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

PCT (1995)  Microsoft

SSLv2 (1994)  Netscape  First release

STLP (1996)  Microsoft

SSLv3 (1995)  Netscape

WTLS (1998)  WAP Forum

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

TLS 1.1 (2006)  

TLS 1.2 (2008)

Source: SSL and TLS, Rescorla
SSL Record Protocol Operation

Figure 7.3 SSL Record Protocol Operation

Source: Network Security Essentials (Stallings)
## SSL Record Format

<table>
<thead>
<tr>
<th>Content Type</th>
<th>Major Version</th>
<th>Minor Version</th>
<th>Compressed Length</th>
</tr>
</thead>
</table>

- Plaintext (optionally compressed)
- MAC (0, 16, or 20 bytes)

*Figure 7.4 SSL Record Format*
RSA Key Exchange Method

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client Hello</strong></td>
<td>[Random_client, Cipher Suites *, SessionID]</td>
</tr>
<tr>
<td><strong>Server Hello</strong></td>
<td>[Random_server, Cipher Suites +, SessionID]</td>
</tr>
<tr>
<td><strong>Server Certificate</strong></td>
<td>chain of X.509 Certs</td>
</tr>
<tr>
<td><strong>Server Hello Done</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Client Key Exchange</strong></td>
<td>[Pre-master secret encrypted with server public key]</td>
</tr>
<tr>
<td><strong>Change Cipher Spec</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Finished</strong></td>
<td>[Encrypted Running Hash]</td>
</tr>
<tr>
<td><strong>Change Cipher Spec</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Finished</strong></td>
<td>[Encrypted Running Hash]</td>
</tr>
</tbody>
</table>
RSA Key Exchange Method

Client

- **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]

Server

- **Client Key Exchange**  [Pre-master secret encrypted with server public key]
- **Certificate Verify**
- **Change Cipher Spec**
- **Finished**  [Encrypted Running Hash]
DHE Key Exchange Method

Client

- **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]

Server

- **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
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)
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.