1) Explain Secure Socket Layer (SSL) architecture.

- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections
- **SSL connection**
  - a transient, peer-to-peer, communications link
  - associated with one SSL session

A session state is defined by the following parameters (definitions from SSL specification):

- 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, DES etc.) and a hash algorithm (such as MD5 or SHA-l) used for MAC calculation. It also defines cryptographic attributes such as the hash size.
- Master secret: 48-byte secret shared between the client and server.
- Is resumable: A flag indicating whether the session can be used to initiate new connections.

A connection state is defined by the following parameters:

- **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 conventional encryption key for data encrypted by the
- server and decrypted by the client.
- **Client write key:** The conventional 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- initialized by the SSL Handshake Protocol.
- **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection.

---

2) #What protocols comprise Secure Socket Layer (SSL)?

3) Explain Secure Socket Layer (SSL) record protocol.

- **confidentiality**
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
  - message is compressed before encryption
- **message integrity**
  - using a MAC with shared secret key
  - similar to HMAC but with different padding

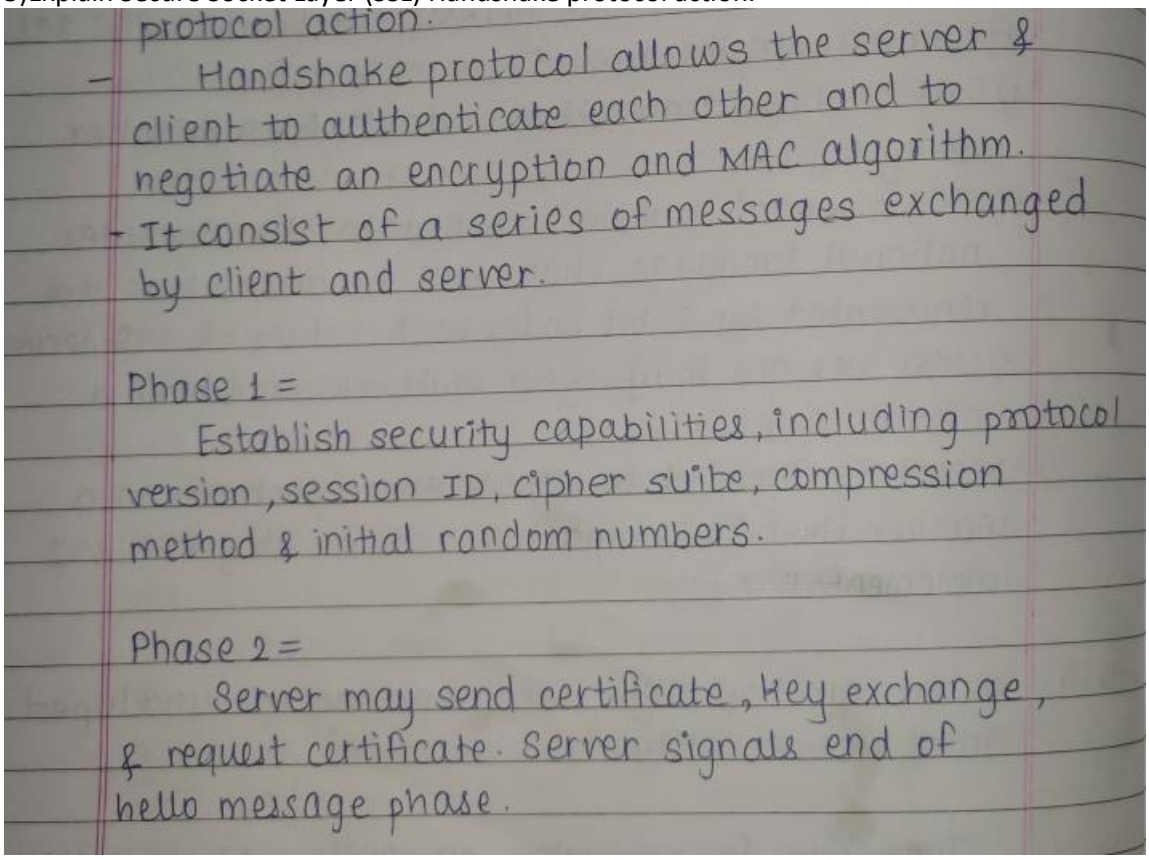
1) **Fragmentation:** Each upper-layer message is fragmented into blocks of 2^{14} bytes (16384 bytes) or less.
2) **Compression:** compression is optionally applied. Compression must be lossless.
3) **Compute MAC:** a shared secret key is used. The calculation is defined as:

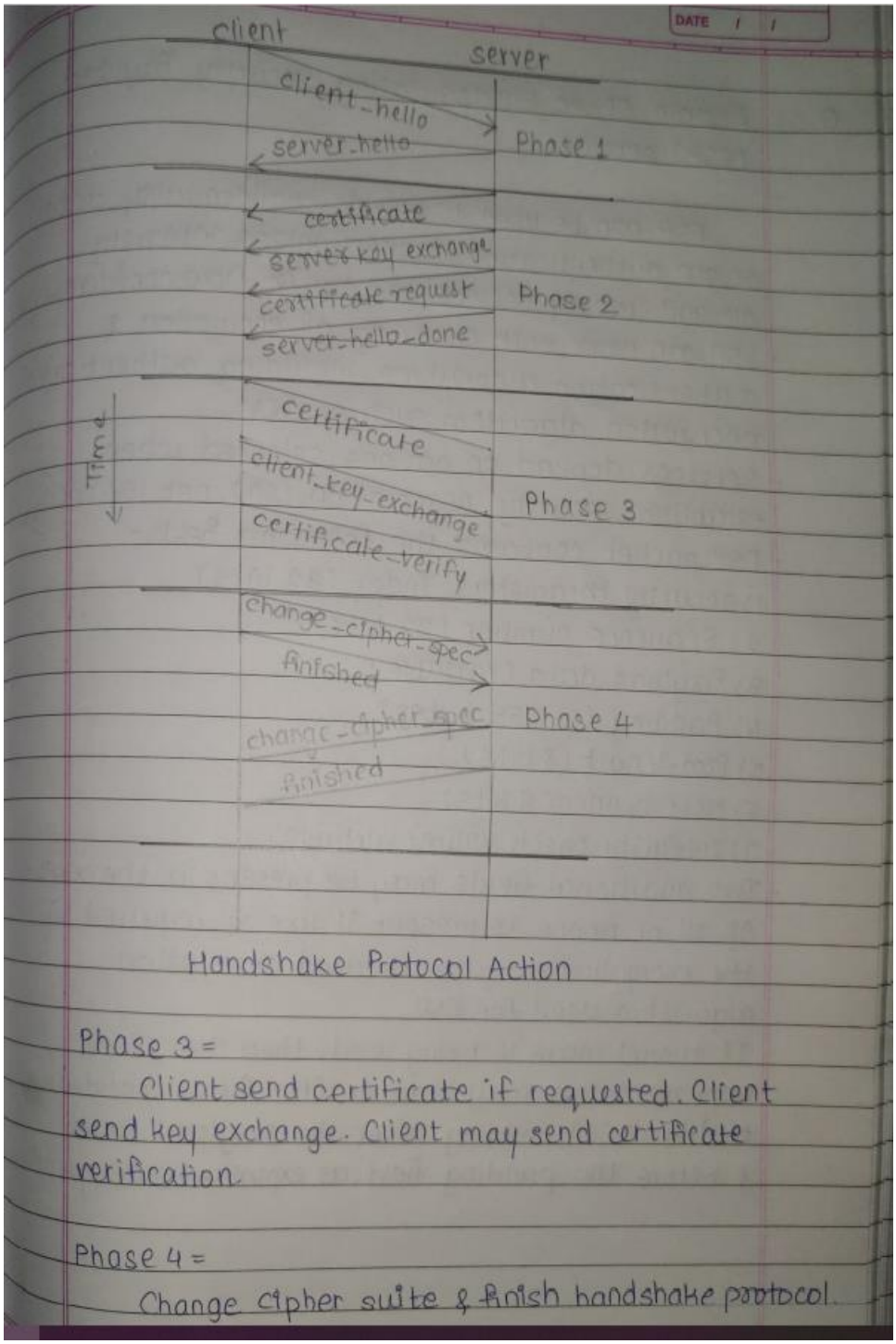
- hash(MAC_write_secret II pad_2 II hash(MAC_write_secret II pad_1 II seq_num II SSLCompressed.type II SSLCompressed.length II SSLCompressed.fragment))

Where,
II = concatenation
MAC- write_secret = shared secret key
Hash = cryptographic hash algorithm; either MD5 or SHA1
pad_1 = the byte 006 (0011 0110) repeated 48 times (384bits)
for MD5 and 40 times (320 bits) for SHA-l
pad_2 = the byte Ox5C (0101 1100) repeated 48 times for MD5 and 40 times for SHA-1
Seq_num = the sequence number for this message
SSLCompressed.type = the higher-level protocol used to process this fragment
SSLCompressed.length = the length of the compressed fragment SSLCompressed.fragment = the compressed fragment (if compression is not used, the plaintext fragment)

4) #What services are provided by the SSL Record Protocol?

5)Explain Secure Socket Layer (SSL) Handshake protocol action.





6) Explain Transport Layer Security (TLS).

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 1.

Message Authentication Code :

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

Defined as follows:

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

Where

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

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-l, block length = 512 bits)

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

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

Pseudorandom Function

- TLS makes use of a pseudo random function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation.

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.

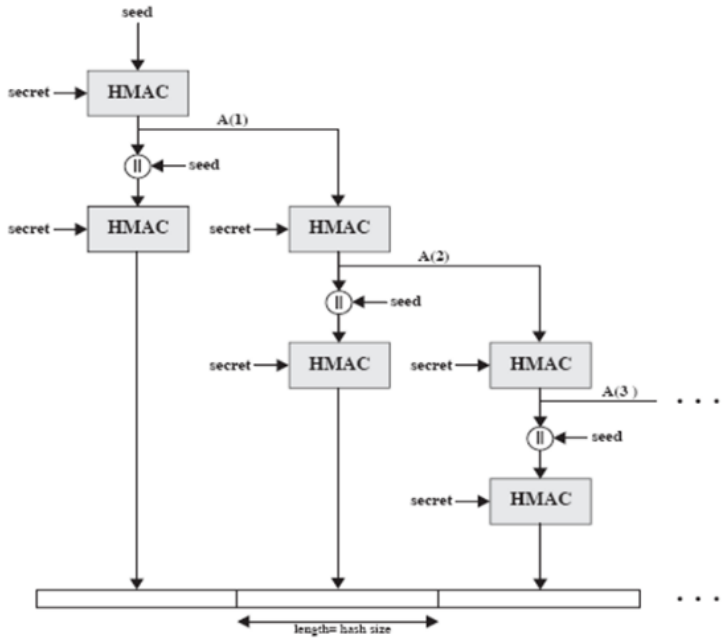


Figure 17.7 TLS Function P_hash (secret, seed)

7) What are the services provided by Pretty Good Privacy (PGP)?

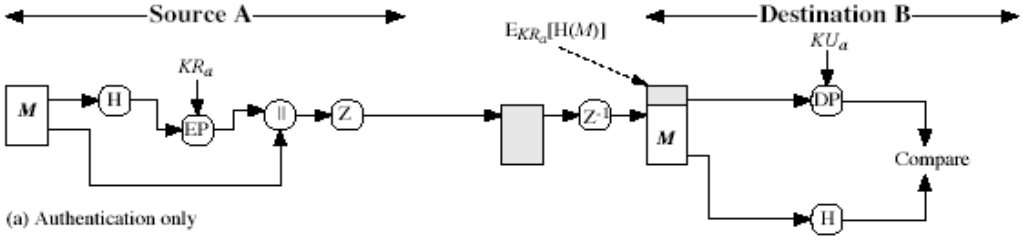
Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman 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.

8) What are the five principal services provided by PGP?

9) Explain Pretty Good Privacy operations.

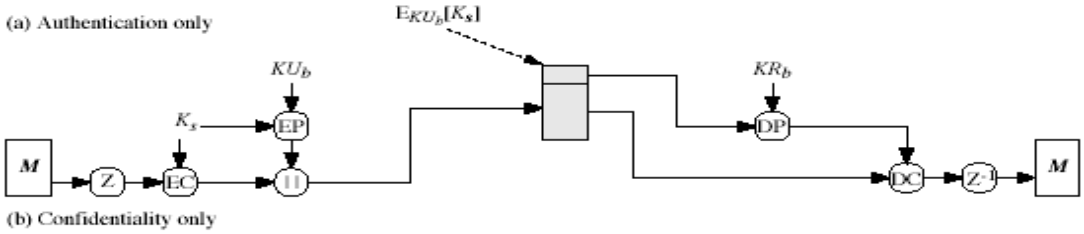
- Authentication:

1. sender creates a message
2. SHA-1 used to generate 160-bit hash code of message
3. hash code is encrypted with RSA using the sender's private key, and result is attached to message
4. receiver uses RSA or DSS with sender's public key to decrypt and recover hash code
5. receiver generates new hash code for message and compares with decrypted hash code, if match, message is accepted as authentic



(a) Authentication only

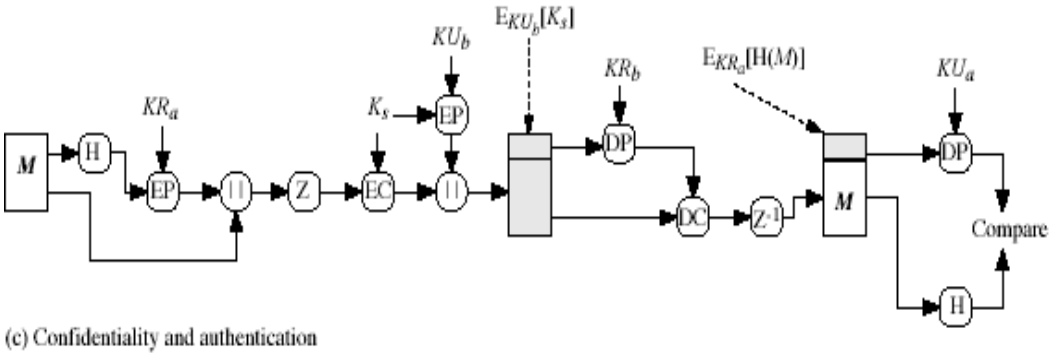
- Confidentiality:
    1. sender generates message and random 128-bit number to be used as session key for this message only
    2. message is encrypted, using CAST-128 / IDEA/3DES with session key
    3. session key is encrypted using RSA with recipient's public key, then attached to message
    4.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



(b) Confidentiality only

- Confidentiality & Authentication:

    uses both services on same message:
    1. create signature & attach to message
    2. encrypt both message & signature
    3. attach RSA encrypted session key



(c) Confidentiality and authentication

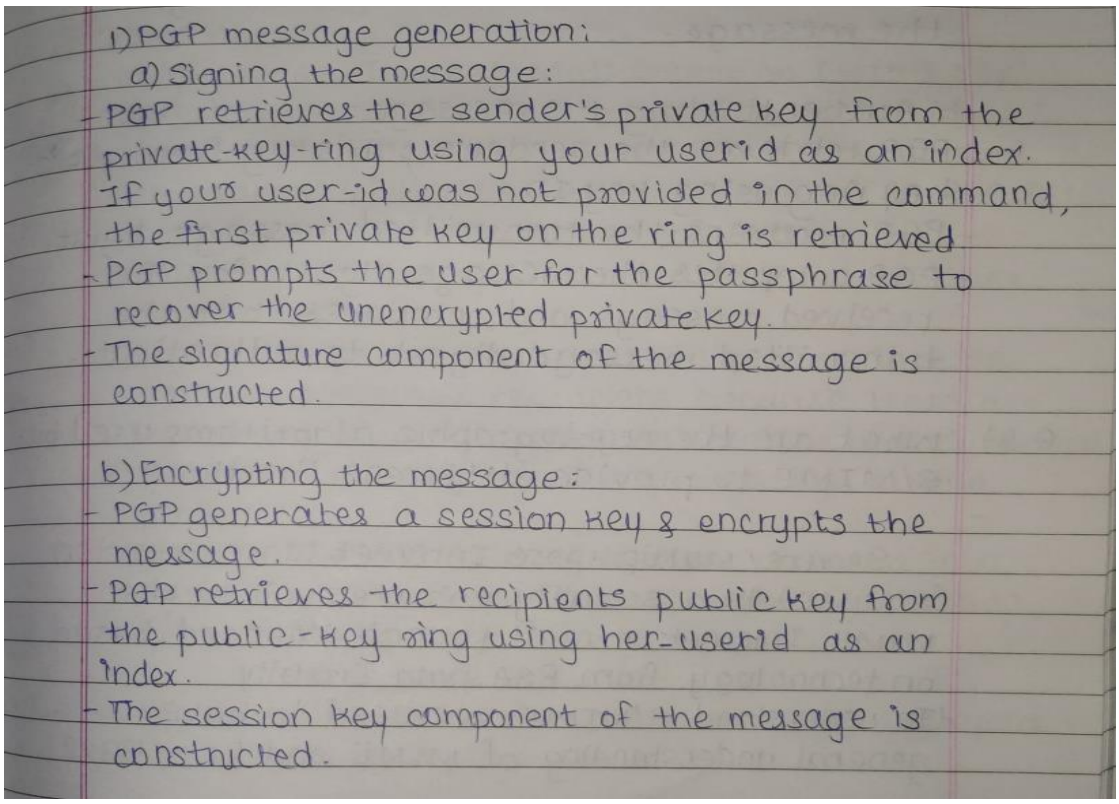
- Compression:
    1. by default PGP compresses message after signing but before encrypting
       -so can store uncompressed message & signature for later verification
       -& because compression is non deterministic
    2. uses ZIP compression algorithm

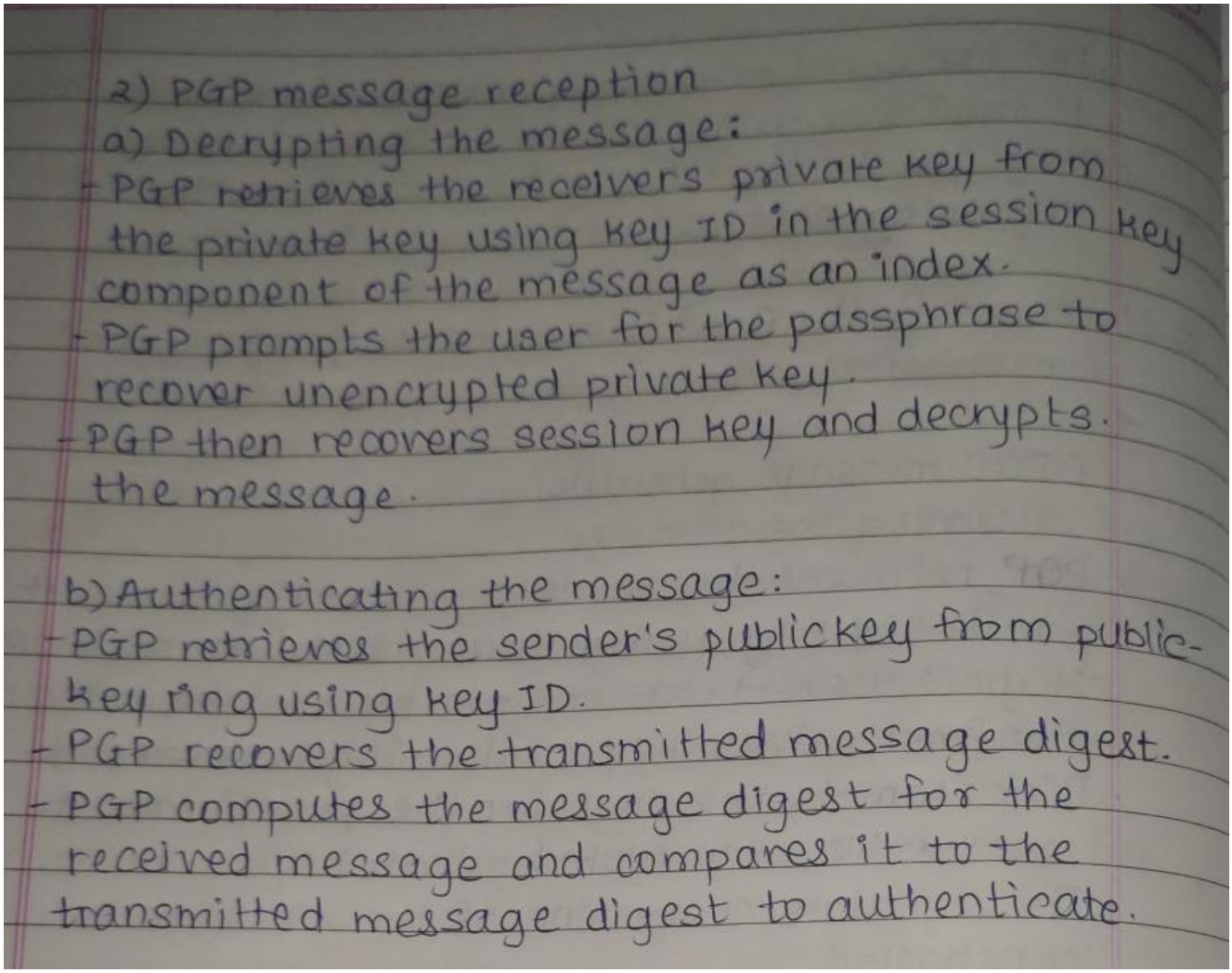
- Email Compatibility:
    1. when using PGP will have binary data to send (encrypted message etc)
    2. however email was designed only for text
    3. hence PGP must encode raw binary data into printable ASCII characters
    4. uses radix-64 algorithm
       -maps 3 bytes to 4 printable chars
   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

1. E-mail facilities often are restricted to a maximum message length.
2. Any longer message must be broken up into smaller segments, each of which is mailed separately.
3. PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail.
4. The segmentation is done after all of the other processing, including the radix-64 conversion.
5. At the receiving end, PGP must strip off all e-mail headers and reassemble the entire original block before performing the required steps at receiving end.

---

10)Explain Pretty Good Privacy message generation and reception process.





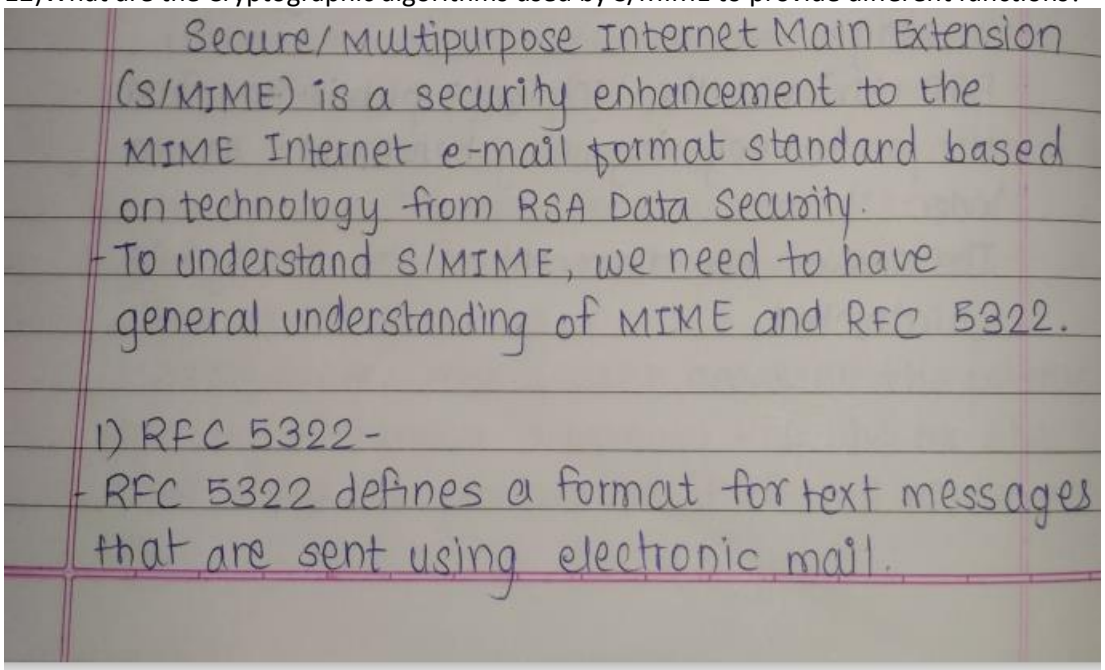
11) What are the functions provided by S/MIME?

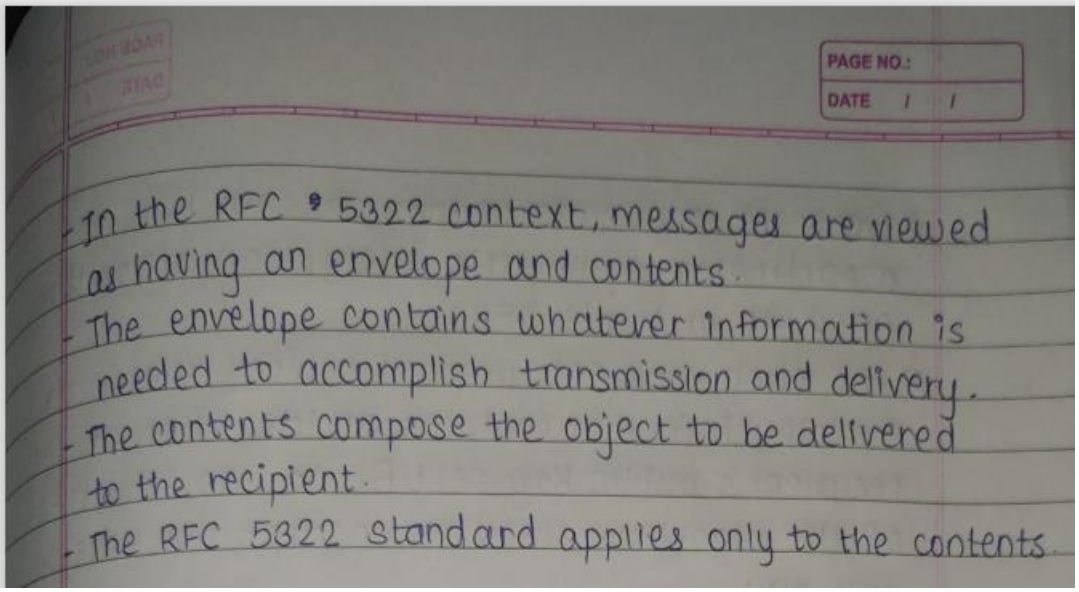
1. **Enveloped data:** This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.
2. **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. 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.

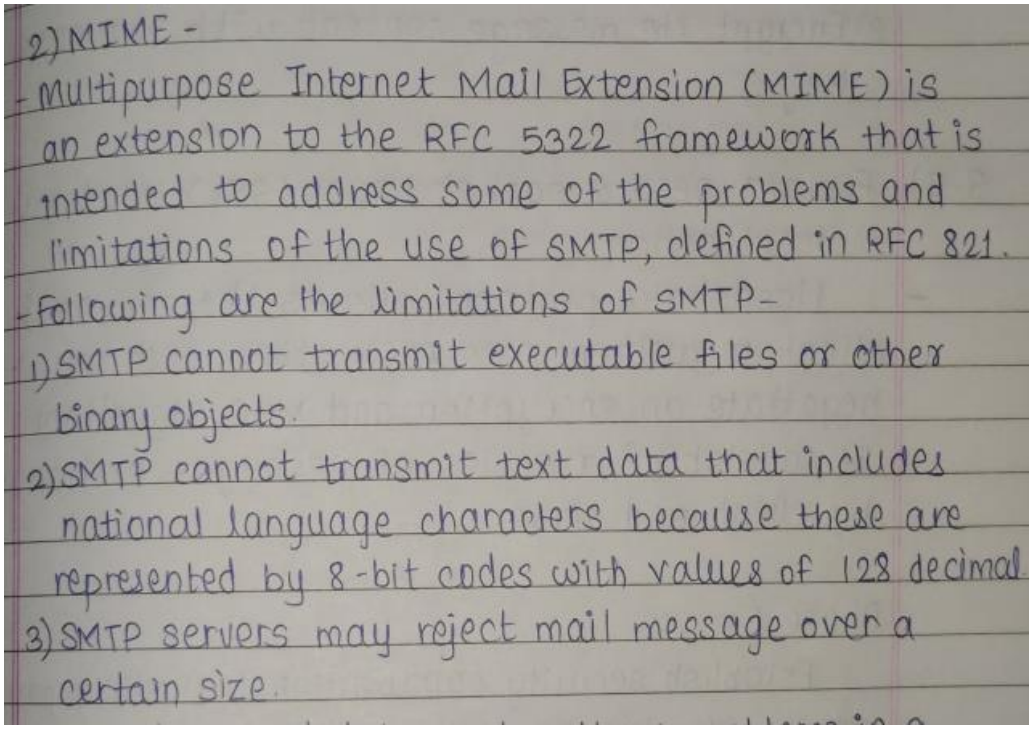
4. **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

---

12)What are the Cryptographic algorithms used by S/MIME to provide different functions?







---

13)What are the steps for preparing an enveloped data MIME entity?

The steps for preparing an enveloped Data MIME entity are as follows:

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or tripleDES).
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as **RecipientInfo** that contains an identifier of the recipient's public-key certificate, an identifier of the algorithm used to encrypt the session key, and the encrypted session key.
4. Encrypt the message content with the session key.

---

14) What are the steps for preparing signed data MIME entity?

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, or hash function, of the content to be signed.
3. Encrypt the message digest with the signer's private key.
4. Prepare a block known as **SignerInfo** 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.

---

15)Explain IPSec architecture.

- IPsec encompasses three functional areas: authentication, confidentiality, and key management
- specification is quite complex, with groups:
    a. Architecture
        i. RFC4301 *Security Architecture for Internet Protocol*
    b. Authentication Header (AH)
        i. RFC4302 *IP Authentication Header*
    c. Encapsulating Security Payload (ESP)
        i. RFC4303 *IP Encapsulating Security Payload (ESP)*

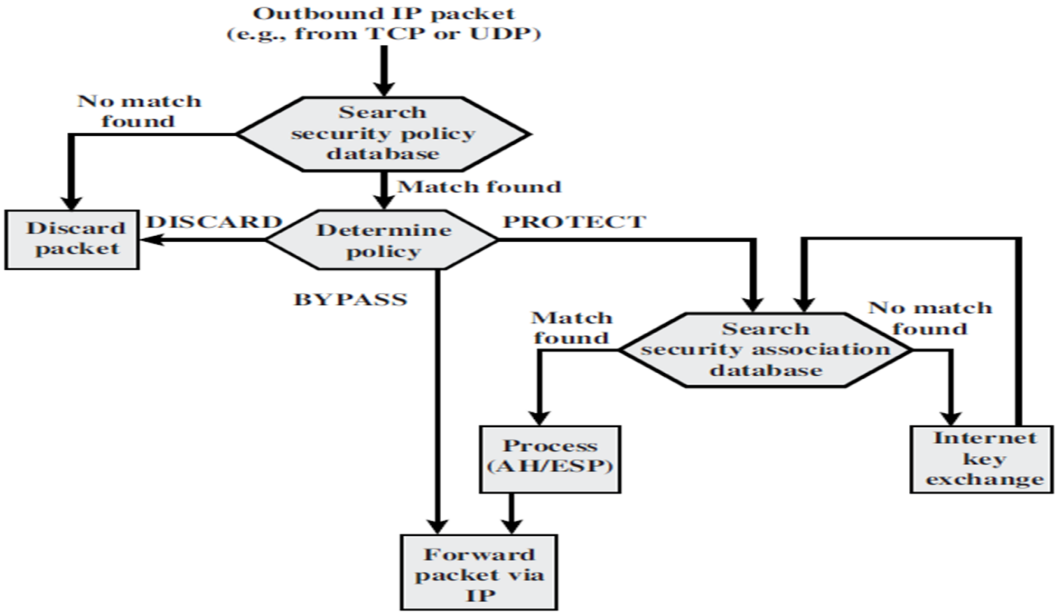
d. Internet Key Exchange (IKE)
    i. RFC4306 *Internet Key Exchange (IKEv2) Protocol*
e. Cryptographic algorithms
f. Other

16)Explain transport and tunnel modes of IPSec.

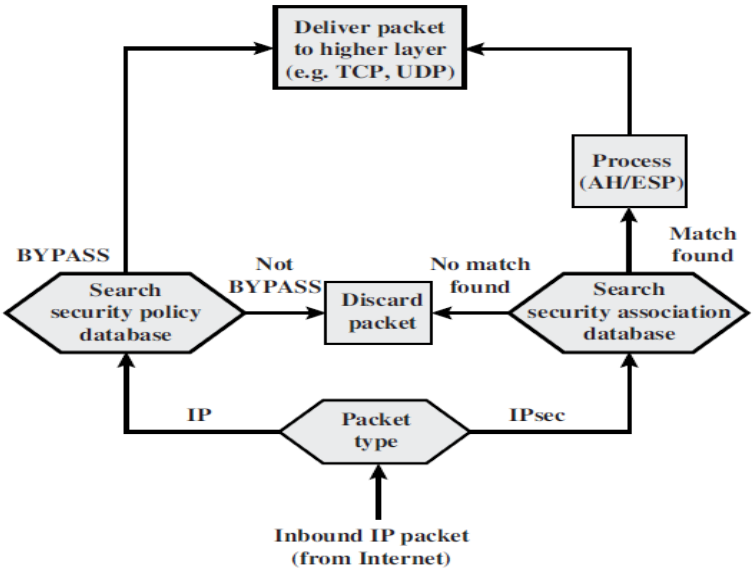
- Transport Mode
  - to encrypt & optionally authenticate IP data
  - can do traffic analysis but is not efficient
  - good for ESP host to host traffic
- Tunnel Mode
  - encrypts entire IP packet
  - add new header for next hop
  - no routers on way can examine inner IP header
  - good for VPNs, gateway to gateway security

	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.

17)Explain processing model for outbound and inbound IPSec packets.



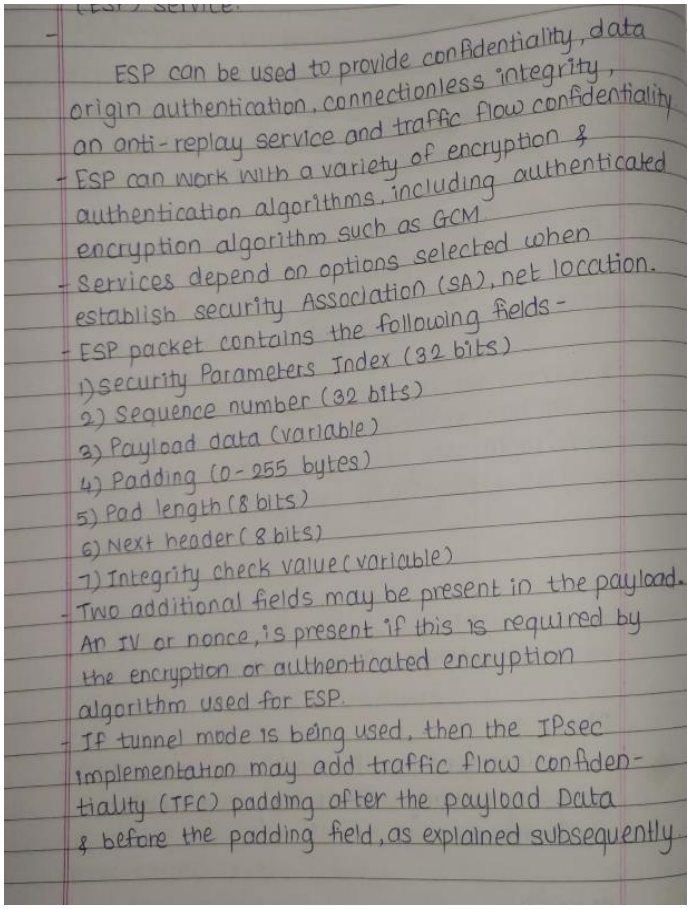
Processing Model for Outbound Packets



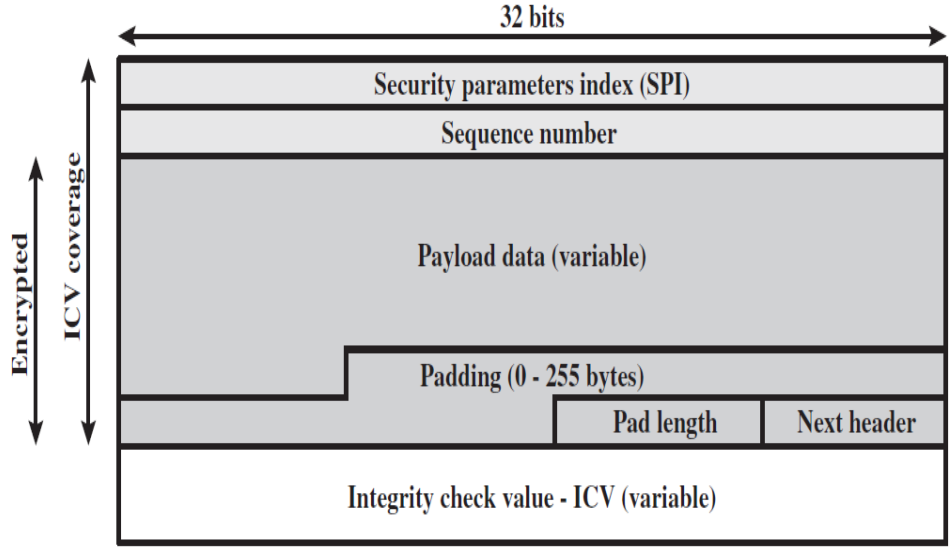
Processing Model for Inbound Packets

18)Explain IPSec Encapsulating Security Payload (ESP) service.

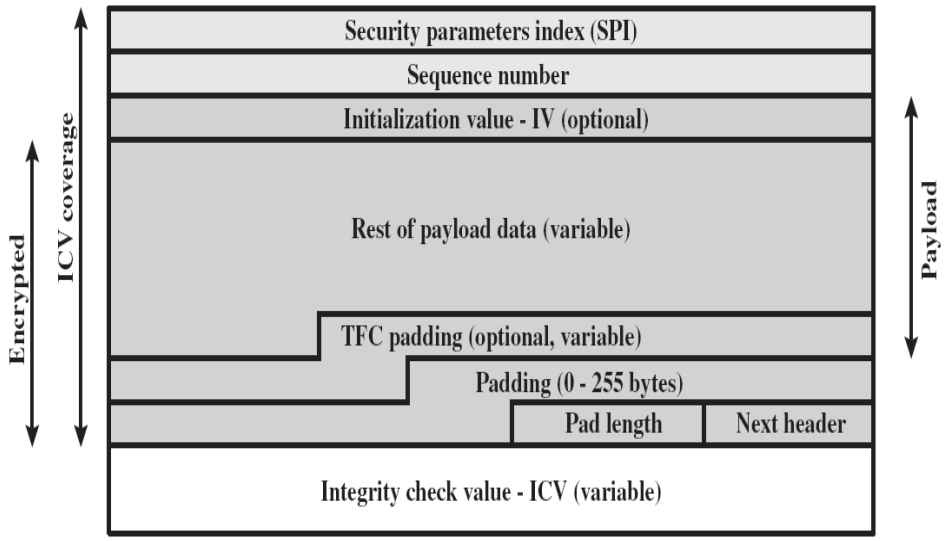
- provides message content confidentiality, data origin authentication, connectionless integrity, an anti-replay service, limited traffic flow confidentiality
- services depend on options selected when establish Security Association (SA), net location
- can use a variety of encryption & authentication algorithms

- 

19)Explain IPSec Encapsulating Security Payload (ESP) packet format..



(a) Top-level format of an ESP Packet



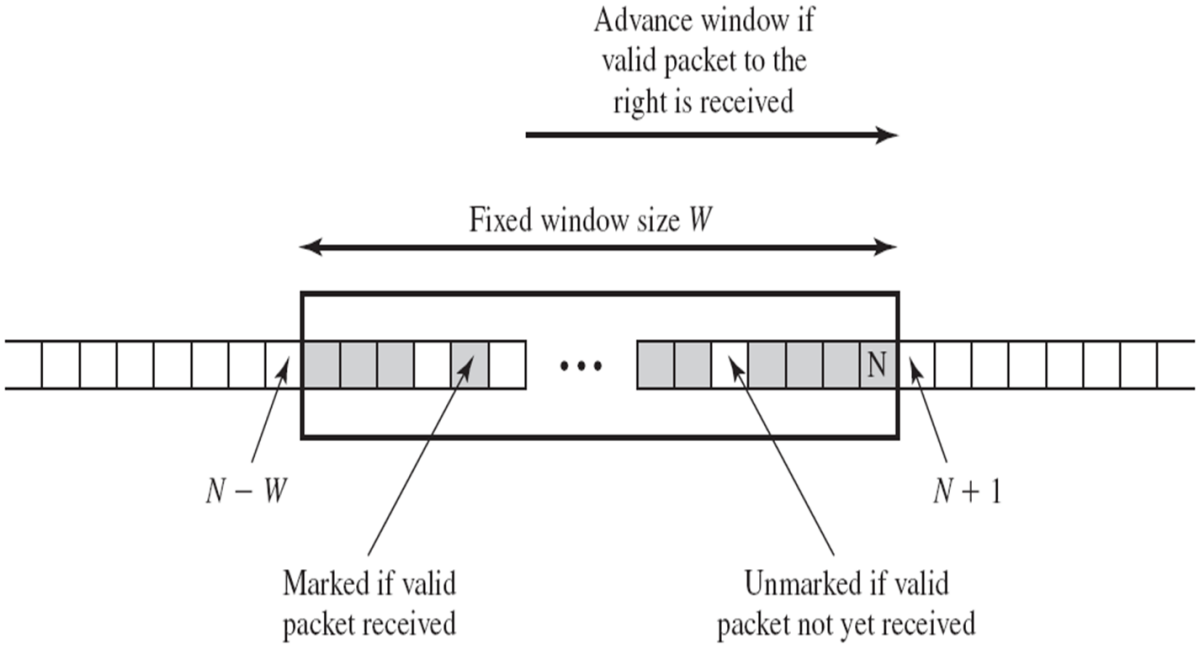
(b) Substructure of payload data

20)Explain IPSec Anti-replay mechanism with neat diagram.

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.



Anti-replay Mechanism