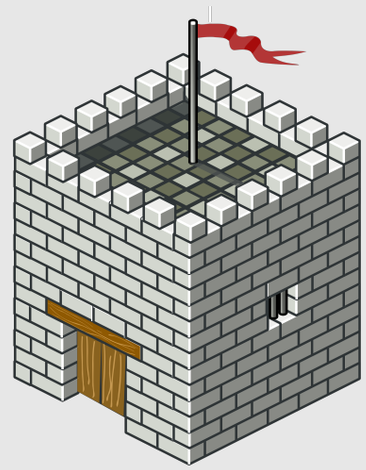
Foundations of Cybersecurity

XI-Public-Key Infrastructures

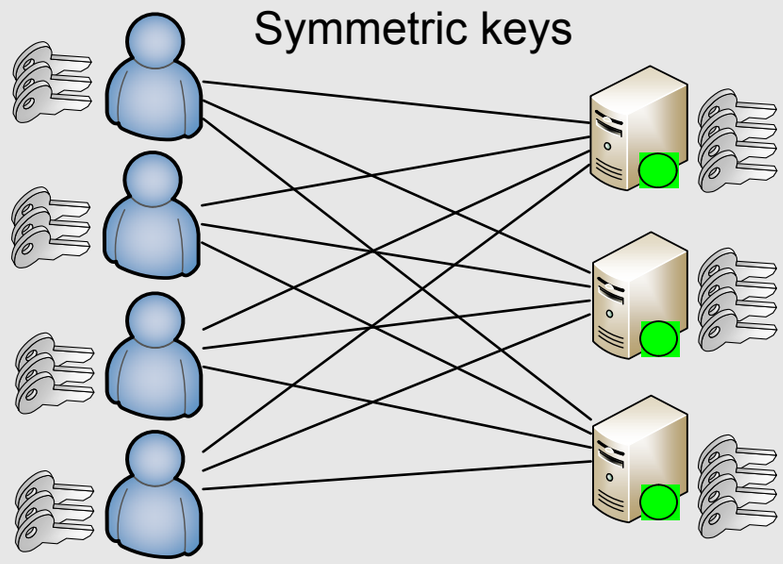


Paweł Szałachowski

2017

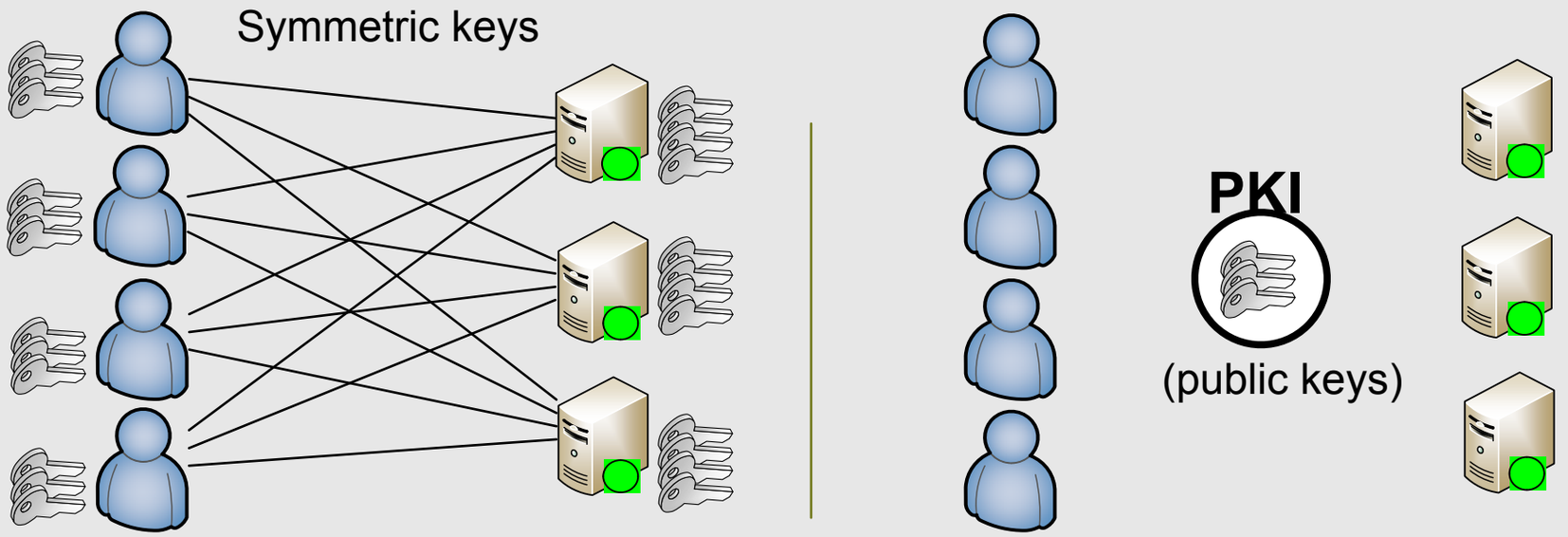
Symmetric Keys (scalability)

- Scalability issues with symmetric crypto
  - Distribution
  - Challenges in managing *n* secrets



Public Key Infrastructure (PKI)

- Scalability issues with symmetric crypto
  - Distribution
  - Challenges in managing *n* secrets
- Asymmetric crypto (DH, RSA, …) solves the scalability problems, ... but creates a new one:
  - **How to ensure that public-key is accessible and authentic ?**



Trust and Trust Models

- Needed to solve scalability issues
- Trust Models
  - Decentralized
  - Centralized
    - Monopoly (Monarchy)
    - Oligopoly (Oligarchy)

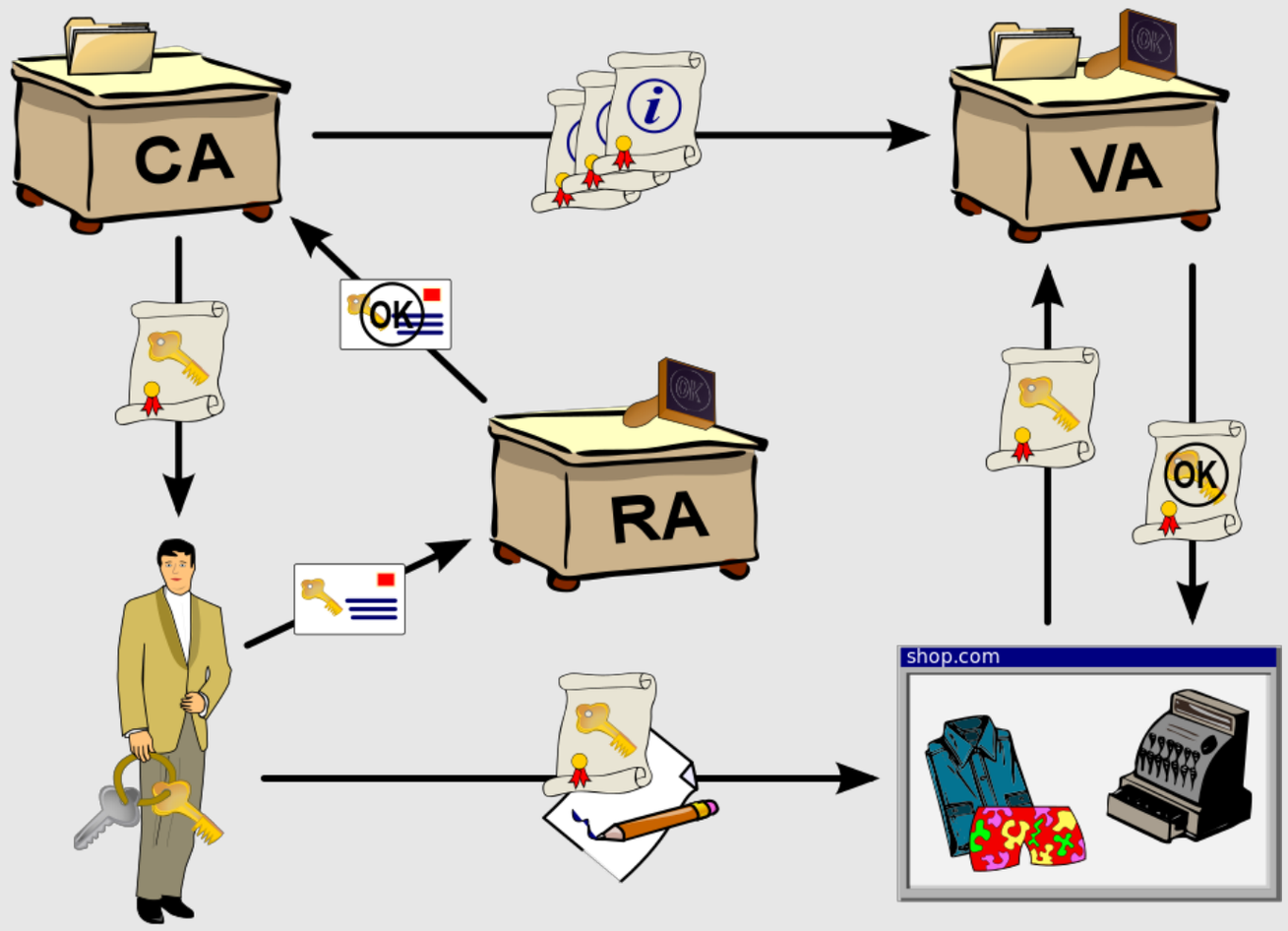
Public-Key Infrastructure (PKI)

- An infrastructure that allows to recognize which public key belongs to whom
- There is a central authority, called the *Certificate Authority* (CA)
  - Everyone trusts the CA and knows its public key

- Alice to join the PKI
  - Generates public/private key pair (PK_A,PK_A^-) and contacts the CA
  - The CA verifies her identity and issues a signed *certificate* that claims that "PK_A belongs to Alice"
  - Alice can now contact Bob sending PK_A and the certificate
  - As Bob trusts the CA, he trusts the certificate

Operations

- Registration Authority
  - verifies identities
- Certificate Authority
  - issues certificates
- Validation Authority
  - informs if a certificate is valid



PKI Examples

- SSL/TLS
  - Web (HTTPS), e-mail, ...
- DNSSEC
- Credit Card Organizations
- Enterprises, companies, organizations, ...

Certificate

- Encoding of a particular data structure **must** be unique
  - ASN.1, canonical JSON/XML/CBOR, ...
- Fields
  - Subject: owner of the certificate
  - Issuer: issuer (CA) of the certificate
  - Not Before: the earliest time on which the certificate is valid
  - Not After: the latest time on which the certificate is valid
  - Public key: public key of the subject
  - Signature: signature of the certificate by the issuer's private key
- Other fields like serial number, key usage, algorithm id, ...

Multilevel Certificates

- For operational reasons certificates form chains
  - Root certificate (trust anchor)
    - self-signed certificate used for signing other certificates
  - Intermediate certificate
    - not self-signed used for signing other certificates
  - Leaf certificate
    - cannot be used for signing other certificates

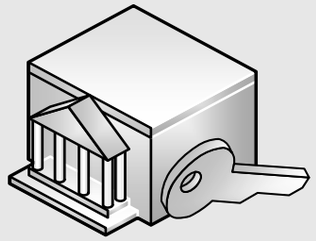
Certificate Revocation

- Sometimes a certificate has to be invalidated (revoked) by the issuing CA
  - How to do this? (One of the hardest problems to solve in a PKI.)
  - What if root/intermediate certificate has to be revoked? (Collateral damage.)

- Requirements
  - Speed of revocation
  - Reliability of revocations
  - Overheads
  - Connectivity

Current SSL/TLS PKI model

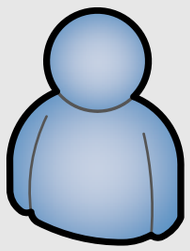
- CA knows whether the certificate is valid or not



CA

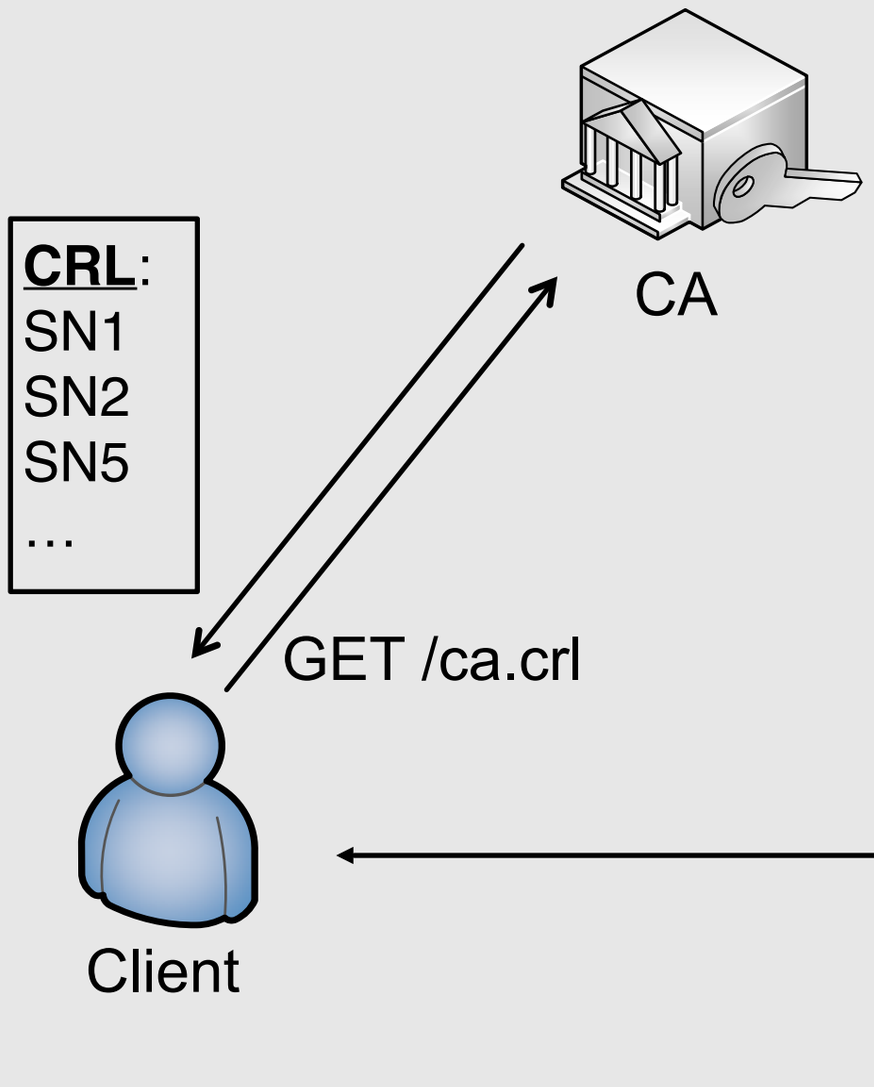
Is the certificate valid?

- *CA*
- *Name*
- *Signatures*
- *Expiration*
- ***Revocation***



Client

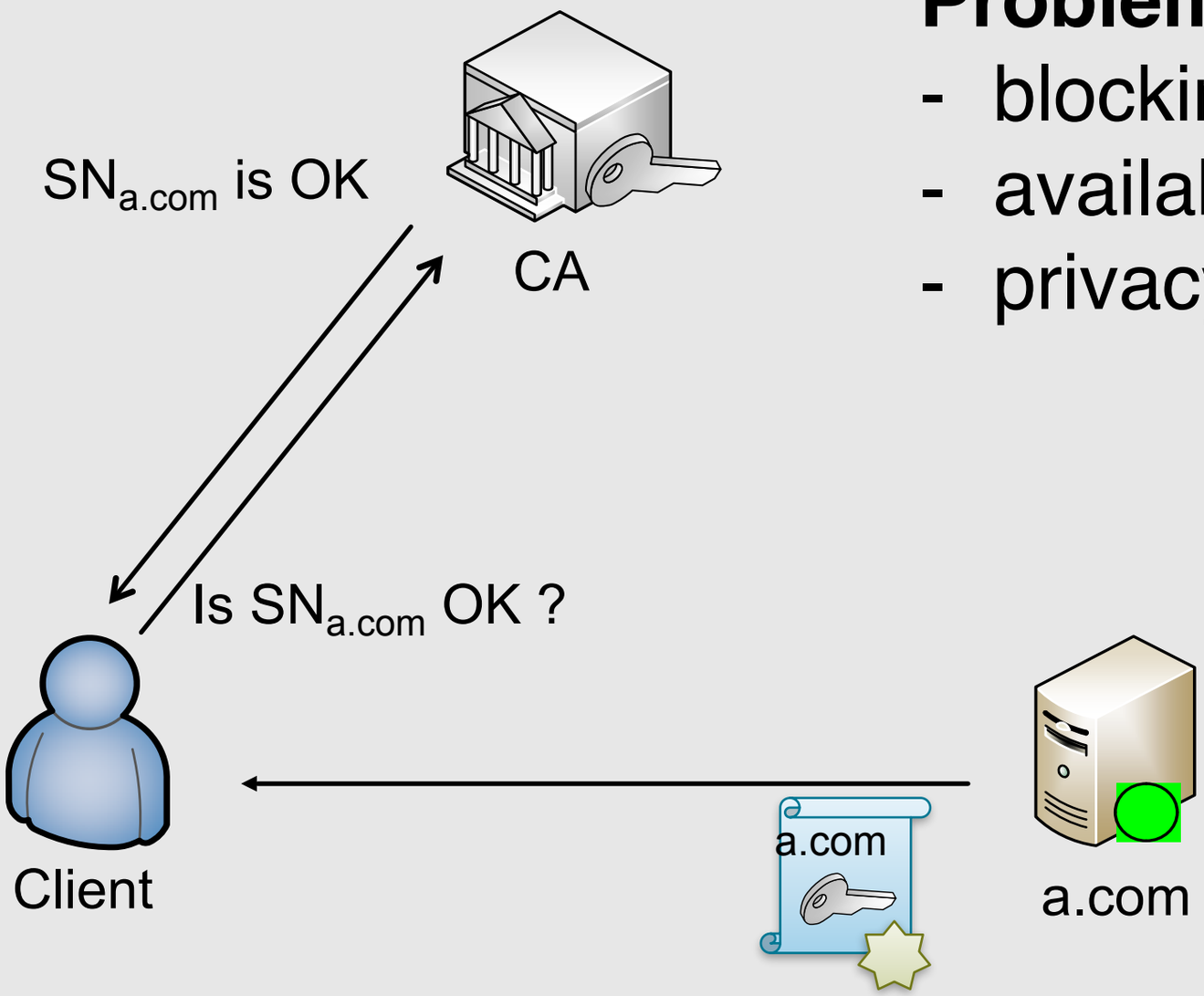
Certificate Revocation List (CRL)



Problems:

- blocking connection
- CRLs are big (expensive)
- 14% CRLs are unavailable
- privacy
- slow revocation after key compromise/loss

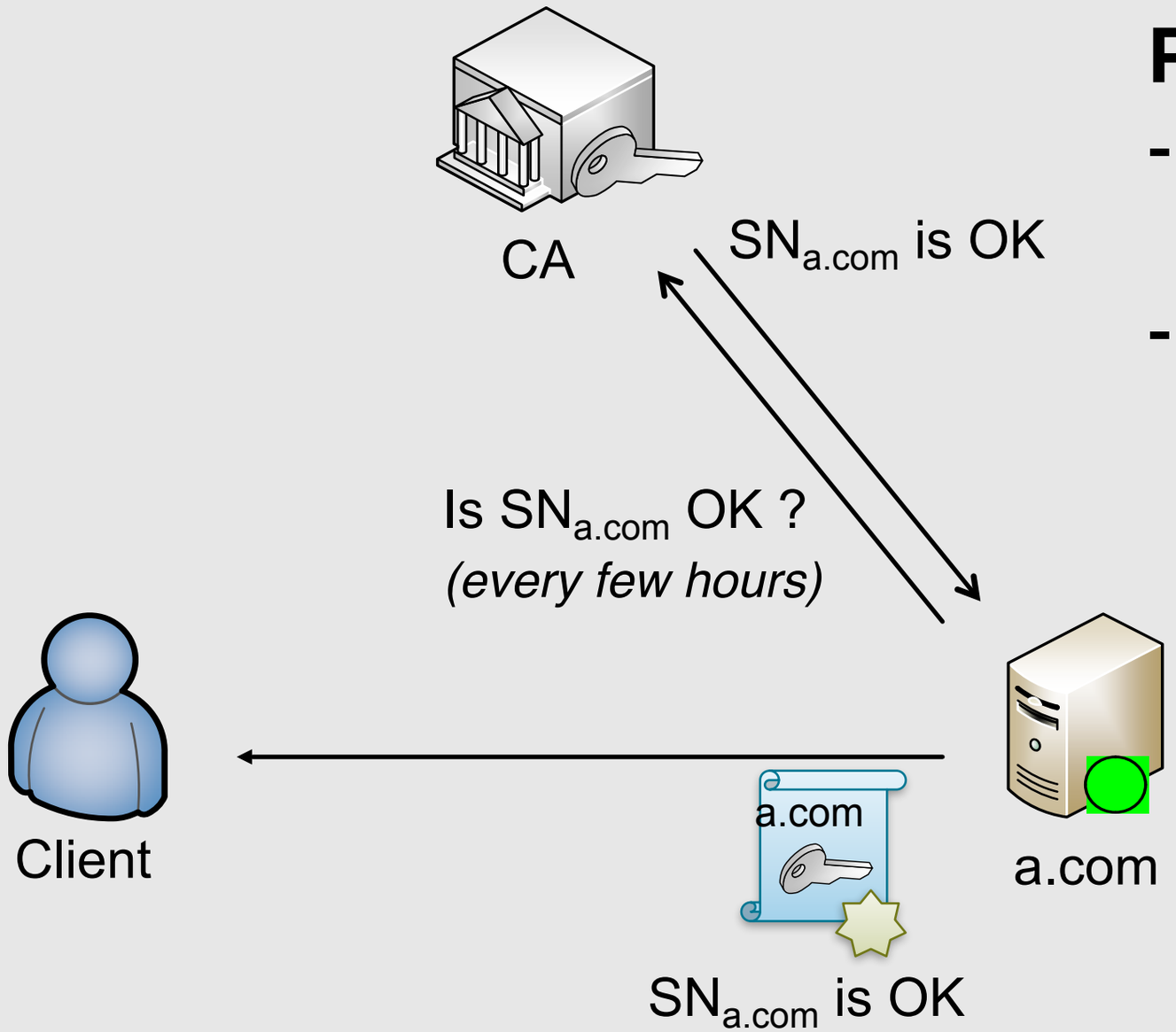
Online Certificate Status Protocol (OCSP)



Problems:

- blocking connection (~350ms)
- availability (18% timeouts)
- privacy

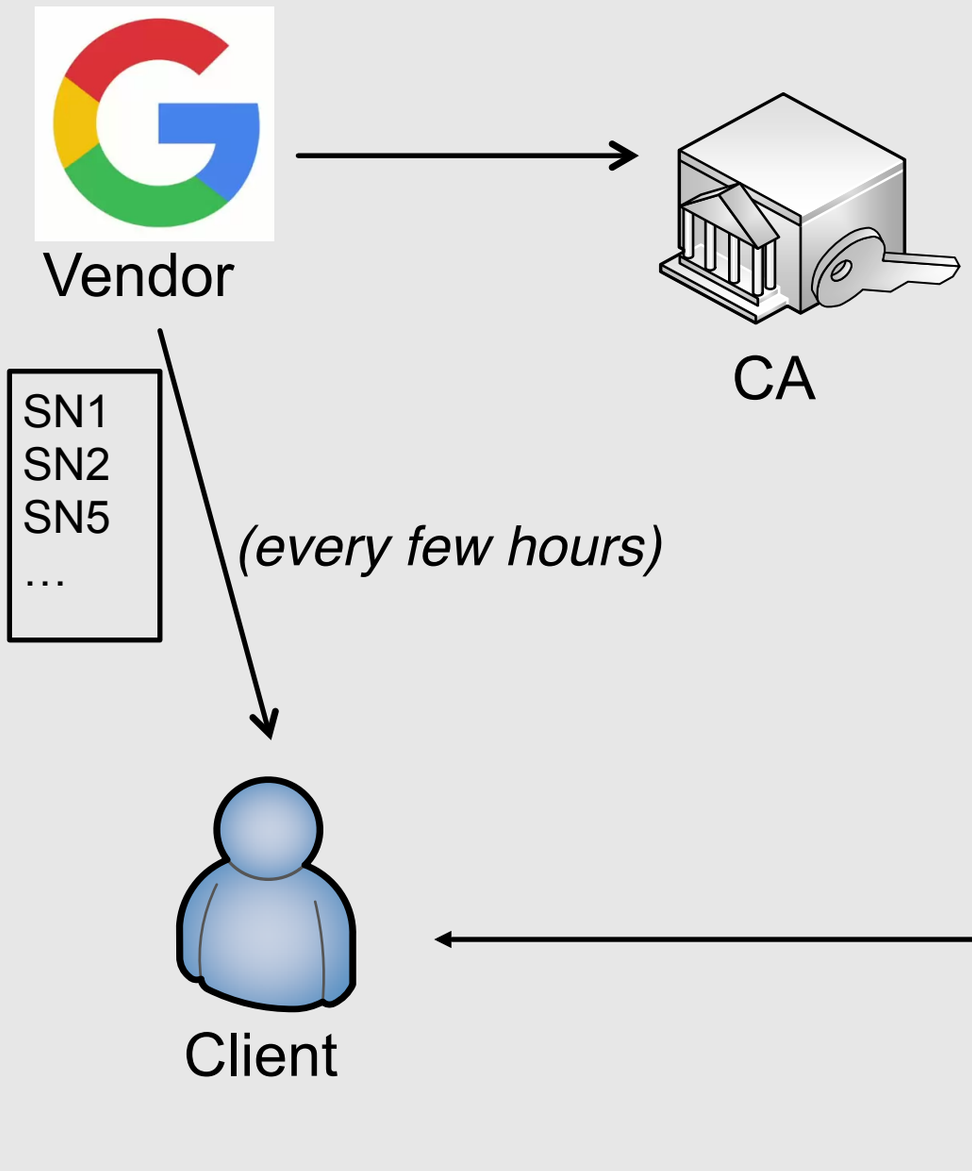
OCSP Stapling



Problems:

- minimal server deployment < 3%
- slow revocation after key compromise/loss

CRLSets



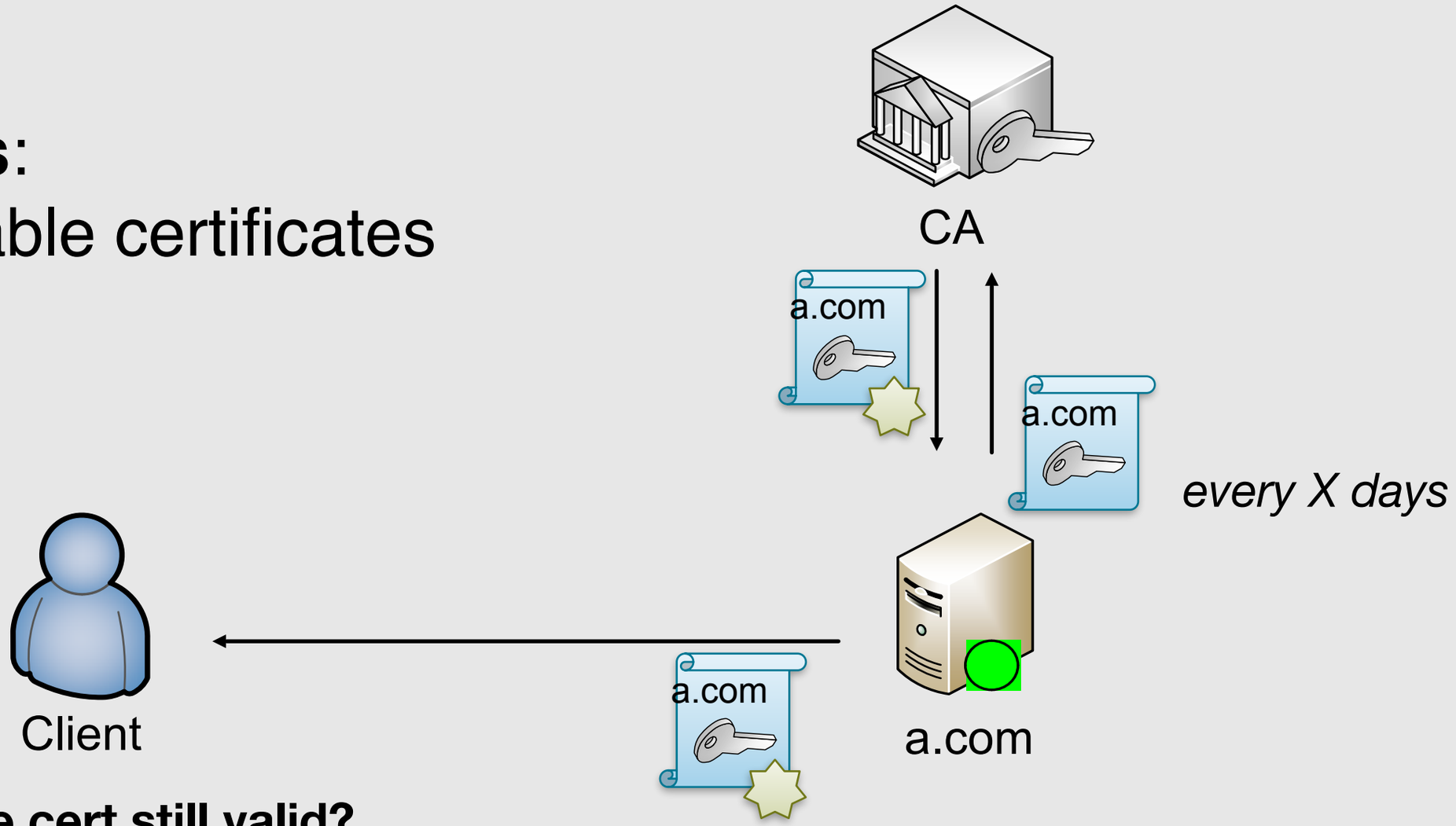
Problems:

- CRLSet is max 250KB
- only 0.35% of all revocations are included
- slow revocation after key compromise/loss

Short-lived Certificates

Problems:

- irrevocable certificates



Is the cert still valid?

Current state

		Desktop Browsers										Mobile Browsers			
		Chrome 44			**Firefox**	**Opera**		**Safari**	**IE**			**iOS**	**Andr.** 4.1–5.1		**IE**
		OS X	Win.	Lin.	40	12.17	31.0	6–8	7–9	10	11	6–8	Stock	Chrome	8.0
	CRL														
Int. 1	Revoked	EV	✓	EV	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	EV	✓	–	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
Int. 2+	Revoked	EV	EV	EV	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	✗	✗	–	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Leaf	Revoked	EV	EV	EV	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	✗	✗	–	✗	✗	✗	✗	✗	A	✓	✗	✗	✗	✗
	OCSP														
Int. 1	Revoked	EV	EV	EV	EV	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	✗	✗	–	✗	✗	L/W	✗	✓	✓	✓	✗	✗	✗	✗
Int. 2+	Revoked	EV	EV	EV	EV	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	✗	✗	–	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Leaf	Revoked	EV	EV	EV	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗
	Unavailable	✗	✗	–	✗	✗	✗	✗	✗	A	✓	✗	✗	✗	✗
Reject `unknown` status		✗	✗	–	✓	✓	✗	✗	✗	✗	✗	–	–	–	–
Try CRL on failure		EV	EV	–	✗	✗	L/W	✓	✓	✓	✓	–	–	–	–
	OCSP Stapling														
Request OCSP staple		✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	I	I	✗
Respect `revoked` staple		✗	✓	–	✓	✓	L/W	–	✓	✓	✓	–	–	–	–

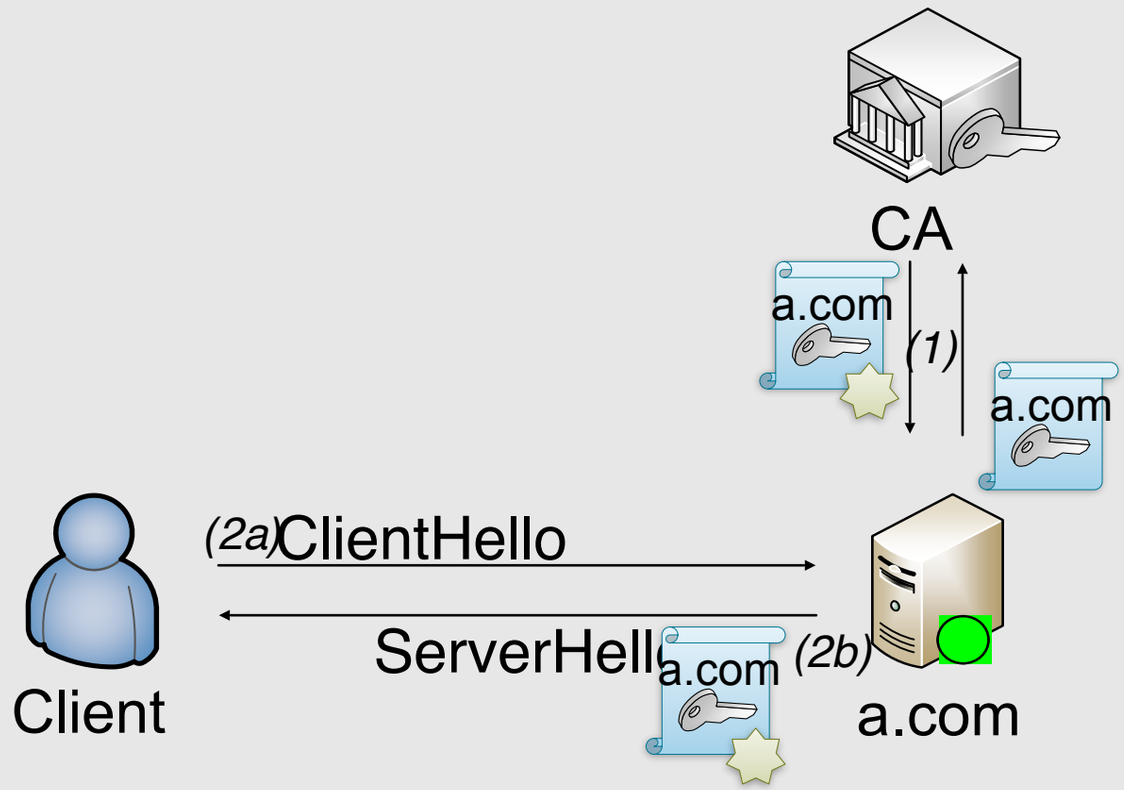
Table 2: Browser test results, when intermediate (Int.) and leaf certificates are either revoked or have revocation information unavailable. ✓ means browser passes test in all cases; ✗ means browser fails test in all cases. Other keys include EV (browser passes only for EV certificates), L/W (browser passes only on Linux and Windows), A (browser pops up an alert), and I (browser requests OCSP staple but ignores the response).

Certificate-Chain Validation

- The root CA certificate is trusted
- All certificates are valid
  - NotBefore < time() < NotAfter
  - not revoked (if revocation is supported)
- The leaf certificate is issued for the contacted party
- Certificates form a *chain of trust*
  - 1st certificate is self signed, and *i*th certificate's issuer is *(i-1)*th certificate's subject
  - 2nd certificate can be verified with the public key of the 1st one, 3rd certificate can be verified with the public key of the 2nd one,..., *i*th certificate can be verified with the public key of the *(i-1)*th

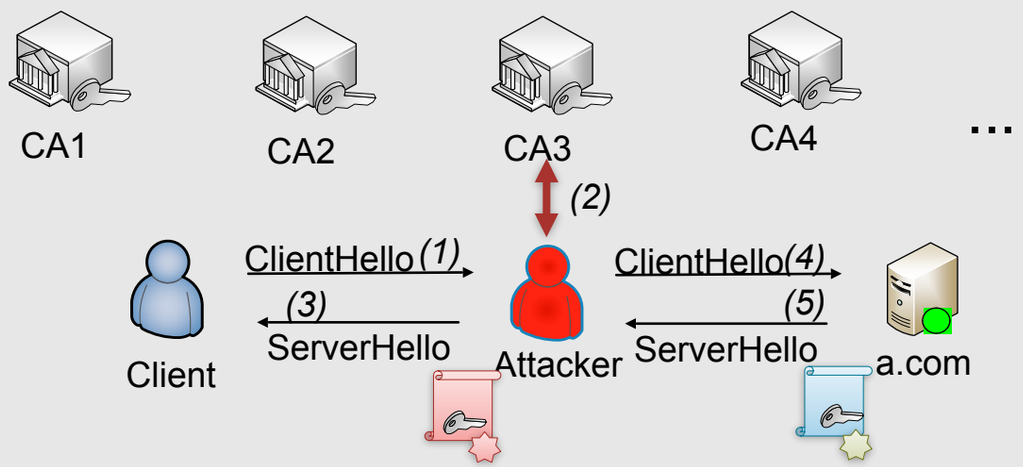
SSL/TLS PKI Model

- SSL/TLS Protocol
- CA is trusted by clients and domains
- Step *(1)* performed one-time per certificate



SSL/TLS PKI: Weak Authentication

- Certificates signed by single CA
  - Currently, cannot sign certificate by multiple CAs
- Weakest-link security with too many *trusted* entities
  - Current browsers trust ~1500 keys that can issue valid certificates



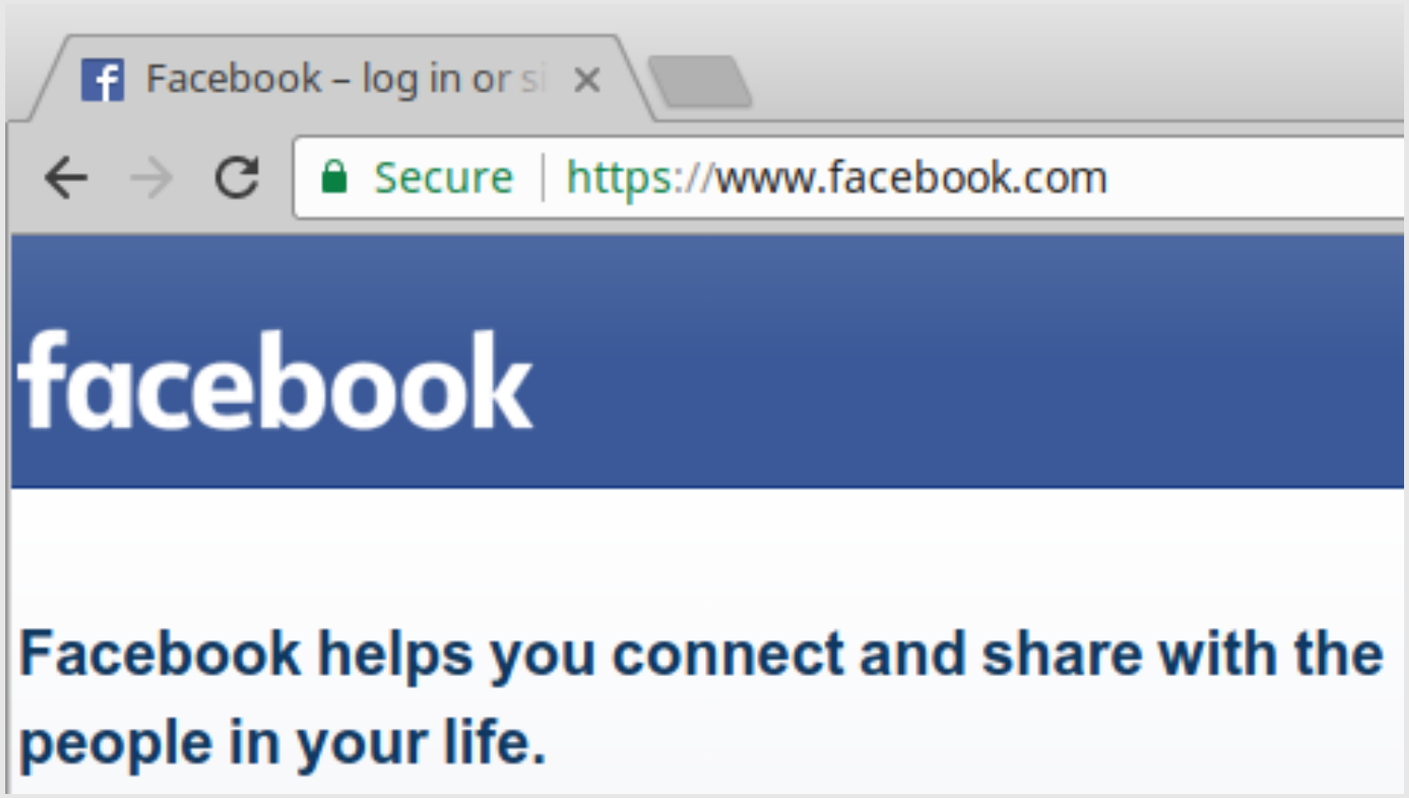
SSL/TLS PKI Problems

SSL/TLS PKI Problems

- Weakest-link security
- Revocation system is insecure and inefficient
  - Various schemes
  - Some CAs are *too-big-to-fail*
- Trust agility
  - Domains cannot state which CAs are trusted
- Transparency
  - CAs' actions are not transparent
- Imbalance
  - CAs have almost unlimited power
- Misconfigurations
  - SSLv2, weak crypto, NULL cipher suites

SSL/TLS as a Secure Channel

- Secure communication
  - Client-Server via HTTP**S**



Discussion&Classwork