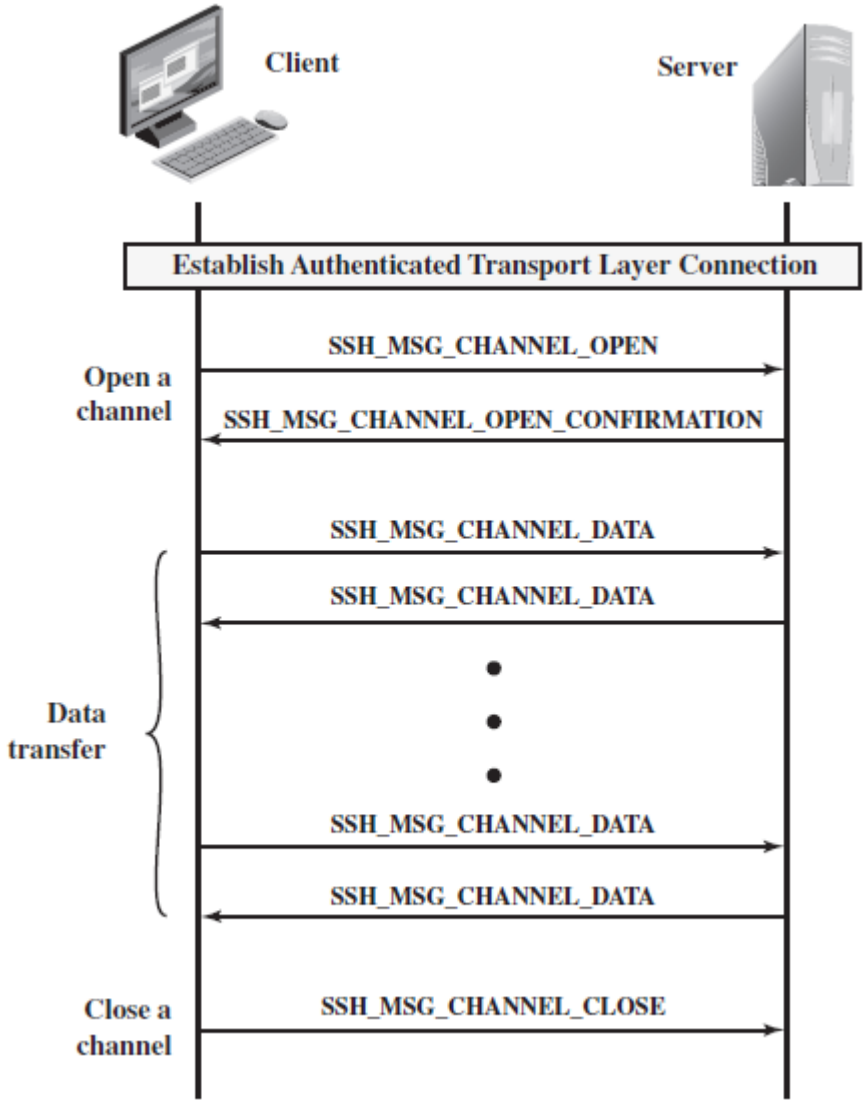


Figure Example of SSH Connection Protocol Message Exchange

6. If the remote side is able to open the channel, it returns a **SSH_MSG_CHANNEL_OPEN_CONFIRMATION** message, which includes the sender channel number, the recipient channel number, and window and packet size values for incoming traffic. Otherwise, the remote side returns a **SSH_MSG_CHANNEL_OPEN_ FAILURE** message with a reason code indicating the reason for failure.
7. Once a channel is open, **data transfer** is performed using a **SSH_MSG_CHANNEL_DATA** message, which includes the recipient channel number and a block of data. These messages, in both directions, may continue as long as the channel is open.

8. When either side wishes to **close a channel,** it sends a SSH_MSG_CHANNEL_ CLOSE message, which includes the recipient channel number.

PORT FORWARDING or SSH TRANSPORT LAYER PACKET EXCHANGES 8m (NOV-2020, SEP-2020)

- One of the most useful features of SSH is port forwarding. In essence, port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. This is also referred to as SSH tunnelling.
- A **port** is an identifier of a user of TCP. So, any application that runs on top of TCP has a port number.
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number. An application may employ multiple port numbers.

Figure (Below) illustrates the basic concept behind port forwarding.

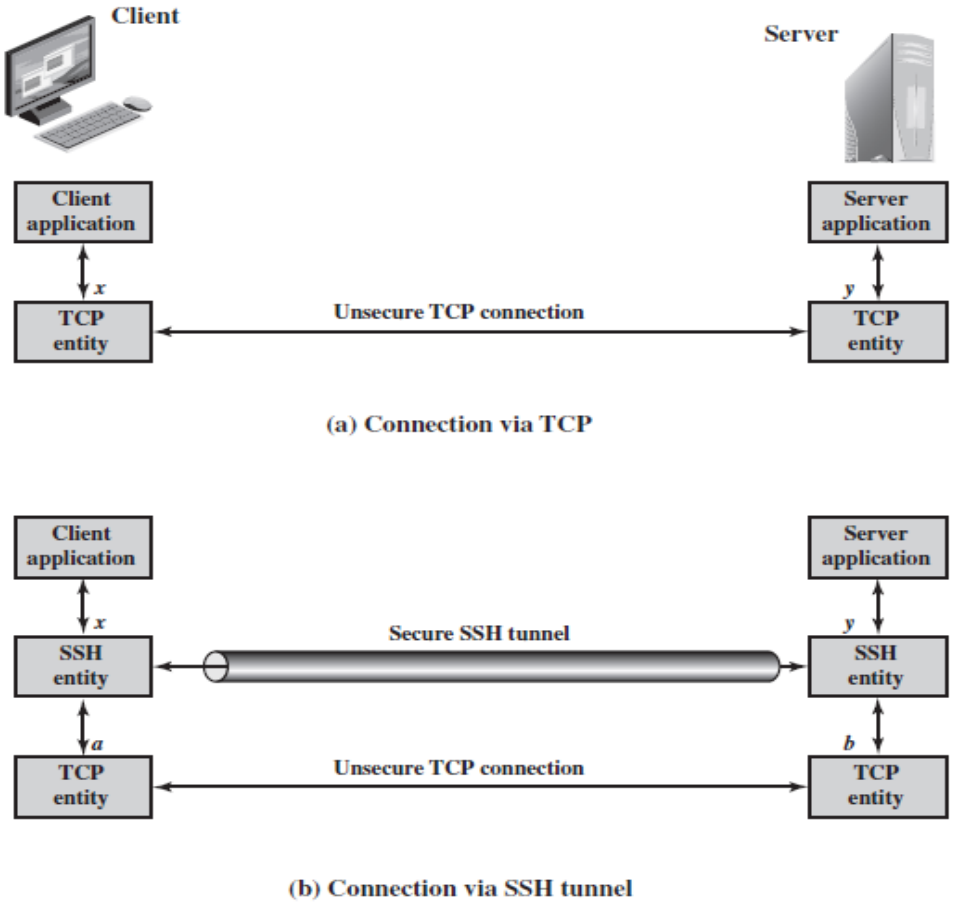


Figure: SSH Transport Layer Packet Exchanges

- We have a client application that is identified by port number *x* and a server application identified by port number *y*.

- At some point, the client application invokes the local TCP entity and requests a connection to the remote server on port *y*.
- The local TCP entity negotiates a TCP connection with the remote TCP entity, such that the connection links local port *x* to remote port *y*.
- To secure this connection, SSH is configured so that the SSH Transport Layer Protocol establishes a TCP connection between the SSH client and server entities, with TCP port numbers *a* and *b*, respectively.
- A secure SSH tunnel is established over this TCP connection.
- Traffic from the client at port *x* is redirected to the local SSH entity and travels through the tunnel where the remote SSH entity delivers the data to the server application on port *y*.
- Traffic in the other direction is similarly redirected.
- SSH supports two types of port forwarding: **local forwarding and remote forwarding**.

LOCAL FORWARDING:

Allows the client to set up a "hijacker" process. This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.

Or

Local forwarding is used to forward a port from the client machine to the server machine. Basically, the SSH client listens for connections on a configured port, and when it receives a connection, it tunnels the connection to an SSH server.

Or

Local port forwarding is the most common type of port forwarding. It is used to let a user connect from the local computer to another server, i.e. forward data securely from another client application running on the same computer as a Secure Shell (SSH) client.

REMOTE FORWARDING

The user's SSH client acts on the server's behalf. The client receives traffic with a given destination port number, places the traffic on the correct port and sends it to the destination the user chooses.

Or

Remote port forwarding is the exact opposite of local port forwarding. It forwards traffic coming to a port on your server to your local computer, and then it is sent to a destination.

ALERT CODES: TRANSPORT LAYER SECURITY 5m (NOV-2020, DEC-2019)

- TLS supports all of the alert codes defined in SSLv3 with the exception of no _ certificate.
- A number of additional codes defined in TLS; of these, the following are always fatal.
  - **Record _ overflow:** A TLS record was received with a payload (cipher text) whose length exceeds 214 + 2048 bytes, or the cipher text decrypted to a length of greater than 214 + 1024 bytes.
  - **Unknown _ ca:** A valid certificate chain or partial chain was received, but the certificate was not accepted because the CA certificate could not be located or could not be matched with a known, trusted CA.
  - **Access _ denied:** A valid certificate was received, but when access control was applied, the sender decided not to proceed with the negotiation.
  - **Decode _ error:** A message could not be decoded, because either a field was out of its specified range or the length of the message was incorrect.
  - **Protocol _version:** The protocol version the client attempted to negotiate is recognized but not supported.
  - **Insufficient _ security:** Returned instead of handshake _ failure when a negotiation has failed specifically because the server requires ciphers more secure than those supported by the client.
  - **Unsupported _ extension:** Sent by clients that receives an extended server hello containing an extension not in the corresponding client hello.
  - **Internal _ error:** An internal error unrelated to the peer or the correctness of the protocol makes it impossible to continue.
  - **Decrypt _ error:** A handshake cryptographic operation failed, including being unable to verify a signature, decrypt a key exchange, or validate a finished message.

The remaining alerts include the following.

- **User _ cancelled:** This handshake is being cancelled for some reason unrelated to a protocol failure.

- **No _ renegotiation:** Sent by a client in response to a hello request or by the server in response to a client hello after initial handshaking. Either of these messages would normally result in renegotiation, but this alert indicates that the sender is not able to renegotiate. This message is always a warning.

COMPARISON OF THREATS ON THE WEB 8m (NOV-2020)

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

BGS Institute of Technology

Department of Electronics and communication Engg

Network and cyber security(17EC835)

Module-1

TRANSPORT LEVEL SECURITY

Jayanth dwijesh H P
Assistant Professor
Dept of ECE,BGSIT

What is a Network?

A Network is two or more systems joined together Via a Switch, Communicating Via a Routing Protocol.

What is the internet?

The biggest World Network is Internet

What is Computer Network?

- Computer Network Is called Data Network
- The inter connection b/w Computers and other Devices

What is Cyber Security?

Cyber Security is the Protection of Internet Connection systems ,Including Hardware, Software and Data from Cyber Attacks.

What are the Data Hiding Techniques?

- Cryptography
- Steganography
- Water Marking
- Hash Function
- Dual-Steganography

E.t.c

Cryptography

- The word Cryptography was derives from combing 2 Greek words ,"Krypto" it Means "Hidden" and "Graphee" Means "Writing"
- Cryptography is The art of Secrete Information Writing or Secrete Data Writing .
- The Main Goal of Cryptography is data secure from unauthorized person or Hackers.

Types Of Cryptography

- Symmetric key cryptography
- Asymmetric Key cryptography
- Hash Function

Steganography

- The word Steganography was derives from combing 2 Greek words ,"Steganos" it Means "Hidden" or "Covered" and "Graphic Means "Written"
- The main Idea of steganography is to Hide secret Messages in the other covered digital Medias such as Text, video, Audio And Image e.t.c. S.T Some one or Hackers or Other person Can not Know the person of the Secret Information .

Dual steganography

- Dual Steganography is the Process of using Steganography Combined with Cryptography
- Dual steganography is the process of hiding confidential data's in the Media files such as Audio, Images and Video E.t.c

Web security

- Wed security means Providing the Security for The Data Which is transmitted To the Network .

A Comparison of Threats on the Web

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

SSL

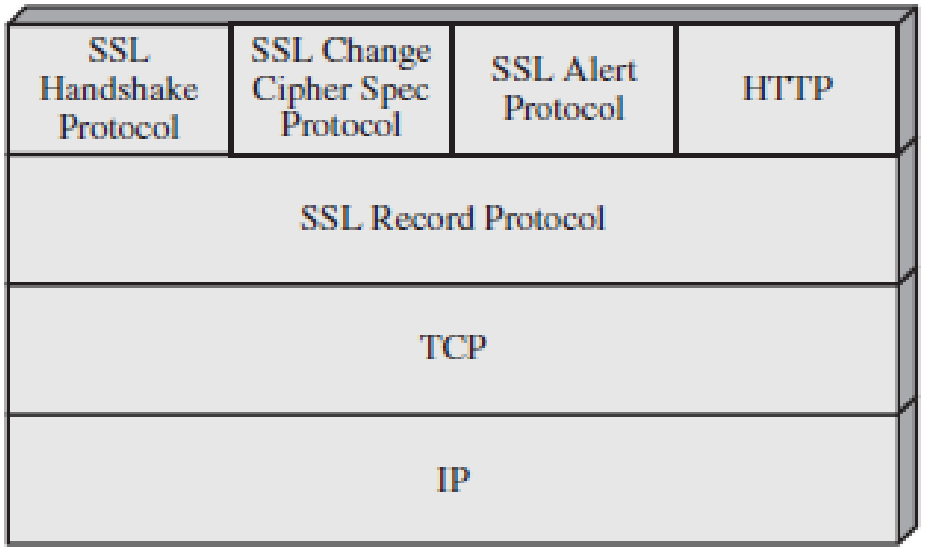


Figure :SSL Protocol Stack

Secure Socket Layer is designed to make use of TCP to provide a reliable end-to end secure service.

Moreover, Secure Socket Layer is not a single protocol but rather two layers of protocols

The SSL Record Protocol provides basic security services to various higher layer protocols.

- In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL.
- Three higher-layer protocols are defined as part of SSL: the **Handshake Protocol, the Change Cipher Spec Protocol, and the Alert Protocol.**

Connection

A connection is a transport that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection associated with one session.

Session

An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

SSL RECORD PROTOCOL: SSL PROTOCOL

- The SSL Record Protocol provides two services for SSL connections: **Confidentiality and Message Integrity.**
- **Confidentiality: The Handshake Protocol defines a shared secret key that is** used for conventional encryption of SSL payloads.
- **Message Integrity: The Handshake Protocol also defines a shared secret key** that is used to form a messa
- Moreover, the overall operation of Record Protocol is:ge authentication code (MAC).
- **Fragmentation: Each upper-layer message fragmented into blocks of 214 bytes** (16384 bytes) or less.
- **Compression: Compression is optionally applied. Compression must be lossless** and may not increase the content length by more than 1024 bytes.

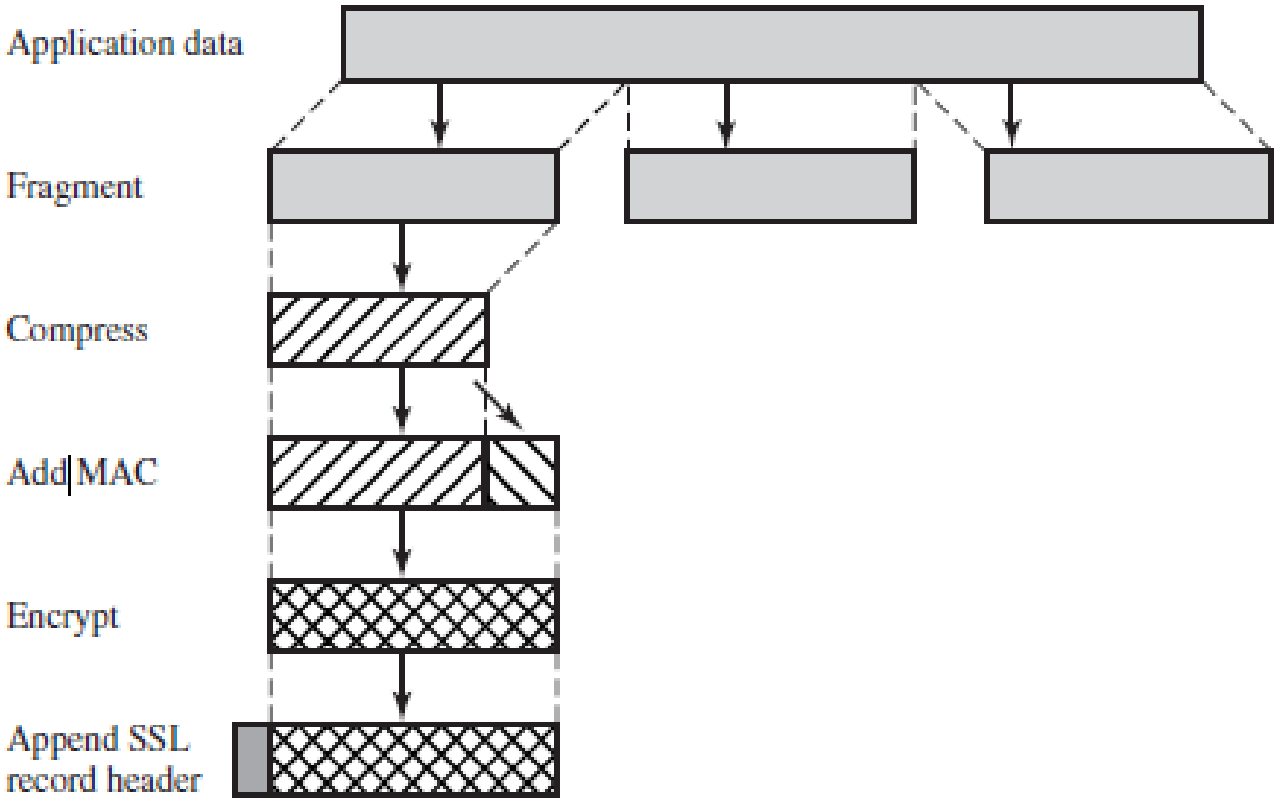


Figure :SSL Record Protocol Operation

- Add message authentication code: MAC calculated over the compressed data by the following expression.

  Hash (MAC_write_secret || pad_2 || hash (MAC_write_secret || pad_1 || seq _ num || SSL Compressed. type || SSL Compressed. length || SSL Compressed. fragment))

- **Encryption: The compressed message plus the MAC encrypted using symmetric** encryption. Algorithms supported are AES, RC4-40, IDEA, RC2, DES, 3DES and Fortezza.

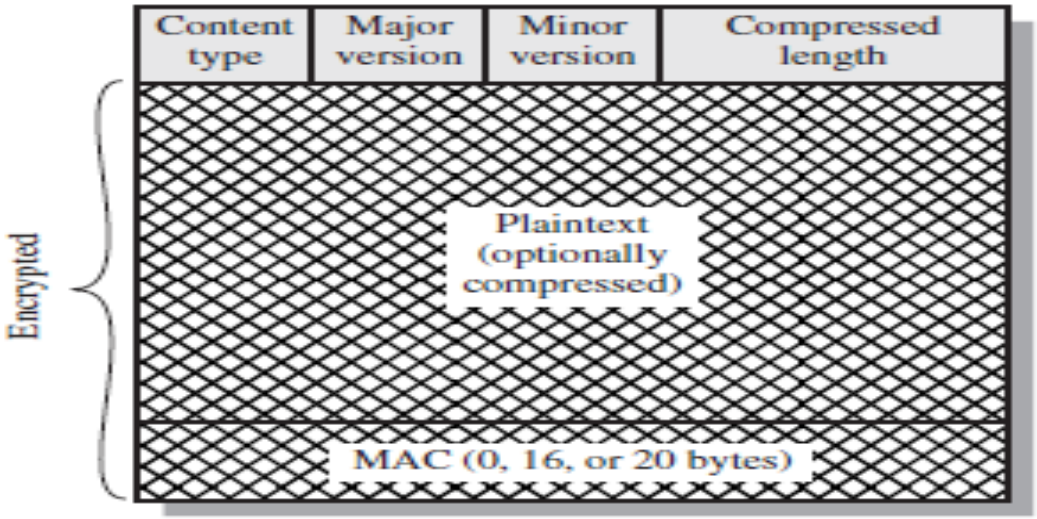


FIG:SSL Record Format

➢The final step of SSL Record Protocol processing is to prepare a header consisting of the following fields:

- **Content Type (8 bits): The higher-layer protocol used to process the fragment.**
- **Major Version (8 bits): Indicates major version of SSL in use. For SSLv3, the** value is 3.
- **Minor Version (8 bits): Indicates minor version in use. For SSLv3, the value is 0.**
- **Compressed Length (16 bits):The length in bytes of the fragment.**

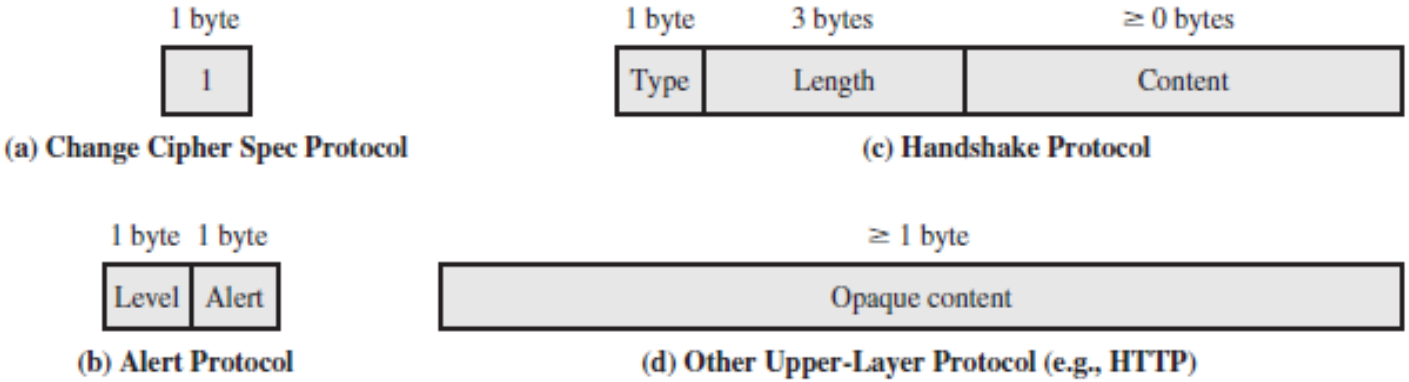


Figure: SSL Record Protocol Payload

- The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a single message of a single byte with the value 1.
- The Alert Protocol used to convey SSL-related alerts to the peer entity keys.

TRANSPORT LAYER SECURITY (TLS)

- TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL.
- Moreover, TLS is defined as a Proposed Internet Standard in RFC 5246. Is very similar to SSLv3.

PSEUDORANDOM FUNCTION

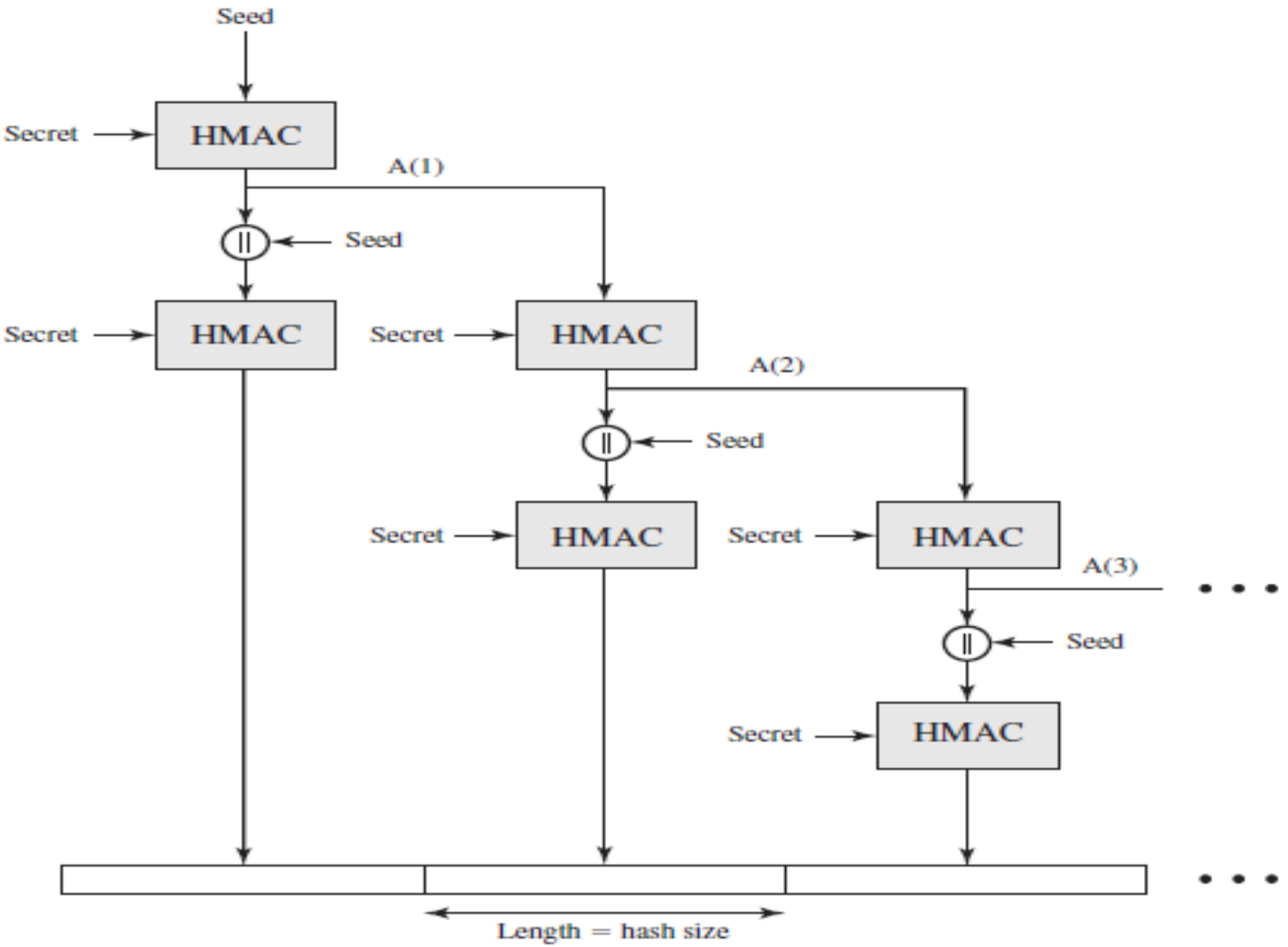


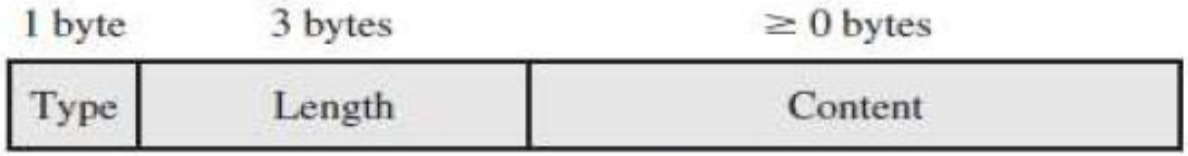
Figure : TLS Function P _ hash (secret, seed)

- TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation.
- Moreover, the objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash function
- The PRF is based on the data expansion function (Above Figure) given as

$$P\_hash(secret, seed) = HMAC\_hash(secret, A(1) \,||\, seed) \,||\, HMAC\_hash(secret, A(2) \,||\, seed) \,||\, HMAC\_hash(secret, A(3) \,||\, seed) \,||\, \ldots$$

Handshake Protocol

- The Main purpose of this handshake protocol is to establish the session.
- handshake protocol is used to authenticate the client with server and server with the client.
- The Handshake Protocol is used before any application data is transmitted.
- The Handshake Protocol consists of a series of messages exchanged by client And server.



Handshake Protocol

As shown in above fig:-

1. Type (1 byte): Indicates one of 10 messages of handshake protocol(as shown in below Table).
2. Length (3 bytes): The length of the message in bytes.
3. Content (bytes): The parameters associated with this message.

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Table: SSL Handshake Protocol Message Types

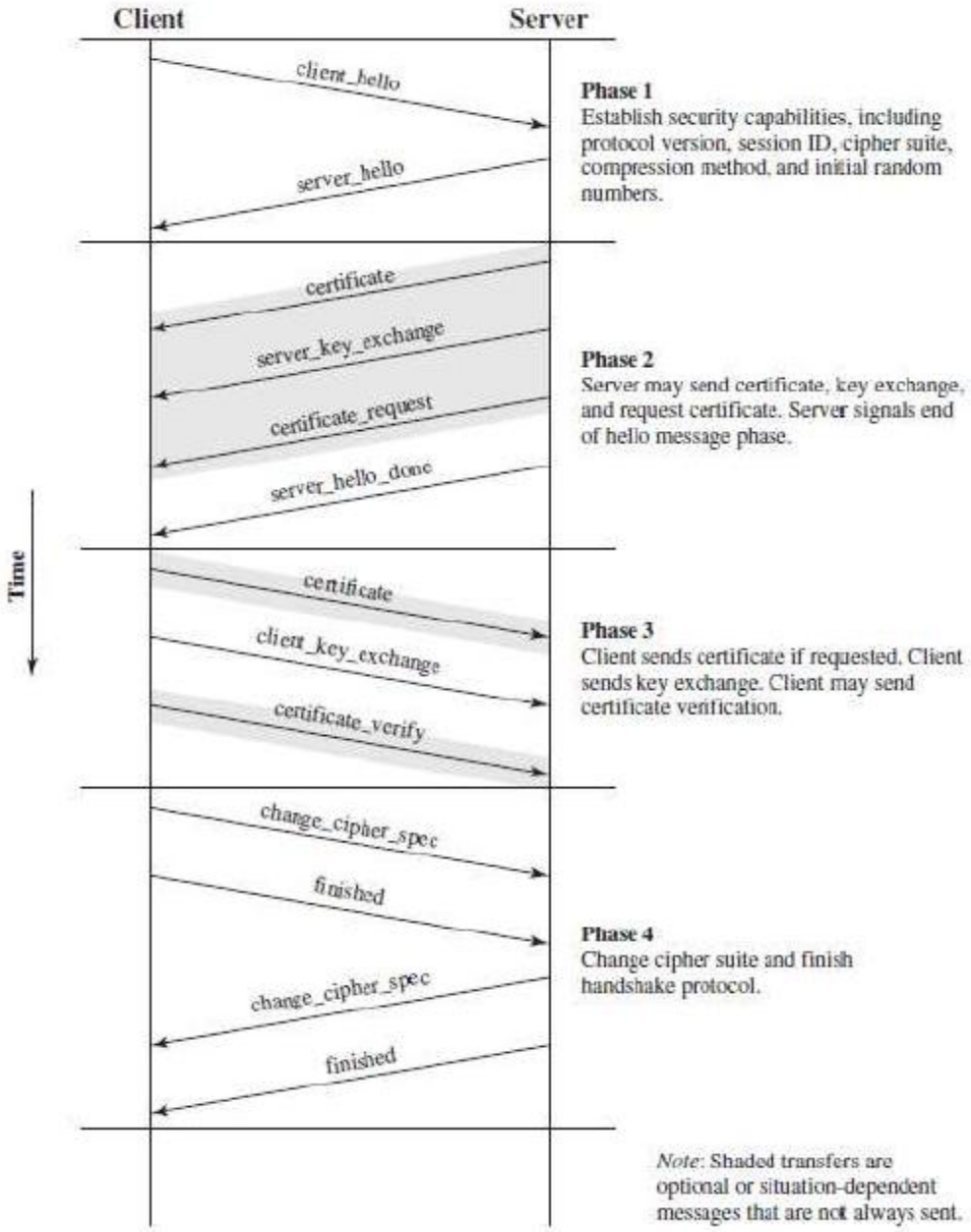


Figure: Handshake Protocol Actions

HTTPS

- HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server.
- The HTTPS capability is built into all modern Web browsers.
- Its use depends on the Web server supporting HTTPS communication. For example, search engines do not support HTTPS.
- The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://.

SECURE SHELLS (SSH)

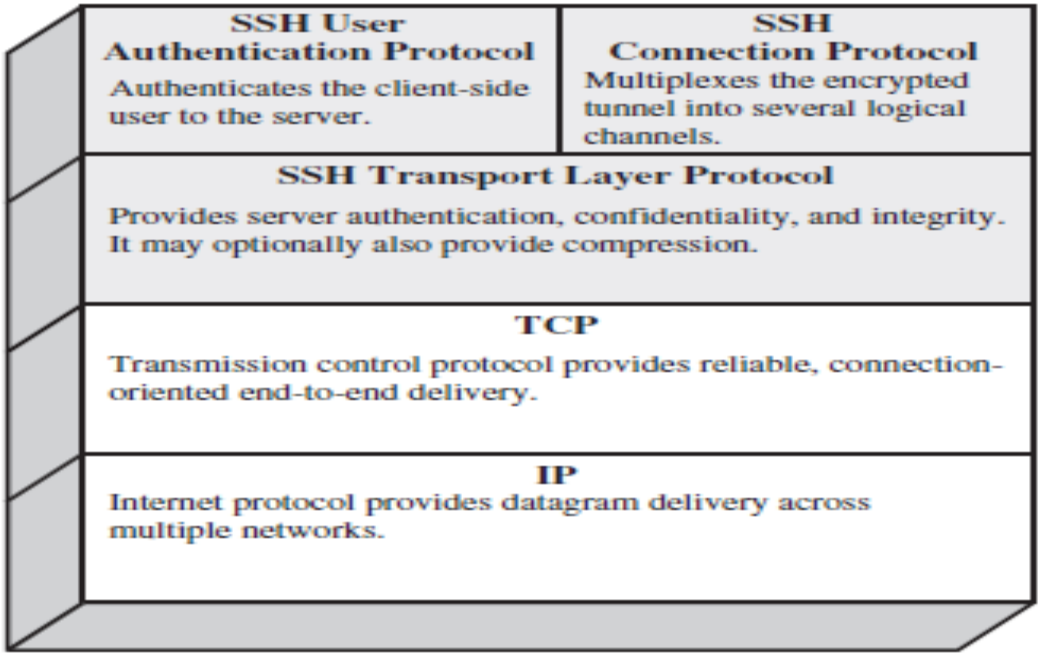


Figure : SSH Protocol Stack

Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. A new version, SSH2, fixes a number of security flaws in the original scheme. SSH2 is documented as a proposed standard in IETF RFCs 4250 through 4256.

SSH Transport Layer Protocol Packet Exchanges

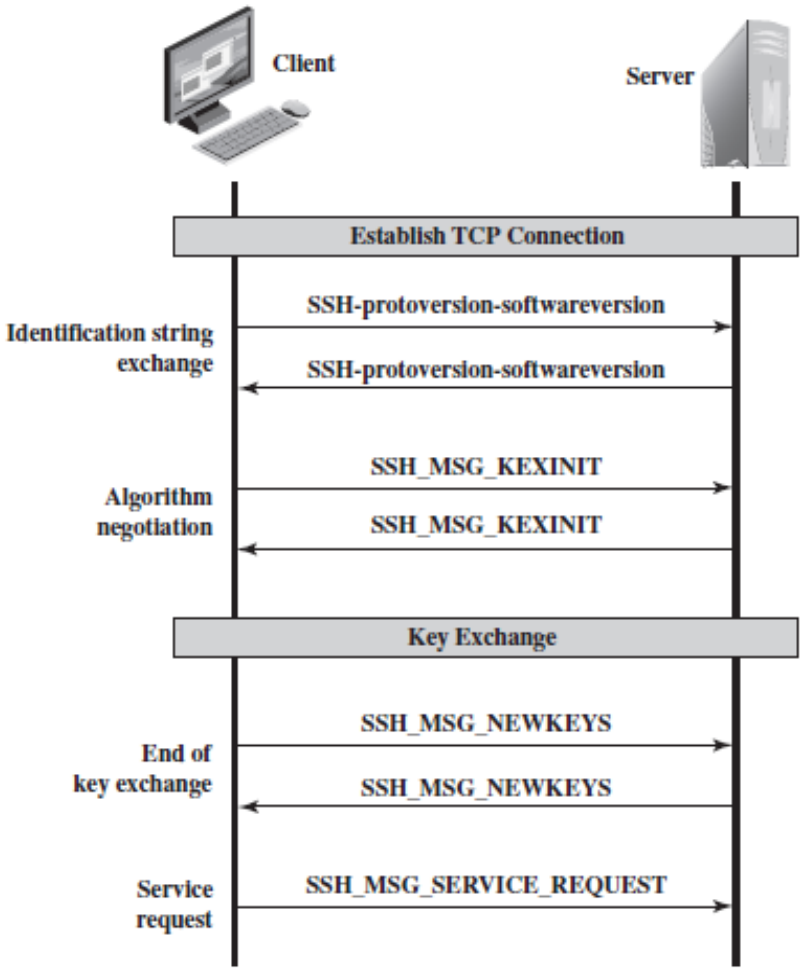


FIG:SSH Transport Layer Protocol Packet Exchanges

The SSH Transport Layer packet exchange consists of a sequence of steps

- The first step, the identification string exchange, begins with the client sending a packet with an identification string of the form:

SSH-protoversion-software version SP comments CR LF

- Next comes algorithm negotiation.
- The next step is key exchange.
- The end of key exchange is signalled by the exchange of SSH_MSG_NEWKEYS packets. At this point, both sides may start using the keys generated from *K, as* discussed subsequently.
- The final step is service request. The client sends an SSH_MSG_SERVICE_ REQUEST packet to request either the User Authentication or the Connection Protocol. Subsequent to this, all data is exchanged as the payload of an SSH Transport Layer packet, protected by encryption and MAC.

SSH Transport Layer Protocol Packet Formation

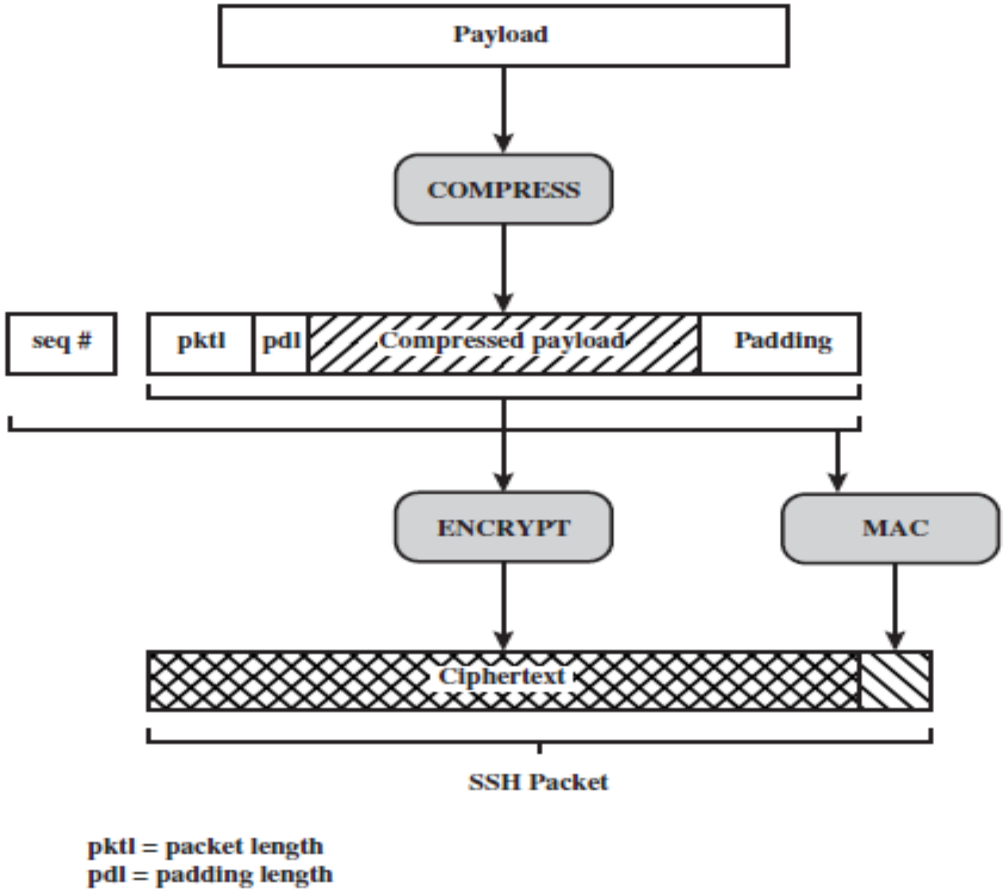


Figure :SSH Transport Layer Protocol Packet Formation

Each packet is in the following Format

- Packet length:
- Padding length
- Payload
- Random padding
- Message authentication code (MAC)

SSH Connection Protocol Message Exchange

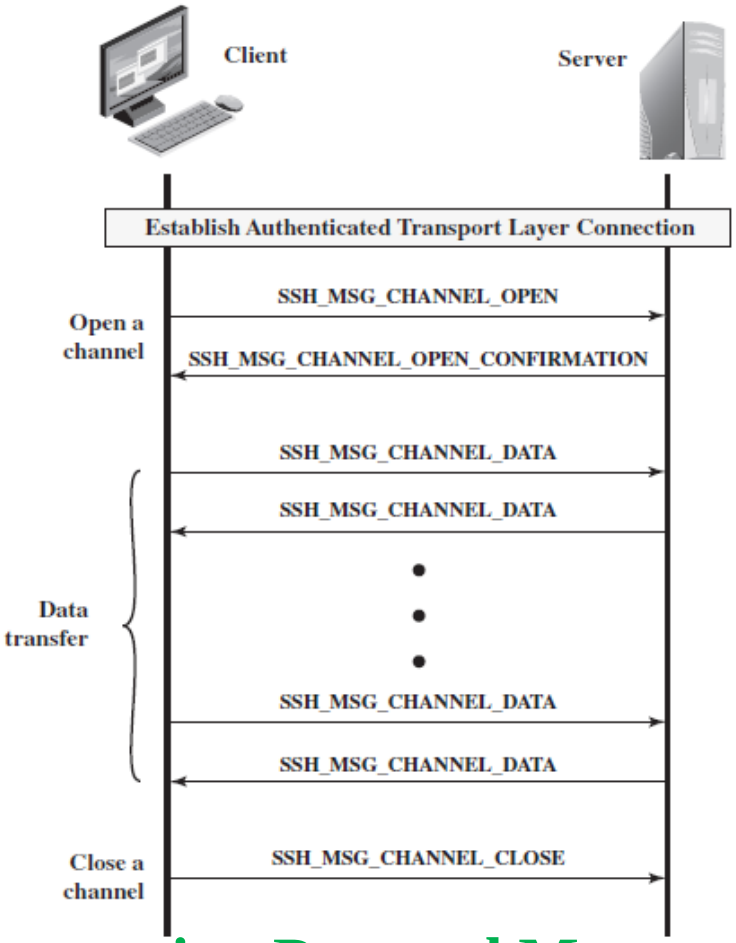


Fig: SSH Connection Protocol Message Exchange

Port Forwarding

- One of the most useful features of SSH is port forwarding. In essence, port forwarding provides the ability to convert any insecure TCP connection into a secure SSH connection. This is also referred to as SSH tunnelling.
- A **port is an identifier of a user of TCP.**
- So, any application that runs on top of TCP has a port number.
- Incoming TCP traffic is delivered to the appropriate application on the basis of the port number.
- An application may employ multiple port numbers.

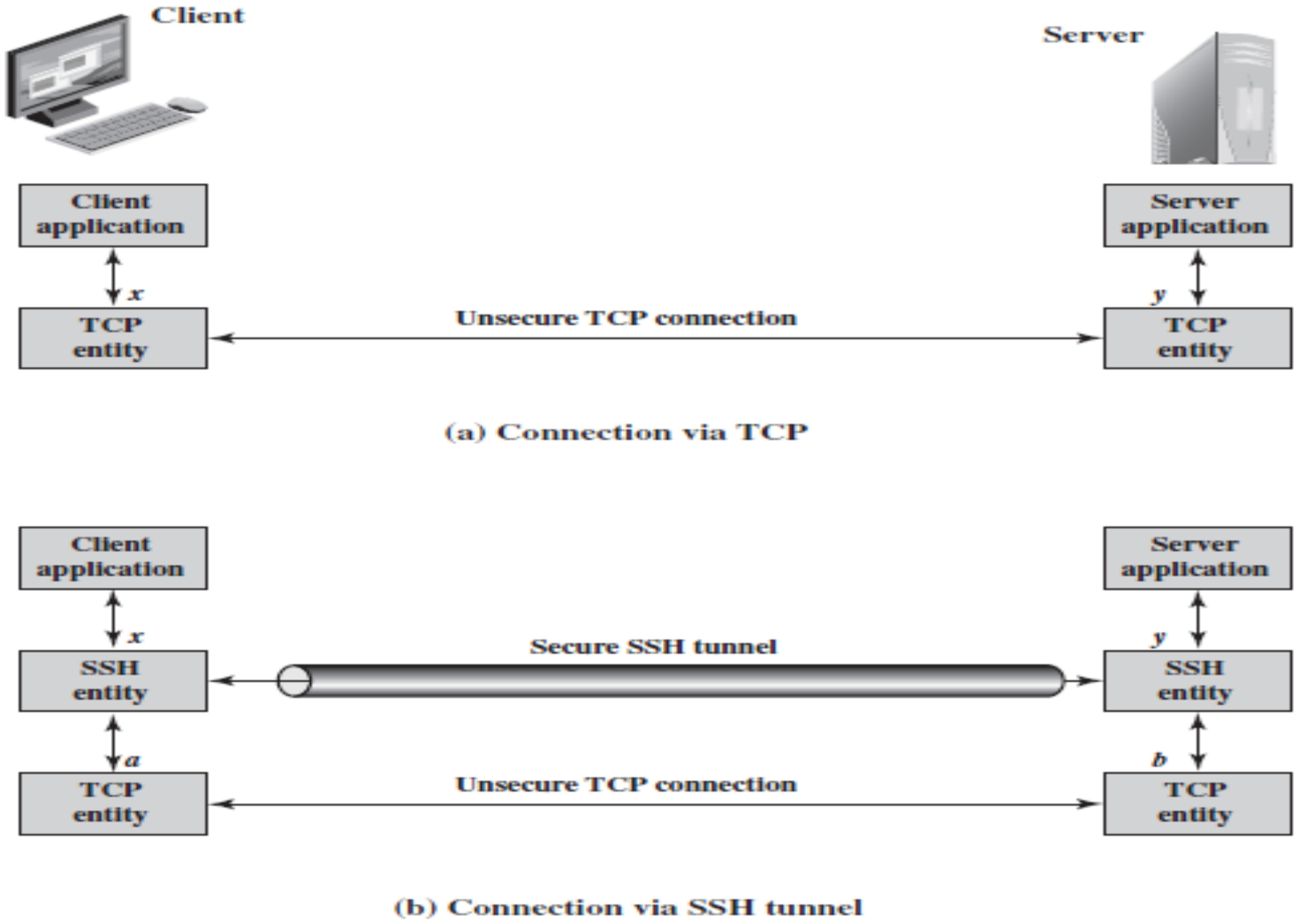


Figure: SSH Transport Layer Packet Exchanges

LOCAL FORWARDING:

Allows the client to set up a "hijacker" process. This will intercept selected application-level traffic and redirect it from an unsecured TCP connection to a secure SSH tunnel.

REMOTE FORWARDING:

The user's SSH client acts on the server's behalf. The client receives traffic with a given destination port number, places the traffic on the correct port and sends it to the destination the user chooses.

THANK YOU

NETWORK AND CYBER SECURITY
(15EC835, 17EC835)

8TH SEM E&C

JAYANTH DWIJESH H P **BE (ECE), M.tech (DECS).**

Assistant Professor – Dept of E&CE, BGSIT.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

B.G.S INSTITUTE OF TECHNOLOGY (B.G.S.I.T)

B.G Nagara, Nagamangala Tq, Mandya District- 571448

NETWORK AND CYBER SECURITY

MODULE–2

MODULE–2

E-mail Security: Pretty Good Privacy, S/MIME, and Domain keys identified mail.

TEXT BOOK:

1. **William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325-1877-3.**
2. **Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.**

REFERENCE BOOKS:

1. **Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.**
2. **Cryptography and Network Security, Atul Kahate, TMH, 2003.**

MODULE-2

E-mail Security: Pretty Good Privacy, S/MIME, and Domain keys identified mail.

1. PRETTY GOOD PRIVACY

IMP QS (question)-06M

PGP is a remarkable phenomenon. Largely the effort of a single person, Phil Zimmermann, PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

In essence, Zimmermann has done the following:

1. Selected the best available cryptographic algorithms as building blocks.
2. Integrated these algorithms into a general-purpose application that is independent of operating system and processor and that is based on a small set of easy-to-use commands.
3. Made the package and its documentation, including the source code, freely available via the Internet, bulletin boards, and commercial networks such as AOL (America on Line).
4. Entered into an agreement with a company (Via crypt, now Network Associates) to provide a fully compatible, low-cost commercial version of PGP.

PGP has grown explosively and is now widely used. A number of reasons can be cited for this growth

1. It is available free worldwide in versions that run on a variety of platforms, including Windows, UNIX, Macintosh, and many more. In addition, the commercial version satisfies users who want a product that comes with vendor support.
2. It is based on algorithms that have survived extensive public review and are considered extremely secure. Specifically, the package includes RSA, DSS, and Diffie-Hellman for public-key encryption; CAST-128, IDEA, and 3DES for symmetric encryption; and SHA-1 for hash coding.
3. It has a wide range of applicability, from corporations that wish to select and enforce a standardized scheme for encrypting files and messages to individuals who wish to communicate securely with others worldwide over the Internet and other networks.
4. It was not developed by, nor is it controlled by, any governmental or standards organization. For those with an instinctive distrust of "the establishment," this makes PGP attractive.

5. PGP is now on an Internet standards track (RFC 3156; MIME Security with Open PGP). Nevertheless, PGP still has an aura of an antiestablishment endeavour.

1.1 NOTATION

Most of the notation used in this chapter has been used before, but a few terms are new. It is perhaps best to summarize those at the beginning. The following symbols are used.

Ks = session key used in symmetric encryption scheme
PRa = private key of user A, used in public-key encryption scheme
PUa = public key of user A, used in public-key encryption scheme
EP = public-key encryption
DP = public-key decryption
EC = symmetric encryption
DC = symmetric decryption
H = hash function
||= concatenation
Z = compression using ZIP algorithm
R64 = conversion to radix 64 ASCII format

1.1 OPERATIONAL DESCRIPTION JULY-2019(10M) DEC-2019(10M), SEPT-2020.

The actual operation of PGP, as opposed to the management of keys, consists of four services: authentication, confidentiality, compression, and e-mail compatibility (Table 1).

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or 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 included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

Table 1: Summary of PGP Services

1.1.1 PGP Operation- Authentication

IMP QS (question)-06M

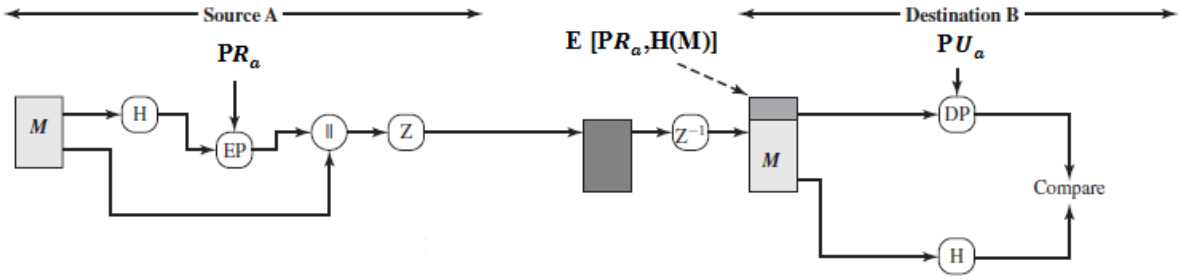


Figure 1 Authentication only

Above Figure 1:-

SENDER:

1. The sender creates a message.
2. SHA-1 is used to generate a 160-bit hash code of the message.
3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.

RECEIVER:

4. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
5. The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

1.1.2 PGP Operation- Confidentiality

IMP QS (question)-06M

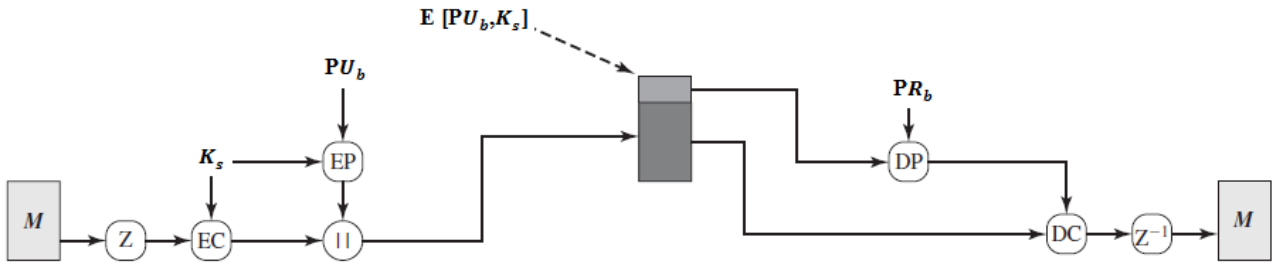


Figure 2 Confidentiality only

Above Figure 2:-

SENDER:

1. The sender generates a message and a random 128-bit number to be used as a session key for this message only.

2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.
3. The session key is encrypted with RSA using the recipient's public key and is prepended to the message.

RECEIVER:

4. The receiver uses RSA with its private key to decrypt and recover the session key.
5. The session key is used to decrypt the message.

1.1.3 PGP Operation- Confidentiality and Authentication

IMP QS (question)-06M

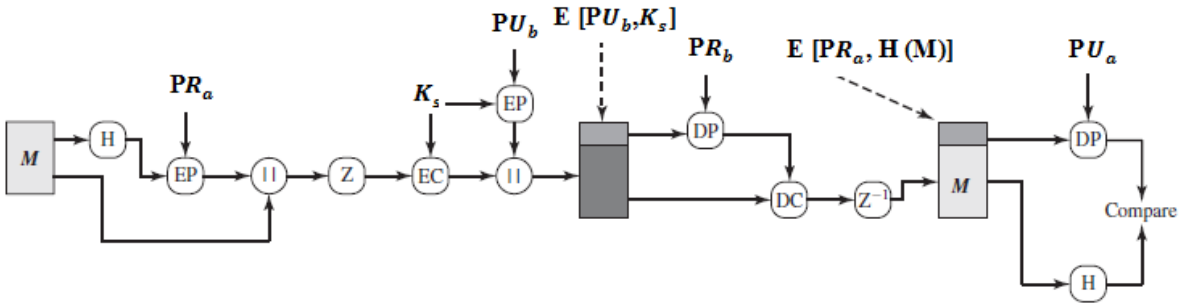


Figure 3 Confidentiality and authentication

Above Figure 3:-

1. Both services may be used for the same message. First, a signature is generated for the plaintext message and prepended to the message.
2. Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA .This sequence is preferable to the opposite: encrypting the message and then generating a signature for the encrypted message.
3. It is generally more convenient to store a signature with a plaintext version of a message. Furthermore, for purposes of third-party verification, if the signature is performed first, a third party need not be concerned with the symmetric key when verifying the signature.

In summary, when both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and finally encrypts the session key with the recipient's public key.

NOTE:- Figure 1 Authentication only, Figure 2 **Confidentiality only**, Figure 3 Confidentiality and authentication = PGP Cryptographic Functions

1.1.4 Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage. The placement of the compression algorithm, indicated by Z for compression and $Z^{-1}$for decompression in figure (1, 2 &3) critical. The compression algorithm used is ZIP.

1. The signature is generated before compression for two reasons:

a. so that one can store only the uncompressed message together with signature for later verification

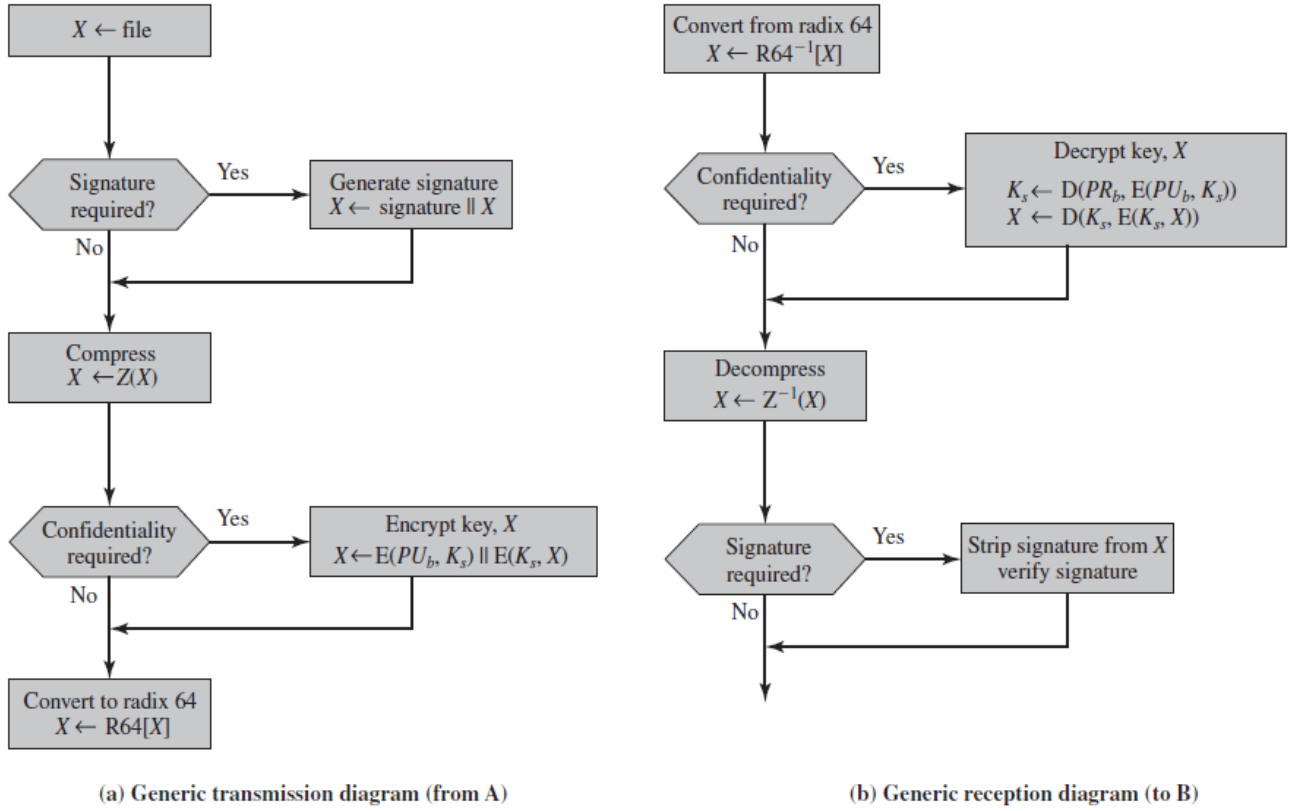
b. Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm as the PGP compression algorithm is not deterministic

2. Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.

1.1.5 PGP Operation – Email Compatibility DEC-2019(10M), SEPT-2020(10M), JULY-2019[10M]

When PGP is used, at least part of the block to be transmitted is encrypted, and thus consists of a stream of arbitrary 8-bit octets.

- However many electronic mail systems only permit the use of ASCII text. To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters.
- It uses radix-64 conversion, in which each group of three octets of binary data is mapped into four ASCII characters. This format also appends a CRC to detect transmission errors. The use of radix 64 expands a message by 33%, but still an overall compression of about one- third can be achieved.



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

Figure 4 Transmissions and Reception of PGP Messages

TRANSMISSIONS

1. On transmission (if it is required), a signature is generated using a hash code of the uncompressed plaintext.
2. Then the plaintext (plus signature if present) is compressed. Next, if confidentiality is required, the block (compressed plaintext or compressed signature plus plaintext) is encrypted and prepended with the public key-encrypted symmetric encryption key.
3. Finally, the entire block is converted to radix-64 format.

RECEPTION

4. On reception, the incoming block is first converted back from radix-64 format to binary.
5. If the message is encrypted, the recipient recovers the session key and decrypts the message. The resulting block is then decompressed.
6. If the message is signed, the recipient recovers the transmitted hash code and compares it to its own calculation of the hash code.

1.2 PGP OPERATION - SEGMENTATION/REASSEMBLY

- E-mail facilities often are restricted to a maximum message length.
- For example, many of the facilities accessible through the Internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately.
- To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.
- The segmentation is done after all of the other processing, including the radix- 64 conversion.
- The session key component and signature component appear only once, at the beginning of the first segment.
- Reassembly at the receiving end is required before verifying signature or decryption.

1.3 KEY IDENTIFIERS OR PGP MESSAGE FORMAT

IMP QS (question)-08M

The concept of key ID has been introduced; we can take a more detailed look at the format of a transmitted message, which is shown in Figure 5

- A message consists of three components: **the message component, a signature (Optional), and a session key component (optional).**
- The **message component** includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation.
- The **signature component** includes the following:
  - **Timestamp:** The time at which the signature was made.
  - **Message digest:** The 160-bit SHA-1 digest, encrypted with the sender's private signature key.
  - **Leading two octets of message digest:** To enable the recipient to determine if the correct public key was used to decrypt the message digest for authentication, by comparing this plaintext copy of the first two octets with the first two octets of the decrypted digest. These octets also serve as a 16-bit frame check sequence for the message.
  - **Key ID of sender's public key:** Identifies the public key that should be used to decrypt the message digest and, hence, identifies the private key that was used to encrypt the message digest.

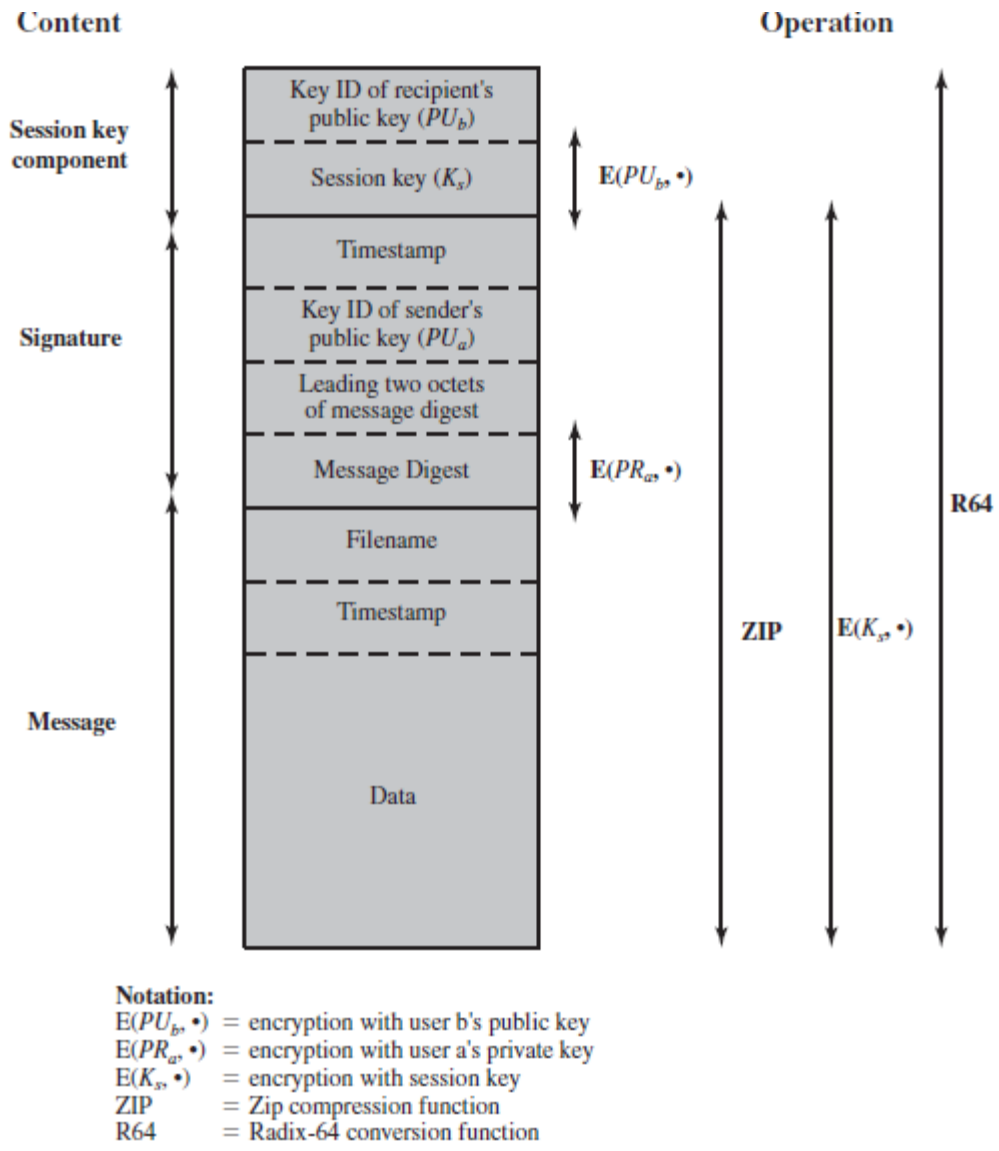


Figure 5 General Format PGP Message (from A to B)

- The **session key component** includes the session key and the identifier of the recipient's public key that was used by the sender to encrypt the session key.
- The entire block is usually encoded with radix-64 encoding.

1.4 PGP MESSAGE GENERATIONS OR PGP MESSAGE TRANSMISSION AND RECEPTION GENERATIONS OR KEY RINGS

IMP QS (question)-10M , NOV -2020(10M)

1.4.1 Message transmission

The following figure 6 shows the steps during message transmission assuming that the Message is to be both signed and encrypted.

The sending PGP entity performs the following steps

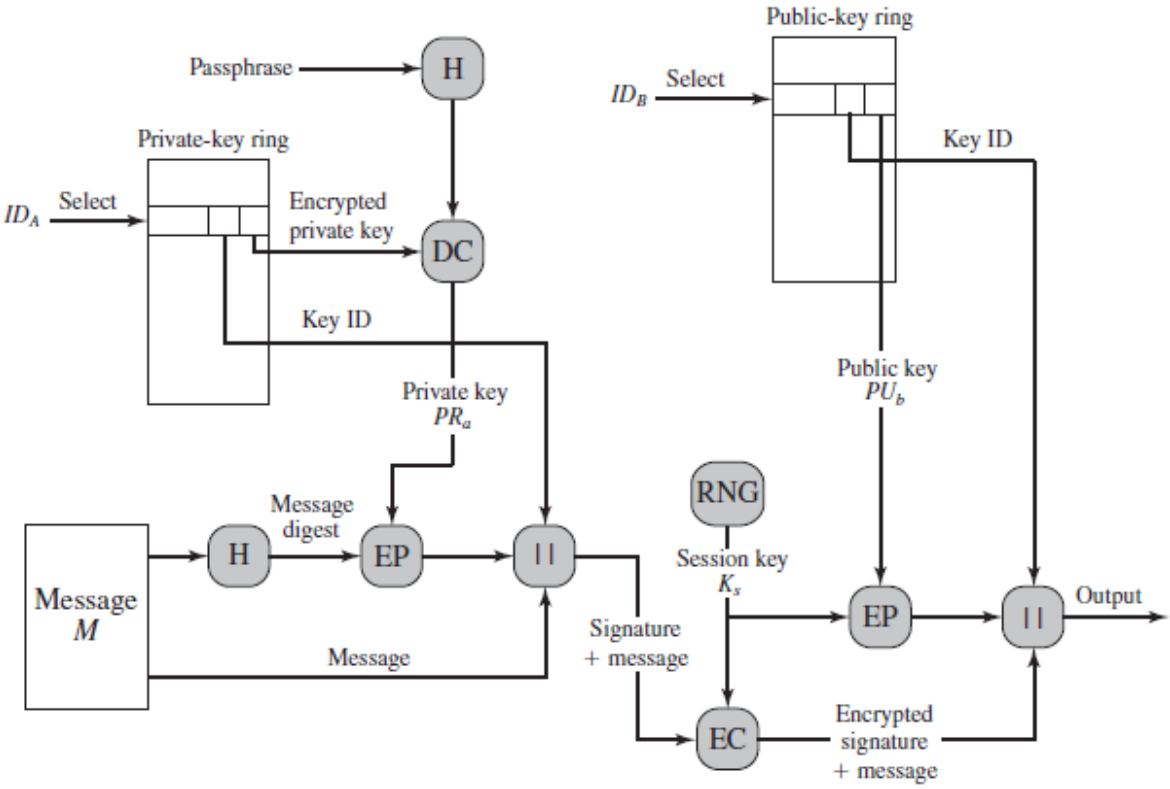


Figure 6 PGP Message Generation (from User A to User B: no compression or radix-64 Conversion)

SIGNING THE MESSAGE

a. PGP retrieves the sender's private key from the private-key ring using your _ user id as an index. If your_ user id was not provided in the command, the first private key on the ring is retrieved.
b. PGP prompts the user for the passphrase to recover the unencrypted private key.
c. The signature component of the message is constructed.

ENCRYPTING THE MESSAGE

a. PGP generates a session key and encrypts the message.
b. PGP retrieves the recipient's public key from the public-key ring using her _ user id as an index.
c. The session key component of the message is constructed.

1.4.2 Message Reception

The receiving PGP entity performs the following steps (Figure 7)

a. PGP retrieves the receiver's private key from the private-key ring using the Key ID field in the session key component of the message as an index
b. PGP prompts the user for the passphrase to recover the unencrypted private key.

c. PGP then recovers the session key and decrypts the message.

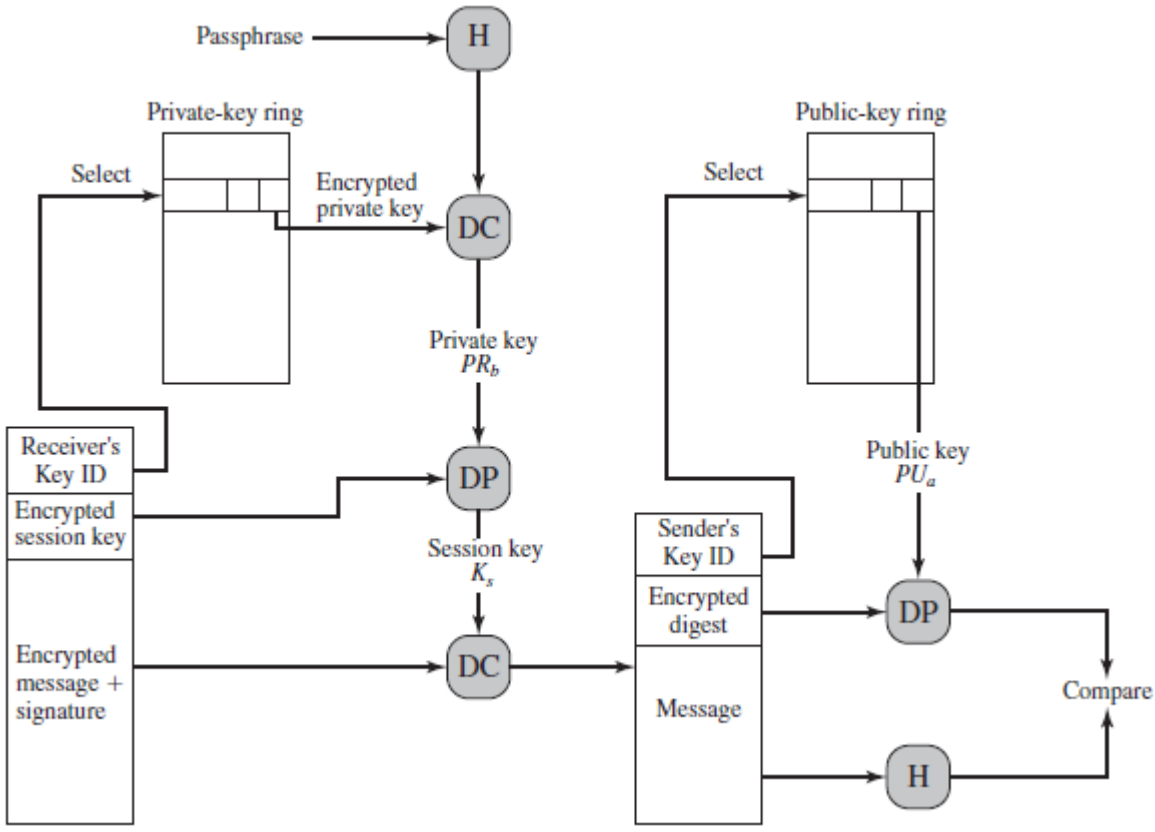


Figure 7 PGP Message Reception (from User A to User B; no compression or radix-64 conversion)

AUTHENTICATING THE MESSAGE

a. PGP retrieves the sender's public key from the public-key ring, using the Key ID field in the signature key component of the message as an index.
b. PGP recovers the transmitted message digest.
c. PGP computes the message digest for the received message and compares it to the transmitted message digest to authenticate.

1.5 RADIX-64 CONVERSION JULY-2020(6M)IMP QS (question)-06M

Both PGP and S/MIME make use of an encoding technique referred to as radix-64 Conversion. This technique maps arbitrary binary input into printable character output. The form of encoding has the following relevant characteristics:

1. The range of the function is a character set that is universally represent able At all sites, not a specific binary encoding of that character set. Thus, the characters themselves can be encoded into whatever form is needed by a specific system.

For example, the character "E" is represented in an ASCII based system as hexadecimal 45 and in an EBCDIC-based system as hexadecimal C5.

2. The character set consists of 65 printable characters, one of which is used for Padding. With 26 = 64 available characters, each character can be used to represent bits of input.
3. No control characters are included in the set. Thus, a message encoded in radix 64 can traverse mail-handling systems that scan the data stream for control characters.
4. The hyphen character "-" is not used. This character has significance in the RFC 5322 format and should therefore be avoided.

Table 2 shows the mapping of 6-bit input values to characters. The character set consists of the alphanumeric characters plus "+" and "/". The "=" character is used as the padding character.

6-bit Value	Character Encoding	6-bit Value	Character Encoding	6-bit Value	Character Encoding	6-bit Value	Character Encoding
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

Table 2 Radix-64 Encoding

Figure 8 illustrates the simple mapping scheme. Binary input is processed in blocks of 3 octets (24 bits). Each set of 6 bits in the 24-bit block is mapped into a character. In the figure, the characters are shown encoded as 8-bit quantities. In this typical case, each 24-bit input is expanded to 32 bits of output.

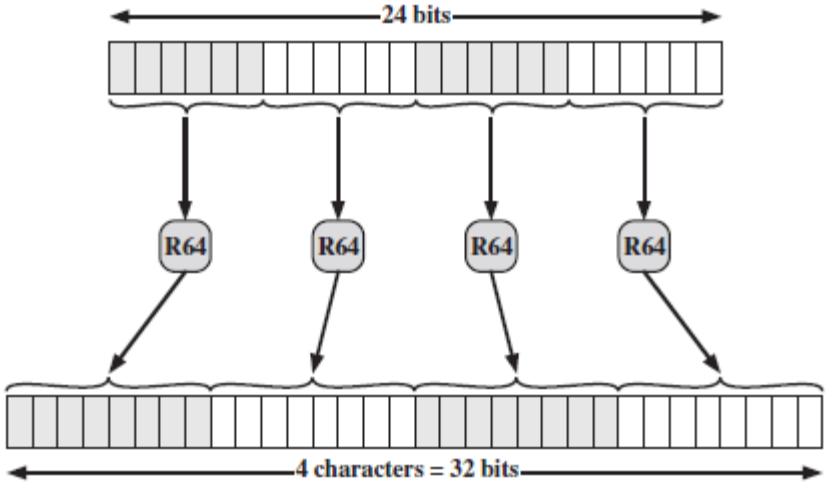


Figure 8 Printable Encoding of Binary Data into Radix-64 Format

For example, consider the 24-bit raw text sequence 00100011 01011100 10010001, which can be expressed in hexadecimal as 235C91. We arrange this input in blocks of 6 bits: 001000 110101 110010 010001

The extracted 6-bit decimal values are 8, 53, 50, and 17. Looking these up in Table 2 yields the radix-64 encoding as the following characters: I1yR. If these characters are stored in 8-bit ASCII format with parity bit set to zero, we have 01001001 00110001 01111001 01010010

In hexadecimal, this is 49317952. To summarize:

Input Data	
Binary representation	00100011 01011100 10010001
Hexadecimal representation	235C91
Radix-64 Encoding of Input Data	
Character representation	I1yR
ASCII code (8 bit, zero parity)	01001001 00110001 01111001 01010010
Hexadecimal representation	49317952

2. S/MIME

IMP QS (question)-06M

- Secure/Multipurpose Internet Mail Extension (S/MIME) is a security enhancement to the MIME Internet e-mail format standard based on technology from RSA Data Security.
- Both PGP and S/MIME are on an IETF standards track, it appears likely that S/MIME will emerge as the industry standard for commercial and organizational

use, while PGP will remain the choice for personal e-mail security for many users.

- S/MIME is defined in a number of documents—most importantly RFCs 3370, 3850, 3851, and 3852.

2.1 RFC 5322

IMP QS (question)-06M, JULY-2020(6M)

- RFC 5322 defines a format for text messages that are sent using electronic mail.
- It has been the standard for Internet-based text mail messages and remains in common use.
- In the RFC 5322 context, messages are viewed as having an envelope and contents.
- The envelope contains whatever information is needed to accomplish transmission and delivery.
- The contents compose the object to be delivered to the recipient.
- The RFC 5322 standard applies only to the contents.
- The content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.
- The overall structure of a message that conforms to RFC 5322 is very simple.
- A message consists of some number of header lines (*the header*) followed by unrestricted text (*the body*).
- The header is separated from the body by a blank line. Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system.

2.2 MULTIPURPOSE INTERNET MAIL EXTENSIONS

IMP QS (question)-10M

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail.

Lists the following limitations of the SMTP/5322 scheme.

1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX Uuencode/ Uudecode scheme. However, none of these is a standard or even a de facto standard.
2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle non textual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include:

- Deletion, addition, or reordering of carriage return and linefeed
- Truncating or wrapping lines longer than 76 characters
- Removal of trailing white space (tab and space characters)
- Padding of lines in a message to the same length
- Conversion of tab characters into multiple space characters

The MIME specification includes the following elements.

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

The five header fields defined in MIME are

1. **MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
2. **Content-Type:** Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to

represent the data to the user or otherwise deal with the data in an appropriate manner.

3. **Content-Transfer-Encoding:** Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
4. **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
5. **Content-Description:** A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

2.3 MIME CONTENT TYPES

IMP QS (question)-10M

- The bulk of the MIME specification is concerned with the definition of a variety of content types.
- Table 3 lists the content types specified in RFC 2046. There are seven different major types of content and a total of 15 subtypes.
- In general, a content type declares the general type of data, and the subtype specifies a particular format for that type of data.
- For **the text type**
  - **The text type** of body, the primary subtype is plain text, which is simply a string of ASCII characters or ISO 8859 characters. The enriched subtype allows greater formatting flexibility.
- For **multipart type**
  - **Multipart type** indicates that the body contains multiple, independent parts.
  - The Content-Type header field includes a parameter (called a boundary) that defines the delimiter between body parts.
  - This boundary should not appear in any parts of the message.
  - Each boundary starts on a new line and consists of two hyphens followed by the boundary value.
  - The final boundary, which indicates the end of the last part, also has a suffix of two hyphens. Within each part, there may be an optional ordinary MIME header.
- There are four subtypes of the **multipart type,** all of which have the same overall syntax.

1. The **multipart/mixed subtype,** is used when there are multiple independent body parts that need to be bundled in a particular order.
2. The **multipart/ parallel subtype,** the order of the parts is not significant. If the recipient's system is appropriate, the multiple parts can be presented in parallel.
3. The **multipart/alternative subtype**, the various parts are different representations of the same information.

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript format.
	octet-stream	General binary data consisting of 8-bit bytes.

Table 3: MIME Content Types

4. The **multipart/digest subtype**, is used when each of the body parts is interpreted as an RFC 5322 message with headers. This subtype enables the construction of a message whose parts are individual messages.

- The **message type** provides a number of important capabilities in MIME.
  1. The **message/rfc822** subtype indicates that the body is an entire message, including header and body.
  2. The **message/partial subtype** Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.

3. The **message/external-body subtype** indicates that the actual data to be conveyed in this message are not contained in the body. Instead, the body contains the information needed to access the data. As with the other message types, the message/external-body subtype has an outer header and an encapsulated message with its own header.

- The **application type** refers to other kinds of data, typically either uninterpreted binary data or information to be processed by a mail-based application.

2.4 MIME TRANSFER ENCODINGS

IMP QS (question)-06M

The other major component of the MIME specification, in addition to content type specification, is a definition of transfer encodings for message bodies. The objective is to provide reliable delivery across the largest range of environments.

- The MIME standard defines two methods of encoding data. **The Content-Transfer-Encoding field** can actually take on six values, as listed in Table 4.
- However, three of these values (7bit, 8bit, and binary) indicate that no encoding has been done but provide some information about the nature of the data.

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present, but the lines are not neces-sarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

Table 4: MIME Transfer Encodings

- For SMTP transfer, it is safe to use the **7bit form**.
- The **8bit and binary** forms may be usable in other mail transport contexts.
- **Another Content-Transfer-Encoding value is x-token,** which indicates that some other encoding scheme is used for which a name is to be supplied.
- The **quoted-printable** transfer encoding is useful when the data consists largely of octets that correspond to printable ASCII characters. In essence, it represents non safe characters by the hexadecimal representation of their code and introduces reversible (soft) line breaks to limit message lines to 76 characters.

- The **base64 transfer encoding,** also known as radix-64 encoding, is a common one for encoding arbitrary binary data in such a way as to be invulnerable to the processing by mail-transport programs.

2.5 NATIVE AND CANONICAL FORM

IMP QS (question)-04M

An important concept in MIME and S/MIME is that of canonical form. Canonical form is a format, appropriate to the content type that is standardized for use between systems. This is in contrast to native form, which is a format that may be peculiar to a particular system. Shown in below table 5

Native Form	The body to be transmitted is created in the system's native format. The native character set is used and, where appropriate, local end-of-line conventions are used as well. The body may be a UNIX-style text file, or a Sun raster image, or a VMS indexed file, or audio data in a system-dependent format stored only in memory, or anything else that corresponds to the local model for the representation of some form of information. Fundamentally, the data is created in the "native" form that corresponds to the type specified by the media type.
Canonical Form	The entire body, including "out-of-band" information such as record lengths and possibly file attribute information, is converted to a universal canonical form. The specific media type of the body as well as its associated attributes dictate the nature of the canonical form that is used. Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types. If character set conversion is involved, however, care must be taken to understand the semantics of the media type, which may have strong implications for any character set conversion (e.g., with regard to syntactically meaningful characters in a text subtype other than "plain").

Table 5: Native and Canonical Form

2.6 S/MIME FUNCTIONALITY

IMP QS (question)-04M

In terms of general functionality, S/MIME is very similar to PGP. Both offer the ability to sign and/or encrypt messages.

S/MIME provides the following functions

1. **Enveloped data:** This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
2. **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.

3. **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
4. **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

2.7 CRYPTOGRAPHIC ALGORITHMS OR CRYPTOGRAPHIC ALGORITHMS USED IN S/MIME

IMP QS (question)-06M

Table 6 summarizes the cryptographic algorithms used in S/MIME. S/MIME uses the following terminology taken from RFC 2119 (Key Words for use in RFCs to Indicate Requirement Levels) to specify the requirement level:

MUST: The definition is an absolute requirement of the specification. An implementation must include this feature or function to be in conformance with the specification.

SHOULD: There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

The S/MIME specification includes a discussion of the procedure for deciding which content encryption algorithm to use. In essence, a sending agent has two decisions to make. First, the sending agent must determine if the receiving agent is capable of decrypting using a given encryption algorithm. Second, if the receiving agent is only capable of accepting weakly encrypted content, the sending agent must decide if it is acceptable to send using weak encryption. To support this decision process, a sending agent may announce its decrypting capabilities in order of preference for any message that it sends out. A receiving agent may store that information for future use.

The following rules, in the following order, should be followed by a sending agent.

1. If the sending agent has a list of preferred decrypting capabilities from an intended recipient, it SHOULD choose the first (highest preference) capability on the list that it is capable of using.
2. If the sending agent has no such list of capabilities from an intended recipient but has received one or more messages from the recipient, then the outgoing

message SHOULD use the same encryption algorithm as was used on the last signed and encrypted message received from that intended recipient.

3. If the sending agent has no knowledge about the decryption capabilities of the intended recipient and is willing to risk that the recipient may not be able to decrypt the message, then the sending agent SHOULD use triple DES.

Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-1. Receiver SHOULD support MD5 for backward compatibility.
Encrypt message digest to form a digital signature.	Sending and receiving agents MUST support DSS. Sending agents SHOULD support RSA encryption. Receiving agents SHOULD support verification of RSA signatures with key sizes 512 bits to 1024 bits.
Encrypt session key for transmission with a message.	Sending and receiving agents SHOULD support Diffie-Hellman. Sending and receiving agents MUST support RSA encryption with key sizes 512 bits to 1024 bits.
Encrypt message for transmission with a one-time session key.	Sending and receiving agents MUST support encryption with tripleDES. Sending agents SHOULD support encryption with AES. Sending agents SHOULD support encryption with RC2/40.
Create a message authentication code.	Receiving agents MUST support HMAC with SHA-1. Sending agents SHOULD support HMAC with SHA-1.

Table 6: Cryptographic Algorithms Used in S/MIME

4. If the sending agent has no knowledge about the decryption capabilities of the intended recipient and is not willing to risk that the recipient may not be able to decrypt the message, then the sending agent MUST use RC2/40.

If a message is to be sent to multiple recipients and a common encryption algorithm cannot be selected for all, then the sending agent will need to send two messages. However, in that case, it is important to note that the security of the message is made vulnerable by the transmission of one copy with lower security.

2.8 S/MIME MESSAGES

IMP QS (question)-04M

S/MIME makes use of a number of new MIME content types, which are shown in Table 7. All of the new application types use the designation PKCS. This refers to a set of public-key cryptography specifications issued by RSA Laboratories and made available for the S/MIME effort.

Type	Subtype	smime Parameter	Description
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the signature.
Application	pkcs7-mime	signedData	A signed S/MIME entity.
	pkcs7-mime	envelopedData	An encrypted S/MIME entity.
	pkcs7-mime	degenerate signedData	An entity containing only public-key certificates.
	pkcs7-mime	CompressedData	A compressed S/MIME entity.
	pkcs7-signature	signedData	The content type of the signature subpart of a multipart/signed message.

Table 7: S/MIME Content Types

2.9 ENVELOPED DATA OR MIME ENVELOPED DATA

The steps for preparing an enveloped Data MIME entity are

1. Generate a pseudorandom session key for a particular symmetric encryption Algorithm (RC2/40 or triple DES).
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as Recipient Info that contains an identifier of the recipient's public-key certificate,2 an identifier of the algorithm used to encrypt the session key, and the encrypted session key.
4. Encrypt the message content with the session key.

2.10 SIGNED DATA

The signed Data smime-type can be used with one or more signers. For clarity, we confine our description to the case of a single digital signature. The steps for preparing a signed Data MIME entity are as follows

1. Select a message digest algorithm (SHA or MD5).
2. Compute the message digest (hash function) of the content to be signed.
3. Encrypt the message digest with the signer's private key.
4. Prepare a block known as Signer Info that contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest. The signed Data entity consists of a series of blocks, including a message digest algorithm identifier, the message being signed, and Signer Info.

The signed Data entity consists of a series of blocks, including a message digest algorithm identifier, the message being signed, and Signer Info.

2.11CLEAR SIGNING

- Clear signing is achieved using the multipart content type with a signed subtype.
- As was mentioned, this signing process does not involve transforming the message to be signed, so that the message is sent "in the clear."
- Thus, recipients with MIME capability but not S/MIME capability are able to read the incoming message.
- A multipart/signed message has two parts.
  - The first part can be any MIME type but must be prepared so that it will not be altered during transfer from source to destination.
  - The Second part has a MIME content type of application and a subtype of pkcs7-signature.

2.12 REGISTRATION REQUEST

- Typically, an application or user will apply to a certification authority for a public-key certificate.
- The application/pkcs10 S/MIME entity is used to transfer a certification request.
- The certification request includes **certification Request Info** block, followed by an identifier of the public-key encryption algorithm, followed by the signature of **the certification Request Info** block made using the sender's private key. The **certification Request Info** block includes a name of the certificate subject (the entity whose public key is to be certified) and a bit-string representation of the user's public key.

2.13 CERTIFICATES-ONLY MESSAGE

A message containing only certificates or a certificate revocation list (CRL) can be sent in response to a registration request. The message is an application/pkcs7-mime type/subtype with a smime-type parameter of degenerate. The steps involved are the same as those for creating a signed Data message, except that there is no message content and the signer Info field is empty.

2.14 S/MIME CERTIFICATE PROCESSING IMP QS (question)-10M

S/MIME uses public-key certificates that conform to version 3 of X.509.

User Agent Role

An S/MIME user has several key-management functions to perform.

1. **Key generation:** The user of some related administrative utility (e.g., one associated with LAN management) MUST be capable of generating separate Diffie-Hellman and DSS key pairs and SHOULD be capable of generating RSA key pairs. Each key pair MUST be generated from a good source of nondeterministic random input and be protected in a secure fashion. A user agent SHOULD generate RSA key pairs with a length in the range of 768 to 1024 bits and MUST NOT generates a length of less than 512 bits.
2. **Registration:** A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.
3. **Certificate storage and retrieval:** A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages. Such a list could be maintained by the user or by some local administrative entity on behalf of a number of users.

VeriSign Certificates

- There are several companies that provide certification authority (CA) services.
- VeriSign provides a CA service that is intended to be compatible with S/MIME and a variety of other applications.
- VeriSign issues X.509 certificates with the product name VeriSign Digital ID.
- The information contained in a Digital ID depends on the type of Digital ID and its use. At a minimum, each Digital ID contains
  1. Owner's public key
  2. Owner's name or alias
  3. Expiration date of the Digital ID
  4. Serial number of the Digital ID
  5. Name of the certification authority that issued the Digital ID
  6. Digital signature of the certification authority that issued the Digital ID
- Digital IDs can also contain other user-supplied information, including
  1. Address
  2. E-mail address

3. Basic registration information (country, zip code, age, and gender)

	Class 1	Class 2	Class 3
Summary of Confirmation of Identity	Automated unambiguous name and e-mail address search.	Same as Class 1, plus automated enrollment information check and automated address check.	Same as Class 1, plus personal presence and ID documents plus Class 2 automated ID check for individuals; business records (or filings) for organizations.
IA Private Key Protection	PCA: trustworthy hardware; CA: trustworthy software or trustworthy hardware.	PCA and CA: trustworthy hardware.	PCA and CA: trustworthy hardware.
Certificate Applicant and Subscriber Private Key Protection	Encryption software (PIN protected) recommended but not required.	Encryption software (PIN protected) required.	Encryption software (PIN protected) required; hardware token recommended but not required.
Applications Implemented or Contemplated by Users	Web-browsing and certain e-mail usage.	Individual and intra- and inter-company e-mail, online subscriptions, password replacement, and software validation.	E-banking, corp. database access, personal banking, membership-based online services, content integrity services, e-commerce server, software validation; authentication of LRAAs; and strong encryption for certain servers.

IA = Issuing Authority
CA = Certification Authority
PCA = VeriSign public primary certification authority
PIN = Personal Identification Number
LRAA = Local Registration Authority Administrator

Table 8: VeriSign Public-Key Certificate Classes

- VeriSign provides three levels, or classes, of security for public-key certificates, as summarized in Table 8. A user requests a certificate online at VeriSign's Web site or other participating Web sites. Class 1 and Class 2 requests are processed on line, and in most cases take only a few seconds to approve. Briefly, the following procedures are used.

1. For **Class 1 Digital IDs,** VeriSign confirms the user's e-mail address by sending a PIN and Digital ID pick-up information to the e-mail address provided in the application.
2. For **Class 2 Digital IDs**, VeriSign verifies the information in the application through an automated comparison with a consumer database in addition to performing all of the checking associated with a Class 1 Digital ID. Finally, confirmation is sent to the specified postal address alerting the user that a Digital ID has been issued in his or her name.

3. For **Class 3 Digital IDs**, VeriSign requires a higher level of identity assurance. An individual must prove his or her identity by providing notarized credentials or applying in person.

2.15 ENHANCED SECURITY SERVICES

As of this writing, three enhanced security services have been proposed in an Internet draft. The details of these may change, and additional services may be added. The three services are

1. **Signed receipts:** A signed receipt may be requested in a Signed Data object. Returning a signed receipt provides proof of delivery to the originator of a message and allows the originator to demonstrate to a third party that the recipient received the message. In essence, the recipient signs the entire original Message plus the original (sender's) signature and appends the new signature to form a new S/MIME message.
2. **Security labels:** A security label may be included in the authenticated attributes of a Signed Data object. A security label is a set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation. The labels may be used for access control, by indicating which users are permitted access to an object. Other uses include priority (secret, confidential, restricted, and so on) or role based, describing which kind of people can see the information (e.g., patient's health-care team, medical billing agents, etc.).
3. **Secure mailing lists:** When a user sends a message to multiple recipients, a certain amount of per-recipient processing is required, including the use of each recipient's public key. The user can be relieved of this work by employing the services of an S/MIME Mail List Agent (MLA). An MLA can take a single incoming message, perform the recipient-specific encryption for each recipient, and forward the message. The originator of a message need only send the message to the MLA with encryption performed using the MLA's public key.

3 DOMAIN KEYS IDENTIFIED MAIL IMP QS (question)-06M

- Domain Keys Identified Mail (DKIM) is a specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream.

- Message recipients (or agents acting in their behalf) can verify the signature by querying the signer's domain directly to retrieve the appropriate public key and thereby can confirm that the message was attested to by a party in possession of the private key for the signing domain.
- DKIM is a proposed Internet Standard (RFC 4871: Domain Keys Identified Mail (DKIM) Signatures). DKIM has been widely adopted by a range of e-mail providers, including corporations, government agencies, Gmail, yahoo, and many Internet Service Providers (ISPs).

3.1 INTERNET MAIL ARCHITECTURE IMP QS (question)-10M JULY-2019(10M)

To understand the operation of DKIM, it is useful to have a basic grasp of the Internet mail architecture, which is currently defined in RFC 5598.

At its most fundamental level, the Internet mail architecture consists of a user world in the form of Message User Agents (MUA), and the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTA).

Figure 9 illustrates the key components of the Internet mail architecture, which include the following.

1. **Message User Agent (MUA):** Operates on behalf of user actors and user applications. It is their representative within the e-mail service. Typically, this function is housed in the user's computer and is referred to as a client e-mail program or a local network e-mail server. The author MUA formats a message and performs initial submission into the MHS via a MSA. The recipient MUA processes received mail for storage and/or display to the recipient user.
2. **Mail Submission Agent (MSA):** Accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards. This function may be located together with the MUA or as a separate functional model. In the latter case, the Simple Mail Transfer Protocol (SMTP) is used between the MUA and the MSA.
3. **Message Transfer Agent (MTA):** Relays mail for one application-level hop. It is like a packet switch or IP router in that its job is to make routing assessments and to move the message closer to the recipients. Relaying is performed by a sequence of MTAs until the message reaches a destination MDA. An MTA also

adds trace information to the message header. SMTP is used between MTAs and between an MTA and an MSA or MDA.

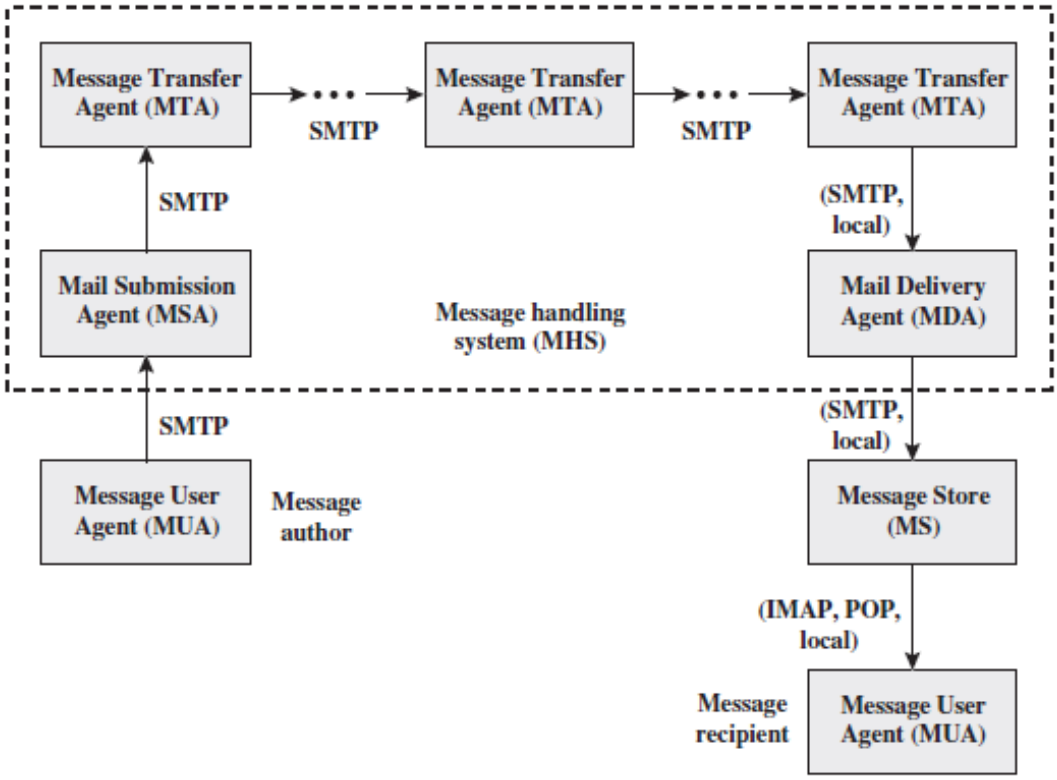


Figure 9: Function Modules and Standardized Protocols Used Between Them or Internet mail architecture

4. **Mail Delivery Agent (MDA):** Responsible for transferring the message from the MHS to the MS.
5. **Message Store (MS):** An MUA can employ a long-term MS. An MS can be located on a remote server or on the same machine as the MUA. Typically, an MUA retrieves messages from a remote server using POP (Post Office Protocol) or IMAP (Internet Message Access Protocol).

Two other concepts need to be defined

1. An **administrative management domain (ADMD)** is an Internet e-mail provider.
2. The **Domain Name System (DNS)** is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address.

3.2 E-MAIL THREATS

IMP QS (question)-10M

RFC 4686 (Analysis of Threats Motivating Domain Keys Identified Mail) describes the threats being addressed by DKIM in terms of the **characteristics, capabilities, and location** of potential attackers.

3.2.1 Characteristics

RFC 4686 characterizes the range of attackers on a spectrum of three levels of threat.

1. At the low end are attackers who simply want to send e-mail that a recipient does not want to receive. The attacker can use one of a number of commercially available tools that allow the sender to falsify the origin address of messages. This makes it difficult for the receiver to filter spam on the basis of originating address or domain.
2. At the next level are professional senders of bulk spam mail. These attackers often operate as commercial enterprises and send messages on behalf of third parties. They employ more comprehensive tools for attack, including Mail Transfer Agents (MTAs) and registered domains and networks of compromised computers (zombies) to send messages and (in some cases) to harvest addresses to which to send.
3. The most sophisticated and financially motivated senders of messages are those who stand to receive substantial financial benefit, such as from an e- mail-based fraud scheme. These attackers can be expected to employ all of the above mechanisms and additionally may attack the Internet infrastructure itself, including DNS cache-poisoning attacks and IP routing attacks.

3.2.2 Capabilities

RFC 4686 lists the following as capabilities that an attacker might have.

1. Submit messages to MTAs and Message Submission Agents (MSAs) at multiple locations in the Internet.
2. Construct arbitrary Message Header fields, including those claiming to be mailing lists, resenders, and other mail agents.
3. Sign messages on behalf of domains under their control.
4. Generate substantial numbers of either unsigned or apparently signed messages that might be used to attempt a denial-of-service attack.

5. Resend messages that may have been previously signed by the domain.
6. Transmit messages using any envelope information desired.
7. Act as an authorized submitter for messages from a compromised computer.
8. Manipulation of IP routing. This could be used to submit messages from specific IP addresses or difficult-to-trace addresses, or to cause diversion of messages to a specific domain.
9. Limited influence over portions of DNS using mechanisms such as cache poisoning. This might be used to influence message routing or to falsify advertisements of DNS-based keys or signing practices.
10. Access to significant computing resources, for example, through the conscription of worm-infected "zombie" computers. This could allow the "bad actor" to perform various types of brute-force attacks.
11. Ability to eavesdrop on existing traffic, perhaps from a wireless network.

3.2.3 Location

- DKIM focuses primarily on attackers located outside of the administrative units of the claimed originator and the recipient.
- These administrative units frequently correspond to the protected portions of the network adjacent to the originator and recipient.
- It is in this area that the trust relationships required for authenticated message submission do not exist and do not scale adequately to be practical.
- Conversely, within these administrative units, there are other mechanisms (such as authenticated message submission) that are easier to deploy and more likely to be used than DKIM.
- External "bad actors" are usually attempting to exploit the "any-to-any" nature of e-mail that motivates most recipient MTAs to accept messages from anywhere for delivery to their local domain. They may generate messages without signatures, with incorrect signatures, or with correct signatures from domains with little traceability. They may also pose as mailing lists, greeting cards, or other agents that legitimately send or resend messages on behalf of others.

3.3 DKIM STRATEGY

IMP QS (question)-10M

DKIM is designed to provide an e-mail authentication technique that is transparent to the end user.

- In essence, a user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates. The signature covers all of the content of the message and some of the RFC 5322 message headers.
- At the receiving end, the MDA can access the corresponding public key via a DNS and verify the signature, thus authenticating that the message comes from the claimed administrative domain.
- Thus, mail that originates from somewhere else but claims to come from a given domain will not pass the authentication test and can be rejected. This approach differs from that of S/MIME and PGP, which use the originator's private key to sign the content of the message.
- The motivation for DKIM is based on the following reasoning.

1. S/MIME depends on both the sending and receiving users employing S/MIME. For almost all users, the bulk of incoming mail does not use S/MIME, and the bulk of the mail the user wants to send is to recipients not using S/MIME.
2. S/MIME signs only the message content. Thus, RFC 5322 header information concerning origin can be compromised.
3. DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need take no action.
4. DKIM applies to all mail from cooperating domains.
5. DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders.

The operation of DKIM or Simple Example of DKIM Deployment

Figure 10 is a simple example of the operation of DKIM.

- We begin with a Message generated by a user and transmitted into the MHS to an MSA that is within the user's administrative domain.
- An e-mail message is generated by an e-mail client program. The content of the message, plus selected RFC 5322 headers, is signed by the e-mail provider using the provider's private key.

- The signer is associated with a domain, which could be a corporate local network, an ISP, or a public e-mail facility such as Gmail.
- The signed message then passes through the Internet via a sequence of MTAs. At the destination, the MDA retrieves the public key for the incoming signature and verifies the signature before passing the message on to the destination e-mail client.

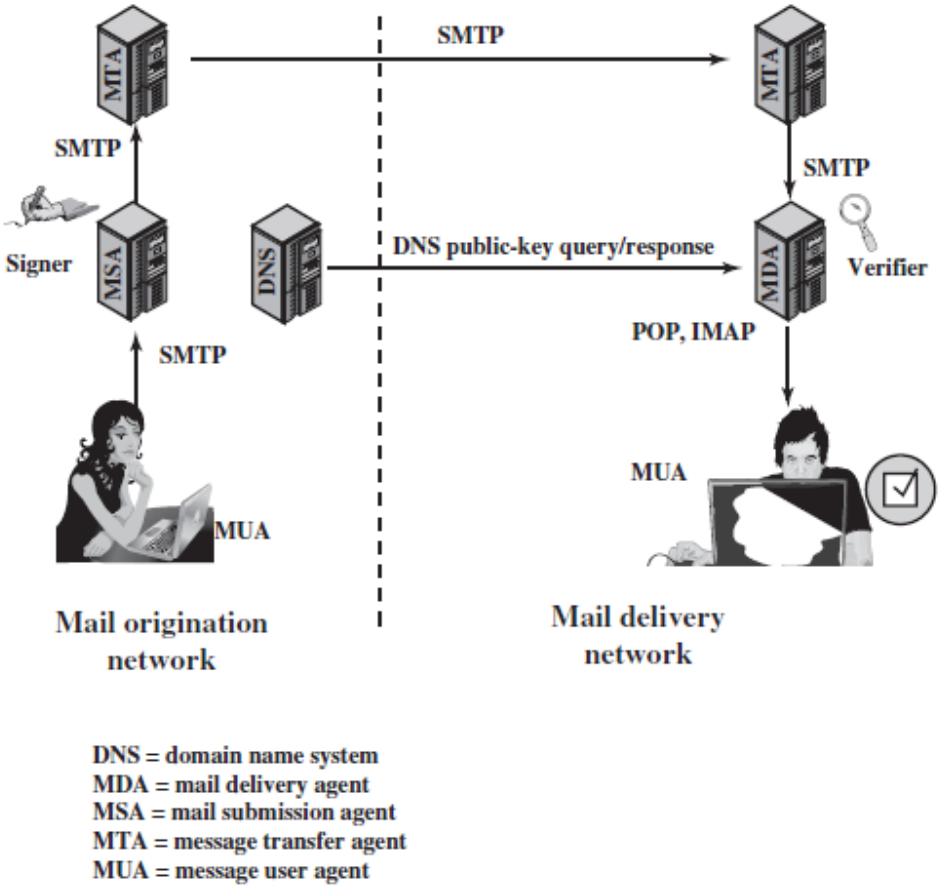


Figure 10 Simple Example of DKIM Deployment

- The default signing algorithm is RSA with SHA-256. RSA with SHA-1 also may be used.

3.4 DKIM FUNCTIONAL FLOW

IMP QS (question)-06M

- **Figure 11** provides a more detailed look at the elements of DKIM operation. Basic message processing is divided between a signing Administrative Management Domain (ADMD) and a verifying ADMD.
- At its simplest, this is between the originating ADMD and the delivering ADMD, but it can involve other ADMDs in the handling path.

- Signing is performed by an authorized module within the signing ADMD and uses private information from a Key Store. Within the originating ADMD, this might be performed by the MUA, MSA, or an MTA.

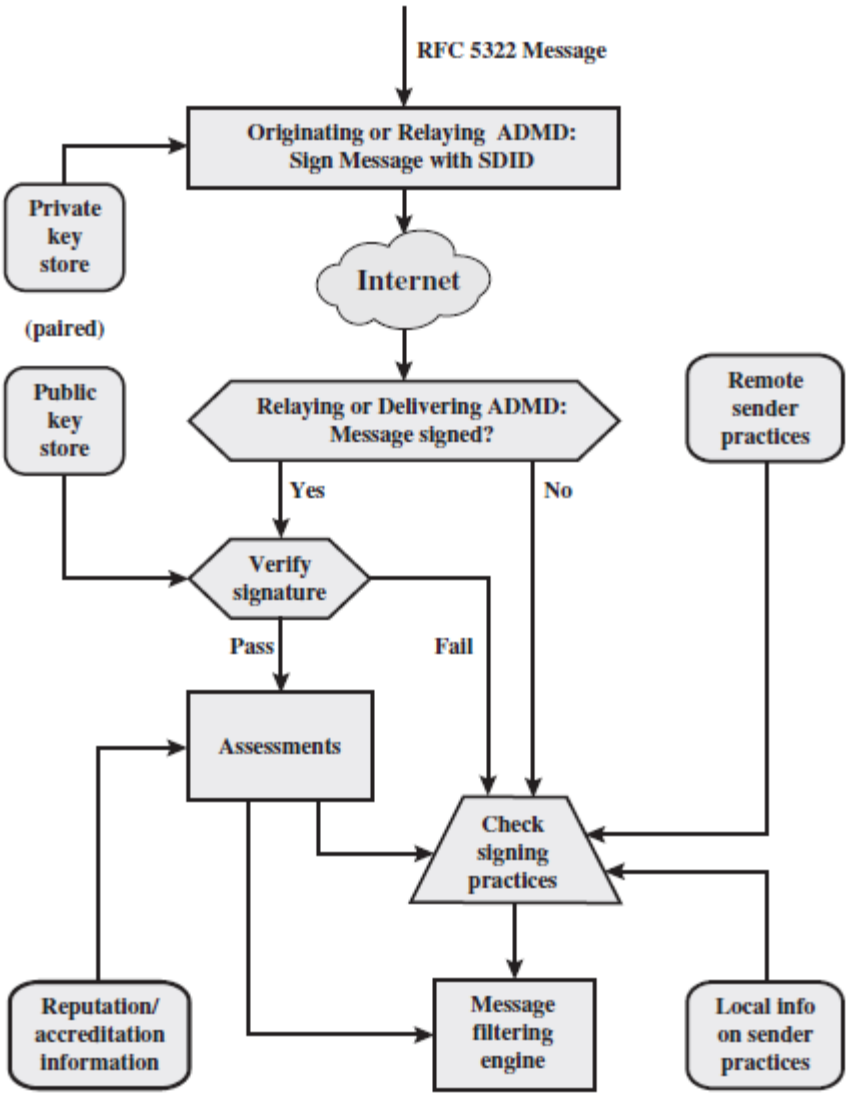


Figure 11 DKIM Functional Flow

- Verifying is performed by an authorized module within the verifying ADMD. Within a delivering ADMD, verifying might be performed by an MTA, MDA, or MUA.
- The module verifies the signature or determines whether a particular signature was required.
- Verifying the signature uses public information from the Key Store.
- If the signature passes, reputation information is used to assess the signer and that information is passed to the message filtering system.

- If the signature fails or there is no signature using the author's domain, information about signing practices related to the author can be retrieved remotely and/or locally, and that information is passed to the message filtering system.(For example, if the sender (e.g., Gmail) uses DKIM but no DKIM signature is present, then the message may be considered fraudulent)
- The signature includes a number of fields. Each field begins with a tag consisting of a tag code followed by an equals sign and ends with a semicolon. The fields include the following:

- **v** = DKIM version.
- **a** = Algorithm used to generate the signature; must be either rsa-sha1 or rsa-sha256.
- **c** = Canonicalization method used on the header and the body.
- **d** = A domain name used as an identifier to refer to the identity of a responsible person or organization. In DKIM, this identifier is called the Signing Domain Identifier (SDID). In our example, this field indicates that the sender is using a Gmail address.
- **s** = In order that different keys may be used in different circumstances for the same signing domain (allowing expiration of old keys, separate departmental signing, or the like), DKIM defines a selector (a name associated with a key), which is used by the verifier to retrieve the proper key during signature verification.
- **h** = Signed Header fields. A colon-separated list of header field names that identify the header fields presented to the signing algorithm. Note that in our example above, the signature covers the domain key-signature field. This refers to an older algorithm (since replaced by DKIM) that is still in use.
- **bh** = The hash of the canonicalized body part of the message. This provides additional information for diagnosing signature verification failures.
- **b** = the signature data in base64 format; this is the encrypted hash code.

QUESTION BANK – NETWORK AND CYBER SECURITY

MODULE-2

1. Explain PGP. **06M**
2. With a neat diagrams, Explain PGP Cryptographic Functions or PGP Functions (Authentication, Confidentiality, Confidentiality and Authentication). **14M**
3. With a neat diagram, Explain E-mail Compatibility or Transmission and Reception of PGP Messages. **08M**
4. With a neat diagram, explain key identifiers or PGP message format. **08M**
5. With a neat diagram, Explain PGP message generations or PGP message transmission and reception or key rings. **12M**
6. With a neat diagram, explain RADIX-64 conversion. **06M OR 08M**
7. Explain S/MIME. **06M**
8. Explain RFC 5322. **06M**
9. Discuss multipurpose internet mail extensions (MIME). **10M**
10. Discuss MIME content types. **08M or 10M**
11. Short note on 1) MIME transfer encodings 2) native and canonical form 3) S/MIME functionality 4) S/MIME messages. **12M or 14M**
12. Discuss cryptographic algorithms or cryptographic algorithms used in S/MIME .**06M**
13. Discuss S/MIME certificate processing. **8M or 10M**
14. Explain domain keys identified mail. **06M**
15. With a neat diagram, explain internet mail architecture. **10M**
16. Discuss E-MAIL threats**.** **10M**
17. With a neat diagram, explain DKIM strategy OR DKIM Deployment. **10M**
18. With a neat diagram, explain DKIM functional flow. **10M**

NETWORK AND CYBER SECURITY
(15EC835, 17EC835)

8TH SEM E&C

JAYANTH DWIJESH H P **BE (ECE), M.tech (DECS).**

Assistant Professor – Dept of E&CE, BGSIT.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

B.G.S INSTITUTE OF TECHNOLOGY (B.G.S.I.T)

B.G Nagara, Nagamangala Tq, Mandya District- 571448

NETWORK AND CYBER SECURITY

MODULE–3

MODULE–3

IP SECURITY: IP Security Overview, IP Security Policy, Encapsulation Security Payload (ESP), combining security Associations, Internet Key Exchange. Cryptographic Suites

TEXT BOOK:

1. **William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325-1877-3.**
2. **Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.**

REFERENCE BOOKS:

1. Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.

2. Cryptography and Network Security, Atul Kahate, TMH, 2003.

MODUL-3:- IP SECURITY: IP Security Overview, IP Security Policy, Encapsulation Security Payload (ESP), combining security Associations, Internet Key Exchange. Cryptographic Suites

1 IP SECURITY OVERVIEW

IMP QS (question)-03M

In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). The report identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user- to-end-user traffic using authentication and encryption mechanisms.

To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6. This means that vendors can begin offering these features now, and many vendors now do have some IPsec capability in their products. The IPsec specification now exists as a set of Internet standards.

1.1 APPLICATIONS OF IPsec

IMP QS (question)-06M

IPsec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include:

1. **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
2. **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for travelling employees and telecommuters.
3. **Establishing extranet and intranet connectivity with partners:** IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
4. **Enhancing electronic commerce security:** Even though some Web and electronic Commerce applications have built-in security protocols; the use of

IPsec enhances that security. IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.

1.2 IP SECURITY SCENARIO

IMP QS (question)-08M

Figure 1 is a typical scenario of IPsec usage.

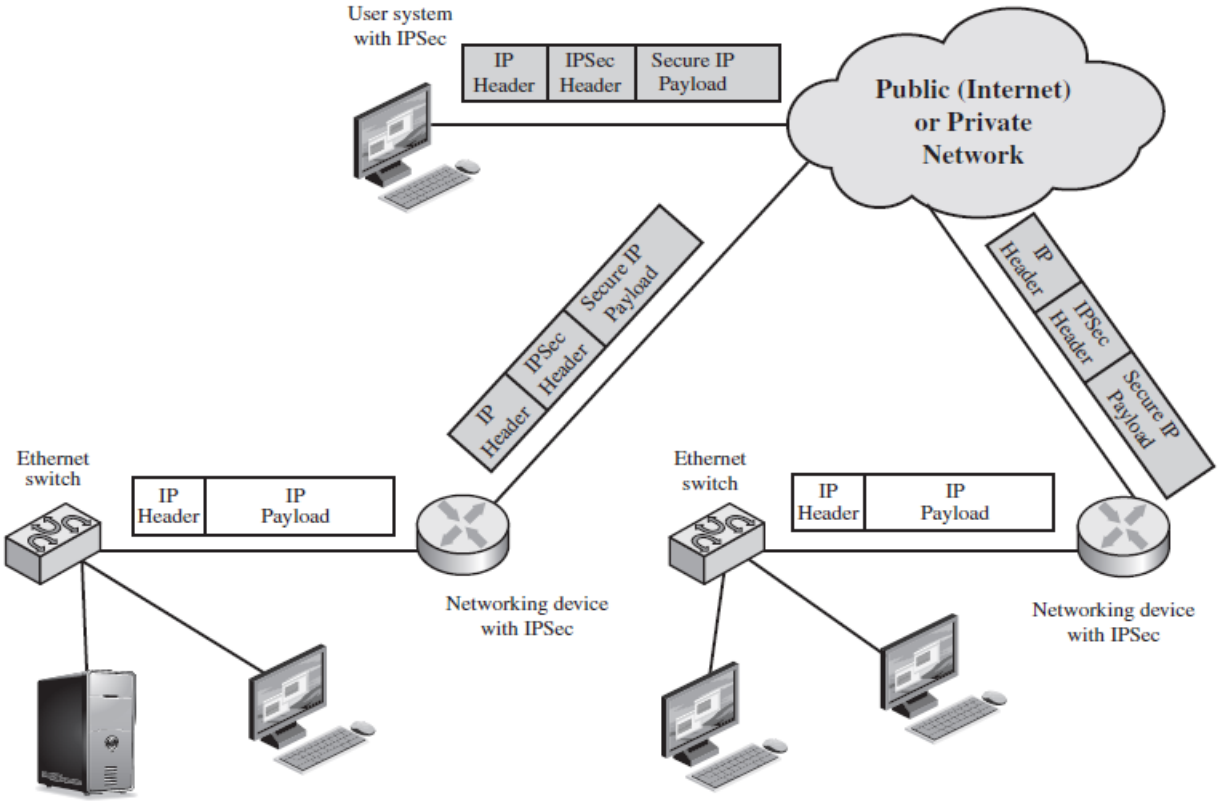


Figure 1: IP Security Scenario

- An organization maintains LANs at dispersed locations. Non secure IP traffic is conducted on each LAN.
- For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.
- These protocols operate in networking devices, such as a router or firewall, that Connect each LAN to the outside world.
- The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN.
- Secure transmission is also possible with individual users who dial into the WAN.

Such user workstations must implement the IPsec protocols to provide security.

1.3 BENEFITS OF IPSEC

IMP QS (question)-04M

Some of the benefits of IPsec:

- hen IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter. Traffic within a company or workgroup does not incur the overhead of security-related processing.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- IPsec is below the transport layer (TCP, UDP) and so is transparent to applications.
- There is no need to change software on a user or server system when IPsec is implemented in the firewall or router.
- Even if IPsec is implemented in end systems, upper-layer software, including applications, is not affected.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying material on a per-user basis, or revoke keying material when users leave the organization.
- IPsec can provide security for individual users if needed. This is useful for offsite workers and for setting up a secure virtual sub network within an organization for sensitive applications.

1.4 ROUTING APPLICATIONS

In addition to supporting end users and protecting premises systems and networks, IPsec can play a vital role in the routing architecture required for internetworking. Lists the following examples of the use of IPsec. IPsec can assure that

- A router advertisement (a new router advertises its presence) comes from an authorized router.
- A neighbour advertisement (a router seeks to establish or maintain a neighbour Relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial IP packet was sent.

- A routing update is not forged.

Routing protocols such as Open Shortest Path First (OSPF) should be run on top of security associations between routers that are defined by IPsec.

1.5 IPsec DOCUMENTS

IMP QS (question)-05M

IPsec encompasses three functional areas: authentication, confidentiality, and key management.

The documents can be categorized into the following groups.

- **Architecture:** Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology. The current specification is RFC 4301, Security Architecture for the Internet Protocol.
- **Authentication Header (AH):** AH is an extension header to provide message Authentication. The current specification is RFC 4302, IP Authentication Header. Because message authentication is provided by ESP, the use of AH is deprecated. It is included in IPsecv3 for backward compatibility but should not be used in new applications. We do not discuss AH in this chapter.
- **Encapsulating Security Payload (ESP):** ESP consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication. The current specification is RFC 4303, IP Encapsulating Security Payload (ESP).
- **Internet Key Exchange (IKE):** This is a collection of documents describing the key management schemes for use with IPsec. The main specification is RFC 5996, Internet Key Exchange (IKEv2) Protocol, but there are a number of related RFCs.
- **Cryptographic algorithms:** This category encompasses a large set of documents that define and describe cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange.
- **Other:** There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content.

1.6 IPsec SERVICES

- IPsec provides security services at the IP layer by enabling a system to select required Security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.
- **Two protocols** are used to provide security: an authentication protocol designated by the header of the protocol. Authentication Header (AH); and a combined encryption/ authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP).
- RFC 4301 lists the following services:
  - Access control.
  - Connectionless integrity.
  - Data origin authentication.
  - Rejection of replayed packets (a form of partial sequence integrity).
  - Confidentiality (encryption).
  - Limited traffic flow confidentiality.

1.7 TRANSPORT AND TUNNEL MODES IMP QS (question)-9M

Both Authentication Headers (AH) and Encapsulating Security Payload (ESP). support two modes of use: transport and tunnel mode.

1.7.1 Transport Mode

- Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.
- When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header.
- For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.
- **ESP in transport mode** encrypts and optionally authenticates the IP payload but not the IP header.

- **AH in transport mode** authenticates the IP payload and selected portions of the IP header.

TABLE 1 SUMMARIZES TRANSPORT AND TUNNEL MODE FUNCTIONALITY.

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

Table 1: Tunnel Mode and Transport Mode Functionality

1.7.2 Tunnel Mode

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

2 IP SECURITY POLICY

IMP QS (question)-04M

Fundamental to the operation of IPsec is the concept of a security policy applied to each IP packet that transits from a source to a destination.

IPsec policy is determined primarily by the interaction of two databases, the **security association database (SAD)** and the **security policy database (SPD)**.

This section provides an overview of these two databases and then summarizes their use during IPsec operation. **Figure 2** illustrates the relevant relationships.

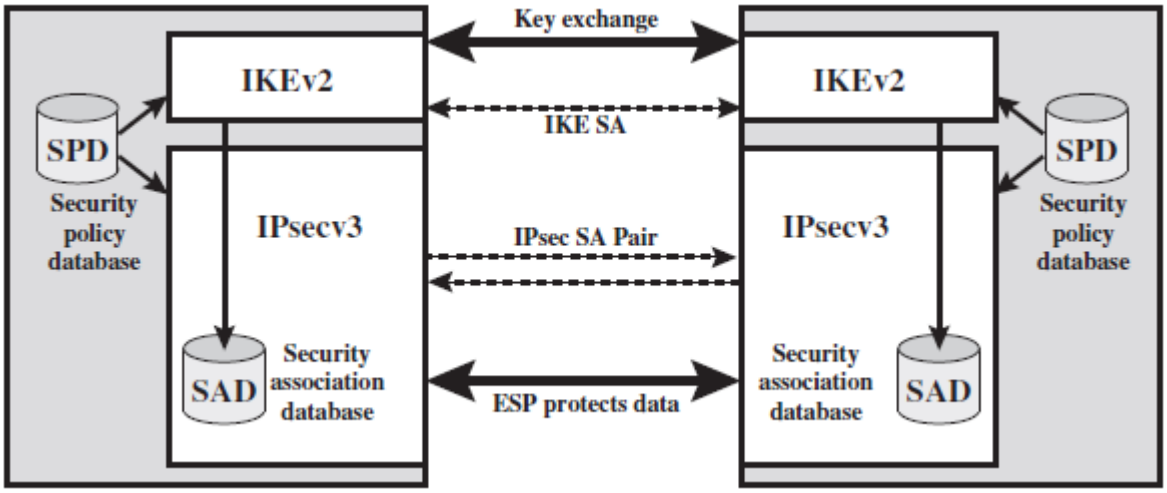


Figure 2 IPsec Architecture

2.1 SECURITY ASSOCIATIONS

IMP QS (question)-03M

- A key concept that appears in both the authentication and confidentiality mechanisms for IP is the security association (SA).
- An association is a one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it.
- If a peer relationship is needed for two-way secure exchange, then two security associations are required.
- A security association is uniquely identified by three parameters.
  - **Security Parameters Index (SPI):** A 32-bit unsigned integer assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
  - **IP Destination Address:** This is the address of the destination endpoint of the SA, which may be an end-user system or a network system such as a firewall or router.
  - **Security Protocol Identifier:** This field from the outer IP header indicates whether the association is an AH or ESP security association.

- Hence, in any IP packet, the security association is uniquely identified by the Destination Address in the IPv4 or IPv6 header and the SPI in the enclosed extension header (AH or ESP).

2.2 SECURITY ASSOCIATION DATABASE

IMP QS (question)-05M

- In each IPsec implementation, there is a nominal2 Security Association Database that defines the parameters associated with each SA.
- A security association is normally defined by the following parameters in an SAD entry.

1. **Security Parameter Index:** A 32-bit value selected by the receiving end of an SA to uniquely identify the SA. In an SAD entry for an outbound SA, the SPI is used to construct the packet's AH or ESP header. In an SAD entry for an inbound SA, the SPI is used to map traffic to the appropriate SA.
2. **Sequence Number Counter:** A 32-bit value used to generate the Sequence Number field in AH or ESP headers, described in Section 20.3 (required for all implementations).
3. **Sequence Counter Overflow:** A flag indicating whether overflow of the Sequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA (required for all implementations).
4. **Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is a replay. (Required for all implementations).
5. **AH Information:** Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations).
6. **ESP Information:** Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations).
7. **Lifetime of this Security Association:** A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur (required for all implementations).
8. **IPsec Protocol Mode:** Tunnel, transport, or wildcard.

9. **Path MTU:** Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations).

2.3 SECURITY POLICY DATABASE

IMP QS (question)-06M

- The means by which IP traffic is related to specific SAs (or no SA in the case of traffic allowed to bypass IPsec) is the nominal Security Policy Database (SPD).
- In its simplest form, an SPD contains entries, each of which defines a subset of IP traffic and points to an SA for that traffic.
- In more complex environments, there may be multiple entries that potentially relate to a single SA or multiple SAs associated with a single SPD entry.
- Each SPD entry is defined by a set of IP and upper-layer protocol field values, called selectors.
- In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA.
- Outbound processing obeys the following general sequence for each IP packet.
  1. Compare the values of the appropriate fields in the packet (the selector fields) against the SPD to find a matching SPD entry, which will point to zero or more SAs.
  2. Determine the SA if any for this packet and its associated SPI.
  3. Do the required IPsec processing (i.e., AH or ESP processing).
- The following selectors determine an SPD entry:
  - **Remote IP Address:** This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one destination system sharing the same SA (e.g., behind a firewall).
  - **Local IP Address:** This may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address. The latter two are required to support more than one source system sharing the same SA (e.g., behind a firewall).
  - **Next Layer Protocol:** The IP protocol header (IPv4, IPv6, or IPv6 Extension) includes a field (Protocol for IPv4, Next Header for IPv6 or IPv6 Extension) that designates the protocol operating over IP. This is an individual protocol

number, ANY, or for IPv6 only, OPAQUE. If AH or ESP is used, then this IP protocol header immediately proceeds the AH or ESP header in the packet.

- **Name:** A user identifier from the operating system. This is not a field in the IP or upper-layer headers but is available if IPsec is running on the same operating system as the user.
- **Local and Remote Ports:** These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

Table 2 provides an example of an SPD on a host system (as opposed to a network system such as a firewall or router).

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

Table 2: Host SPD Example

2.4 IP TRAFFIC PROCESSING

IMP QS (question)-10M

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

OUTBOUND PACKETS

- Figure 3 highlights the main elements of IPsec processing for outbound traffic.
- A block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body. Then the following steps occur:-

1. IPsec searches the SPD for a match to this packet.
2. If no match is found, then the packet is discarded and an error message is generated.
3. If a match is found, further processing is determined by the first matching entry in the SPD. If the policy for this packet is DISCARD, then the packet is discarded.

If the policy is BYPASS, then there is no further IPsec processing; the packet is forwarded to the network for transmission.

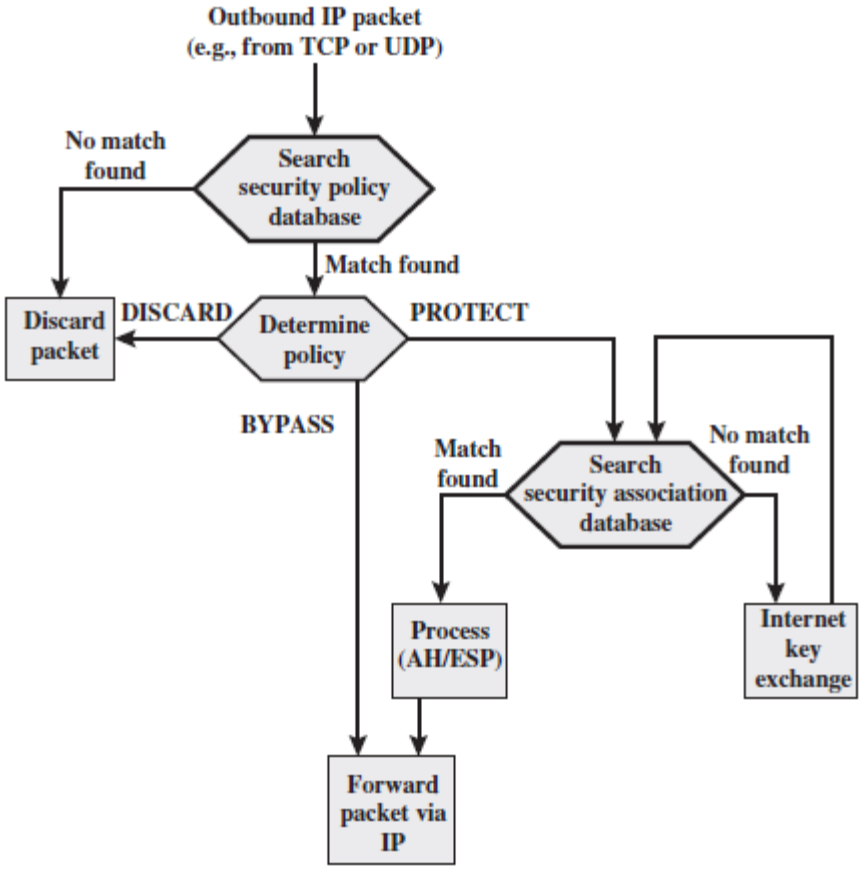


Figure 3 Processing Model for Outbound Packets

4. If the policy is PROTECT, then a search is made of the SAD for a matching entry. If no entry is found, then IKE is invoked to create an SA with the appropriate keys and an entry is made in the SA.
5. The matching entry in the SAD determines the processing for this packet. Encryption, authentication, or both can be performed, and either transport or tunnel mode can be used. The packet is then forwarded to the network for transmission.

INBOUND PACKETS

Figure 4 highlights the main elements of IPsec processing for inbound traffic. An incoming IP packet triggers the IPsec processing. The following steps occur:

1. IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6).

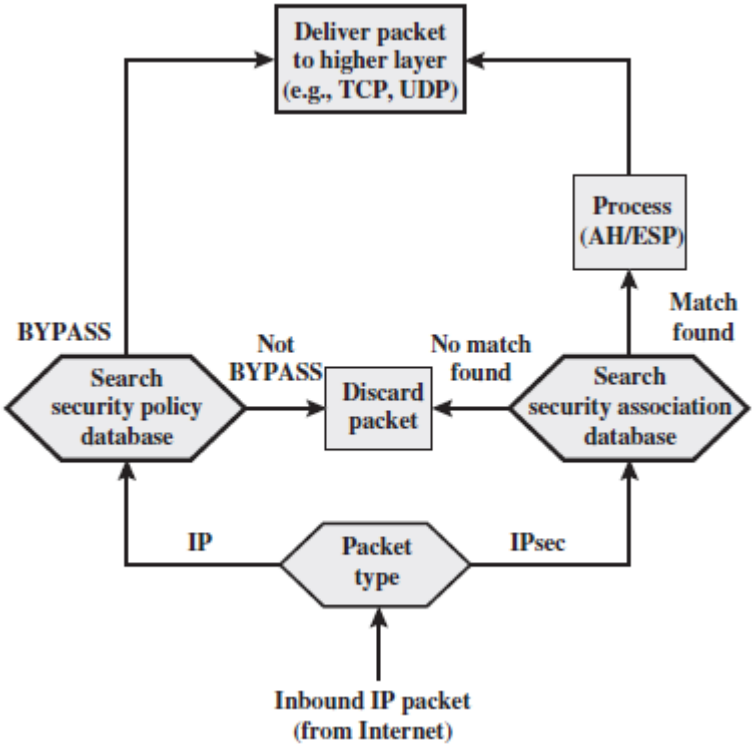


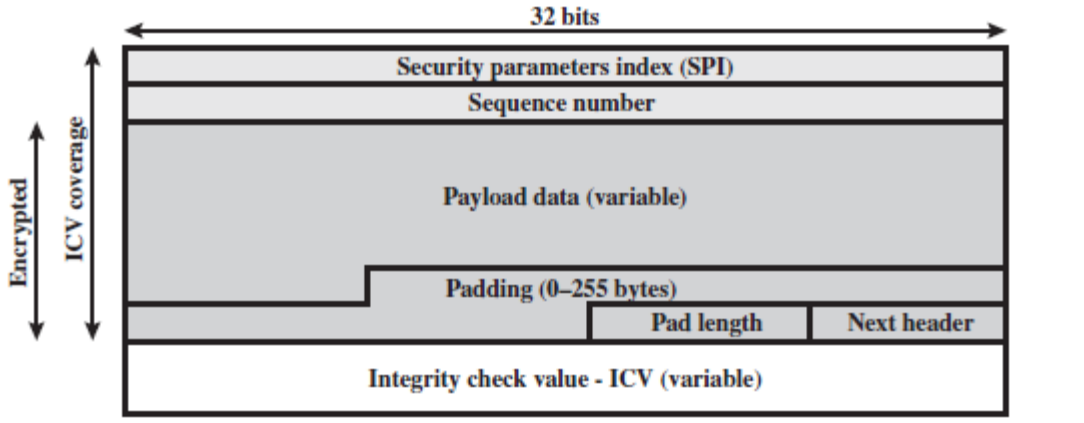
Figure 4 Processing Model for Inbound Packets

2. If the packet is unsecured, IPsec searches the SPD for a match to this packet. If the first matching entry has a policy of BYPASS, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP. If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.
3. For a secured packet, IPsec searches the SAD. If no match is found, the packet is discarded. Otherwise, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP.

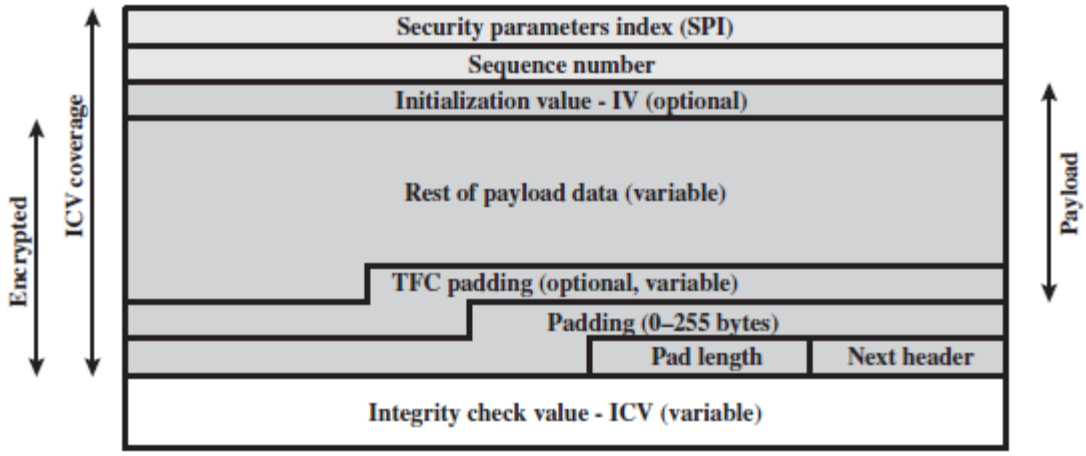
3 ENCAPSULATING SECURITY PAYLOAD IMP QS (question)-03M

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

3.1 ESP Format IMP QS (question)-08M



(a) Top-level format of an ESP Packet



(b) Substructure of payload data

Figure 5 ESP Packet Format

Figure 5(a) shows the top-level format of an ESP packet. It contains the following Fields.

1. **Security Parameters Index (32 bits):** Identifies a security association.
2. **Sequence Number (32 bits):** A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
3. **Payload Data (variable):** This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
4. **Padding (0–255 bytes):** The purpose of this field is discussed later.
5. **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding this field.
6. **Next Header (8 bits):** Identifies the type of data contained in the payload data field by identifying the first header in that payload (e.g., an extension header in IPv6, or an upper-layer protocol such as TCP).
7. **Integrity Check Value (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

Figure 5(b).

Two additional fields may be present in the payload **figure 5(b).** an **initialization value (iv)**, or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for esp. if tunnel mode is being used, then the IPsec implementation may add **traffic flow confidentiality (TFC)** padding after the payload data and before the padding field, as explained subsequently.

3.2 ENCRYPTION AND AUTHENTICATION ALGORITHMS

- The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service.
- If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an **initialization vector (IV**), then these data may be carried explicitly at the beginning of the Payload Data field.
- If included, an **initialization vector (IV**), is usually not encrypted, although it is often referred to as being part of the cipher text.

- The ICV field is optional. It is present only if the integrity service is selected and is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV.
- The **Integrity check value (ICV)** is computed after the encryption is performed.
- Note that the ICV is not protected by encryption a keyed integrity algorithm must be employed to compute the **Integrity check value (ICV).**

3.3 PADDING

The Padding field serves several purposes:

- If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.
- The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the cipher text must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.
- Additional padding may be added to provide partial traffic-flow confidentiality by concealing the actual length of the payload.

3.4 ANTI – REPLY SERVICE

IMP QS (question)-06M

- A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence.
- The sequence number field is used to thwart the reply attack.

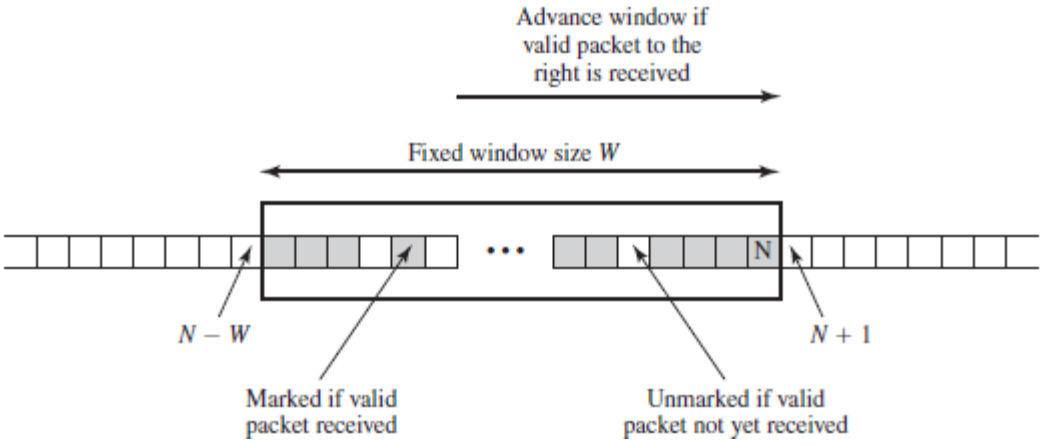


Figure 6 Anti-replay Mechanism

FIGURE 6:–

- The sequence number is set to zero with a new SA (Security Associations) established
- The number is incremented by 1 for each packet sent on the SA.
- The SA is terminated or negotiated with a new key is $N=2^{32}-1$
- A window of size W is implemented in order for IP packed to be delivered in reliable manner (with a default of w=64).
- The right edge of the window represents the highest sequence number, *N*, so far received for a valid packet.
- For any packet with a sequence number in the range from *N* - *W* + 1 to *N* that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked (Figure 6). Inbound processing proceeds as follows when a packet is received:
  1. If the received packet falls within the window and is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked.
  2. If the received packet is to the right of the window and is new, the MAC is Checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked.
  3. If the received packet is to the left of the window or if authentication fails, the packet is discarded; this is an auditable event.

3.5 TRANSPORT AND TUNNEL MODES

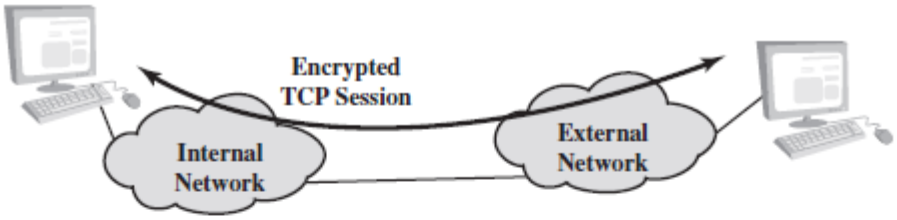
IMP QS (question)-08M

FIGURE 7 SHOWS

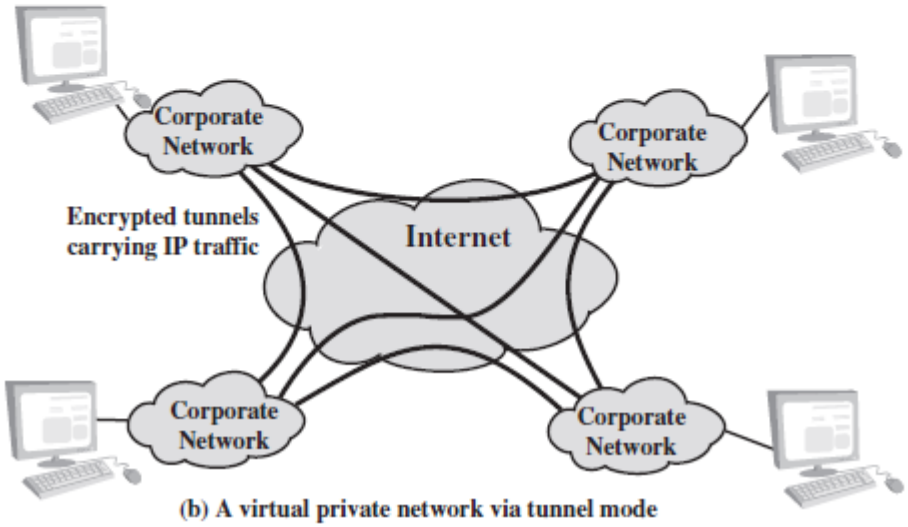
- Two ways in which the IPsec ESP service can be used.
- In the **upper part of the figure**, encryption (and optionally authentication) is provided directly between two hosts.
- **Figure 7(b)** shows how tunnel mode operation can be used to set up a **virtual private network**.

In this example

- An organization has four private networks interconnected across the Internet. Hosts on the internal networks use the Internet for transport of data but do not interact with other Internet-based hosts.
- By terminating the tunnels at the security gateway to each internal network, the configuration allows the hosts to avoid implementing the security capability.



(a) Transport-level security



(b) A virtual private network via tunnel mode

Figure 7 Transport-Modes versus Tunnel-Mode Encryption

- The former technique is supported by a transport mode SA, while the latter technique uses a tunnel mode SA.

➢ The scope of ESP for the two modes. The considerations are somewhat different for IPv4 and IPv6. We use the packet formats of **Figure 8(a)** as a starting point.

3.5.1 Transport Mode ESP

IMP QS (question)-08M

Transport mode ESP is used to encrypt and optionally authenticate the data carried by IP (e.g., a TCP segment), as shown in Figure 8(b.) .

AS SHOWN IN FIGURE 8(B.)

IPv4:-

- For this mode using IPv4, the ESP header is inserted into the IP packet immediately prior to the transport-layer header (e.g., TCP, UDP, ICMP), and an ESP trailer (Padding, Pad Length, and Next Header fields) is placed after the IP packet.
- If authentication is selected, the ESP Authentication Data field is added after the ESP trailer.
- The entire transport-level segment plus the ESP trailer are encrypted. Authentication covers all of the cipher text plus the ESP header.

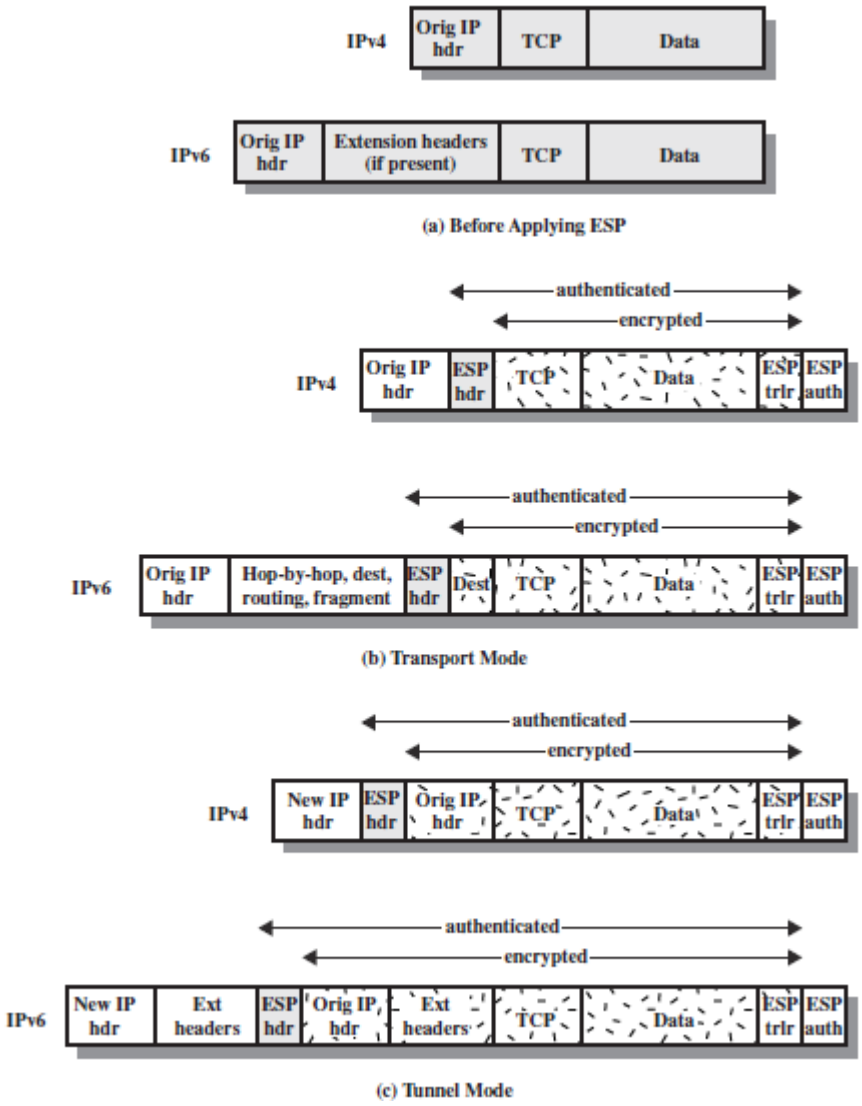


Figure 8 Scopes of ESP Encryption and Authentication

IPv6:-

- In the context of IPv6, ESP is viewed as an end-to-end payload; that is, it is not examined or processed by intermediate routers.
- Therefore, the ESP header appears after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers.
- The destination options extension header could appear before or after the ESP header, depending on the semantics desired.
- For IPv6, encryption covers the entire transport-level segment plus the ESP trailer plus the destination options extension header if it occurs after the ESP header.
- Again, authentication covers the cipher text plus the ESP header.

TRANSPORT MODE OPERATION MAY BE SUMMARIZED AS FOLLOWS.

1. At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its cipher text to form the IP packet for transmission. Authentication is added if this option is selected.
2. The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but does not need to examine the cipher text.
3. The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment.

ADVANTAGES AND DRAWBACKS

- Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application.
- One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets.

3.5.2 Tunnel Mode ESP

IMP QS (question)-06M

- Tunnel mode ESP is used to encrypt an entire IP packet **(FIGURE 8C).**

- For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted.
- This method can be used to counter traffic analysis. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header.
- Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus cipher text plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis.

FIGURE 9 SHOWS THE PROTOCOL ARCHITECTURE FOR THE TWO MODES.

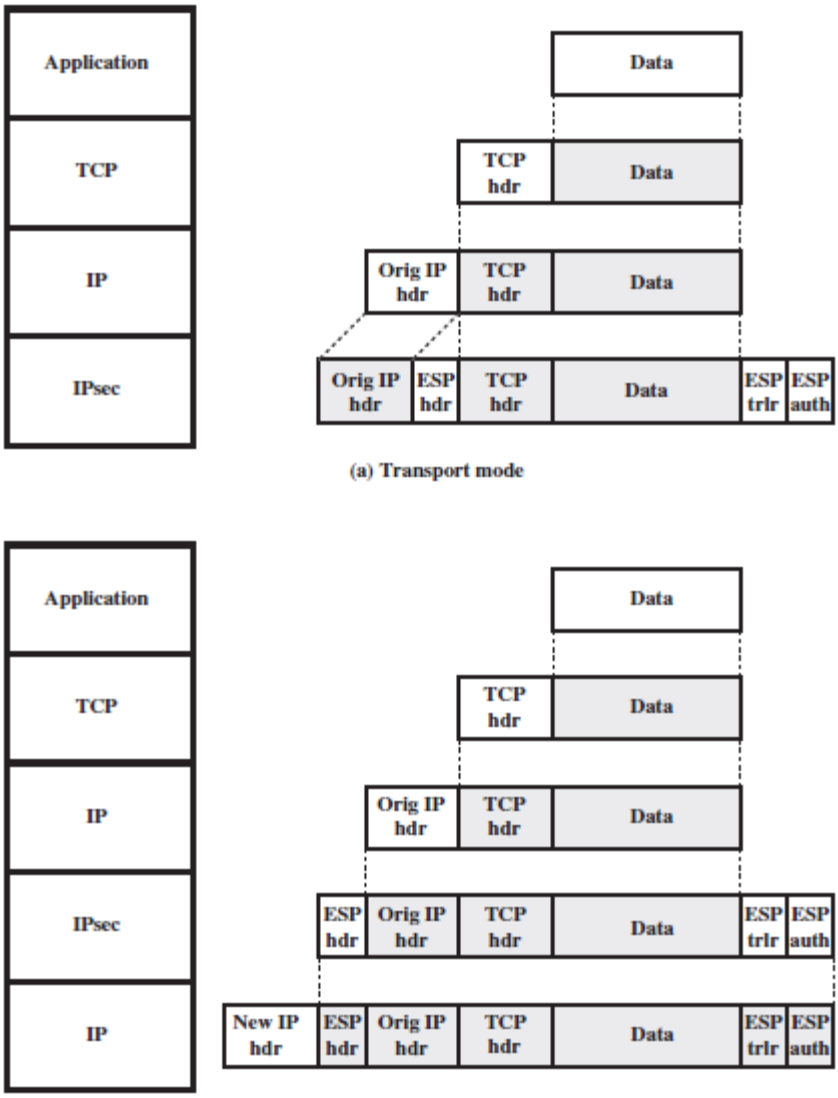


Figure 9 Protocol Operation for ESP

- Consider a case in which an external host wishes to communicate with a host on an internal network protected by a firewall, and in which ESP is implemented in the external host and the firewalls.
- The following steps occur for transfer of a transport-layer segment from the external host to the internal host.

1. The source prepares an inner IP packet with a destination address of the target internal host. This packet is prefixed by an ESP header; then the packet and ESP trailer are encrypted and Authentication Data may be added. The resulting block is encapsulated with a new IP header (base header plus optional extensions such as routing and hop-by-hop options for IPv6) whose destination address is the firewall; this forms the outer IP packet.
2. The outer packet is routed to the destination firewall. Each intermediate router needs to examine and process the outer IP header plus any outer IP extension headers but does not need to examine the cipher text.
3. The destination firewall examines and processes the outer IP header plus any outer IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext inner IP packet. This packet is then transmitted in the internal network.
4. The inner packet is routed through zero or more routers in the internal network to the destination host.

4. COMBINING SECURITY ASSOCIATIONS IMP QS (question)-06M

- An Individual SA **(Security Associations)** can implement either the AH **(Authentication Header)** or ESP (**encapsulating Security Payload)** protocol but not both.
- Sometimes a particular traffic flow will call for the services provided by both AH and ESP. Further, a particular traffic flow may require IPsec services between hosts and, for that same flow, separate services between security gateways, such as firewalls.
- In all of these cases, multiple SAs must be employed for the same traffic flow to achieve the desired IPsec services.
- The SAs in a bundle may terminate at different endpoints or at the same endpoints.

- Security associations may be combined into bundles in two ways:

1. **Transport adjacency:** Refers to applying more than one security protocol to the same IP packet without invoking tunnelling. This approach to combining AH and ESP allows for only one level of combination; further nesting yields no added benefit since the processing is performed at one IPsec instance: the (ultimate) destination.
2. **Iterated tunnelling:** Refers to the application of multiple layers of security protocols affected through IP tunnelling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IPsec site along the path.

4.1 AUTHENTICATION PLUS CONFIDENTIALITY IMP QS (question)-12M

Encryption and authentication can be combined in order to transmit an IP packet that has both confidentiality and authentication between hosts.

4.1.1 ESP wit h Authentication Option

- In this approach, the user first applies ESP to the data to be protected and then appends the authentication data field.
- There are actually two sub cases:
  1. **Transport mode ESP:** Authentication and encryption apply to the IP payload delivered to the host, but the IP header is not protected.
  2. **Tunnel mode ESP:** Authentication applies to the entire IP packet delivered to the outer IP destination address (e.g., a firewall), and authentication is performed at that destination. The entire inner IP packet is protected by the privacy mechanism for delivery to the inner IP destination.
- For both cases, authentication applies to the cipher text rather than the plaintext.

4.1.2 Transport Adjacency

- Another way to apply authentication after encryption is to use two bundled transport SAs, with the inner being an ESP, SA and the outer being an AH, SA.
- In this case, ESP is used without its authentication option. Because the inner SA is a transport SA, encryption is applied to the IP payload.
- The resulting packet consists of an IP header (and possibly IPv6 header extensions) followed by an ESP.

- AH is then applied in transport mode, so that authentication covers the ESP plus the original IP header (and extensions) except for mutable fields.

ADVANTAGE AND DISADVANTAGE

- The advantage of this approach over simply using a single ESP SA with the ESP authentication option is that the authentication covers more fields, including the source and destination IP addresses.
- The disadvantage is the overhead of two SAs versus one SA.

4.1.3 Transport-Tunnel Bundle

- The use of authentication prior to encryption might be preferable for several reasons.
  - First, because the authentication data are protected by encryption, it is impossible for anyone to intercept the message and alter the authentication data without detection.
  - Second, it may be desirable to store the authentication information with the message at the destination for later reference.
- It is more convenient to do this if the authentication information applies to the unencrypted message; otherwise the message would have to be re encrypted to verify the authentication information.

4.2 BASIC COMBINATIONS OF SECURITY ASSOCIATIONS IMP QS (question)-08M

- The IPsec Architecture document lists four examples of combinations of SAs that must be supported by compliant IPsec hosts (e.g., workstation, server) or security gateways (e.g., firewall, router).

THESE ARE ILLUSTRATED IN FIGURE 10.

- The lower part of each case in the figure represents the physical connectivity of the elements; the upper part represents logical connectivity via one or more nested SAs.
- Each SA can be either AH or ESP.

- For host-to-host SAs, the mode may be either transport or tunnel; otherwise it must be tunnel mode.

CASES

- **Case 1**. All security is provided between end systems that implement IPsec.

For any two end systems to communicate via an SA, they must share the appropriate secret keys. Among the possible combinations are

1. AH in transport mode
2. ESP in transport mode
3. ESP followed by AH in transport mode (an ESP SA inside an AH SA)
4. Any one of a, b, or c inside an AH or ESP in tunnel mode

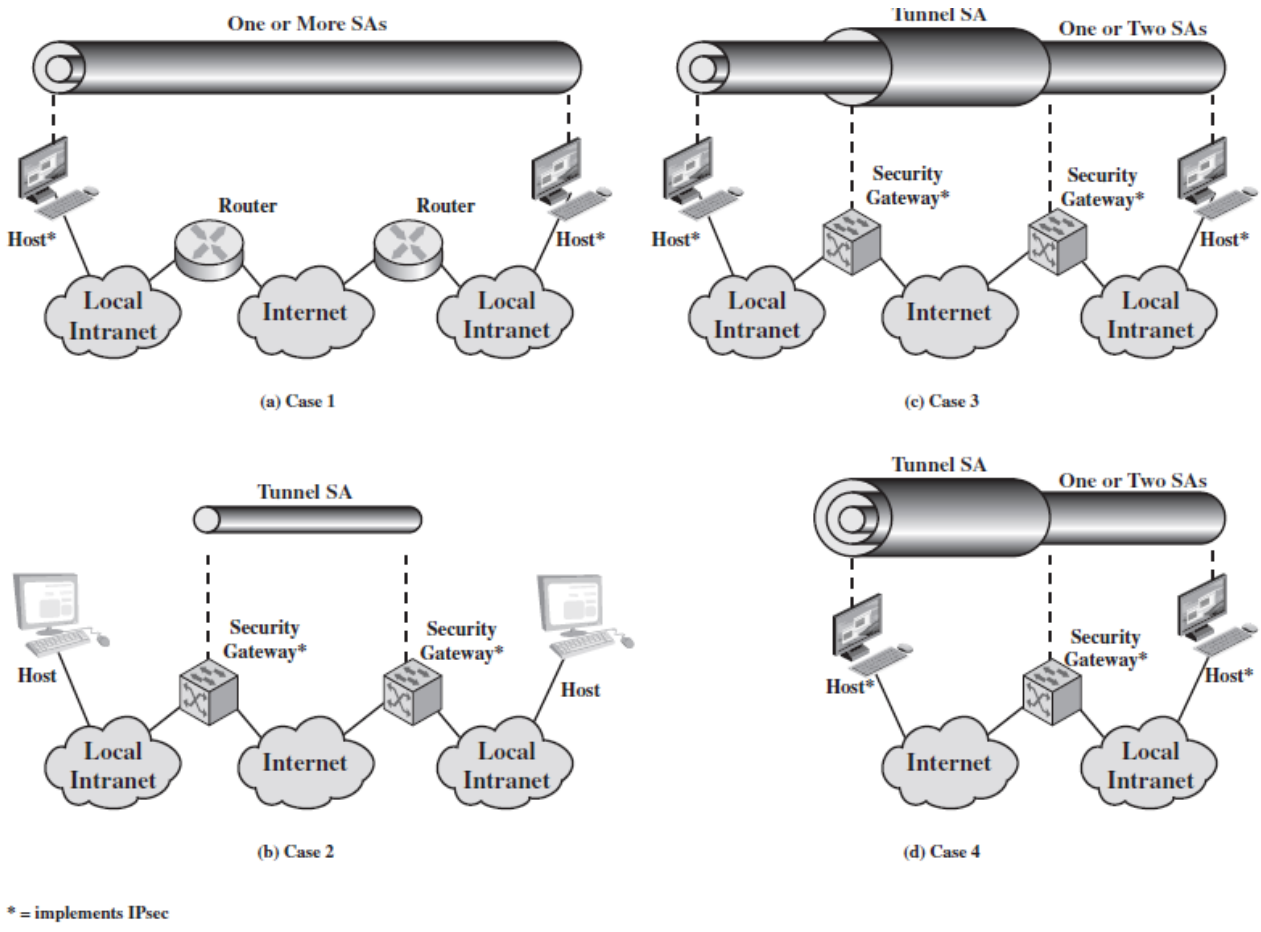


Figure 10 Basic Combinations of Security Associations

- **Case 2.** Security is provided only between gateways (routers, firewalls, etc.) and no hosts implement IPsec. This case illustrates simple virtual private network support. The security architecture document specifies that only a single tunnel SA is needed for this case. The tunnel could support AH, ESP, or ESP with the

authentication option. Nested tunnels are not required, because the IPsec services apply to the entire inner packet.

- **Case 3**. This builds on case 2 by adding end-to-end security. The same combinations discussed for cases 1 and 2 are allowed here. The gateway-to-gateway tunnel provides either authentication, confidentiality, or both for all traffic between end systems. When the gateway-to-gateway tunnel is ESP, it also provides a limited form of traffic confidentiality. Individual hosts can implement any additional IPsec services required for given applications or given users by means of end-to end SAs.
- **Case 4.** This provides support for a remote host that uses the Internet to reach an organization's firewall and then to gain access to some server or workstation behind the firewall. Only tunnel mode is required between the remote host and the firewall. As in case 1, one or two SAs may be used between the remote host and the local host.

5 INTERNET KEY EXCHANGE

IMP QS (question)-05M

- The key management portion of IPsec involves the determination and distribution of secret keys.
- A typical requirement is four keys for communication between two applications: transmit and receive pairs for both integrity and confidentiality.
- The IPsec Architecture document mandates support for two types of key management:
  - **Manual:** A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
  - **Automated:** An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration.
- The default automated key management protocol for IPsec is referred to as ISAKMP/Oakley and consists of the following elements:
  - **Oakley Key Determination Protocol:** Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats.

- **Internet Security Association and Key Management Protocol (ISAKMP):** ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes.

5.1 KEY DETERMINATION PROTOCOL

- IKE key determination is a refinement of the Diffie-Hellman key exchange algorithm.
- The Diffie-Hellman involves the following interaction between users A and B.
- There is prior agreement on two global parameters: q, a large prime number; And α, a primitive root of q.
- A selects a random integer X_A as its private key and transmits to B its public key $Y_A = \alpha^{X_A} \bmod q$.
- Similarly, B selects a random integer X_B as its private key and transmits to A its public key $Y_B = \alpha^{X_B} \bmod q$. Each side can now compute the secret session key:

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

- The Diffie-Hellman algorithm has two attractive features:
  - Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability.
  - The exchange requires no pre-existing infrastructure other than an agreement on the global parameters.
- However, there are a number of weaknesses to Diffie-Hellman, as pointed out in [HUIT98].
  - It does not provide any information about the identities of the parties.
  - It is subject to a man-in-the-middle attack, in which a third party C impersonates B while communicating with A and impersonates A while communicating with B. Both A and B end up negotiating a key with C, which can then listen to and pass on traffic. The man-in-the-middle attack proceeds as

1. B sends his public key Y_B in a message addressed to A (see Figure 11).
2. The enemy (E) intercepts this message. E saves B's public key and sends a message to A that has B's User ID but E's public key Y_E. This message is sent in such a way that it appears as though it was sent from B's host system. A

receives E's message and stores E's public key with B's User ID. Similarly, E sends a message to B with E's public key, purporting to come from A.

3. B computes a secret key $K1$ based on B's private key and Y_E. A computes a secret key $K2$ based on A's private key and Y_E. E computes $K1$ using E's secret key X_E and Y_B and computers $K2$ using X_E And Y_A.

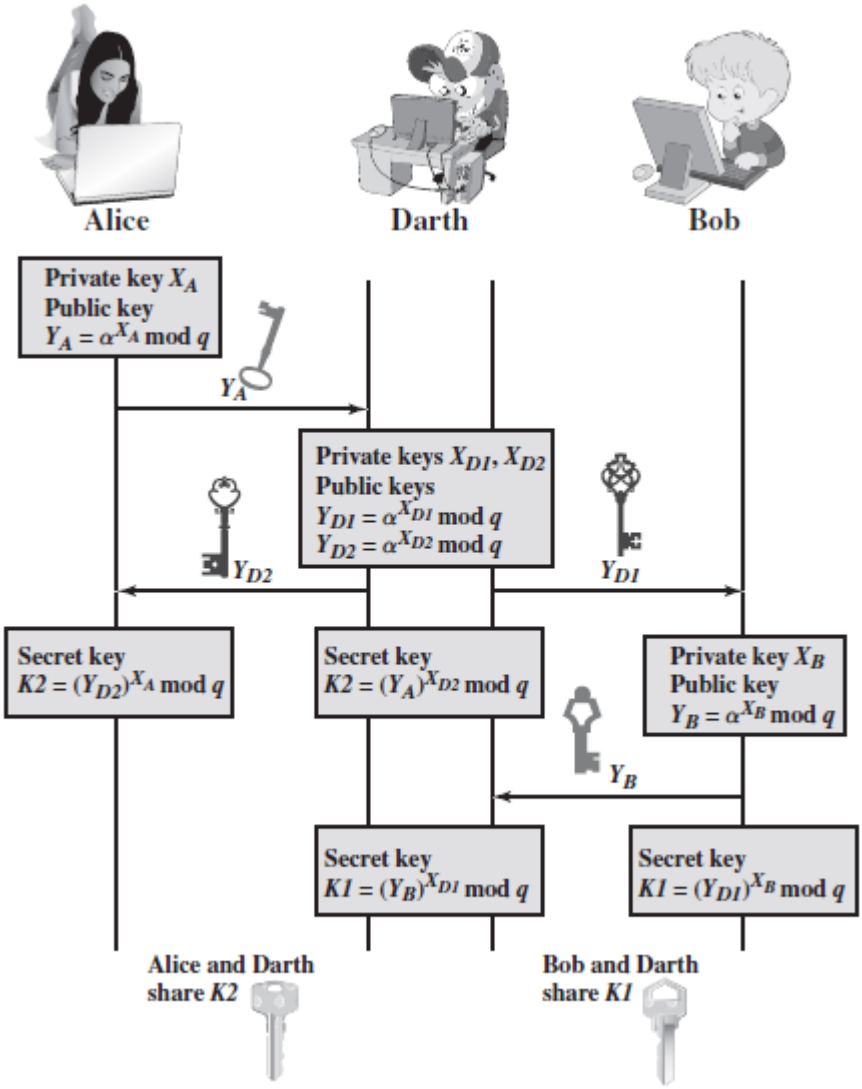


Figure 11 Man-in-the-Middle Attacks

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

- It is computationally intensive. As a result, it is vulnerable to a clogging attack, in which an opponent requests a high number of keys. The victim spends considerable computing resources doing useless modular exponentiation rather than real work.

5.1.1 FEATURES OF IKE KEY DETERMINATION IMP QS (question)-08M

THE IKE KEY DETERMINATION ALGORITHM IS CHARACTERIZED BY FIVE IMPORTANT FEATURES:

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

IKE MANDATES THAT COOKIE GENERATION SATISFY THREE BASIC REQUIREMENTS:

1. The cookie must depend on the specific parties. This prevents an attacker from obtaining a cookie using a real IP address and UDP port and then using it to swamp the victim with requests from randomly chosen IP addresses or ports.
2. It must not be possible for anyone other than the issuing entity to generate cookies that will be accepted by that entity. This implies that the issuing entity will use local secret information in the generation and subsequent verification of a cookie. It must not be possible to deduce this secret information from any particular cookie. The point of this requirement is that the issuing entity need not save copies of its cookies, which are then more vulnerable to discovery, but can verify an incoming cookie acknowledgment when it needs to.
3. The cookie generation and verification methods must be fast to thwart attacks intended to sabotage processor resources.

IKE KEY DETERMINATION SUPPORTS THE USE OF DIFFERENT GROUPS FOR THE DIFFIE–HELLMAN KEY EXCHANGE. EACH GROUP INCLUDES THE DEFINITION OF THE TWO GLOBAL PARAMETERS AND THE IDENTITY OF THE ALGORITHM. THE CURRENT SPECIFICATION INCLUDES THE FOLLOWING GROUPS.

- Modular exponentiation with a 768-bit modulus

$$q = 2^{768} - 2^{704} - 1 + 2^{64} \times (\lfloor 2^{638} \times \pi \rfloor + 149686)$$
$$\alpha = 2$$

- Modular exponentiation with a 1024-bit modulus

$$q = 2^{1024} - 2^{960} - 1 + 2^{64} \times (\lfloor 2^{894} \times \pi \rfloor + 129093)$$
$$\alpha = 2$$

- Modular exponentiation with a 1536-bit modulus
- Parameters to be determined
- Elliptic curve group over 2^{155}
- Generator (hexadecimal): X = 7B, Y = 1C8
- Elliptic curve parameters (hexadecimal): A = 0, Y = 7338F
- Elliptic curve group over 2^{185}
- Generator (hexadecimal): X = 18, Y = D
- Elliptic curve parameters (hexadecimal): A = 0, Y = 1EE9

The first three groups are the classic Diffie-Hellman algorithm using modular exponentiation. The last two groups use the elliptic curve analog to Diffie-Hellman.

THREE DIFFERENT AUTHENTICATION METHODS CAN BE USED WITH IKE KEY DETERMINATION:

- **Digital signatures:** The exchange is authenticated by signing a mutually obtainable hash; each party encrypts the hash with its private key. The hash is generated over important parameters, such as user IDs and nonce's.
- **Public-key encryption:** The exchange is authenticated by encrypting parameters such as IDs and nonce's with the sender's private key.
- **Symmetric-key encryption:** A key derived by some out-of-band mechanism can be used to authenticate the exchange by symmetric encryption of exchange parameters.

5.1.2 IKE V2 EXCHANGES

IMP QS (question)-06M

The IKEv2 protocol involves the exchange of messages in pairs.

THE FIRST TWO PAIRS OF EXCHANGES ARE REFERRED TO AS THE INITIAL EXCHANGES FIGURE 12(A)

In the first exchange, the two peers exchange information concerning cryptographic algorithms and other security parameters they are willing to use along with nonces and Diffie-Hellman (DH) values. The result of this exchange is to set up a special SA called the IKE SA.

In the second exchange, the two parties authenticate one another and set up a first IPsec SA to be placed in the SADB and used for protecting ordinary (i.e. non-IKE) communications between the peers. Thus, four messages are needed to establish the first SA for general use.

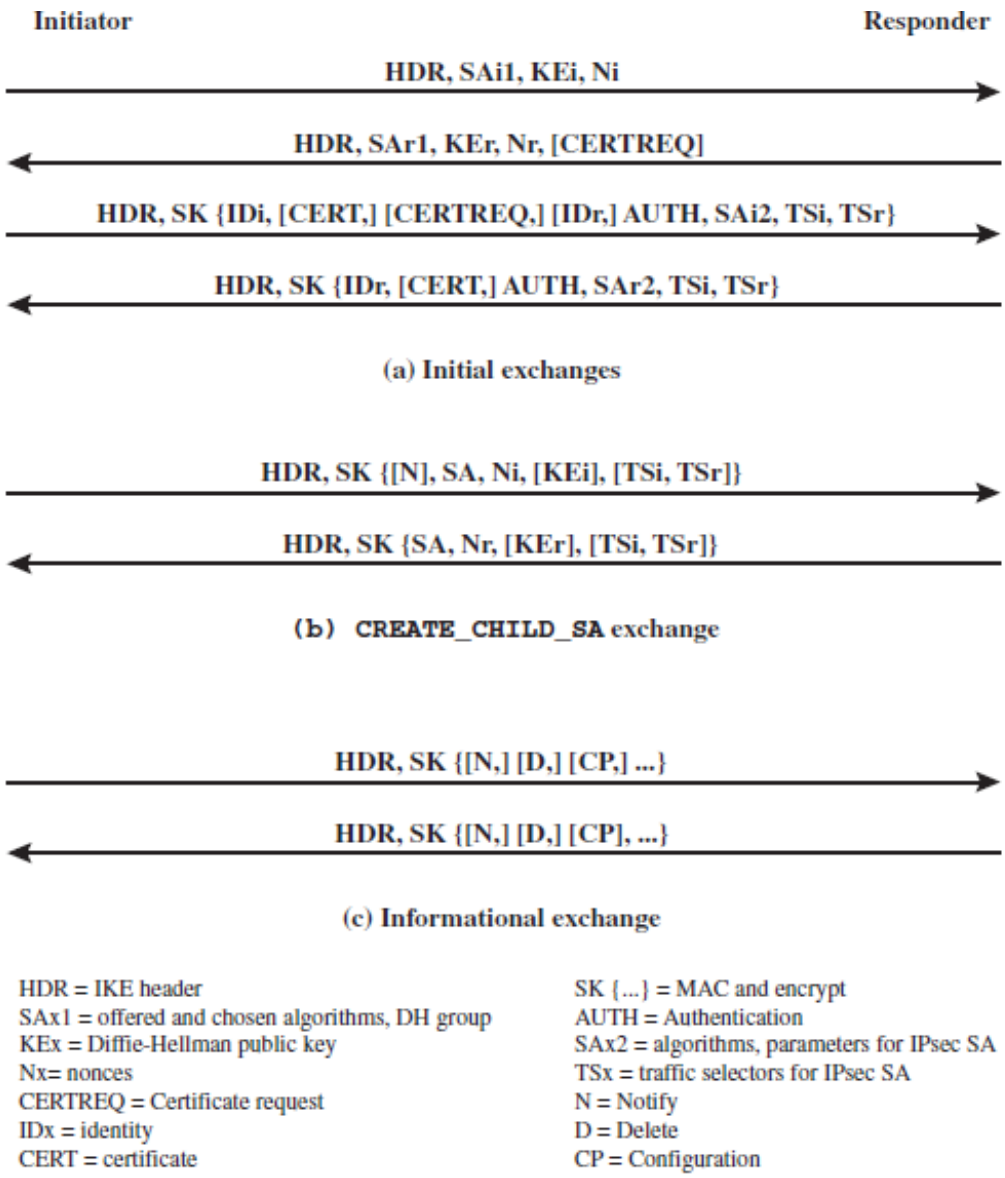


Figure 12 IKEv2 Exchanges

FIGURE 12(B) AND FIGURE 12(C)

The CREATE_CHILD_SA exchange can be used to establish further SAs for protecting traffic. The informational exchange is used to exchange management information, IKEv2 error messages, and other notifications.

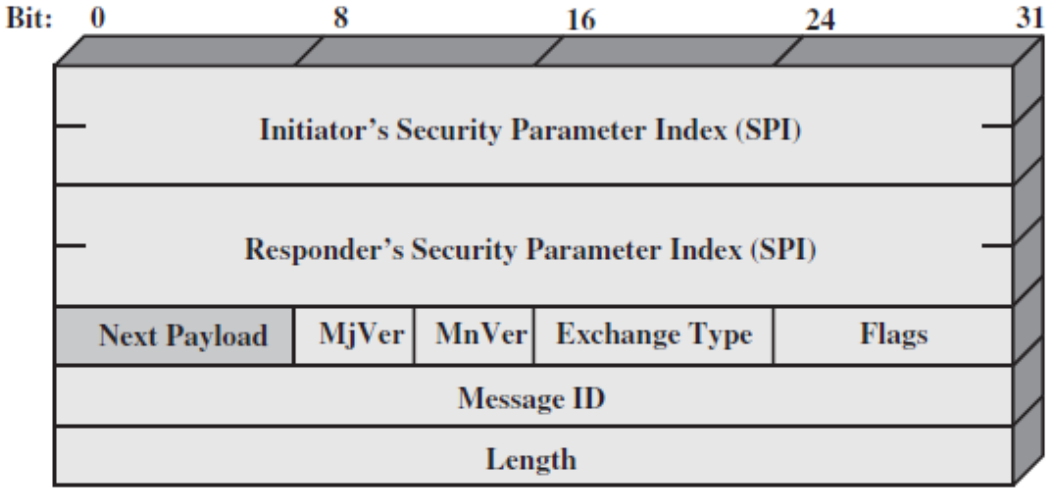
5.2 HEADER AND PAYLOAD FORMATS

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

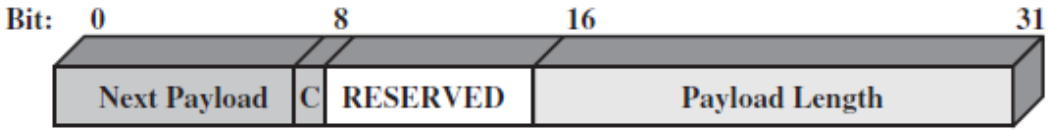
5.2.1 IKE HEADER FORMAT

IMP QS (question)-06M

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



(a) IKE header



(b) Generic Payload header

Figure 13 IKE Formats

Figure 13(A) shows the header format for an IKE message. It consists of the following fields.

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

2. **Responder SPI (64 bits):** A value chosen by the responder to identify a unique IKE SA.
3. **Next Payload (8 bits):** Indicates the type of the first payload in the message; payloads are discussed in the next subsection.
4. **Major Version (4 bits):** Indicates major version of IKE in use.
5. **Minor Version (4 bits):** Indicates minor version in use.
6. **Exchange Type (8 bits):** Indicates the type of exchange; these are discussed later in this section.
7. **Flags (8 bits):** Indicates specific options set for this IKE exchange. Three bits are defined so far. The initiator bit indicates whether this packet is sent by the SA initiator. The version bit indicates whether the transmitter is capable of using a higher major version number than the one currently indicated. The response bit indicates whether this is a response to a message containing the same message ID.
8. **Message ID (32 bits):** Used to control retransmission of lost packets and matching of requests and responses.
9. **Length (32 bits):** Length of total message (header plus all payloads) in octets.

5.2.2 IKE PAYLOAD TYPES

IMP QS (question)-06M

- All IKE payloads begin with the same generic payload header shown in Figure **Figure 13(B).**
- The Next Payload field has a value of 0 if this is the last payload in the message; otherwise its value is the type of the next payload.
- The Payload Length field indicates the length in octets of this payload, including the generic payload header.
- These elements are formatted as substructures within the payload as follows.
  1. **Proposal:** This substructure includes a proposal number, a protocol ID (AH, ESP, or IKE), an indicator of the number of transforms, and then a transform substructure. If more than one protocol is to be included in a proposal, then there is a subsequent proposal substructure with the same proposal number.
  2. **Transform:** Different protocols support different transform types. The transforms are used primarily to define cryptographic algorithms to be used with a particular protocol.

3. **Attribute:** Each transform may include attributes that modify or complete the specification of the transform. An example is key length.

IKE PAYLOAD TYPES

- The **Key Exchange payload** can be used for a variety of key exchange techniques, including Oakley, Diffie-Hellman, and the RSA-based key exchange used by PGP.
- The **Identification payload** is used to determine the identity of communicating peers and may be used for determining authenticity of information. Typically the ID Data field will contain an IPv4 or IPv6 address.
- The **Certificate payload** transfers a public-key certificate.

Type	Parameters
Security Association	Proposals
Key Exchange	DH Group #, Key Exchange Data
Identification	ID Type, ID Data
Certificate	Cert Encoding, Certificate Data
Certificate Request	Cert Encoding, Certification Authority
Authentication	Auth Method, Authentication Data
Nonce	Nonce Data
Notify	Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data
Delete	Protocol-ID, SPI Size, # of SPIs, SPI (one or more)
Vendor ID	Vendor ID
Traffic Selector	Number of TSs, Traffic Selectors
Encrypted	IV, Encrypted IKE payloads, Padding, Pad Length, ICV
Configuration	CFG Type, Configuration Attributes
Extensible Authentication Protocol	EAP Message

Table 3 IKE Payload Types

- At any point in an IKE exchange, the sender may include a **Certificate Request** payload to request the certificate of the other communicating entity.
- The **Authentication** payload contains data used for message authentication purposes.
- The **Nonce** payload contains random data used to guarantee liveness during an exchange and to protect against replay attacks.
- The **Notify** payload contains either error or status information associated with this SA or this SA negotiation.

- The **Delete** payload indicates one or more SAs that the sender has deleted from its database and that therefore are no longer valid.
- The **Vendor ID** payload contains a vendor-defined constant.
- The **Traffic Selector** payload allows peers to identify packet flows for processing by IPsec services.
- The **Encrypted** payload contains other payloads in encrypted form.
- The **Configuration** payload is used to exchange configuration information between IKE peers.
- The **Extensible Authentication Protocol (EAP)** payload allows IKE SAs to be authenticated using EAP,

6 CRYPTOGRAPHIC SUITES

IMP QS (question)-08 or 10M

RFC 4308 defines two cryptographic suites for establishing virtual private networks.

Suite VPN-A matches the commonly used corporate VPN security used in older IKEv1 implementations at the time of the issuance of IKEv2 in 2005.

Suite VPN-B provides stronger security and is recommended for new VPNs that implement IPsecv3 and IKEv2.

Table 4 (a) lists the algorithms and parameters for the two suites. There are several points to note about these two suites. Note that for symmetric cryptography, VPN-A relies on 3DES and HMAC, while VPN-B relies exclusively on AES.

THREE TYPES OF SECRET–KEY ALGORITHMS ARE USED:

1. **Encryption:** For encryption, the cipher block chaining (CBC) mode is used.
2. **Message authentication:** For message authentication, VPN-A relies on HMAC with SHA-1 with the output truncated to 96 bits. VPN-B relies on a variant of CMAC with the output truncated to 96 bits.
3. **Pseudorandom function:** IKEv2 generates pseudorandom bits by repeated use of the MAC used for message authentication.

RFC 6379 defines four optional cryptographic suites that are compatible with the United States National Security Agency's Suite B specifications.

The four suites defined in RFC 4869 provide choices for ESP and IKE.

The four suites are differentiated by the choice of cryptographic algorithm strengths and a choice of whether ESP is to provide both confidentiality and integrity or integrity only.

All of the suites offer greater protection than the two VPN suites defined in RFC 4308.

	VPN-A	VPN-B
ESP encryption	3DES-CBC	AES-CBC (128-bit key)
ESP integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE encryption	3DES-CBC	AES-CBC (128-bit key)
IKE PRF	HMAC-SHA1	AES-XCBC-PRF-128
IKE Integrity	HMAC-SHA1-96	AES-XCBC-MAC-96
IKE DH group	1024-bit MODP	2048-bit MODP

(a) Virtual private networks (RFC 4308)

	GCM-128	GCM-256	GMAC-128	GMAC-256
ESP encryption/ Integrity	AES-GCM (128-bit key)	AES-GCM (256-bit key)	Null	Null
ESP integrity	Null	Null	AES-GMAC (128-bit key)	AES-GMAC (256-bit key)
IKE encryption	AES-CBC (128-bit key)	AES-CBC (256-bit key)	AES-CBC (128-bit key)	AES-CBC (256-bit key)
IKE PRF	HMAC-SHA-256	HMAC-SHA-384	HMAC-SHA-256	HMAC-SHA-384
IKE Integrity	HMAC-SHA-256-128	HMAC-SHA-384-192	HMAC-SHA-256-128	HMAC-SHA-384-192
IKE DH group	256-bit random ECP	384-bit random ECP	256-bit random ECP	384-bit random ECP

(b) NSA Suite B (RFC 4869)

Table 4 Cryptographic Suites for IPsec

Table 4(b) lists the algorithms and parameters for the two suites. As with RFC 4308, three categories of secret key algorithms are listed:

- **Encryption:** For ESP, authenticated encryption is provided using the GCM mode with either 128-bit or 256-bit AES keys. For IKE encryption, CBC is used, as it was for the VPN suites.
- **Message authentication:** For ESP, if only authentication is required, then GMAC is used. As discussed in Chapter 12, GMAC is simply the authentication portion of GCM. For IKE, message authentication is provided using HMAC with one of the SHA-3 hash functions.

- **Pseudorandom function:** As with the VPN suites, IKEv2 in these suites generates pseudorandom bits by repeated use of the MAC used for message authentication.

QUESTION BANK – NETWORK AND CYBER SECURITY

MODULE-3

1.	Discuss IP security overview.	**06M**
2.	Discuss applications of IPsec.	**06M**
3.	With neat diagrams, explain ip security scenario.	**08M**
4.	Discuss benefits of IPsec.	**04M**
5.	Discuss IPsec documents.	**05M**
6.	Discuss transport and tunnel modes.	**09M**
7.	Discuss ip security policy.	**04M**
8.	Discuss security associations.	**03M**
9.	Discuss security association database.	**05M**
10.	Discuss security policy database.	**05M**
11.	With a neat diagram, explain ip traffic processing.	**10M**
12.	Discuss encapsulating security payload.	**03M**
13.	With a neat diagram, explain ESP format.	**08M**
14.	With a neat diagram, explain anti – reply service.	**06M**
15.	With a neat diagram, explain transport and tunnel modes.	**08M**
16.	With a neat diagram, explain transport mode ESP.	**08M**
17.	With a neat diagram, explain tunnel mode ESP.	**06M**
18.	Discuss authentication plus confidentiality.	**10M**
19.	With a neat diagram, explain basic combinations of security associations.	**10M**
20.	Discuss internet key exchange.	**05M**
21.	Discuss features of IKE key determination.	**08M**
22.	With a neat diagram, explain IKE v2 exchanges.	**10M**
23.	With a neat diagram, explain IKE header format.	**08M**
24.	Discuss IKE payload types.	**06M**
25.	Discuss cryptographic suites.	**10M**

NETWORK AND CYBER SECURITY
(15EC835, 17EC835)

8TH SEM E&C

JAYANTH DWIJESH H P **BE (ECE), M.tech (DECS).**

Assistant Professor – Dept of E&CE, BGSIT.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

B.G.S INSTITUTE OF TECHNOLOGY (B.G.S.I.T)

B.G Nagara, Nagamangala Tq, Mandya District- 571448

NETWORK AND CYBER SECURITY

MODULE–4

MODULE–4

Cyber network security concepts: Security Architecture, Antipattern: signature based malware detection versus polymorphic threads, document driven certification and accreditation, policy driven security certifications. Refactored solution: reputational, behavioural and entropy based malware detection.

The problems: cyber antipatterns concept, forces in cyber antipatterns, cyber anti pattern templates, cyber security Antipattern catalog.

TEXT BOOK:

1. **William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325-1877-3.**
2. **Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.**

REFERENCE BOOKS:

1. Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.

2. Cryptography and Network Security, Atul Kahate, TMH, 2003.

MODULE–4

Cyber network security concepts: Security Architecture, Antipattern: signature based malware detection versus polymorphic threads, document driven certification and accreditation, policy driven security certifications. Refactored solution: reputational, behavioural and entropy based malware detection.

The problems: cyber antipatterns concept, forces in cyber antipatterns, cyber anti pattern templates, cyber security Antipattern catalog.

CHAPTER -1 CYBER NETWORK SECURITY CONCEPTS

1. SECURITY ARCHITECTURE

IMP QS (question)-04M

- The cyber security crisis is a fundamental failure of architecture. Many of the Networked technologies we depend upon daily have no effective security whatsoever.
- The architecture of the Internet and the vast majority of deployed software create significant opportunities for malicious exploitation.
- It is worth stating that if infrastructure and software technologies were engineered properly, they would be built to withstand known and manage unknown risks, and they would be significantly more secure than current-day technologies.
- the Zachman Framework for Enterprise Architecture and applies it to securing enterprises.
- The Zachman Framework is a powerful intellectual tool that enables complex organizations to describe themselves, including their mission, business, and information technology (IT) assets. With this self-knowledge comes awareness of risks and mitigations, and ways of engineering security into solutions from inception.
- The Zachman Framework serves as an overarching structure that organizes the problem-solving patterns catalog.

2. ANTIPATTERN: SIGNATURE-BASED MALWARE DETECTION VERSUS POLYMORPHIC THREATS

IMP QS (question)-06M

- The conventional wisdom is that all systems with up-to-date antivirus signatures will be safe.
- However, many popular antivirus solutions are nearly obsolete, with many missing the majority of new malware.
- Current signature-based antivirus engines miss 30 percent to 70 percent of malicious code, and nearly 100 percent of zero day infections, which, by definition, are unreported exploits.
- Malicious signature growth is exploding from 5 new ones per day in 2000 to 1,500 per day in 2007 and more than 15,000 per day in 2009, according to Symantec (from a 2010 conference briefing on reputational anti-malware), which is an average of 200 percent to 300 percent cumulative growth per year.
- Malware variability has grown so rapidly that signature-based detection is rapidly becoming obsolete.
- The proliferation of malware signatures is exploding primarily due to polymorphic malware techniques.
    - For example, hash functions used by signature-based detectors yield very different values with only slight changes to a malicious file. Changing a string literal in the file is sufficient to trigger a false negative.
    - Other polymorphic techniques include varying character encodings, encryption, and random values in the files.
- One interesting online application from VirusTotal.com runs more than 30 antivirus programs on each file that any Internet user can submit. You can witness just how haphazard antivirus tests are.

3. ANTIPATTERN: DOCUMENT-DRIVEN CERTIFICATION AND ACCREDITATION

IMP QS (question)-06M

- Some of the most flagrant antipatterns involve the IT security industry itself.
- Assessment and Authorization (A&A), formerly called Certification and Accreditation (C&A), has attracted much public criticism because it has a

reputation as a paper-driven process that does not secure systems from real threats.

- A&A is the process of assuring the information security of systems before they are deployed.
- Certification is an assessment and testing phase that identifies and confirms vulnerabilities.
- Accreditation is an executive approval process that accepts risks discovered during certification.
- Precertification is often an arduous process of security documentation and reviews. In many organizations, certification is problematic. Often testing is waivered or done very superficially with policy scanners that check registry and configuration settings.
- In the more rigorous practice of penetration testing (pen testing), vulnerabilities are thoroughly explored with state of the art tools, followed by actual exploitation and malicious user tests where unauthorized accesses are the goal.
- Although A&A is formalized in government organizations, it is also widely practiced in industry. For example, payment card industry (PCI) standards require businesses that process credit cards (in other words, virtually all retail companies), to conduct penetration tests and other formal assessments.
- Refactored solutions for this Antipattern can be derived from the practical security testing and investigation techniques.

4. ANTIPATTERN: POLICY-DRIVEN SECURITY CERTIFICATIONS DO NOT ADDRESS THE THREAT.

IMP QS (question)-06M

- The gold standard of professional security certifications is the Certified Information System Security Professional (CISSP). It is an entirely paper-based qualification, requiring a great deal of memorization in 10 diverse security domains, such as physical security, communications security, and systems security.
- CISSP is required by the U.S. Department of Defense (DoD) for both management and technical security workers, and demanded in the job market.

- Anecdotally, the presumed goal of this certification is to produce articulate security professionals who can communicate effectively with upper management, but what does that have to do with combating emerging cyber threats? This paradox was addressed by the Center for Strategic and International Studies (CSIS), which released a Presidential Commission report: A Human Capital Crisis in Cyber security (July, 2010).
- The report states clearly that "the current professional certification regime is not merely inadequate; it creates a dangerously false sense of security" with an overemphasis on security compliance on paper versus combating threats.
- Many people in the cyber security community view this finding as controversial because their careers, reputations, and credentials are invested in security compliance policies and procedures. This is the industry that drives A&A, risk management, security controls compliance, and other labor-intensive security activities.
- Unfortunately, for most professionals, it is much easier to turn a highly technical person into a policy person, whereas it is very difficult (or impossible) to turn a policy person into a highly technical one. It is a one-way street.

5. REFACTORED SOLUTION: REPUTATIONAL, BEHAVIOURAL AND ENTROPY BASED MALWARE DETECTION.

IMP QS (question)-06M

Vendors are developing innovative techniques that can detect zero day and polymorphic malware. Several promising approaches for the future include:

- Symantec is harnessing a 100M+ global customer base to identify potential malware signatures. The technique, called reputation-based signatures, is able to identify 240 million new malware signatures by comparing binaries across millions of systems for anomalous variations.
- Fire Eye has created a behavioral intrusion detection system (IDS) that uses elements of honey pots and forensics to automatically identify malicious content as it flows across corporate networks. Behavioral IDS techniques simulate the execution of sniffed content in a virtual machine, which then observes resulting configuration changes, such as changes in registry settings, services, and the file

system. There are other emerging behavioural antivirus products, for example, from ThreatFire.com.

- An emerging field of research called entropy-based malware detection looks for mathematical similarity to known malware signatures. Hash functions that are used by most antivirus programs detect subtle differences between a file and its known hash. Minor changes to a file, such as modification of strings or encodings can cause a hash match to fail. Entropy-based matching uses mathematical functions that measure similarity rather than differences. If a suspicious file nearly matches the same entropy measure as malware, there is a high likelihood that the malware is present.

CHAPTER-2 THE PROBLEMS

1. ANTIPATTERNS CONCEPT

IMP QS (question)-06M

Design forces are the competing concerns, priorities, and technical factors that influence the choice of solutions. In antipatterns, there are two solutions: the Antipattern solution and the Refactored solution.

1. An **Antipattern solution** represents a commonplace dysfunctional situation or configuration. The Antipattern solution may be the result of multiple choices over an extended system lifecycle, or it may have evolved inadvertently. Every solution or design choice yields benefits and consequences.

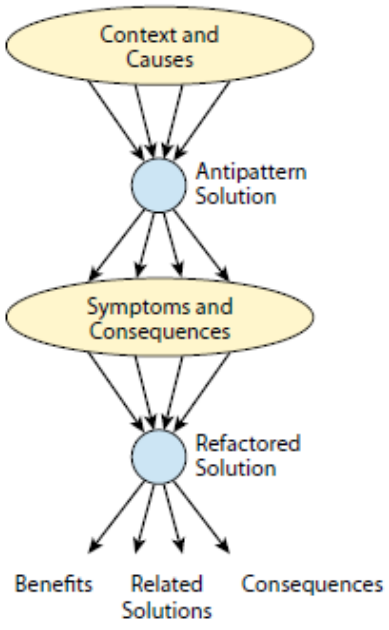


Figure 1: Antipattern concept

2. The **Refactored solution** results from reconsideration of the design forces and the selection of a more effective solution (see Figure 1). The Refactored solution yields benefits that outweigh its consequences. There may also be related solutions or variations that also resolve the design forces beneficially.

2. FORCES IN CYBER ANTIPATTERNS

IMP QS (question)-08M

The major types of forces in antipatterns include primal, horizontal, and vertical forces.

1. Primal forces are pervasive design forces present in almost every design decision.
2. Horizontal forces are forces that can apply in all domains.
3. Vertical forces are domain or system specific design forces.

The primal design forces in the cyber security domain include:

- Management of functionality
- Management of confidentiality
- Management of integrity
- Management of availability

You probably recognize this formulation as the famous **Confidentiality, Integrity, and Availability (CIA)** from IT security.

The **functionality** design force is added because it drives the other forces. Systems are granted accreditation with respect to a defined level of **functionality**. **Functionality** is tested and verified by the developers prior to security testing.

- **Confidentiality**
  - Is the protection of information on the system.
  - In most current systems, the information is the primary resource being secured and the sensitivity of the information defines the level of risk and security priority for each system or database element.
- **Integrity**
  - Is protection of the coherence of the data and system metadata (for example, configuration).
  - The significant threat of damage to data can be very costly to remediate.
  - This threat affects even the most sensitive systems that have very limited connectivity to external networks because data, e-mail, and removable

media with malware can migrate to those systems through normal and erroneous operations.

- **Availability**
  - is the continuous readiness of the system to execute its functionality in response to users and other systems' requests, and the ability to continually access its data.
  - Availability is an aspect of the more general concept of Quality of Service (QOS).
  - QOS is a service-level requirement for the system, such as guarantees of throughput bandwidth or user request response time.

To assure the security of a system, testing is a necessary evil. A test is a comparison of two things, such as a security specification and a system implementation. As a result, requirements for functionality, confidentiality, integrity, and availability should be clearly identified in the system documentation.

3. CYBER ANTIPATTERN TEMPLATES

IMP QS (question)-10M

The **two templates** include the **micro-Antipattern template and the full cyber Antipattern template**. You use **the micro-template** for simpler patterns in which detailed explanation is not necessary.

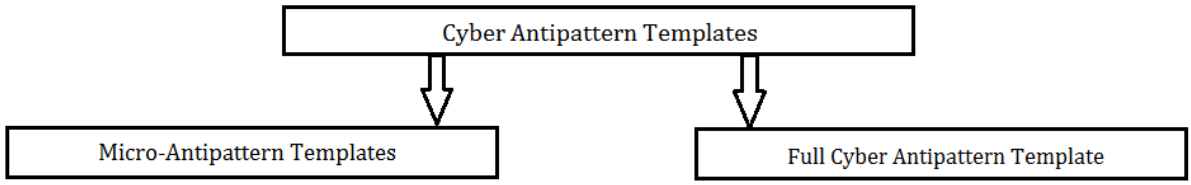


Fig2: Cyber Antipattern templates types

You use the **full template** for the more complex and more important antipatterns, providing a much more complete coverage of the problem and the solution.

3.1 MICRO-ANTIPATTERN TEMPLATES

IMP QS (question)-04M

The micro-Antipattern It is a flexible and informal way to present antipatterns. The components of a micro-Antipattern template are:

1. **Name:** The name of the micro-Antipattern is usually a pejorative term, suggesting the negative consequences of the antipatterns presence.
2. **Antipattern Problem:** The problem section summarizes the micro-antipatterns symptoms, consequences, and characterization.
3. **Refactored Solution:** The solution section summarizes alternative ways To resolve the Antipattern design forces with improved benefits.

Because the micro-Antipattern template is so simple, it can be presented without the formality of templates at all.

3.2 FULL CYBER ANTIPATTERN TEMPLATE IMP QS (question)-08M

- The full cyber Antipattern template has two main parts: a header and a body.
- The header gives a quick sense of the Antipattern and the solution, inviting the reader to dive deeper.
- The body sections contain the pattern details.
- The full cyber Antipattern template It allows for a more structured and comprehensive definition with additional Antipattern attributes.
- Many of the attributes are considered optional, depending on the particulars of the Antipattern concerned.
- The heading fields in the full cyber Antipattern template are
  - **Antipattern Name:** The name is a unique pejorative noun phrase. The intent is to make this Antipattern a well-known phenomenon, easily recognizable, with an organizational reputation as an important security gap.
  - **Also Known As:** Many antipatterns are known by various names across different organizations. Some known names or analogous names from different domains are listed here. A given organization might want to adopt a name from this list if their members find it suitable.
  - **Refactored Solution Names:** One or more names of alternative solutions are listed here. The purpose is to give the reader a sense of the direction that this pattern write-up is heading toward and to promote a common terminology for the solution identity associated with the Antipattern.
  - **Unbalanced Primal Forces:** This field lists the primal design forces that are poorly resolved by this Antipattern.

- **Anecdotal Evidence:** These are some quips that characterize this Antipattern. These phrases are sometimes heard when the Antipattern is present and in the early recognition of it.

➢ The body fields in the full cyber Antipattern template are

- **Background:** This optional field provides contextual explanations that are potentially useful or of general interest but are not central to the Antipattern and its Refactored solution.
- **Antipattern Solution:** This field defines the Antipattern solution through diagrams, explanations, examples, and discussions of design forces. The Antipattern solution is a commonly occurring situation or configuration with significant security implications, such as risks, threats, and vulnerabilities.
- **Causes, Symptoms, and Consequences:** This bulleted section lists the typical causes, common symptoms, and resulting consequences of the Antipattern solution. The intent is to make it easier to recognize the Antipattern and understand how and why its replacement is necessary.
- **Known Exceptions:** If there are some situations where the Antipattern solution might be desirable, this section identifies them. For example, if the consequences are acceptable in a context or if replacement is not worthwhile.
- **Refactored Solution and Examples:** This field defines the Refactored solution. The Refactored solution is proposed as an alternative to the Antipattern solution. Refactoring is a process of replacing or reworking a given solution into an alternative solution. The new solution resolves the design forces differently, particularly providing a more effective solution that resolves design forces more satisfactorily.
- **Related Solutions:** If there are other potential solutions to the Antipattern, they are identified in this section. Often there are different approaches to resolving the same problem that don't conveniently fall under the umbrella of the chosen Refactored solution.

4. CYBER SECURITY ANTIPATTERN CATALOG

IMP QS (question)-06M

➢ The concept of antipatterns as a way to motivate organizational and behavioral changes.

- the following general antipatterns informally, along with potential solutions:
  - Signature-Based Malware Detection Versus Polymorphic Threats
  - Document-Driven Certification and Accreditation
  - Proliferating IA Standards with No Proven Benefits
  - Policy-Driven Security Certifications Do Not Address the Threat
- Cyber mistakes and bad security habits with these prevalent antipatterns:
  - Can't Patch Dumb
  - Unpatched Applications
  - Never Read the Logs
  - Networks Always Play by the Rules
  - Hard on the Outside, Gooey in the Middle
  - Webify Everything
  - No Time for Security
- The antipatterns are intended to be light reading to raise awareness of major security gaps created by how current practitioners develop and manage systems and networks.

4.1 CAN'T PATCH DUMB

IMP QS (question)-07M

Antipattern Name: Can't Patch Dumb

Also Known As: Social Engineering, Phishing, Spam, Spyware, Drive-by Malware, Ransom-Ware, Auto play Attacks

Refactored Solution Names: Security Awareness

Unbalanced Primal Forces: Confidentiality (for example, divulging private information), integrity (for example, rootkits)

Anecdotal Evidence: "Technology is not the problem; people are the problem," and "Technology is easy; people are difficult."

Antipattern Solution

- The end user's lack of security awareness puts his personal information and the organization's competitiveness at risk. Social engineering—the art of extracting sensitive information from people—exploits a human's inherent tendency to want to help other people.

- Unaware end users are easily fooled into opening malicious e-mail attachments, responding to spam offers, and downloading spyware, ransom-ware, and malicious websites.
- The spyware problem is much more widespread than people realize because it involves not just spyware applications, but also spyware that is running in browsers that is served up knowingly by Top 100 websites such as ESPN.com and Disney.com.
- There are thousands of web-tracking companies making money spying on your web activities and vacuuming information from all your browser tabs.
- About 9,000 malicious websites offer free antivirus solutions. When you install these applications, what you get instead is a threat and a warning to pay the vendor or your computer becomes unusable.
- Drive-by malware are computer infections that are loaded automatically when you visit a malicious website. Some of these sites are from legitimate businesses that have been hacked and exploited to download malware.
- Legitimate websites can also spread malware from their advertisements, which are controlled by third parties.
- Auto play infections result when malware is introduced by a user from a Universal Serial Bus (USB) memory stick, often in violation of organizational policies. By default, Microsoft Windows auto plays programs on memory sticks whenever one is inserted.
- Memory sticks with auto play infections are distributed accidentally by legitimate companies at trade shows. They have propagated viruses from educational networks to home networks, and were used to spread malware attacks such as Stuxnet.
- The Antipattern occurs when organizations do not take adequate precautions to keep their end users from inadvertently compromising their systems or divulging information to strangers.

Causes, Symptoms, and Consequences

Causes and symptoms of this Antipattern are a lack of a recurring security awareness training program for all end users, including a test assessment.

Refactored Solution and Examples

- Security awareness training should be mandatory for every person in an organization. Training should be completed before a person is given computer access, and then the organization should conduct annual refresher courses.
- The courses should include training on social engineering skills as well as Internet safety.
- The training should articulate the organization's policies on what information can be divulged to which groups of customers or co-workers.
- An online training program with integrated testing is ideal and ensures that the required skills are acquired. Test answers can be used for accountability, proving that particular policies were known by specific individuals.

Related Solutions

End users should have website advisors installed, perhaps as part of the antivirus suite. Users should take even further precautions, such as using Google to reach websites.

Google constantly scans the Internet for malware. Before a user clicks through to a suspected malicious site, Google gives a warning message on the search page, and presents a challenge page to further persuade the user to avoid the website.

Solutions such as the Firefox extension No Script are too all-or-nothing to effectively dissuade users from unsafe surfing behaviors. No Script stops web scripts by default, requiring user permission to enable them on each webpage.

After some experience, many users will enable scripts everywhere, or at least on many Top 100 websites where the biggest spyware threats reside.

4.2 UNPATCHED APPLICATIONS

IMP QS (question)-07M

Antipattern Name: Unpatched Applications

Also Known As: Vendor-Specific Updates, Default Configuration

Refactored Solution Names: Patch Management

Unbalanced Primal Forces: Management of integrity

Anecdotal Evidence: "Most new attacks are going after the applications, not the operating systems."

Antipattern Solution

According to SANS Institute's 2010 list of top security vulnerabilities, Unpatched applications are one of the biggest security risks. Add-on applications such as QuickTime for Windows, Acrobat, Chrome, and many others are frequent sources of security warnings from the United States Computer Emergency Readiness Team (US-CERT). Vendors try to release patches for the problems at the same time that the defects are announced.

The lag between the patch release and the installed update creates a vulnerability window for attackers. Announcement of the vulnerability and the binary patch gives attackers clues about how to exploit the weakness. Eventually, security researchers may even release the exploit publically.

SANS found that enterprises are very effective at keeping operating system patches current, but they are ineffective at keeping application patches up to date.

Causes, Symptoms, and Consequences

The causes, symptoms, and consequences of this antipattern include

- Automatic update disabled on any application where it's available
- Never visit vendor websites to search for updates
- No inventory of applications and vendors
- No update maintenance schedule
- Not reviewing the US-CERT bulletins
- No governance of application versions

Known Exceptions

If software product support has expired (as it has for early versions of Windows) and there are no further vendor updates, migration to a supported version is strongly recommended.

Each organization should maintain a list of approved standard versions of all software applications. Some vendors will continue to support the product with security patches for an additional fee.

Refactored Solution and Examples

A first step toward managing your patches is obtaining an inventory of systems and installed software packages. On small networks, you can enable automatic updates on Windows and applications such as Acrobat and Firefox. For other applications, such as video drivers, you might have to update from the vendor's website.

Keep an eye on the US-CERT bulletins. If there are serious vulnerabilities announced for your applications, follow the Patch Available links and install the patches. In some environments, sophisticated users can maintain their own systems in this manner.

For larger networks, patch management tools can maintain hundreds or thousands of machines with minimal effort. Some best-in-class vendors of these technologies include LANDesk, BMC, Altiris, and HP.

For even greater assurance, many shops are adopting vulnerability scanning tools, such as Retina from eEye, Nessus from Tenable, and NeXpose from Rapid7. Tools are often used for security certification testing prior to system release.

Some can be configured for automatic scans, such as quarterly or daily. Tools can check for patches, policy configuration issues, and network vulnerabilities.

Related Solutions

Some technically advanced organizations are using their data center provisioning environments to assure patch management and policy configurations.

By creating locked-down standard system images, data centers are able to deploy virtual servers which conform to security baselines, and perform mass updates to these configurations to apply patches and other changes.

4.3 NEVER READ THE LOGS

IMP QS (question)-07M

Antipattern Name: Never Read the Logs

Also Known As: Guys Watching Big Network Displays Miss Everything, Insider Threat, Advanced Persistent Threat (APT), Network Operations Center (NOC)

Refactored Solution Names: Advanced Log Analysis

Unbalanced Primal Forces: Management of confidentiality

Anecdotal Evidence: Nick Leeson at Barings Bank, Wikileaks, Aurora Cyber Intrusions

Antipattern Solution

Network operating centers (NOC) are facilities with large colorful displays of system and network status. System, network, and security devices send messages about events (audit logs) to centralized management applications, which test for alarm conditions and generate the big displays.

The alerting rules are usually set to eliminate false positive alarms. For example, Intrusion Detection System (IDS) rules and Intrusion Prevention Systems (IPS) that cause false alarms are disabled. Frequently logged events, such as configuration changes on end-user systems are not alarmed.

All this is fine and good, assuming that it actually works, but those colourful displays give a false sense of security.

Causes, Symptoms, and Consequences

The causes, symptoms, and consequences of this Antipattern include

- Nobody responsible for reading network, system, and security logs.
- No health and status monitoring of syslog events.
- No alarm rules for Windows configurations.
- New IDS yields numerous alerts.
- Many IDS rules disabled.

Refactored Solution and Examples

Reading the logs is an essential periodic activity; without it, you miss a lot of unusual, suspicious, and erroneous activity on your networks. Depending on the criticality of the applications, it might be necessary to review the logs daily or multiple times throughout the day.

Review the system security event logs, system logs, network device logs, and IDS/IPS logs regularly. Do not always depend on the versions in the centralized log manager, but periodically audit the local logs and make sure that they are accurately reflected in the central logs.

4.4 NETWORKS ALWAYS PLAY BY THE RULES IMP QS (question)-07M

Antipattern Name: Networks Always Play by the Rules

Also Known As: Trust All Servers, Trust All Clients, Do You Believe in Magic?

Refactored Solution Names: System Hardening, State-of-the-Art Wireless Security Protocols

Unbalanced Primal Forces: Management of confidentiality and integrity

Anecdotal Evidence: In wireless, the access point with the strongest signal is the one that user devices will trust, even if it's malicious.

Antipattern Solution

The Internet was not designed with security in mind; neither were many wireless technologies. For example, both Wi-Fi-enabled laptops and Global System for Mobile Communications (GSM) cell phones accept any base station that knows the respective protocols. There is a free security tool called Karma that can turn any Wi-Fi-enabled laptop into an imposter wireless access point. Described as Internet in a box, Karma fools other laptops into sharing their cookies for major websites and other purposes.

Yersinia is a security research tool that generates network layer 2 attacks, the data link layer. Protocols on this layer generally do not authenticate other systems, meaning whatever frames they are sent are accepted as valid and acted upon. This is the general vulnerability responsible for ARP cache poisoning (corrupting machine and Internet addresses). Yersinia can perform host spoofing and monkey-in-the-middle attacks on six additional protocols including Dynamic Host Configuration Protocol (DHCP), which is responsible for assigning Internet Protocol addresses to machines.

Many of the security issues on the Internet are due to software assuming that all others are playing by the rules; for example, assuming that other programs follow all of the Internet Requests for Comments standards specifications and always exchange reasonable parameters. Cyber exploit code and malware exploit these design assumptions by deliberately breaking the rules and catching the technology off guard, causing the targeted software to perform operations it was not designed to do and on behalf of the attacker.

Causes, Symptoms, and Consequences

The causes, symptoms, and consequences of this Antipattern include

- Lack of server authentication (HTTP, Wi-Fi, GSM, DNS, SMTP)
- Lack of client authentication (HTTP, HTTPS)
- Not monitoring networks for malformed protocols and packets

Refactored Solution and Examples

There are many inherent weaknesses in Internet technologies that you cannot mitigate. What you can do is use cyber security best practices to make your systems hard targets.

For example, harden system configurations according to best-practice guidelines. Use the most advanced, updated solutions for antivirus, anti-spyware, IDS, IPS, and Host-Based Security System (HBSS). Configure systems such as Wi-Fi-enabled laptops to require host authentication. Engineer security into the system from the beginning of the development lifecycle.

Related Solutions

Some authorities have argued for a fundamental rethinking of the Internet with much stronger support for delegation of trust and attribution of user actions.

4.5 HARD ON THE OUTSIDE, GOOEY IN THE MIDDLE

IMP QS (question)-06M

Antipattern Name: Hard on the Outside, Gooey in the Middle

Also Known As: Tootsie Pop, Defense in Depth, Perimeter Security, Protect Everything from All Threats

Refactored Solution Names: HBSS, Network Enclaves

Unbalanced Primal Forces: Management of confidentiality

Anecdotal Evidence: "Each user's browser is sending thousands of spyware beacons every day!"; Advanced Persistent Threat; "Our networks are totally secure; we have a firewall."

Antipattern Solution

Traditional network architectures include three major domains: the Internet boundary (or DMZ), the data center Storage Area Network (SAN), and the rest of the network (intranet). Between the DMZ and intranet, there are network security devices, including a firewall and possibly an IDS/IPS. Network security is concentrated at the firewall, and firewalls are assumed to protect the entire network.

In drive-by malware, attackers take advantage of the fact that most Internet browsers are configured to execute scripted code by default. If your browser encounters a malware-infected site, attacker code is executed on your system. Drive-by malware sites are widespread on the Internet. For example, there are more than 9,000 sites that distribute free antivirus protection, which is really malware in disguise.

One malware variety, ransom ware, locks up your system and demands payment. Inside the firewall, there are few internal protections on intranets. However, the greatest threat of all, the insider threat, is inside the firewall. Insider threats are most dangerous because they have legitimate network credentials, and they know about the most valuable information.

External threats penetrate networks and get inside the firewall, often through stealthy means. In a common APT scenario, specific employees are studied using their online information, such as Facebook pages, LinkedIn profiles, and other public data. In phishing attacks, targeted e-mails are crafted with malware attachments. Preying on the gullibility and curiosity of people, the malware is opened and the system infected with, for example, a key logger. The key logger sends keystroke data back to the attacker on port 80, quickly gaining the user's login credentials, and eventually a system administrator's credentials when they log in to perform maintenance. With administrative credentials, the malware can be propagated to many other machines inside the firewall. In effect, the entire intranet is owned by the attacker.

Causes, Symptoms, and Consequences

The causes, symptoms and consequences of this Antipattern include

- No protected network enclaves inside the firewall on the intranet
- No HBSS
- No configuration monitoring

- Other cyber antipatterns such as Never Read the Logs

Known Exceptions

For small networks, of perhaps fewer than 50 users, a traditional network architecture might be workable. However, additional measures, such as system hardening and HBSS, should be implemented.

Refactored Solution and Examples

For larger networks, with extensive information assets, intranet security should be carefully designed. What are the most critical information assets in the enterprise? These deserve additional protection, such as a separate firewalled network enclave, with IDS/IPS network monitoring. Security should be focused on the assets most deserving of additional safeguards.

State-of-the-art security solutions include continuous configuration monitoring. There are tools (such as Tripwire) that monitor changes to key system files (files such as the kernel and dynamic link libraries). Other tools encapsulate both file system changes and Application Program Interface (API) calls, preventing malicious actions, such as the McAfee HBSS. Some tools perform periodic security vulnerability and configuration testing, such as Retina from eEye, Nessus from Tenable, and NeXpose from Rapid7.

4.6 WEBIFY EVERYTHING

IMP QS (question)-07M

Antipattern Name: Webify Everything

Also Known As: Cross-site scripting, Cross-site Request Forgery, US Power Grid on Internet, Global Financial System on Internet

Refactored Solution Names: Physical Separation, Out of Band Separation

Unbalanced Primal Forces: Management of integrity and availability

Anecdotal Evidence: "Why the hell would they put the electrical power grid on the Internet?"

Antipattern Solution

The "webify everything" mindset defies common sense when it proliferates web interfaces for critical infrastructure. Does it make sense to proliferate easily maintained

and massively replicable remote interfaces to control electric power plants and core network devices? The so-called "Smart Grid" does webify its control devices and screens, and major providers of network devices Webify their control interfaces.

The problem is compounded by the common malware technique called cross site scripting (XSS). What Internet browsers do is execute remote code in the form of HTML, JavaScript, and other static and dynamic scripting notations.

Hypertext Mark-up Language (HTML) can no longer be considered a benign static notation. The introduction of HTML 5.0 exacerbates security issues by adding facilities for remote code execution and read-write access to local-browser client disks.

When remote code executes in an Internet browser, it has complete access to all the open browser windows, all their data, and all their implied authorities. XSS attacks combined with webified remote infrastructure control is a recipe for disaster. Whenever remote administrative interfaces are logged in and the user surfs to additional Internet sites, there is a distinct possibility that XSS attacks could gain control of highly valued infrastructure targets.

Supervisory Control and Data Acquisition (SCADA) systems are the core control systems of machines, utilities, and manufacturing infrastructure. The Stuxnet worm—which proliferated widely in the Middle East and Asia but only targeted very specific SCADA devices—proved that targeted attacks on SCADA systems are much more than theoretical.

In theory, due to the ambitions and capabilities of cyber warriors in dozens of countries, there are root kits, back doors, and logic bombs infecting much of our modern infrastructure, power plants, public works, financial systems, defense systems, and possibly air traffic control systems.

Causes, Symptoms, and Consequences

The causes, symptoms and consequences of this Antipattern include

- Web browsers are a user interface platform for applications, called thin clients. Thin clients are used ubiquitously and are convenient for system administrators because there is no client software installation or client software updates.
- Users are in the habit of opening multiple browser tabs and connecting with multiple websites. Websites with malicious content are a significant and

prevalent threat. Malicious content (such as malware scripts) can be embedded in the site or served up through advertisements supplied by third parties.

Refactored Solution and Examples

Software Virtual Private Networks (VPNs) provide out-of-band separation of communications across public networks. This means that interception of packets using technologies like network sniffers is essentially prevented.

VPNs are a widely deployed technology; one wonders why VPNs aren't used universally.

For example, providers that send unencrypted e-mails across the Internet are inviting exploitation of the data and possible future attacks.

Related Solutions

To prevent XSS and other attacks, the American Banker's Association recommends using a dedicated, physically separate computer for all financial transactions.

Although this seems like an extreme solution, it's a realistic response to the threats. Chances are that a hardened, well-patched computer, which is only used to connect to trusted financial sites, is much less likely to be compromised by malware than a computer that is exposed to general Internet surfing.

4.7 NO TIME FOR SECURITY IMP QS (question)-07M

Antipattern Name: No Time for Security

Also Known As: Add Security Last, Blame Security for Schedule Slippage, Deliver It Now!

Refactored Solution Names: Security Requirements Are Real Requirements, Cyber Risk Management

Unbalanced Primal Forces: Management of confidentiality, integrity, and availability

Anecdotal Evidence: "Wait until it's time to test the system, and then worry about security."

Background

Security is usually the final consideration in the development of a system. Sometimes security is left out altogether in the rush to get products out the door.

Antipattern Solution

Developers of software projects, and now also widget developers, often wait until the end of the development lifecycle to address security. Near the date that the enterprise release process will test security vulnerabilities, managers and developers begin a madcap cover-up process to obscure inherently insecure software, user account, and configuration practices. When confronted, developers can claim ignorance; they are not security experts after all.

Causes, Symptoms, and Consequences

The causes, symptoms, and consequences of this Antipattern include

- Security was never part of the requirements.
- Saving on development costs and time at the expense of security.
- Project is behind schedule.
- Shared administrator accounts.
- Not training the developers to be security aware.

Known Exceptions

If the software is out-of-the-box, it is already near the end of the development cycle and can be configured for security just prior to deployment. However, you are taking it on faith that the original software developers accounted for security and built in appropriate configuration settings.

Refactored Solution and Examples

Security risks and requirements should be analyzed early in the development cycle at the same time as functional requirements. This is not as difficult or expensive as it sounds. Business stakeholders should categorize the system, such as: confidentiality high, integrity medium, and availability medium (see NIST SP 800-30, -37, -53 or CNSSI 1253). You can select the baseline security requirements rather mechanically using these profiles and the NIST 800-53 controls catalog. This delivers a set of security

requirements close to the desired set, which you can tailor to your specific situation during development. The security requirements should be given first class status in the overall requirements set.

Related Solutions

You can select security and audit controls using the Committee on Sponsoring Organizations (COSO) and Control Objectives for Information and Related Technology (COBIT) frameworks for commercial systems and to satisfy Sarbanes Oxley requirements.

QUESTION BANK – NETWORK AND CYBER SECURITY

MODULE-4

1. Discuss Security Architecture. 5M
2. Discuss Antipattern: signature based malware detection versus polymorphic Discuss threads, document driven certification and accreditation. 6M
3. Discuss policy driven security certifications. 6M
4. Discuss Refactored solution: reputational, behavioural and entropy based malware detection. Detection versus polymorphic threads. 6M
5. With a neat diagram, explain Antipatterns Concept. 6M
6. Discuss forces in cyber antipatterns. 6M
7. Discuss Cyber Antipattern Templates and its types.10-12M 10-12M
8. Discuss cyber security Antipattern catalog.
9. Discuss Can't Patch Dumb. 8M
10. Discuss Unpatched Applications. 8M
11. Discuss Never Read the Logs. 8M
12. Discuss Networks Always Play by the Rules. 8M
13. Discuss Hard on the Outside Gooey in the Middle. 8M
14. Discuss Webify Everything. 8M
15. Discuss No Time for Security. 8M
16. Short note on 1) Can't Patch Dumb. 2) Hard on the Outside, Gooey in the Middle. 3) Webify Everything. 4) No Time for Security. 12M
17. Short note on 1) Unpatched Applications.2) Never Read the Logs.3) Networks Always Play by the Rules.4) Hard on the Outside Gooey in the Middle. 12M

NETWORK AND CYBER SECURITY
(15EC835, 17EC835)

8TH SEM E&C

JAYANTH DWIJESH H P **BE (ECE), M.tech (DECS).**

Assistant Professor – Dept of E&CE, BGSIT.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

B.G.S INSTITUTE OF TECHNOLOGY (B.G.S.I.T)

B.G Nagara, Nagamangala Tq, Mandya District- 571448

NETWORK AND CYBER SECURITY

MODULE–5

MODULE–5

Cyber network security concepts contd.

Enterprise security using Zachman framework

Zachman framework for enterprise architecture, primitive models versus composite models, architectural problem solving patterns, enterprise workshop, matrix mining, mini patterns for problem solving meetings.

Case study: cyber security hands on – managing administrations and root accounts, installing hardware, reimaging OS, installing system protection/ antimalware, configuring firewalls.

TEXT BOOK:

1. **William Stallings, "Cryptography and Network Security Principles and Practice", Pearson Education Inc., 6th Edition, 2014, ISBN: 978-93-325-1877-3.**
2. **Thomas J. Mowbray, "Cyber Security – Managing Systems, Conducting Testing, and Investigating Intrusions", Wiley.**

REFERENCE BOOKS:

1. **Cryptography and Network Security, Behrouz A. Forouzan, TMH, 2007.**
2. **Cryptography and Network Security, Atul Kahate, TMH, 2003.**

MODULE–5

Cyber network security concepts contd.

Enterprise security using Zachman framework

Zachman framework for enterprise architecture, primitive models versus composite models, architectural problem solving patterns, enterprise workshop, matrix mining, mini patterns for problem solving meetings.

Case study: cyber security hands on – managing administrations and root accounts, installing hardware, reimaging OS, installing system protection/ antimalware, configuring firewalls.

CHAPTER -1 ENTERPRISE SECURITY USING ZACHMAN FRAMEWORK

1. THE ZACHMAN FRAMEWORK FOR ENTERPRISE ARCHITECTURE

IMP QS (question)-10M

The Zachman Framework, invented by John A. Zachman, is an intellectual tool for describing enterprises. Because enterprises are inherently complex, you Need a powerful framework to describe them—a framework that divides and Conquers complexity.

Figure1:-

- The Zachman Framework (see Figure 1) slices and dices complexity into rows and Columns.
- The columns are the six basic questions you could ask about any subject. These interrogatives include: What? How? Where? Who? When? Why? These are the same questions journalists ask to write newspaper stories. When a journalist has answered these six questions, he or she can claim to have a complete story.
- The Zachman Framework further slices and dices complexity into rows.
- The rows represent a general overview of the human roles. The hierarchy of every complex enterprise has: executives, business management, architects, engineers, Technicians and users. Each of these roles can ask the same six questions; hence six cells per row.
- Each row-column intersection in the Zachman Framework is a cell to be populated with models and specifications, which are representations of the enterprise. Everyone has their own specifications.

- A populated row represents the enterprise architecture from that row-wise perspective.

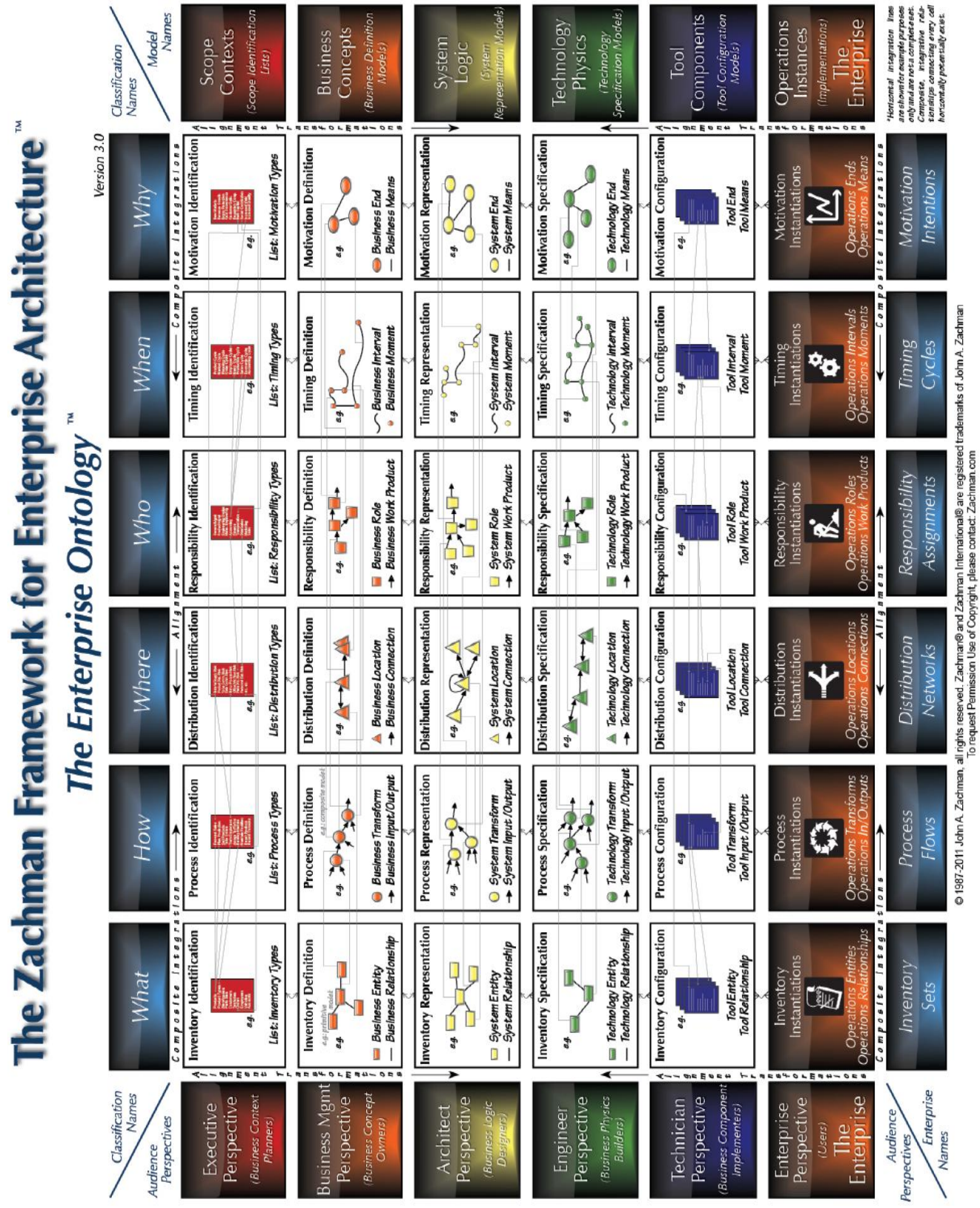


Figure 1: The Zachman Framework

2. PRIMITIVE MODELS VERSUS COMPOSITE MODELS

IMP QS (question)-08M

Primitive model:-

- A primitive model is one that only includes entities from a single column.
- Row 1 contains lists of primitives from each column. Row 2 has hierarchies built from those lists (refer to Figure 1).
- Keeping the columns separate in our models has many advantages. It makes the framework conceptually simple; the model based upon the Zachman Framework can also be simple; and because the columns are independent, you can populate them in parallel.
- Primitive models are relatively stable and slow-changing.

Example (primitive model): -

Consider the information models in Column 1 of Figure 1; our models can include a list of data entities (Row 1), a data hierarchy (Row 2), a conceptual schema (Row 3), a logical schema (Row 4), and a physical schema (Row 5), and record instances (Row 6).

Certainly, Row 6 changes constantly in the implementation, but the other cells would be relatively stable. How often do you change the physical schema requiring software re-engineering? Probably not that often, in most enterprises.

Composite models:-

- Composite models combine primitives from multiple columns.

Example (Composite models):-

An example is matrix is a matrix that shows relationships between primitives, such as processes versus data.

With that matrix you can answer the question, “Which processes use which data?” Composite models are necessary for assessing impacts between columns (such as what data is manipulated by which processes) and for describing implementations.

3. ARCHITECTURAL PROBLEM SOLVING PATTERNS

IMP QS (question)-08M

- People have been applying the Zachman Framework pervasively since its public release two decades ago.
- An early method, documented by Stephen Spewak in the classic book Enterprise Architecture Planning, explained how to populate the first few rows with models. It was a laborious process, easily taking six months or more, and the business benefits of doing the exercise were not exactly clear, but the customer's sweat equity should create a psychological buy in.
- Twenty years have passed and the techniques have evolved through thousands of engagements. The key techniques are as follows:
  - **Business Question Analysis:**
    - Gather knowledge from enterprise subject matter experts (SMEs—for example, experienced business modelers) to find out what questions the business management has.
    - Analyze each question to understand which columns are involved to answer the questions, and which columns need to be mapped to which others. This analysis determines which hierarchies and matrices are needed.
  - **Document Mining:**
    - Obtain as much enterprise documentation as possible. Choose a column and go through each document finding examples.
    - Keeps a list of what you found, the enterprises' text defining it, and the document and page number where it can be found for traceability). Completing document mining for all the interrogatives will populate Row 1.
  - **Hierarchy Formation:**
    - Play a cards-on-the-wall exercise with small groups and organize each list into a hierarchy, possibly inventing some new categories in the middle of the tree.
    - Redraw this electronically and print it as a readable poster. When complete, you have six hierarchies that populate Row 2.

- **Enterprise Workshop:**
  - Bring the posters and some binders with the Row 1 definitions to a workshop with enterprise stakeholders.
  - Have the enterprise take ownership of the meeting and walk through each hierarchy to validate the models. This workshop is usually done one Hierarchy at a time.
- **Matrix Mining:**
  - Carefully review the documents for cross-column relationships, that is, a sentence involving more than one column.
  - Keep track of each relationship, including document, quoted text, and page number.
  - Then conduct an enterprise workshop to validate the matrices.

➢ These techniques use documents rather than interviews because documents are multi-person vetted content.

➢ Interview results are less reliable because they depend upon a single person's opinion in whatever situation that person is in that day.

➢ In the following catalog of Architectural Problem Solving Patterns, there are primary modes of thinking: divergent, convergent, and information sharing (see Figure 2). These correspond to generating ideas, selecting ideas, and defining ideas.

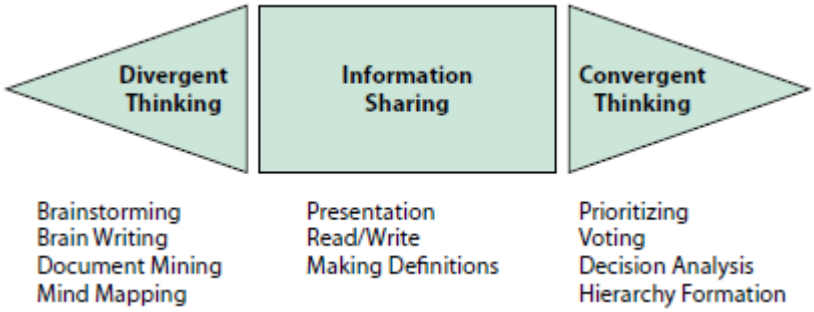


Figure 2: Problem-solving types

4. ENTERPRISE WORKSHOP

IMP QS (question)-06M

Also Known As: Large Group Consensus Forming, Review Meeting.

Problem Solving Type: Information Sharing, Convergent Thinking.

Process Roles: Customer Lead (ideally) or Facilitator.

Content Roles: Customer Review Team.

Communication Techniques: Hierarchy Posters, Written Descriptions on PowerPoint, Sticky Wall.

Range of Durations: 1 and 1/2 hour to 6 and 1/2 hours.

Background

The customer review team is a larger group than the customer leads, whose consensus will be required to proceed to implementing the problem solution. The purpose of this workshop is to share information and build consensus by soliciting input into the solution.

Preparation

Create large-format posters for the hierarchies. Create a three-ring binder for all Other materials, such as the Excel listings of the primitive definitions and their sources. Prepare headings and items for the sticky wall. 72-point Times New Roman works for a sticky wall. Include some information that ties each sticky wall item back to its definition.

Procedure

Bring the posters and some binders with the Row 1 definitions to a workshop with enterprise stakeholders. Have the enterprise take ownership of the meeting and walk through each hierarchy validating the models. This workshop is usually done with one hierarchy at a time.

5. MATRIX MINING

IMP QS (question)-06M

Also Known As: Creating Another Matrix.

Problem Solving Type: Convergent Thinking.

Process Roles: Task Lead who is EA Problem Solving SME.

Content Roles: The Team.

Communication Techniques: Silent document review; internal review workshop.

Range of Durations: 1 hour to 2 days (depends on the documentation).

Background

Matrices are composite models that show relationships and effects between columns of the Zachman Framework.

Preparation

Reuse the documents collected for document mining.

Procedure

Carefully review the documents for cross-column relationships—that is, a sentence involving more than one column, for example, a role and a process. Keep track of each relationship, including document, quoted text, and the page number. Then conduct an enterprise workshop to validate the matrices.

6. MINIPATTERNS FOR PROBLEM SOLVING MEETINGS

IMP QS (question)-08M

These mini patterns are additional techniques to round out your meeting facilitation skills. Techniques such as breakouts and the idea parking lot are classic approaches for conducting effective meetings.

Get Organized

If you have no agenda, brainstorm these two questions on a flipchart:

- Why are we here?
- What outcomes do we want?

Breakouts

- Meetings are least productive when only one person talks and everyone else does nothing but listen and take notes. In general, people's creativity is inhibited in groups larger than five.
- The facilitator can ask that the group form small discussions to address a particular question, and then have them report back their conclusions subgroup by subgroup.

- Another approach is to quickly generate a list of topics or concerns (brainstorming or brain writing) and then have each breakout take one problem to solve as a subgroup before debriefing the general session.

Flipcharts

- Unlike a computer or a whiteboard, flipcharts give a group unlimited space for creativity. When a page of a flipchart is filled, it is moved and taped to a nearby wall.
- Flipcharts are group notes; people do not need to be taking their own notes; they can have their heads up and be fully engaged in the meeting.
- Flipcharts are also highly portable, unlike whiteboards.

Time Management

- If you plan an agenda, plan the time of each meeting topic, and stick to it. Or ask the group if they want to extend the time. Assign a time keeper to remind the group. Make sure there is a highly visible clock in the meeting room. Time consciousness keeps people focused on problem solving.

Ground Rules

Have some ground rules for each meeting so that distractions are minimized, and the group doesn't waste time.

Idea Parking Lot

Post a separate flipchart to capture ideas that are outside the meeting's purpose. Revisit these ideas at the end of the meeting and decide as a group how they should be addressed.

Other Problem-Solving Catalogs

General problem solving and business meeting facilitation are similar disciplines, and they share common catalogs of techniques. Perhaps the most widely used and respected catalog of techniques is Techniques of Structured Problem Solving by Arthur VanGundy (Springer, 1988, ISBN 978-0-442-28847-1).

New techniques are constantly being developed. Beyond problem solving there is a new generation of techniques based upon possibility thinking. More than 60 possibility-thinking methods have been published (Holman & Devane, 2007). Using techniques such as The Circle Way, The World Cafe, Open Space, and Appreciative Inquiry, groups are encouraged to explore new possibilities based upon organizational strengths rather than dwelling on problems and weaknesses.

CHAPTER -2 CASE STUDY

1. MANAGING ADMINISTRATOR AND ROOT ACCOUNTS

IMP QS (question)-06M

- As a network administrator, you are granted a privileged user account on many networked systems and devices.
- Privileged or administrative accounts can exercise unlimited authority on your systems and networks. Some key best practices for managing privileged accounts include:
  - All users, including network administrators, should normally use unprivileged, non administrative accounts.
  - Administrative operations should be effectively separated from other user activities.
- **For example,** e-mail and Internet browsing should not be performed using administrative accounts or while managing devices/services remotely. Careful use of legitimate websites to perform software updates and upgrades is acceptable. These policies are essential for network security for the following reasons:
  - Logged in with a privileged account, a user receives an unexpected but authentic-looking e-mail and opens its attachment, which installs a root kit. A root kit is malicious software that takes complete control of an account for a remote attacker. By compromising a privileged account, the entire system (and possibly the entire local area network [LAN]), all its accounts, computing power, and data are compromised.
  - A network administrator, logged in as root super user, visits a drive-by malware website; a root kit is installed unknowingly. Now the attackers have administrative privileges on the network.

- A network administrator has web browser windows open for managing Cisco routers, Oracle databases, and the company's website. Over the lunch hour, the administrator does some personal Internet browsing and stumbles upon a website that performs a cross-site scripting (XSS) attack. XSS attacks involve running malicious scripts inside the administrator's browser; the scripts have all the authority of the network administrator in all open browser windows and tabs.

- If users had followed proper security policies, only individual, non privileged accounts—rather than entire systems, networks, and remote devices/services—would be compromised.
  - Example nightmare scenarios of XSS attacks include an administrator's browser remotely managing an electrical power grid, an air traffic control system, life-critical hospital systems, manufacturing controls, banking systems, or military weapons systems. Common sense dictates that these systems should be rigorously protected or completely separated from the Internet.
- It is common practice to never use the default administrative accounts and set them up as honey pots, create new administrator (or root/su access) accounts and audit the default ones so you can see who is attempting to use them

1.1 Windows

IMP QS (question)-04M

- Windows accounts are assigned either administrative or normal user privileges.
- You can log out as a normal user and log in as an administrative user to gain administrative privileges.
- You can then repeat the process in reverse to return to a normal account.
- On newer Windows systems, you can switch user accounts and gain temporary administrative privileges through User Account Control (UAC).
- UAC was first introduced in Windows Vista and Windows Server 2008.
- When you attempt a privileged operation, UAC challenges you for an administrative account password if you are logged into a system with a non administrative account and attempt to do something on the system that requires an elevated level of permissions.

- By default, windows command shells are non privileged. To create a privileged shell, choose Start ➪ All Programs ➪ Accessories and then right-click Cmd (command shell) and select Run as Administrator.

1.2 LINUX AND UNIX

IMP QS (question)-02M

` The administrative account in Linux and UNIX is username root. From a non root Account, Linux users can switch to root with the super user command, su -, which Challenges you for the root password. The dash (-) means also adopt the root's environment variables and home directory. Use the exit command to de-escalate back to the user account. Use the super user command to log into other accounts, such as *su - my account.*

1.3 VMware

IMP QS (question)-03M

- Of the three VMware platforms covered in this chapter, only ESXi has administrative accounts, whereas VMware Player and VMware Workstation are single-user desktop applications without user authentication.
- ESXi is a sparse operating system, whose entire purpose is to administer virtual machines.
- It does not have Internet browsers or other user applications. As such, it makes sense for all network administrators to use privileged accounts on this operating system.

2. INSTALLING HARDWARE

IMP QS (question)-10M

- Computer hardware is generally not OS specific, although some platforms are unable to boot certain OSes.
- The basic components for desktop systems are pedestal (processor, memory, and peripherals), display/monitor, keyboard, pointing device, uninterruptable power supply (UPS) systems, and cables for all of the components plus a network cable.
- Some systems are more bundled, such as laptops and Apple Macs, which bundle everything except the wireless mouse and wireless keyboard.

- For pointing devices and keyboards, the standard connector is the Universal Serial Bus (USB), except that the obsolete IBM PS2 connectors are still in use in some shops.
- Wired network CAT 5 cables use RJ-45 connectors.
- Although wired network cables are standardized on RJ-45, there are dozens of competing standards for fiber optic cables.
- Each shop must adopt one of the standards compatible with the Network Interface Card (NIC) and network switch components.
- Computer monitors usually conform to one of the prevalent standards: Video Graphics Adapter (VGA) or Digital Video Interface (DVI).
- Some monitors have both types of connectors and support switching between pedestals.
- The standard power cable is called the pigtail and conforms to each country's electrical standards. Usually, there are separate pigtails for the monitor and the pedestal.
- You should consult your manufacturer's instructions for specifics about setting up your system, but a typical hardware setup sequence for a desktop pedestal Includes the following:
    1. Position the components on top of the desk. Optionally, if the system has a separate floor pedestal then it is placed underneath the desk along with the UPS, with the backside and connector ports facing the installer.
    2. Connect the monitor pigtail and display cable and secure the thumbscrews, if any.
    3. Feed the monitor, mouse, and keyboard cables down through a desktop opening, or around the back/side.
    4. Connect the network, monitor, mouse, and display cables to the pedestal.
    5. For a new UPS, you may need to connect the battery by removing a panel and an instructions sticker and then fastening an internal plug.
    6. Connect the pigtail to the pedestal and then connect it to the UPS. Connect the UPS to the electrical outlet. Turn on the UPS. Double-check all cable connections.
    7. Always verify your work. Make no assumptions. Turn on the computer; verify that the monitor displays the boot sequence on the screen. If an operating

system is installed, continue booting to check keyboard, mouse, and network functionality. Alternatively you can test the system using bootable CD/DVD test tools, such as Back Track, Caine, or Helix.

- For server rooms, vertical rack capacity is measured in rack-mountable units.
- Standard 19″ server racks range from about 3′ (19 U) to nearly 7′ (42 U) in height, and 3′ in depth. Nineteen inches is the outside width of the mounting brackets.
- For server rooms, rack-mounted systems generally come without monitors or keyboards.
- A single user interface (UI) can serve an entire rack using a Keyboard Video Mouse (KVM) switch.
- A rack console is typically 1 U and mounted on movable rails with an integrated trackball (rather than mouse), keyboard, and display.
- KVM switches use standard connectors for the server and console cables, and some KVM use adapters and proprietary connectors on the KVM end.
- You can convert many pedestals to rack-mounted servers.
- The following is a typical installation sequence for a sliding-rails rack server:
  1. Unpack the server and its components, such as server rails and cables.
  2. Affix the server rails to the 19″ rack. Some rails require screws, and some are screw less (spring loaded). A rack with square holed brackets may require special fasteners (nuts) with clips to affix to the rack bracket.
  3. From the front of the rack, extend the rails out of the rack and snap the server into the rails.
  4. Retract the server into the rack.
  5. From the back of the rack, connect the network and KVM switch cables. Oracle Sun systems usually have a single USB cable for monitor and keyboard.
  6. Connect the pigtail to the server and UPS or UPS-connected power strip. Double-check that all cable connections are fastened properly.
  7. From the front of the rack, powers up the server. Some servers and many network devices are dedicated and always on.
  8. Verify your work. Using the KVM, verify the functionality of the keyboard, mouse, and display connections.

9. Note: Do not connect the system to the network until it is patched and secured with anti-malware (e.g. antivirus, software firewall); after you have taken those precautions, you can attach the network cable and test network connectivity.

- Common network devices are modems, firewalls, routers, and switches.
- Modems connect an internal network to telecommunications networks, such as Digital Subscriber Line (DSL), T1, E1, T2, and E2 telecommunications provider services.
- Firewalls are network boundary devices enforcing rules for separation between internal and external network communications.
- Routers interconnect network segments by routing and switching data using Internet Protocol (IP) addresses between segments.
- Routers also translate IP addresses and Media Access Control (MAC) addresses within network segments.
- Each network interface card (NIC) has a unique hardware (MAC) address.
- For Ethernet, a network segment is considered a broadcast domain between computers, whose NICs operate according to the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) algorithm.
- The terms network segment, subnet, and LAN are used interchangeably.
- Network switches are repeater devices that extend network segments by mirroring all broadcast signals to additional cable connections.

3. RE-IMAGING OPERATING SYSTEMS

- In any network or security shop, rebuilding systems is a way of life.
- People rebuild systems to recycle hardware or after a system failure.
- People also rebuild a system if it is seriously contaminated by malware, such as a root kit.
- If systems are networked at a hacker convention or cyber security class, a rebuild is strongly recommended.

3.1 WINDOWS

IMP QS (question)-06M

- As a product of Microsoft, Windows is a licensed software package that must be purchased and installed legally.

- Ideally, you retain the original manufacturer installation disks that come with the system purchase.
- The original disks help you avoid a key headache: Microsoft Activation.
- If you install from a non-manufacturer source, Microsoft requires that you activate your system. You can perform the activation via the Internet or a phone call to Microsoft's interactive voice response (IVR) system.
- One-off Windows purchases allow one activation per license key.
- Other ways to buy Windows may give a fixed number of activations, such as 10 from the same license key from Microsoft Action Pack Solution Provider Subscription (MAPS) or Microsoft Developer Network (MSDN) subscription.
- If activation is required, you see onscreen instructions after rebooting.
- The following is a typical re-imaging sequence for the Windows OS:
    1. Obtain the installation disks for Windows. Each network shop should retain a set of master re-imaging disks from each system purchase.
    2. Verify that the system in not connected to the network. You will secure the system before you allow it on the production network.
    3. Power on the machine and open the CD/DVD drive. Insert the first Windows install disk. Reboot or power down and restart. The first screen is the hardware boot screen (sometimes called the BIOS screen). Options will be offered for accessing the Boot Device Menu (usually F9 or F12). If the system does not automatically boot off the DVD then recycle the power and access the Boot Device Menu. In the Boot Device Menu, select CD/ DVD using the arrow keys and and then press Enter.
    4. Follow the on-screen instructions for installation. It is usually better to set the Administrator password after Windows boots because it is easy to get an inaccurate password at this stage of the installation, and that would require you to completely redo the installation process.
    5. Insert a DVD containing device drivers for this system. Driver disks are provided with the hardware. Alternatively, you can download updated drivers from the manufacturer's website, burn them to disk, and load them offline. Make sure that you install drivers for all of the major devices: display, keyboard, mouse, network, sound, CD/DVD, USB, and miscellaneous, depending upon what hardware and drivers are available.

6. The next three sections continue the installation instructions, including explaining how to download patches, burn CDs, transfer files, secure the network with anti-malware tools, and install applications.

3.2 LINUX

IMP QS (question)-06M

- The two largest Linux families are descendants of Debian and Red Hat, commonly called distributions or distros.
- There are also optional desktops for Linux variants, some of the most popular being KDE and Gnome. GUI applications are specialized for each desktop. Here are various sites where you can download popular Linux distributions for free:
  - www.backtrack-linux.org: Back Track/ Ubuntu /KDE is a Debian Linux distribution with an entire suite of penetration test tools preinstalled.
  - www.ubuntu.com/: Ubuntu is a very popular OS, primarily for its command line application installer, the apt-get command. An expanding open source community is creating numerous Ubuntu applications. Ubuntu installs Gnome desktop by default.
  - www.opensuse.org/en/: open SUSE is the open source version of Novell's SUSE Linux. It is Red Hat derived and features optional KDE and Gnome desktops (or shells) configured by the installer. Open SUSE includes a large number of pre-installed applications.
  - www.centos.org/: Centos is descended from the Red Hat open source code and runs Gnome desktop by default. Centos bills itself as an enterprise OS. The optional Yum extender is a GUI tool that manages updates and application installations.
  - http://fedoraproject.org/en/get-fedora: Fedora is an open source version of Red Hat Enterprise Linux, and it runs Gnome desktop.
- Linux installation procedures are similar in sequence to Windows re-imaging (described earlier in this chapter). Depending upon the packages you choose, only the first two disks may be required for Red Hat Enterprise installation, or perhaps all six disks, as expected.
- Ubuntu, running as an ISO DVD can install itself onto a hard disk through a command like the following:

```
# Ubiquity --desktop %k gtk_ui
```

- This command launches a graphical installer GUI, which includes various installation options such as selection and resizing of disk partitions.

3.3 VMware

IMP QS (question)-02M

- VMware Player cannot install new VMs, but VMware Workstation and ESXi can. It's easier to install a new VM with VMware Workstation because it works locally on your PC desktop.
- Use Workstation to create a new VM and then connect to the hardware CD/DVD on the host device.
- Insert the installer CD and install exactly the way you would install a physical machine. In Workstation choose VM ➪ Install VMware tools to get VM-specific drivers into the OS for networking and other functions.

3.4 Other OSes

IMP QS (question)-02M

Solaris installation works somewhat differently than other OSes, in that it enables you to perform network setup during the installation procedure. This requires more information upfront from local network administrators. It is a good idea to use these installation options, as manual setup requires significant research.

4. INSTALLING SYSTEM PROTECTION / ANTI MALWARE

IMP QS (question)-08M

- An unprotected system directly exposed to the Internet has a life expectancy of about 10 minutes. That is the principle behind passive honey pots, which are machines purposefully exposed to the Internet in order to capture malware.
- Antivirus is the most obvious protection that you should install, enable, and set for automatic updating (or at least semi-automatic updating).
- But there are several other important protections, and the range of protections needed is constantly rising as threats escalate. Because perimeter security is insufficient to combat current and future threats, the trend in protection is toward host-based security (HBS), which provides a full array of network defenses on each machine.

- The security industry has not settled on the full scope of HBS, and actually implements its principles fairly poorly.
- HBS can be implemented with a combination of location protections and services and enterprise services that manage local configurations and services.
- For example, an enterprise antivirus solution can update malware signature
- Databases on machines throughout the network. A full-scope HBS would include technologies such as
  - Antivirus
  - Anti-spyware
  - Firewall
  - Intrusion detection
  - Intrusion prevention
  - Blacklisting
  - Real-time integrity checking
  - Periodic policy scanning
  - Rootkit detection
  - Patch management
- Antivirus protection scans for malicious files. There are several types of scans: on-demand, scheduled, and continuous.
- On-demand and scheduled scans start at a discrete time and continue until all the requested files are scanned.
- Continuous scans are performed in near-real time as files are added or modified.
- Traditional antivirus protection recognizes malware through signatures, usually by matching the hash function of files with a known malware database. These databases must be updated frequently to catch the latest malware. It's recommended that you configure for automatic updating.
- Anti-spyware searches for suspicious applications that might be collecting data without the users' knowledge.
- Spyware applications are often installed covertly, as the user is surfing a website.
- Both antivirus and anti-spyware programs either quarantine or remove the malicious file.
- A quarantined file is temporarily disabled, usually by moving it to a sandboxed directory. An administrator may restore it.

- An Intrusion Detection System (IDS) scans network traffic for potentially malicious packets and sends alerts and the packet to log files.
- An Intrusion Prevention System (IPS) can dynamically block network traffic based upon alerts. A blacklist is a list of blocked domains, IPs, or IP address blocks.
- A real-time integrity check monitors key OS files for changes and alerts when it detects potentially malicious changes. A periodic policy scan checks the security settings of registry keys, Group Policy objects, services, and applications, alerting when it detects variation from accepted standards for secure systems.
- Root kit detection is a scan that seeks serious malicious system infections, which might evade other defenses by cloaking their activities from normal observations—for example, directory listings.
- Patch management ensures that the OS and applications have the latest developer-recommended updates. Although OS patching is reasonably prompt in most organizations and homes, application patching remains a major vulnerability.
- Microsoft initiated a ritual monthly update called Patch Tuesday. It is the second Tuesday of each month and coincides with patch updates from many vendors.

4.1 WINDOWS

IMP QS (question)-02M

- Some of the free antivirus packages for Windows are available from www.avast .com, www.clamav.net, http://free.avg.com, and www.malwarebytes.org.
- Some free anti-spyware packages are available from http://superantispyware.com and http://lavasoft.com (Ad-Aware).
- Free root kit detection tools are available from www.safer-networking.org (Spybot S&D) and www.microsoft.com (Malicious Software Removal Tool).
- As a first step, always update your antivirus databases to the latest releases, even if they're newly installed, and enable automatic updates.

4.2 LINUX

IMP QS (question)-04M

- The anti-malware market for Linux is considerably smaller than Windows; hence there are fewer offerings of both free and pay solutions.

- Some of the free antivirus packages for Linux are available from www.clamav.net and www.free .avg.com. Linux packages in general are less turnkey than Windows. For example, you can install Clamav on Ubuntu with the following command:

```
# apt-get install clamav
```

- Similarly on Red Hat Linux, you can search and install Clamav using Yum extender. To update the Clamav antivirus signature databases twice a day, use the following command:

```
# freshclam –d –c 2
```

- The fresh clam command does an immediate update if invoked with no arguments. To run a Clamav scan on the entire file system, use the following:

```
# clamscan –r /
```

- You can substitute a specific directory path to scan a subset of the file system. The best way to verify an ISO or downloaded data is to run a hash against it.
- A hash is the output of a program that calculates a unique digital value for a block of program or data. Even a single bit change can significantly change hash values.

4.3 VMware

IMP QS (question)-02M

- The VMware infrastructure, which in the ESXi release contains no other OS code, generally requires no anti-malware on its own. However, the guest VM operating systems require the same protection that you would afford any OS.

5. CONFIGURING FIREWALLS

IMP QS (question)-10M

- Firewall setup is usually delegated to specialists with vendor certifications.
- However, it is useful for you to expose yourself to how firewall configuration is performed, which is similar to network switch configuration.
- From time to time uncertified network administrators and other security professionals will be called upon to verify firewall configurations. This example is for configuring a Cisco ASA 5000 series firewall.

- A Windows host is used as the console terminal. Connect it directly to the firewall using the Cisco console rollover cable if the Windows system has a serial port. Otherwise, use a USB to serial patch the cable attached to the console cable.
- On Windows, choose Start ➪ All Programs ➪ Accessories ➪ System Tools ➪ HyperTerminal.
- Assuming the connection is setup by default, the commands look like the following:

```
$ enable
Password:
# show run
# config t
(config)# interface vlan 2
(config-if)# name if inside
(config-if)# Security –level 100
(config-if)# ip address 10.10.100.1 255.255.255.0
(config-if)# no shut
(config-if)# exit
(config)#
(config)# interface vlan 3
(config-if)# nameif outside
(config-if)# security –level 0
(config-if)# ip address 192.168.10.2 255.255.255.0
(config-if)# no shut
(config-if)# exit
(config)# route outside 0.0.0.0 0.0.0.0 192.168.10.1
(config)# int e0/1
(config-if)# switchport access vlan 2
(config-if)# speed 100
(config-if)# duplex full
(config-if)# no shut
(config-if)# exit
(config)# int e0/2
(config-if)# switchport access vlan 3
```

(config-if)# speed 100
(config-if)# duplex full
(config-if)# no shut
(config-if)# exit
(config)# wr mem
(config)# exit
show run
exit

- The previous commands set up inside (vlan 2) and outside (vlan 3) virtual local area networks (VLANs).
- Ports 1 and 2 are then configured and associated with the VLANs at 100 megabits per second full duplex. By convention, v lan 1 is avoided because it exists on all Cisco switches.
- Communications on VLAN interfaces are denied by default, so access rules must be established to enable communications. The following commands configure access rules for a straightforward network:

$ enable
Password:
Show run
config t
(config)# access-list in2out extended permit tcp 10.10.100.0
255.255.255.0 any eq http
(config)# access-list in2out extended permit tcp 10.10.100.0
255.255.255.0 any eq https
(config)# access-list in2out extended permit tcp 10.10.100.0
255.255.255.0 any eq domain
(config)# access-list in2out extended permit udp 10.10.100.0
255.255.255.0 any eq domain
(config)# access-group in2out in int inside
(config)# access-list out2in extended permit tcp host 192.168.10.101
10.10.100.0 255.255.255.0 eq ssh
(config)# access-group out2in in int outside
(config)# wr mem

(config)# exit

show run

exit

- An access list (in2out) is defined with rules allowing internal hosts (10.10.100.0/24) to communicate using HTTP, HTTPS, and DNS protocols with any address.
- The access-list commands define the rules, and the access-group command assigns the rules on the interface "inside." An external-maintenance IP address (192.168.10.101) is allowed to connect to any host inside the network using SSH. That rule is applied to the interface "outside."
- The firewall can be set up to shun or block specific external IP addresses. To set up a shun of an outside host and then remove it, use these commands:

(config)# shun 64.94.107.0

(config)# no shun 64.94.107.0

- You can block an address range by using the 0 as a wildcard. Search the firewall command reference for additional operations.
- Make sure that the commands apply to your specific firewall model number, as commands vary significantly even within one series of devices.
- Increasingly, most outbound traffic is destined for ports 80 (HTTP) and 443 (SSL).
- Due to firewall conventions, most inbound connections are denied. Malware takes advantage of this fact to disguise malicious connections by originating them from inside networks and using ports 80 and 443.

QUESTION BANK – NETWORK AND CYBER SECURITY

MODULE-5

1. With a neat diagram, explain the zachman framework for enterprise architecture. 10M
2. Discuss primitive models versus composite models. 8M
3. Discuss architectural problem solving patterns. 8M
4. Discuss enterprise workshop. 6M
5. Discuss matrix mining. 6M
6. Discuss mini patterns for problem solving meetings. 8M
7. Discuss managing administrator and root accounts. 8M
8. Short note on 1) windows (managing administrator and root accounts),2)Linux and unix,3)VMware. 8M
9. Discuss installing hardware. 10M
10. Short note on 1)windows (re-imaging operating systems),2)linux,3)VMware b4)other oses . 8M
11. Discuss windows (re-imaging operating systems). 6M
12. Discuss installing system protection / anti malware. 8M
13. Short note on 1) windows (installing system protection / anti malware), 2) Linux, 3) VMware. 8M
14. Discuss Configuring firewalls. 10M