Certificates

- Alternative:
  - Trust on first use
  - When we receive a public key for the first time, we assume there is no attack and then remember it
  - Like when you ssh for the first time and you have to answer yes / no
- We want to avoid having to do this TOFU every time

- Certificate authority:
  - Stores mapping from server names to public keys
  - Provides signed certificates to the server that the server can send with its public key
  - The certificate verifies that this server truly corresponds to that public key
- CA doesn't talk directly with the client; it just signs things and the server is in charge of sending it out

Issuing Certificates

- Domain validator
- ACME: challenge to prove control over domain

1. Server generates PK and SK
2. Server requests a certificate from CA
3. CA challenges by asking the server to put a specific nonce on their website
4. You put the file there and then tell the CA you are ready
5. CA then checks this URL to see if that nonce was put in

- This is known as domain-validated certificates
  - Depends on DNS / IP working properly
  - These are designed to defend against people who are able to subvert DNS / IP
  - Good CAs will access the domain from multiple points to ensure that they are not being fooled by local DNS / IP re-routing
- Alternative: extended validation
  - CA vets not just domain control but tries to ensure they are communicating with a legitimate representative
  - Very expensive and most users don't know the difference

Auditing

- CAs store receipts for all certificates issued so people can go back and look at which certificates are issued in their name
  - Certificate Transparency Log
- Server operators can see if anyone is issuing certificates on their behalf / if their certificate was able to make it out in time
- Allows operators to catch these problems after the fact even if the CA wasn't able to find it in time

CA Trustworthiness

- Problem: some CAs can be untrustworthy
- There are hundreds of CAs and they can each generate certificates for any website
  - Multiple CAs can have certificates which is desirable if a website wants to switch CAs
  - But attackers can trick CAs if they are sloppy / maybe bribe / coerce them

Revoking Certificates

- First tool: certificates have expiration dates
- How does a CA revoke a certificate if all it does is give out certificates and clients don't talk to the CA?
- Answer 1: CA issues a CRL which are sent daily and include a list of all revoked certificates
  - They can also be sent immediately after a certificate has been revoked
  - Problem: often incredibly large and browsers ignore them
- Answer 2: CA runs a service called OCSP (online certificate status protocol) that you can query for whether a certificate is valid
  - Can be down, causing availability to suffer
  - Browsers can cache these responses for a week but then that's the problem of not checking very often
- New solution:
  - Proprietary, browser-specific CRLs
    - Examples include Mozilla's CRLite and Chrome's CRLSets
  - Basic concept involves browser vendors centrally downloading and processing CRLs into smaller formats like Bloom filters
  - Compressed objects then pushed to all browser instances via rapid update mechanisms
    - Firefox updates pushed as frequently as every 6 hours
  - Enables browsers to download revocation lists in advance, maintaining fast page loads
  - Localizes revocation checks, bypassing delays associated with OCSP caches
  - Immediate updates take effect without waiting for OCSP cache expiration
- A New Life for Certificate Revocation Lists - Let's Encrypt