

CSE 484 / CSE M 584: Computer Security and Privacy

Finish Cryptography; Start Web Security

Autumn 2020

Franziska (Franzi) Roesner

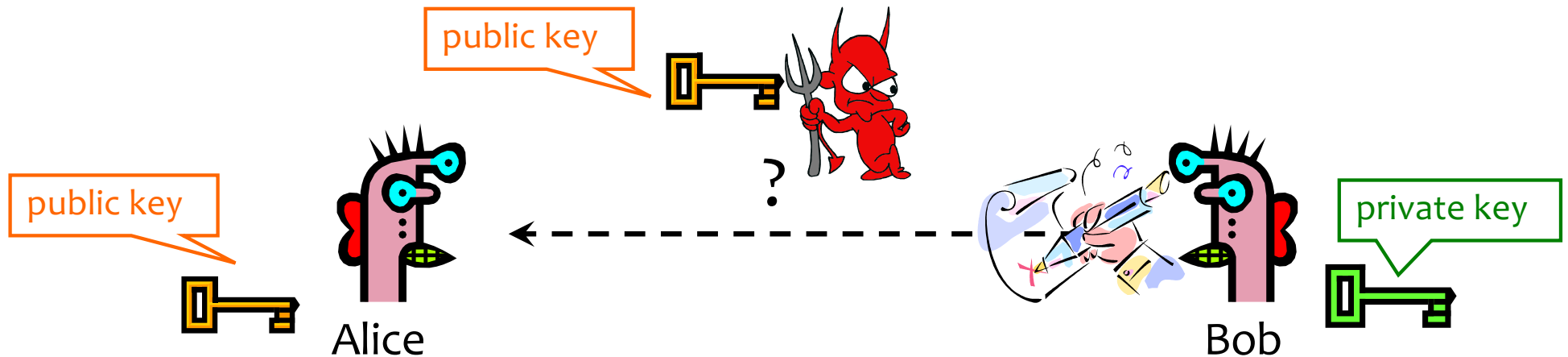
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, Ada Lerner, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Admin

- Lab 1 due today (10/30)
- Homework 2 due in a week (11/6)
- Final Project checkpoint #1 due in 2 weeks (11/13)

Digital Signatures: Basic Idea



Given: Everybody knows Bob's **public key**
Only Bob knows the corresponding **private key**

Goal: Bob sends a “digitally signed” message

1. To compute a signature, must know the private key
2. To verify a signature, only the public key is needed

RSA Signatures

- Public key is (n, e) , private key is (n, d)
- To **sign** message m : $s = m^d \bmod n$
 - Signing & decryption are same **underlying** operation in RSA
 - It's infeasible to compute s on m if you don't know d
- To **verify** signature s on message m :
verify that $s^e \bmod n = (m^d)^e \bmod n = m$
 - Just like encryption (for RSA primitive)
 - Anyone who knows n and e (public key) can verify signatures produced with d (private key)
- **In practice, also need padding & hashing**
 - Standard padding/hashing schemes exist for RSA signatures

DSS Signatures

- Digital Signature Standard (DSS)
 - U.S. government standard (1991, most recent rev. 2013)
- Public key: $(p, q, g, y=g^x \bmod p)$, private key: x
- Security of DSS requires hardness of discrete log
 - If could solve discrete logarithm problem, would extract x (private key) from $g^x \bmod p$ (public key)
- Again: We've discussed discrete logs modulo integers; significant advantages to using elliptic curve groups instead.

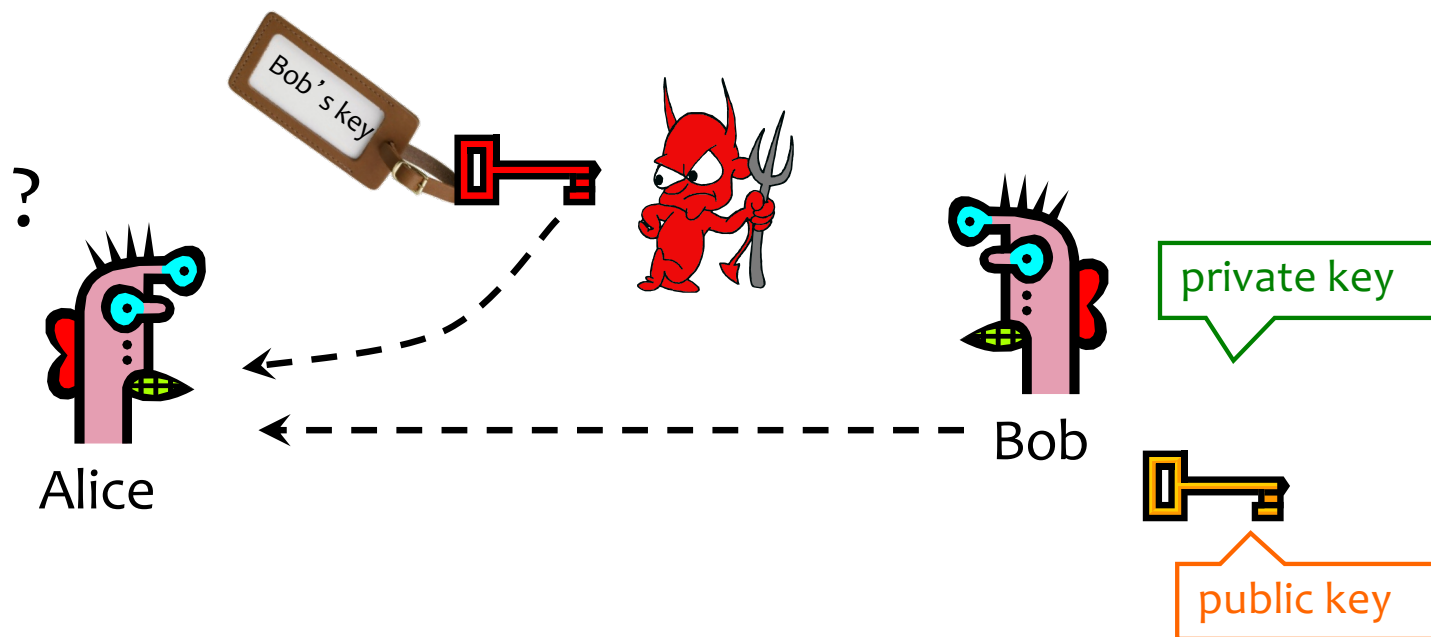
Cryptography Summary

- Goal: Privacy
 - Symmetric keys:
 - One-time pad, Stream ciphers
 - Block ciphers (e.g., DES, AES) → modes: EBC, CBC, CTR
 - Public key crypto (e.g., Diffie-Hellman, RSA)
- Goal: Integrity
 - MACs, often using hash functions (e.g, SHA-256)
- Goal: Privacy and Integrity
 - Encrypt-then-MAC
- Goal: Authenticity (and Integrity)
 - Digital signatures (e.g., RSA, DSS)

Want More Crypto?

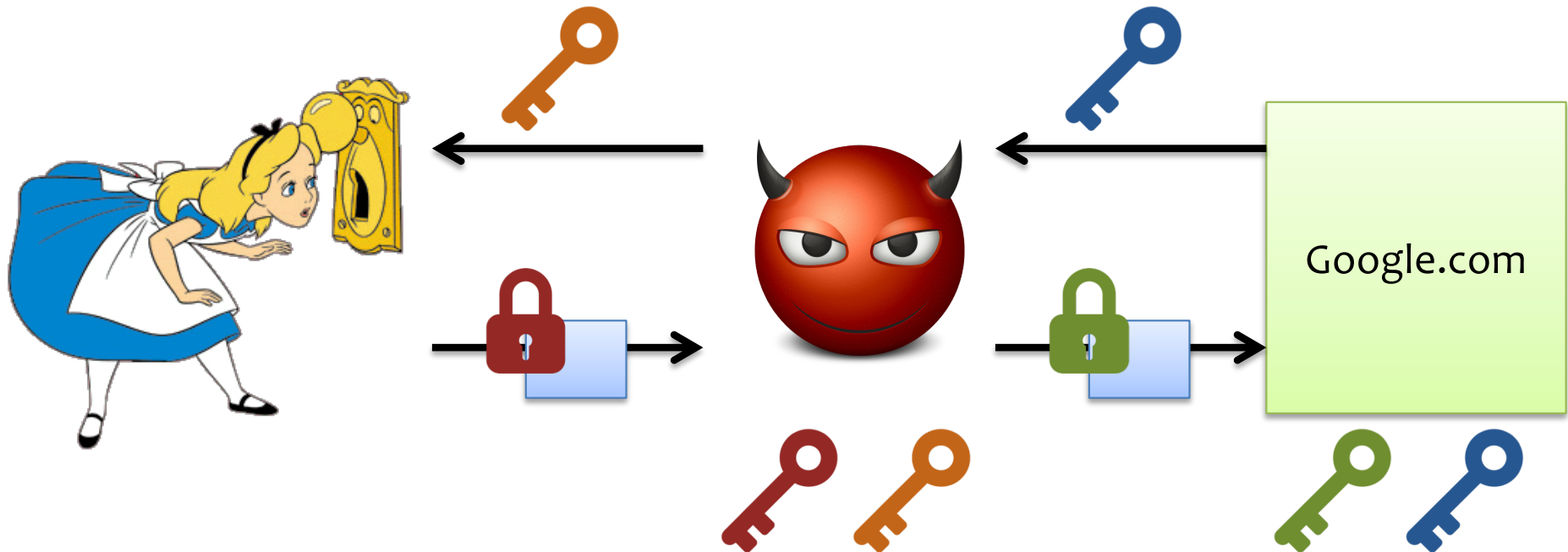
- Some suggestions:
 - CSE 490C (Rachel Lin):
<https://courses.cs.washington.edu/courses/cse490c/20au/>
 - Stanford Coursera (Dan Boneh):
<https://www.coursera.org/learn/crypto>

Authenticity of Public Keys



Problem: How does Alice know that the public key she received is really Bob's public key?

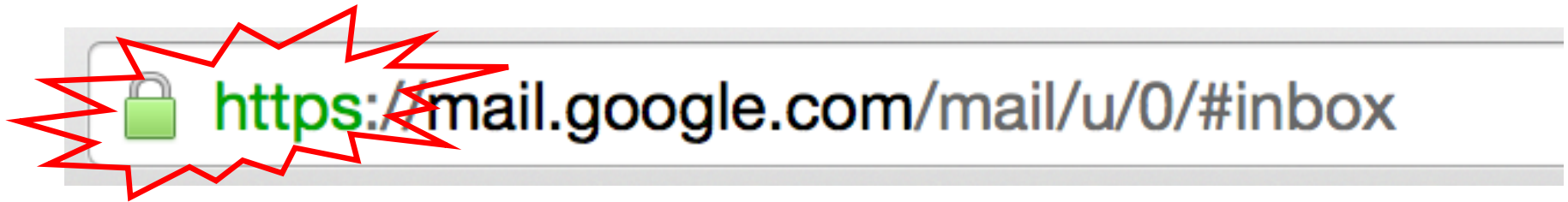
Threat: Person-in-the Middle



Distribution of Public Keys

- Public announcement or public directory
 - Risks: forgery and tampering
- Public-key certificate
 - Signed statement specifying the key and identity
 - $\text{sig}_{\text{CA}}(\text{"Bob"}, \text{PK}_B)$
- Common approach: certificate authority (CA)
 - Single agency responsible for certifying public keys
 - After generating a private/public key pair, user proves his identity and knowledge of the private key to obtain CA's certificate for the public key (offline)
 - Every computer is pre-configured with CA's public key

You encounter this every day...




SSL/TLS: Encryption & authentication for connections

SSL/TLS High Level

- SSL/TLS consists of **two** protocols
 - Familiar pattern for key exchange protocols
- Handshake protocol
 - Use **public-key cryptography** to establish a shared secret key between the client and the server
- Record protocol
 - Use the **secret symmetric key** established in the handshake protocol to protect communication between the client and the server

Example of a Certificate

GeoTrust Global CA
↳ Google Internet Authority G2
↳ *.google.com

 ***.google.com**
Issued by: Google Internet Authority G2
Expires: Monday, July 6, 2015 at 5:00:00 PM Pacific Daylight Time
✓ This certificate is valid

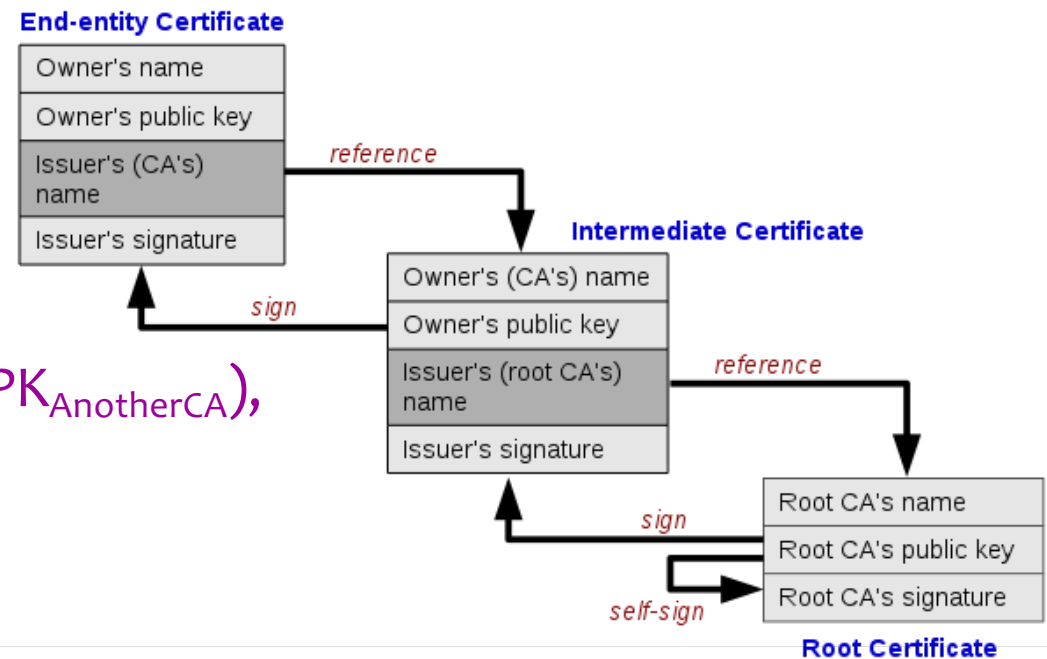
▼ **Details**

Subject Name	
Country	US
State/Province	California
Locality	Mountain View
Organization	Google Inc
Common Name	*.google.com
Issuer Name	
Country	US
Organization	Google Inc
Common Name	Google Internet Authority G2
Serial Number	6082711391012222858
Version	3

Signature Algorithm	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
Parameters	none
Not Valid Before	Wednesday, April 8, 2015 at 6:40:10 AM Pacific Daylight Time
Not Valid After	Monday, July 6, 2015 at 5:00:00 PM Pacific Daylight Time
Public Key Info	
Algorithm	Elliptic Curve Public Key (1.2.840.10045.2.1)
Parameters	Elliptic Curve secp256r1 (1.2.840.10045.3.1.7)
Public Key	65 bytes : 04 CB DD C1 CE AC D6 20 ...
Key Size	256 bits
Key Usage	Encrypt, Verify, Derive
Signature	256 bytes : 34 8B 7D 64 5A 64 08 5B ...

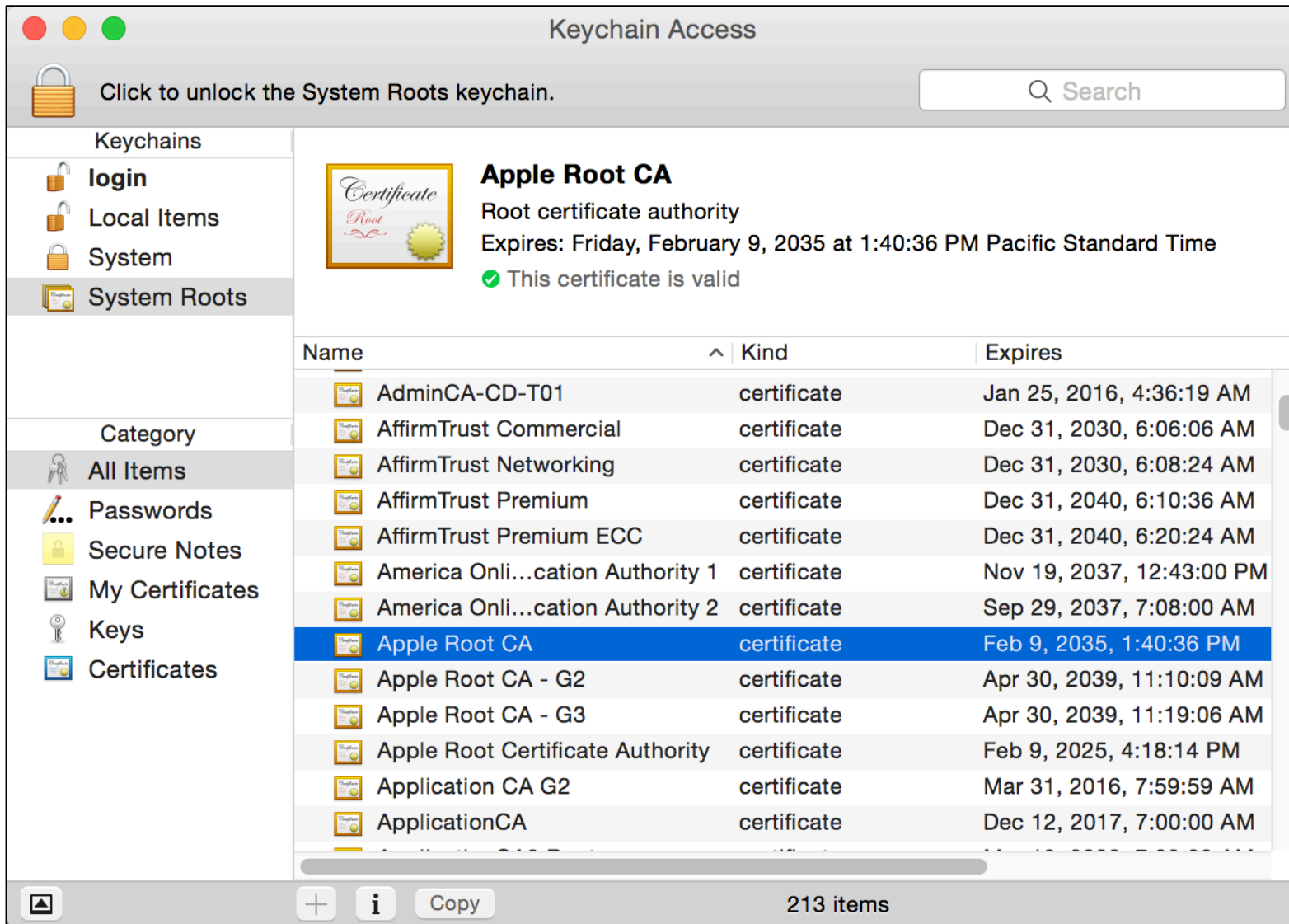
Hierarchical Approach

- Single CA certifying every public key is impractical
- Instead, use a trusted **root authority** (e.g., Verisign)
 - Everybody must know the root's public key
 - Instead of single cert, use a **certificate chain**
 - $\text{sig}_{\text{Verisign}}(\text{"AnotherCA"}, \text{PK}_{\text{AnotherCA}})$,
 $\text{sig}_{\text{AnotherCA}}(\text{"Alice"}, \text{PK}_A)$



– What happens if root authority is ever compromised?

Trusted(?) Certificate Authorities



Turtles All The Way Down...



The saying holds that the world is supported by a chain of increasingly large turtles. Beneath each turtle is yet another: it is "turtles all the way down".

[Image from Wikipedia]

Many Challenges...

- Hash collisions
- Weak security at CAs
 - Allows attackers to issue rogue certificates
- Users don't notice when attacks happen
 - We'll talk more about this later in the course
- How do you revoke certificates?

DigiNotar is a Dutch Certificate Authority. They sell SSL certificates.



Attacking CAs

Security of DigiNotar servers:

- All core certificate servers controlled by a single admin password (Prod@dm1n)
- Software on public-facing servers out of date, unpatched
- No anti-virus (could have detected attack)

Somehow, somebody managed to get a rogue SSL certificate from them on **July 10th, 2011**. This certificate was issued for domain name **.google.com**.

What can you do with such a certificate? Well, you can impersonate Google — assuming you can first reroute Internet traffic for google.com to you. This is something that can be done by a government or by a rogue ISP. Such a reroute would only affect users within that country or under that ISP.

Consequences

- Attacker needs to first divert users to an attacker-controlled site instead of Google, Yahoo, Skype, but then...
 - For example, use DNS to poison the mapping of mail.yahoo.com to an IP address
- ... “authenticate” as the real site
- ... decrypt all data sent by users
 - Email, phone conversations, Web browsing

Attempt to Fix CA Problems:

Certificate Transparency

- **Problem:** browsers will think nothing is wrong with a rogue certificate until revoked
- **Goal:** make it impossible for a CA to issue a bad certificate for a domain *without the owner of that domain knowing*
 - (Then what?)
- **Approach:** auditable certificate logs

www.certificate-transparency.org

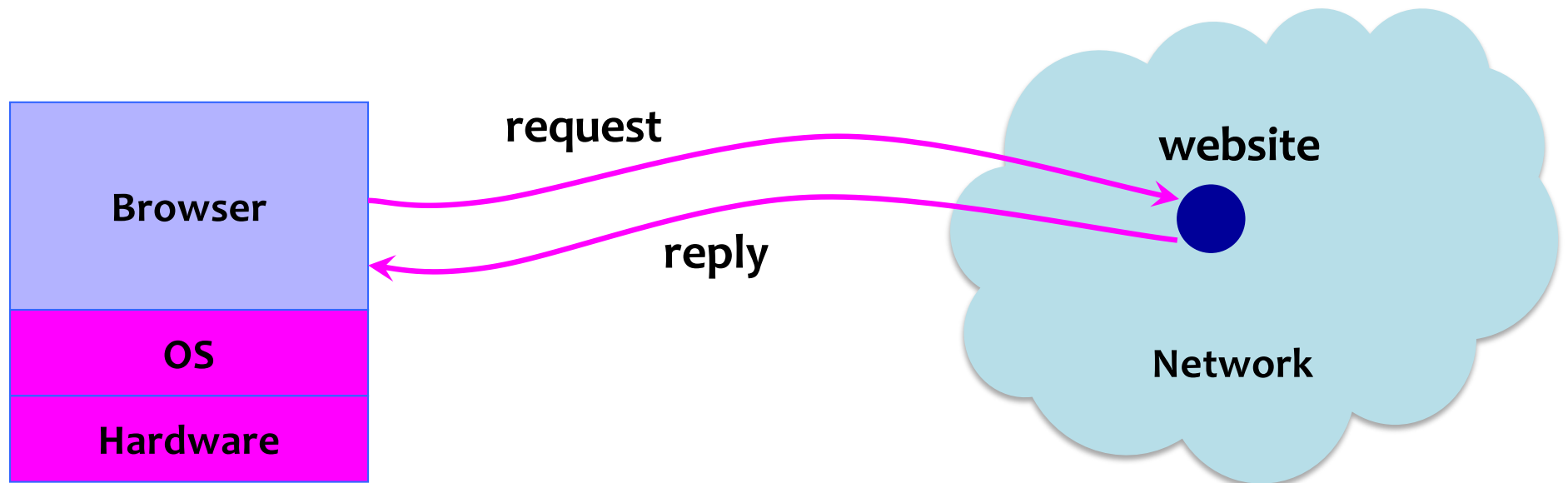
Attempt to Fix CA Problems:

Certificate Pinning

- **Trust on first access:** tells browser how to act on subsequent connections
- HPKP – HTTP Public Key Pinning
 - Use these keys!
 - HTTP response header field `Public-Key-Pins`
- HSTS – HTTP Strict Transport Security
 - Only access server via HTTPS
 - HTTP response header field `Strict-Transport-Security`

Next Major Topic!
Web+Browser Security

Big Picture: Browser and Network



Where Does the Attacker Live?

