CSE 484 / CSE M 584: Computer Security and Privacy

Web Security
[Certificates and Overview]

Spring 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 ...

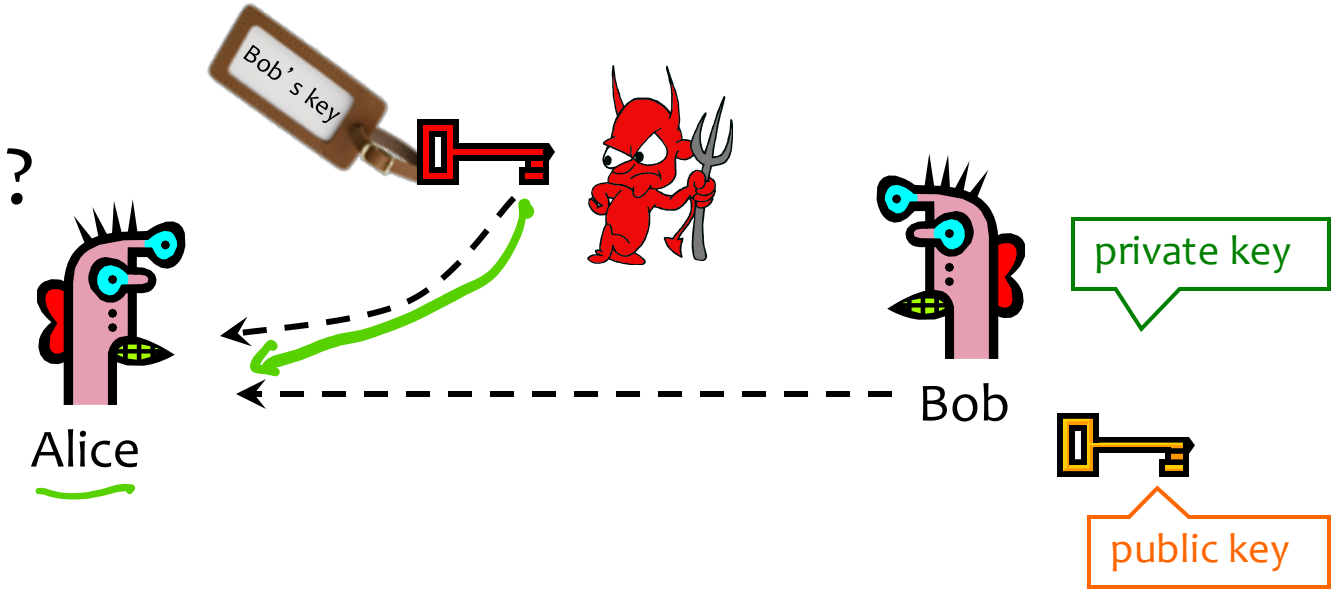
Admin

- Today:
  - Transition to web security
- Lab 1 due on Friday
  - See FAQs on discussion board

Cryptography Summary

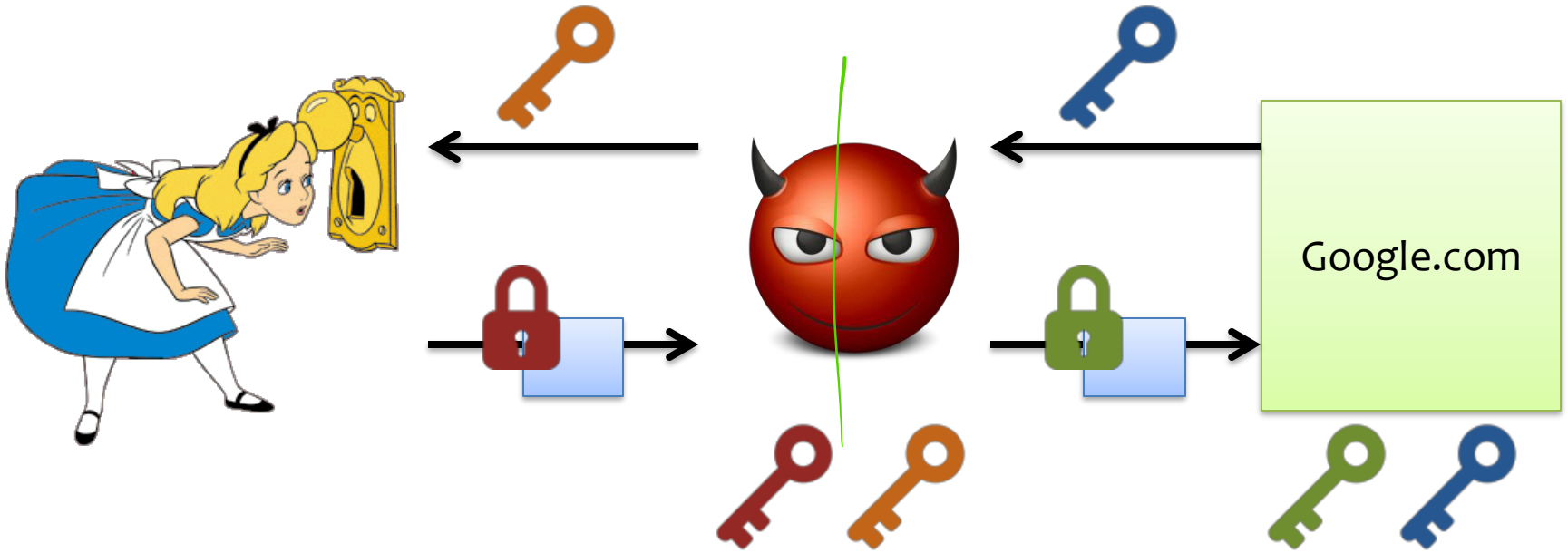
- Goal: Privacy
  - Symmetric keys:
    - One-time pad, Stream ciphers
    - Block ciphers (e.g., DES, AES) → modes: EBC, CBC, CTR
  - Public key crypto (e.g., Diffie-Hellman, RSA)
- Goal: Integrity
  - MACs, often using hash functions (e.g, SHA-256)
- Goal: Privacy and Integrity
  - Encrypt-then-MAC → not encrypt AND MAC
- Goal: Authenticity (and Integrity)
  - Digital signatures (e.g., RSA, DSS)

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

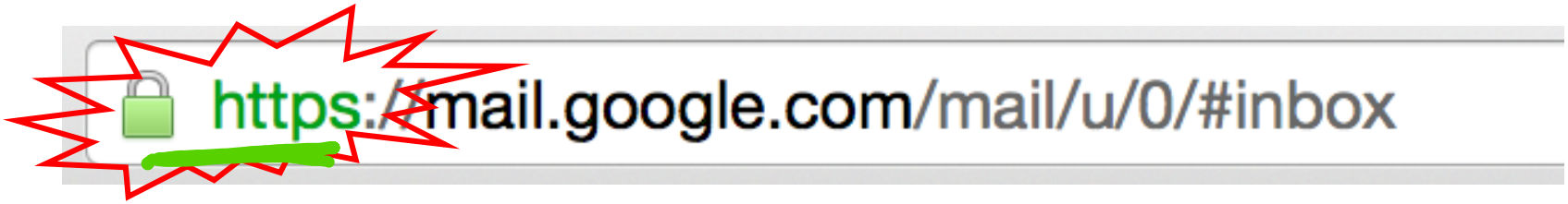


Distribution of Public Keys

- Public announcement or public directory
  - Risks: forgery and tampering
- Public-key certificate
  - Signed statement specifying the key and identity
    - $sig_{CA}(\text{“Bob”}, PK_B)$
- 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

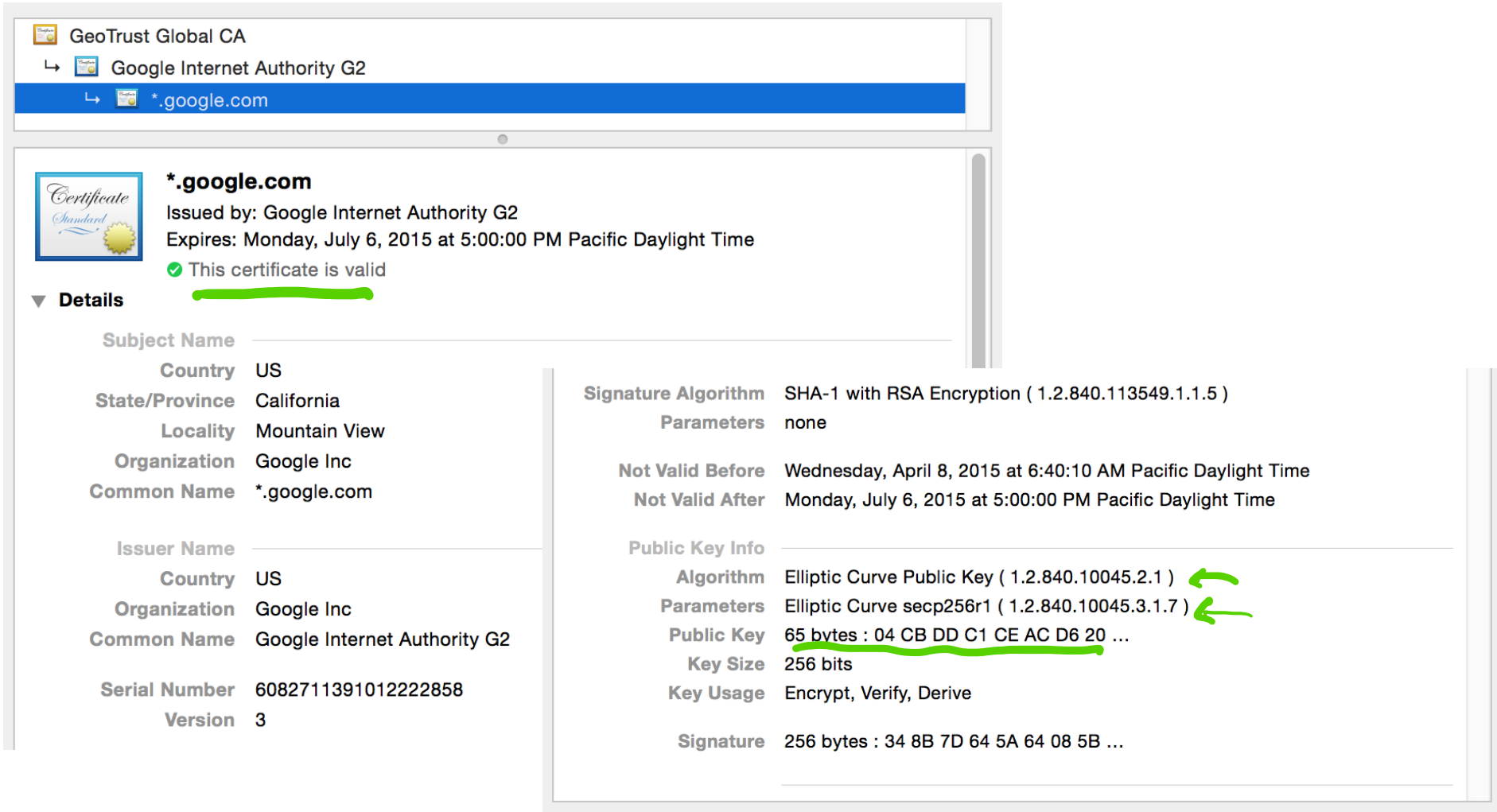
verisign

You encounter this every day…



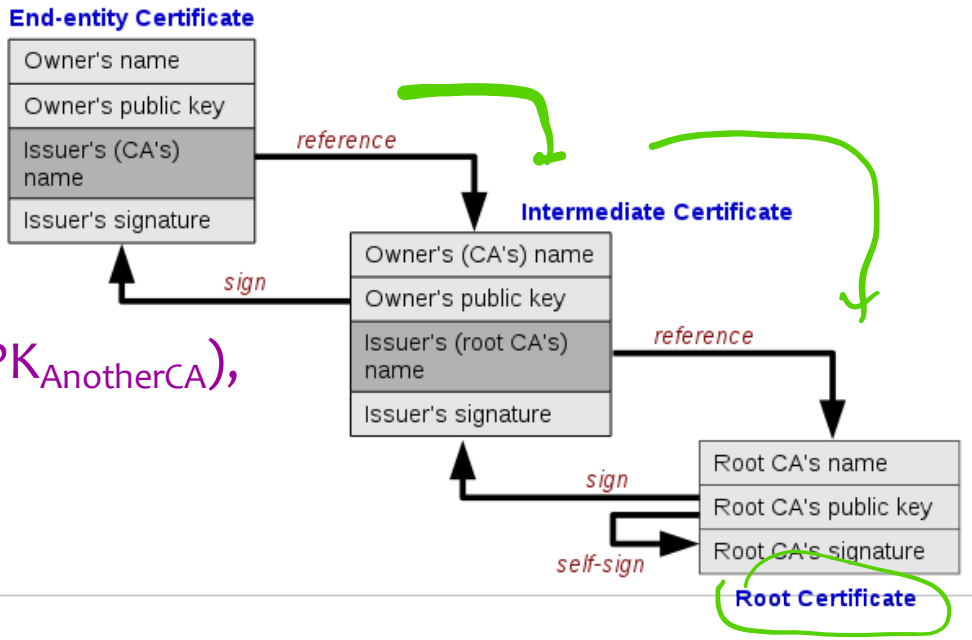
SSL/TLS: Encryption & authentication for connections

Example of a Certificate



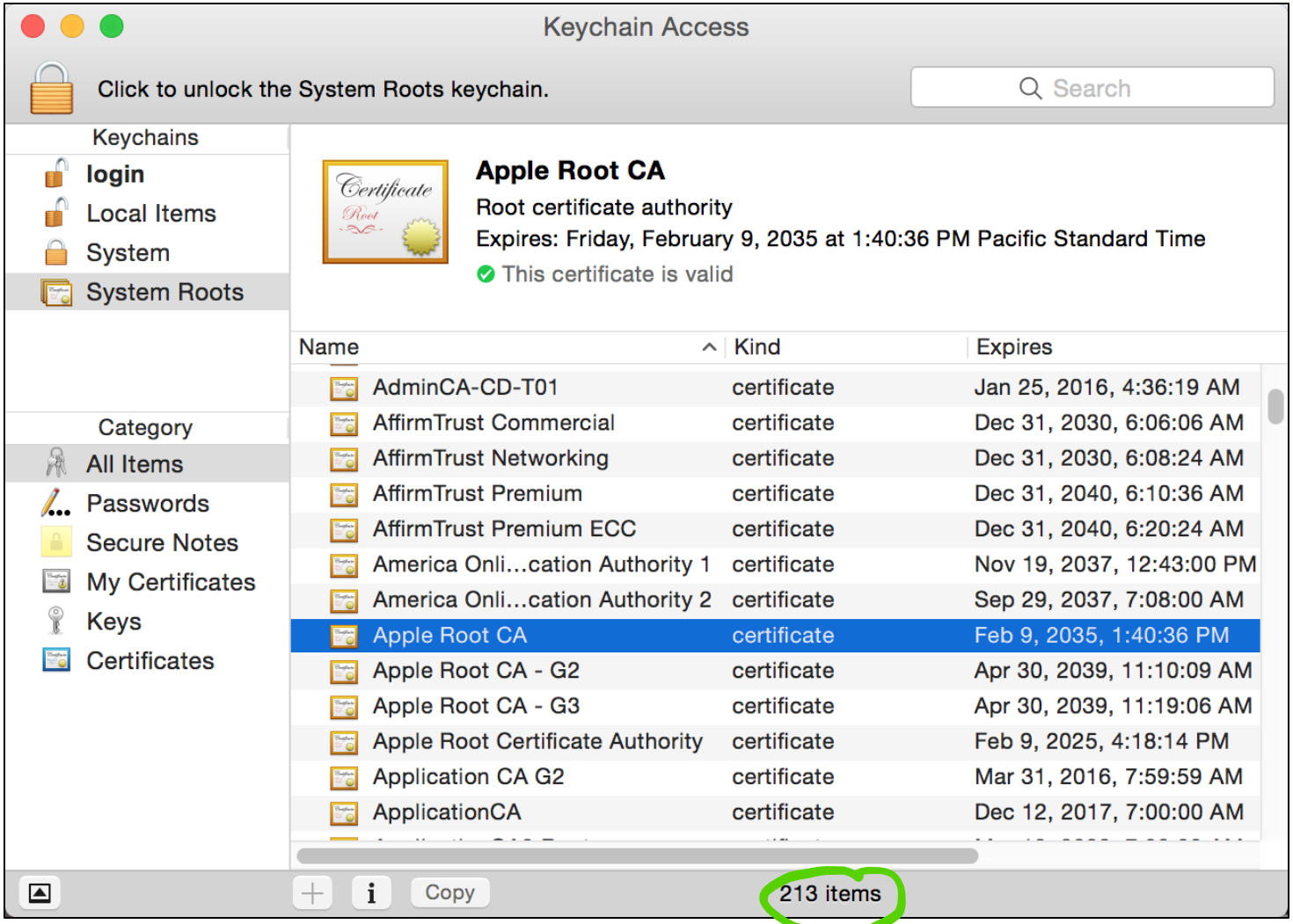
Hierarchical Approach

- 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
    - $sig_{Verisign}(\text{“AnotherCA”}, PK_{AnotherCA})$, $sig_{AnotherCA}(\text{“Alice”}, PK_A)$



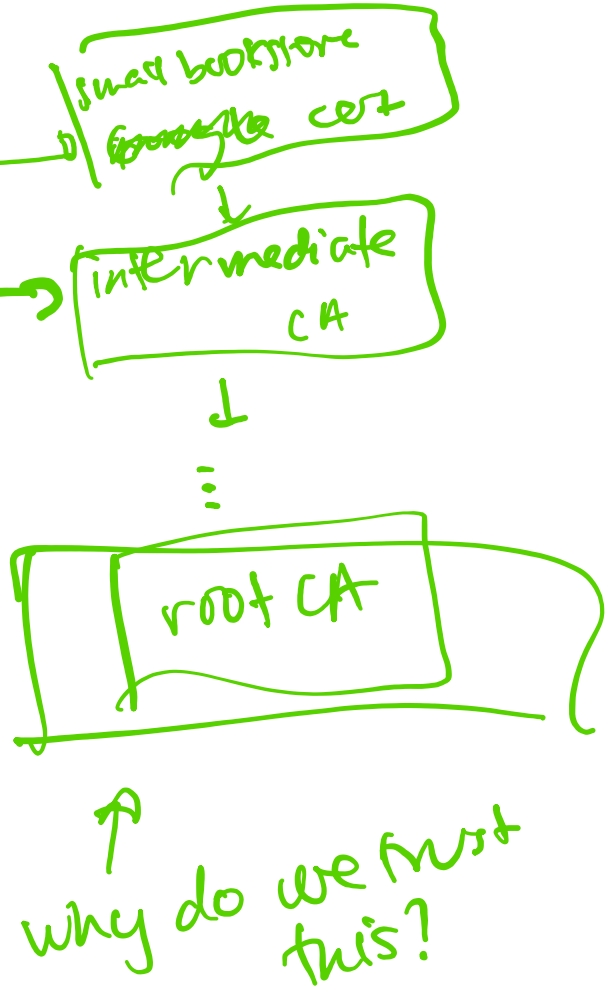
  - What happens if root authority is ever compromised?

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 ← md5 ← explicitly forge a cert
- 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?

Attacking CAs

DigiNotar Hacked by Black.Spook and Iranian Hackers Posted by Mikko @ 09:05 GMT | Comments

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

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.

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)

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

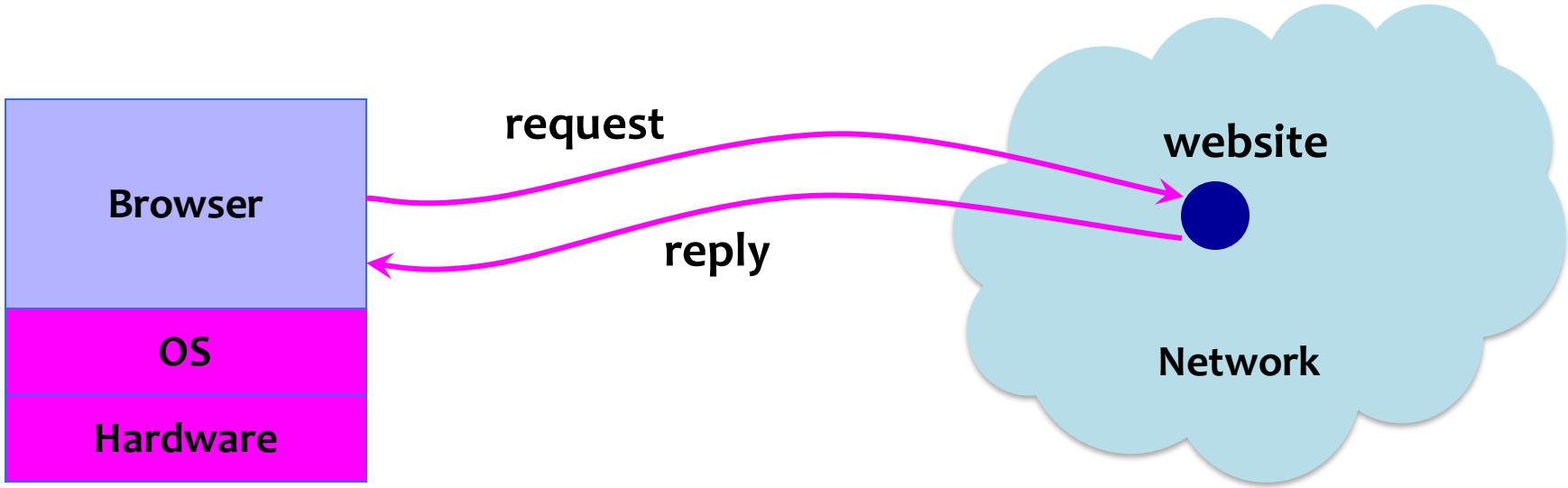
Assumes attack are rare

- 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`"

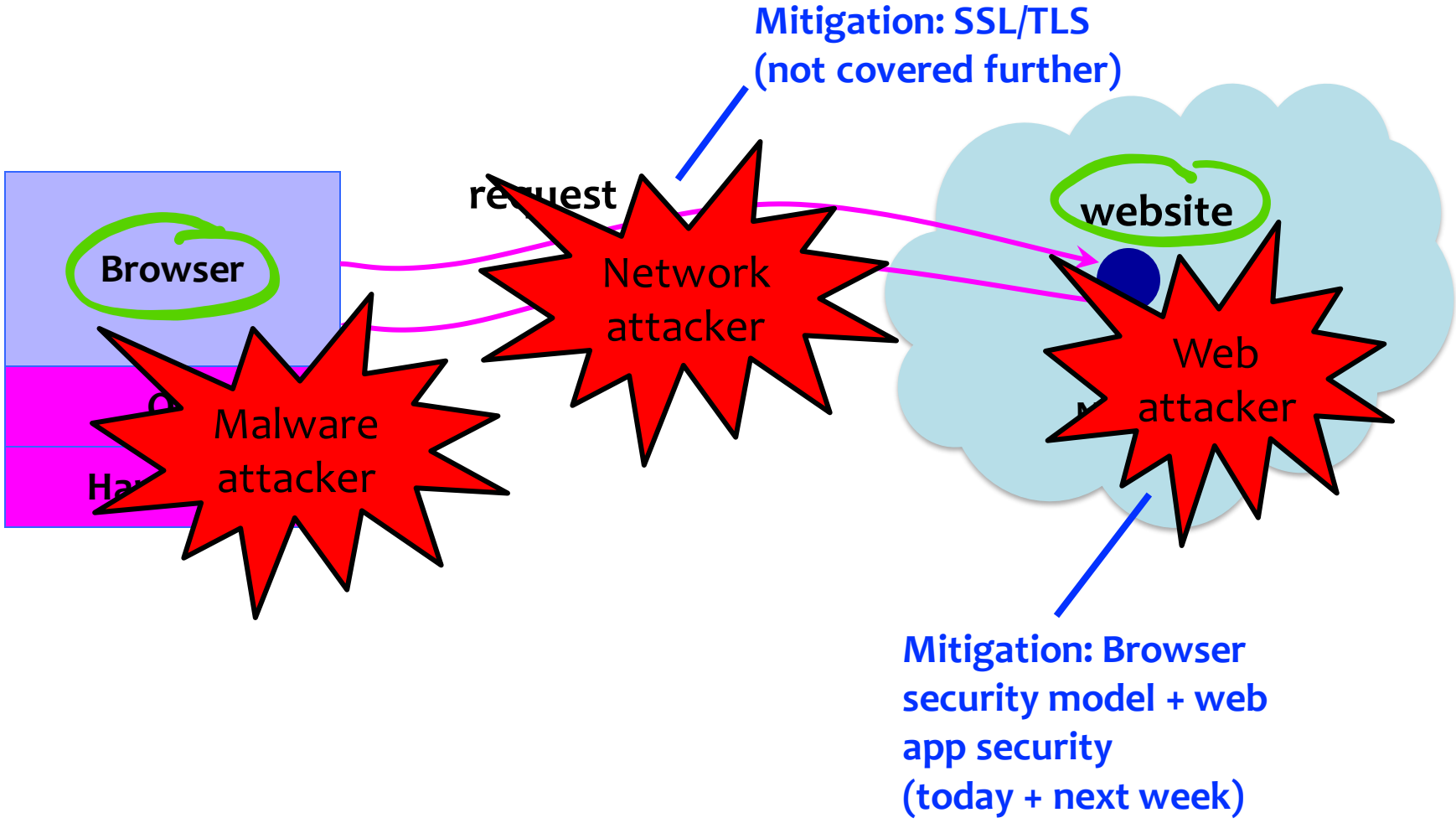
Let's Encrypt (EFF)

Web+Browser Security

Big Picture: Browser and Network



Where Does the Attacker Live?



Web Attacker

- Controls a malicious website (**attacker.com**)
  - Can even obtain SSL/TLS certificate for site
- User visits attacker.com – why?
  - Phishing email, enticing content, search results, placed by an ad network, blind luck …
- Attacker has no other access to user machine!
- Variation: good site **honest.com**, but:
  - An iframe with malicious content included
  - Website has been compromised