Client Hello  
Version Number:  
- Client sends the version number corresponding to the highest version it supports. In this case TLS version 1.0.  
  
Randomly Generated Data:  
- A 32 byte value consisting of a 4-byte number that contains the client’s date and time, plus a 28 byte randomly generated number that ultimately will be used with the server random value to generate a master secret.  
Encryption keys will be derived from this master secret.

Session Identification:  
- The sessionID is included to enable the client to resume a previous session. This can be useful because creating a new session requires public key operations which is CPU-intensive.

Cipher Suite:  
A list of the client’s available cipher suits.  
TLS\_RSA\_WITH\_DES\_CBC\_SHA: TLS is the protocol version, RSA is the algorithm that will be used for the key exchange, DES\_CBC is the encryption algorithm (using a 56-bit key in CBC mode), and SHA is the hash function.

Server Hello  
  
Version Number:  
- The server sends the highest version number supported by BOTH sides.

Randomly Generated Data:  
- 32-byte value. 4 bytes is the server’s time and date, 28 bytes is a randomly generated number.  
This number will eventually be used with the client random number to generate the master secret, from which the encryption keys will be derived.

SessionID:  
- This can only be one of three choices:  
 New Session ID – The client did not indicate a session to resume, so a new is created.  
 Resumed Session ID – The ID is the same as indicated in the client hello. So, resume.  
 Null: This is a new session, but the server is not willing to resume it at a later time.

Cipher Suite:  
- The server will choose the strongest cipher that BOTH the client and server support. If no cipher is supported by both the client and server, the session is ended with a “Handshake Failure” alert.

Server Certificate

- The server sends its certificate to the client. This certificate contains the server’s public key, which the client will use to authenticate and encrypt the premaster key.  
The client also checks the name of the server in the certificate to verify that is matches the name the client used to connect.  
  
Example: If the client types [www.facebook.com](http://www.facebook.com) in the browser, the certificate should contain a subject name of “www.facebook.com” or “\*.facebook.com”.  
Most browsers will warn the user if these names don’t match.

Server Key Exchange  
This is an optional step in which the server creates and sends a temporary key to the client. This key can be used by the client to encrypt the Client Key Exchange message later in the process.

This step is only required when the public key algorithm does not provide the key material necessary to encrypt the Client Key Exchange message, such as when the server’s certificate does not contain a public key.

Server Hello Done  
This message indicated that the server is finished and awaiting a response from the client.

Client Key Exchange  
The client sends a Client Key Exchange message after computing the premaster secret using both random values.  
The premaster secret is encrypted by the public key from the server’s certificate before being transmitted to the server.  
Both the client and server will compute the master secret locally and derive the session key from it.  
  
If the server can decrypt this data and complete the protocol, the client is assured that the server has the correct private key.  
This step is crucial to prove the authenticity of the server – only the server with the private key that matches the public key in the certificate can decrypt this data and continue the protocol negotiation.

This message also includes the protocol version.  
This is because the server can verify that it matches the original value sent in the Client Hello message.  
This measure guards against Rollback Attacks[1].

Change Cipher Spec (Client)  
This message notifies the server that all messages that follow the Client Finished message will be encrypting using the keys and algorithms just negotiated.

Client Finished

Change Cipher Spec (Server)  
This message notifies the client that the server will encrypt all the following messages with the keys just negotiated.

Server Finished Message  
This message is a hash of the entire exchange to this point using the session key and the MAC secret.  
If the client is able to successfully decrypt this message, and validate the contained hashes, it now is sure that the SSL/TLS handshake was successful, and that both the client keys and server keys match.

[1]: Rollback Attacks works by manipulating the message in order to cause the server and client to use a less secure, earlier version of the protocol.