CS 465

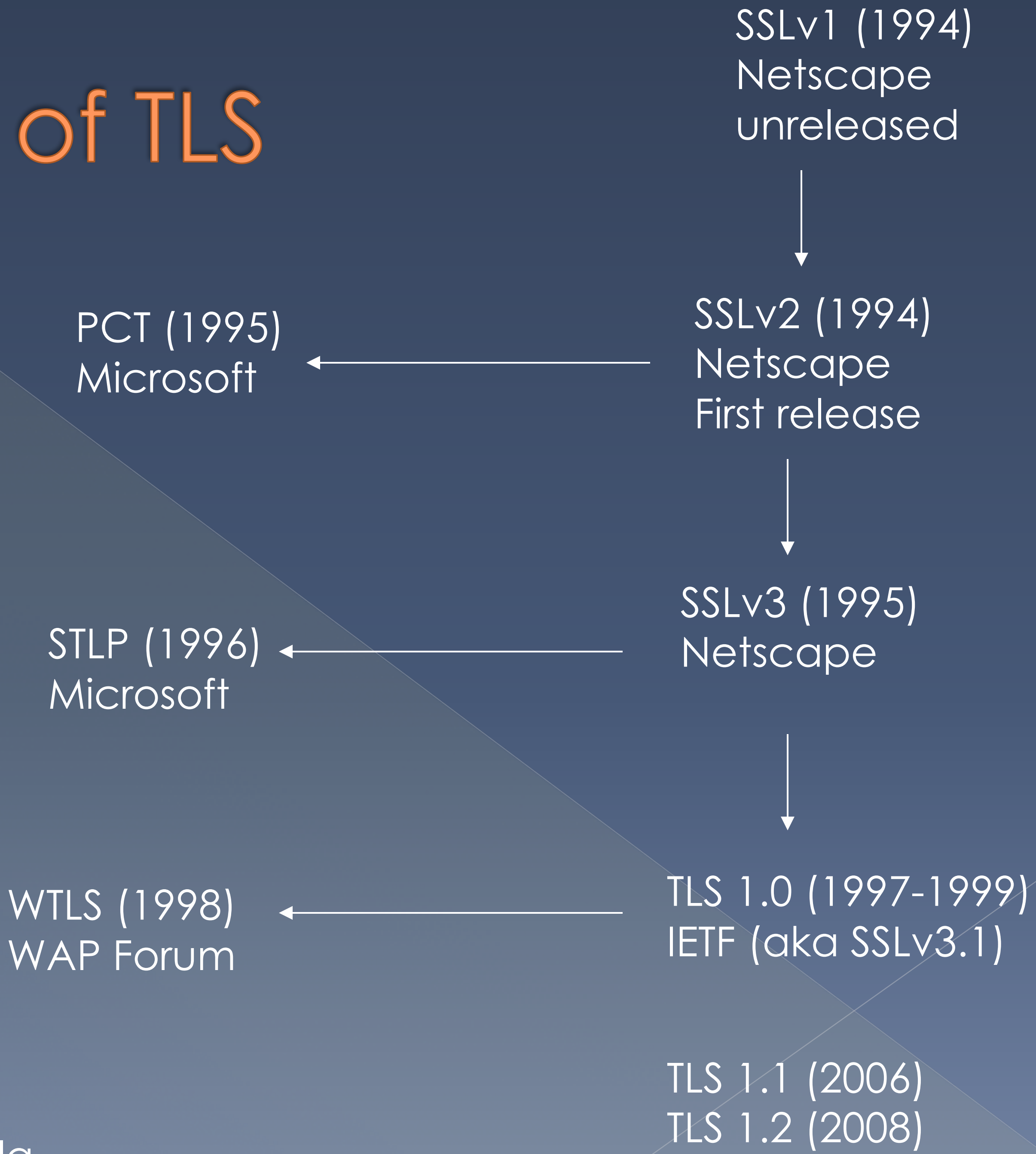
TLS

Last Updated: Oct 31, 2017

Student Learning Goals

- Understand the TLS handshake
- Understand client/server authentication in TLS
  - RSA key exchange
  - DHE key exchange
  - Explain certificate ownership proofs in detail
  - What cryptographic primitives are used and why?
- Understand session resumption
- Understand the limitations of TLS

Genesis of TLS



SSLv1 (1994)
Netscape
unreleased

SSLv2 (1994)
Netscape
First release

PCT (1995)
Microsoft

SSLv3 (1995)
Netscape

STLP (1996)
Microsoft

TLS 1.0 (1997-1999)
IETF (aka SSLv3.1)

WTLS (1998)
WAP Forum

TLS 1.1 (2006)
TLS 1.2 (2008)

Source: SSL and TLS, Rescorla

SSL RECORD PROTOCOL OPERATION

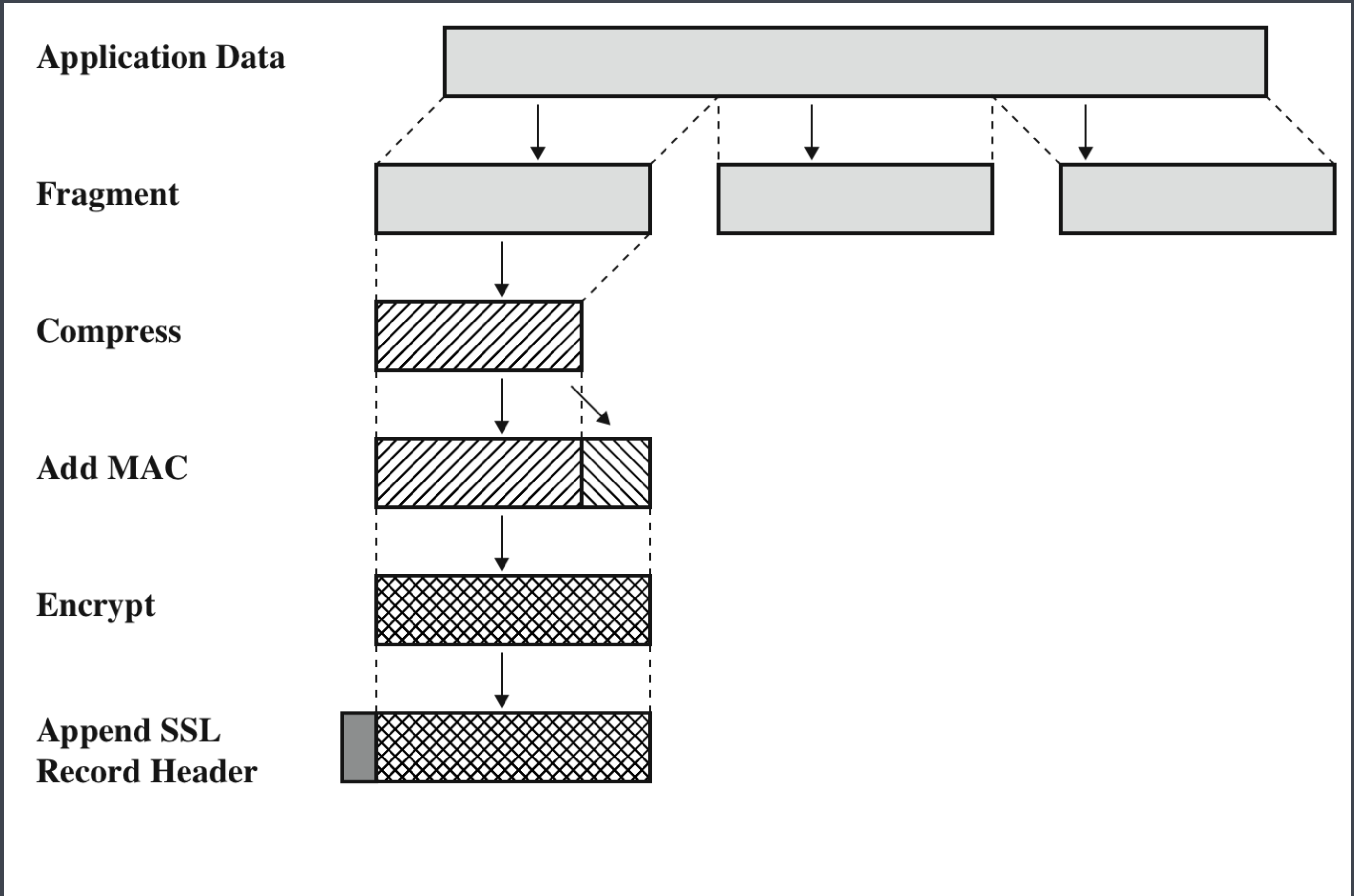


Figure 7.3 SSL Record Protocol Operation

SSL Record Protocol Operation

Source: Network Security Essentials (Stallings)

SSL Record Format

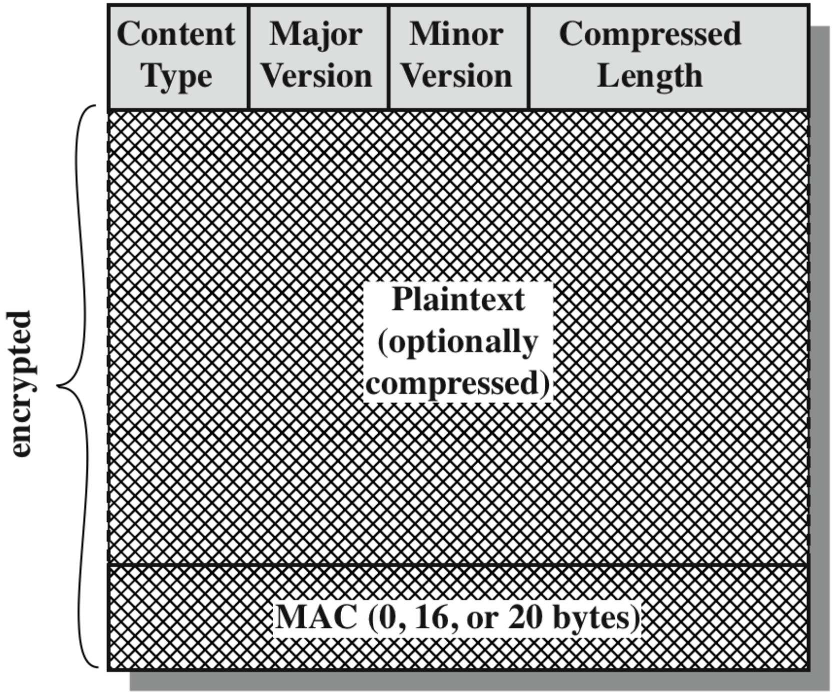
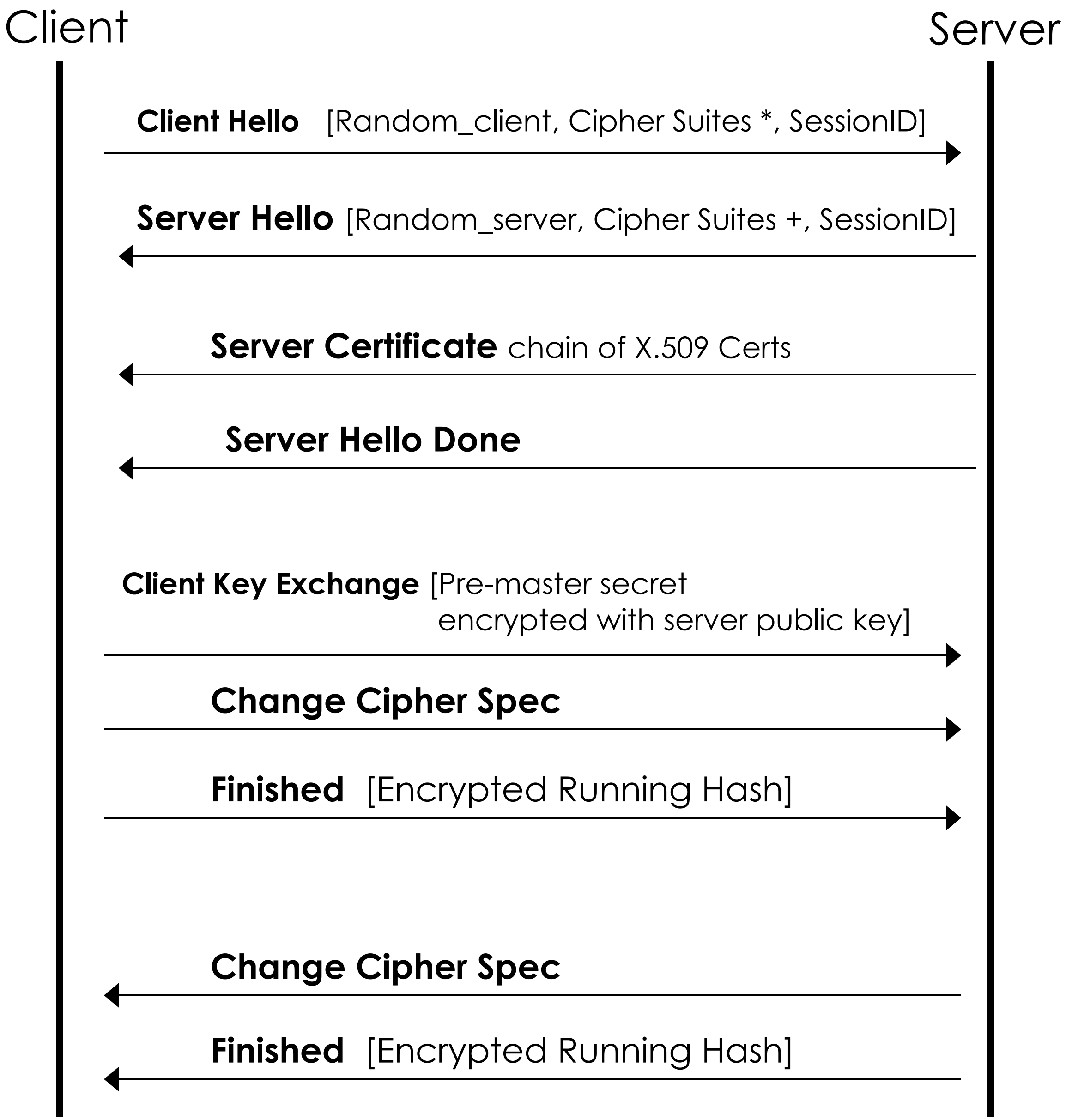


Figure 7.4 SSL Record Format

SSL Record Format

Source: Network Security Essentials (Stallings)

RSA Key Exchange Method



Client

Server

Client Hello [Random_client, Cipher Suites *, SessionID]

Server Hello [Random_server, Cipher Suites +, SessionID]

Server Certificate chain of X.509 Certs

Server Hello Done

Client Key Exchange [Pre-master secret
encrypted with server public key]

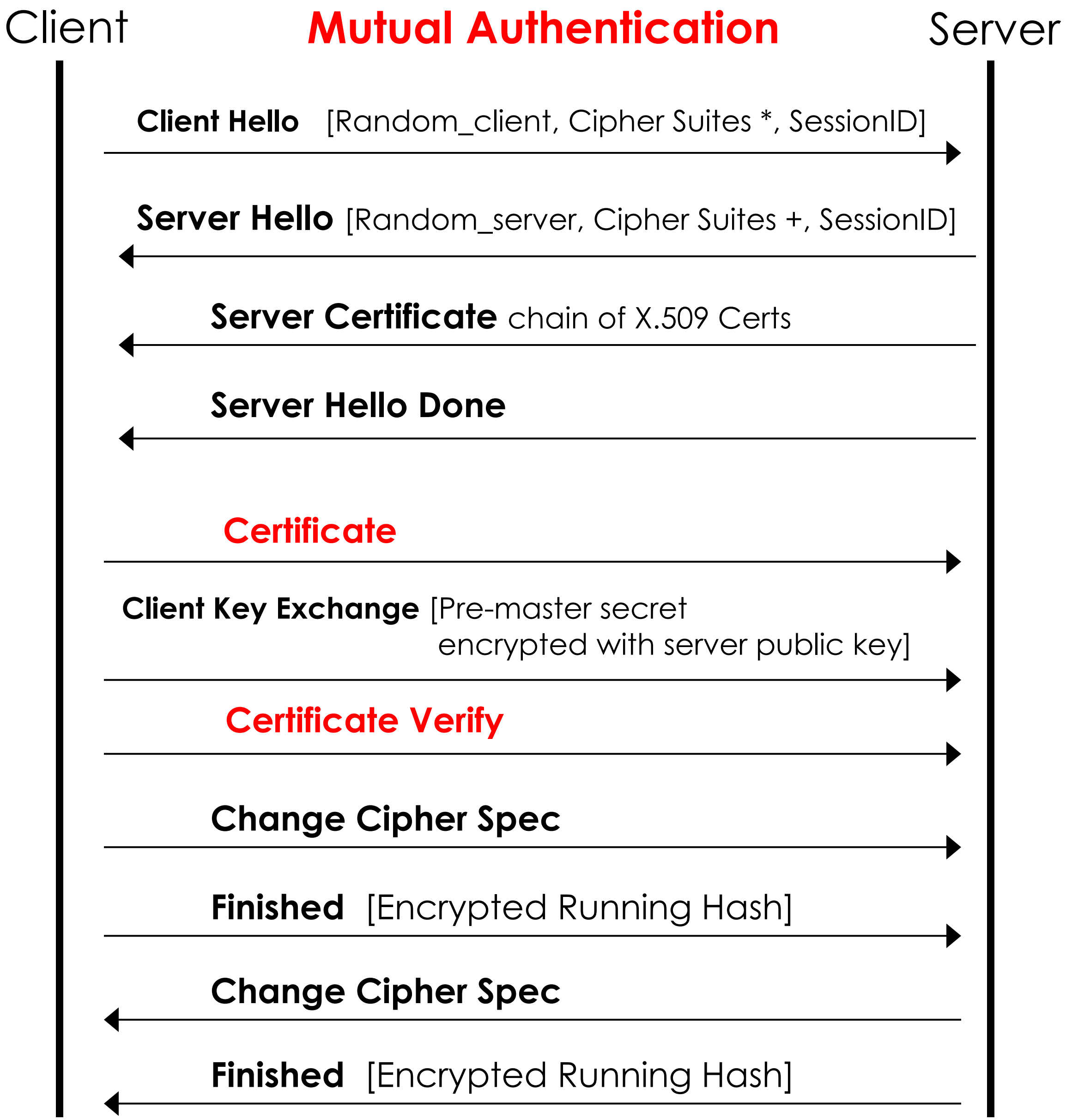
Change Cipher Spec

Finished [Encrypted Running Hash]

Change Cipher Spec

Finished [Encrypted Running Hash]

RSA Key Exchange Method



Client

Mutual Authentication

Server

Client Hello [Random_client, Cipher Suites *, SessionID]

Server Hello [Random_server, Cipher Suites +, SessionID]

Server Certificate chain of X.509 Certs

Server Hello Done

Certificate

Client Key Exchange [Pre-master secret
encrypted with server public key]

Certificate Verify

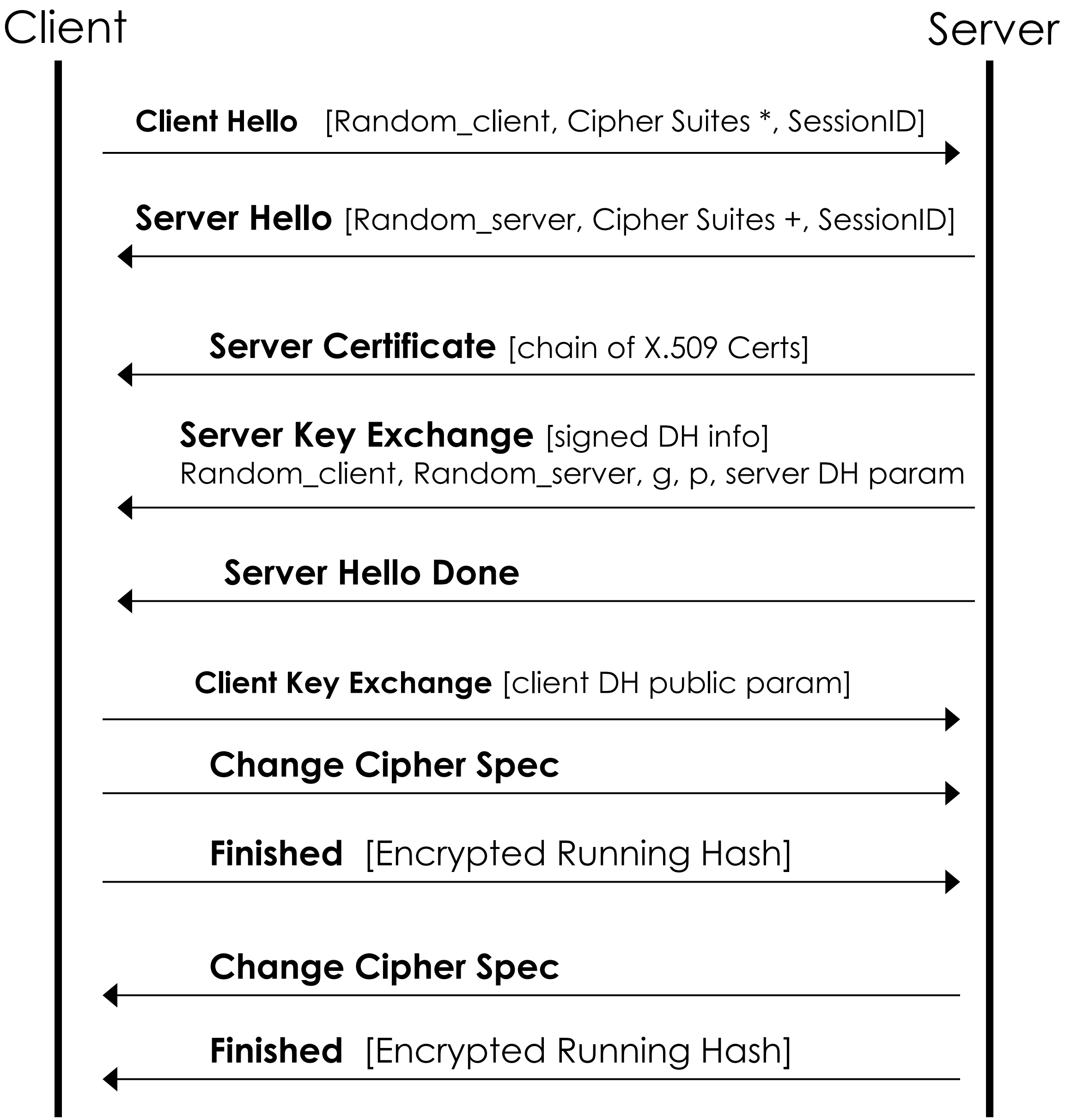
Change Cipher Spec

Finished [Encrypted Running Hash]

Change Cipher Spec

Finished [Encrypted Running Hash]

DHE Key Exchange Method



Client

Server

Client Hello [Random_client, Cipher Suites *, SessionID]

Server Hello [Random_server, Cipher Suites +, SessionID]

Server Certificate [chain of X.509 Certs]

Server Key Exchange [signed DH info]
Random_client, Random_server, g, p, server DH param

Server Hello Done

Client Key Exchange [client DH public param]

Change Cipher Spec

Finished [Encrypted Running Hash]

Change Cipher Spec

Finished [Encrypted Running Hash]

Key Material for TLS

- RSA
  - Client generates pre-master secret
  - Sends to server encrypted with servers public key
- DHE
  - DH shared key is the pre-master secret
- Pre-master secret and random values used to compute master secret
- Master secret and random values used to compute key block material
  - Key block contains 4 or 6 keys
  - Two keys for AES, 2 keys for MAC, 2 keys (IV) for block cipher mode if needed

Perfect Forward Secrecy

- In vanilla RSA, the premaster secret is encrypted with the server's public key
  - If the server's private key is compromised all past and future sessions are also compromised
  - Majority of TLS uses vanilla RSA
- Alternatives
  - Ephemeral Diffie-Hellman (DHE-RSA)
  - Elliptic curve variation is faster (ECDHE)

Forward Secrecy

- Using an ephemeral key
  - Even if the server's private key is later compromised, past sessions cannot be decrypted, even if captured and stored by a third party

TLS 1.3

- https://blog.cloudflare.com/tls-1-3-overview-and-q-and-a/
  - Reduced round trips in the handshake
  - Certificates are encrypted
  - Quick session resumption

Review Questions

- How many shared keys are derived between a client and a server that establish a TLS session?
- How does the server prove ownership of its private key?
- How does the client prove ownership of its private key when client authentication is (rarely) used?
- What is the pre-master secret?
  - Who creates it?
  - How is it securely transmitted?
- What is session resumption?
  - How does it differ from a regular SSL handshake?
- When do the client and server start encrypting traffic using symmetric encryption?

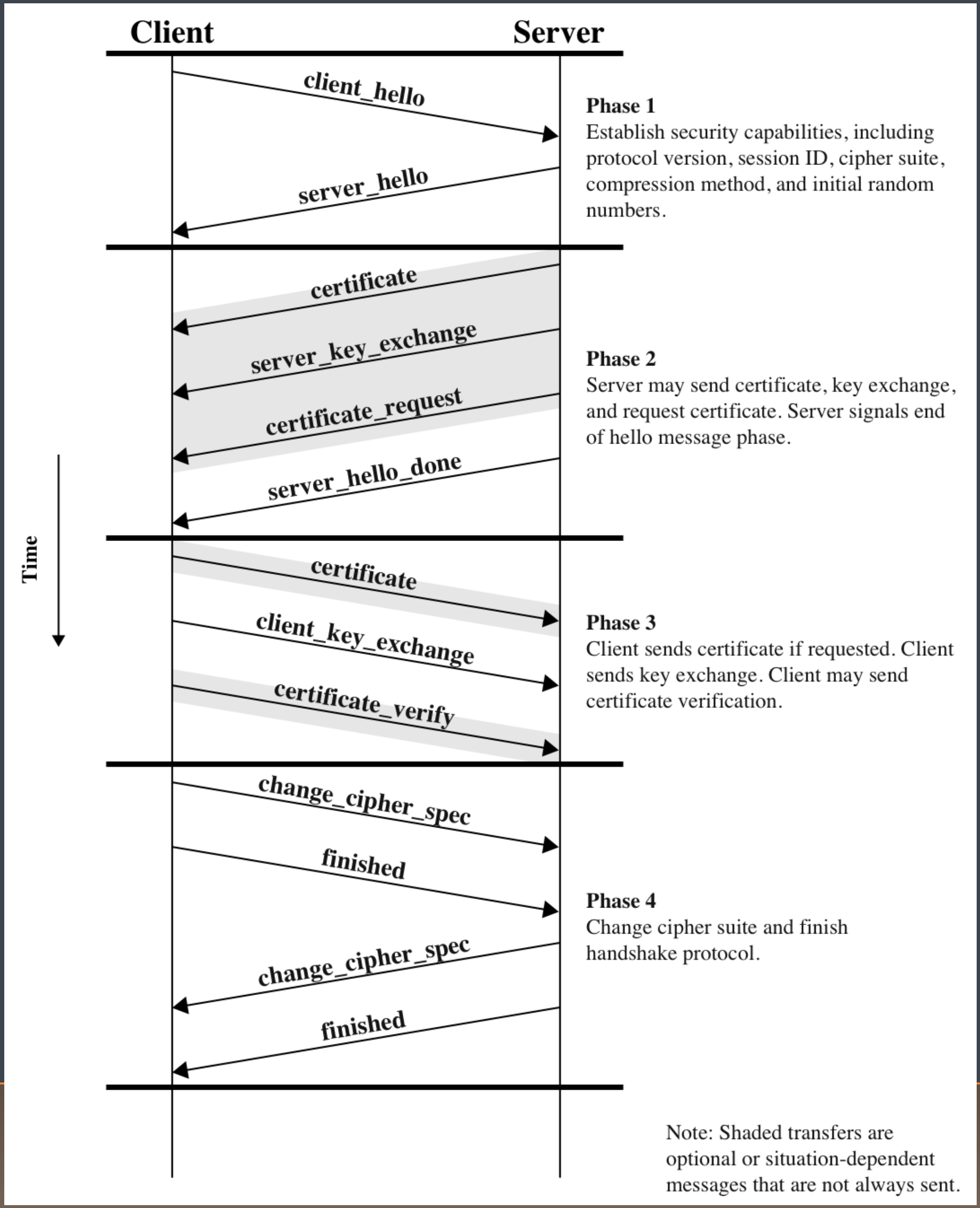
Review Questions

- How many shared keys are derived between a client and a server that establish a TLS session?
  - Each side generates 4-6 keys
- How does the server prove ownership of its private key?
  - Implicitly by decrypting the pre-master secret and finishing handshake
- How does the client prove ownership of its private key when client authentication is (rarely) used?
  - Send digital signature to the server
- What is the pre-master secret?
  - Who creates it?
  - How is it securely transmitted?
- What is session resumption?
  - How does it differ from a regular SSL handshake?
- When do the client and server start encrypting traffic using symmetric encryption?
  - Finished message

Limitations/Issues

- Certificate Authority system
- TLS Proxies
- TLS Inspection
  - Proxies, Middleboxes
- Other approaches
  - Pinning (TOFU)
  - Notaries (Crowd)
  - DANE (DNS-based)

SSL HANDSHAKE



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Source: Network Security Essentials (Stallings)