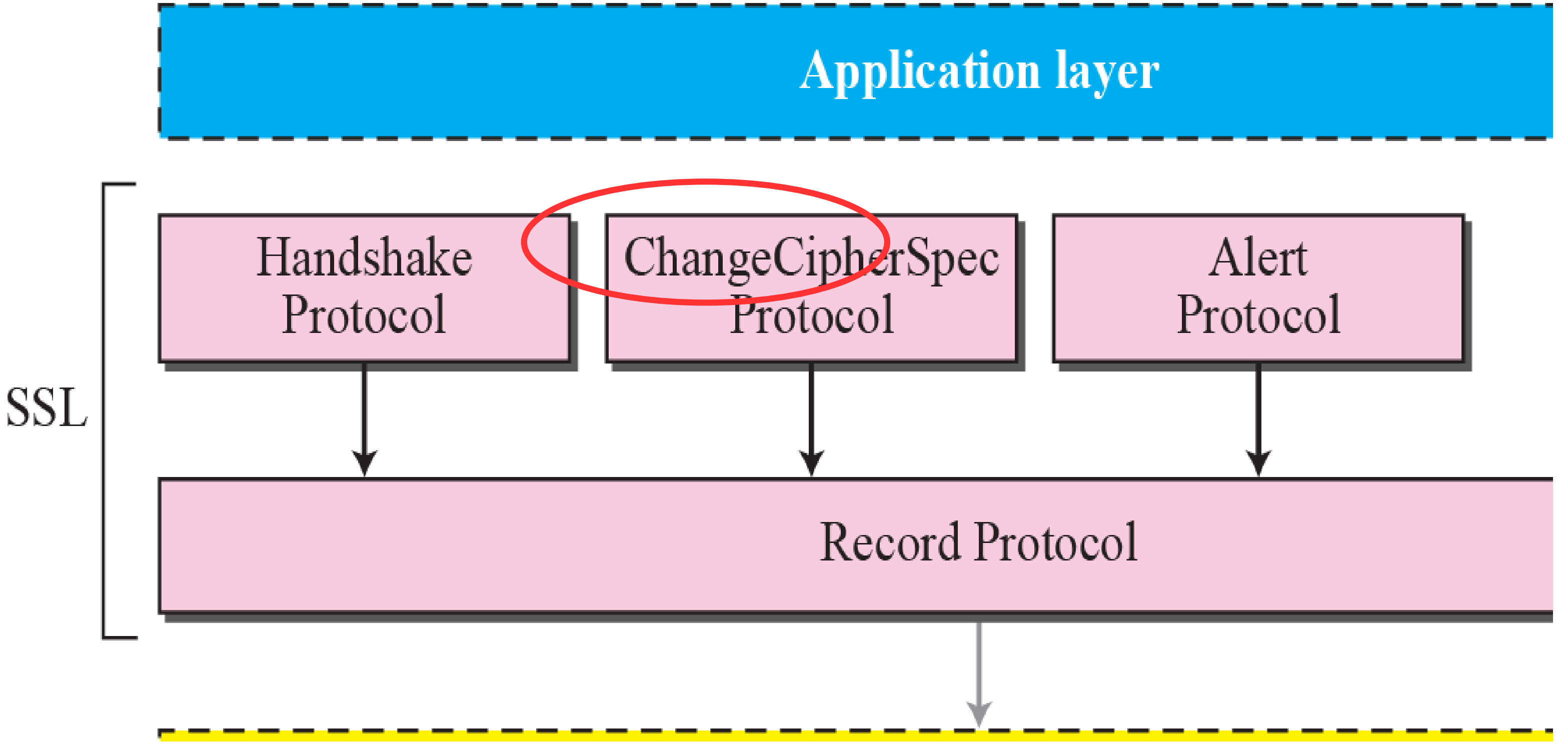
Module 5

Network Security and Applications

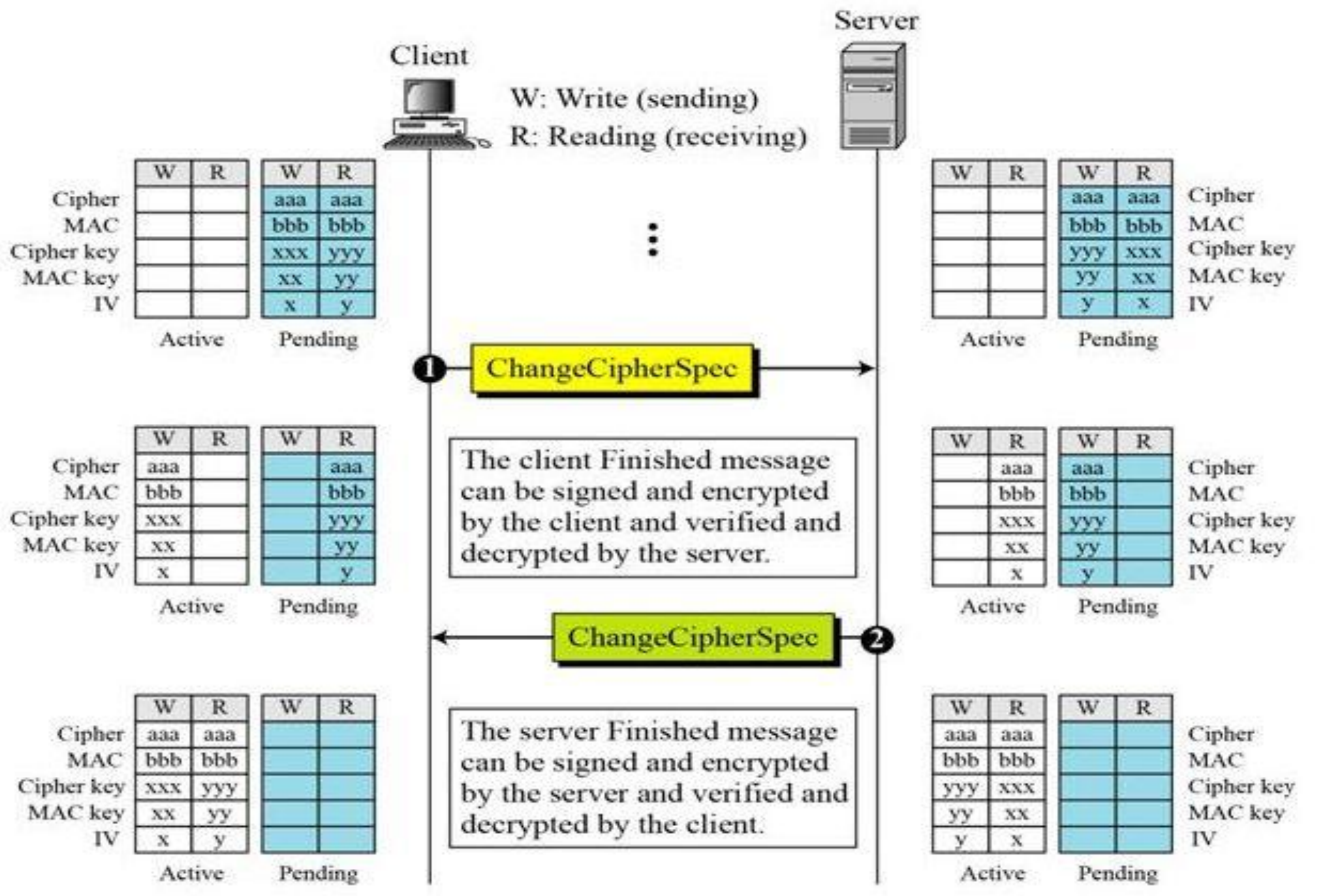
SSL protocols



SSL-ChangeCipherSpec Protocol

- The two parties can’t use the parameter secrets until and unless they have sent or received a special message.
- This special message is ChangeCipherSpec message which is exchanged during Handshake Protocol Phase-IV and defined in ChangeCipherSpec Protocol
- The main reason behind this is client and server needs to have two states:
  - **Pending state:** It keeps track of the parameters and secrets
  - **Active state:**It holds the parameters and secrets used by Record Protocol to sign/verify or encrypt/decrypt messages
- Each state holds two sets of values:
  - Read
  - Write

SSL-ChangeCipherSpec Protocol



SSL-ChangeCipherSpec Protocol

Step-1

- The client sends a ChangeCipherSpec message to the server
- After that it moves the write parameters from pending to active state
- The client can now use these parameters to sign or encrypt outbound messages

Step-2

- The server receives the message and moves the read parameters from pending to active state.
- The server can verify and decrypt the inbound messages

SSL-ChangeCipherSpec Protocol

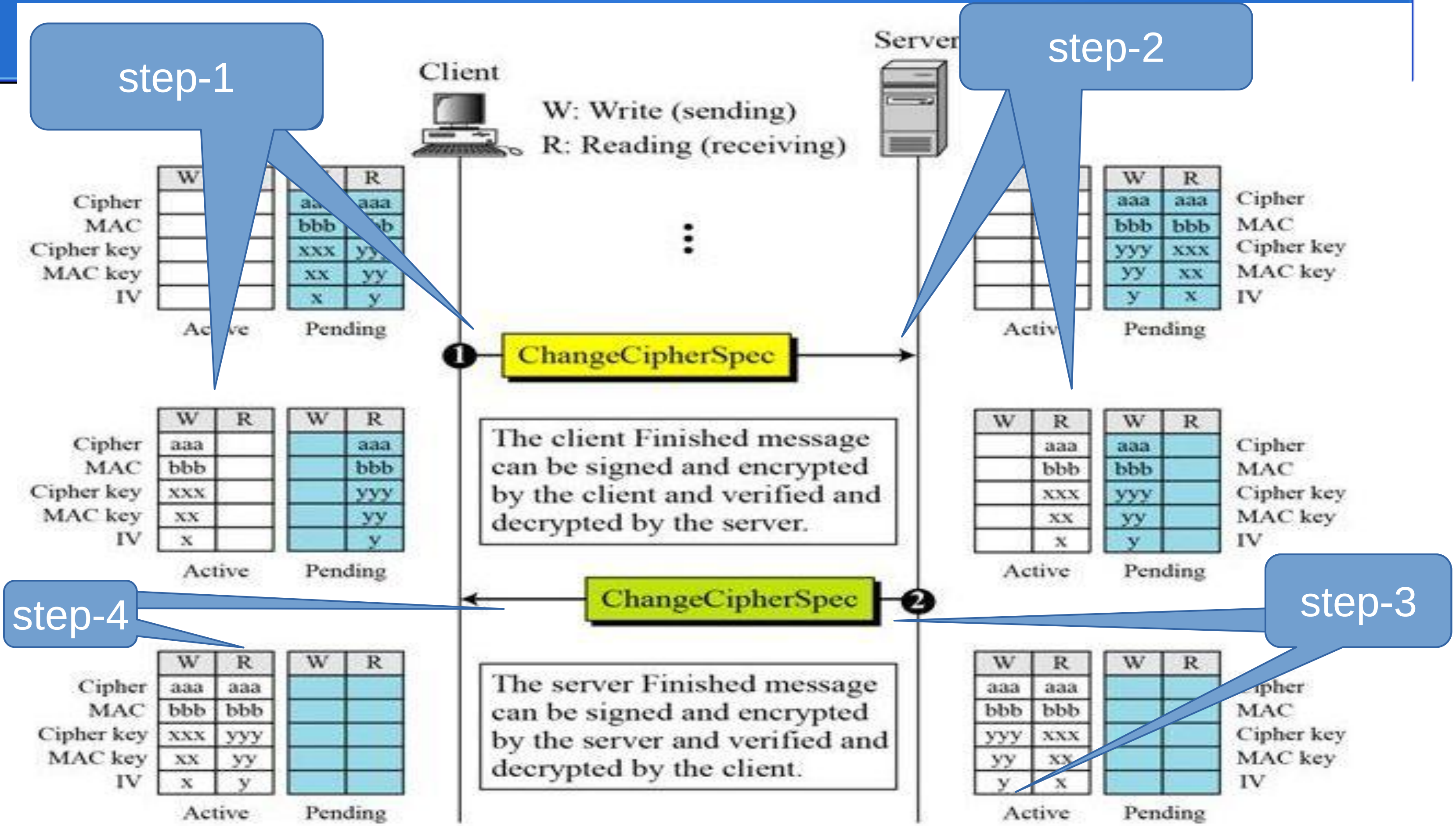
Step-3

- The server sends a ChangeCipherSpec message after the client has sent a Finished message.
- After that it moves the write parameters from pending to active state
- The server can now use these parameters to sign or encrypt outbound messages.

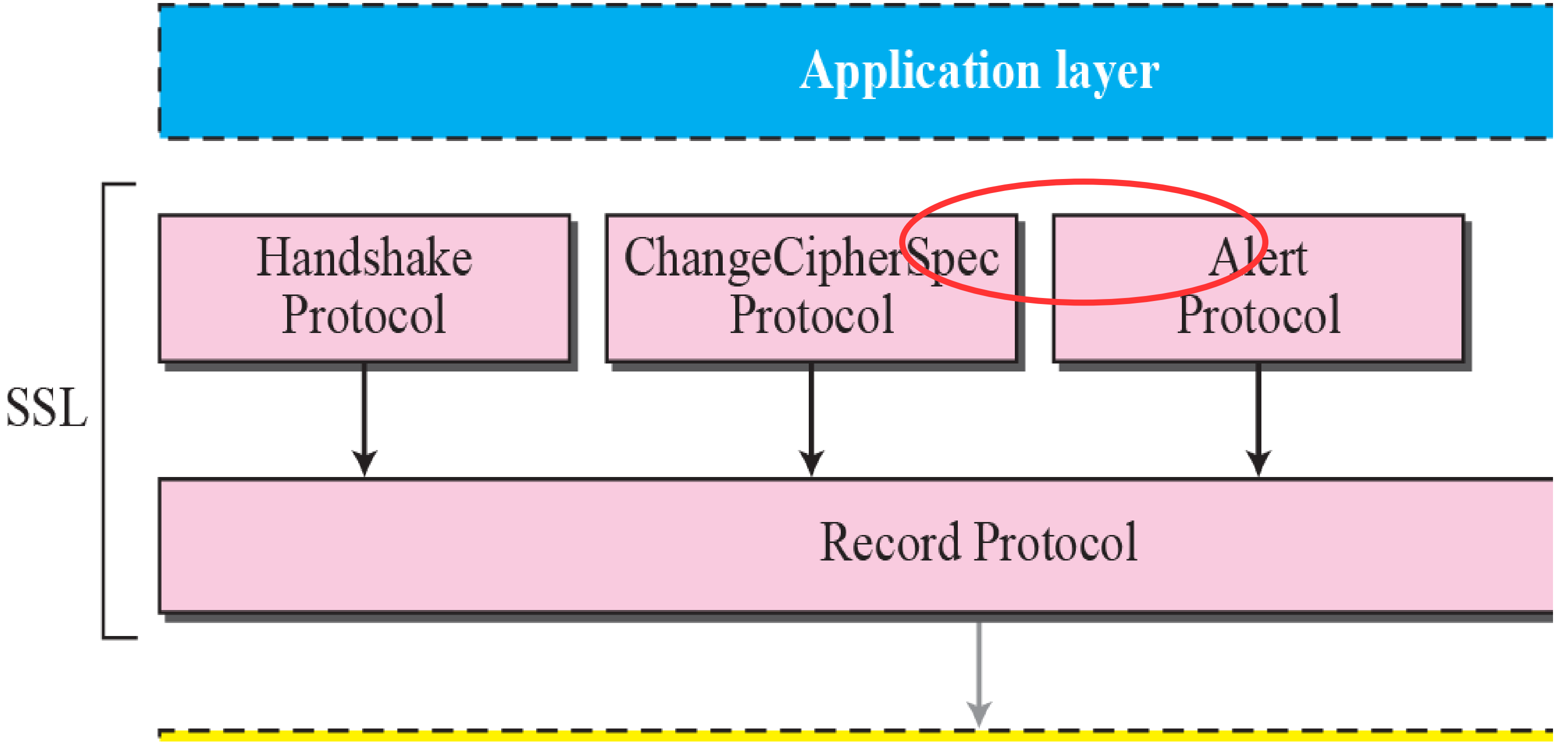
Step-4

- The client receives this message and moves the read parameters from pending to active state.
- Now the client can verify and decrypt the messages

SSL-ChangeCipherSpec Protocol



SSL protocols



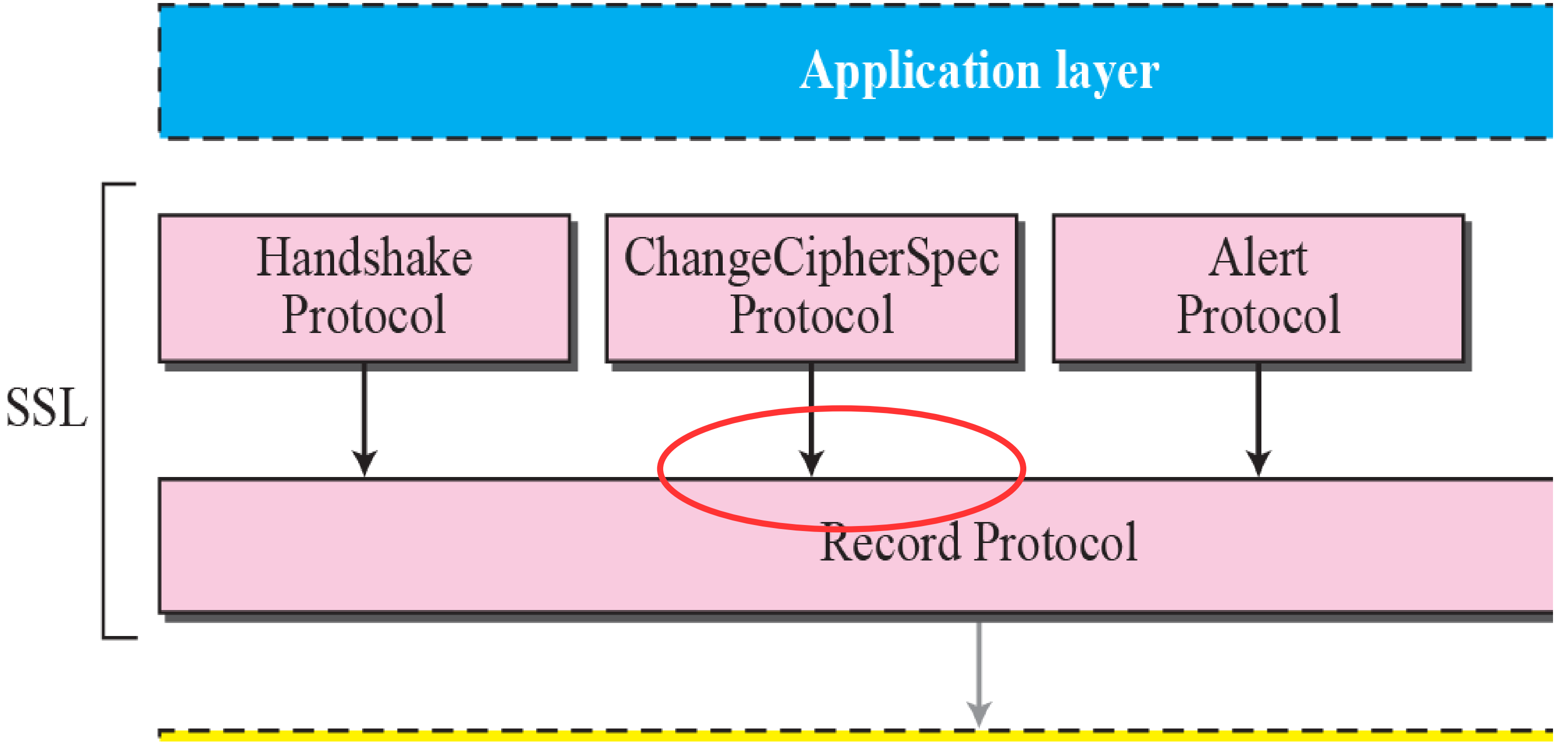
SSL-Alert Protocol

- SSL uses the Alert Protocol for reporting errors and abnormal conditions
- It has only one message type i.e. the ALERT message that describes the problem and its level (warning or fatal)

Table 17.4 *Alerts defined for SSL*

Value	Description	Meaning
0	*CloseNotify*	Sender will not send any more messages.
10	*UnexpectedMessage*	An inappropriate message received.
20	*BadRecordMAC*	An incorrect MAC received.
30	*DecompressionFailure*	Unable to decompress appropriately.
40	*HandshakeFailure*	Sender unable to finalize the handshake.
41	*NoCertificate*	Client has no certificate to send.
42	*BadCertificate*	Received certificate corrupted.
43	*UnsupportedCertificate*	Type of received certificate is not supported.
44	*CertificateRevoked*	Signer has revoked the certificate.
45	*CertificateExpired*	Certificate expired.
46	*CertificateUnknown*	Certificate unknown.
47	*IllegalParameter*	An out-of-range or inconsistent field.

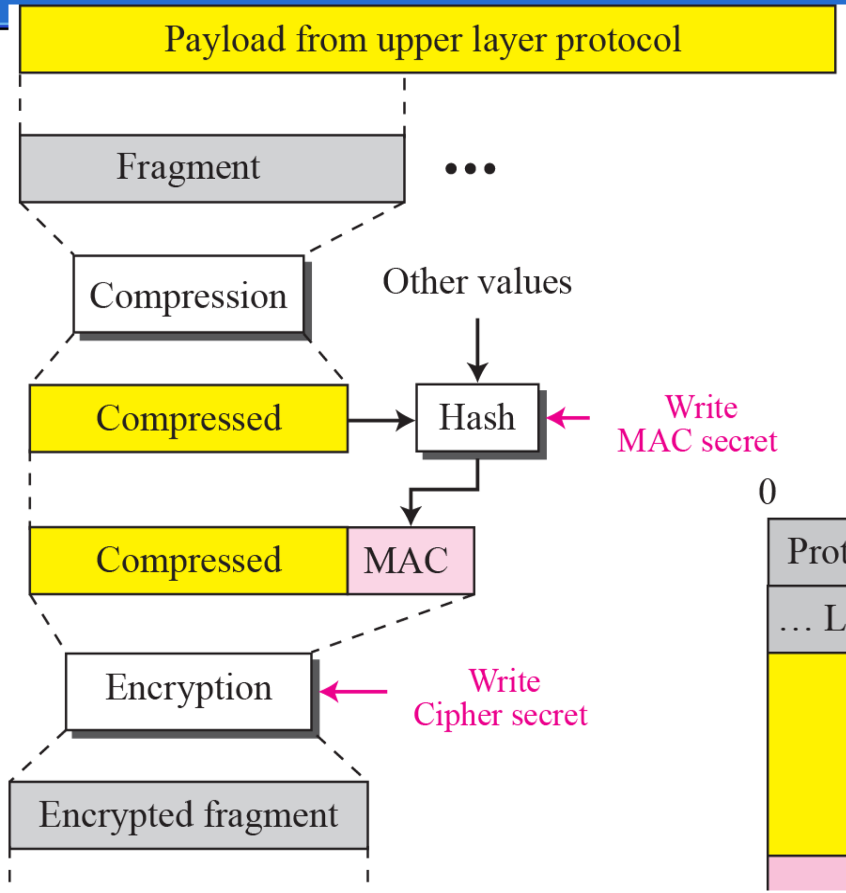
SSL protocols



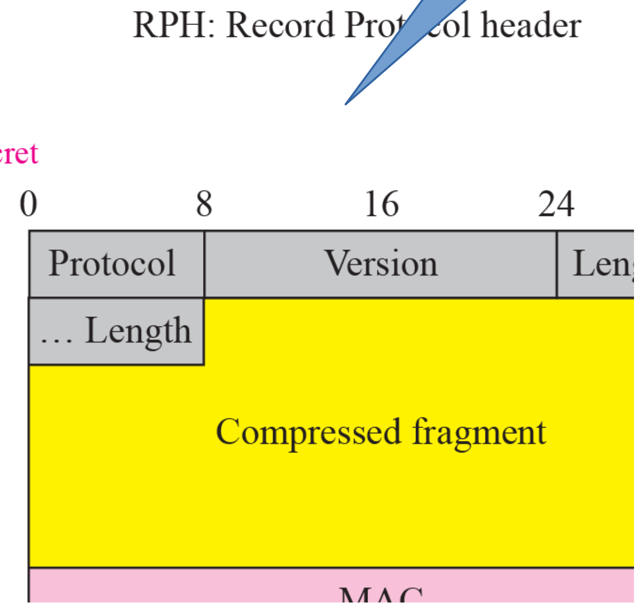
SSL-Record Protocol

- The Record protocol carries message from upper layer protocols
- The message is fragmented and optionally compressed.
- A MAC is added to the compressed message using the negotiated hash algorithm
- The compressed fragment and the MAC are encrypted using the negotiated encryption algorithm
- The SSL header is added to the encrypted message

SSL-Record Protocol



SSL Record Format



SSL-Record Protocol

Client Side

- Fragmentation:- A message from application layer is fragmented into blocks of 2^{14} bytes where the last block is less than this size

Receiver's side

- Combination:- The fragments are combined to make a replica of the original message

SSL-Record Protocol

Client Side

- Compression:-
  - All application layer fragments are compressed by the compression method negotiated during handshaking.
  - The size of the fragments must not exceed 1024 bytes

Receiver's side

- Decompression:-
  - The compressed fragment is decompressed to create a replica of the original
  - If the size of decompressed fragment exceeds 2^{14} then a fatal decompression Alert message is issued

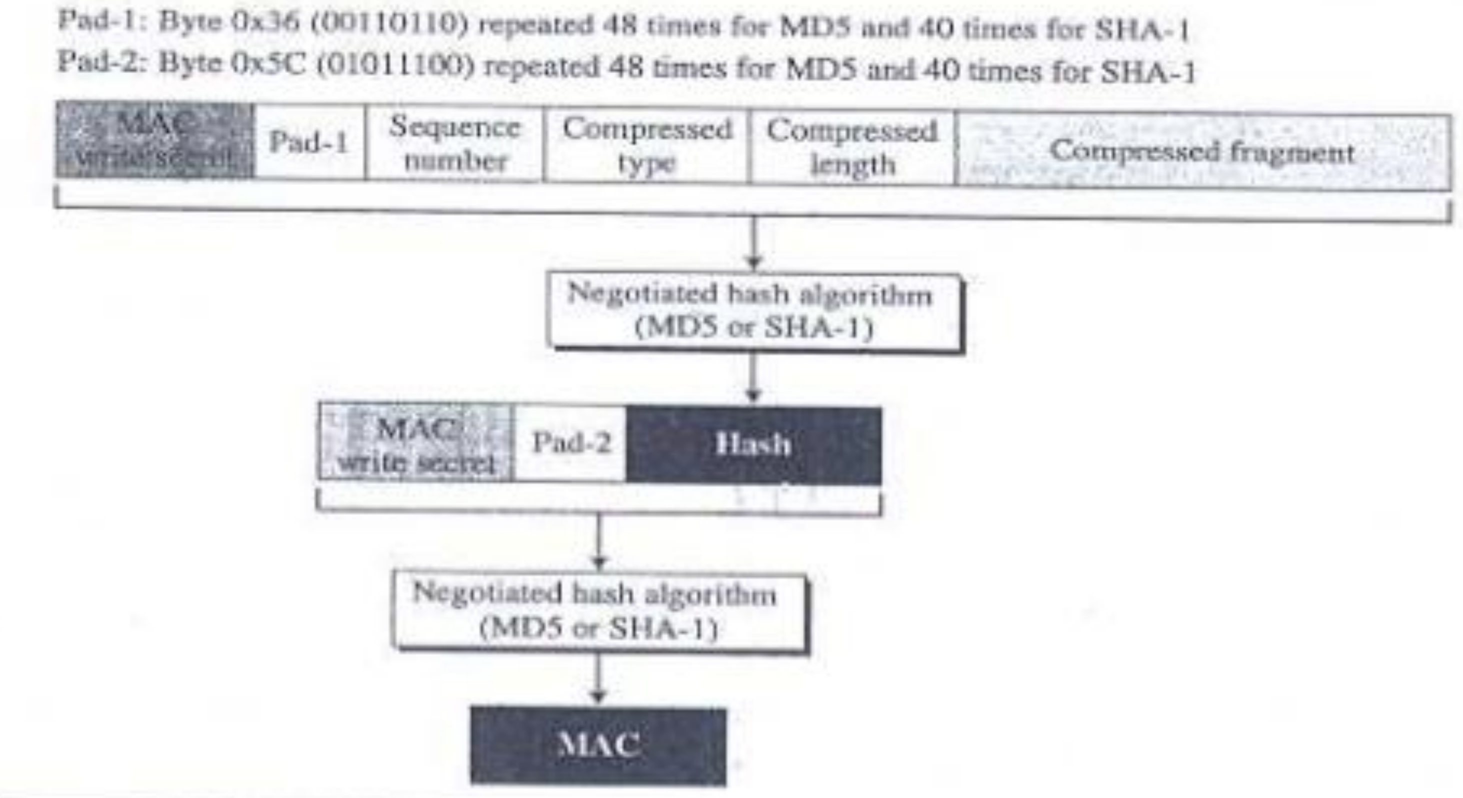
SSL-Record Protocol

Client Side

- Signing
  - The authentication method defined during the handshake protocol creates a signature
  - This signature is known as the MAC (Message Authentication Code)

SSL-Record Protocol

Figure 17.22 *Calculation of MAC*



SSL-Record Protocol

Receiver's Side

- Verifying
  - It is done by calculating a new hash and comparing it with the received hash

SSL-Record Protocol

<u>Sender's Side</u>

- Encryption
  - The compressed fragment and the hash are encrypted using the cipher write secret

<u>Receiver's Side</u>

- Decryption
  - The received message is decrypted using the cipher read secret

SSL-Record Protocol

Sender's Side

- Framing
  - After encryption, the Record Protocol Header is added

Receiver's Side

  - Deframing
    - Before decryption,the header is removed

SSL

How SSL provides authentication?

- For server authentication, the client uses the server's public key to encrypt the data that is used to compute the secret key.
- The server can generate this secret key only if it can decrypt that data with the correct private key.
- For client authentication, the server uses the public key in the client certificate to decrypt the data which the client sends during handshake.
- The exchange of finished messages that are encrypted with the secret key confirms that authentication is complete.
- If any of the authentication steps fail, the handshake fails and the session terminates.
- The exchange of digital certificates during the SSL handshake is part of the authentication process.

SSL

- The certificates required are as follows:
  - A certificate authority(CA), X issues the certificate to the SSL client, and certificate authority (CA), Y issues the certificate to the SSL server:

  - . For both server and client authentication,
    <u>Server needs:</u>
    - The personal certificate issued to the server by CA Y
    - The server's private key
    - The CA certificate for X

    <u>Client needs:</u>
    - The personal certificate issued to the client by CA X
    - The client's private key
    - The CA certificate for Y

SSL

How SSL provides confidentiality?

- SSL uses a combination of symmetric and asymmetric encryption to ensure message privacy.
- During the SSL handshake, client and server agree an encryption algorithm and a shared secret key to be used for one session only.
- All messages transmitted between the client and server are encrypted using that algorithm and key, ensuring that the message remains private even if it is intercepted.
- The SSL Record protocol performs the encryption of SSL payloads using the shared secret key generated by Handshake Protocol
- Since, SSL uses asymmetric encryption when transporting the shared secret key, there is no key distribution problem

SSL

How SSL provides integrity?

- SSL provides data integrity by calculating Machine Authentication Code
- The Handshake Protocol defines a shared secret key that is used to form a MAC
- Calculation of MAC is done by the Record Protocol
- Use of SSL does ensure data integrity, provided that the CipherSpec uses a hash algorithm
- If data integrity is a concern,we should avoid choosing a CipherSpec whose hash algorithm is listed as "NULL".