CSE 484 / CSE M 584: Web Security: Certificates and Browser Security Model

Fall 2023

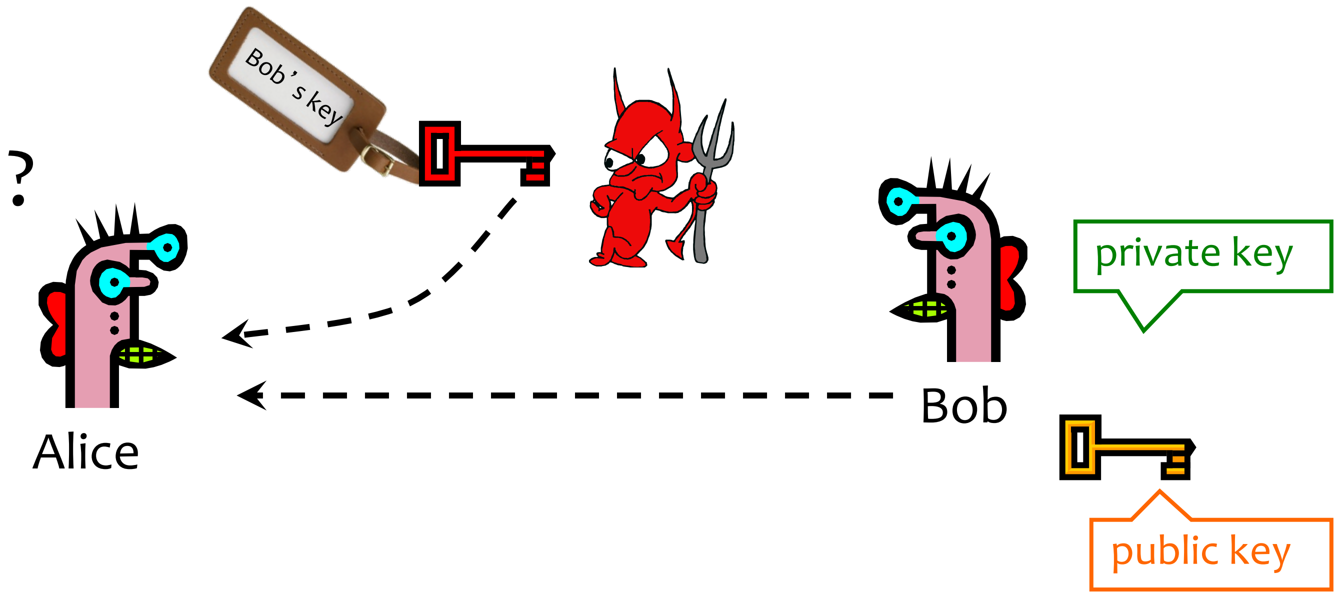
Franziska (Franzi) Roesner
franzi@cs

UW Instruction Team: David Kohlbrenner, Yoshi Kohno, Franziska Roesner. Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Announcements

- Homework 2 due in 1 week
- Lab 2 (web security) out mid next week
- New and improved office hour schedule!
  - Mondays 11:30-12:30pm: Franzi @ Gates 314
  - Mondays 12:30-1:30pm: Akash @ Allen 3rd floor breakout
  - Tuesdays 1-3pm: Basia, Evan, Shaoqi @ Gates 345
  - Wednesdays 5-6pm: Lin @ Allen 5th floor breakout
  - Thursday 10:30am-11:30am: Kirsten @ Allen 3rd floor breakout
  - Fridays 3-4pm: Sara @ Allen Center 624
  - Fridays 5-6pm: Sonia @ Zoom

Review: Authenticity of Public Keys

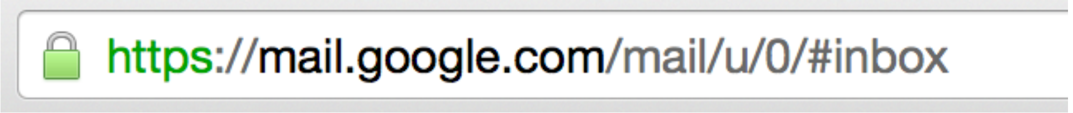


Problem: How does Alice know that the public key she received is really Bob's public key?

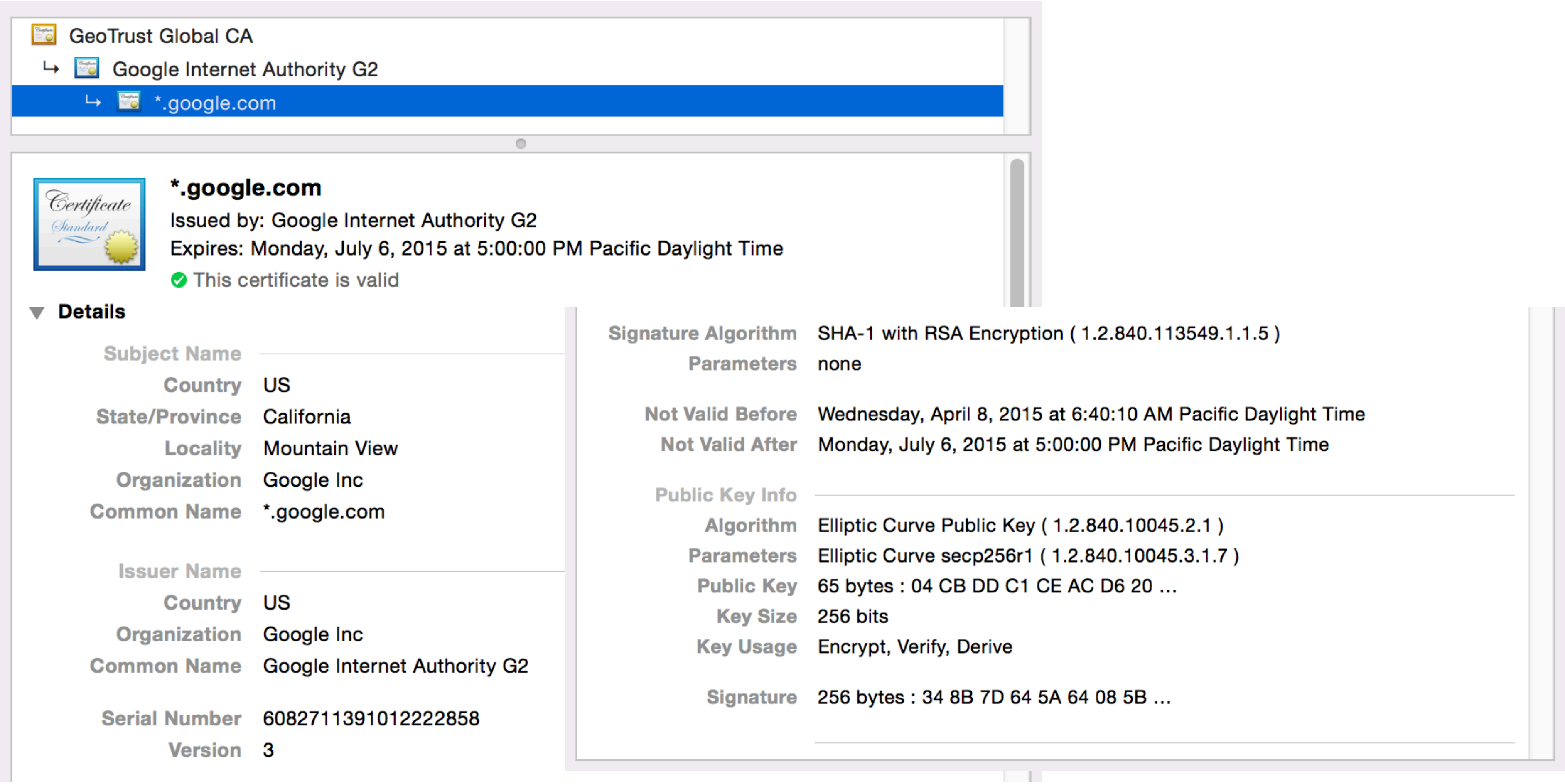
Review: Distribution of Public Keys

- Public announcement or public directory
  - Risks: forgery and tampering
- Public-key certificate
  - Signed statement specifying the key and identity
    - sig_{CA}(“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

Review: Example of a Certificate



https://mail.google.com/mail/u/0/#inbox



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

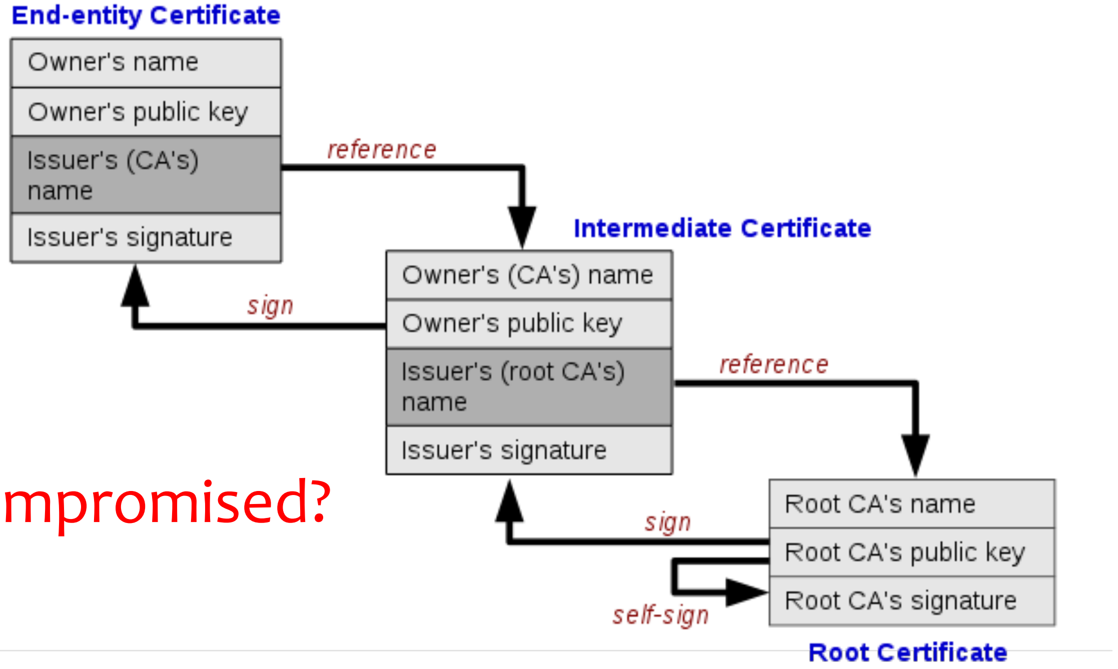
▼ **Details**

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

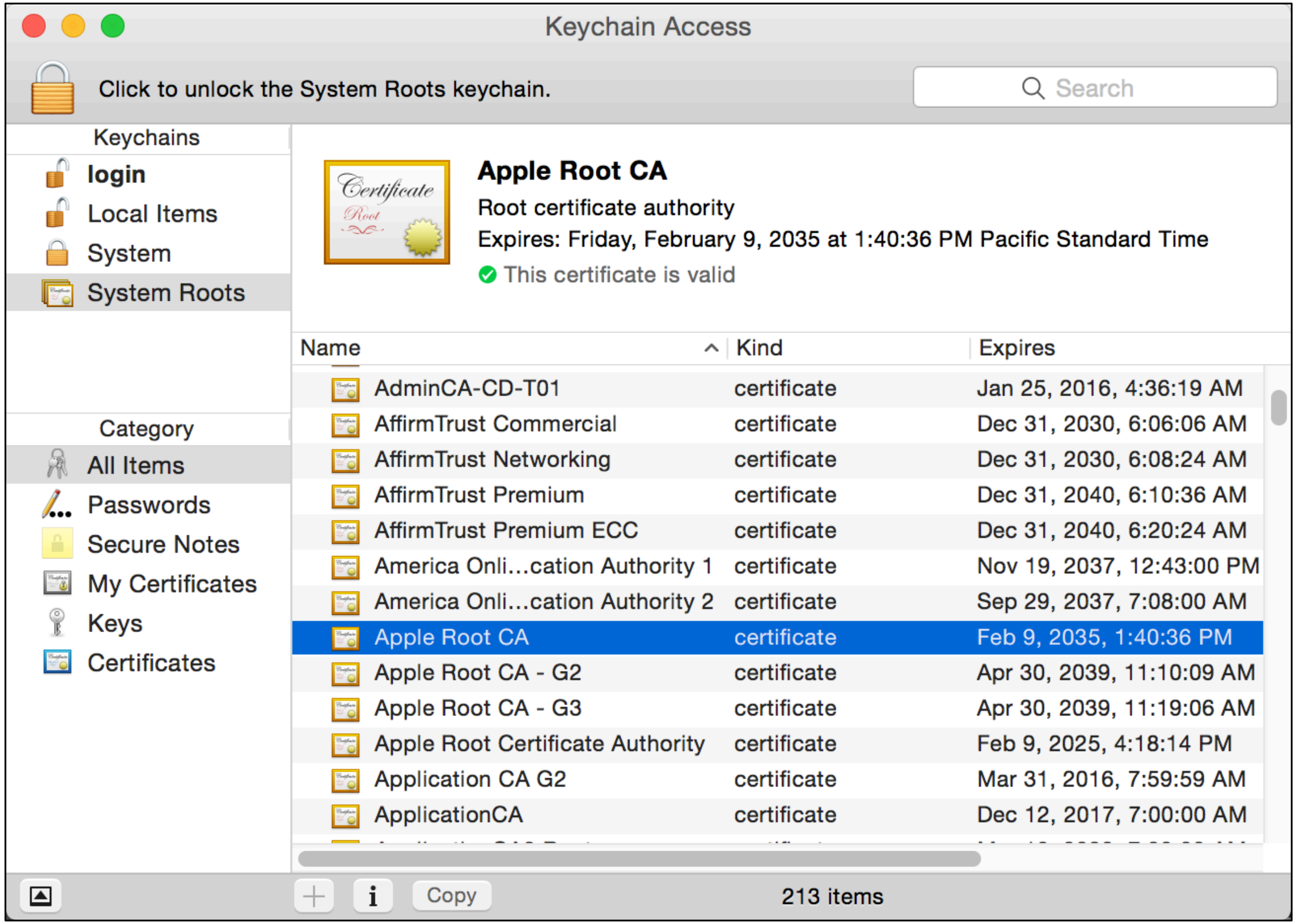
Signature Algorithm	SHA-1 with RSA Encryption (1.2.840.113549.1.1.5)
Parameters	none
Not Valid Before	Wednesday, April 8, 2015 at 6:40:10 AM Pacific Daylight Time
Not Valid After	Monday, July 6, 2015 at 5:00:00 PM Pacific Daylight Time
Public Key Info	
Algorithm	Elliptic Curve Public Key (1.2.840.10045.2.1)
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
    - $\text{sig}_{\text{Verisign}}$("AnotherCA", $\text{PK}_{\text{AnotherCA}}$), $\text{sig}_{\text{AnotherCA}}$("Alice", PK_{A})
  - Not shown in figure but important:
    - Signed as part of each cert is whether party is a CA or not
  - What happens if root authority is ever compromised?



Trusted(?) Certificate Authorities



Apple Root CA

Root certificate authority

Expires: Friday, February 9, 2035 at 1:40:36 PM Pacific Standard Time

✓ This certificate is valid

Name	Kind	Expires
AdminCA-CD-T01	certificate	Jan 25, 2016, 4:36:19 AM
AffirmTrust Commercial	certificate	Dec 31, 2030, 6:06:06 AM
AffirmTrust Networking	certificate	Dec 31, 2030, 6:08:24 AM
AffirmTrust Premium	certificate	Dec 31, 2040, 6:10:36 AM
AffirmTrust Premium ECC	certificate	Dec 31, 2040, 6:20:24 AM
America Onli…cation Authority 1	certificate	Nov 19, 2037, 12:43:00 PM
America Onli…cation Authority 2	certificate	Sep 29, 2037, 7:08:00 AM
Apple Root CA	certificate	Feb 9, 2035, 1:40:36 PM
Apple Root CA - G2	certificate	Apr 30, 2039, 11:10:09 AM
Apple Root CA - G3	certificate	Apr 30, 2039, 11:19:06 AM
Apple Root Certificate Authority	certificate	Feb 9, 2025, 4:18:14 PM
Application CA G2	certificate	Mar 31, 2016, 7:59:59 AM
ApplicationCA	certificate	Dec 12, 2017, 7:00:00 AM

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".

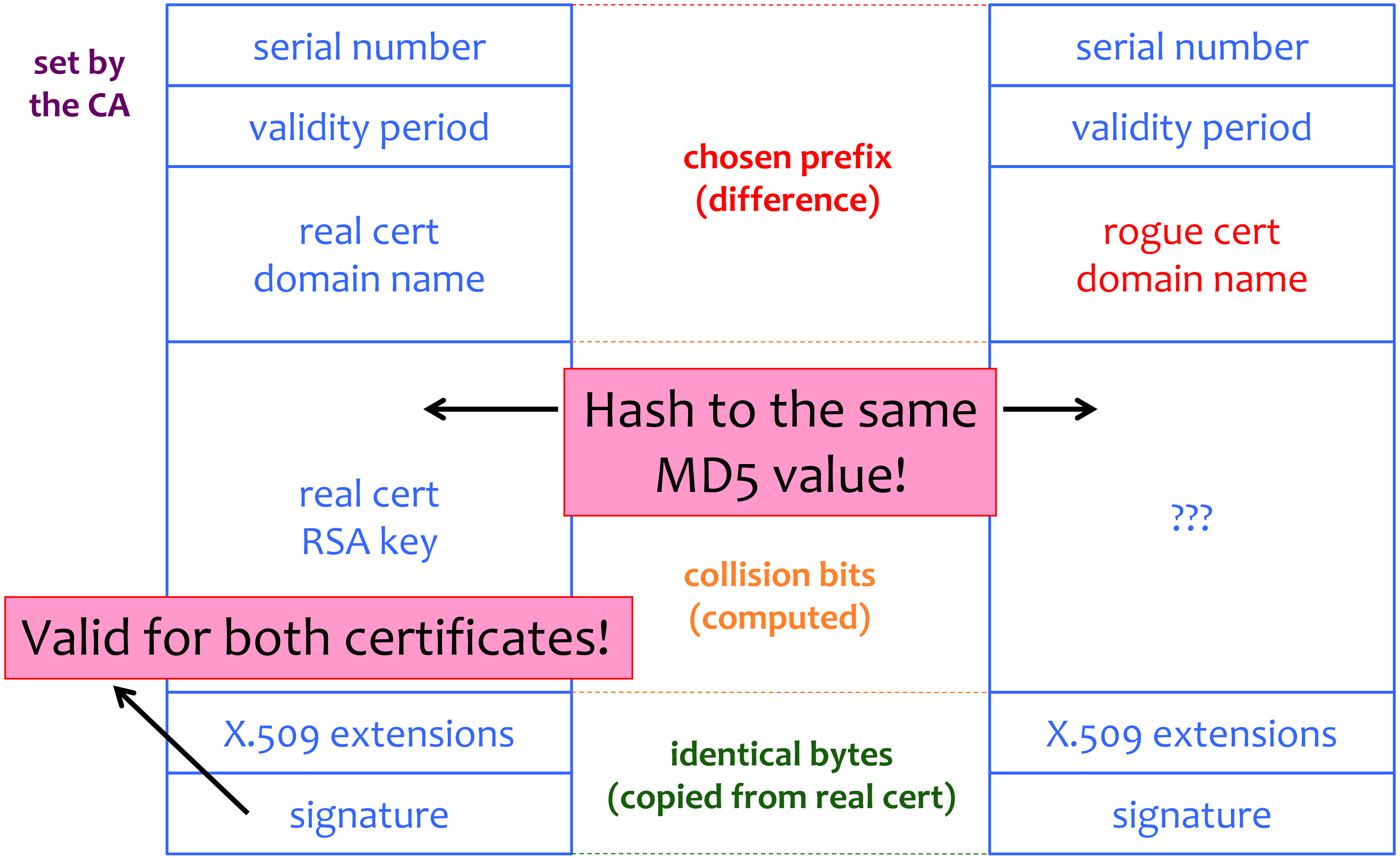
[Image from Wikipedia]

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?

[Sotirov et al. “Rogue Certificates”]

Colliding Certificates



set by the CA

serial number

validity period

real cert domain name

real cert RSA key

X.509 extensions

signature

chosen prefix (difference)

Hash to the same MD5 value!

collision bits (computed)

identical bytes (copied from real cert)

serial number

validity period

rogue cert domain name

???

X.509 extensions

signature

Valid for both certificates!

Attacking CAs

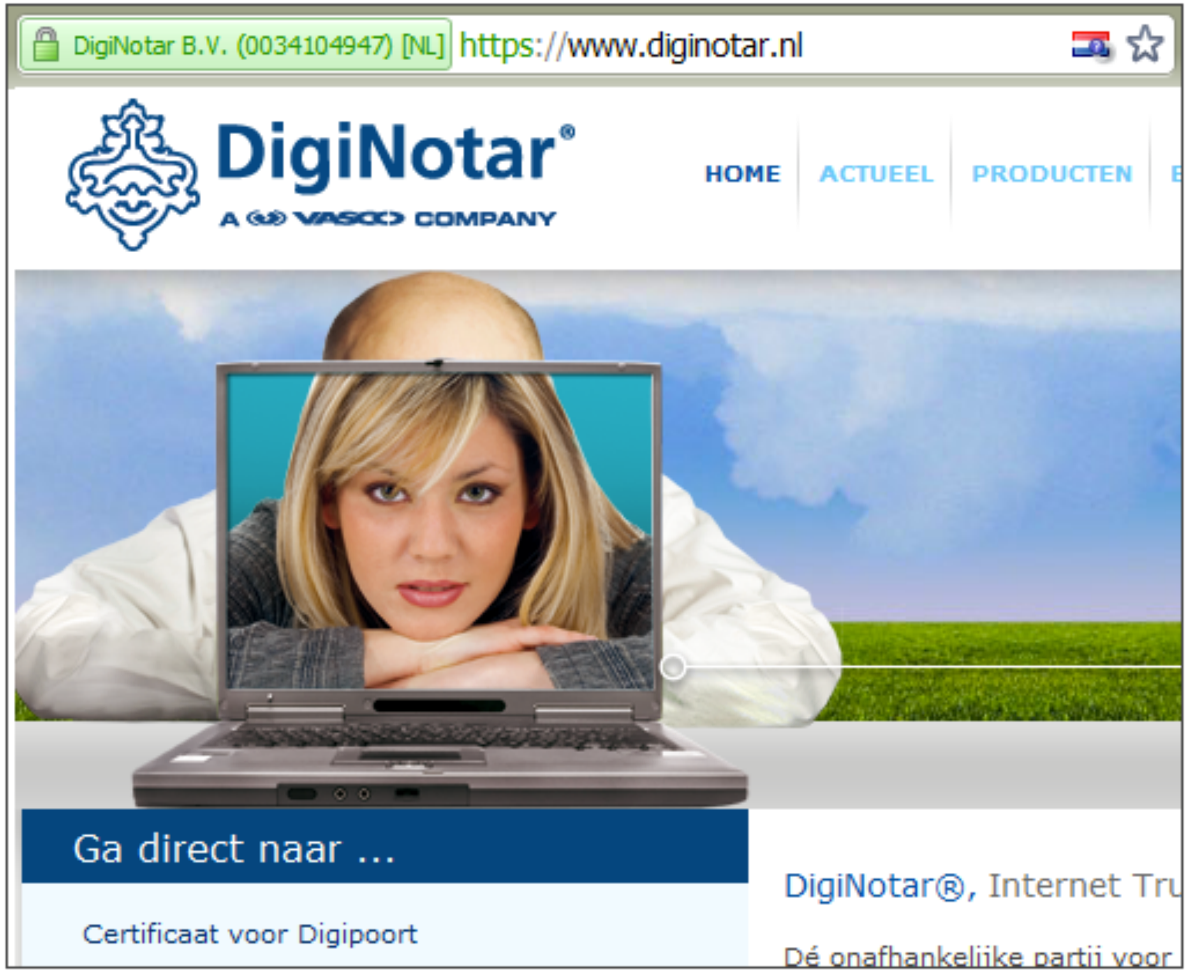
Security of DigiNotar servers:

- 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)

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.



DigiNotar B.V. (0034104947) [NL] https://www.diginotar.nl

Ga direct naar ...

Certificaat voor Digipoort

DigiNotar®, Internet Tru

Dé onafhankelijke partij voor

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

More Rogue Certs

- In Jan 2013, a rogue *.google.com certificate was issued by an intermediate CA that gained its authority from the Turkish root CA TurkTrust
  - TurkTrust accidentally issued intermediate CA certs to customers who requested regular certificates
  - Ankara transit authority used its certificate to issue a fake *.google.com certificate in order to filter SSL traffic from its network
- This rogue *.google.com certificate was trusted by every browser in the world
- There are plenty more stories like this…

Certificate Revocation

- Revocation is <u>very</u> important
- Many valid reasons to revoke a certificate
  - Private key corresponding to the certified public key has been compromised
  - User stopped paying their certification fee to this CA and CA no longer wishes to certify them
  - CA's private key has been compromised!
- Expiration is a form of revocation, too
  - Many deployed systems don't bother with revocation
  - Re-issuance of certificates is a big revenue source for certificate authorities

Certificate Revocation Mechanisms

- Certificate revocation list (CRL)
  - CA periodically issues a signed list of revoked certificates
    - Credit card companies used to issue thick books of canceled credit card numbers
  - Can issue a "delta CRL" containing only updates
- Online revocation service
  - When a certificate is presented, recipient goes to a special online service to verify whether it is still valid
    - Like a merchant dialing up the credit card processor

Attempt to Fix CA Problems:

Certificate Pinning

- Trust on first access: tells browser how to act on subsequent connections
- HPKP – HTTP Public Key Pinning
  [obsolete, but pinning idea persists e.g. in mobile apps]
  - 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`"

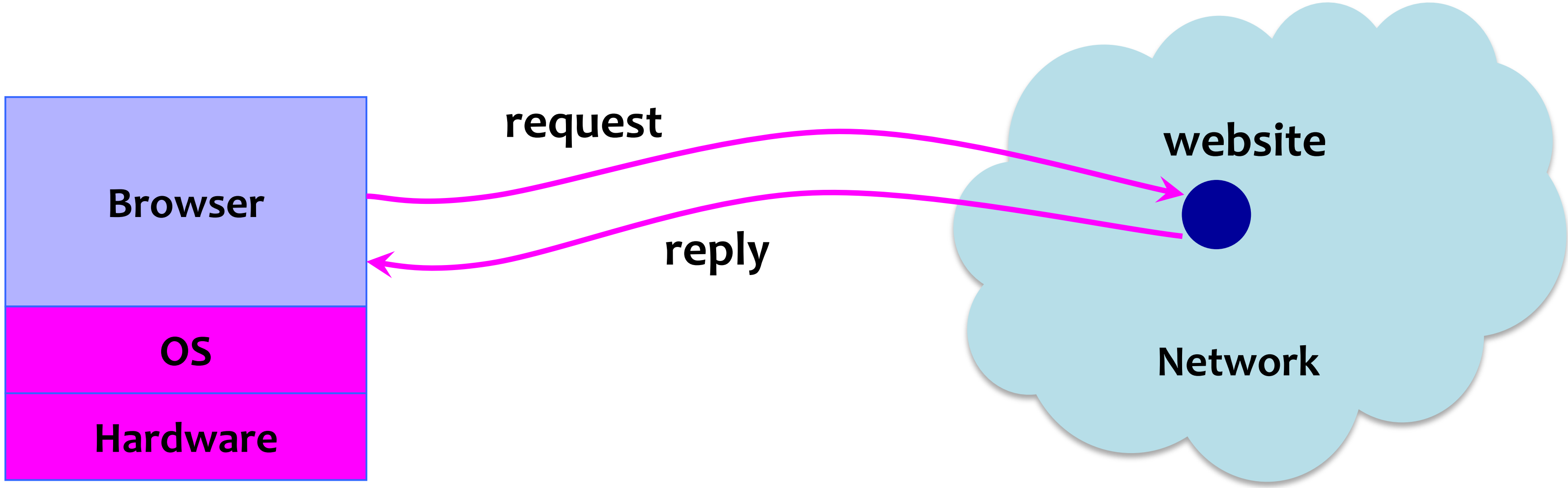
Attempt to Fix CA Problems:

Certificate Transparency

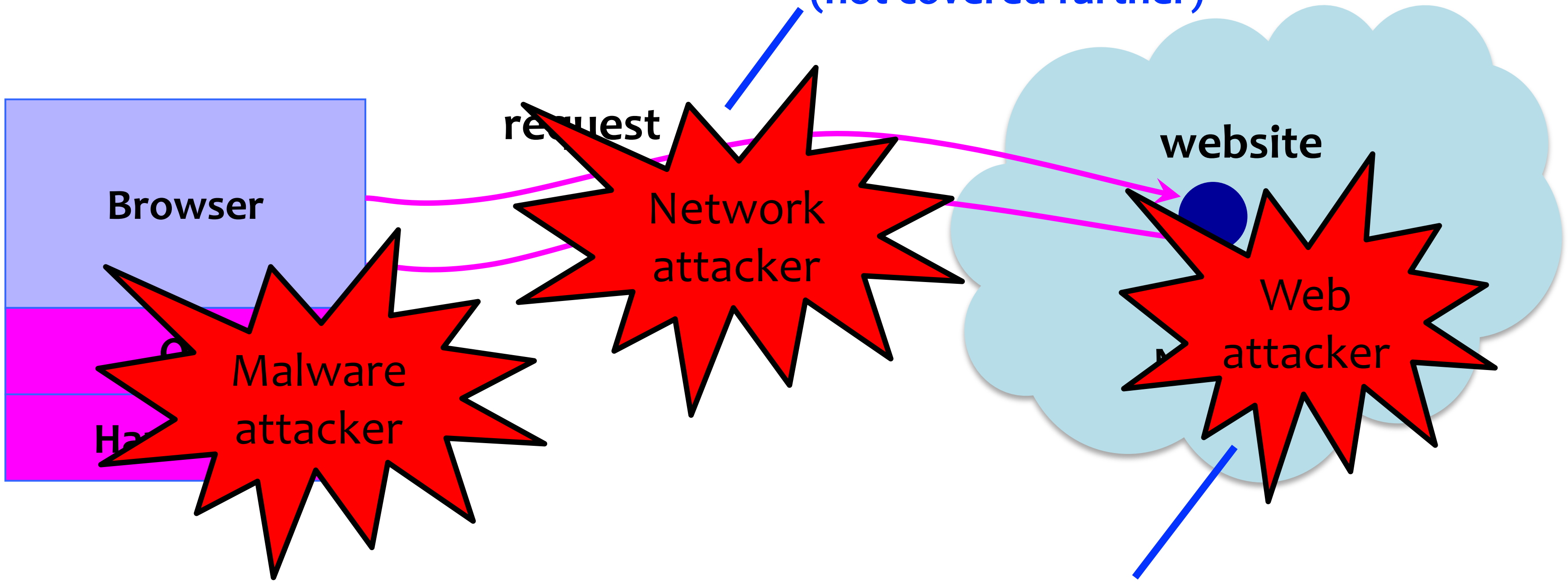
- **Problem:** browsers will think nothing is wrong with a rogue certificate until revoked
- **Goal:** make it impossible for a CA to issue a bad certificate for a domain *without the owner of that domain knowing*
- **Approach:** auditable certificate logs
  - Certificates published in public logs
  - Public logs checked for unexpected certificates

www.certificate-transparency.org

Big Picture: Browser and Network



Where Does the Attacker Live?



Mitigation: SSL/TLS (not covered further)

Mitigation: Browser security model + web app security (next week)

Question: *Why/how would someone visit a malicious site?*

Two Sides of Web Security

(1) Web browser

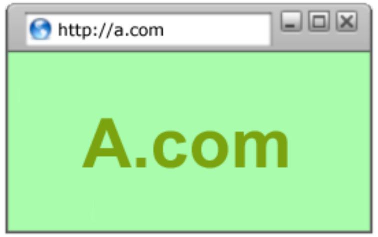
- Responsible for securely confining content presented by visited websites

(2) Web applications

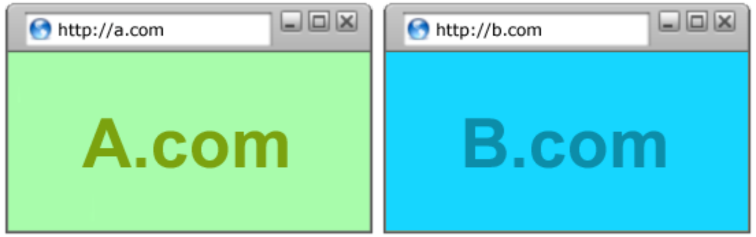
- Online merchants, banks, blogs, Google Apps …
- Mix of server-side and client-side code
  - Server-side code written in PHP, JavaScript, C++ etc.
  - Client-side code written in JavaScript (… sort of)
- Many potential bugs: XSS, XSRF, SQL injection

Browser: All of These Should Be Safe

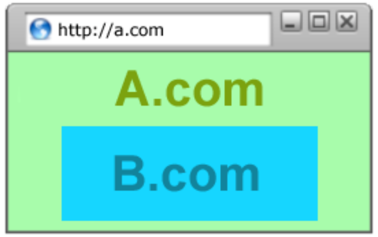
- Safe to visit an evil website



- Safe to visit two pages
  - Simultaneously
  - Sequentially



- Safe delegation



Browser Security Model

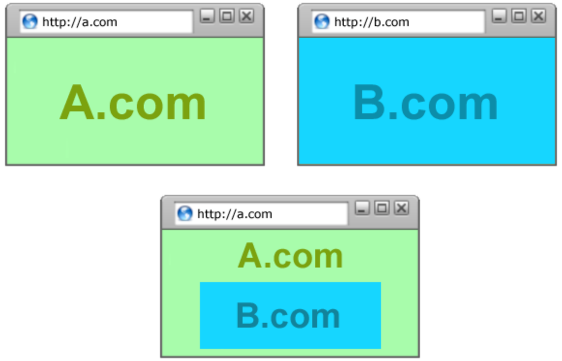
Goal 1: Protect local system from web attacker

→ Browser Sandbox

http://a.com
A.com

Goal 2: Protect/isolate web content from other web content

→ Same Origin Policy

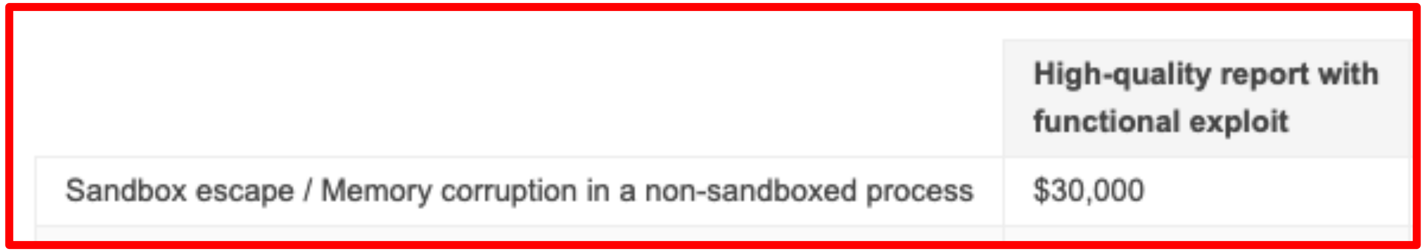


Browser Sandbox

Goals: (1) Protect local system from web attacker; (2) Protect websites from each other

- E.g., safely execute JavaScript provided by a website
- No direct file access, limited access to OS, network, browser data, content from other websites
- Tabs **(newer: also iframes!)** in their own processes
- Implementation is browser and OS specific*

*For example, see: https://chromium.googlesource.com/chromium/src/+/master/docs/design/sandbox.md



	High-quality report with functional exploit
Sandbox escape / Memory corruption in a non-sandboxed process	$30,000

From Chrome Bug Bounty Program