Module 2.4
SSL Handshake Protocol

Ravi Sandhu

Spring 2021
SSL session negotiated by handshake protocol

- session ID
  - chosen by server
- X.509 public-key certificate of peer
  - possibly null
- compression algorithm
- cipher spec
  - encryption algorithm
  - message digest algorithm
- master secret
  - 48 byte shared secret
- is resumable flag
  - can be used to initiate new connections
  - each session is created with one connection, but additional connections within the session can be further created
SSL Handshake Protocol

Fig. 1 - Message flow for a full handshake

* Indicates optional or situation-dependent messages that are not always sent.

Plaintext: Until and including ChangeCipherSpec message

Encrypted and MAC-ed: All subsequent messages

4-round message exchange
these handshake messages must occur in order

optional messages can be eliminated

change_cipher_spec is a separate 1 message protocol

functionally just like a message in the handshake protocol

post-handshake message to renegotiate

hello_request

can be sent anytime from server to client to request client to start handshake protocol to renegotiate session
SSL 2-Way Handshake with RSA

Phase 1

Client
ClientHello  -------->  Server
ServerHello

Phase 2

Certificate*
CertificateRequest*
ServerKeyExchange*

Phase 3

Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished

Phase 4

[ChangeCipherSpec]
Finished

Application Data
Application Data

Record Protocol

Fig. 1 - Message flow for a full handshake

* Indicates optional or situation-dependent messages that are not always sent.

Plaintext: Until and including ChangeCipherSpec message

Encrypted and MAC-ed: All subsequent messages

4-round message exchange
SSL 2-Way Handshake with RSA

- **Phase 1:**
  - Establish security capabilities
- **Phase 2:**
  - Server authentication
- **Phase 3:**
  - Client authentication and key exchange
- **Phase 4:**
  - Finish
Phase 1: Establish security capabilities

- **client hello message**
  - 4 byte timestamp, 28 byte random value
  - session ID:
    - non-zero for new connection on existing session
    - zero for new connection on new session
  - client version: highest version
  - cipher_suite list: ordered list
    - key exchange method, encryption method, MAC method
  - compression list: ordered list

- **server hello message**
  - 32 byte random value
  - session ID:
    - new or reuse
  - version
    - lower of client suggested and highest supported
  - cipher_suite list: single choice
  - compression list: single choice
Phase 2: Server authentication

- **certificate message**
  - server’s X.509v3 certificate along with chain of certificates
  - required for RSA

- **certificate request message**
  - request a certificate from client
  - specifies Certificate Type and Certificate Authorities

- **server done message**
  - ends phase 2, always required
Phase 3: Client authentication and key exchange

- certificate message
  - client’s X.509v3 certificate along with chain of certificates
- client key exchange message
  - client generates 48-byte pre-master secret, encrypts with server’s RSA public key
- certificate verify message
  - signs hash of master secret (established by key exchange) and all handshake messages so far

- client and server compute 48 byte master secret
  - using 48-byte pre-master secret, ClientHello.random, ServerHello.random
- client and server compute 4 symmetric keys from master secret
Phase 4: Finish and move to record protocol

- change cipher spec message
  - not considered part of handshake protocol but in some sense is part of it
  - 1 byte message

- Finished message
  - sent under new algorithms and keys
  - content is MAC of all previous messages with master secret and constant “client finished” or “server finished”
SSL 2-way RSA Handshake Result

Client (Browser)

Key 1 for MAC
Key 2 for encrypt

Server

Key 3 for MAC
Key 4 for encrypt

RSA signature certificate

RSA encryption certificate

Mutually authenticated secure channel
## SSL 1-Way Handshake with RSA

### Phase 1

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientHello</td>
<td>--------&gt;</td>
</tr>
<tr>
<td>ServerHello</td>
<td>Certificate*</td>
</tr>
<tr>
<td>Certificate*</td>
<td>ServerKeyExchange*</td>
</tr>
<tr>
<td>CertificateRequest*</td>
<td>ServerHelloDone</td>
</tr>
</tbody>
</table>

### Phase 2

| Certificate* |
| ClientKeyExchange |
| CertificateVerify* |

### Phase 3

| [ChangeCipherSpec] |
| Finished |

### Phase 4

| ChangeCipherSpec |
| Finished |

| Application Data | Application Data |

---

**Fig. 1** - Message flow for a full handshake

* Indicates optional or situation-dependent messages that are not always sent.

---

**Plaintext:** Until and including ChangeCipherSpec message

**Encrypted and MAC-ed:** All subsequent messages

**4-round message exchange**
Phase 1:
- Establish security capabilities

Phase 2:
- Server authentication

Phase 3:
- Client authentication and key exchange

Phase 4:
- Finish
Phase 1: Establish security capabilities

- **client hello message**
  - 4 byte timestamp, 28 byte random value
  - session ID:
    - non-zero for new connection on existing session
    - zero for new connection on new session
  - client version: highest version
  - cipher_suite list: ordered list
    - key exchange method, encryption method, MAC method
  - compression list: ordered list

- **server hello message**
  - 32 byte random value
  - session ID:
    - new or reuse
  - version
    - lower of client suggested and highest supported
  - cipher_suite list: single choice
  - compression list: single choice

No change relative to SSL RSA 2-way Handshake Phase 1
Phase 2: Server authentication

- certificate message
  - server’s X.509v3 certificate along with chain of certificates
  - required for RSA

- certificate request message
  - request a certificate from client
  - specifies Certificate Type and Certificate Authorities

- server done message
  - ends phase 2, always required
Phase 3: Client authentication and key exchange

- **Certificate message**
  - Client’s X.509v3 certificate along with chain of certificates

- **Client key exchange message**
  - Client generates 48-byte pre-master secret, encrypts with server’s RSA public key

- **Certificate verify message**
  - Signs hash of master secret (established by key exchange) and all handshake messages so far

- Client and server compute 48-byte master secret
  - Using 48-byte pre-master secret, ClientHello.random, ServerHello.random

- Client and server compute 4 symmetric keys from master secret
Phase 4: Finish and move to record protocol

- change cipher spec message
  - not considered part of handshake protocol but in some sense is part of it
  - 1 byte message

- Finished message
  - sent under new algorithms and keys
  - content is MAC of all previous messages with master secret and constant “client finished” or “server finished”

No change relative to SSL RSA 2-way Handshake Phase 4
SSL 1-way RSA Handshake Result

Server-to-client authenticated secure channel

Client-to-server authentication via user ID and password