Module 5

Network Security and Applications

Internet Security Protocols

- SSL
- IPSEC
- PGP

Security Sockets Layer(SSL)

- Security at transport layer is provided by SSL
- Transport layer security provides end-to-end secirity services for application that uses TCP protocol
- The idea is to provide security services for transaction on the Internet

Security Sockets Layer(SSL)

- Example of an Internet transaction
  - When a customer does online shopping, the following security services are desired:-
    - Customer needs to be sure that the server belongs to the actual vendor (authentication)
    - Customer and vendor need to be sure that contents of message are not modified during transmission (message integrity)
    - Customer and vendor needs to be sure that an attacker does not intercept sensitive information such as credit card information (confidentiality)

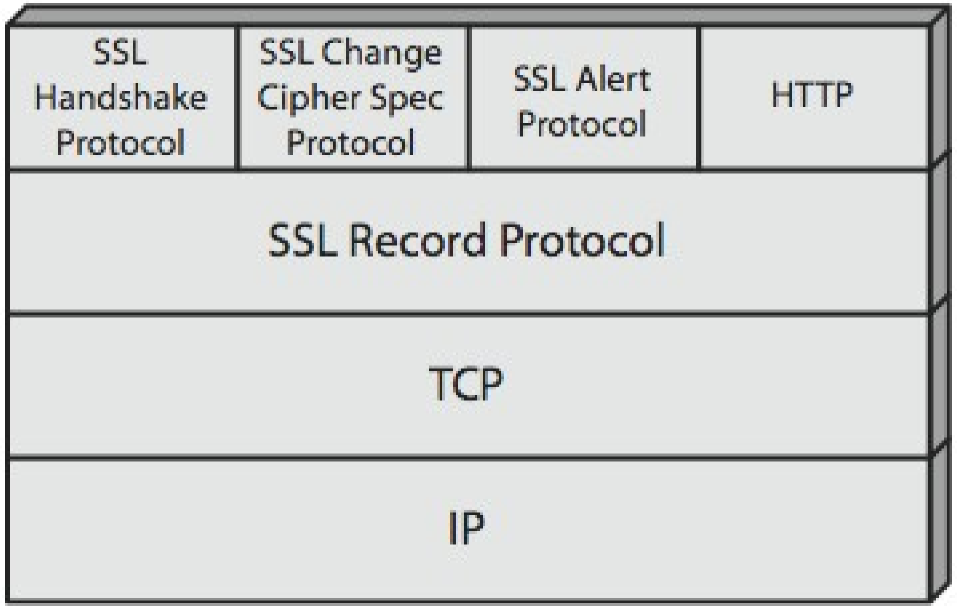
Secure Sockets Layer(SSL)

- Goal of SSL protocol is to provide:-
  - Server and client authentication
  - Data confidentiality
  - Data integrity

- SSL is designed to provide security and compression services to data generated from application layer

- Data recieved from application layer is compressed(if required), signed and encrypted

SSL architecture



SSL services

- Fragmentation
- Compression
- Message Integrity
- Confidentiality
- Framing

SSL services

- **Fragmentation**- SSL divides the data into blocks of 2^{14} bytes or less

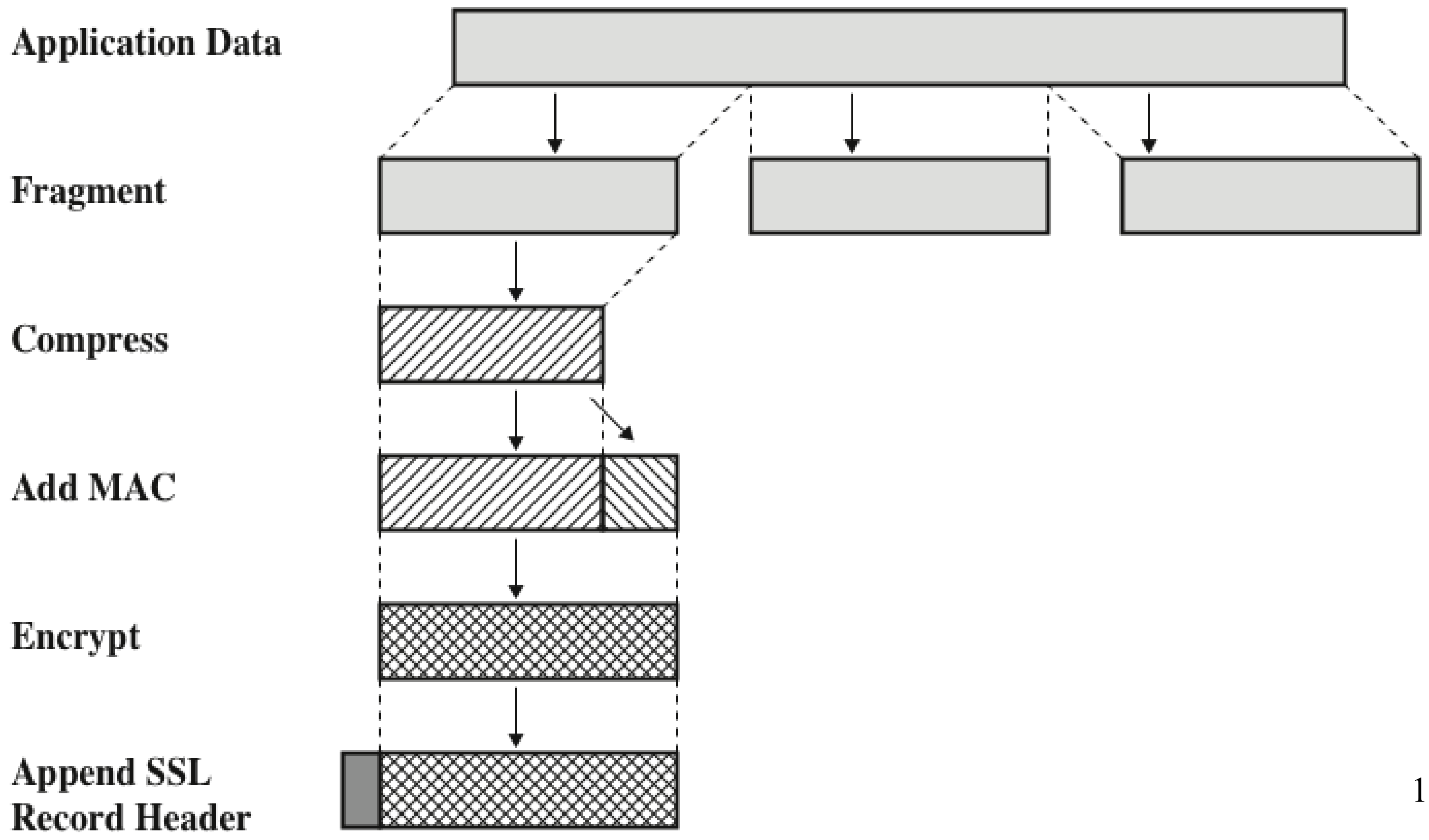
- **Compression(optional)**-Each fragment of data is compressed using one of the lossless compression methods negotiated between client and server.
  - Lossless compression methods can be Run Length Encoding(RLE) or Huffman Coding

  - **Message Integrity**-To preserve integrity of data, SSL uses keyed-hash function to create a MAC (Message Authentication Code)

SSL services

- **Confidentiality**-To provide confidentiality, the original data and the MAC are encrypted using symmetry key cryptography
- **Framing**-A header is added to the encrypted payload. The payload is then passed to a reliable transport layer protocol.
  - The header identifies the source and destination of the packet, while the actual data is referred to as the payload.

SSL services



SSL session and connection

- SSL session
  - an association between client & server
  - created by the Handshake Protocol
  - defines 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

SSL session state

- A session is defined by a session state, a set of parameters established between the server and the client
- Parameters:
  - Session ID
  - Peer Certificate
  - Compression method
  - Cipher suite
  - Master secret
  - Is resumable

SSL session state

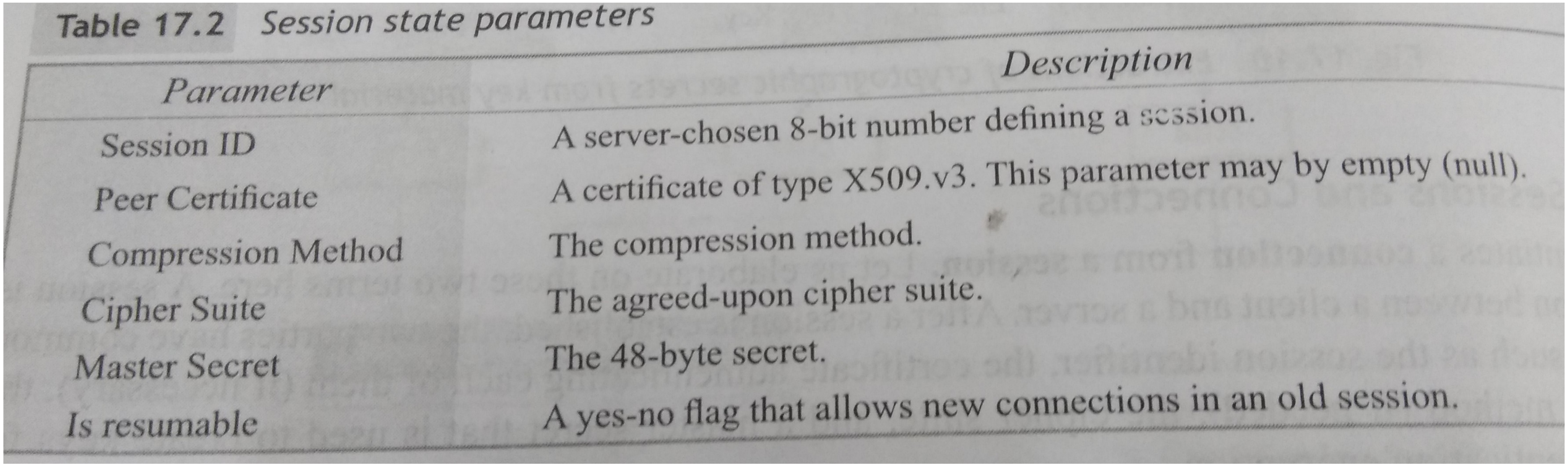


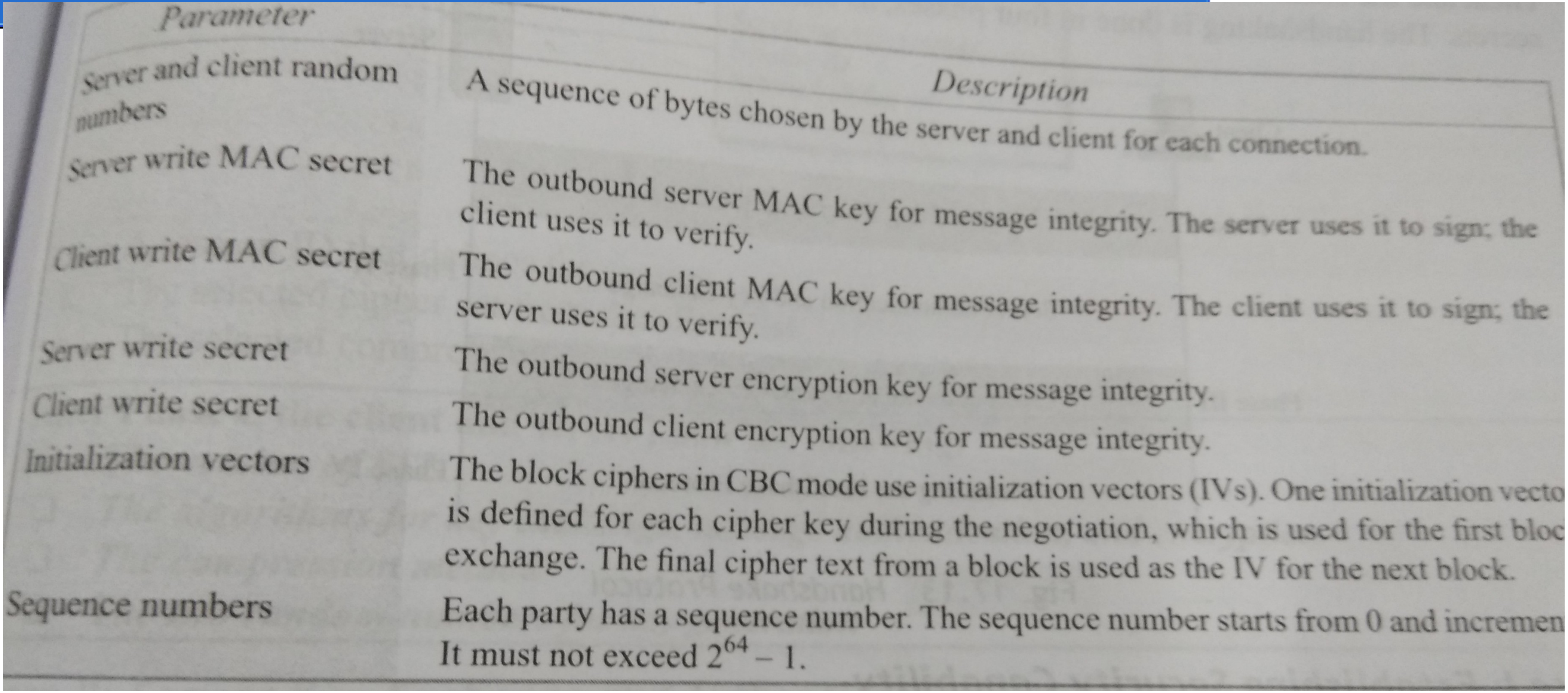
Table 17.2 *Session state parameters*

Parameter	*Description*
Session ID	A server-chosen 8-bit number defining a session.
Peer Certificate	A certificate of type X509.v3. This parameter may by empty (null).
Compression Method	The compression method.
Cipher Suite	The agreed-upon cipher suite.
Master Secret	The 48-byte secret.
Is resumable	A yes-no flag that allows new connections in an old session.

SSL connection states

- A connection is defined by a connection state, a set of parameters established between two peers.
- Parameters:
  - Server and client random numbers
  - Server write MAC secret
  - Client write MAC secret
  - Server write secret
  - Client write secret
  - Initialization vectors
  - Sequence numbers

SSL connection states



Parameter	Description
Server and client random numbers	A sequence of bytes chosen by the server and client for each connection.
Server write MAC secret	The outbound server MAC key for message integrity. The server uses it to sign; the client uses it to verify.
Client write MAC secret	The outbound client MAC key for message integrity. The client uses it to sign; the server uses it to verify.
Server write secret	The outbound server encryption key for message integrity.
Client write secret	The outbound client encryption key for message integrity.
Initialization vectors	The block ciphers in CBC mode use initialization vectors (IVs). One initialization vecto is defined for each cipher key during the negotiation, which is used for the first bloc exchange. The final cipher text from a block is used as the IV for the next block.
Sequence numbers	Each party has a sequence number. The sequence number starts from 0 and incremen It must not exceed $2^{64} - 1$.

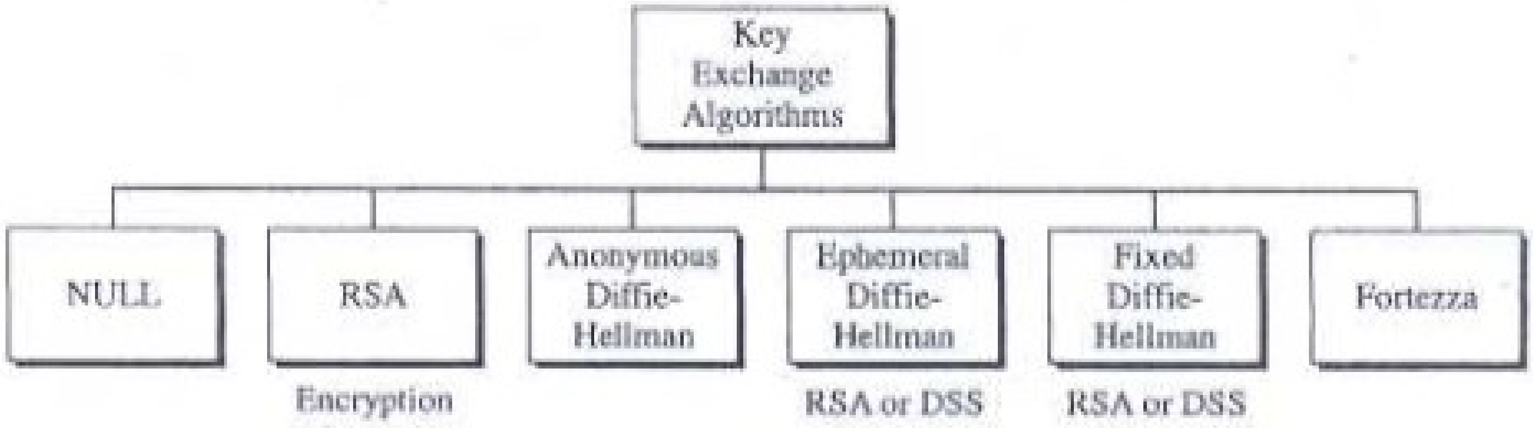
SSL

- To exchange an authenticated and confidential message, the client and the server each need six cryptographic secrets:
  - Client authentication key
  - Server authentication key
  - Client encryption key
  - Server encryption key
  - Client initiation vector
  - Server initiation vector

- To create this secrets, one pre-master secret must be established between the two parties

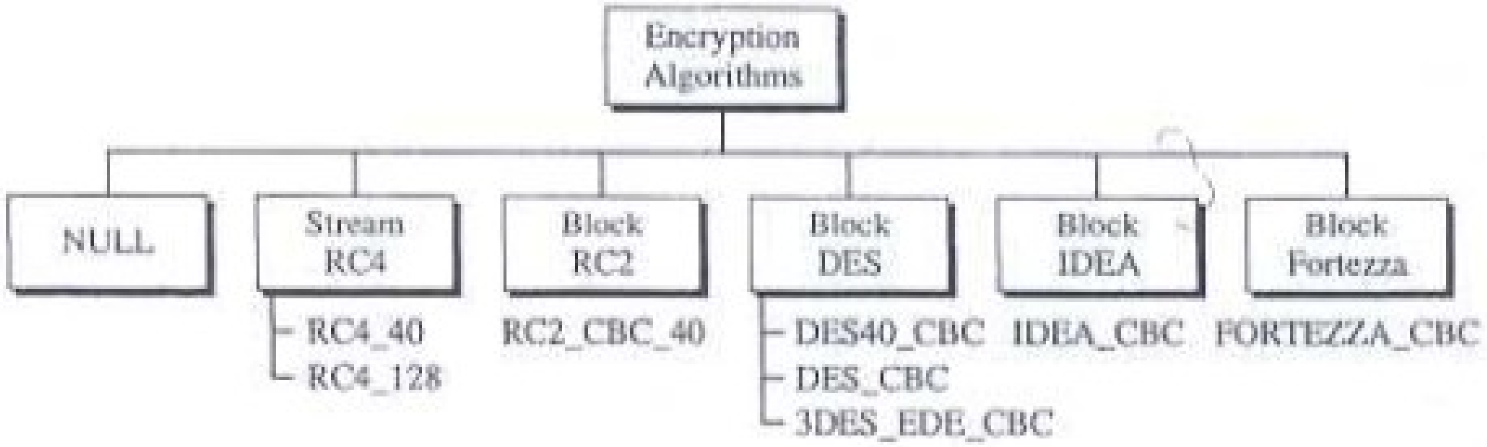
SSL-Key Exchange

- SSL defines six key-exchange methods to establish this pre-master secret



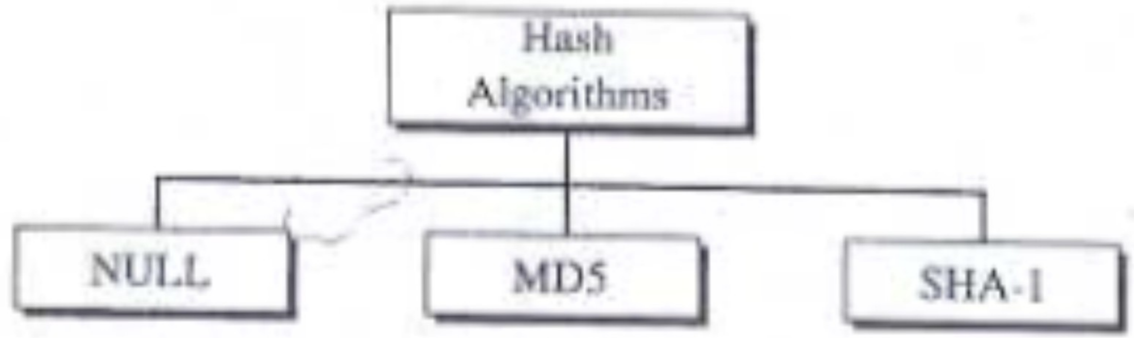
SSL-Encryption-Decryption

- For encryption and decryption, the following algorithms can be used:



SSL-Hash Algorithms

- SSL uses hash algorithms to provide message integrity. The following hash algorithms can be used

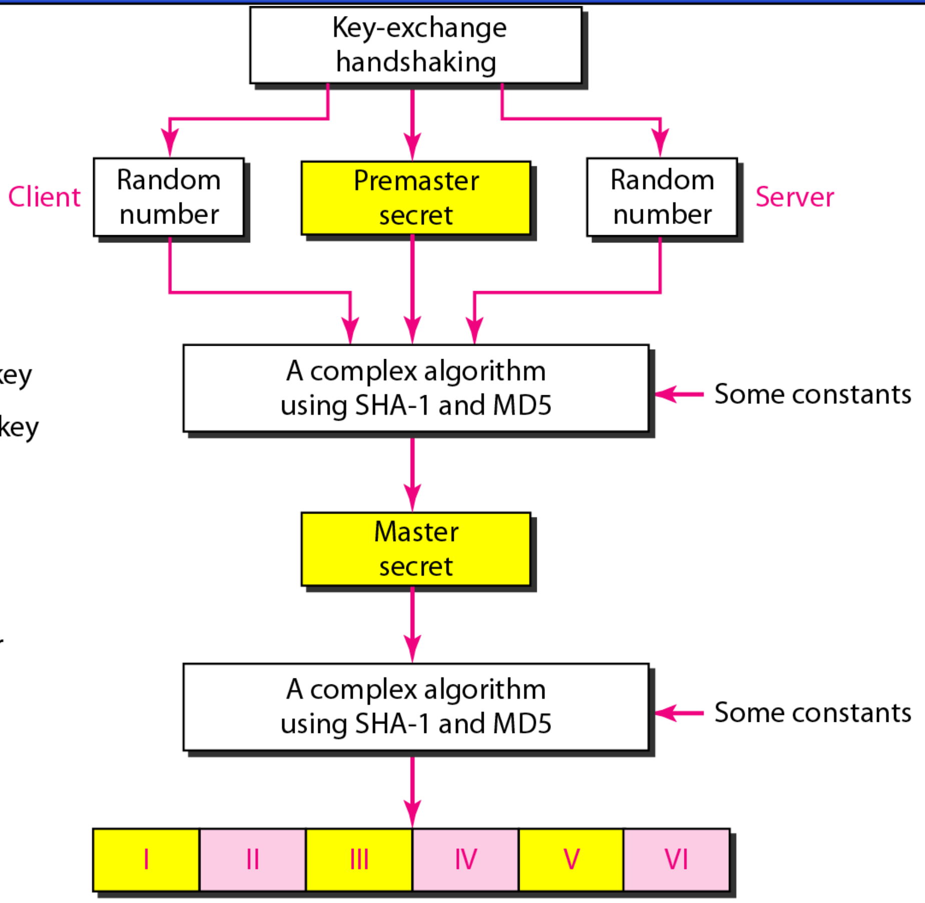


SSL-Cipher Suite

- The combination of <u>key exchange,hash, encryption algorithm defines a cipher suite</u> for each SSL session
- For example:
  SSL_DHE_RSA_WITH_DES_CBC_SHA

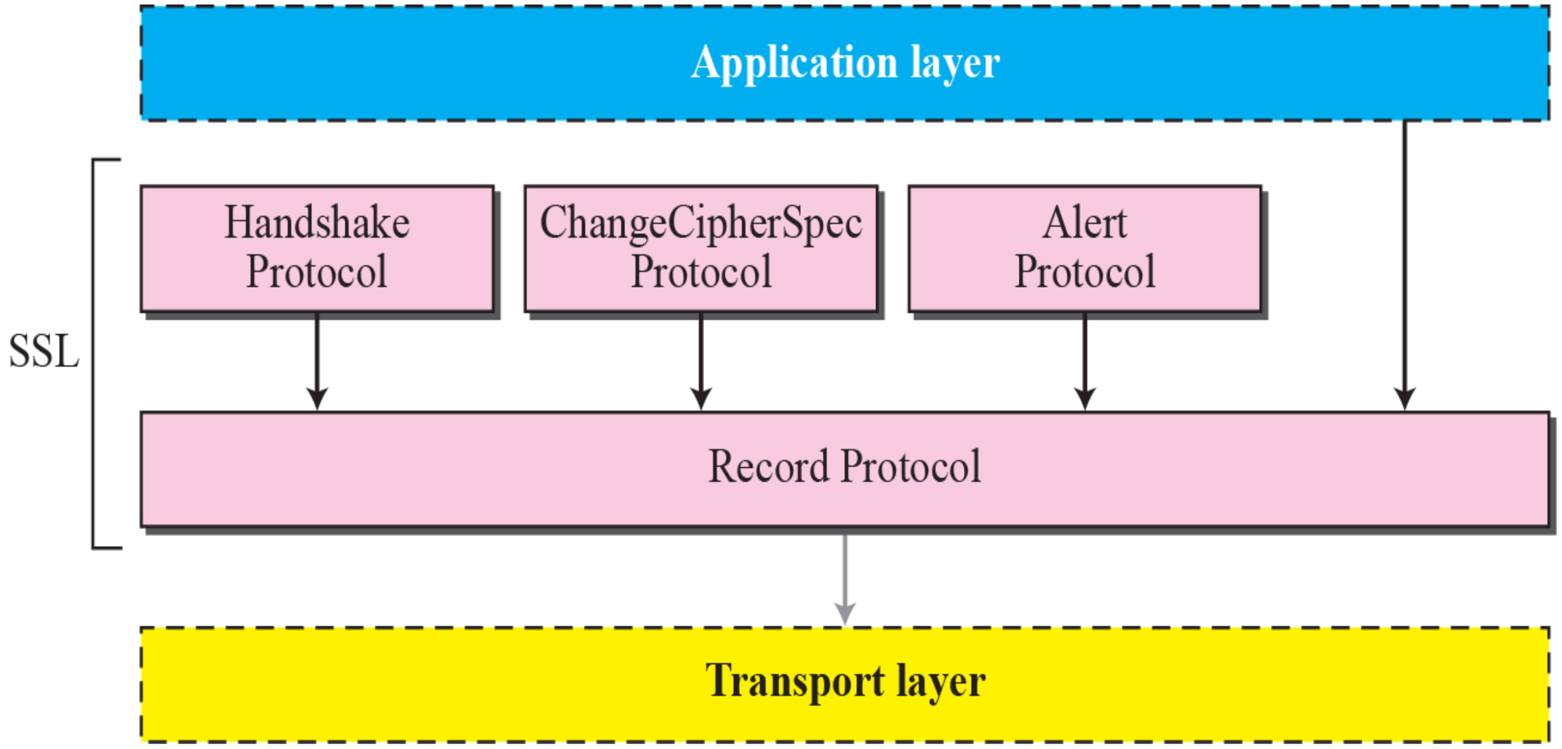
  - Each cipher suite starts with the term “SSL”
  - The word “WITH” separates the key exchange algorithm from encryption and hash algorithm

Client and Server cryptographic secrets



I	Client authentication key
II	Server authentication key
III	Client encryption key
IV	Server encryption key
V	Client initiation vector
VI	Server initiation vector

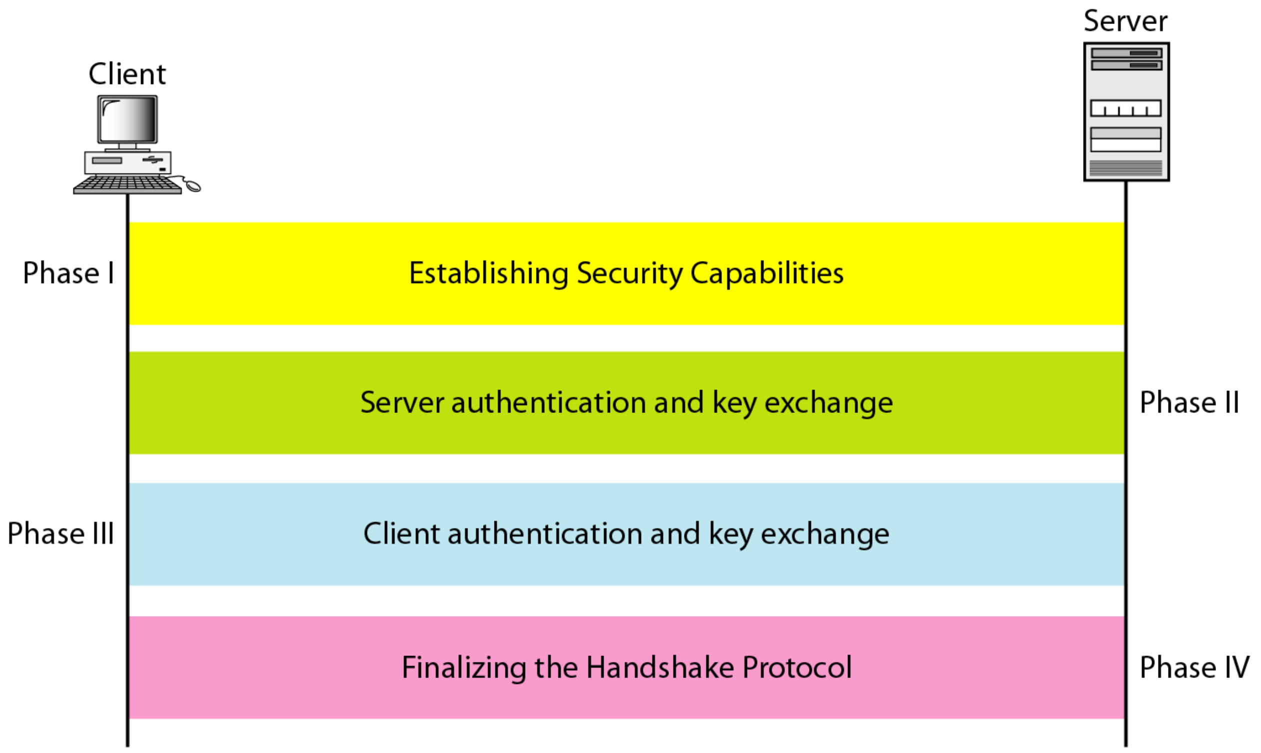
SSL protocols



SSL protocols

- **Handshake protocol-**
  - It provides security parameters for the Record Protocol
  - It establishes a cipher set and provides keys and security parameters
  - It authenticates the server to the client and client to the server
- **ChangeCipherSpec protocol-**
  - It is used for signalling the readiness of cryptographic secrets
- **Alert Protocol-**
  - It is used to report abnormal conditions
- **Record protocol-**
  - It is the carrier which carries messages from other three protocols and data coming from application layer to transport layer

Handshake Protocol

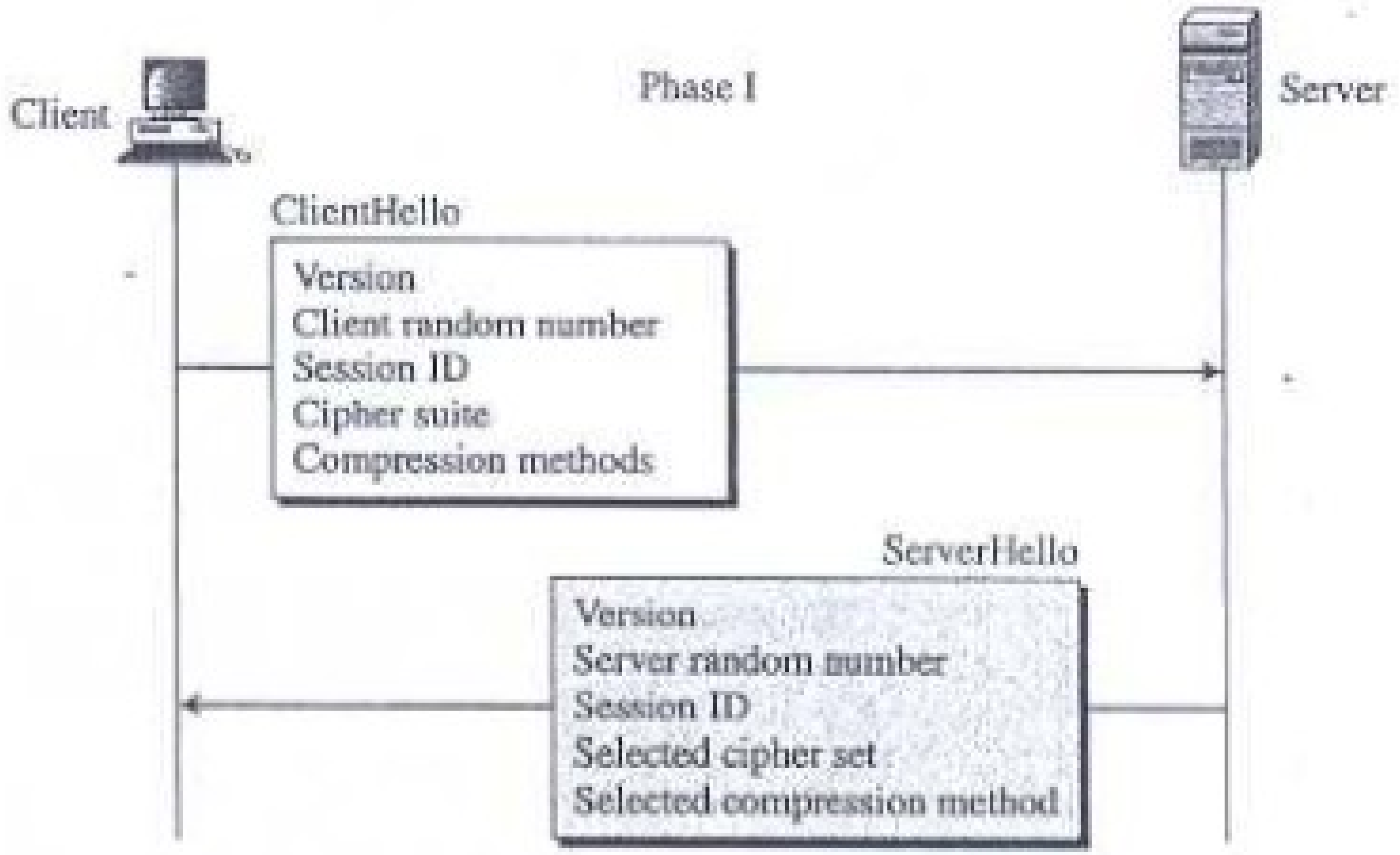

Server
Client
Phase I
Establishing Security Capabilities
Server authentication and key exchange
Phase II
Phase III
Client authentication and key exchange
Finalizing the Handshake Protocol
Phase IV

Handshake protocol-Phase I

Establishing Security Capability

- Client and server announce their security capabilities and choose those that are convenient for both
- A session id is established and a cipher suite is chosen
- The parties agree upon a particular compression method
- Two random numbers (one by the client and another by the server) are selected which are used for creating a master secret
- Two messages are exchanged:-
  - ClientHello
  - ServerHello

Handshake Protocol-Phase I



Handshake protocol-Phase I

- After Phase I, the client and server know the following:-
  - The version of SSL,
  - The algorithms for key exchange, message authentication and encryption
  - The compression method, and
  - The two random numbers for key generation

Handshake protocol-Phase II

Server Authentication and Key Exchange

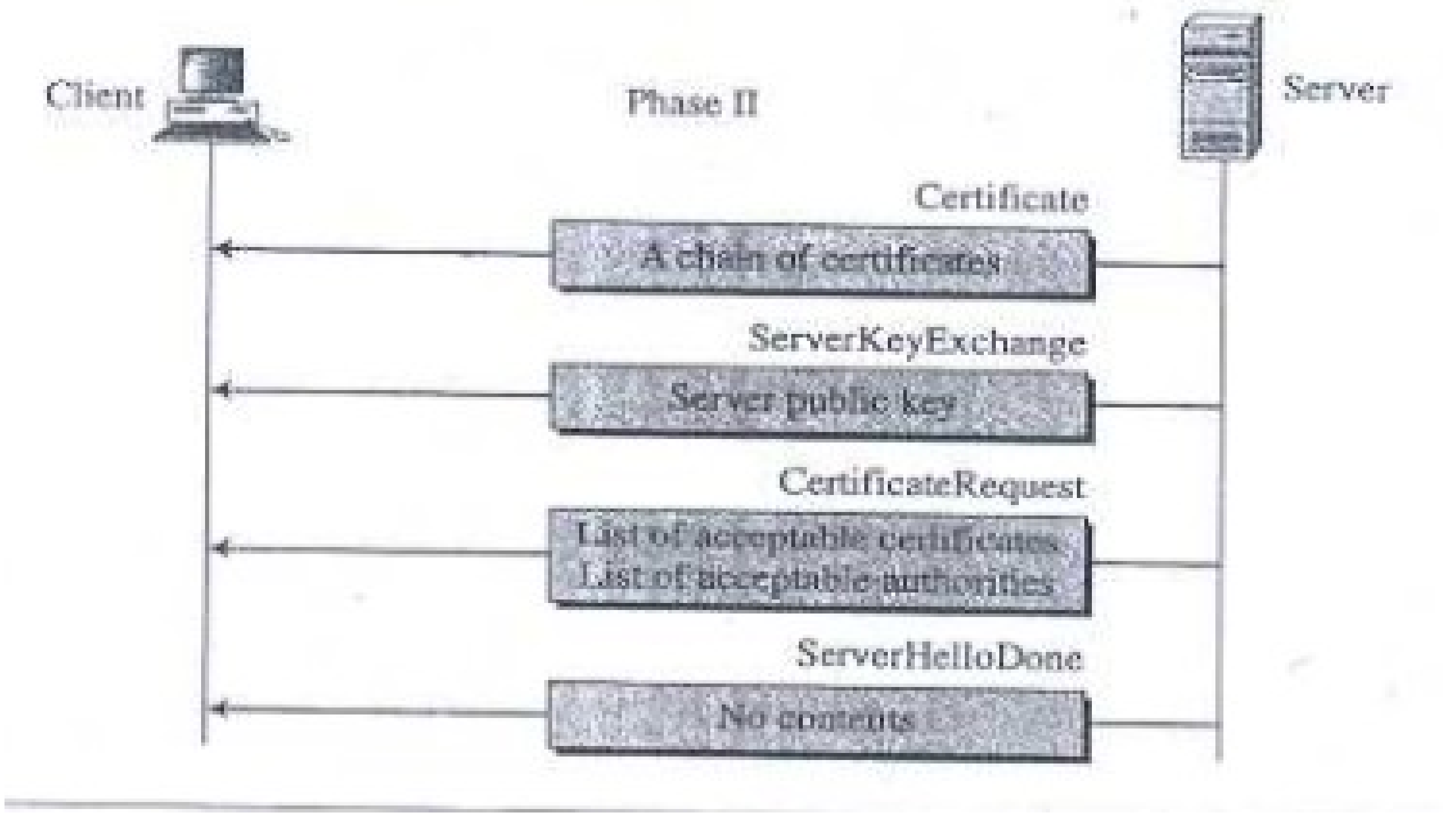
- The server authenticates itself
- The server sends its certificates, public key and request certificates from the client
- The server announces that the serverHello process is done
- Four messages are exchanged

Handshake protocol-Phase II

Server Authentication and Key Exchange

- **Certificate**: The server sends a Certificate Message to authenticate itself. The message includes a list of certificates of type X.509
- **ServerKeyExchange**: The server sends a ServerKeyExchange message tat includes its contribution to the pre-master secret
- **CertificateRequest**: The server requires the client to authenticate itself. The serevr sends a CertificateRequest message that asks for certification in Phase III from the client
- **ServerHelloDone**:It is a signal to the client that Phase II is over and the client has to start Phase III

Handshake protocol-Phase II



Handshake protocol-Phase II

After Phase II:

- the server is authenticated to the client, and
- the client knows the public key of the server if required.

Handshake protocol-Phase III

Client Authentication and Key Exchange

- This phase is designed for client authentication
- Three messages are sent from client to the server

Handshake protocol-Phase III

Client Authentication and Key Exchange

- **Certificate:**
  - To certify itself, the client sends a Certificate message.
  - It includes the chain of certificates that certify the client.
  - This message is sent only if the server has requested a certificate in Phase II
  - If there is a request and client has no Certificate to send then it sends a Alert message using Alert Protocol with a warning that there is no certificate.
  - The server after receiving the warning may continue with the session or may decide to abort

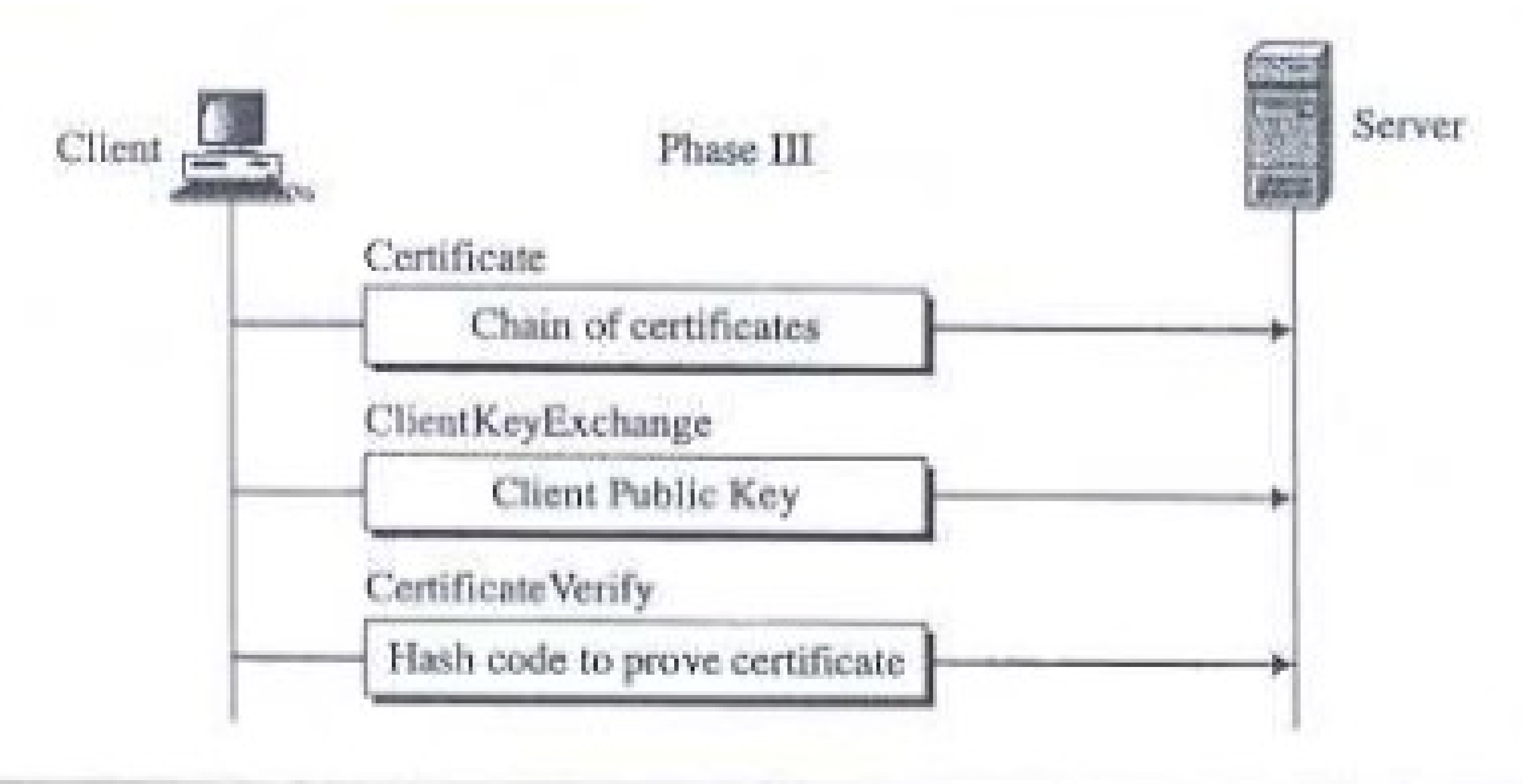
Handshake protocol-Phase III

<u>Client Authentication and Key Exchange</u>

- **ClientKeyExchange:**
  - The client sends a ClientKeyExchange message which includes its contribution to pre-master secret

- **CertificateVerify:**
  - If the client has sent a certificate declaring that it owns the public key then it has to prove that it knows the corresponding private key.
  - This proof is done by creating a message a message and signing it with the private key
  - The server can verify the message with the public key to ensure that the certificate belongs to the client

Handshake protocol-Phase III



Handshake protocol-Phase III

After Phase III,

- The client is authenticated for the server, and
- Both the client and the server know the pre-master secret.

Handshake protocol-Phase IV

Finalizing the Handshake Protocol

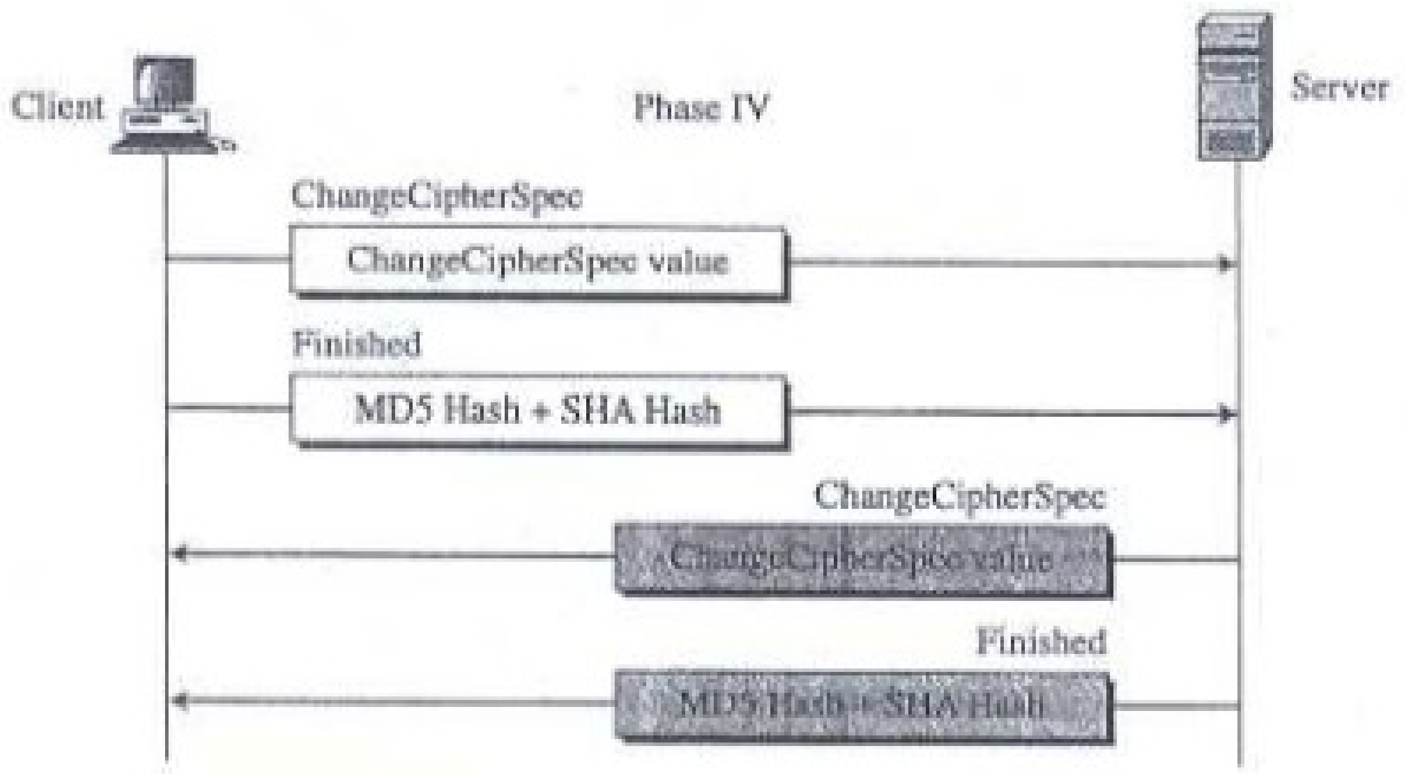
- The client and server send messages to change cipher specification
- They also send messages to finish the handshaking protocol
- Four messages are exchanged in this phase

Handshake protocol-Phase IV

<u>Finalizing the Handshake Protocol</u>

- **ChangeCipherSpec:**The client sends a ChangeCipherSpec message to show that it has moved all of the cipher suite set and the parameters from pending state to active state
- **Finished:** This message is sent by client that announces the end of handshaking protocol by the client
- **ChangeCipherSpec:**The server sends a ChangeCipherSpec message to show that it has also moved all of the cipher suite set and the parameters from pending state to active state
- **Finished:** The server send the Finished message to show that handshaking is totally completed

Handshake protocol-Phase IV


Client
Phase IV
Server
ChangeCipherSpec
ChangeCipherSpec value
Finished
MD5 Hash + SHA Hash
ChangeCipherSpec
ChangeCipherSpec value
Finished
MD5 Hash + SHA Hash