Trust Infrastructure of SSL

BİL 448/548
Internet Security Protocols
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

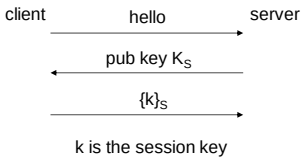
- Brought cryptography to the service of the masses
- Until SSL and spread of Internet, cryptography saw very limited use by common people.
- In the first 20 years of PKC (1976-96), the technology had a very limited penetration.
- This all changed in the second half of the 1990s with SSL.

Success of SSL

- Trust infrastructure has an autonomous and self-governing structure, consisting of
  - browser / OS 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:



Active Attacks & Certificates

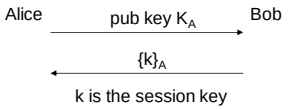
- Simple public key encryption solves the key distribution problem against passive attackers (i.e., an attacker that just eavesdrops).
- Active attackers can send a fake public key & become a “man in the middle” (MitM).

Notation:

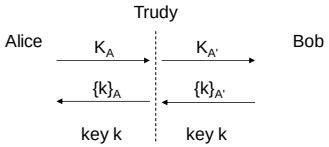
- $[M]_X$: message M signed with the prv. key of X
- $\{M\}_X$: message M enc. with the pub. key of X

MitM Attack

Normal op:



MitM attack:

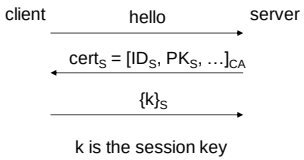


Certificates

- 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., $cert_A = [ID_A, PK_A, exp.date, \ldots]_{CA}$

Certified Encrypted Key Exchange

~ SSL key exchange protocol:



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:
  - https://wiki.mozilla.org/CA:How_to_apply
  - http://www.chromium.org/Home/chromium-security/root-ca-policy
  - http://technet.microsoft.com/en-us/library/cc751157.aspx

Certification Authorities

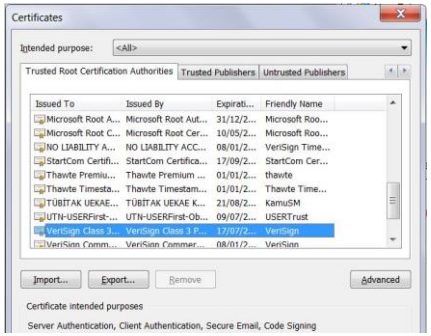
- Browser makers require CA firms to be audited and accredited according to some standards:
  - WebTrust
  - ETSI TS 101/102
  - ISO 21188:2006
- Public key infrastructure of SSL:
  - Oligarchy model: A number of trusted root CAs,
  - which issue certificates to intermediate CAs, or to end users (SSL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

Example: IE Browser

- Tools > Internet options > Content > Certificates

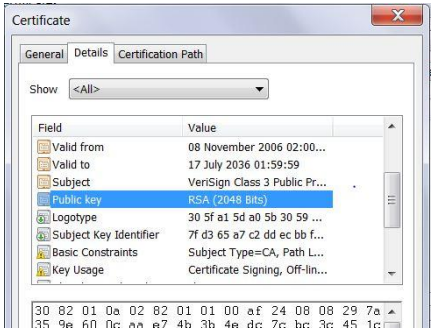
Example: IE Browser

- Trusted root CAs:



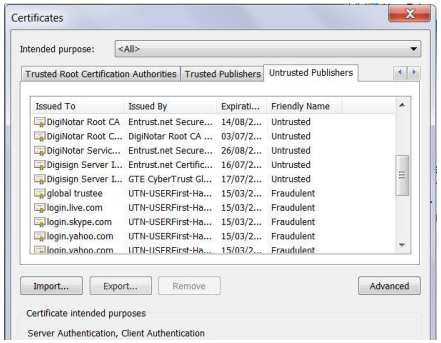
Example: IE Browser

- E.g., VeriSign root certificate:



Example: IE Browser

- Untrusted / revoked certificates:



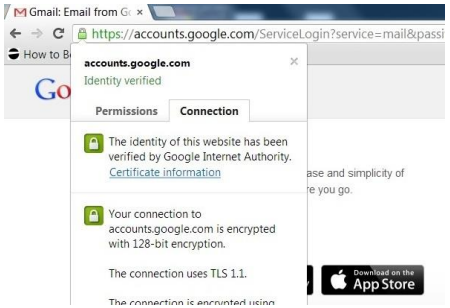
Certificates & Validation

- 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.

http://en.wikipedia.org/wiki/Certification_path_validation_algorithm

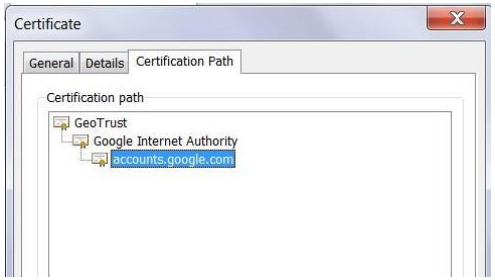
Example Client Certificate

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



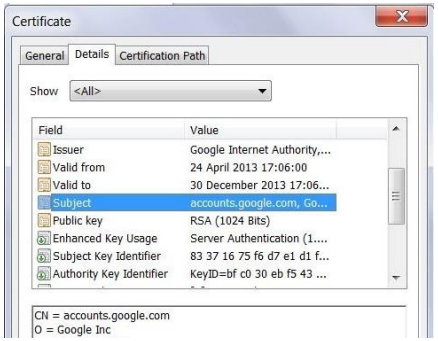
Example Client Certificate

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



Example Client Certificate

- Example Client Certificate



SSL/TLS in Practice

SSL/TLS:

- 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…