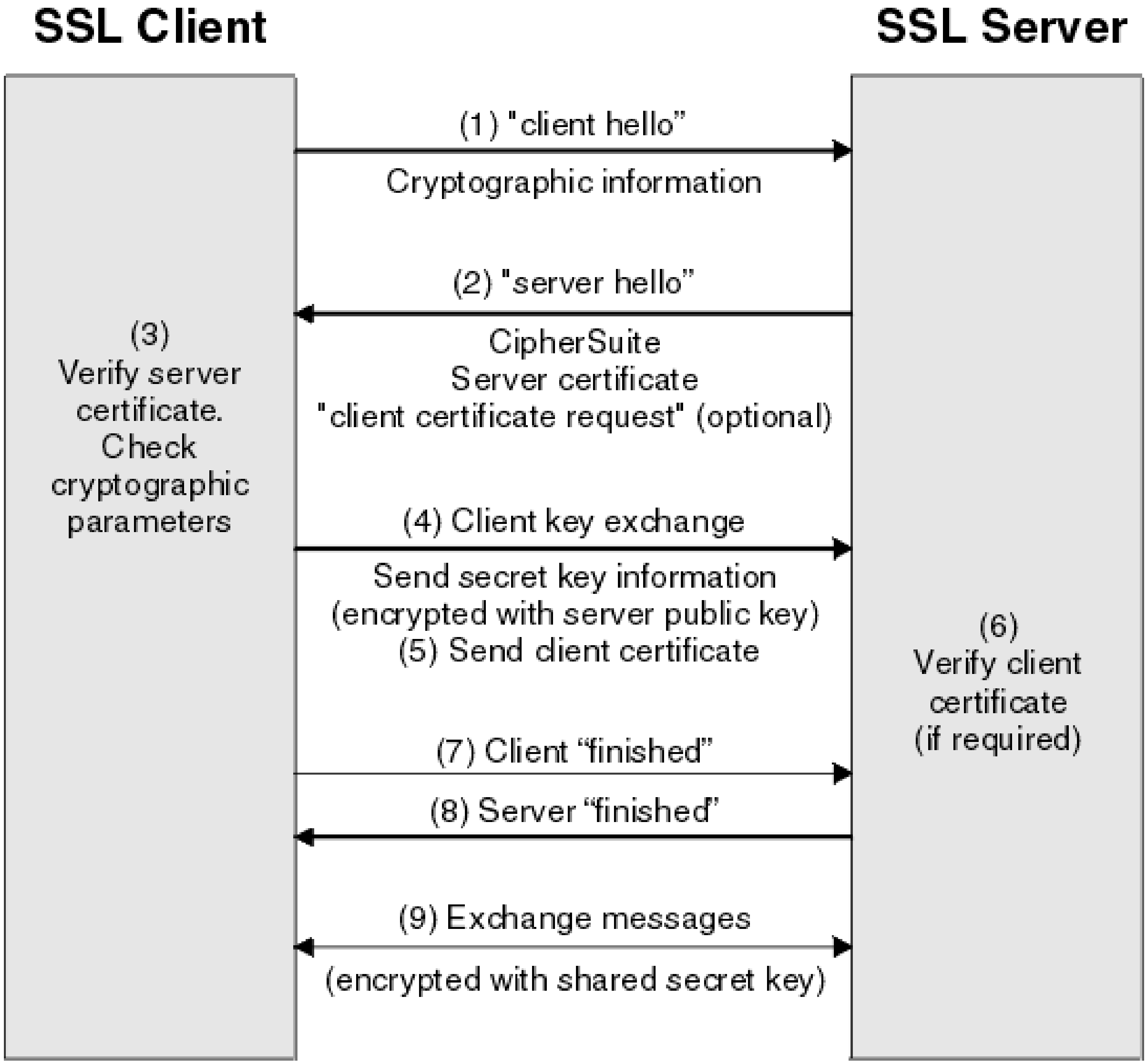
Secure Communication Protocols

TLS (or SSL) Communication Set-up

1. The TLS client sends a client hello message that lists cryptographic information such TLS version and, in the client's order of preference, the CipherSuites supported by the client, and a client random.
2. The TLS server responds with a server hello message that contains the CipherSuite chosen by the server from the list provided by the client, the session ID, and a server random. The server also sends its digital certificate. If the server requires a digital certificate for client authentication, the server sends a client certificate request.
3. The TLS client verifies the server's digital certificate.
4. The TLS client sends another random byte string (known as the pre-master secret) that enables both the client and the server to compute the secret key to be used for encrypting subsequent message data. The random byte string itself is encrypted with the server's public key.

TLS Communication (cont’d)

5. If the TLS server sent a client certificate request, the client sends a random byte string encrypted with the client's private key.
6. The TLS server verifies the client's certificate.
7. Client and Server generate secret session keys based off random number.
8. The TLS client sends the server a finished message, which is encrypted with the secret key, indicating that the client part of the handshake is complete.
9. The TLS server sends the client a finished message, which is encrypted with the secret key, indicating that the server part of the handshake is complete.
10. For the duration of the TLS session, the server and client can now exchange messages that are symmetrically encrypted with the shared secret key.



SSL Client

SSL Server

(1) "client hello"

Cryptographic information

(2) "server hello"

CipherSuite
Server certificate
"client certificate request" (optional)

(3)
Verify server
certificate.
Check
cryptographic
parameters

(4) Client key exchange

Send secret key information
(encrypted with server public key)
(5) Send client certificate

(6)
Verify client
certificate
(if required)

(7) Client "finished"

(8) Server "finished"

(9) Exchange messages

(encrypted with shared secret key)

Internet Key Exchange (IKEv2)

- Popular for VPNs
- Step 1 : Negotiation - The peer that has traffic that should be protected will initiate the IKE negotiation. The two peers will negotiate about the following items:
  - Hashing: we use a hashing algorithm to verify the integrity.
  - Authentication: each peer has to prove who he is. Two commonly used options are a pre-shared key or digital certificates.
  - DH (Diffie Hellman) group: the DH group determines the strength of the key that is used in the key exchange process. The higher group numbers are more secure but take longer to compute.
  - Lifetime: how long does the IKE phase 1 tunnel stand up? the shorter the lifetime, the more secure it is because rebuilding it means we will also use new keying material. Each vendor uses a different lifetime, a common default value is 86400 seconds (1 day).
  - Encryption: what algorithm do we use for encryption? For example, AES.
- Step 2: DH Key Exchange
  - Once the negotiation has succeeded, the two peers will know what policy to use. They will now use the DH group that they negotiated to exchange keying material. The end result will be that both peers will have a shared key.

IKEv2 (cont’d)

- Step 3: Authentication
- The last step is that the two peers will authenticate each other using the authentication method that they agreed upon on in the negotiation. When the authentication is successful, we have completed IKE phase 1. The end result is a IKE phase 1 tunnel (aka ISAKMP tunnel) which is bidirectional. This means that both peers can send and receive on this tunnel.

- IKEv2 is common in peer-to-peer secure protocols