CS 465
Computer Security

Public Key Infrastructure and Certificates

Daniel Zappala, adapted from Kent Seamons
Fall 2018

Public Key Infrastructure

- Hardware, software, and policies needed to create, store, manage, distribute, and revoke public keys and digital certificates

- Key management includes this and also management of private keys and symmetric keys

Digital Certificates

- Certificates are designed to bind a subject to the subject's public key
  - A subject needs an identifier, such as a company name, person's name, email address, etc.
  - Digitally signed by another entity, in some cases another person and most commonly an organization called a Certificate Authority
- Can solve the key distribution problem for public keys by narrowing the problem to the secure distribution of public keys for the CAs

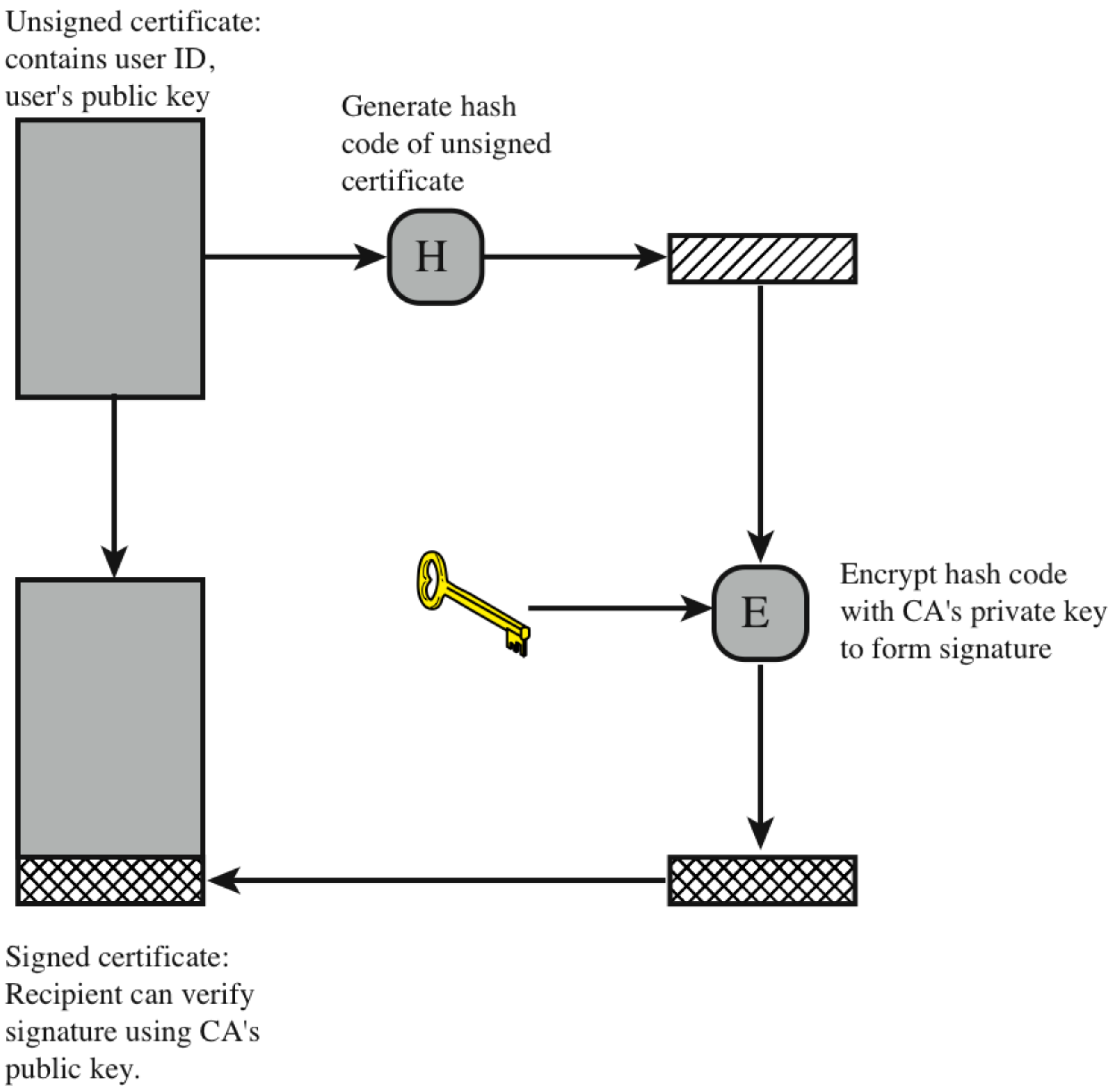
Certificate Signing

- Digital signature = encrypt hash of certificate with private key of CA

- Usually the subject generates the key pair and the CA only sees the public key. The CA challenges for ownership of the private key.

- Corporate software may keep a copy of the private key so that you can't lose it or so they can inspect encrypted traffic

- Certificates typically include an expiration date


Unsigned certificate: contains user ID, user's public key
Generate hash code of unsigned certificate
H
E
Encrypt hash code with CA's private key to form signature
Signed certificate: Recipient can verify signature using CA's public key.

Source: Stallings, Network Security Essentials

Certificate Revocation

- What if your private key is lost or stolen?
- Ask the CA to revoke your certificate
  - they will investigate
  - see, e.g. https://www.digicert.com/certificate-revocation.htm
- Often put on a Certificate Revocation List (CRL) that can be queried

Certificate Verification

- Steps performed by a relying party (e.g., web browser)
  - Integrity — verify the has has been signed by a CA you trust and the subject is the one you are expecting
    - May require checking a chain of certificates
  - Expiration — check the expiration date
  - Revocation — check the CRL or other revocation mechanisms
  - Limits on the keys — e.g. whether the certificate allows it to be used for signing additional certificates
  - Ownership — does the entity presenting the certificate have access to the associated private key?

Certificate Hierarchies

- The set of valid certificates forms a tree
    - Digicert issues a certificate to BYU
    - BYU CA issues certificates to College CAs
    - College CAs issue certificates to Department CAs
    - Departments issue certificates to students
- Verifying a certificate chain
    - The relying party could only have the BYU public key
    - The client or server has to discover the certificate chain – one method is for the client to deliver the chain to the server, another is to include links to where next certificate in the chain can be downloaded

Certificate Hierarchies

- What if the college has a private key compromised?
    - College has to generate a new key pair and get BYU to sign a certificate with the new public key, plus revoke old certificate
    - College has to sign department public keys again with the new key private key, revoke and re-issue department certificates
    - Student certificates are OK
- What if the department has it’s key compromised?
- Only have to re-sign certificates one level below in the hierarchy. Don’t need to re-create the entire hierarchy

PKI Reality

- Names – how to identify subjects?
- Authorities – who can sign certificates?
- Trust – who do we trust as authorities?
- Revocation – hardest PKI problem to solve

Source: Cryptography Engineering, Ferguson et al., Chapter 19

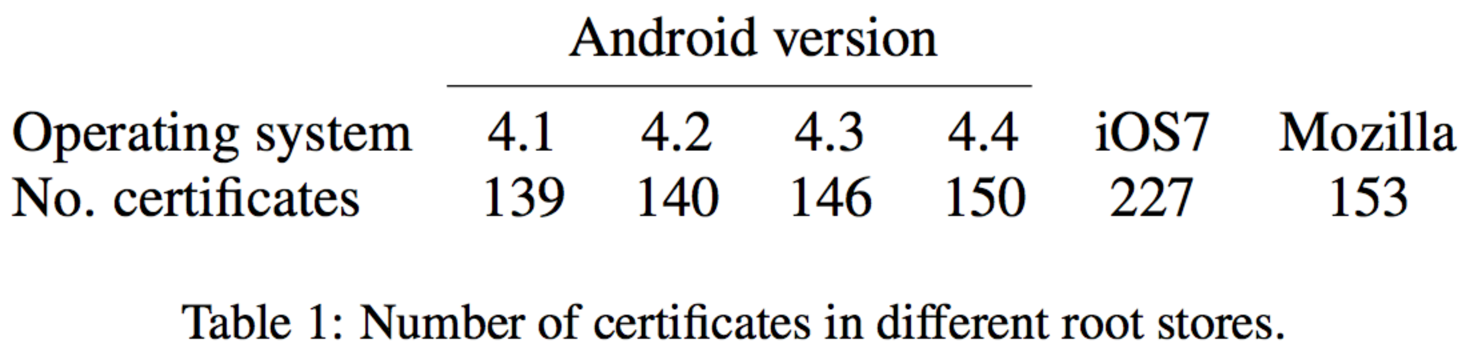
Certificate Authority System

Certificate Authority System

- Trusted root authorities are bundled with operating system or browser

- These authorities can provide certificates with signing authority to additional authorities

- Browser (or other software) checks for a valid certificate chain, rooted in the trusted root store

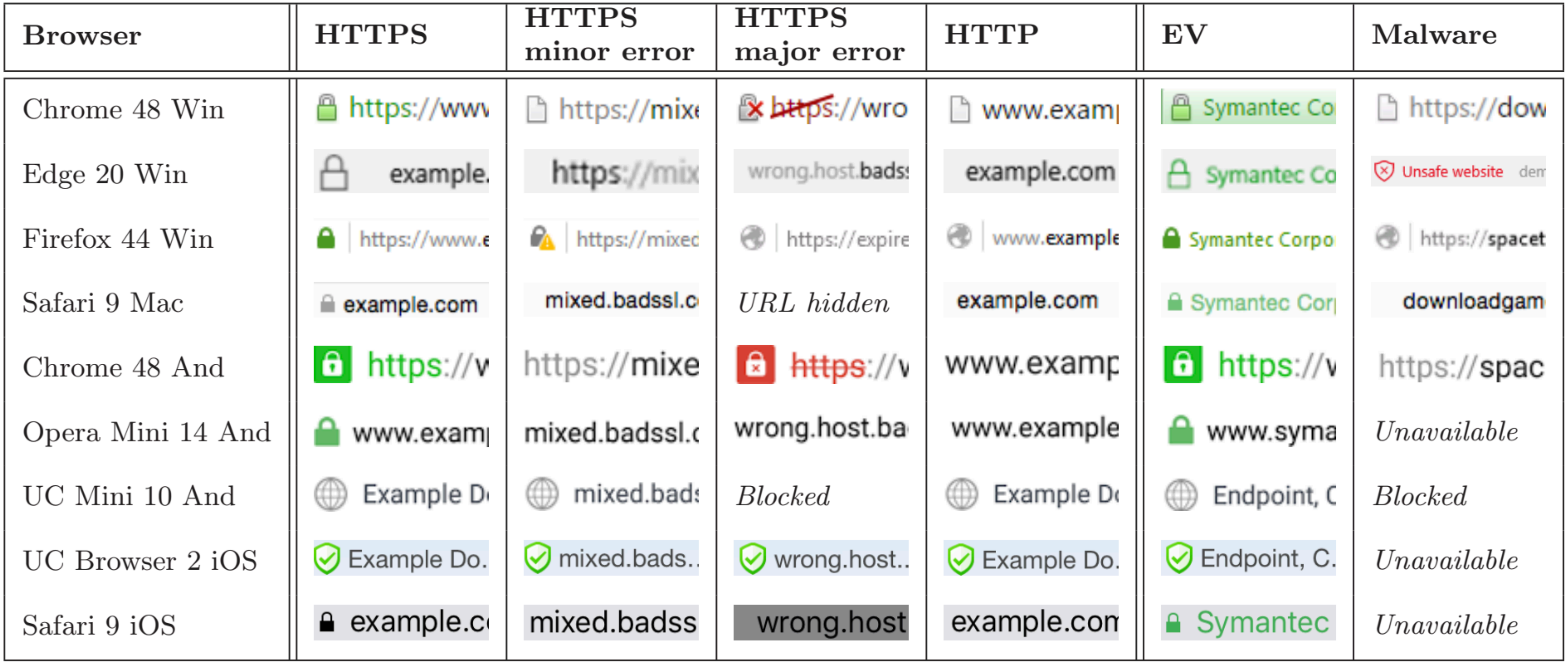


	Android version					
Operating system	4.1	4.2	4.3	4.4	iOS7	Mozilla
No. certificates	139	140	146	150	227	153

Table 1: Number of certificates in different root stores.

Durumeric, Zakir, James Kasten, Michael Bailey, and J. Alex Halderman. "Analysis of the HTTPS certificate ecosystem." In *Proceedings of the 2013 conference on Internet measurement conference*, pp. 291-304. ACM, 2013.

Browser Security Indicators (2016)



Browser	HTTPS	HTTPS minor error	HTTPS major error	HTTP	EV	Malware
Chrome 48 Win	https://www	https://mixe	~~https~~://wro	www.examp	Symantec Co	https://dow
Edge 20 Win	example.	https://mix	wrong.host.badss	example.com	Symantec Co	Unsafe website der
Firefox 44 Win	https://www.e	https://mixed	https://expire	www.example	Symantec Corpo	https://spacet
Safari 9 Mac	example.com	mixed.badssl.c	*URL hidden*	example.com	Symantec Corp	downloadgam
Chrome 48 And	https://w	https://mixe	~~https~~://w	www.examp	https://w	https://spac
Opera Mini 14 And	www.examp	mixed.badssl.c	wrong.host.ba	www.example	www.syma	*Unavailable*
UC Mini 10 And	Example D	mixed.bads	*Blocked*	Example D	Endpoint, C	*Blocked*
UC Browser 2 iOS	Example Do.	mixed.bads..	wrong.host..	Example Do.	Endpoint, C.	*Unavailable*
Safari 9 iOS	example.c	mixed.badss	wrong.host	example.com	Symantec	*Unavailable*

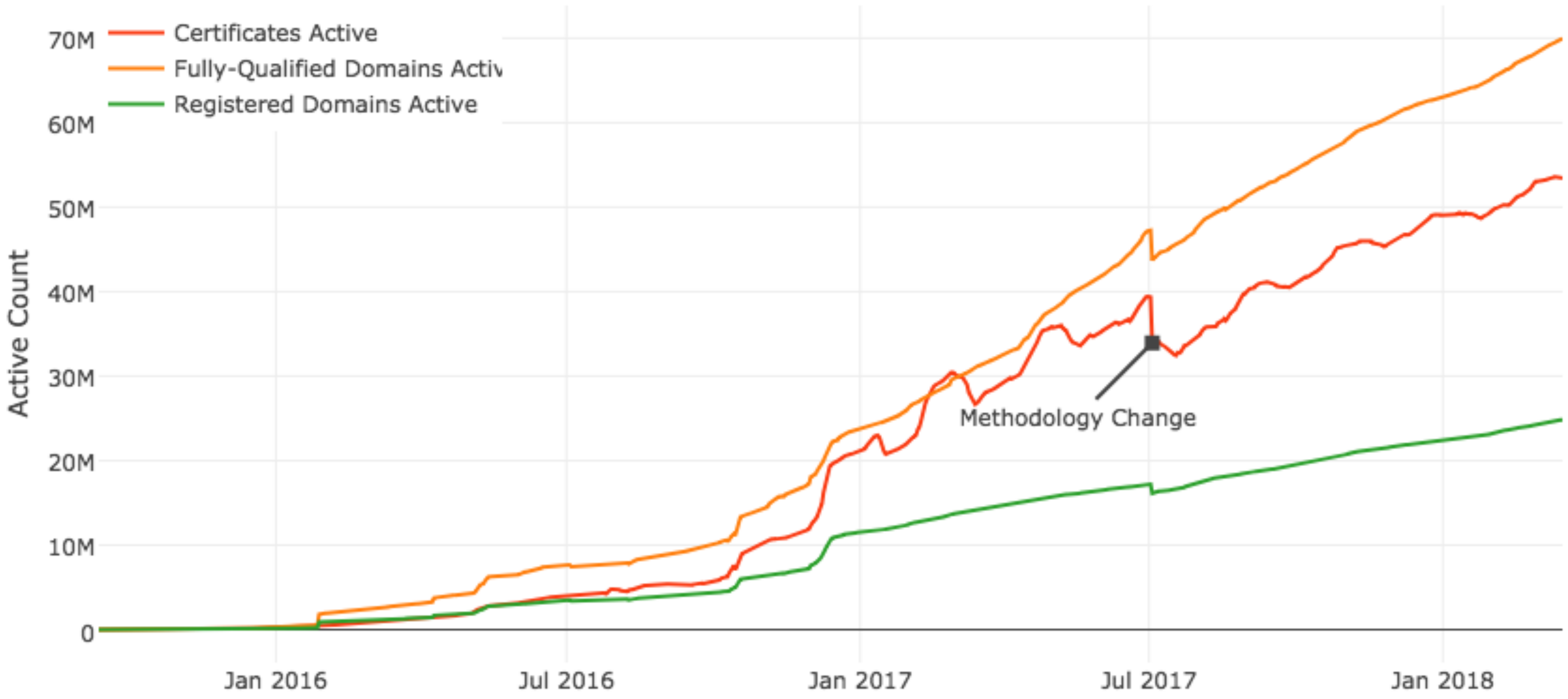
Figure 2: Security indicators for major browsers on Windows (Win), Mac, Android (And), and iOS. For categories that trigger warnings (e.g., malware), we include the security indicator state during the warning.

Felt, Adrienne Porter, Robert W. Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Embre Acer, Elisabeth Morant, and Sunny Consolvo. "Rethinking Connection Security Indicators." In *SOUPS*, pp. 1-14. 2016.

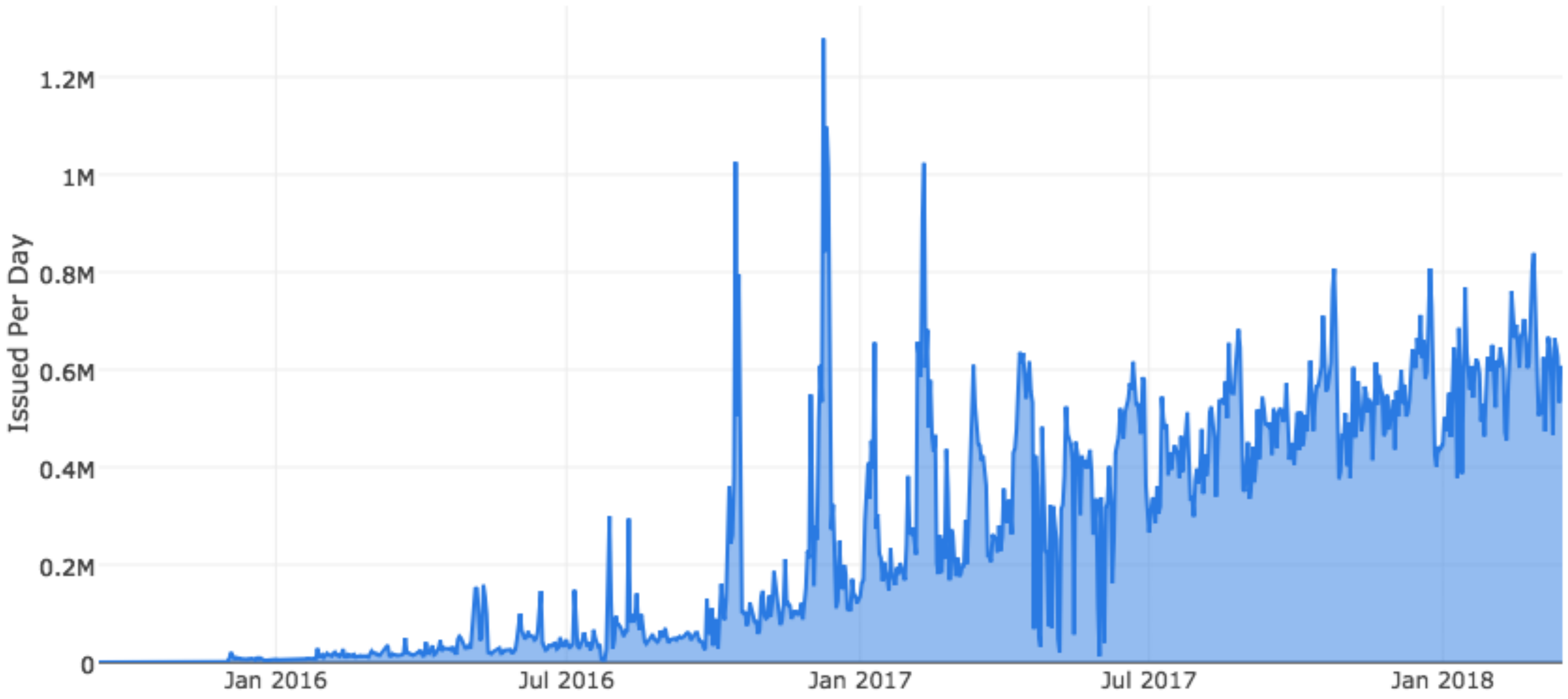
Certificate Authorities

- How do you prove you own a domain (and get a certificate issued)?
  - Domain Validation — prove you own a domain by putting a special record into DNS or posting a special file on your website — partially automated
  - Extended Validation — go through additional procedures to validate you own a company — considered worthless by some security experts, e.g. https://scotthelme.co.uk/are-ev-certificates-worth-the-paper-theyre-written-on/
- Let’s Encrypt https://letsencrypt.org/
  - Completely automated certificate issuance
  - Now the largest CA, by some measures

Let’s Encrypt Growth



Let’s Encrypt Certificates Issued Per Day



Application Errors

We demonstrate that SSL certificate validation is completely broken in many security-critical applications and libraries. Vulnerable software includes Amazon’s EC2 Java library and all cloud clients based on it; Amazon’s and PayPal’s merchant SDKs responsible for transmitting payment details from e-commerce sites to payment gateways; integrated shopping carts such as osCommerce, ZenCart, Ubercart, and PrestaShop; AdMob code used by mobile websites; Chase mobile banking and several other Android apps and libraries; Java Web-services middleware—including Apache Axis, Axis 2, Codehaus XFire, and Pusher library for Android—and all applications employing this middleware. Any SSL connection from any of these programs is insecure against a man-in-the-middle attack.

Georgiev, Martin, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. "The most dangerous code in the world: validating SSL certificates in non-browser software." In *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 38-49. ACM, 2012.

Interception Middleware and Malware

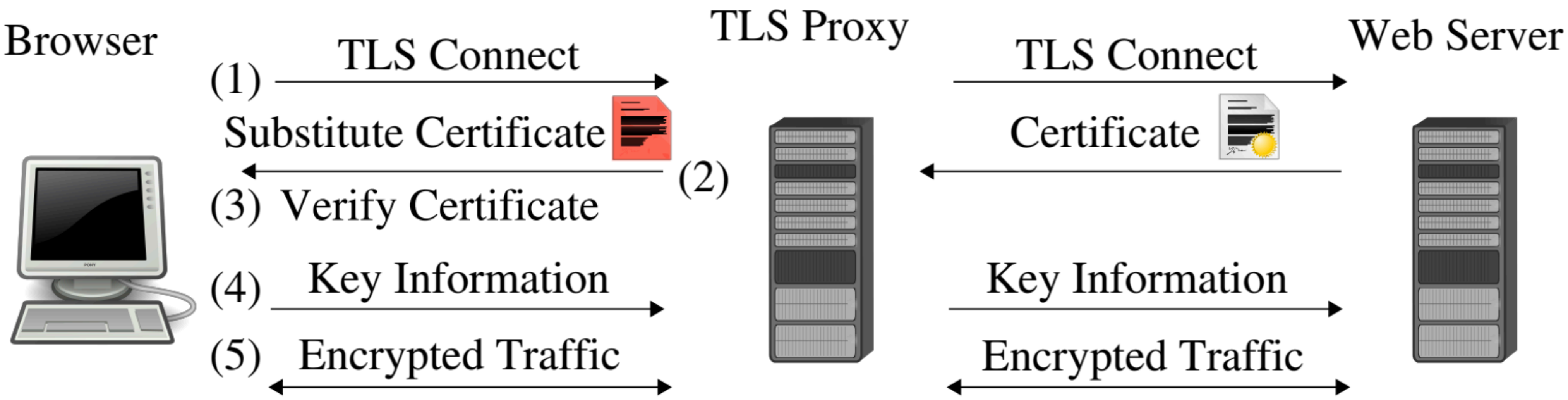


Figure 1: "Secure" session establishment with involving a TLS proxy

O'Neill, Mark, Scott Ruoti, Kent Seamons, and Daniel Zappala. "TLS proxies: Friend or foe?." In *Proceedings of the 2016 Internet Measurement Conference*, pp. 551-557. ACM, 2016.

Compromised CAs, Malpractice, Etc.

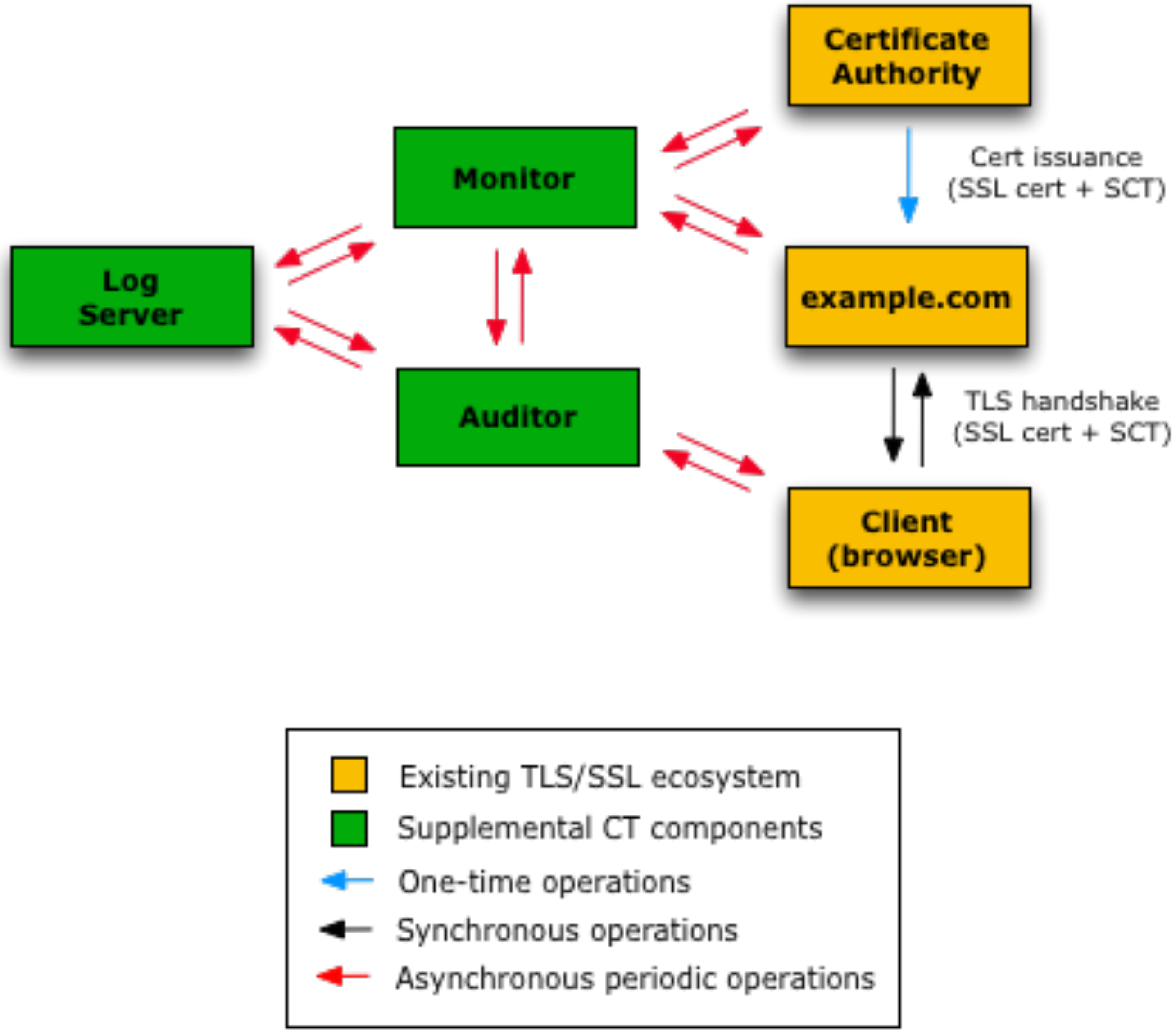
- The system is only as strong as the weakest link
- Attack
    - 2001: Verisign issued two fraudulent Microsoft certificates
        - No revocation infrastructure, so Microsoft patch had to explicitly blacklist these two certificates in the verification code
    - 2011: Dutch CA DigiNotor was compromised
        - Led to man-in-the-middle attack on 300,000 Iranian citizens, including Gmail accounts

Compromised CAs, Malpractice, Etc.

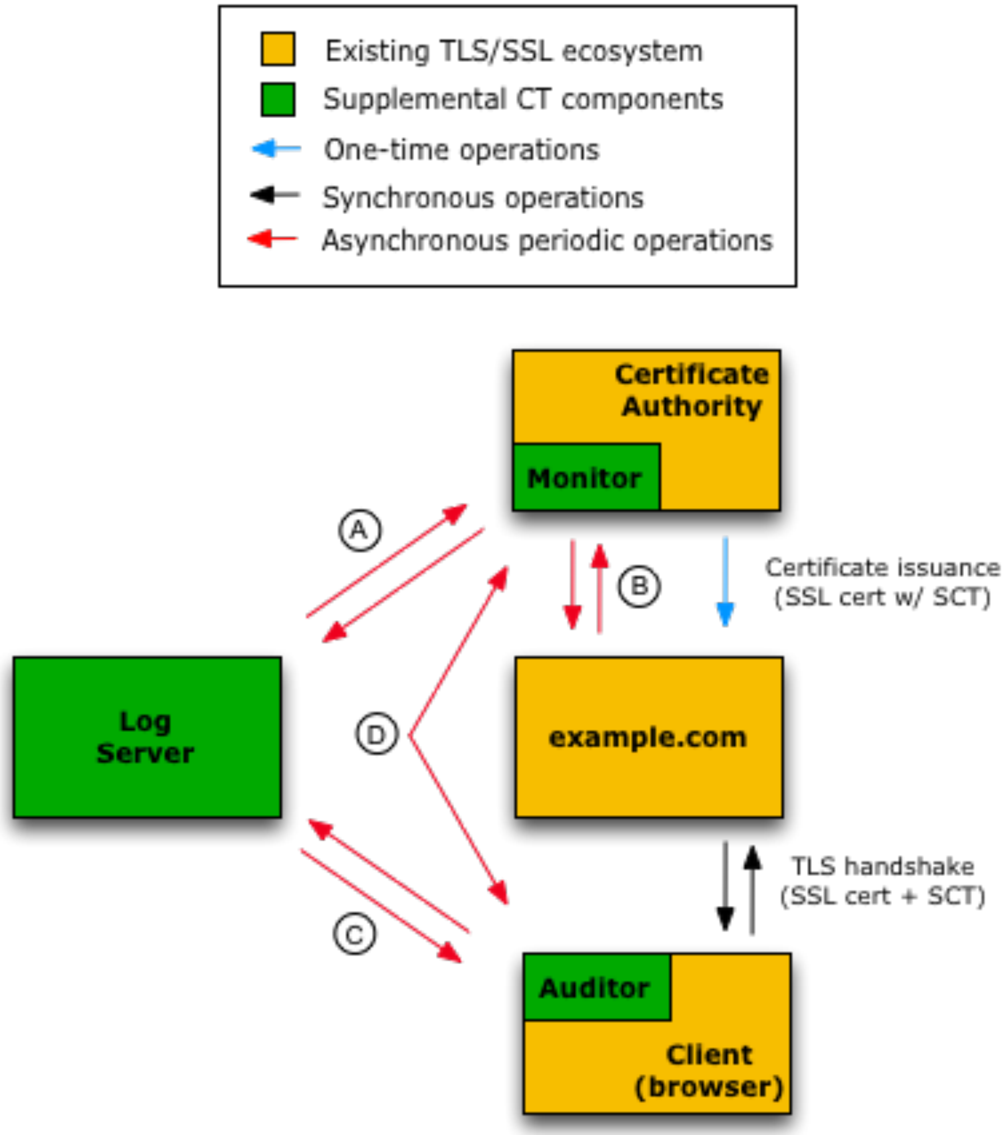
- Malpractice
  - Best practices such as the **principle of least privilege** and **defense in depth** are not being followed
  - Turktrust accidentally issued a signing certificate to one of its customers that ultimately signed a valid certificate for *.google.com. If name or path constraints had been applied to Turktrust’s CA intermediate certificate, the incident could have been avoided or, at the very least, reduced in scope.
    - Durumeric, Zakir, James Kasten, Michael Bailey, and J. Alex Halderman. "Analysis of the HTTPS certificate ecosystem." In Proceedings of the 2013 conference on Internet measurement conference, pp. 291-304. ACM, 2013
- Government Ownership
  - *compelled certificate creation attack*, in which government agencies may compel a certificate authority to issue false SSL certificates that can be used by intelligence agencies to covertly intercept and hijack traffic
    - Soghoian, Christopher, and Sid Stamm. "Certified lies: Detecting and defeating government interception attacks against SSL." In Proceedings of ACM Symposium on Operating Systems Principles, pp. 1-18. 2010.

Certificate Transparency

detects certificate mis-issuance


Certificate Authority
Monitor
Log Server
Auditor
example.com
Client (browser)
Cert issuance (SSL cert + SCT)
TLS handshake (SSL cert + SCT)
🟧 Existing TLS/SSL ecosystem
🟩 Supplemental CT components
One-time operations
Synchronous operations
Asynchronous periodic operations

https://www.certificate-transparency.org/how-ct-works


🟧 Existing TLS/SSL ecosystem
🟩 Supplemental CT components
One-time operations
Synchronous operations
Asynchronous periodic operations
Certificate Authority
Monitor
Log Server
example.com
Auditor
Client (browser)
A
B
C
D
Certificate issuance (SSL cert w/ SCT)
TLS handshake (SSL cert + SCT)

Ⓐ Monitors watch logs for suspicious certs and verify that all logged certs are visible.

Ⓑ Certificate owners query monitors to verify that nobody has logged illegitimate certs for their domain.

Ⓒ Auditors verify that logs are behaving properly; they can also verify that a particular cert has been logged.

Ⓓ Monitors and auditors exchange information about logs to help detect forked or branched logs.

Figure 3

Revisit: PKI Reality

- Names – how to identify subjects?
    - Domain Names
- Authorities – who can sign certificates?
    - CAs, as long as the chain resolves to a root certificate and it is in the CT logs
- Trust – who do we trust as authorities?
    - Whoever the browser and OS vendors tell us to trust
- Revocation – hardest PKI problem to solve
    - Uh…

Revocation

- Certificate Revocation List (CRL)
  - large (76 MB recently) — hard for mobile to download
- Online Certificate Status Protocol (OCSP)
  - check status of individual cert
  - adds latency, violates privacy
- OCSP Stapling
  - send revocation status with original certificate during TLS handshake — but can simply be dropped by attacker and client doesn't know it should be there
- OCSP Must Staple
  - certificate includes a field indicating OCSP stapling must be there — but if a website administrator forgets to include it, then their site is offline — also a DoS attack against OCSP responders can block access to web sites
- CRLSet (Google) and OneCRL (Mozilla)
  - small list of revoked certificates where risk of compromise is suspected
  - does not cover entire certificate space
- Most non-mobile browsers have disabled revocation with CRLs in favor of CRLSet and OneCRL, mobile browsers do not check revocation status