Transport Layer Security (TLS)

Originally, encryption across socket connections was devised as Secure Sockets Layer (SSL). SSL was a proprietary standard created by Taher Elgamal and his team at Netscape. Elgamal, who also created ElGamal encryption and the Elgamal signature scheme, is known as the father of SSL.

SSL v3.0 was modified and became Transport Layer Security(TLS) v1.0 in RFC 2246, making it a public standard instead of proprietary.

TLS is designed to provide three security services:

- encryption of transmissions;
- authentication of content and entities; and
- integrity of message contents, i.e., detection of tampering and forgery

TLS connections are initiated using a handshake protocol (Figure 2):

1. The client connects over TCP with a server.
2. The client sends a client_hello message containing the following:
    a) highest version of TLS that the client understands
    b) 60 bytes of pseudorandom data, consisting of a 32 byte datestamp and 28 bytes from a pseudorandom number generator; this value is used to prevent replay attacks
    c) a zero if starting a new session; otherwise, a session ID for creating or modifying a connection on an existing session
    d) ciphersuite list, which is a list of ciphersuites supported by the client. A ciphersuite is a combination of a key exchange algorithm, a bulk encryption algorithm, a MAC algorithm, and a pseudorandom number generator algorithm.
    e) compression method, which is a list of data compression methods supported by the client

3. In response, the server sends a server_hello message with the same basic structure:
    a) the version is the highest mutually-supported version of TLS
    b) the session id is either the id of the newly-created session or the id of the resumed session
    c) the CipherSuite is the one selected from the client's list, presumably the most secure mutually-supported one
    d) the Compression method is the one selected from the client's list
4. The server then sends its certificate to the client, if authentication is required.
5. The server sends a key exchange message.
6. If the server demands authentication of the client, it will then send a certificate request message to the client.
7. The server sends a server_hello_done message to the client.
8. If requested of it, the client remits its certificate to the server.
9. The client responds to the key exchange message. Through these steps, the client and the server use the pseudorandom numbers to create a master secret that is shared and can be used for encryption.

10. If the client authenticated the server, it sends a certificate_verify message to the server.
11. Assuming everything goes as planned, the client then sends a change_cipher_spec message, indicating that the agreed upon cipherSuite will be used from there on to encrypt all further communications.
12. The client sends a finished message to the server.
13. The server responds with a change_cipher_spec message, indicating that it, too, has switched to encrypted communication.
14. Finally, the server sends a finished message to the client.

Once the handshake is completed, TLS divides packets into records for sending; see the Record Protocol (Figure 1).

- The session id is an association with the shared master secret. The client also stores the server's IP address and port number with the session id.
- All TLS packets are sent using a Record structure.
- TLS records may be compressed, contain a HMAC of the payload, or be encrypted, depending on the state of the connection.
- The Alert protocol is used to send process messages to the other side, such as when closing down a connection or when an error occurs.