TLS/SSL

- TLS (Transport Layer Security)
  - A suite of protocols to provide secure communication
    - Confidentiality by applying block & stream ciphers
    - Integrity with MACs
    - Authenticity with certificates
- Predecessor: SSL (secure sockets layer)
  - TLS was proposed as an upgrade
  - All versions of SSL are considered insecure (recently, the POODLE—padding oracle—attack)

TCP/IP: Host A and B can send packets to one another.

TLS/SSL: operate “over” TCP/IP to ensure security/authenticity.
TLS/SSL protocol (high level)

**Browser** (initiates connection)

- **Client hello**
  - Version, crypto options, *nonce*

**Server** (authenticates itself)

- **Server hello + server cert (PKs)**
  - Version, crypto options, *nonce*,
    - Signed certificate containing
      - the server’s public key PKs

- **Server key exchange (when using DH)**

- **Client key exchange**
  - PreMaster secret encrypted with server’s PKs

- **Switch to negotiated cipher**

**Data transmission**

Compute K based on nonces & PreMaster

Compute K based on nonces & PreMaster
HTTPS

TCP/IP: Host A and B can send packets to one another

TLS/SSL: operate “over” TCP/IP to ensure security/authenticity

HTTPS is HTTP operating on top of TLS/SSL

Why not HTTPS everywhere?
HTTPS everywhere?

• Takes more time to initiate connections

• In-network services want to look at the traffic
  • To compress it (cannot compress encrypted traffic)
  • To cache it (especially if it’s static content)
  • To stick their own ads into it (i.e., there is pushback)
  • Any other ideas?

• Google has moved its services over to https (even the ones you’re not logged into)
  • Didn’t want others “transcoding” (reducing the quality of) their videos, or sticking in their own ads
Certificates in the wild

The lock icon indicates that the browser was able to authenticate the other end, i.e., validate its certificate.
Certificate chain

**Subject** (who owns the public key)

- **Common name:** the URL for which this cert is valid (can contain wildcards, e.g., *.wellsfargo.com)

**Issuer** (who verified the identity and signed this certificate)
Wildcard certificates

Certificates with wildcards are “wildcard certs”. Wildcards (*) only match a single level of a domain

*.bar.com
✔️ foo.bar.com
✘ not.this.bar.com

The X.509 protocol (which defines the format of these certificates) advises wildcard policies

(e.g., don’t allow *.*
Serial number: Uniquely identifies this cert with respect to the issuer (look for this in CRLs)

Signature algorithm: How the issuer will sign parts of the cert

Not valid before/after: When to start and stop believing this cert (start & expiration dates)

The public key: And the issuer’s signature of the public key
Subject Alternate Names:
Other URLs for which this cert should be considered valid.
(wellsfargo.com is not the same as www.wellsfargo.com)

Can include wildcards, e.g., *.google.com

CRL & OCSP:
Where to go to check if this certificate has been revoked

Non-cryptographic checksums
Root certificates
### Keychain Access

Click to unlock the System Roots keychain.

<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
<th>Expires</th>
<th>Keychain</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeriSign Class 3 Public Primary Certification Authority - G5</td>
<td>certificate</td>
<td>Jul 16, 2036, 7:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Trust2408 OCES Primary CA</td>
<td>certificate</td>
<td>Dec 3, 2037, 8:11:34 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Trusted Certificate Services</td>
<td>certificate</td>
<td>Dec 31, 2028, 6:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Trustis FPS Root CA</td>
<td>certificate</td>
<td>Jan 21, 2024, 6:36:54 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>TÜBİTAK UEKA Kök S...izmet Sağlayıcısı - Sürüm 3</td>
<td>certificate</td>
<td>Aug 21, 2017, 7:37:07 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı</td>
<td>certificate</td>
<td>Sep 16, 2015, 6:07:57 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı</td>
<td>certificate</td>
<td>Dec 22, 2017, 1:37:19 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>TWCA Global Root CA</td>
<td>certificate</td>
<td>Dec 31, 2030, 10:59:59 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>TWCA Root Certification Authority</td>
<td>certificate</td>
<td>Dec 31, 2030, 10:59:59 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UCA Global Root</td>
<td>certificate</td>
<td>Dec 30, 2037, 7:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UCA Root</td>
<td>certificate</td>
<td>Dec 30, 2029, 7:00:00 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UTN - DATACorp SGC</td>
<td>certificate</td>
<td>Jun 24, 2019, 3:06:30 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UTN-USERFirst-Client Authentication and Email</td>
<td>certificate</td>
<td>Jul 9, 2019, 1:36:58 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UTN-USERFirst-Hardware</td>
<td>certificate</td>
<td>Jul 9, 2019, 2:19:22 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UTN-USERFirst-Network Applications</td>
<td>certificate</td>
<td>Jul 9, 2019, 2:57:49 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>UTN-USERFirst-Object</td>
<td>certificate</td>
<td>Jul 9, 2019, 2:40:36 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VAS Latvijas Pasts SSI(RCA)</td>
<td>certificate</td>
<td>Sep 13, 2024, 5:27:57 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VeriSign Class 1 Public...Certification Authority - G3</td>
<td>certificate</td>
<td>Jul 16, 2036, 7:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VeriSign Class 2 Public...Certification Authority - G3</td>
<td>certificate</td>
<td>Jul 16, 2036, 7:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VeriSign Class 3 Public...Certification Authority - G4</td>
<td>certificate</td>
<td>Jan 18, 2038, 6:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VeriSign Class 4 Public...Certification Authority - G3</td>
<td>certificate</td>
<td>Jul 16, 2036, 7:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VeriSign Universal Root Certification Authority</td>
<td>certificate</td>
<td>Dec 1, 2037, 6:59:59 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Visa eCommerce Root</td>
<td>certificate</td>
<td>Jun 23, 2022, 8:16:12 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>Visa Information Delivery Root CA</td>
<td>certificate</td>
<td>Jun 29, 2025, 1:42:42 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>VRG Kov. Root CA</td>
<td>certificate</td>
<td>Dec 18, 2023, 8:51:08 AM</td>
<td>System Roots</td>
</tr>
<tr>
<td>WellsSecure Public Root Certificate Authority</td>
<td>certificate</td>
<td>Dec 13, 2022, 7:07:54 PM</td>
<td>System Roots</td>
</tr>
<tr>
<td>XRamp Global Certification Authority</td>
<td>certificate</td>
<td>Jan 1, 2035, 12:37:19 AM</td>
<td>System Roots</td>
</tr>
</tbody>
</table>

213 items
Certificate types

Why are these different?

This is an EV (extended validation) certificate; browsers show the full name for these kinds of certs.
Proper reaction to Heartbleed

1. Patch the software

2. “Reissue” a new key (get a new one and load it onto your servers)

3. Revoke the old key

Order matters!
If we reissued and then patched, then our new key would be compromised, too.

If we revoked first, we’d be offline.