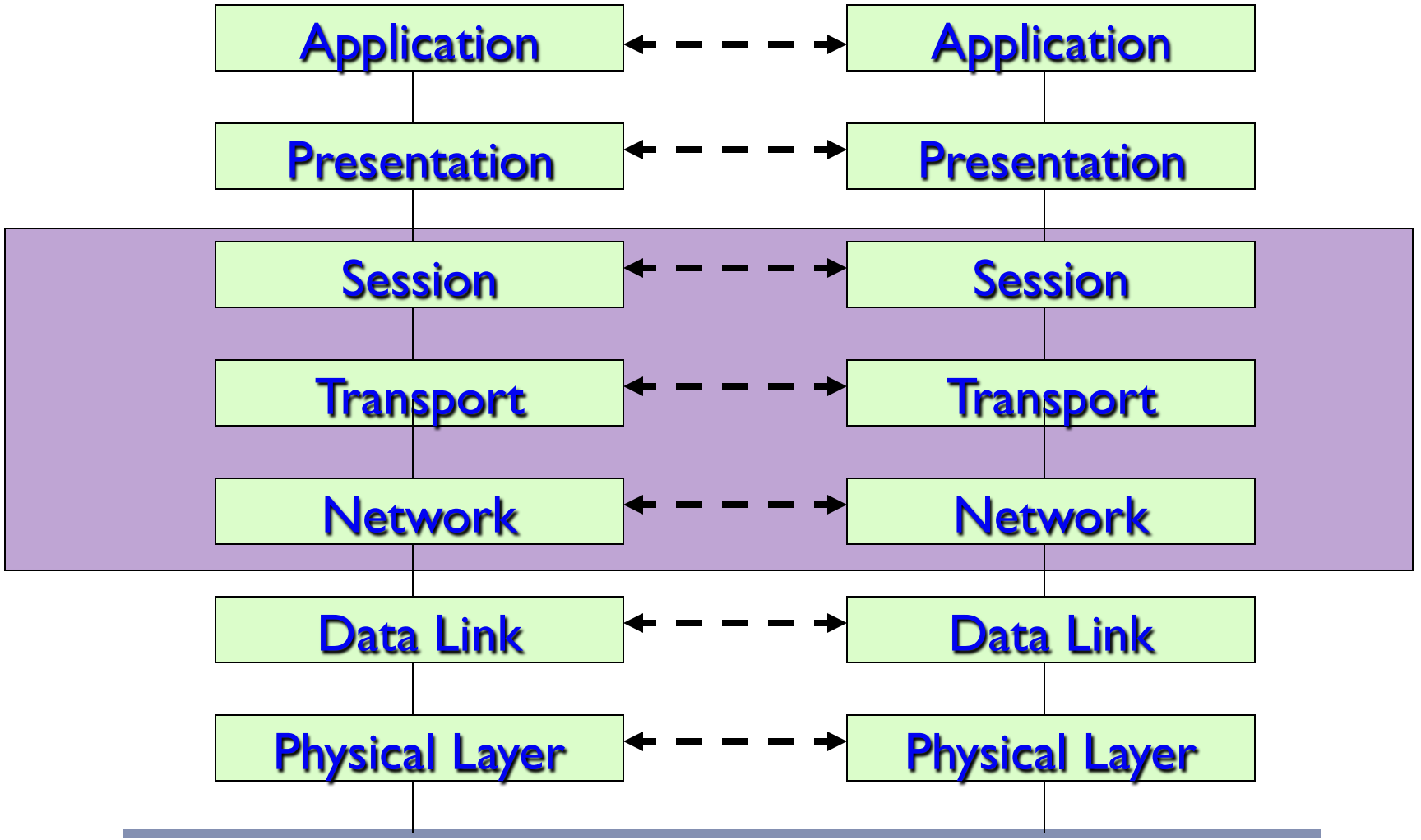
Cristina Nita-Rotaru

CS355: Cryptography

Lecture 19: TLS

OSI/ISO Model



Internet Protocol - IP

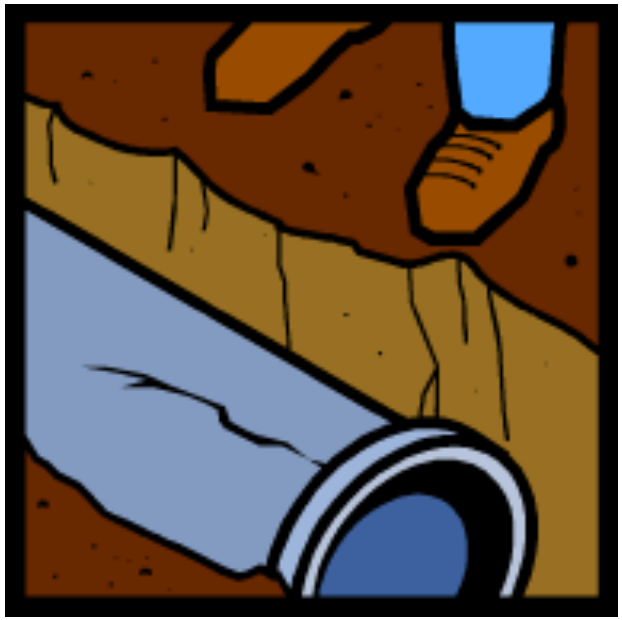
- IP is the current **delivery** protocol on the Internet, between **hosts**.
- IP provides ‘best effort’, unreliable delivery of packets.
- There are two versions:
  - IPv4 is the current routing protocol on the Internet
  - IPv6, a newer version, still not totally embraced by the community

Transport Protocols

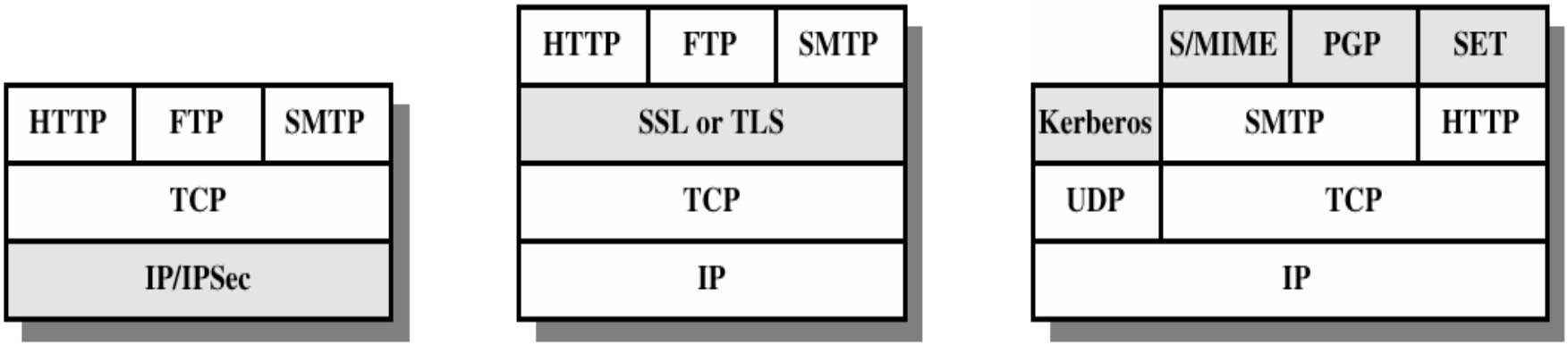
- Provides communication between **processes** running on hosts
- The most common transport protocols are **UDP and TCP.**
- OS provides support for developing applications on top of UDP and TCP.

Establishing a ``Secure Channel''

- Services provides: confidentiality, integrity and authentication
- At what level in the stack?
- What are advantages disadvantages based on the level
- Two protocols: SSL (TLS) and IPSec



Providing Security



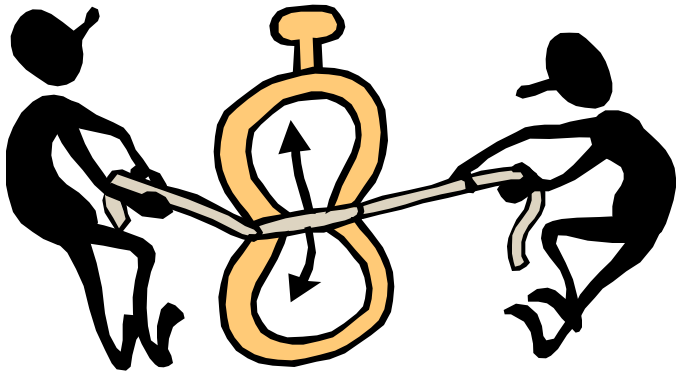
(a) Network Level

(b) Transport Level

(c) Application Level

SSL and TLS History

- SSL was originated by Netscape
- SSLv3 was designed with public comment and made available, known as TLS
- An working group was formed within IETF
- http://www.ietf.org/html.charters/tls-charter.html

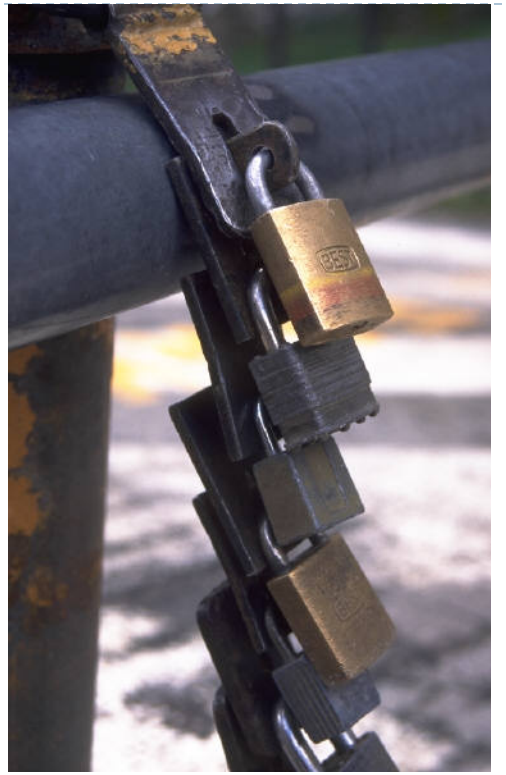


What is Transport Layer Security

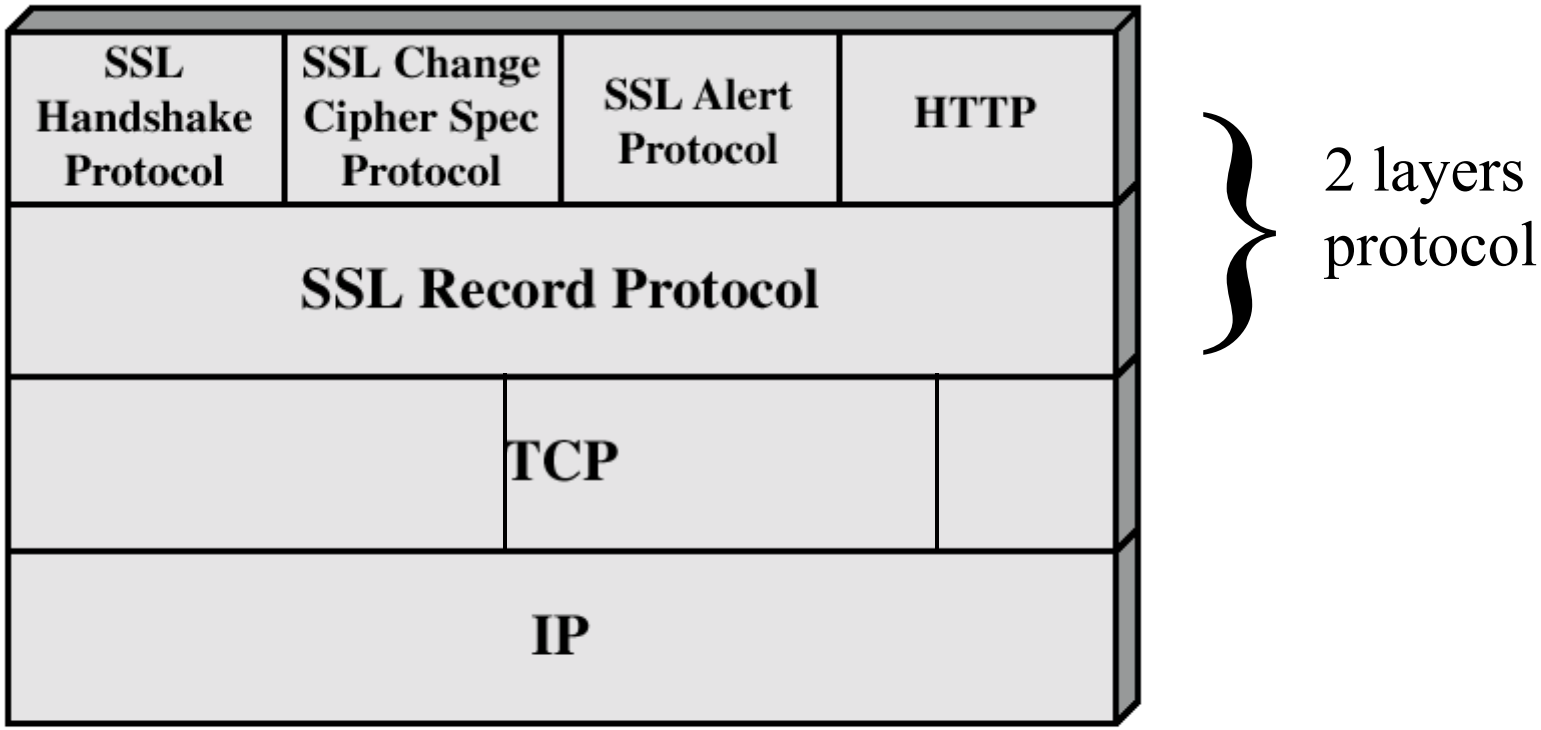
- Protocol that allows to establish an end-to-end secure channel, providing: confidentiality, integrity and authentication
- Defines how the characteristics of the channel are negotiated: key establishment, encryption cipher, authentication mechanism
- Requires reliable end-to-end protocol, so it runs on top of TCP
- It can be used by other session protocols (such as HTTPS)
- Several implementations: for example open source implementation (www.openssl.org)

TLS (cont.)

- **Confidentiality**: Achieved by encryption using DES, 3DES, RC2, RC4, IDEA.
- **Integrity**: Achieved by computing a MAC and send it with the message; MD5, SHA1.
- **Key exchange**: relies on public key encryption.



TLS: Protocol Architecture



Session and Connection

- **Session**:
  - association between a client and a server;
  - created by the Handshake Protocol;
  - defines secure cryptographic parameters that can be shared by multiple connections.
- **Connection**:
  - end-to-end reliable secure communication;
  - every connection is associated with a session.

Session

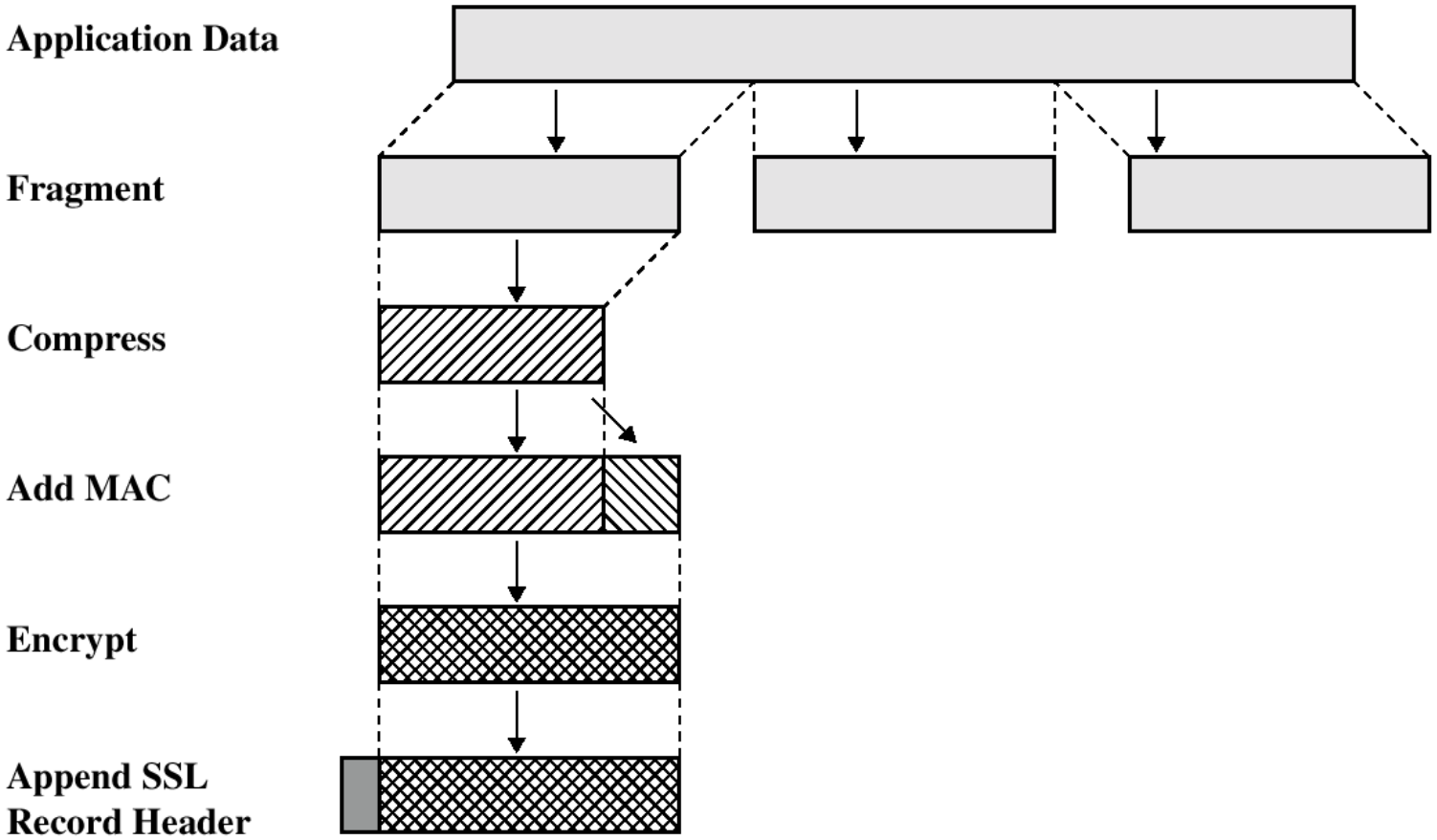
- Session identifier: generated by the server to identify an active or resumable session.
- Peer certificate: X 509v3 certificate.
- Compression method: algorithm used to compress the data before encryption.
- Cipher spec: encryption and hash algorithm, including hash size.
- Master secret: 48 byte secret shared between the client and server.
- Is resumable: indicates if the session can be used to initiate new connections.

Connection

- Server and client random: chosen for each connection.
- Server write MAC secret: shared key used to compute MAC on data sent by the server.
- Client write MAC secret: same as above for the client
- Server write key: shared key used by encryption when server sends data.
- Client write key: same as above for the client.
- Initialization vector: initialization vectors required by encryption.
- Sequence numbers: both server and client maintains such a counter to prevent replay, **cycle is $2^{64} - 1$**.

TLS: SSL Record Protocol

- Provides confidentiality and message integrity using shared keys established by the Handshake Protocol



TLS: SSL Record Protocol

- Fragments have size 16384.
- Compression done such that expansion is not more than 1024 bytes (for small messages, compression might expand data because of alignments.
- Currently in TLS no compression scheme specified.
- The maximum packet size id 16384 + 2048 bytes (1024 from compression, 1024 from HMAC).
- TLS uses the HMAC standard for integrity and authentication.

TLS: Change Cipher Spec Protocol

- One message of one byte containing value 1
- When this message is sent the pending state is copied in the current state

SSL Record Packet

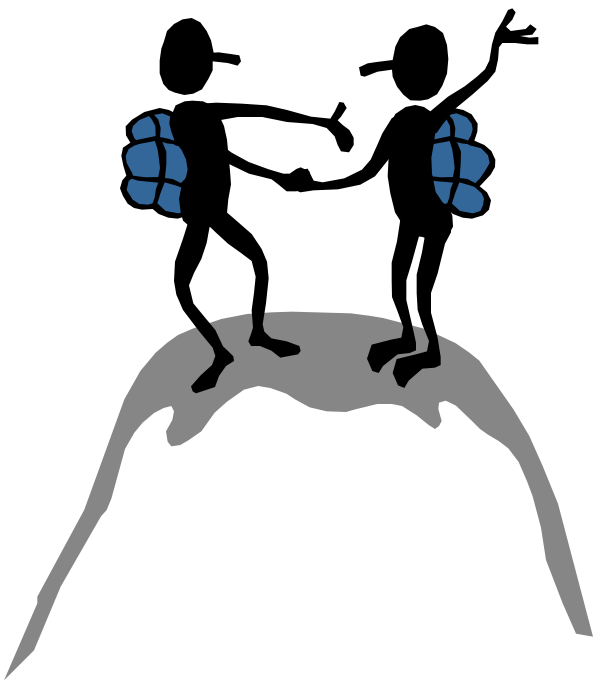
- Header:
  - Content type: what protocol will process the packet (change cipher spec, alert, handshake, application data)
  - Major Version: 3 (for TLS)
  - Minor Version: 0 (for TLS)
  - Compressed length: max is16384 + 2048

Alert Protocol

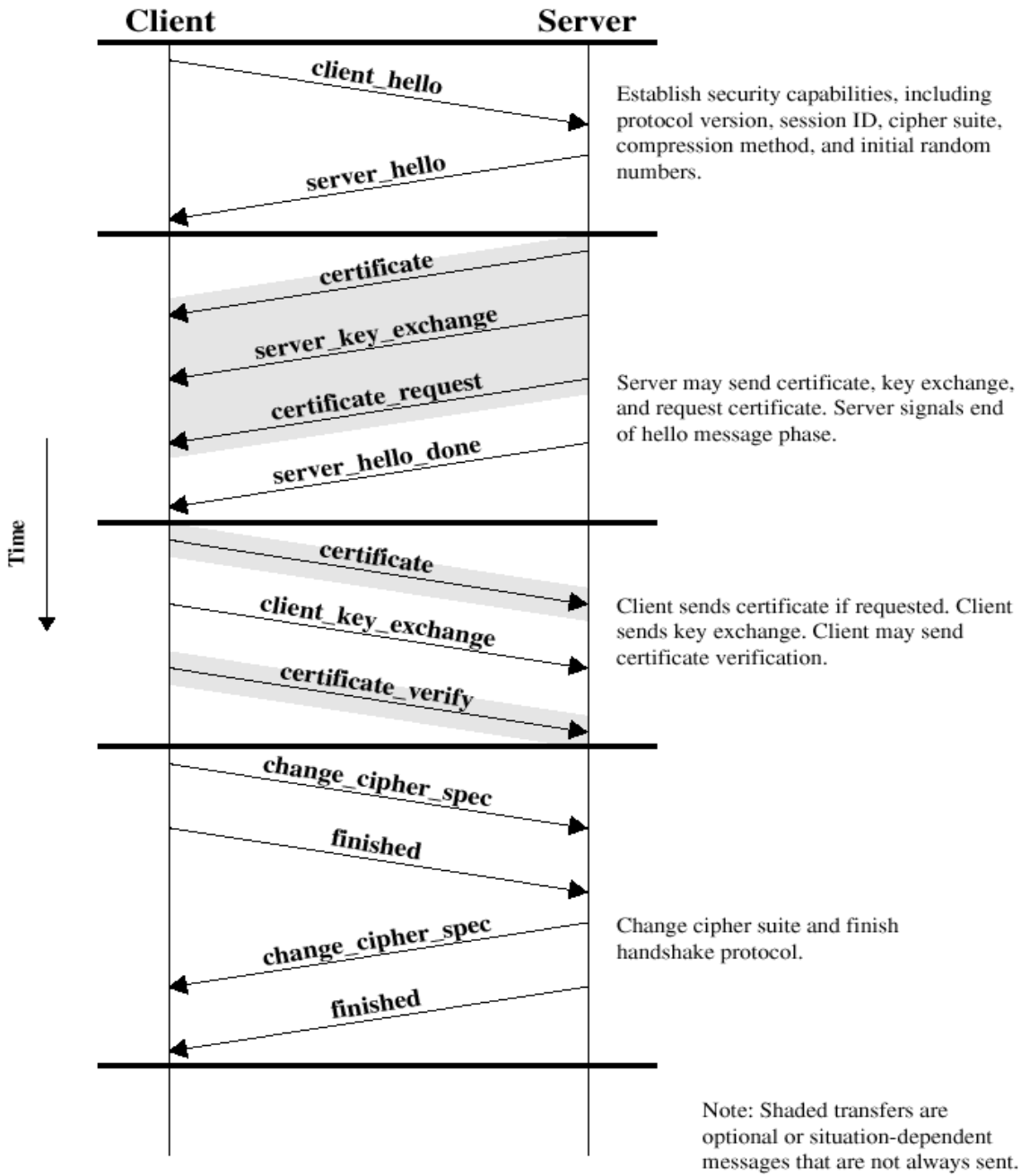
- Used to send TLS related alerts to peers
- **Alert messages are compressed and encrypted**
- Message: two bytes, one defines fatal/warnings, other defines the code of alert
- Fatal errors: decryption_failed, record_overflow, unknown_ca, access_denied, decode_error, export_restriction, protocol_version, insufficient_security, internal_error
- Other errors: decrypt_error, user_cancelled, no_renegotiation

TLS: Handshake Protocol

- Negotiate Cipher-Suite Algorithms
  - Symmetric cipher to use
  - Key exchange method
  - Message digest function
- Establish the shared master secret
- Optionally authenticate server and/or client



Handshake Protocol



Handshake Protocol: Hello

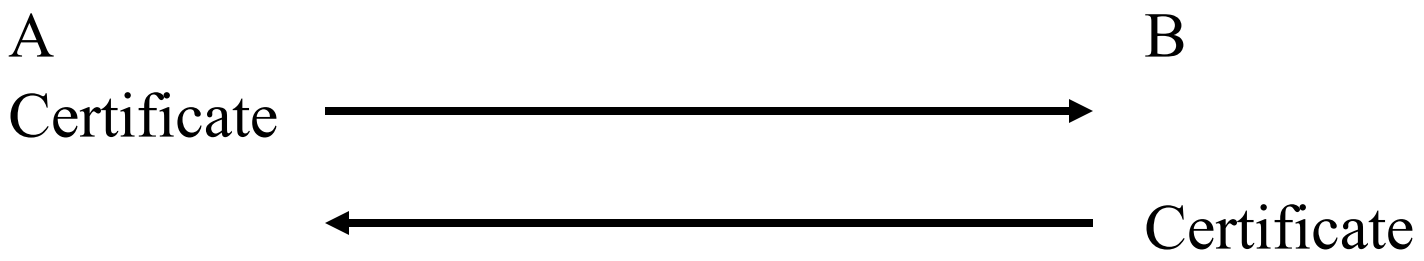
- Client_hello_message has the following parameters:
  - Version
  - Random: timestamp + 28-bytes random
  - Session ID
  - CipherSuite: cipher algorithms supported by the client, first is key exchange
  - Compression method
- Server responds with the same
- Client may request use of cached session
  - Server chooses whether to accept or not

Handshake Protocol: Key Exchange

- Supported key exchange methods:
  - RSA: shared key encrypted with RSA public key
  - Fixed Diffie-Hellman; public parameters provided in a certificate
  - Ephemeral Diffie-Hellman: the best; Diffie-Hellman with temporary secret key, messages signed using RSA or DSS
  - Anonymous Diffie-Hellman: vulnerable to man-in-the-middle

TLS: Authentication

- Verify identities of participants
- Client authentication is optional
- Certificate is used to associate identity with public key and other attributes



TLS: Change Cipher Spec/Finished

- Change Cipher Spec completes the setup of the connections.
- Announce switch to negotiated algorithms and values
- The client sends a message under the new algorithms, allows verification of that the handshake was successful.