Trust Infrastructure of SSL

CS 470
Introduction to Applied Cryptography
Ali Aydın Selçuk

SSL/TLS

• The main workhorse of secure Internet communication.

• Everyday, billions of web packets (HTTPS) are encrypted by SSL/TLS.

• Not only web pages: VPN tunneling, electronic banking, cloud services, … all rely on SSL to secure their communications.

Success of SSL

• Brought cryptography to the service of the masses

• Until SSL and spread of Internet, cryptography saw very limited use by common people.

• In the first 20 years of PKC (1976-96), the technology had a very limited penetration.

• This all changed in the second half of the 1990s with SSL.

Success of SSL

• Trust infrastructure has an autonomous and self-governing structure, consisting of
  – browser vendors
  – audit firms and standards bodies
  – certificate authorities
  – SSL servers

• Has been remarkably successful, especially compared to previous efforts such as PEM to secure Internet communications.
A Simple Key Exchange Protocol

~ SSL key exchange protocol:

<table>
<thead>
<tr>
<th>Client</th>
<th>Hello</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pub key (K_s)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(k)</td>
<td></td>
</tr>
</tbody>
</table>

\(k\) is the session key

Active Attacks & Certificates

- Simple public key encryption solves the key distribution problem against passive attackers (i.e., an attacker that just eavesdrops).
- Active attackers can send a fake public key & become a “man in the middle” (MitM).

Notation:
- \([M]_X\) : message \(M\) signed with the prv. key of \(X\)
- \(\{M\}_X\) : message \(M\) enc. with the pub. key of \(X\)

MitM Attack

Normal op:

<table>
<thead>
<tr>
<th>Alice</th>
<th>pub key (K_A)</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(k)</td>
<td></td>
</tr>
</tbody>
</table>

\(k\) is the session key

MitM attack:

<table>
<thead>
<tr>
<th>Alice</th>
<th>Trudy</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>(k)</td>
<td>(k)</td>
</tr>
<tr>
<td>key (k)</td>
<td>key (k)</td>
<td>key (k)</td>
</tr>
</tbody>
</table>

Certificates

- These MitM attacks are possible because a receiver cannot distinguish a fake public key from the real one.
- Certificates: IDs and public keys are signed by a trusted authority (“certification authority”).
- E.g., \(\text{cert}_A = [\text{ID}_A, \text{PK}_A, \text{exp.date}, \ldots ]_\text{CA}\)
Certified Encrypted Key Exchange

SSL key exchange protocol:

```
\[
\text{client} \rightarrow \text{server} \quad \text{hello} = [ID_S, PK_S, \ldots, k_{CA}]
\]
\[
\text{cert}_S = [ID_S, PK_S, \ldots, k_{CA}]
\]
```

\( k \) is the session key

Certification Authorities

- CAs’ public key should be distributed in a trusted way to all the parties in the system in advance.
- In SSL, root CAs are approved by the browser (or the OS) makers, and distributed with the browser/OS code.
- CAs must satisfy certain criteria for this:

- Browser makers require CA firms to be audited and accredited according to some standards:
  - WebTrust
  - ETSI TS 101/102
  - ISO 21188:2006
- Public key infrastructure of SSL:
  - Oligarchy model: A number of trusted root CAs, which issue certificates to intermediate CAs, or to end users (SSL servers)

Example: IE Browser

- Tools > Internet options > Content > Certificates
Example: IE Browser

• Trusted root CAs:

E.g., VeriSign root certificate:

Example: IE Browser

• Untrusted / revoked certificates:

Certificates & Validation

• Valid SSL/TLS certificates are issued to web servers by root or intermediate CAs.
  – E.g., Google’s certificate: GeoTrust (root) → Google Internet Authority → accounts.google.com
• Client (browser) authenticates this chain of certificates beginning from the root CA.
Example Client Certificate

- E.g., gmail.com (or, accounts.google.com)

SSL/TLS in Practice

- A reasonably secure protocol
- with a reasonable trust model
- and commercially viable operation

What may go wrong?

- “Man in the browser” attacks
- Cert. validation software may get it wrong
- Compromised CAs, fake certificates
- and more…