

# CS5321 Network Security

## Week4: PKI Security

**Daisuke MASHIMA**

2022/23 Sem 2

# Public-key Infrastructure for TLS

# Digital certificate

- Signature binding an entity to its public key
- $\text{cert}_{C \rightarrow A}$ : certificate for Alice's key issued by Charlie
  - Alice's key-pair  $(K_A, K_A^{-1})$ , Charlie's key-pair  $(K_C, K_C^{-1})$
  - Say Charlie *knows* that  $K_A$  is Alice's public key, then Charlie computes the following signature:

$$\text{cert}_{C \rightarrow A} = \{\text{Alice}, K_A\}_{K_C^{-1}}$$

- If 'Alice' is not specific enough, use email address, URL, etc
- Alice  $\rightarrow$  Bob:  $(K_A, \text{cert}_{C \rightarrow A})$ ; Bob verifies  $\text{cert}_{C \rightarrow A}$
- If Bob *knows* Charlie's public key  $K_C$  and *trusts* Charlie (i.e., his word 'Alice's key is  $K_A$ '), then Bob can believe that  $K_A$  is indeed Alice's key

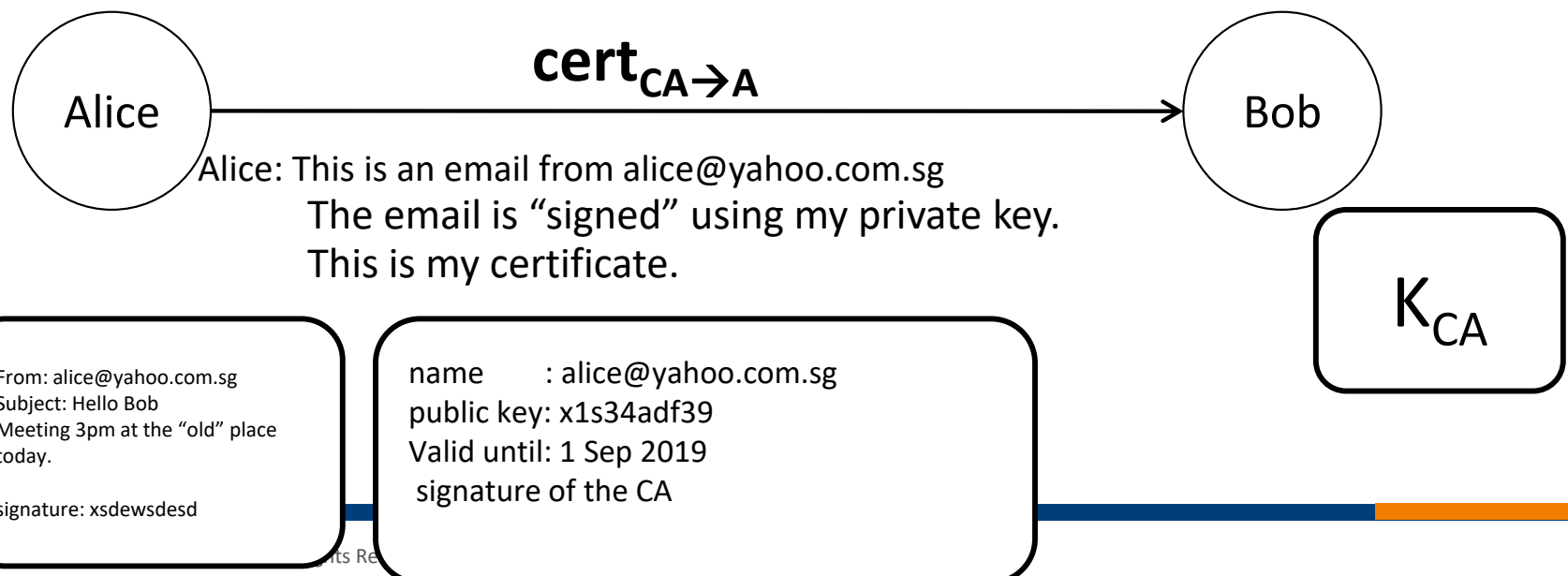
# Certificate authority (CA)

alice@yahoo.com.sg	x1s34adf39
alice@yahoo.com	asd3123411
alice@cs.nyu.edu	2s3dasdf233
apple@google.com	a323fasdfas
.....	

- CA keep the **directory of public keys**. CA also has its own public-private key. We assume that the CA's public key has been securely distributed to all entities involved.

Directory Server (CA)

Bob verifies that the signature in the certificate is indeed signed by the CA. Since no one except the CA can produce the valid signature, the authenticity of the information in the certificate is as good as coming directly from the CA.



# Different PKI models: Single CA


- Single CA
  - Everybody in the system trusts the single CA
    - E.g., organization, government
  - $K_{CA}$  is installed in everyone's machine over authenticated channel (e.g., first day of work, USB stick, pre-installed)

# Different PKI models: Multiple CAs

- Multiple CAs
  - Hard to have single-CA PKI in practice
    - Single, global trusted CA?
    - Single point of failure
  - Multiple CAs issue certificate
    - Alice can obtain  $\text{cert}_{\text{CA1} \rightarrow \text{A}}$ ,  $\text{cert}_{\text{CA2} \rightarrow \text{A}}$ , ...
  - How many CAs do your browser trust?
    - macOS High Sierra: 173 root CAs are trusted  
<https://support.apple.com/en-sg/HT208127>

# Certificate Issuance and Validation Details



- **Self-signed** certificate
  - Self-generated, not CA signed, free
  - E.g. OpenSSL
- **Domain Validated** Certificate
  - Entry-level certificate
  - Only verification check: applicant owns the domain (web site address) associated with certificate
  - No checks to ensure that domain owner is a valid business entity
  - \$400/year (Symantec 2012) or free (e.g., Let's Encrypt)
- **Extended Validation (EV) Certificate** 
  - Much more stringent identity validation
  - URL address bar turns green. Special field with name of web site owner
  - \$1000-\$1500 / year (Symantec 2012)



# Sample Certificate: Gmail



**Equifax Secure Certificate Authority**

↳ **GeoTrust Global CA**

↳ **Google Internet Authority G2**

↳ **mail.google.com**

Expires: Wednesday, May 14, 2014 8:00:00 PM ET  
This certificate is valid

▼ Details

Subject Name

Country US

State/Province California

**Common Name mail.google.com**

Issuer Name

Country US

**Common Name Google Internet Authority G2**

Serial Number 3300100003300983100

Version 3

Signature Algorithm SHA-1 with RSA Encryption ( 1 2 840 113549 1 1 5 )

Parameters none

Not Valid Before Wednesday, January 15, 2014 9:49:14 AM ET

Not Valid After Wednesday, May 14, 2014 8:00:00 PM ET

Public Key Info

Algorithm Elliptic Curve Public Key ( 1 2 840 10045 2 1 )

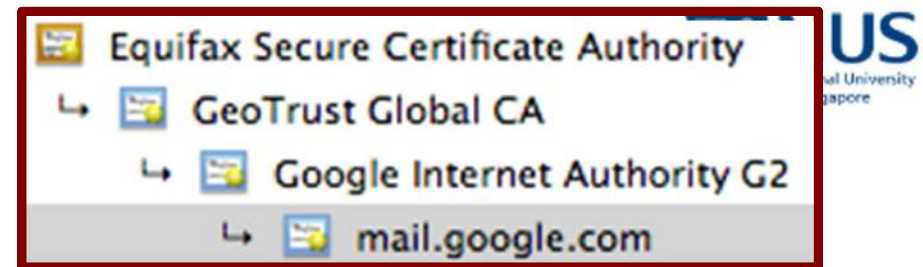
**Public Key 65 bytes : 04 11 85 94 7B 7A 83 DF 7F 88 AE 6  
19 D7 BE E5 5F 4B 30 D7 5B FA DC 1D 26 B1 8**

Key Usage Encrypt, Verify, Sign

Signature 256 bytes : 4E F1 A3 69 BF CF 60 AC ...



# CA Hierarchy



**Root CA**

Equifax CA  $K_{CA}$

“signs”

**Intermediate CA**

GeoTrust:  $\{T, K_T\}_{K_{CA}^{-1}}$

DigiTrust:  $\{B, K_B\}_{K_{CA}^{-1}}$

**Intermediate CA**

IBM:  $\{I, K_I\}_{K_T^{-1}}$

Google IA:  $\{G, K_G\}_{K_T^{-1}}$

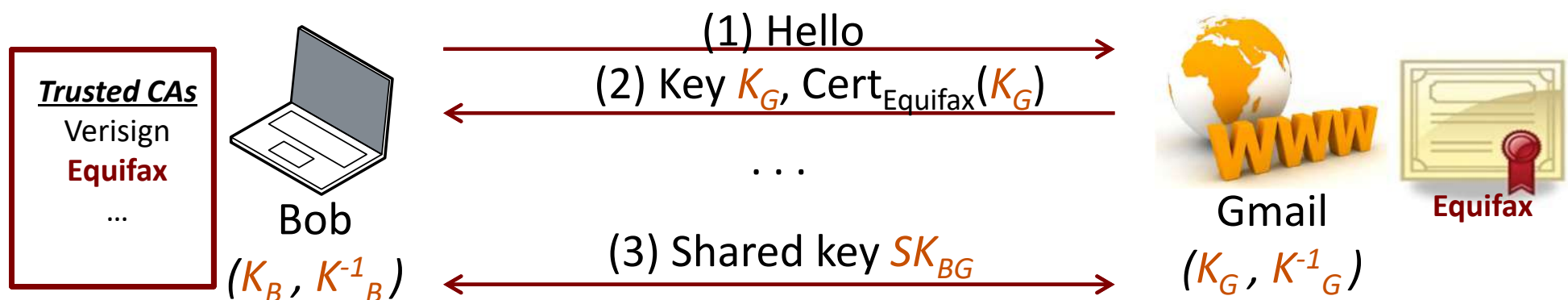
**Domain/service**



Gmail:  $\{M, K_M\}_{K_G^{-1}}$

# Secure Channel using Certificate

- SSL/TLS for encrypted communication
  - Client/browser verifies domain's identity using certificate
    - Browser knows the **public keys of trusted CAs**
  - Uses verified public keys to set up a secure channel



# Problems of Trust Model of Public-key Infrastructure for TLS

# Sample Trusted Root CAs

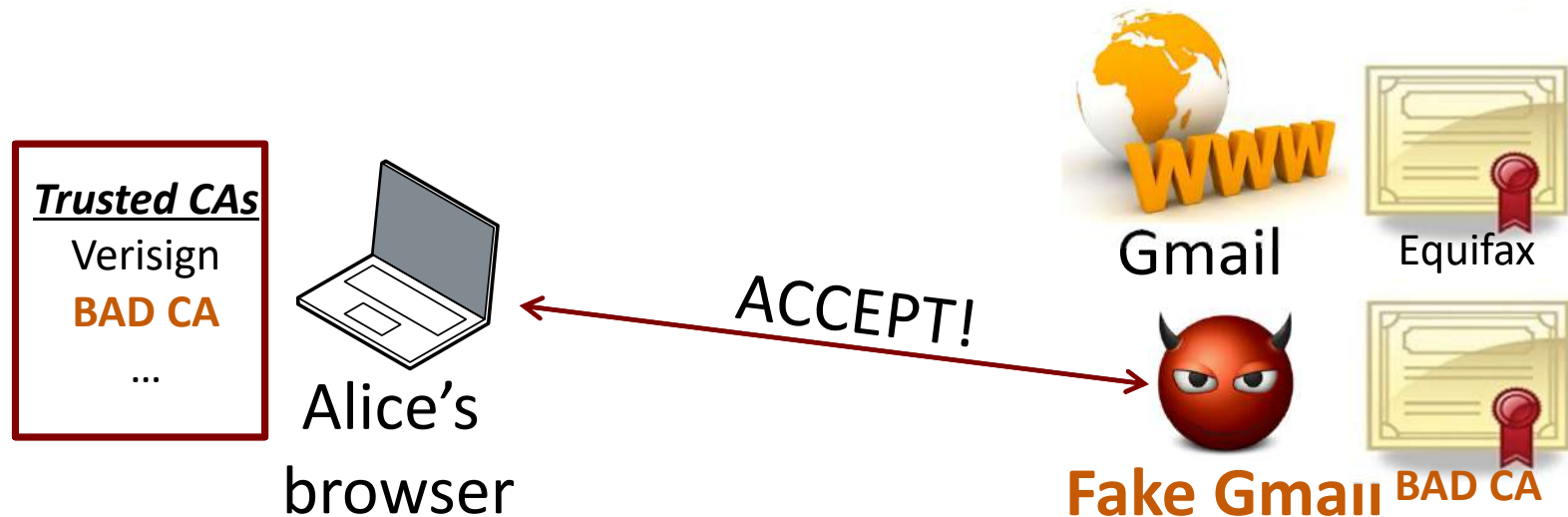
- MSR SV PKI project, 2012 [1]
  - 337 root certificates trusted by Microsoft IE
  - 1,510 intermediate CA certificates
- macOS High Sierra: 173 root CAs are trusted
  - <https://support.apple.com/en-sg/HT208127>

Trusted certificates

Certificate name	Issued by	Type	Key size	Sig alg	Serial number	Expires	EV policy
AAA Certificate Services	AAA Certificate Services	RSA	2048	SHA-256	01	22:59:59	Not EV
Actalis Authentication Root CA	Actalis Authentication Root CA	RSA	4096 bits	SHA-256	57 0A 11 97 42 C4 E3 CC	11:22:02 Sep 22, 2030	1.3.159.1.17.1

What if this CA signs {"mail.google.com",  $K_{\text{attack}}$ }?

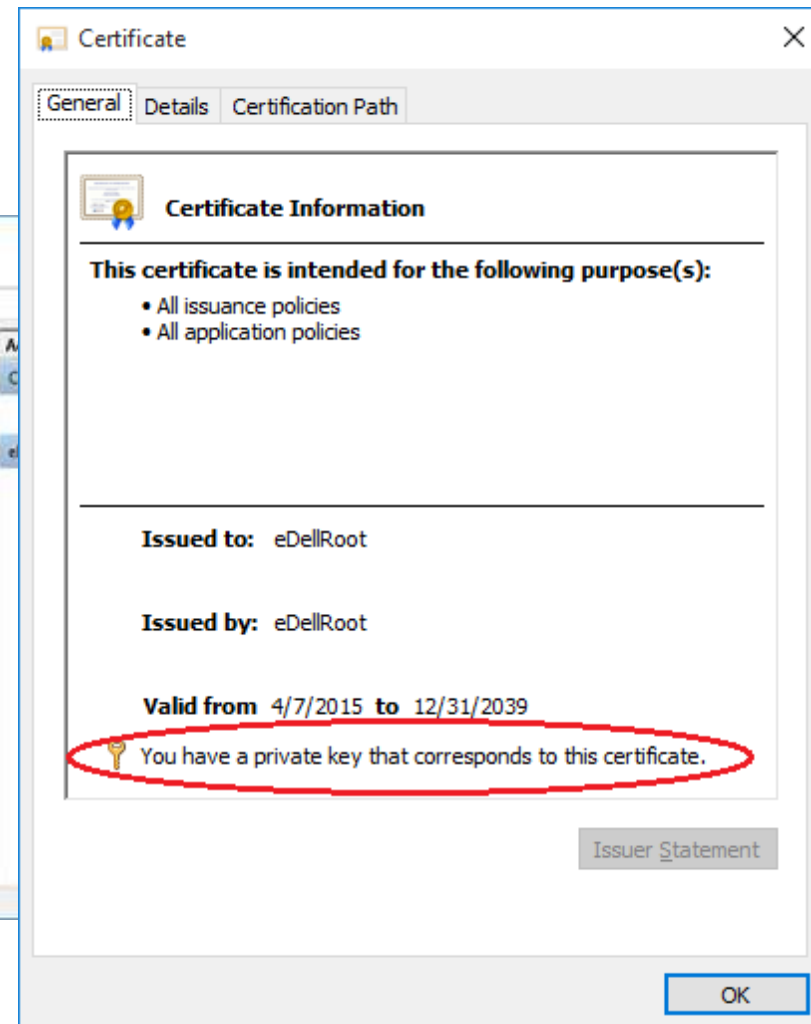
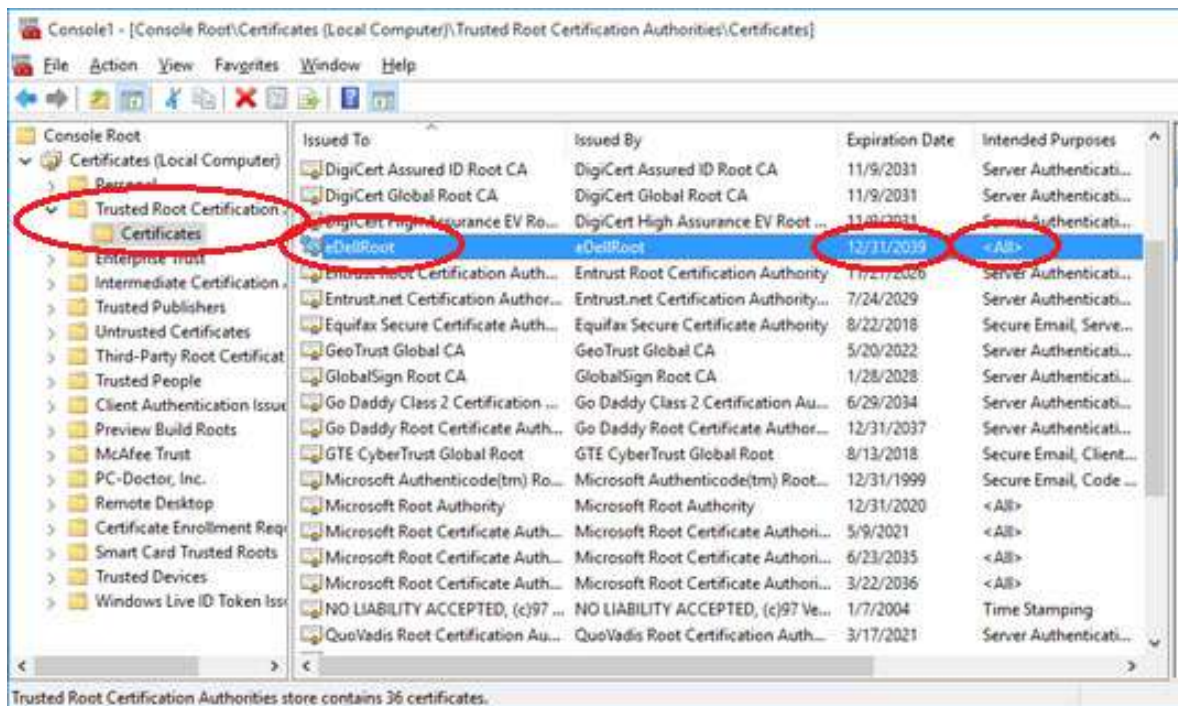
# CA Single Point of Failure



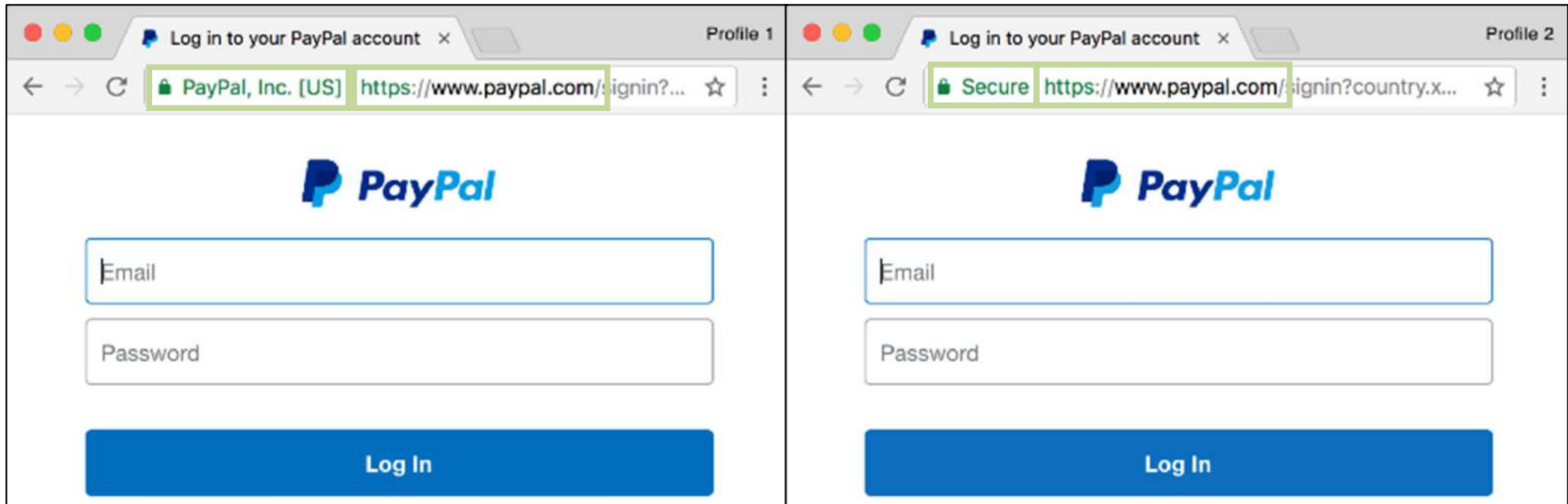
- Security of the weakest link
  - Security breach of a single CA → Compromise security of sites **protected by any other CA!**

# CA Single Point of Failure

- eDellRoot Certificate in 2015
  - Dell shipped laptops that trust self-signed root CA certificates
  - Private key is also included!

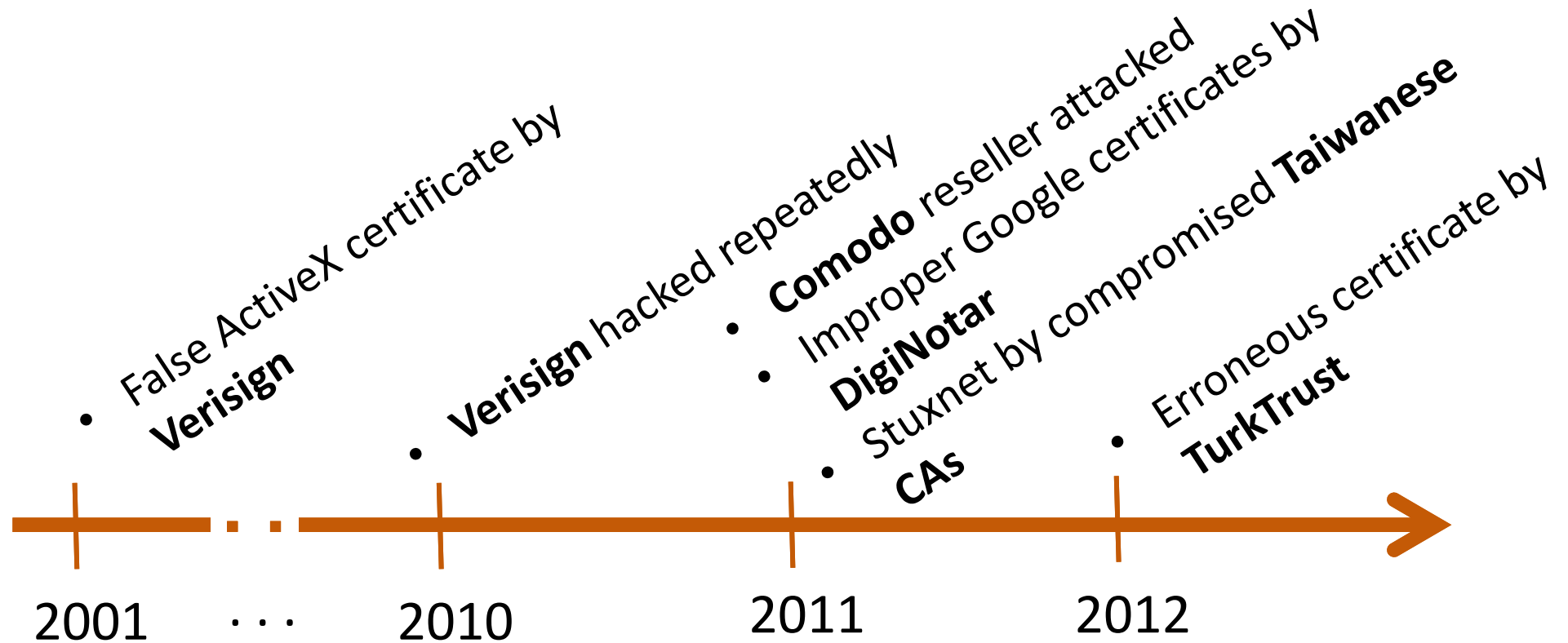


# CA Single Point of Failure



- VeriSign Class 3 Public Primary Certification Authority – G5
  - Symantec Class 3 EV SSL CA – G3
    - www.paypal.com
- **eDell Root**
  - By means of a fake (but successfully verified) certificate issued using eDellRoot, MITM attack is possible.

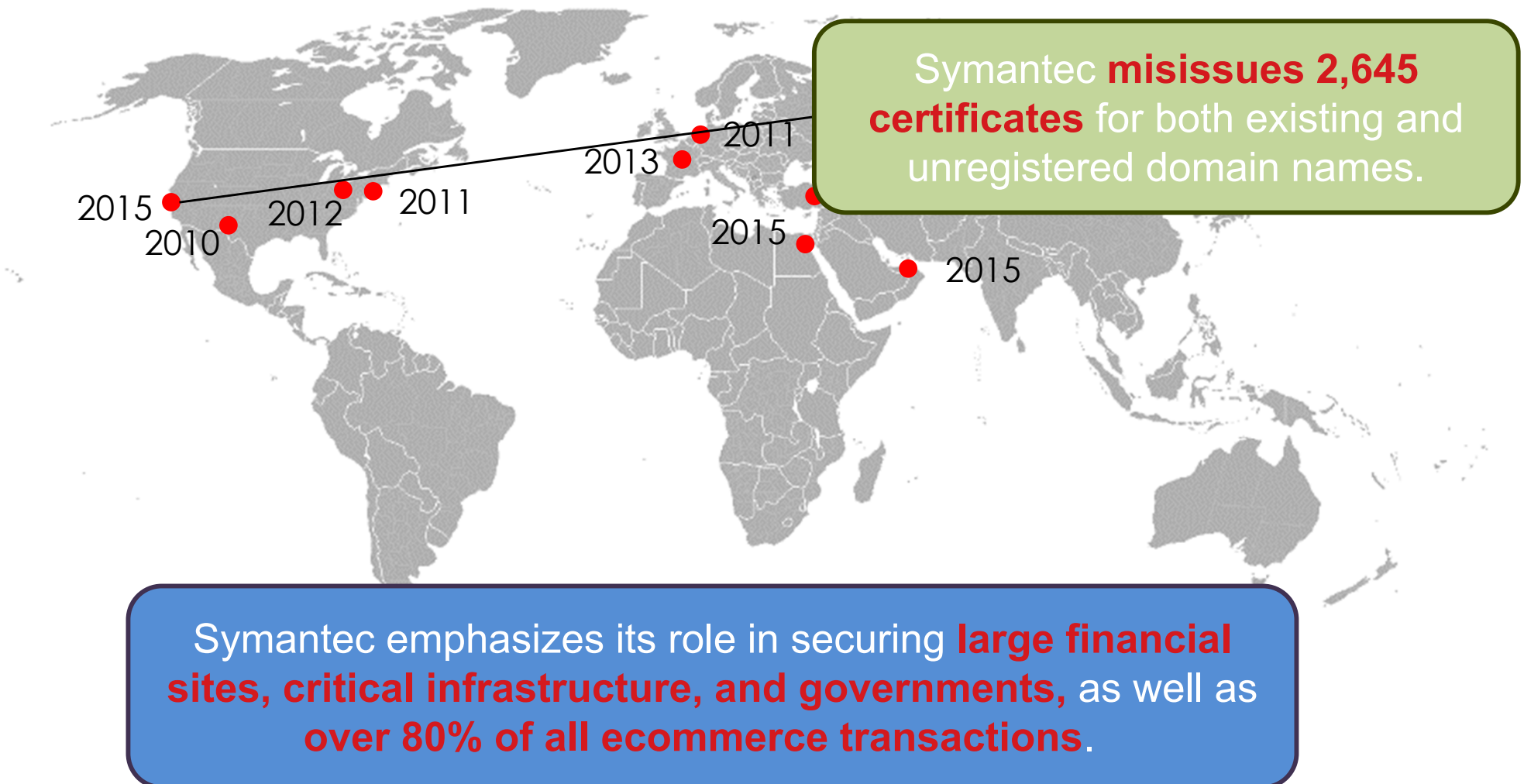
# CA Breach Events



- *Possibly a large number of CA breaches are concealed*



# More Breach Events...



# Compelled Certificates

- Certified Lies: Detecting and Defeating **Government Interception** Attacks against SSL by Christopher Soghoian and Sid Stamm
  - <https://s3.amazonaws.com/files.cloudprivacy.net/ssl-mitm.pdf>
- Compelled certificate: public/private key pair for **law enforcement** (MitM attacks on SSL) with CA certificate enabling private key to sign additional certificates

# MitM Device using Compelled Certificates

PACKET FORENSICS

VOLUME 1 • NO. 1 • 2009

## Man-in-the-Middle Capabilities

Intercept any communication within Secure Socket Layer (SSL) or Transport Layer Security (TLS) sessions

All Packet Forensics targeting and policy capabilities can operate within the encrypted tunnel

### Contacts

Offices in  
Virginia and

capability to become a go-between for any TLS or SSL connections in addition to having access to all unprotected traffic. This allows you to conditionally intercept

the subject a false sense of confidence in its authenticity.

Of course, this is only a concern for communications incorporating PKI. For most

To use our product in this scenario, users have the ability to import a copy of any legitimate key they obtain (potentially by court order) or they can generate “look-alike” keys designed to give the subject a false sense of confidence in its authenticity.

PACKET FORENSICS

## SMALL DEVICES. BIG OPPORTUNITIES.

INTRODUCING THE 5-SERIES

Packet Forensics 5-Series are the most flexible tactical surveillance devices in the world of IP networks. Designed for defense and (counter) intelligence applications, they are fully-embedded without moving parts and available in a variety of sizes, shapes and power footprints, all customized for the client. In under five minutes, they can be configured and installed in-line without knowledge of existing network topology. **Capabilities include:** Keyword, RADIUS, DHCP and behavior-based session identification; filtering, modification and injection of packets; compatibility with existing collection systems. With this modular platform, Packet Forensics



creates mission packages based on customer requirements. Best of all, they're so cost effective, they're disposable--that means less risk to personnel.



### Introduction

The 5-Series is a turnkey intercept solution in an appliance platform. Offering the most flexible approach to network surveillance and novel approaches to rapid deployment and stealthy reporting of captured data, the 5-Series devices are unmatched in the industry.

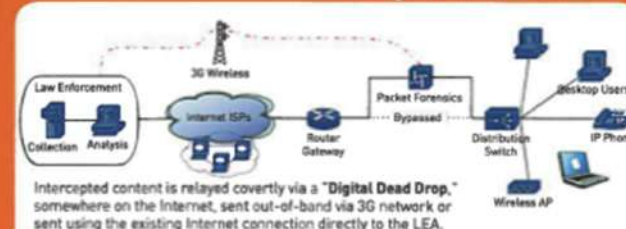
An attractive feature of the 5-Series is its ability to passively discover network topology--this allows an individual to deploy it with no prior knowledge of the target network. The device can be placed in-line and immediately act as a passive bridge while performing its mission. As intelligence is being gathered and the device has an understanding of the network, it uses its stealth reporting techniques to return captured information or accomplish a variety of other missions.

The 5-Series has no MAC address or IP address; it dynamically masquerades as the most appropriate host that sits topologically behind it. The 5-Series can be used to intercept and record matching sessions to internal flash-memory, or report them upstream using a variety of protocols. In the most hostile environments, this upstream reporting can be accomplished using a technique that makes the 5-Series' presence undetectable using standard network security methods.

### The Internet Cafe

The 5-Series is an ideal solution to the "Internet Cafe Problem." Quick deployment and remote control minimize personnel risk and maximize collection capabilities. Small footprint and minimal power requirements make installation easy.

### Solving the Internet Cafe Problem



### Key Advantages

- Customized mission packages
- Small form-factor, solid-state (as small as 4 square inches)
- No moving parts, highly reliable
- Battery, PoE or wired power
- Hardware bypass, fail-safe
- Tamper detection, fail-secure
- Up to Gb/sec throughput
- Deployable with no knowledge of target network topology
- Supports stealth upstream reporting (practically undetectable)
- "Digital Dead Drop" delivery
- Triggers intercepts based on keywords, RADIUS, DHCP, behavior or other subject criteria
- Probe and Mediation capabilities
- Performs dialed digit extraction
- Packet modification, injection and replay capabilities
- Packet Forensics software stack and PeerTalk™ technology
- Advanced firmware-update keeps software up-to-date

Things can go wrong...  
but how to fix it?

***“Revocation!”***



# Certificate Revocation



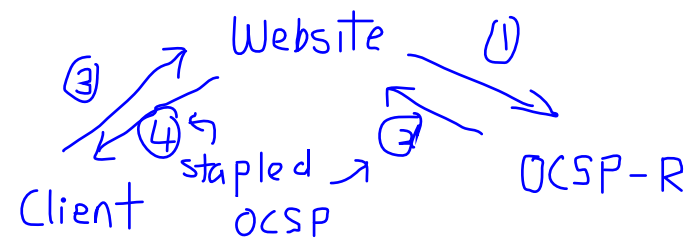
- Certificate revocation is a mechanism to invalidate certificates
  - After a private key is disclosed
  - Employee leaves corporation
  - Certificate expiration time is usually chosen too long (updating certificates is a lot of work)
- CA periodically publishes **Certificate Revocation List (CRL)**
  - Issued in hourly, daily, or weekly basis
  - Delta CRLs only contain changes
  - What to do if we miss CRL update?

# Short-Lived Certificates

- Certs have short validity lifetime (e.g., few days)
  - Servers update certs from CAs daily
  - If unable to update → previous cert is still valid for next few days
- Advantage
  - Timely information regarding the revocation status
- Disadvantage
  - Too much overhead to CAs and domain owners

# OCSP Extension

- **Online Certificate Status Protocol**
  - Client query to get certs' revocation status
- Protocol
  - $A \rightarrow CA_B$ : OCSP request for Bob's  $cert_B$  signed by  $CA_B$
  - $CA_B$ 's OCSP responder: Checks  $CA_B$ 's status database
  - $CA_B$ 's OCSP responder  $\rightarrow A$ : sends a response=good/revoked/unknown
- Advantages
  - Timely information regarding the revocation status
- Disadvantages
  - Traffic overhead to CAs for querying
  - **Privacy**: CA learns user's activity
- **OCSP stapling**: Certificate holder (e.g., website) to provide CA-signed & time-stamped OCSP response along with its own certificate
  - but large response size



# Can we detect malicious/misissued certificates?



# Public Key Pinning

- Google Chrome **used to** maintain **HTTPS pins (HPKP)**:
  - List of trusted public key(s) for each site
    - E.g., “The whitelisted public keys for Google currently include Verisign, Google Internet Authority, Equifax and GeoTrust.”  
(<https://www.imperialviolet.org/2011/05/04/pinning.html>)
  - Certificate chain for a domain must include a **trusted public key**
    - otherwise, fatal error
- Advantage
  - Prevents browser from accepting certs signed by a rogue CA
- Disadvantages
  - No protection against compromised CAs who are pinned
  - Pins should not be easily replaced. Should expire after a certain expiration time.
    - If pinning duration is too long → long period of unavailability
  - Deactivated in 2018

# Perspectives (2008)

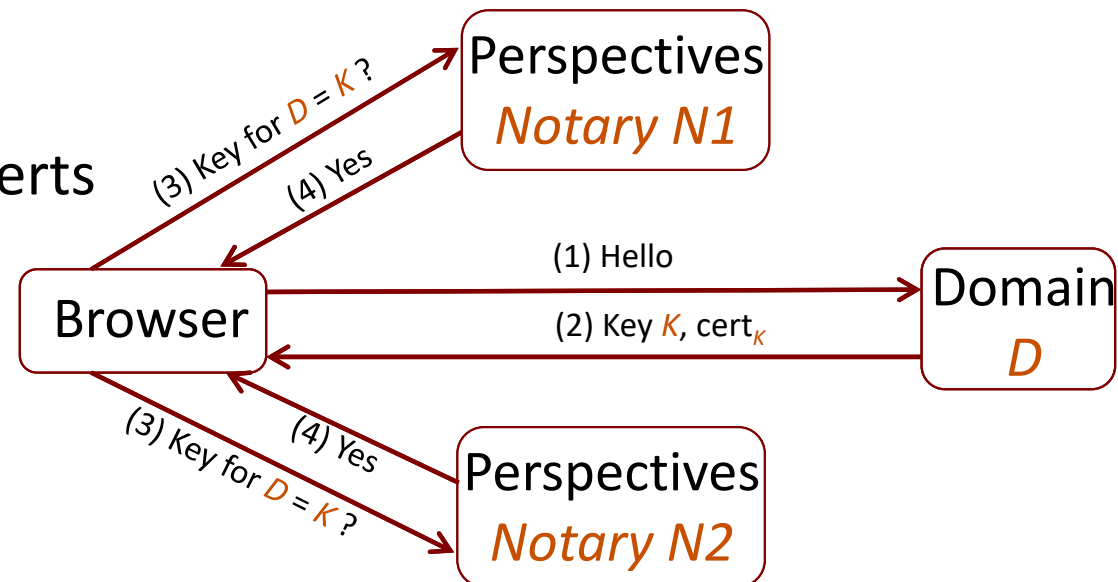
- Globally distributed **notary servers**
  - Contact known SSL/TLS servers once a day
  - Fetch current server's certificate
  - Store entire history of observed certs & support queries
- Users configure a set of trusted notaries
  - No single point of failure

## ADVANTAGE

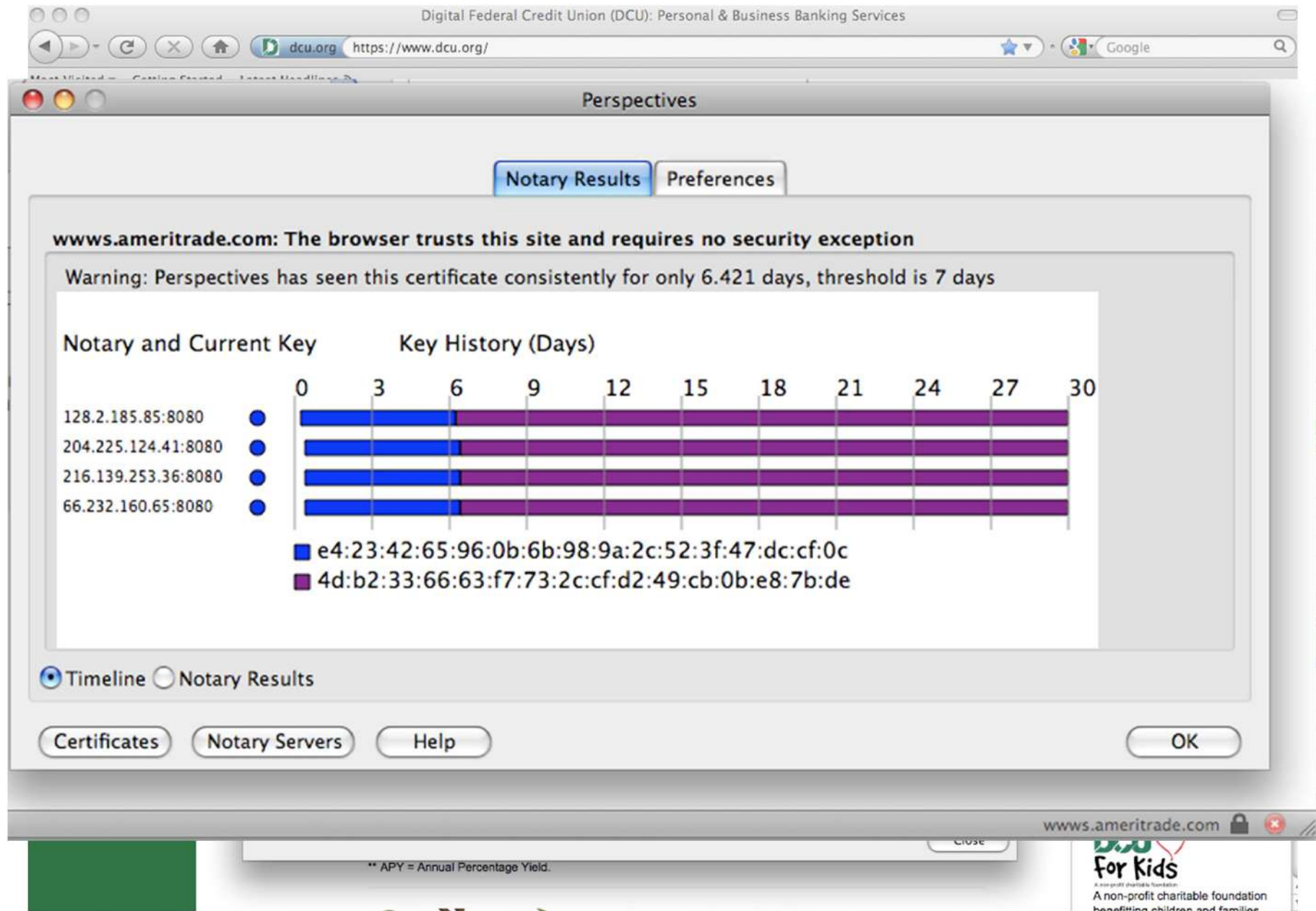
- Detect MitM, compelled certs

## DISADVANTAGES

- Requires additional connection
- Privacy: notary learns site

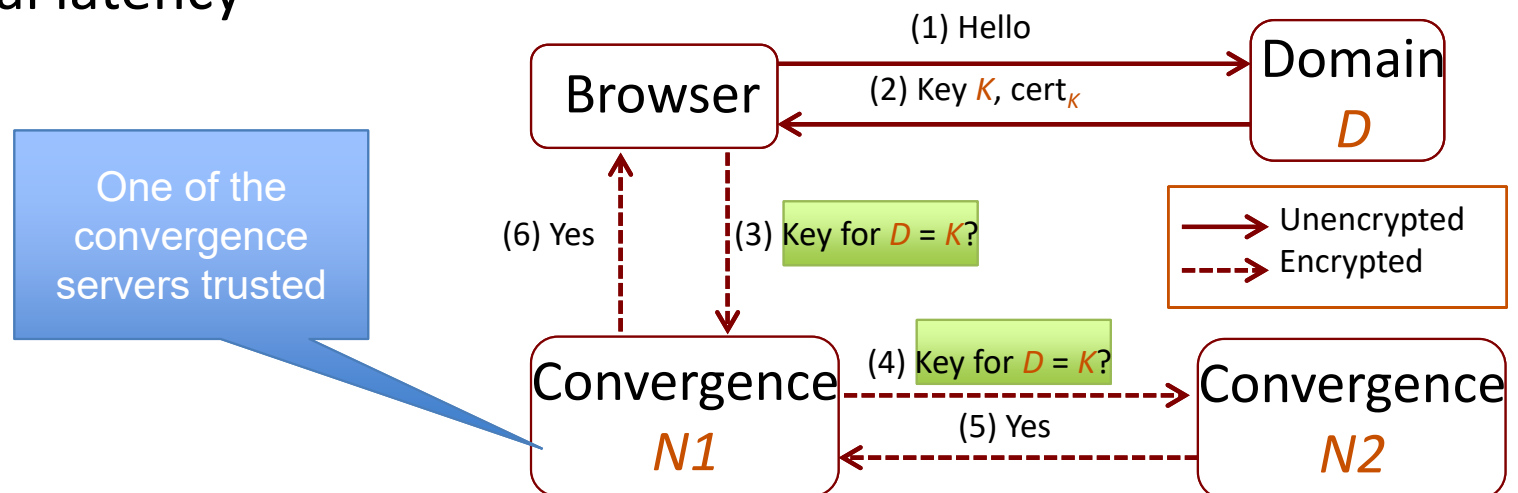


# Perspectives in Action



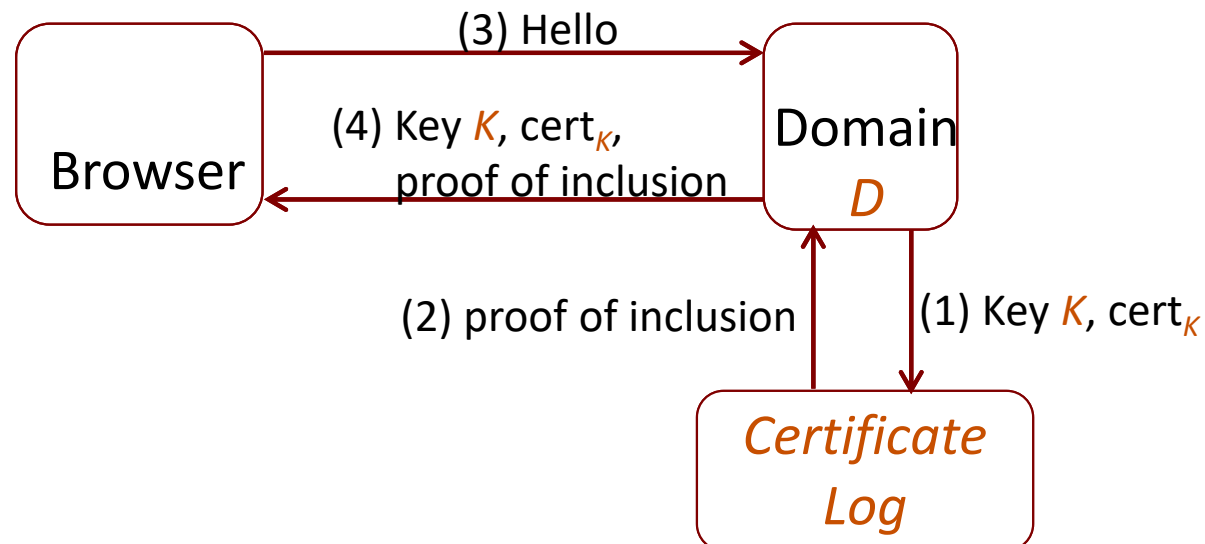
# Convergence

- Enhancement of Perspectives
  - Privacy for cert lookup using 2-step onion routing approach
    - 1<sup>st</sup> Convergence server: redirects query to 2<sup>nd</sup> server
      - Know who you are, but don't know what you are querying
    - 2<sup>nd</sup> Convergence server: responds to 1<sup>st</sup> server
      - Know what you are querying, but don't know who you are
- **ISSUE**
  - Increased performance cost
  - Additional latency



# Certificate Transparency (2013- by Google)

- Can't we just make certificates transparent?
  - If all (either authorized or unauthorized) certificates are visible to everyone, misbehavior would be also visible
- Certificate log (CL)
  - Public, verifiable, and append-only log of TLS certificates
- Browsers *reject* if certificate is NOT in one of the CLs

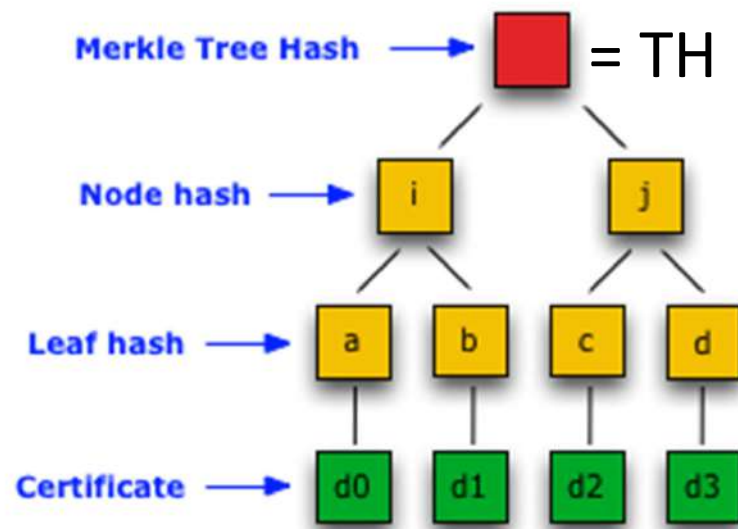


# How to build a public, verifiable, and append-only logs?

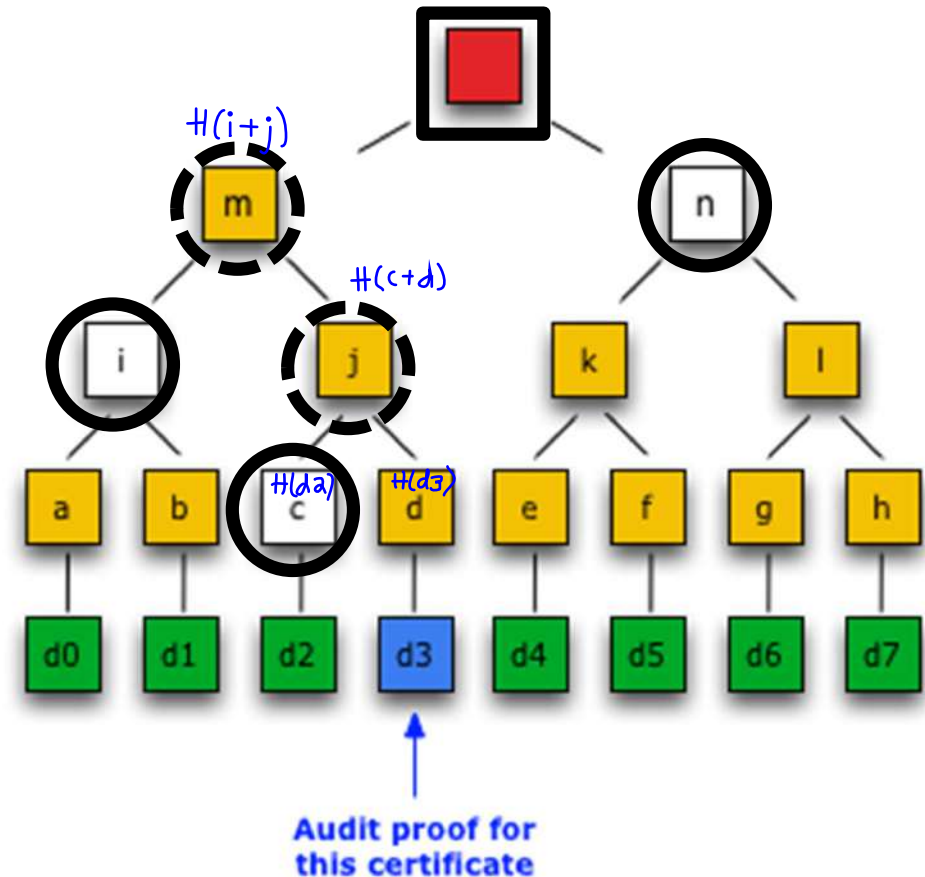
- One obvious approach
  - One global log
  - Each client *continuously downloads* the entire log and check the *append-only property*
  - Clients compare their copies with each other to check that the log is *public*
- Any problem with this approach?
  - Huge waste of resource

# A better approach: Merkle tree

- Merkle Hash Tree Structure
  - Tree head (TH) or root of the tree is a summary of *all* the certificates
  - Distribute signed tree head (STH):  $\{TH\}_{K_{\log}^{-1}}$



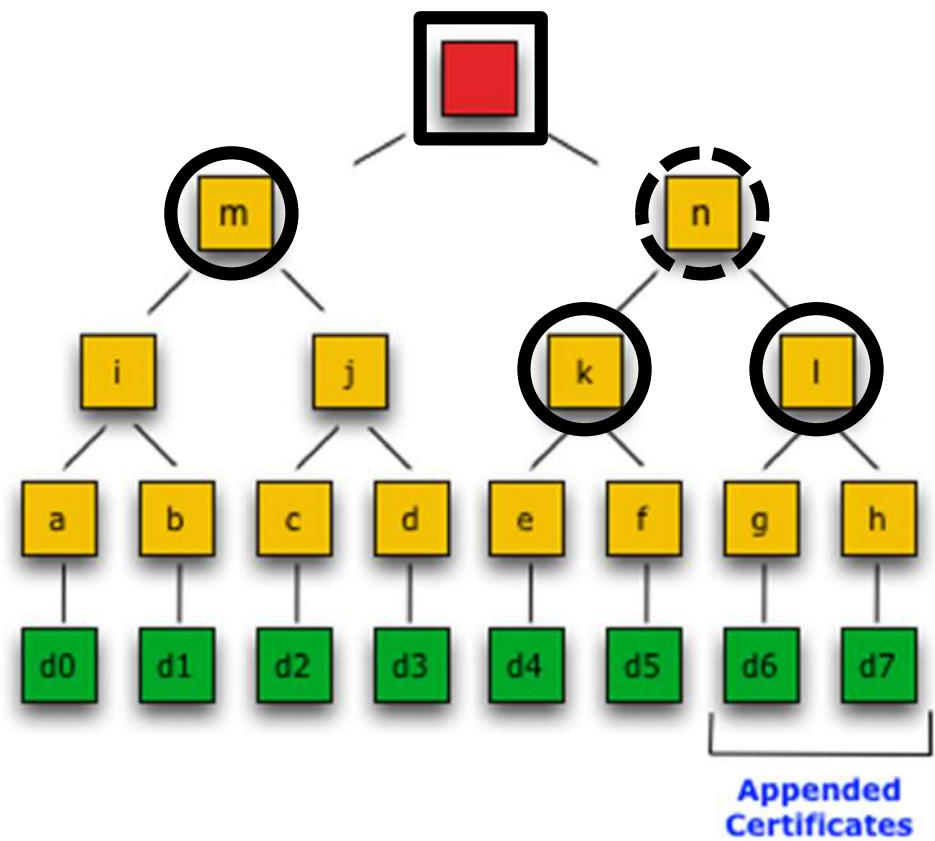
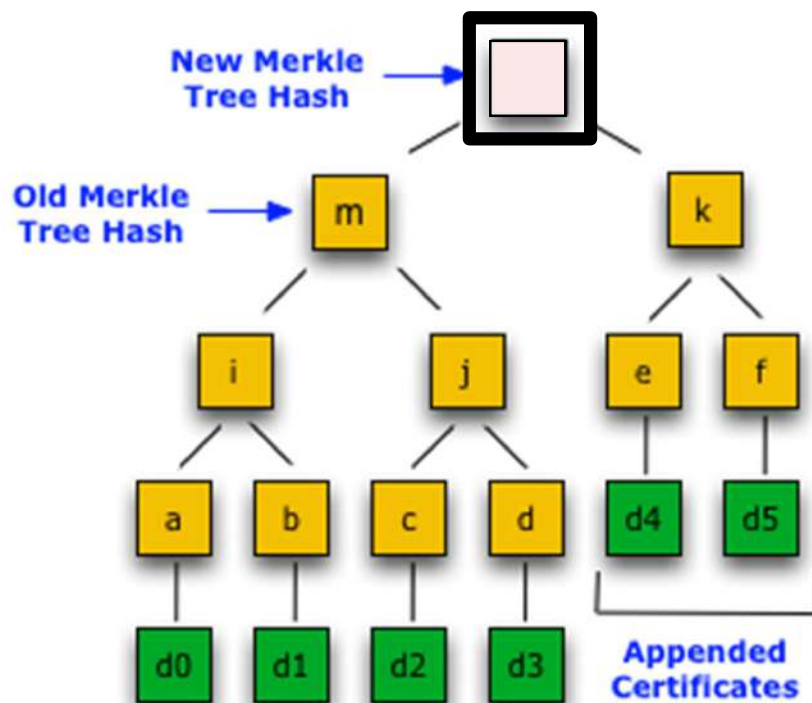
Proof of inclusion?



*Is this enough?*

# Append-only property with Merkel tree

- “Does the later version include the previous version?”
- Verify two versions of logs: (1) ver1 is included in ver2; and (2) new entries come after old ones

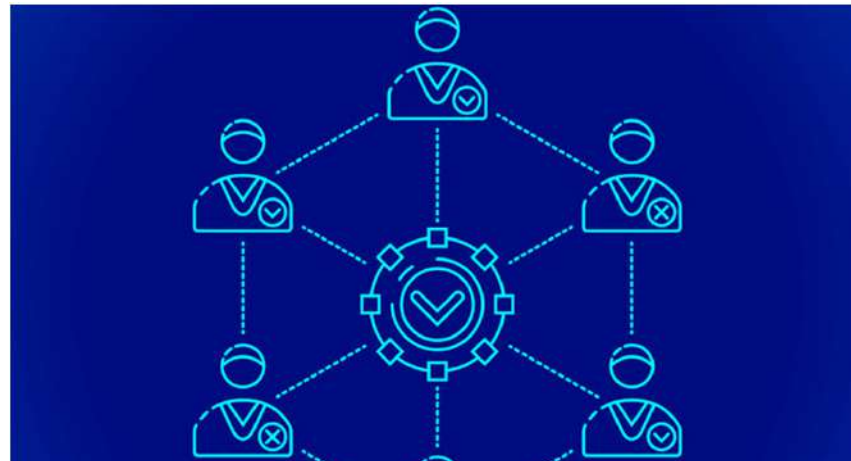


<http://www.certificate-transparency.org/log-proofs-work>



# Availability vs. consistency

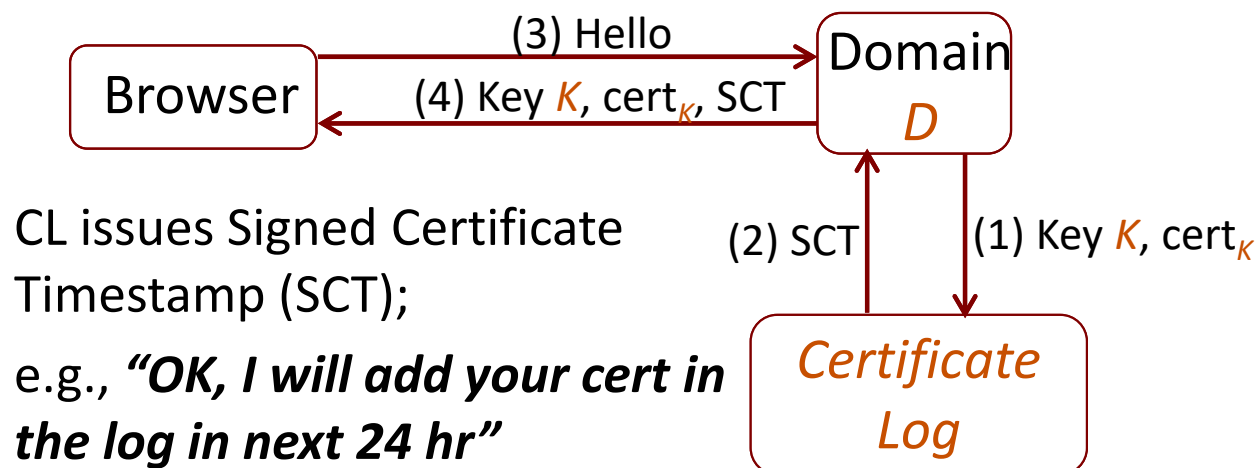
- Log must be highly available
  - Multiple log instances in distributed geographical regions
- How to make these logs consistent?
  - Logs reach a consensus among them?



What about ***latency*** (e.g., time to register a new certificate and get the proof of inclusion)?

# CT's Design choice: SCT and MMD

- When CA wishes to register a certificate to CT:
  - CA sends the certificate to **a number of logs**
  - Each log immediately issues a **signed certificate timestamp (SCT)** as a **promise** that it **will insert** this certificate to its Merkel tree
    - With a promise that the actual insertion will take place within a **maximum merge delay (MMD)** (e.g., 24 hours)
  - CA can use the **certificate + SCT** from that moment

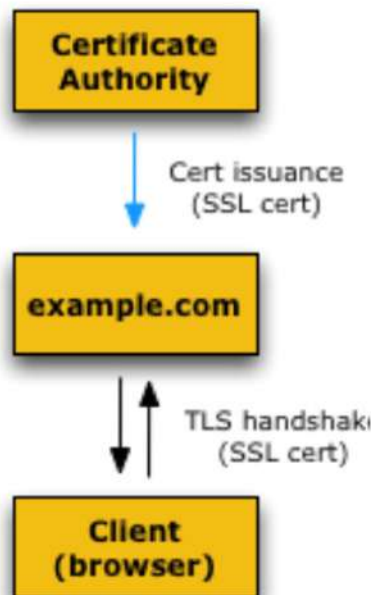


# How SCT can be incorporated

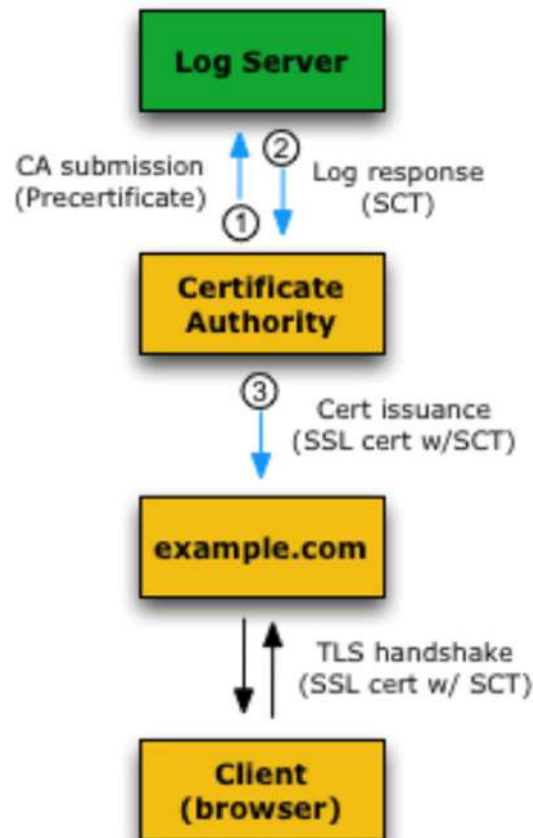
SCT is part of cert.  
Server does not need any change.

- Existing TLS/SSL system
- Supplemental CT components
- One-time operations
- Synchronous operations
- Order of operation

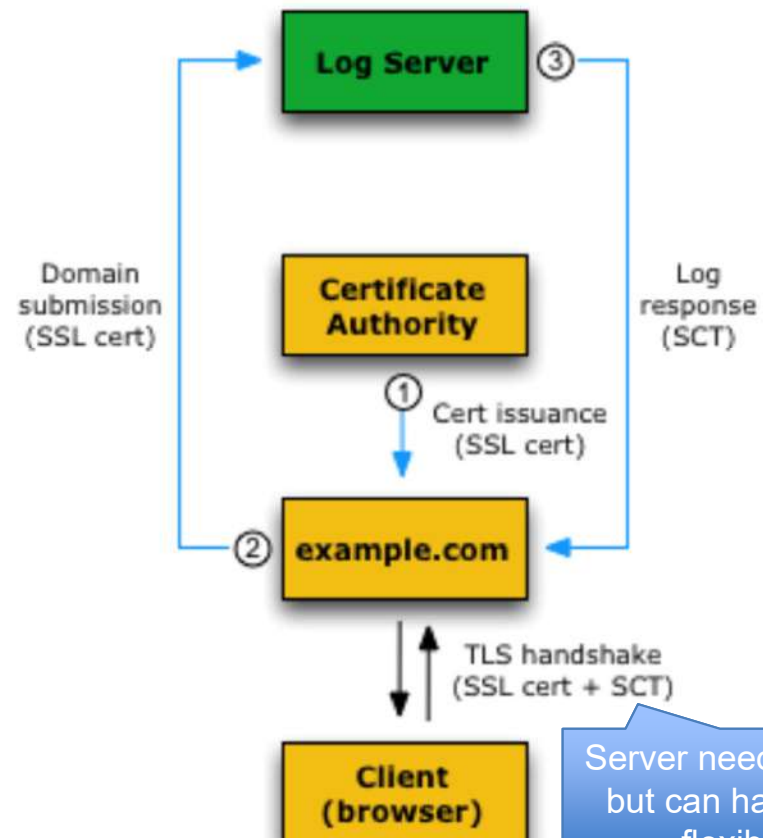
Current TLS/SSL System



TLS/SSL System with Certificate Transparency (X.509v3 Extension)



TLS/SSL System with Certificate Transparency (TLS Extension)



Server needs change but can have more flexibility.

# A good ecosystem is needed

- Any interested parties (e.g., CAs, domain owners, clients) can monitor logs to ensure:
  - CAs do not register improper certificates
  - Logs do not misbehave; for example,
    - violating append-only property
    - not adding certificate within MMD
    - presenting different logs to different clients
- Community effort needed
  - Any violation or misconduct should be known to all
  - Clients may gossip (including signed tree hash)
  - Auditors can identify misbehavior

# Certificate Transparency Operation



- Multiple untrusted Logs can be operated
  - Operated by Google, Symantec, DigiCert, Cloudflare
  - Logs are untrusted (*anyone* can run)
- Browsers
  - Chrome requires CT for certificate since Apr 2018
- Monitors/auditors
  - Check if logs misbehave (e.g., remove certificates, not inserting certificates)
  - Check if logs are consistent when updated
  - Misbehaving logs are detected
- Revocation?
  - CRL/OCSP can be used
  - CT helps misbehavior detection but **not revocation**

So, CT solves all the problems?

# Current state of CT and new concerns\*

- Rapid increase of CT in recent years
  - 33% of established connections supporting CT (as of 2018)
- (-) New security concern: DNS leak
  - non-public domains could be seen by the public with CT
  - **Subdomain enumeration** is possible!
- (-) Attackers are already looking at CT
- (+) Easily detect phishing domains
- More studies are needed

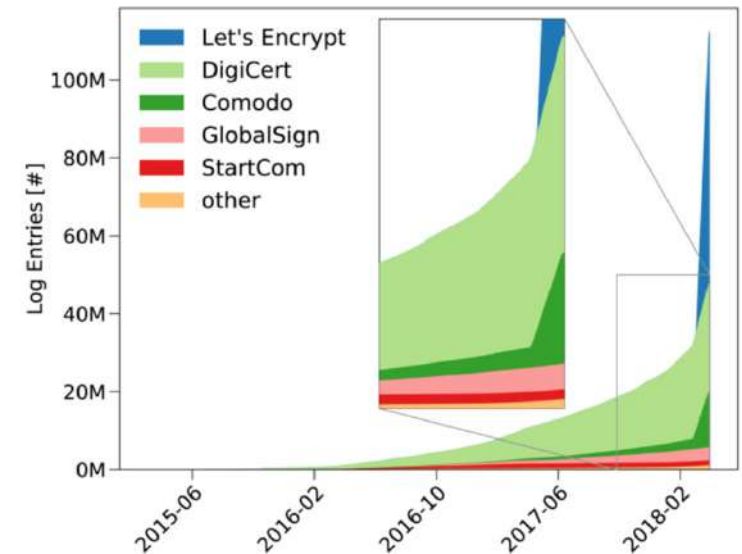


Table 2: Top 20 subdomain labels (SDL) in CT-logged certificates.

	SDL	Count		SDL	Count		SDL	Count
1	www	61.1M	8	shop	303k	15	secure	176k
2	mail	14.4M	9	whm	280k	16	admin	158k
3	webdisk	8.7M	10	dev	256k	17	mobile	156k
4	webmail	8.6M	11	remote	253k	18	server	146k
5	cpanel	8.2M	12	test	249k	19	cloud	141k
6	autodiscover	3.6M	13	api	239k	20	smtp	140k
7	m	310k	14	blog	235k			

Table 3: Potential phishing domains identified in CT.

Service	Count	Example
Apple	63k	<i>appleid.apple.com-7etr6eti.gq</i>
PayPal	58k	<i>paypal.com-account-security.money</i>
Microsoft	4k	<i>www-hotmail-login.live</i>
Google	1k	<i>accounts.google.co.am</i>
eBay	<1k	<i>www.ebay.co.uk.dll7.bid</i>

(\*) Scheitle, Quirin, et al. "The rise of certificate transparency and its implications on the internet ecosystem." Proceedings of the Internet Measurement Conference 2018.

# Summary: PKI Security



- Public-key infrastructure is a de facto security infrastructure for encryption/signature in large system (e.g., web)
- Existing PKI system follows delegated CA + Oligarchy model
  - Poor security: too many trusted entities, hard to handle many attacks
- Certificate Transparency is becoming a new standard
- Many PKI security problems are still open!
  - Understanding the complexity of the problem is necessary.

# NEXT WEEK: TCP/IP SECURITY



# Two papers

- **Accountable Internet Protocol (Sigcomm'08)**
  - Problem: current IP has no accountability
  - Solution: let's make it accountable
    - Useful features of AIP?
    - Too radical?
    - How to address many new problems it'd create?
- **TCP Sequence Number Inference Attack (CCS'12)**
  - TCP seq number should be hard to predict/infer; otherwise, TCP session can be easily hijacked
  - Discovery: many operating systems leak TCP seq numbers, making TCP hijacking possible