UiO : Department of Informatics
University of Oslo

COINS Winter School 2018, Finse
Nils Gruschka, UiO

PKI and Certificate Security

Introduction

- Nils Gruschka
  - University Kiel (Diploma in Computer Science)
  - T-Systems, Hamburg
  - University Kiel (PhD in Computer Science)
  - NEC Laboratories Europe, Bonn + Heidelberg
  - University of Applied Science Kiel
  - University of Oslo
- Contact:
  - Nils.Gruschka@ifi.uio.no
- Areas of interest:
  - Security: Network, Web, Cloud Computing, Industrial Networks
  - Applied Cryptography

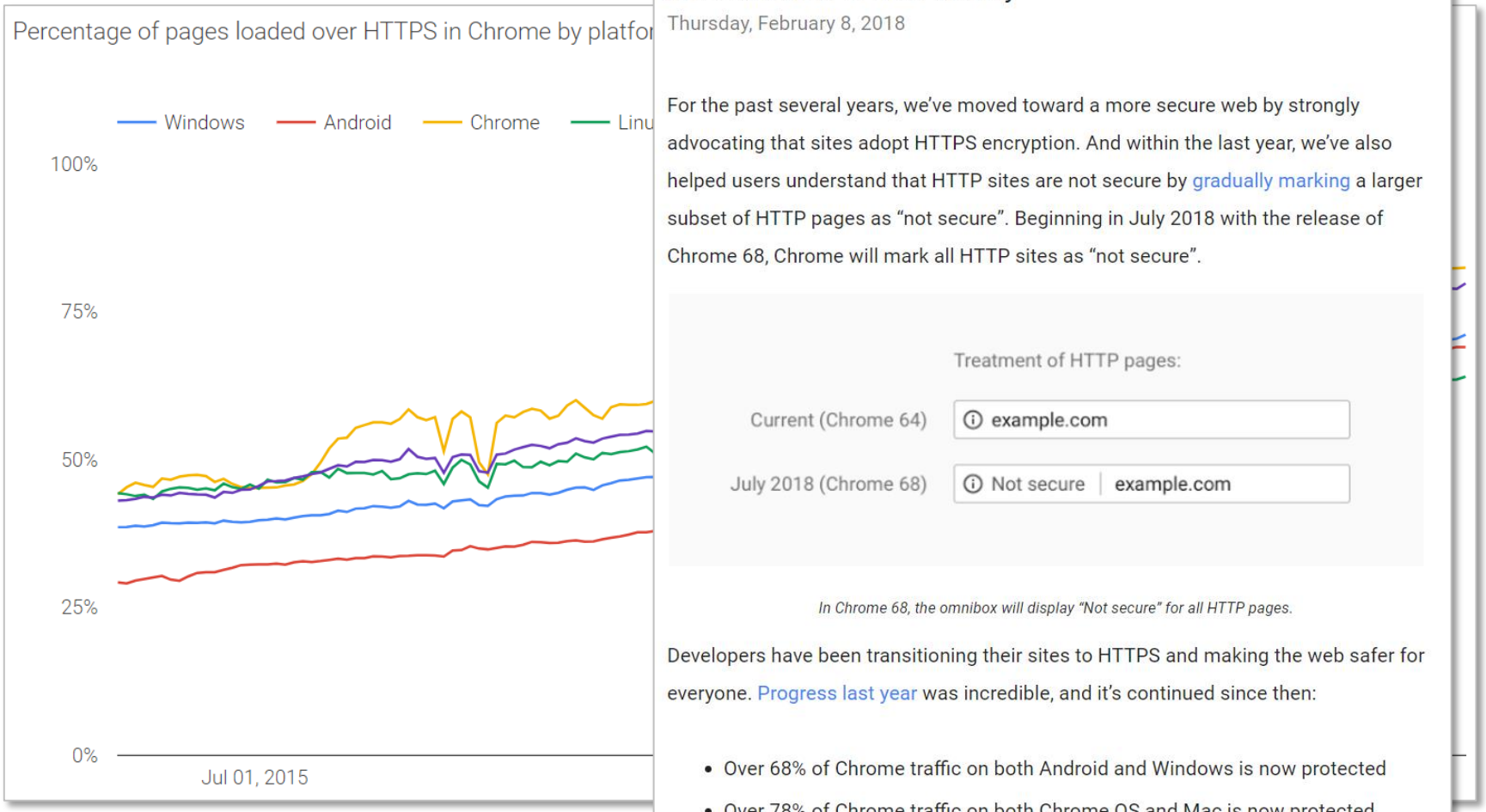
Outline

- Motivation
- Certificates
- Public Key Infrastructure
- Threats to certificates / PKI
- Protecting certificates / PKI

Certificates

Motivation

Motivation: TLS usage



A secure web is here to stay

Thursday, February 8, 2018

For the past several years, we've moved toward a more secure web by strongly advocating that sites adopt HTTPS encryption. And within the last year, we've also helped users understand that HTTP sites are not secure by gradually marking a larger subset of HTTP pages as "not secure". Beginning in July 2018 with the release of Chrome 68, Chrome will mark all HTTP sites as "not secure".

Treatment of HTTP pages:

Current (Chrome 64) example.com

July 2018 (Chrome 68) Not secure | example.com

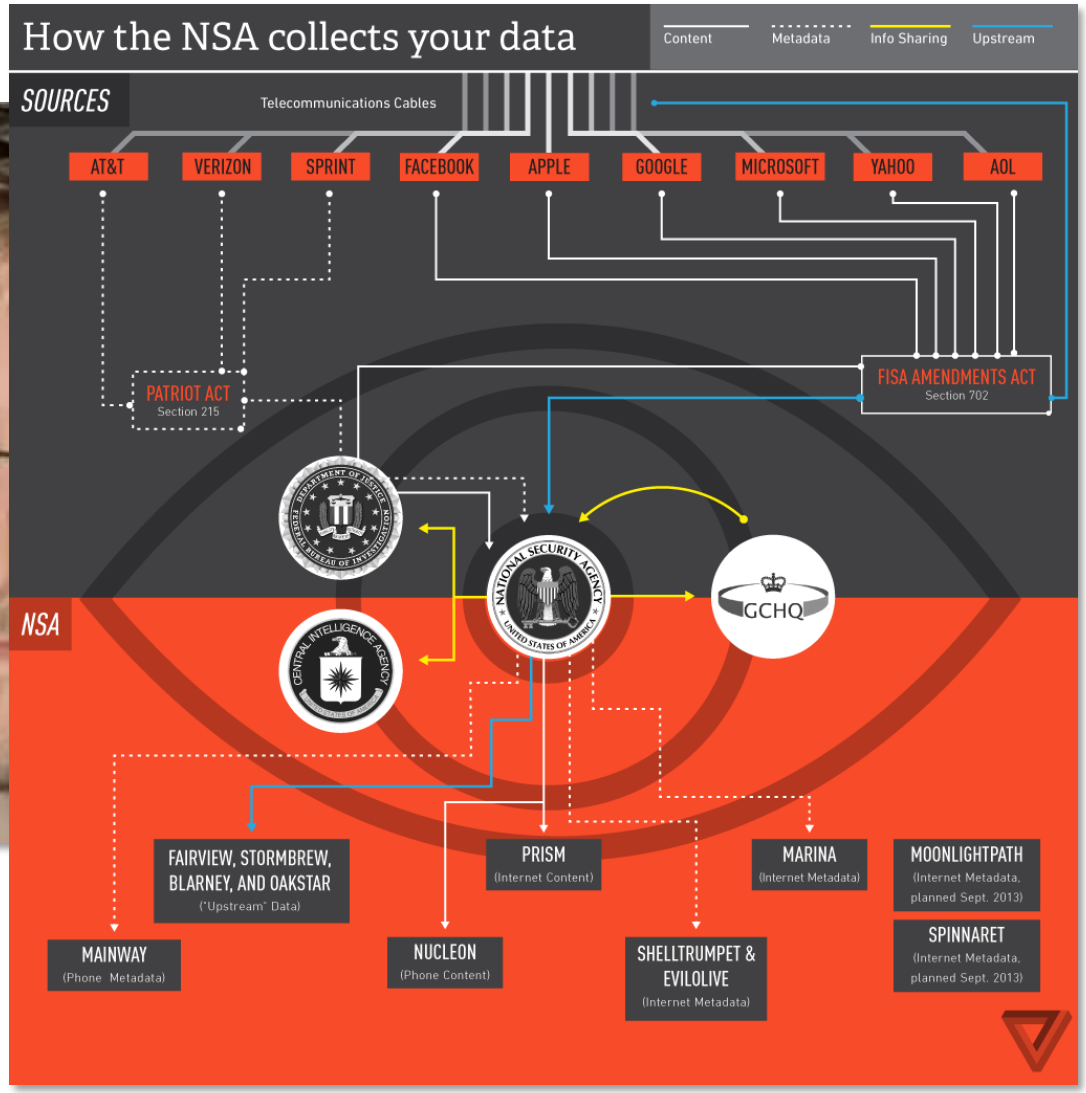
In Chrome 68, the omnibox will display "Not secure" for all HTTP pages.

Developers have been transitioning their sites to HTTPS and making the web safer for everyone. Progress last year was incredible, and it's continued since then:

- Over 68% of Chrome traffic on both Android and Windows is now protected
- Over 78% of Chrome traffic on both Chrome OS and Mac is now protected
- 81 of the top 100 sites on the web use HTTPS by default

Source: https://transparencyreport.google.com/https/overview?hl=en
https://security.googleblog.com/2018/02/a-secure-web-is-here-to-stay.html

Why do we need a “more secure Web”?



Source: https://www.theverge.com/2013/7/17/4517480/nsa-spying-prism-surveillance-cheat-sheet
https://www.theguardian.com/uk-news/2013/dec/20/gchq-targeted-aid-agencies-german-government-eu-commissioner

Why do we need a “more secure Web?

The Intercept_

Photo: Norwegian Intelligence Service

NORWAY USED NSA TECHNOLOGY FOR POTENTIALLY ILLEGAL SPYING

Henrik Moltke

Mar. 1

15

Source: https://theintercept.com/2018/03/01/norway-nsa-victory-garden-surveillance/

TLS and Assymetric Crytography

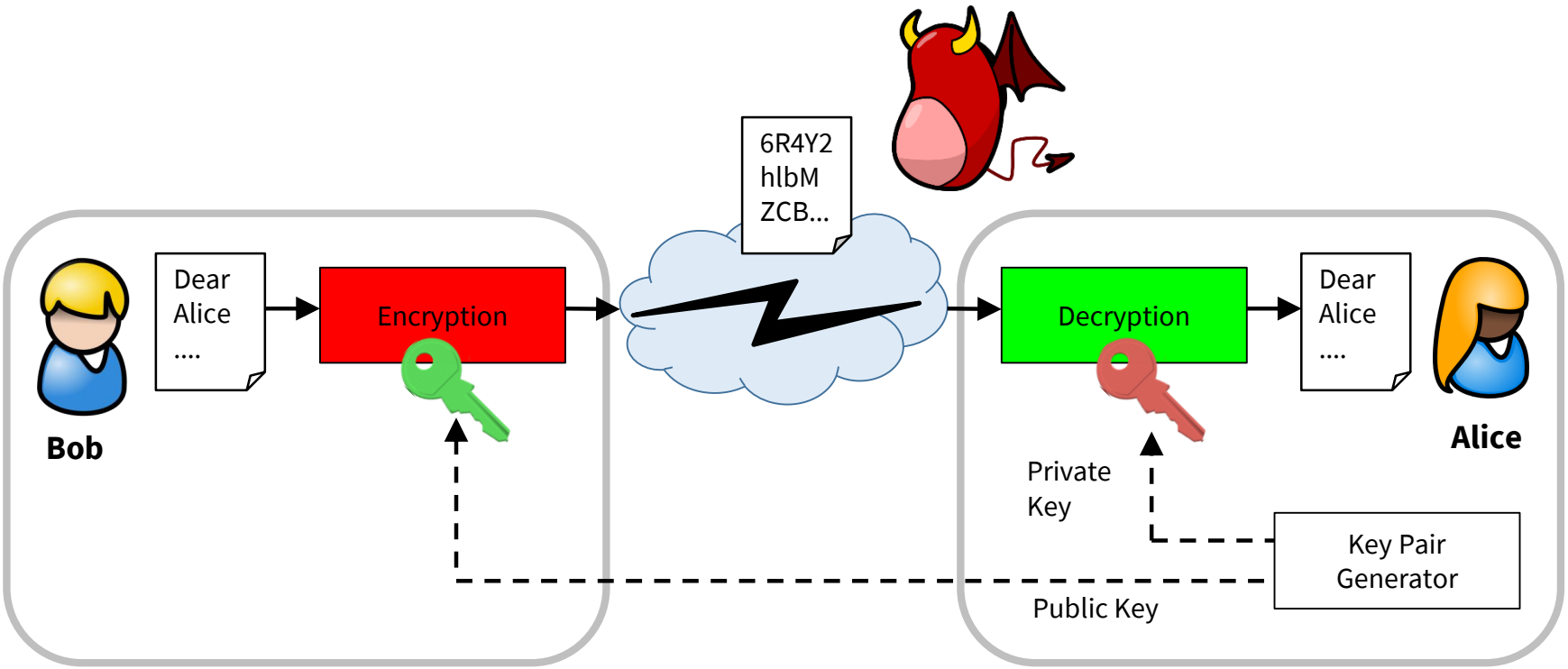
- From RFC 5246 (TLS 1.2):

  When RSA is used for server authentication and key exchange, a 48-byte pre_master_secret is generated by the client, encrypted under the server's public key, and sent to the server. The server uses its private key to decrypt the pre_master_secret.

  When Diffie-Hellman key exchange is used, the server […] use the server key exchange message to send a set of temporary Diffie-Hellman parameters signed with […] RSA […].
  […] the client can verify the […] signature to ensure that the parameters belong to the server.

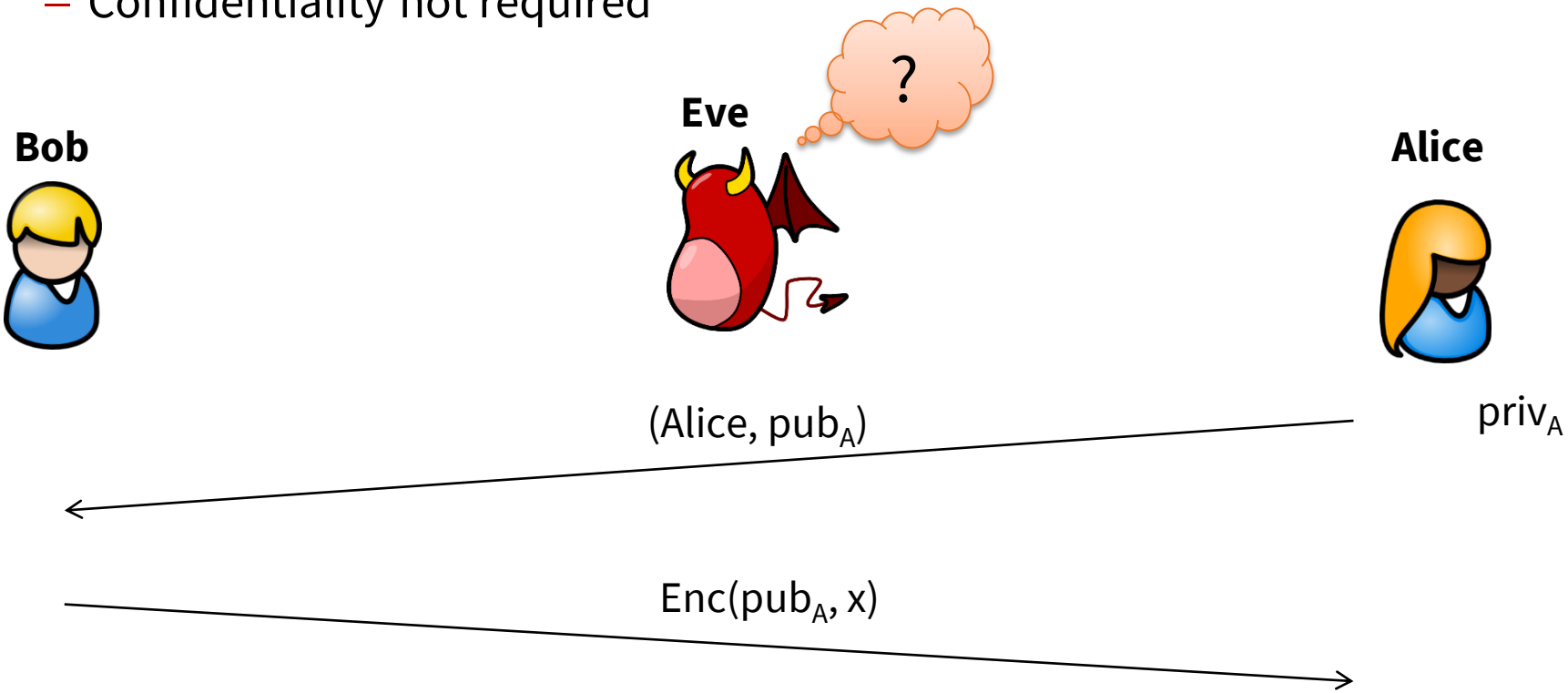
Recapitulation: Asymmetric Encryption

- Two distinct keys (private key and public key) are used for encryption and decryption respectively



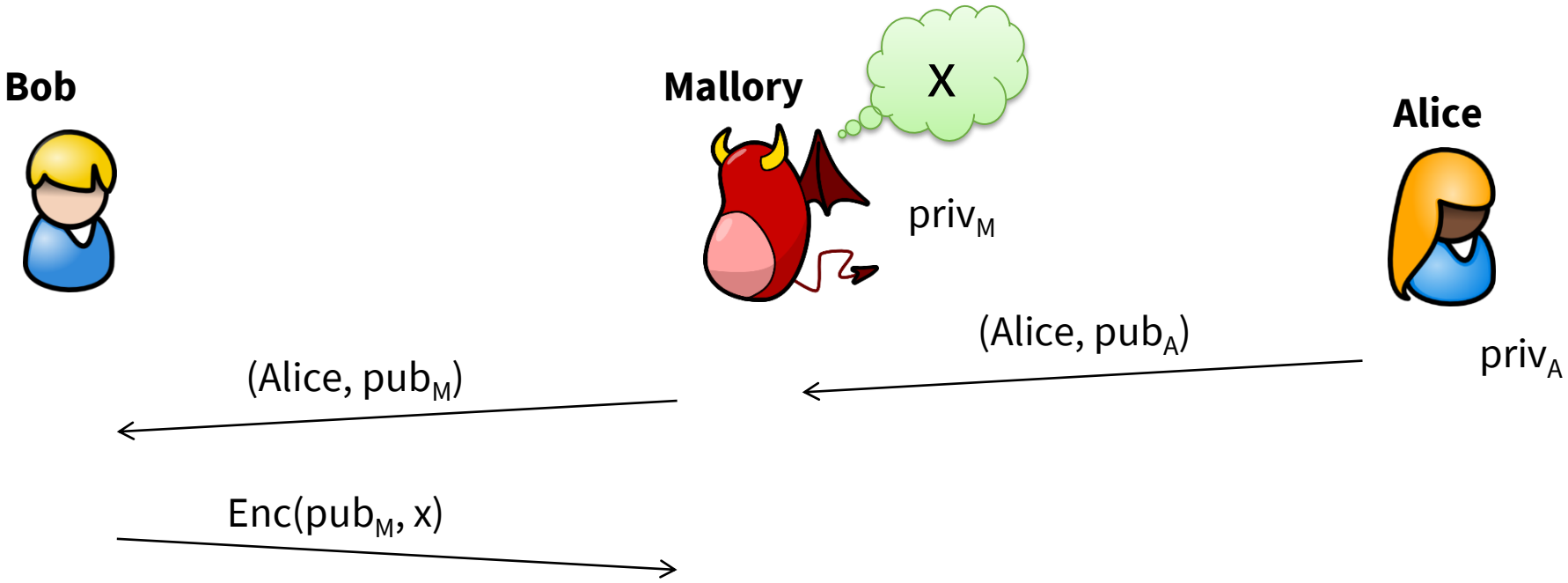
Attack on Key Exchange (Encryption)

- Exchange of public key:
  - Confidentiality not required

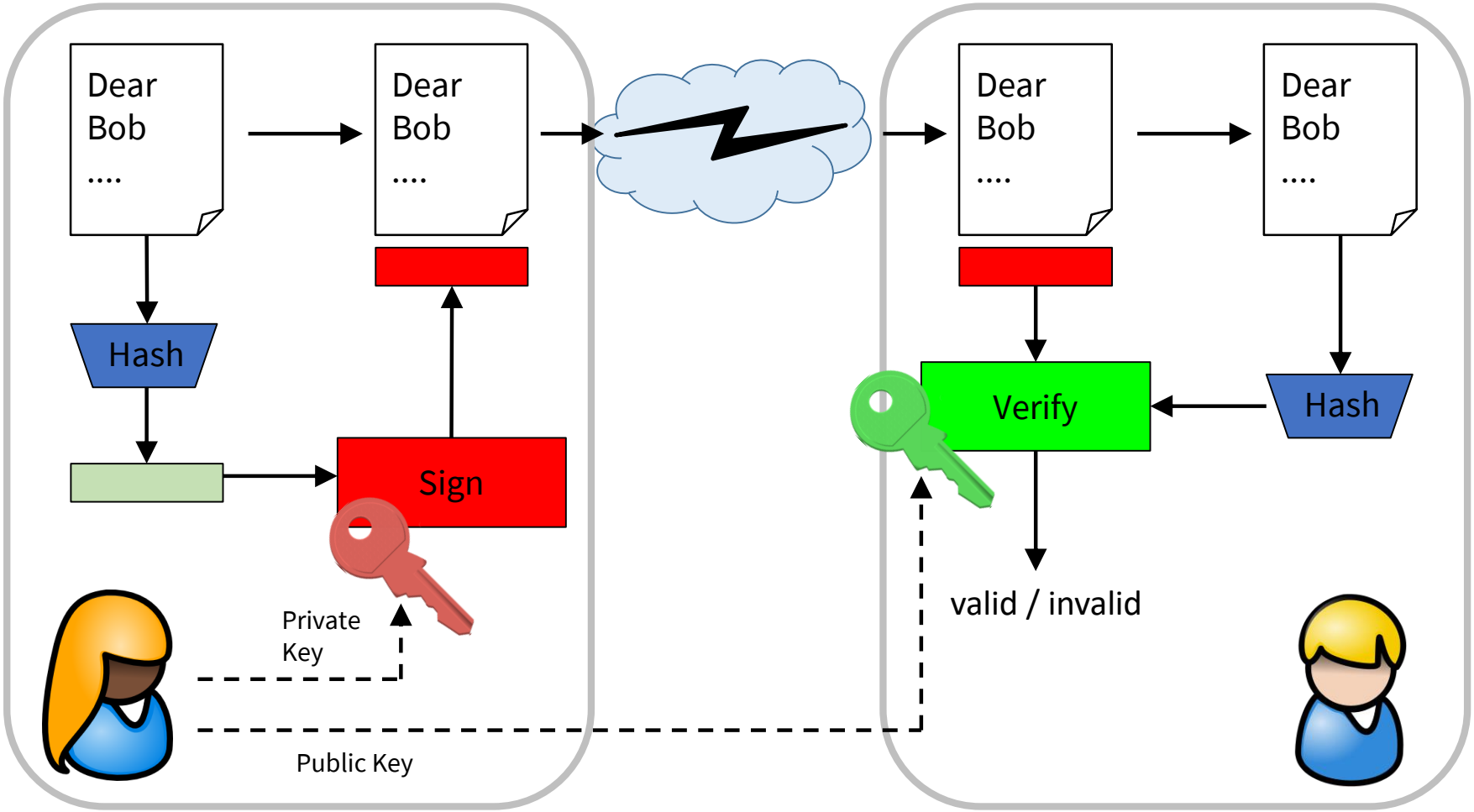


Attack on Key Exchange (Encryption)

- Exchange of public key:
  - Confidentiality not required
  - Integrity/authenticity highly required

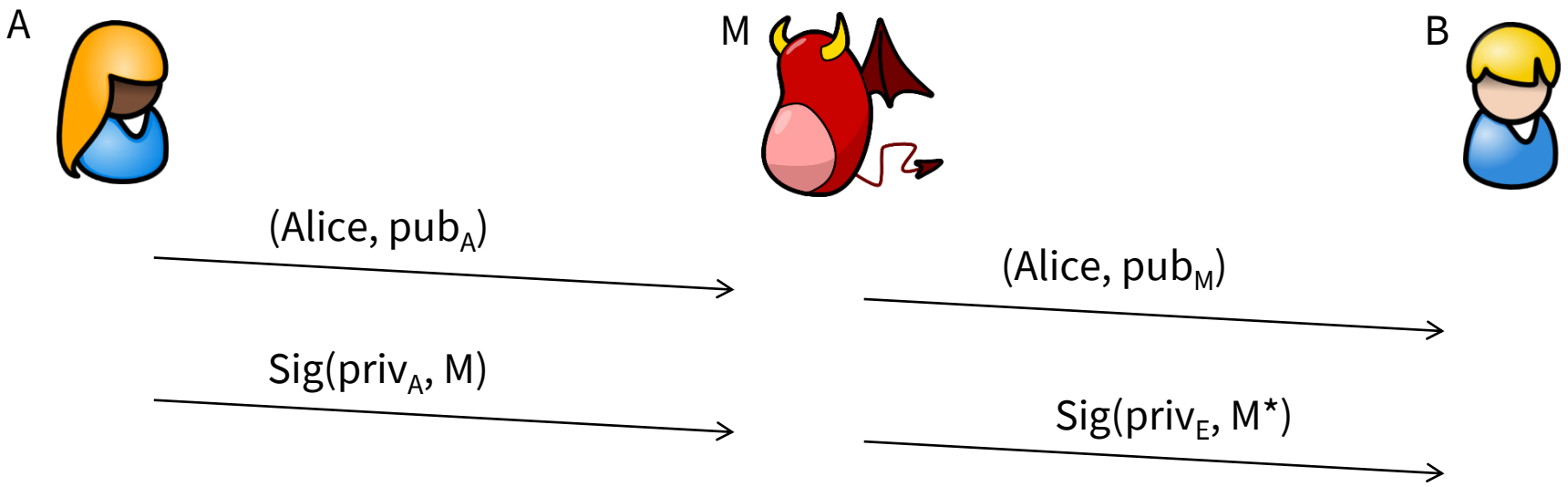


Recapitulation: Digital Signature



Attack on Key Exchange (Digital Signature)

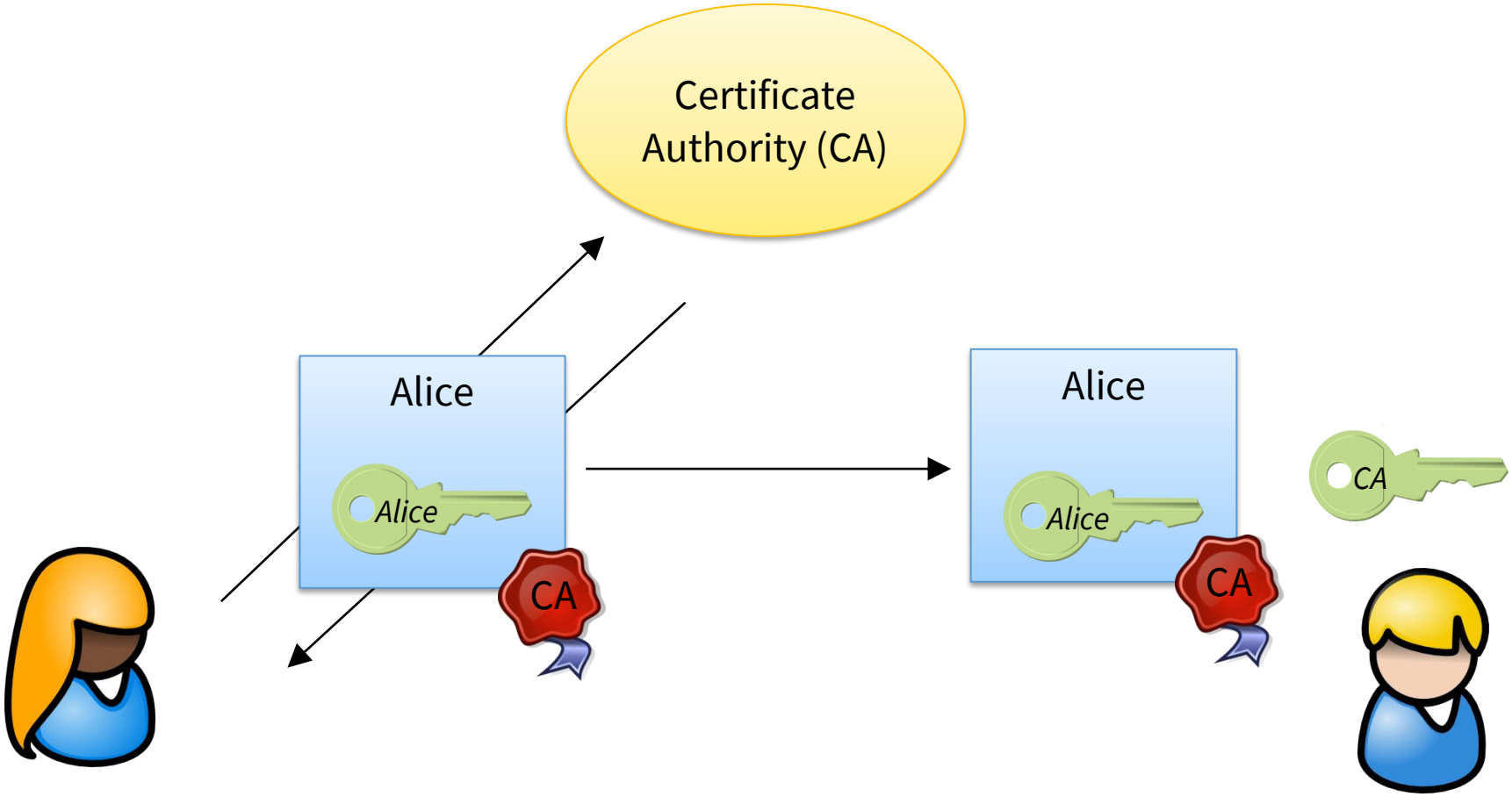
- Exchange of public key:
  - Confidentiality not required
  - Integrity/authenticity highly required



Certificates

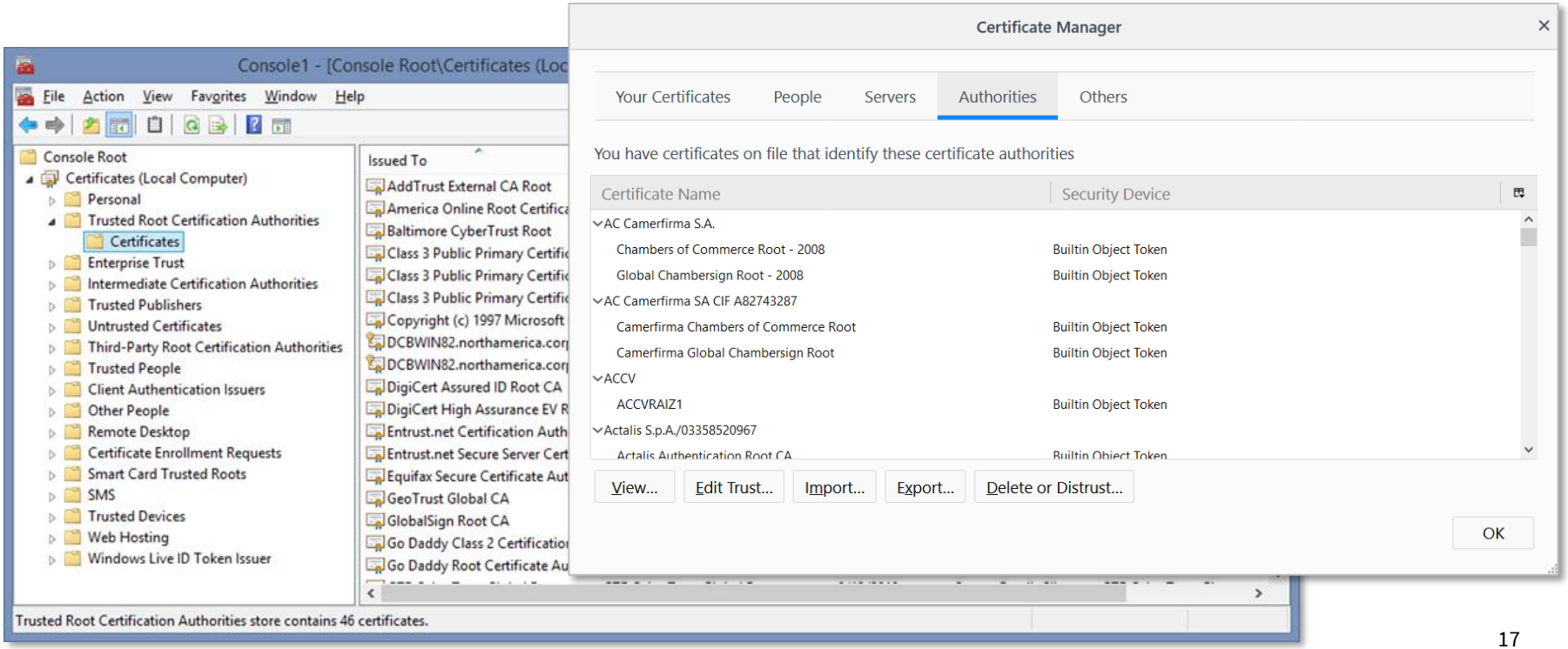
Introduction

Certificates

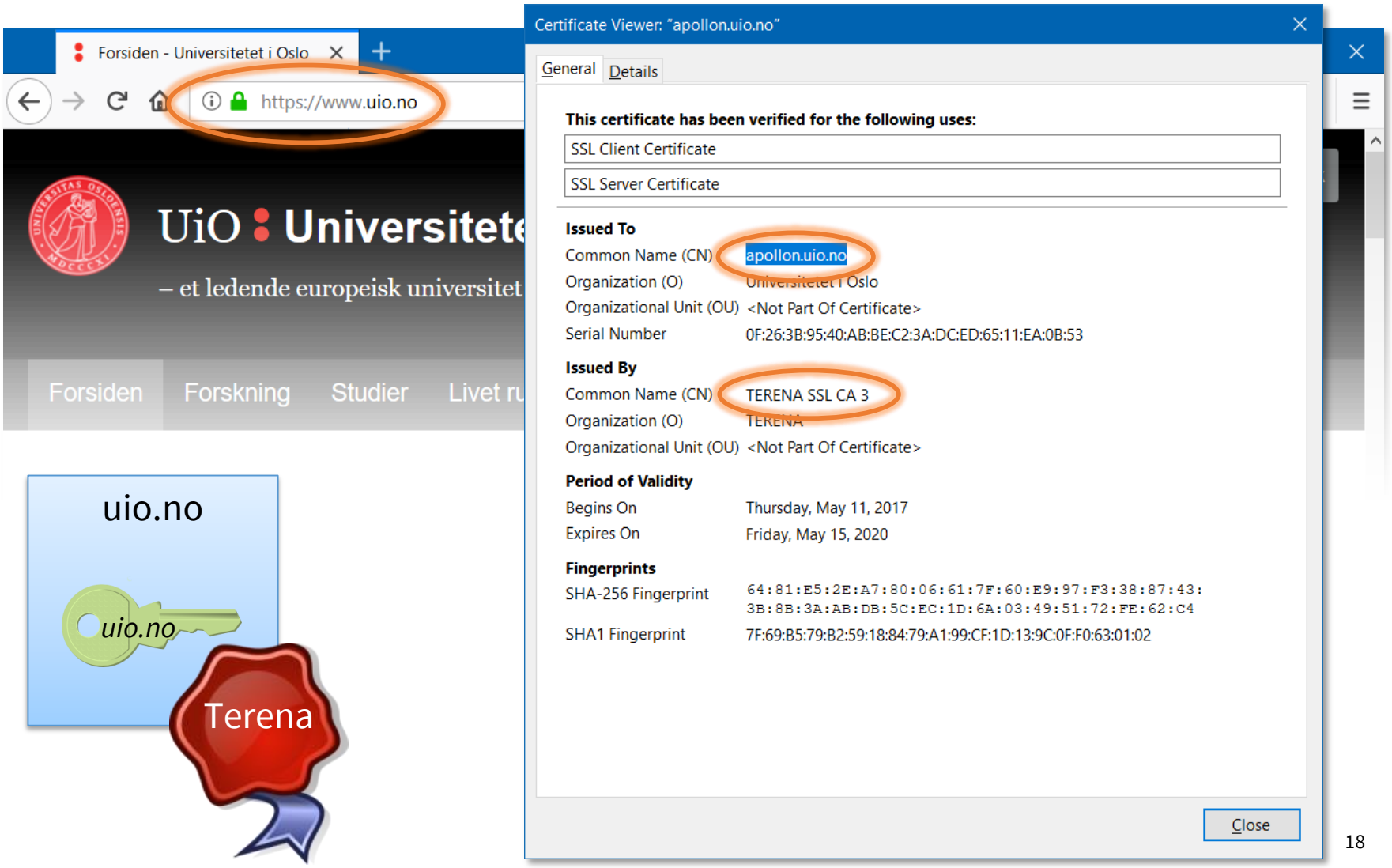


Certificate Trust

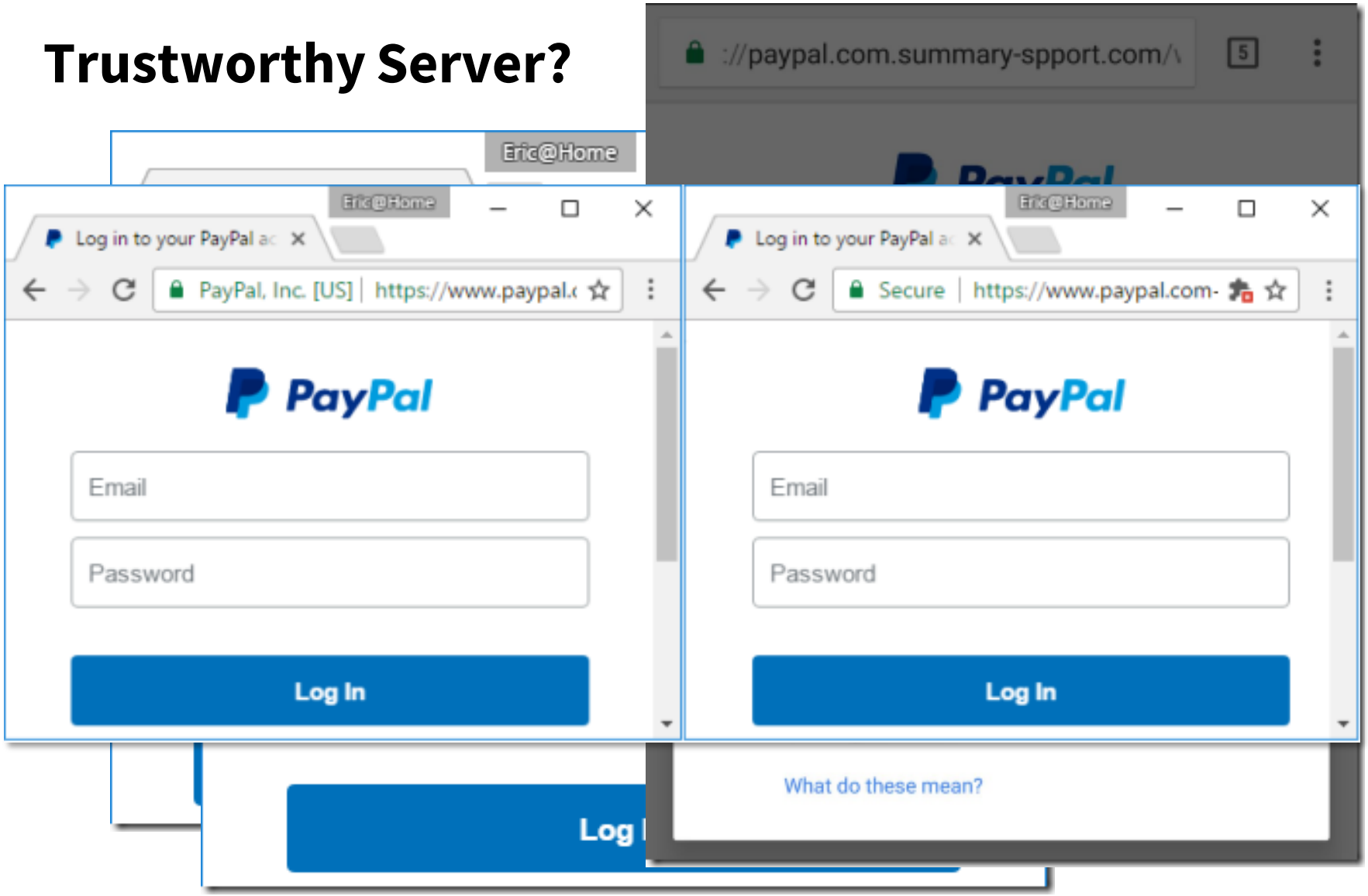
- How does Bob obtain the public key of the CA?
- A set of trusted CAs (root store) is included in the OS or the application (e.g. browser)



Certificates on the Web



Trustworthy Server?



Source: https://textslashplain.com/2017/01/16/certified-malice/

Certificates

Details

X.509 Certificate

- Most common standard for public key management and certificate formats
- X.509 certificates are defined using ASN.1 and can be encoded into different formats:
  - .cer, .crt
    - DER encoding (X.690)
  - .pem
    - Base64 encoded DER
    - enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"
  - .p7b
    - encoding according PKCS#7 standard
  - .p12
    - encoding according PKCS#12 standard
    - includes private key

ASN.1 and DER encoding

- ASN.1
  - Similar to Backus-Naur form
  - Example (grammar)

```
FooQuestion ::= SEQUENCE {
        trackingNumber INTEGER,
        question       IA5String
}
```

  - Example (message)

```
myQuestion FooQuestion ::= {
    trackingNumber     5,
    question           "Anybody there?"
}
```

Source: Wikipedia

ASN.1 and DER encoding

- DER encoding (for ASN.1 messages)
  - Uses a tag – length – value encoding
  - Example:

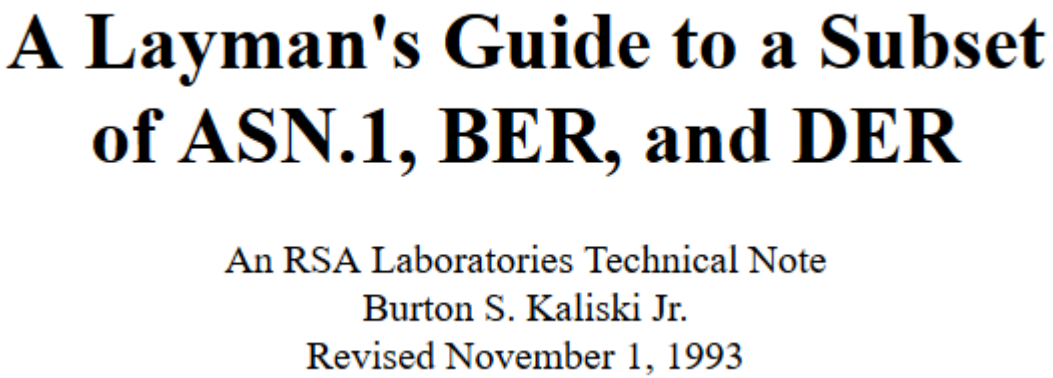
```
30 13 02 01 05 16 0e 41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f

30 – type tag indicating SEQUENCE
13 – length in octets of value that follows
  02 – type tag indicating INTEGER
  01 – length in octets of value that follows
    05 – value (5)
  16 – type tag indicating IA5String
     (IA5 means the full 7-bit ISO 646 set, including variants,
      but is generally US-ASCII)
  0e – length in octets of value that follows
    41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f – value ("Anybody there?")
```

Source: Wikipedia

ASN.1 and DER encoding

- More on this topic: http://luca.ntop.org/Teaching/Appunti/asn1.html


A Layman's Guide to a Subset of ASN.1, BER, and DER

An RSA Laboratories Technical Note
Burton S. Kaliski Jr.
Revised November 1, 1993

X.509 Certificate

- ASN.1 syntax definition (simplified):

```
Certificate ::= SIGNED{TBSCertificate}
TBSCertificate ::= SEQUENCE {
    version             Version DEFAULT v1,
    serialNumber        CertificateSerialNumber,
    signature           AlgorithmIdentifier{{SupportedAlgorithms}},
    issuer              Name,
    validity            Validity,
    subject             Name,
    ...
}
Version ::= INTEGER {v1(0), v2(1), v3(2)}
CertificateSerialNumber ::= INTEGER
Validity ::= SEQUENCE {
    notBefore  Time,
    notAfter Time,
    ...
}
```

Source: ITU-T X.509

X.509 Certificate

- Certificate
  - Version
  - Serial Number
  - Algorithm ID
  - Issuer
  - Validity
    - Not Before
    - Not After
  - Subject
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

Example

```
0000 00 05 24 30 82 05 20 30 82 04 08 a0 03 02 01 02 ..$0.. 0 ........
0010 02 10 05 42 23 6d a9 f7 83 38 46 e7 ce 5b f4 a4 ...B#m.. .8F..[..
0020 62 ee 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05 b.0...*. H.......
0030 00 30 64 31 0b 30 09 06 03 55 04 06 13 02 4e 4c .0d1.0.. .U....NL
0040 31 16 30 14 06 03 55 04 08 13 0d 4e 6f 6f 72 64 1.0...U. ...Noord
0050 2d 48 6f 6c 6c 61 6e 64 31 12 30 10 06 03 55 04 -Holland 1.0...U.
0060 07 13 09 41 6d 73 74 65 72 64 61 6d 31 0f 30 0d ...Amste rdam1.0.
0070 06 03 55 04 0a 13 06 54 45 52 45 4e 41 31 18 30 ..U....T ERENA1.0
0080 16 06 03 55 04 03 13 0f 54 45 52 45 4e 41 20 53 ...U.... TERENA S
0090 53 4c 20 43 41 20 33 30 1e 17 0d 31 35 30 36 32 SL CA 30 ...15062
00A0 36 30 30 30 30 30 30 5a 17 0d 31 38 30 37 30 34 6000000Z ..180704
00B0 31 32 30 30 30 30 5a 30 71 31 0b 30 09 06 03 55 120000Z0 q1.0...U
00C0 04 06 13 02 4e 4f 31 0d 30 0b 06 03 55 04 08 13 ....NO1. 0...U...
00D0 04 4f 73 6c 6f 31 12 30 10 06 03 55 04 07 13 09 .Oslo1.0 ...U....
00E0 30 33 31 33 20 4f 73 6c 6f 31 1d 30 1b 06 03 55 0313 Osl o1.0...U
00F0 04 0a 13 14 55 6e 69 76 65 72 73 69 74 65 74 65 ....Univ ersitete
0100 74 20 69 20 4f 73 6c 6f 31 0d 30 0b 06 03 55 04 t i Oslo 1.0...U.
0110 0b 13 04 55 53 49 54 31 11 30 0f 06 03 55 04 03 ...USIT1 .0...U..
0120 0c 08 2a 2e 75 69 6f 2e 6e 6f 30 82 01 22 30 0d ..*.uio. no0.."0.
0130 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03 82 01 ..*.H... ........
0140 0f 00 30 82 01 0a 02 82 01 01 00 bb c1 e2 ec d8 ..0..... ........
0150 bc d4 bf b4 f0 f5 00 8d cd 5c 86 b7 a3 17 42 5b ........ .\....B[
```

Example

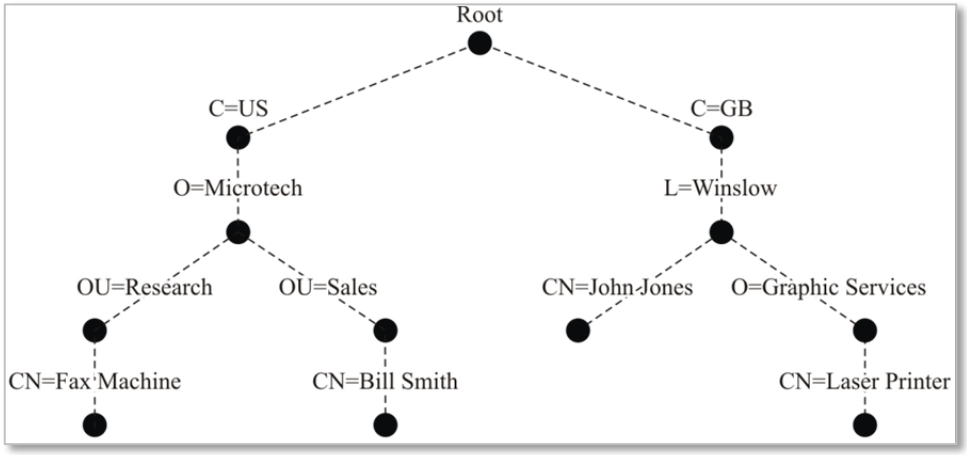
```
Version: 3 (0x2)
Serial Number: 0f:26:3b:95:40:ab:be:c2:3a:dc:ed:65:11:ea:0b:53
Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3
    Validity
        Not Before: May 11 00:00:00 2017 GMT
        Not After : May 15 12:00:00 2020 GMT
    Subject: C=NO, ST=Oslo, L=0313 Oslo, O=Universitetet i Oslo, CN=apollon.uio.no
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
            Modulus: 00:bc:58:...
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Authority Key Identifier:
            keyid:67:FD:88:20:14:27:98:C7:09:D2:25:19:BB:E9:51:11:63:75:50:62
        X509v3 Subject Key Identifier:
            7E:44:F3:39:4F:07:A7:0F:01:0B:52:4B:73:5F:72:00:C4:C7:E9:7A
        X509v3 Subject Alternative Name:
            DNS:apollon.uio.no, ..., DNS:www.mn.uio.no, ..., DNS:www.uio.no
        X509v3 Key Usage: critical
            Digital Signature, Key Encipherment
        X509v3 Extended Key Usage:
            TLS Web Server Authentication, TLS Web Client Authentication
        X509v3 CRL Distribution Points:
            Full Name: URI:http://crl3.digicert.com/TERENASSLCA3.crl
            Full Name: URI:http://crl4.digicert.com/TERENASSLCA3.crl
        X509v3 Certificate Policies:
            Policy: 2.16.840.1.114412.1.1
              CPS: https://www.digicert.com/CPS
            Policy: 2.23.140.1.2.2
        Authority Information Access:
            OCSP - URI:http://ocsp.digicert.com
            CA Issuers - URI:http://cacerts.digicert.com/TERENASSLCA3.crt
        X509v3 Basic Constraints: critical
            CA:FALSE
Signature Algorithm: sha256WithRSAEncryption
     81:fd:a9:...
```

Example – Details

- `Version: 3 (0x2)`
  - X.509 version (1, 2 or 3)
- `Serial Number: 0f:26:3b:95:40:ab:be …`
  - Unique identifier (within the issuing certificate authority)
- `Signature Algorithm: sha256WithRSAEncryption`
  - Algorithm for signing the certificate
  - Here: SHA-2 (256 bit) with RSA
  - Specified as international OID: 1.2.840.113549.1.1.11

OID:	{iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) sha256WithRSAEncryption(11)}
	1.2.840.113549.1.1.11
	/ISO/Member-Body/US/113549/1/1/11

Entity Identification



- X.500:
  - Standardized by ITU-T
  - Hierarchical data model ("directory information tree")
  - Directory access protocol
  - Entity can be uniquely addressed/identified by the distinguished name (DN), e.g.: C=US, O=Microtech, OU=Sales, CN=Bill Smith
- LDAP:
  - Standardized by IETF: RFC 4511
  - Similiar concept
  - Simplified data model and access protocol ("X.500 Lite")
  - Widespread usage, e.g. Microsoft Active Directory

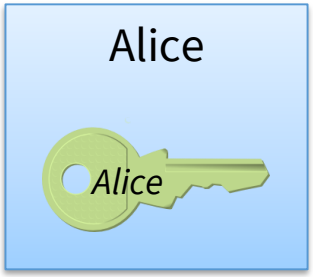
Image source: ITU-T X.500

Example – Details

- `Issuer: C=NL, ST=Noord-Holland, L=Amsterdam, O=TERENA, CN=TERENA SSL CA 3`
  - Issuing certificate authority (X.500 DN)
- `Validity`

  ```
  Not Before: May 11 00:00:00 2017 GMT
  Not After : May 15 12:00:00 2020 GMT
  ```

  - Limited lifetime
    - to reduce the risk of misuse
    - to incorporate the decrease of security of cryptographic algorithms
  - Also a indication when a certificate was created
    - some regulation only apply to certificates creates after a specific date
    - problem: CA can “cheat” and backdate a certificate



Example – Details

- `Subject: C=NO, ST=Oslo, L=0313 Oslo, O=Universitetet i Oslo, CN=apollon.uio.no`
  - Certificate subject (X.500 DN; usually only CN relevant)
  - For Web PKI: CN contains domain name
  - Domain name might contain a "wildcard", e.g. `*.example.com`
- `Subject Public Key Info:`

```
Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
        Modulus: 00:bc:58:...
        Exponent: 65537 (0x10001)
```

  - Public key of certificate subjects
  - Here: RSA (2048 bit), "standard" exponent $2^{16} + 1$

Example – Details

- `X509v3 extensions`
  - Further functionality added later to the standard
  - Instead of changing the data format
    → adding new/optional functions to an extensible data part
  - Not all processing entities understand all extensions
  - RFC 5280:

    *"A certificate-using system MUST reject the certificate if it encounters a **critical** extension that it does not recognize, or a critical extension that contains information that it cannot process.*
    *A **non-critical** extension MAY be ignored if it is not recognized, but MUST be processed if it is recognized."*

Example – Details

- `X509v3 Subject Alternative Name:`

  `DNS:apollon.uio.no, ..., DNS:www.uio.no`
  - Additional hostnames (in addition to CN) which the certificate covers
- `X509v3 CRL Distribution Points:`

  `URI:http://crl3.digicert.com/TERENASSLCA3.crl`

  `URI:http://crl4.digicert.com/TERENASSLCA3.crl`
- `Authority Information Access:`

  `OCSP - URI:http://ocsp.digicert.com`
  - Endpoints for information on revoked certificated → later

Example – Details

- `X509v3 Certificate Policies:`

  `Policy: 2.16.840.1.114412.1.1`

  `CPS: https://www.digicert.com/CPS`

  – Identifier for „Digicert OV“

  `Policy: 2.23.140.1.2.2`

  – Identifier for „Compliant with Baseline Requirements – Organization identity asserted”

- `X509v3 Basic Constraints: critical`

  `CA:FALSE`

  – Indicates a end-entity certificate

  – Only possibility to distinguish from CA certificates!

Example – Details

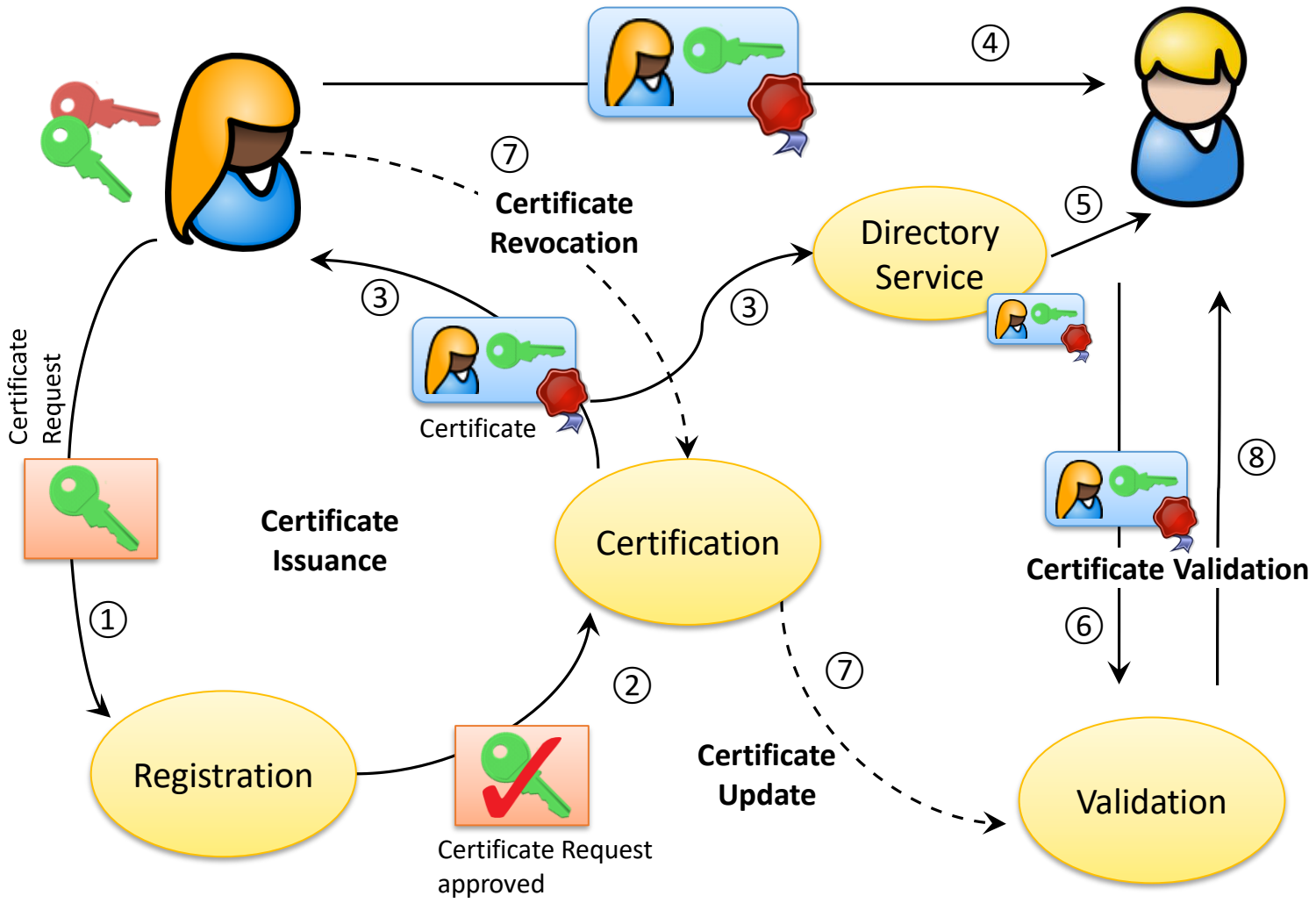
- `Signature Algorithm: sha256WithRSAEncryption`
  `81:fd:a9:...`
  - Digital signature on the certificate created by the CA
  - Here: RSA with SHA2 algorithm

"… some ISO standards have been written by little green monsters from outer space in order to confuse normal human beings and prepare them for the big invasion."
Markus Kuhn, 1995

Certificates

Public Key Infrastructure (PKI)

Components in a Public Key Infrastructure (PKI)

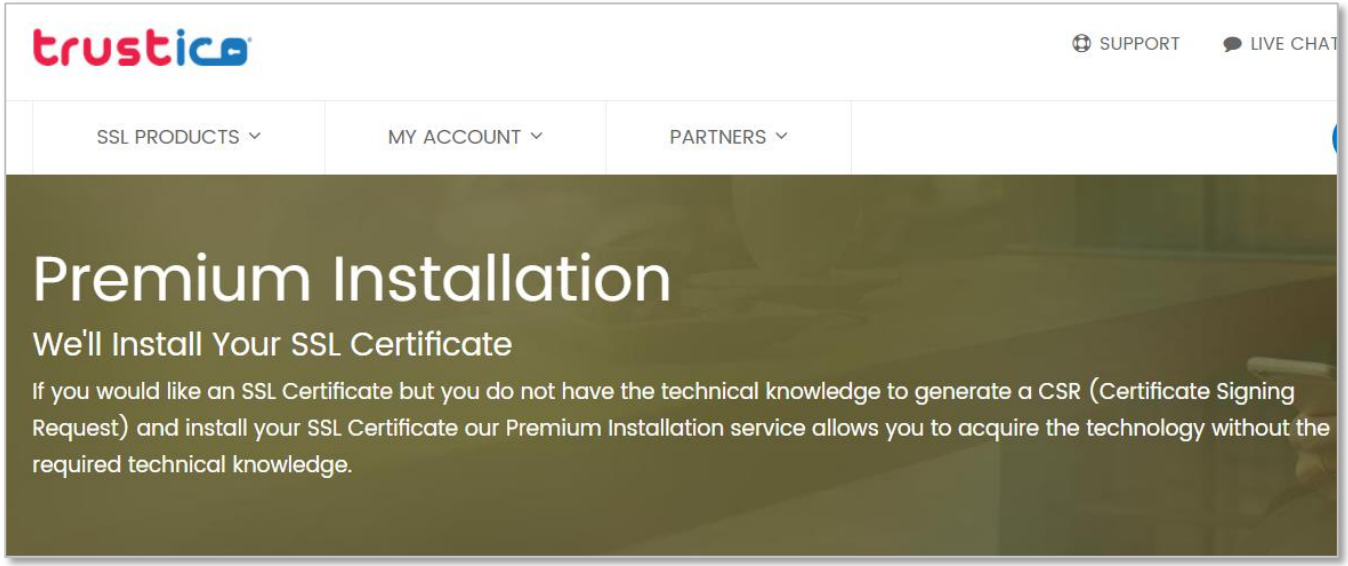


Certificate registration

- Requester must prove the identity to be certified
- For Web certificates:
  - prove ownership of the domain (DV, domain validation)
  - prove organization (OV, organization validation)
  - prove of legal organization registration (EV, extended validation)
- Common methods for domain validation:
  - Put a CA-provided challenge at a specific place on the Web server
  - Put a CA-provided challenge in a DNS record corresponding to the target domain.
  - Receive CA-provided challenge at a (hopefully) administrator-controlled email address corresponding to the domain and then respond to it on the CA's Web page.

Source: https://tools.ietf.org/html/draft-ietf-acme-acme-09

„Premium“ Key Management in a CA



The installation process may involve accessing your hosting account and we will require your hosting account information. If you do not want to reveal your hosting password you may wish to change it temporarily whilst we install your certificate.

If required, we can use your server to generate the CSR & Private Key, and reissue the SSL Certificate so it can be installed onto your server.

Source: https://www.trustico.com/premium/ssl-install-service.php

„Premium“ Key Management in a CA

23,000 Users Lose SSL Certificates in Trustico-DigiCert Spat

By Catalin Cimpanu — February 28, 2018 — 06:30 PM — 11

⇨ **6)** DigiCert claims that on February 27 it received an email from Trustico containing over 23,000 private keys for Trustico customers SSL certificates.

Over 23,000 users will have their SSL certificates revoked by tomorrow morning, March 1, in an incident between two companies —Trustico and DigiCert— that is likely to have a huge impact on the CA (Certificate Authority) industry as a whole in the coming months.

Source: https://www.bleepingcomputer.com/news/security/23-000-users-lose-ssl-certificates-in-trustico-digicert-spat/

CAB: CA/Browser forum

- Consortium of certificate related organizations:
  - Certificate authorities (e.g. DigiCert, Comodo, Let's Encrypt)
  - Browser vendors (e.g. Mozilla, Google)
  - Operating system vendors (e.g. Apple, Microsoft)
- Creates guidelines/best practices for issuances and management of certificates
  - Baseline requirements
  - Extended validation
  - Network and Certificate System Security Requirements

CA/BROWSER FORUM

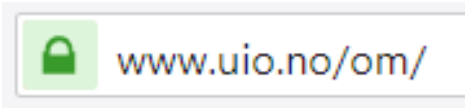
CAB: CA/Browser forum

- Baseline requirements
  - *"The Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates describe a subset of the requirements that a certification authority must meet in order to issue digital certificates for SSL/TLS servers to be publicly trusted by browsers."*
- Extended validation
  - Additionally:
  - *"Identify the legal entity that controls a web site by providing reasonable assurance to the user of an Internet browser that the web site the user is accessing is controlled by a specific legal entity identified in the EV Certificate by name, address of Place of Business, Jurisdiction of Incorporation or Registration and Registration Number or other disambiguating information."*

Source: https://cabforum.org/

Extended Validation Certificates (EV)

- EV certificates are indicated by the browser

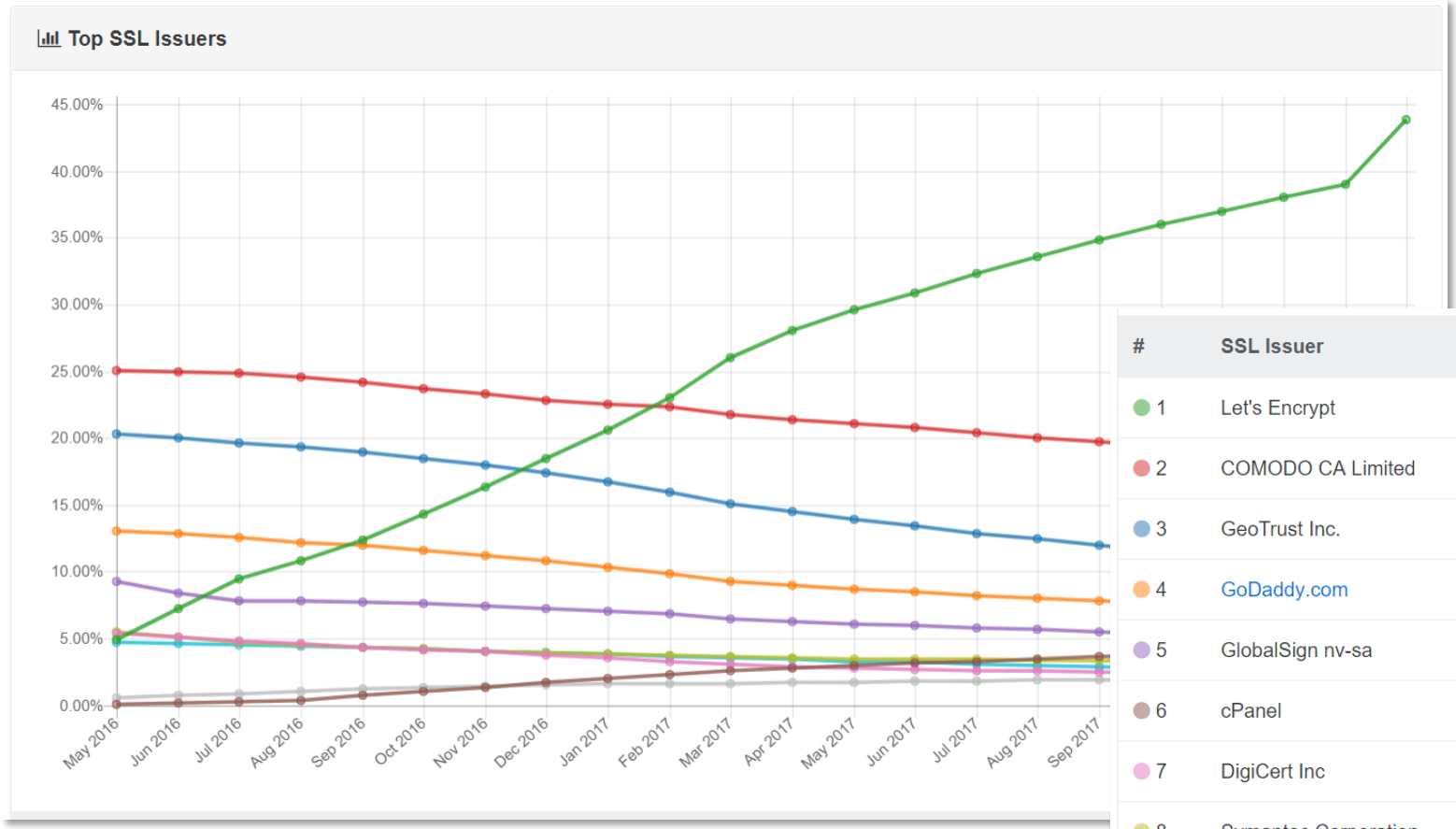


DV/OV certificate

EV certificate

- Increased assurance in identity of certificate subject
- Phishing attacks harder to accomplish
- In case of malicious server → better traceability for law enforcement authorities
- However: still no guarantee for honesty of server (e.g. mafia owned company can get EV certificate)

Top Certificate Issuer



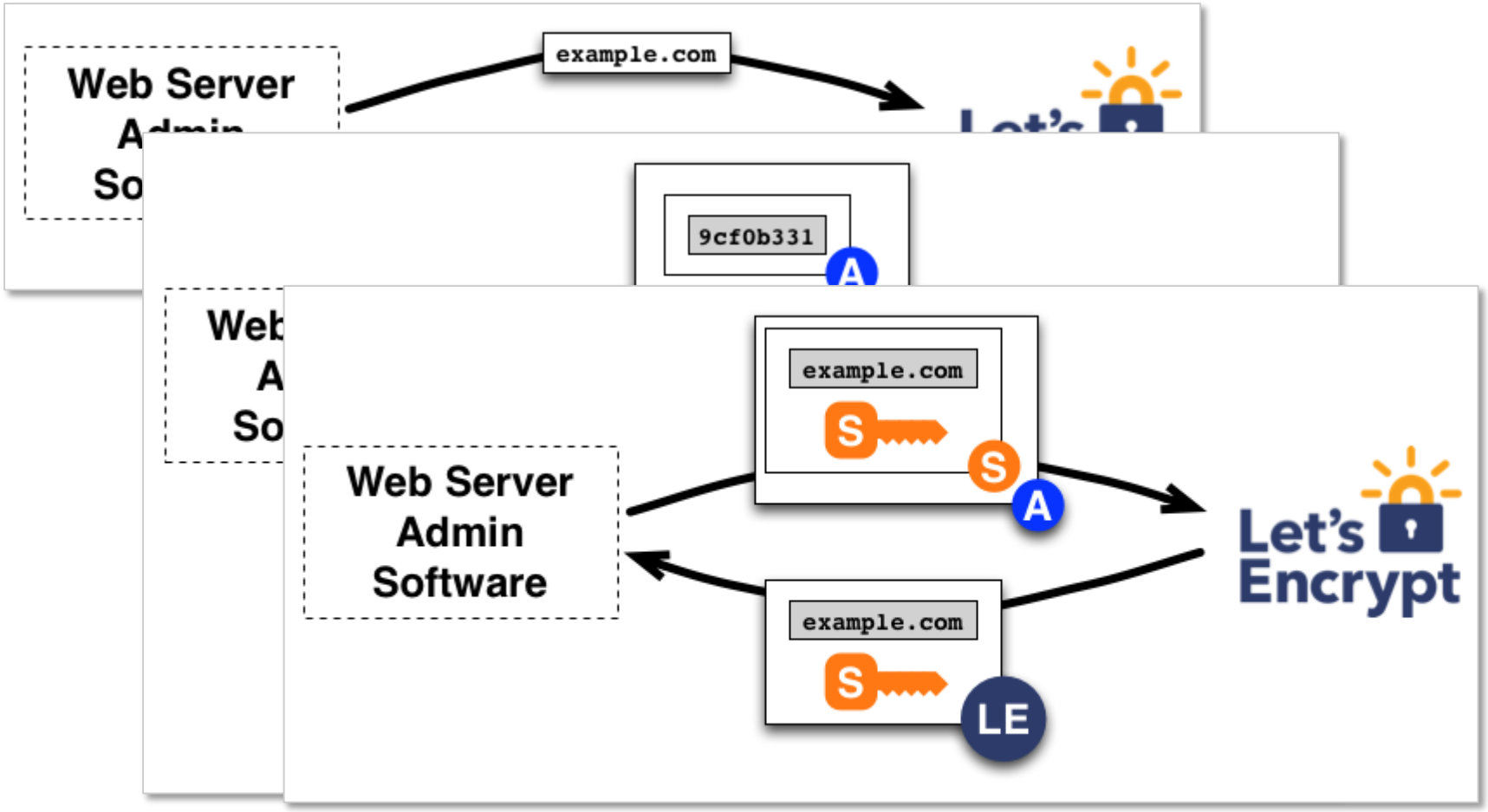
#	SSL Issuer	Percentage
1	Let's Encrypt	43.85%
2	COMODO CA Limited	17.15%
3	GeoTrust Inc.	8.17%
4	GoDaddy.com	6.66%
5	GlobalSign nv-sa	4.47%
6	cPanel	3.95%
7	DigiCert Inc	3.37%
8	Symantec Corporation	2.84%
9	thawte	2.01%
10	Amazon	1.67%

Source: https://nettrack.info/ssl_certificate_issuers.html

Let‘s Encrypt (LE)

- Created by the Internet Security Research Group (ISRG):
  - Akamai
  - Mozilla
  - Cisco
  - Google
  - Electronic Frontier Foundation (EFF)
- Goal: simple and free certificates for TLS
- Automatic certificate issuance and renewal processes
  - Automated Certificate Management Environment (ACME) protocol
  - not feasible for extended validation (EV) certificates

Automated Certificate Management Environment (ACME)



Source: https://letsencrypt.org/

Automated Certificate Management Environment (ACME)

- Different domain validation methods (simplified):
  - HTTP
    - ACME client puts a challenge to a specific location on the Web Server
    - ACME server resolves domain and downloads http://domain/.well-known/acme-challenge/<challenge-file-name>
  - TLS-SNI
    - ACME client installs a self-signed certificate for a subject named "<challenge>.acme.invalid"
    - ACME server resolves domain and initiates TLS connection to retrieve certificate
  - DNS
    - ACME client enters challenge into the TXT resource record of the domain
    - ACME server resolves domain and requests TXT resource record
    - DNS entry proofs possession of complete domain → wildcard certificates possible

Let‘s Encrypt – The Downside

March 20, 2017 21

PayPal Phishing Certificates Far More Prevalent Than Previously Thought

Over 14,000 SSL Certificates issued to PayPal phishing sites.

Earlier this month I discussed the use of Let's Encrypt certificates on PayPal phishing sites. In that article I asked Let's Encrypt to stop issuing certificates containing the term "PayPal" because of the high likelihood they would be used for phishing.

That requested stemmed from the fact that PayPal is a high value target and that Let's Encrypt had already issued nearly 1,000 certificates containing the term "PayPal," more than 99% of which were intended for phishing sites.

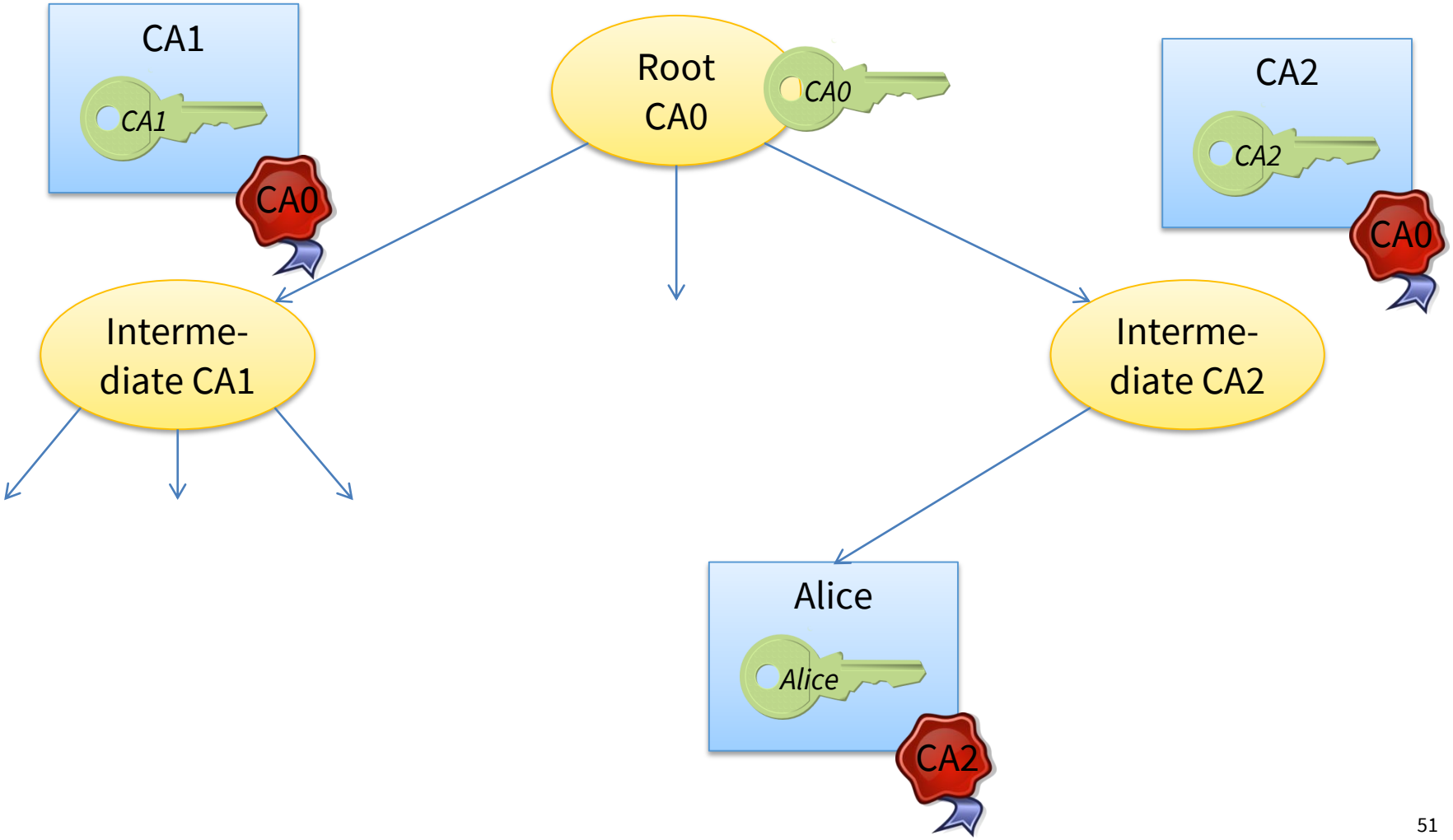
With expanded research, we found our previous claim was a major underestimate. Let's Encrypt has actually issued 15,270 "PayPal" certificates. This reveals the previously unknown extent of the Let's Encrypt phishing phenomenon.

Source: https://www.thesslstore.com/blog/lets-encrypt-phishing/

Certificate Trust

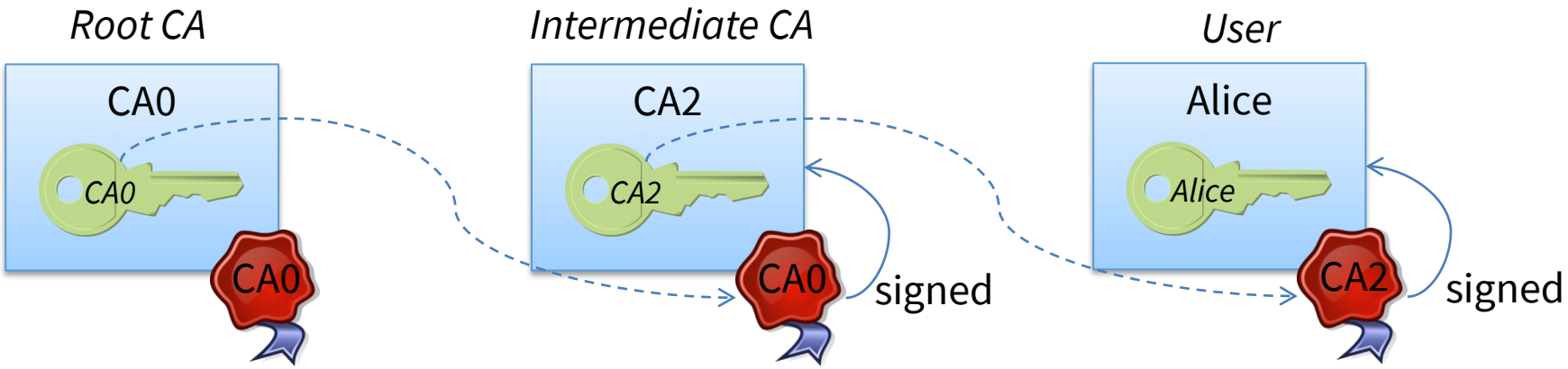
- Until now: only direct trust (CA → Certificate)
- Problem: Every CA must be known to the user
- Does not scale for large amount of certificates
- Solution: Delegation
  - Small amount of **Root CAs** issue certificates for **Intermediate CAs**
  - Intermediate CAs can issue certificates for other CAs or for end-entities
  - Users only need to know Root CAs

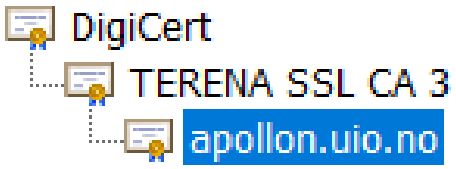
Certificate Trust



Certificate Trust

- When to trust a certificate?
- → there exists a signature chain from a trusted root CA

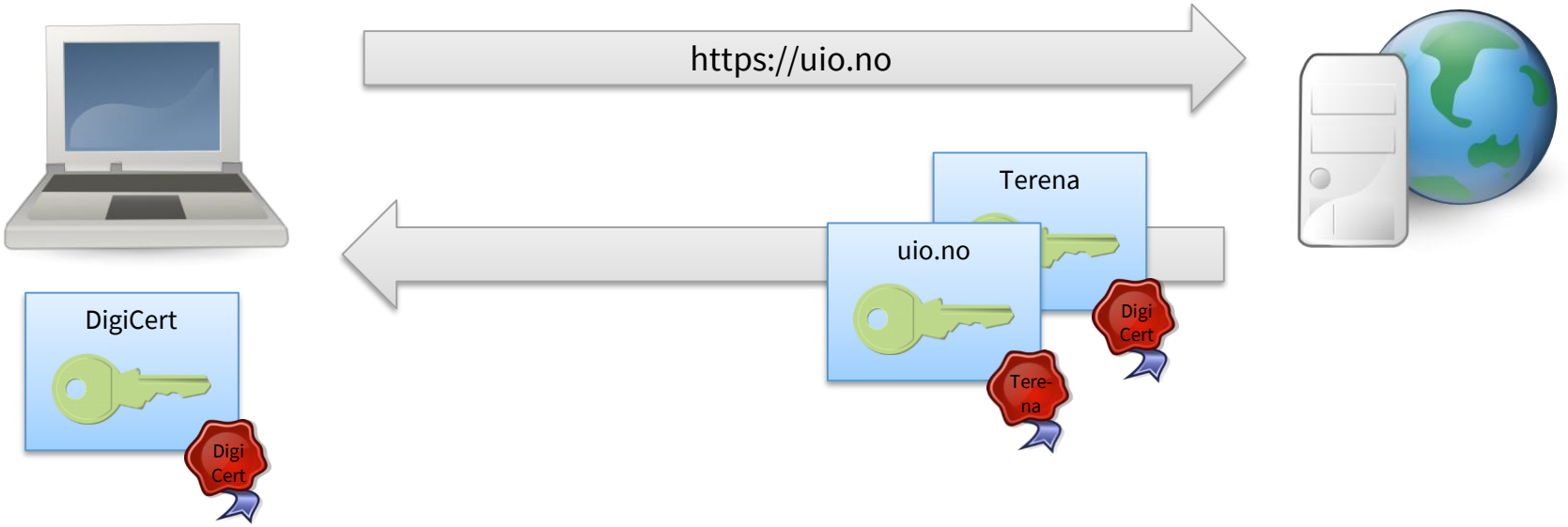




- Validity:
  - Digicert 10/2006 → 10/2031
  - Terena 10/2014 → 10/2024
  - uio.no 05/2017 → 05/2020

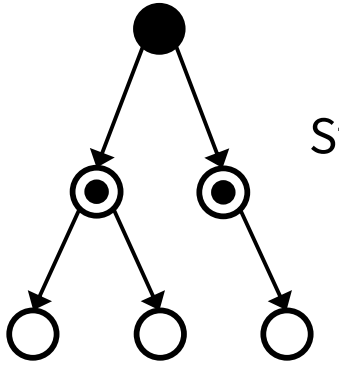
Certificate Trust

- How does the recipient retrieve the intermediate certificate?
- Usually the sender provides the intermediate certificate together with the user certificate
- Example (UiO):

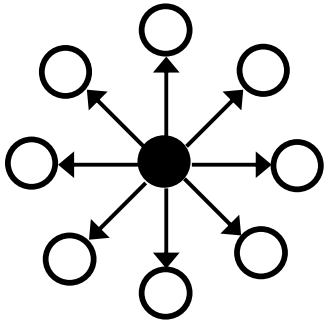


Trust Models

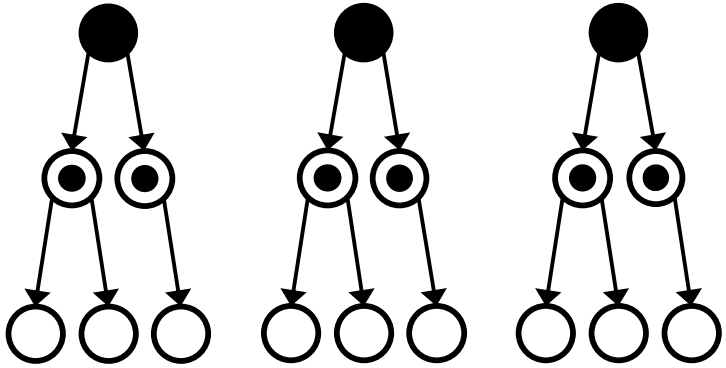
Self-signed root CA certificate
CA-signed intermediate CA certificate
CA-signed custom (leaf) certificate (cannot sign)



Strict hierarchy
e.g. DNSSEC



User-centric PKI
e.g. PGP



Isolated strict hierarchies
e.g. Web PKI

- Advantages of Web PKI trust model:
  - scales very well
  - users can choose from many (intermediate) CAs

PKI / Certificate Security

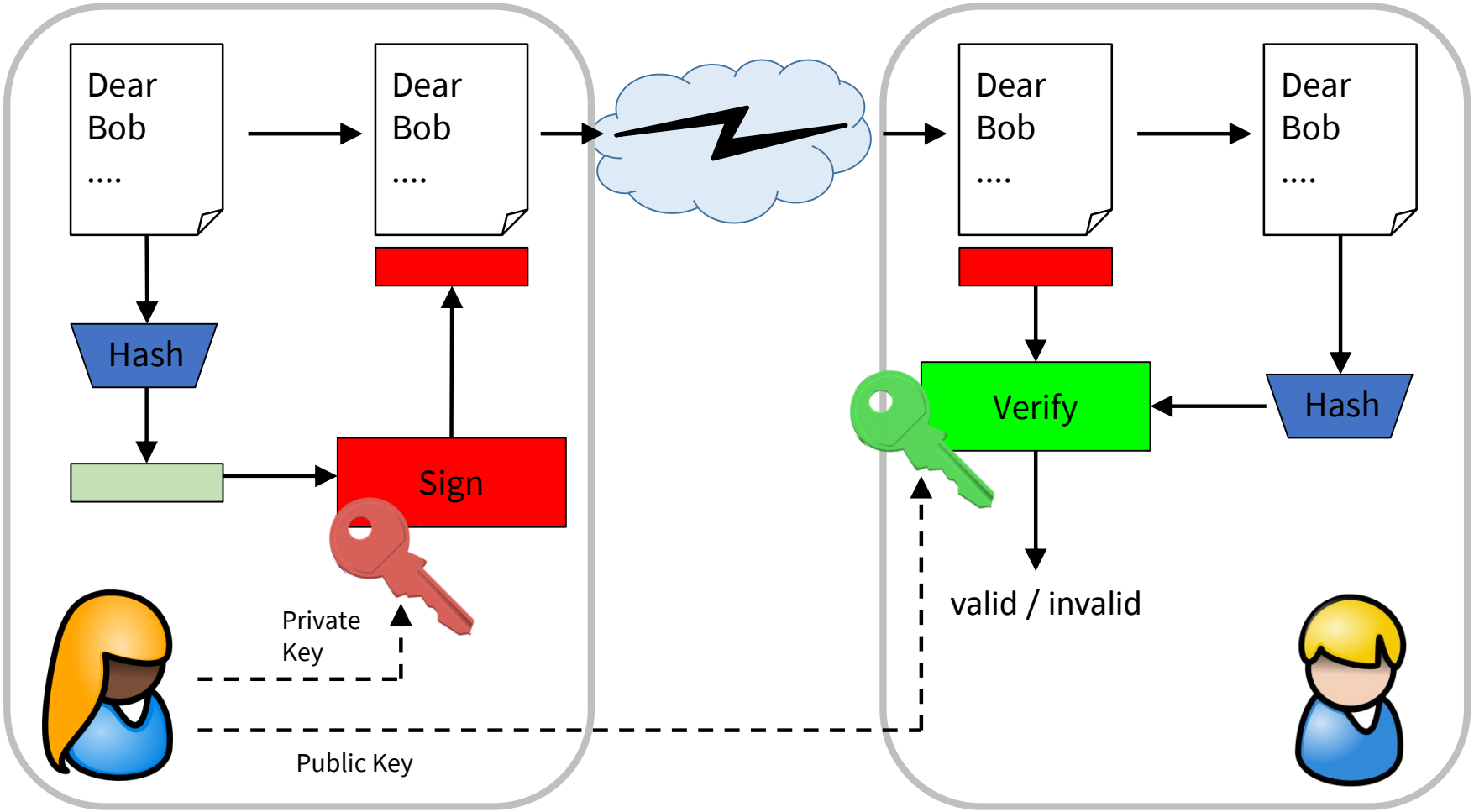
- Problems
  - Trust anchor required (trusted root store)
  - Trusted certificate ≠ trustworthy server
  - No binding between CA and end-entity
    → every CA can issue certificates for any domain
- (Some) threats:
  - Forging certificates
  - MITM attacks
  - Misconfigured client
  - Compromised server/certificate
  - Compromised
  - Sloppy
  - Rogue

  } certificate authority

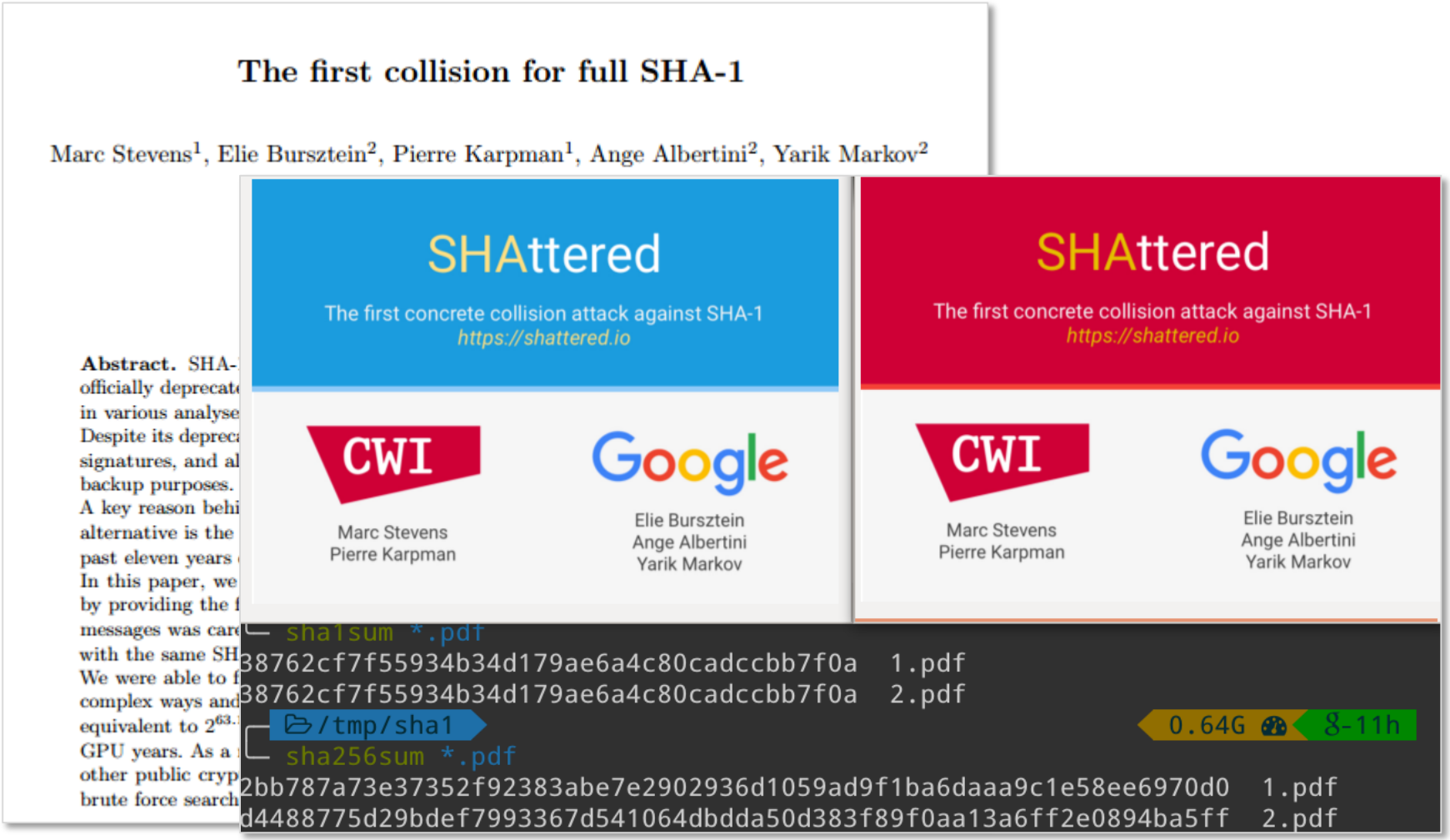
PKI / Certificate Security

General Threats

Recapitulation: Digital Signature

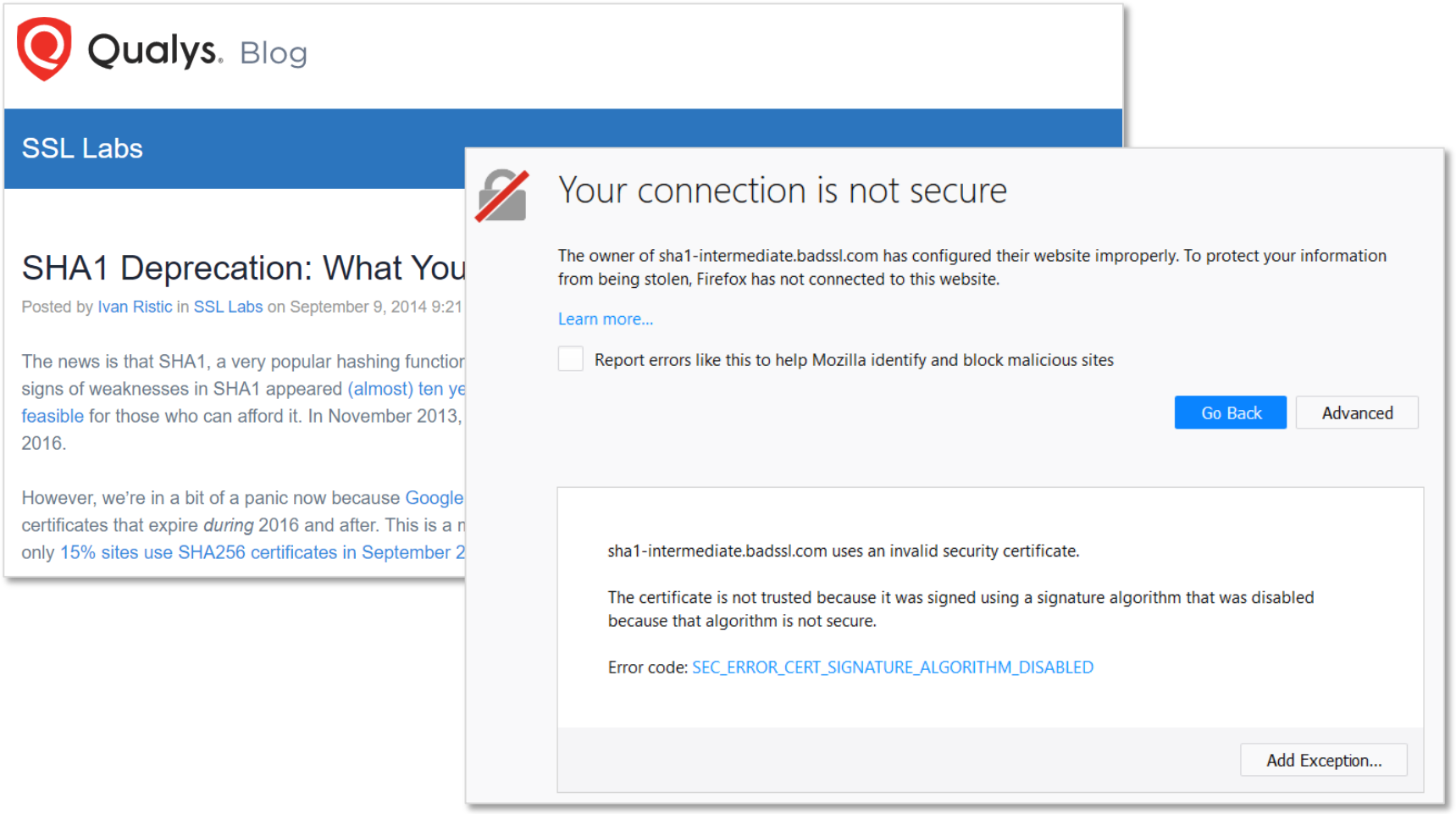


Attack on Hash Algorithm



Source: http://shattered.io

Attack on Hash Algorithm



Source: https://blog.qualys.com/ssllabs/2014/09/09/sha1-deprecation-what-you-need-to-know

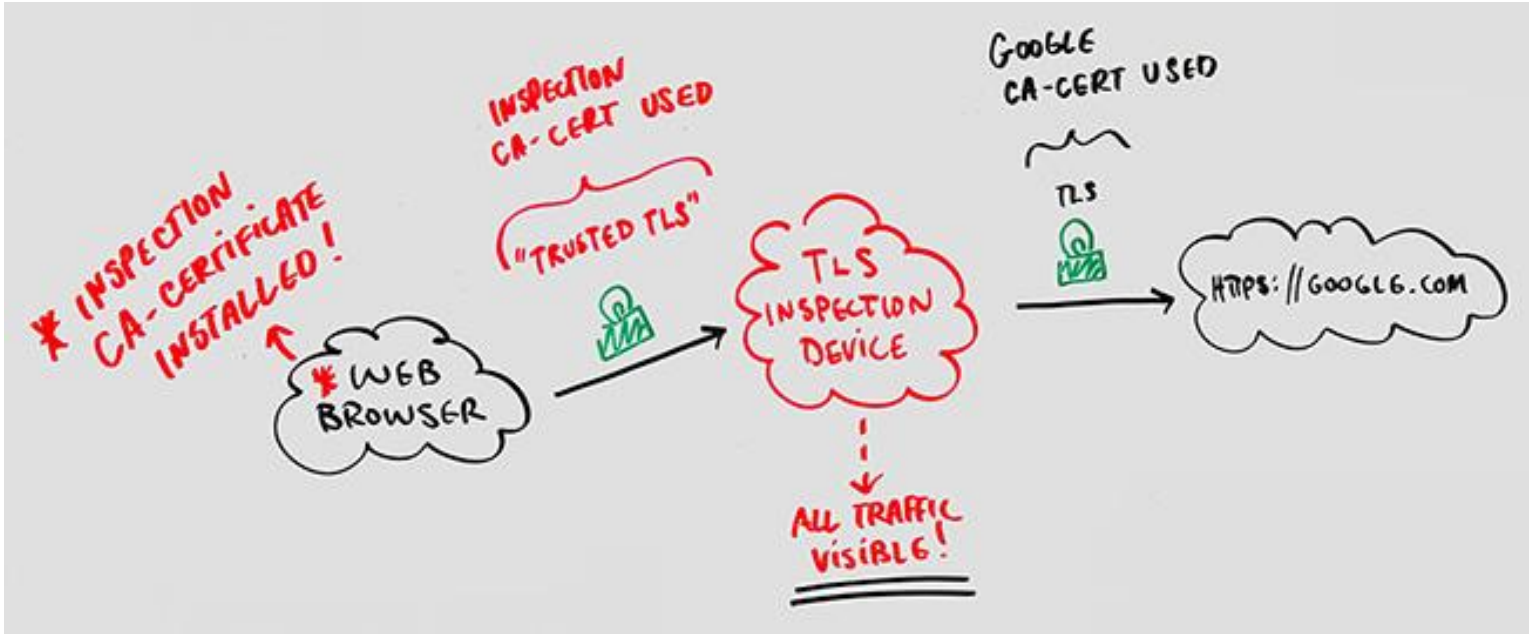
MITM: Downgrade attack

- Force/trick user to non-TLS connection
- Example:
  - Normal pattern:
    - User types „example.com“ in the browser → http://example.com
    - Server sends redirect (302) → https:// example.com
  - Malicious pattern:
    - User types „google.com“ in the browser → http://example.com
    - Attacker drops redirection but requests https://example.com himself
- HTTP Strict Transport Security (HSTS)
  - Server sends (on first visit) HSTS response header, e.g.:

    ```
    Strict-Transport-Security: max-age=31536000
    ```
  - Browser will only allow HTTPS connections for the specified durations
  - Problems:
    - “Trust on first use”
    - Can be misused for Web tracking (“super cookie”)

MITM: TLS/SSL Inspection

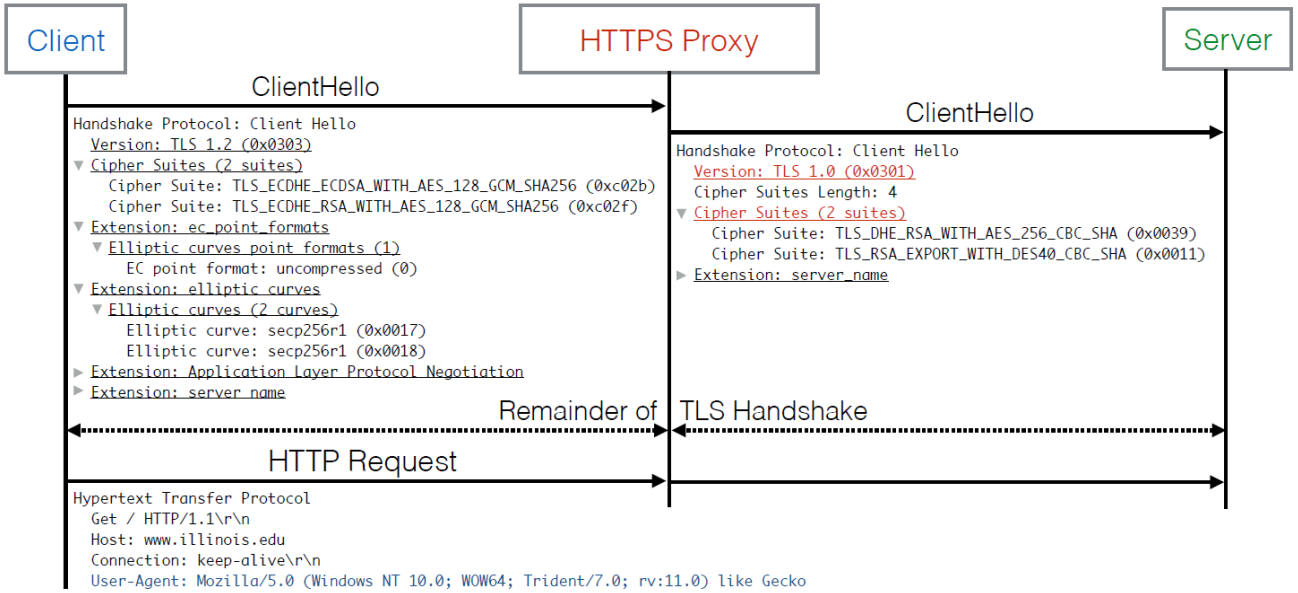
- "Security" proxies are breaking TLS connections and scanning content (e.g. antivirus, company policies)
- Prerequisites:
  proxy includes CA + root certificate installed on clients



Source: https://www.helpnetsecurity.com/2017/05/23/breaking-tls/

MITM: TLS/SSL Inspection

- Problems:
  - End to end confidentiality broken (user assumes “secure connection)
  - Many certificate security mechanisms (e.g. public key pinning, certificate transparency) are inoperable
  - Many proxies reduce the security level of the TLS connection



Source: Z. Durumeric et al.: *The security impact of HTTPS interception*, 2017.

Misconfigured Client

- Preinstalled root certificate (incl. private key!) on Dell computers
- Attacker can issue arbitrary certificates which are accepted by all affected computers

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

HOME | ABOUT US | CAREERS | PUBLICATIONS | ALERTS AND TIPS | RELATED RESOURCES

Dell Computers Contain CA Root Certificate Vulnerability

Original release date: November 24, 2015 | Last revised: November 27, 2015

Print Tweet Send Share

Dell personal computers using the preinstalled certificate authority (CA) root certificate (eDellRoot) contain a critical vulnerability. Exploitation of the vulnerability could allow a remote attacker to read encrypted web browser traffic (HTTPS), impersonate (spoof) any website, or perform other attacks on the affected system.

The eDellRoot certificate originated from an update to the Dell Foundation Services (DFS) application on August 18, 2015. As of November 23, that update is no longer being provided. The certificate was also preinstalled on some systems November 20–23, 2015. Dell is pushing a DFS software update to remove the vulnerable certificate from affected systems.

US-CERT encourages users and administrators to review Vulnerability Note VU#870761 and Dell's blog post for more information and guidance on removing the certificate.

Source: https://www.us-cert.gov/ncas/current-activity/2015/11/24/Dell-Computers-Contain-CA-Root-Certificate-Vulnerability

Sloppy Domain Owner

- User was able to register mail address hostmaster@live.fi
- This was used to request a certificate for domain live.fi

Microsoft Security Advisory 3046310

1 out of 1 rated this helpful - Rate this topic

Improperly Issued Digital Certificates Could Allow Spoofing

Published: March 16, 2015

Version: 1.0

Executive Summary

Microsoft is aware of an improperly issued SSL certificate for the domain "live.fi" that could be used in attempts to spoof content, perform phishing attacks, or perform man-in-the-middle attacks. It cannot be used to issue other certificates, impersonate other domains, or sign code. This issue affects all supported releases of Microsoft Windows. Microsoft is not currently aware of attacks related to this issue.

To help protect customers from potentially fraudulent use of this digital certificate, it has been revoked by the issuing CA and Microsoft is updating the Certificate Trust list (CTL) for all supported releases of Microsoft Windows to remove the trust of certificates that are causing this issue. For more information about these certificates, see the **Frequently Asked Questions** section of this advisory.

On this page

Executive Summary
Advisory Details
Affected Software
Advisory FAQ
Suggested Actions
Other Information

Source: https://docs.microsoft.com/en-us/security-updates/securityadvisories/2015/3046310/

Misusing DNS

- Wrong entry in DNS → domain validation useless
- Example: Cloud “Infrastructure as a Service”
  - Virtual servers are often used only for a short time
  - IPv4 address are quickly reused by the cloud provider for other cloud users
  - Cloud provider offers APIs for requesting free IP addresses
  - DNS entries are not changed immediately or are cached due to long TTL
  - → Attacker can easily (in the conducted experiment: 70 s) instantiate a virtual machine with a specific IP address with an out-dated DNS reference + request a certificate for this domain

Source: Borgolteet et. al: "Cloud Strife: Mitigating the Security Risks of Domain-Validated Certificates," in Proc. Internet Society Symposium on Network and Distributed System Security (NDSS), 2018.

PKI / Certificate Security

Compromised Certificate

Compromised Certificate

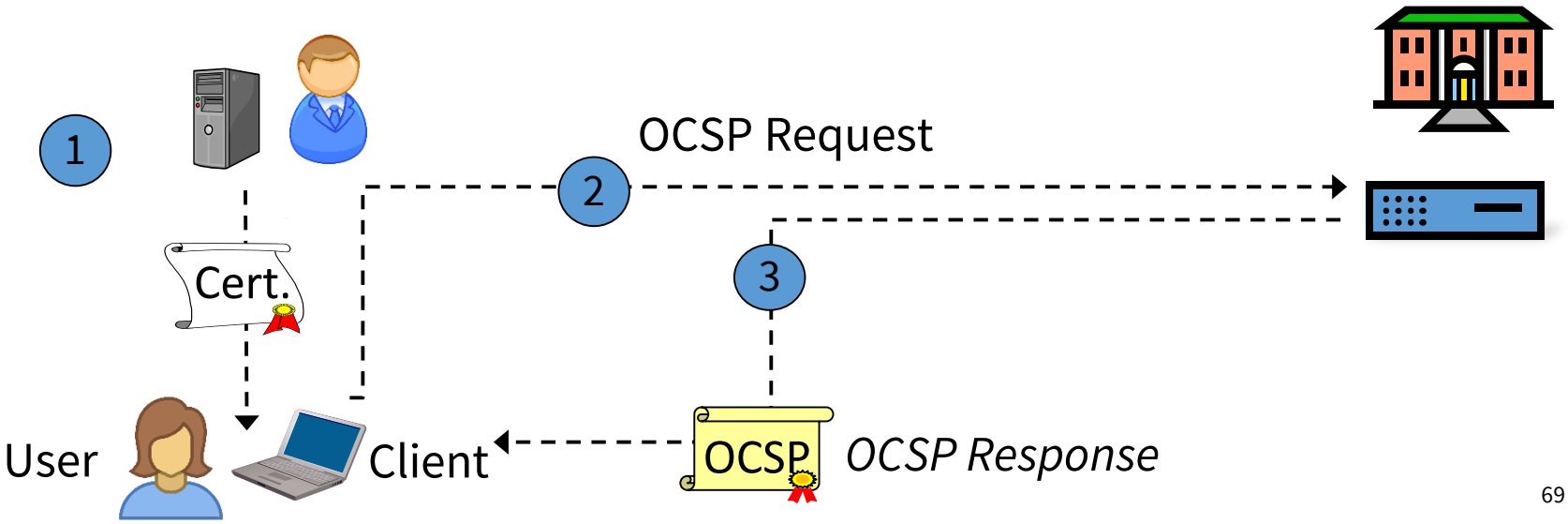
- What happens if certificate owner wants to invalidate a certificate (e.g. lost or stolen private key)?
  - Contact certificate authority
  - CA marks certificate as revoked
- How can the recipient of the certificate know of this revocation?
  - Certificate Revocation List (CRL)
  - Online Certificate Status Protocol (OCSP)

Certificate Revocation List (CRL)

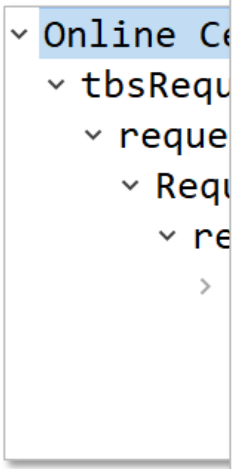
- Server/CA offers the list of revoked certificate for download
- Example (uio.no):
  - http://crl3.digicert.com/TERENASSLCA3.crl
  - http://crl4.digicert.com/TERENASSLCA3.crl
- Problems?
  - Download CRL for every TLS connection → additional delay
  - Download CRL in certain intervals → is CRL still up to date?
  - How often is the CRL updated at the CLR endpoint?
  - CRL can become very large → additional traffic / load
  - What is the browser supposed to do when the CRL endpoint is not accessible (hard-fail/soft-fail)?

Online Certificate Status Protocol (OCSP)

- Interactive protocol to validate if the certificate is still valid
- Example (uio.no):
  - http://ocsp.digicert.com
- Client sends a request to the CA containing the serial number
- CA sends a responds which is digitally signed



Online Certificate Status Protocol (OCSP)



OCSP Req

```
Online Certificate Status Protocol
  responseStatus: successful (0)
  responseBytes
    ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
    BasicOCSPResponse
      tbsResponseData
        responderID: byKey (2)
        producedAt: 2018-02-22 03:01:49 (UTC)
        responses: 1 item
          SingleResponse
            certID
              hashAlgorithm (SHA-1)
              issuerNameHash: 1175295285b7738d52a8e3508fb390c5eec7d46a
              issuerKeyHash: 67fd8820142798c709d22519bbe9511163755062
              serialNumber: 0x0542236da9f7833846e7ce5bf4a462ee
            certStatus: good (0)
            thisUpdate: 2018-02-22 03:01:49 (UTC)
            nextUpdate: 2018-03-01 02:16:49 (UTC)
      signatureAlgorithm (sha256WithRSAEncryption)
      Padding: 0
      signature: 03f30f15f7e6428a5eb60b97fd706031aa366bfd517d32ea...
```

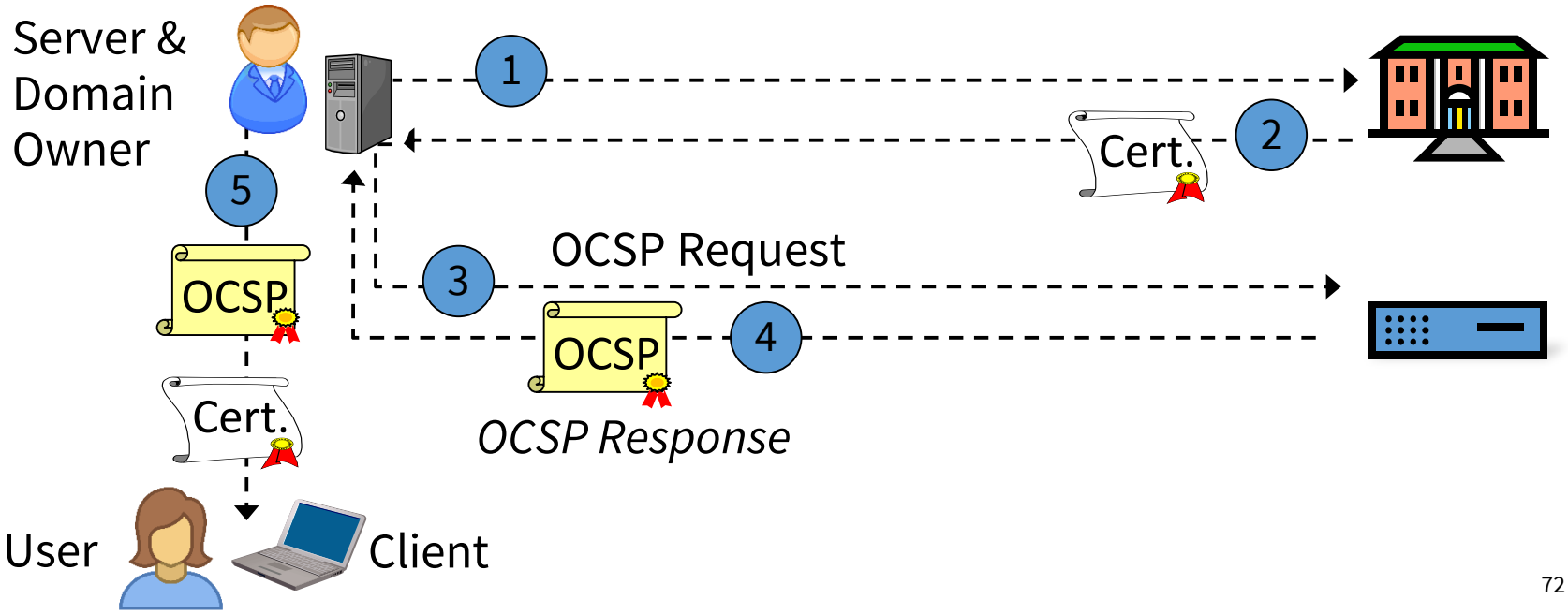
OCSP Response

Online Certificate Status Protocol (OCSP)

- Advantages compared to CRL?
  - Allows (theoretically) realtime access to certificate status
  - Reduced traffic
- Problems remaining?
  - Often implemented at the CA using a CRL
  - Delay in TLS connection setup
  - Attacker can block access to the OCSP endpoint
  - What is the browser supposed to do when the OCSP endpoint is not accessible?
- New problems?
  - CA learns which (HTTPS) Web pages have been accessed by the client

OCSP stapling

- Extension of the TLS protocol
- OCSP Certificate is **not** requested by the client at the CA
- Server request OCSP Certificate at the CA and send it during the TLS handshake to the client



OCSP stapling

```
Extension: status_request (len=5)
    Type: status_request (5)
    Length: 5
    Certificate Status Type: OCSP (1)
    Responder ID list Length: 0
    Request Extensions Length: 0
```

Status request from Client (inside TLS ”Client Hello” message)

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 5985
  > Handshake Protocol: Server Hello
  > Handshake Protocol: Certificate
  > Handshake Protocol: Certificate Status
  > Handshake Protocol: Server Key Exchange
  > Handshake Protocol: Server Hello Done
```

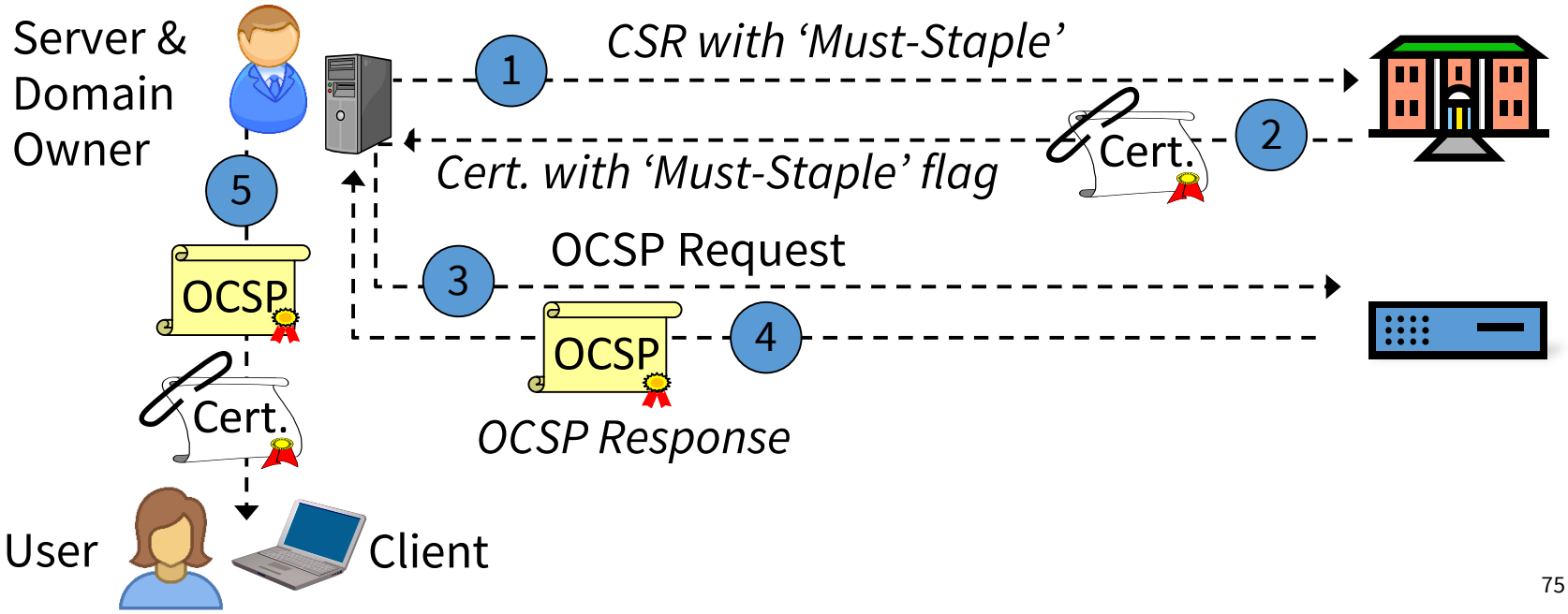
Certificate Status from server (after TLS ”Certificate” message)

OCSP stapling

- Advantages compared to OCSP?
  - Client does not contact the CA → no privacy issue
- Problems remaining?
  - Attacker („owner“ of private key for the compromised certificate) can deliver the certificate without the OCSP status

OCSP "Must-Staple"

- The certificate is issued with a flag indicating a mandatory OCSP status response

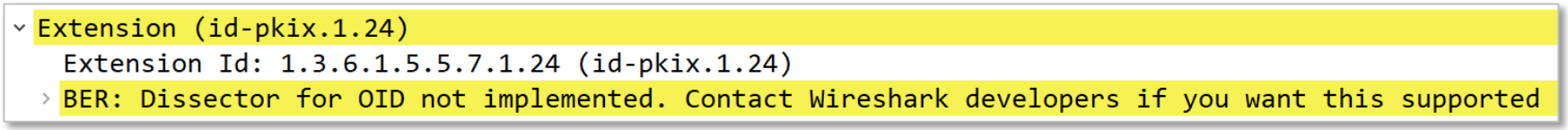


OCSP “Must-Staple”

- Advantages compared to OCSP stapling?
  - Client detects a missing OCSP status
- Problems remaining?
  - What is the browser supposed to do when the OCSP status is missing?
  - Insufficient implementation support (client, server, network tools)
  - Not used by any major Web site

OCSP “Must-Staple”

- Advantages compared to OCSP stapling?
  - Client detects a missing OCSP status
- Problems remaining?
  - What is the browser supposed to do when the OCSP status is missing?
  - Insufficient implementation support (client, server, network tools)
  - No widespread use yet

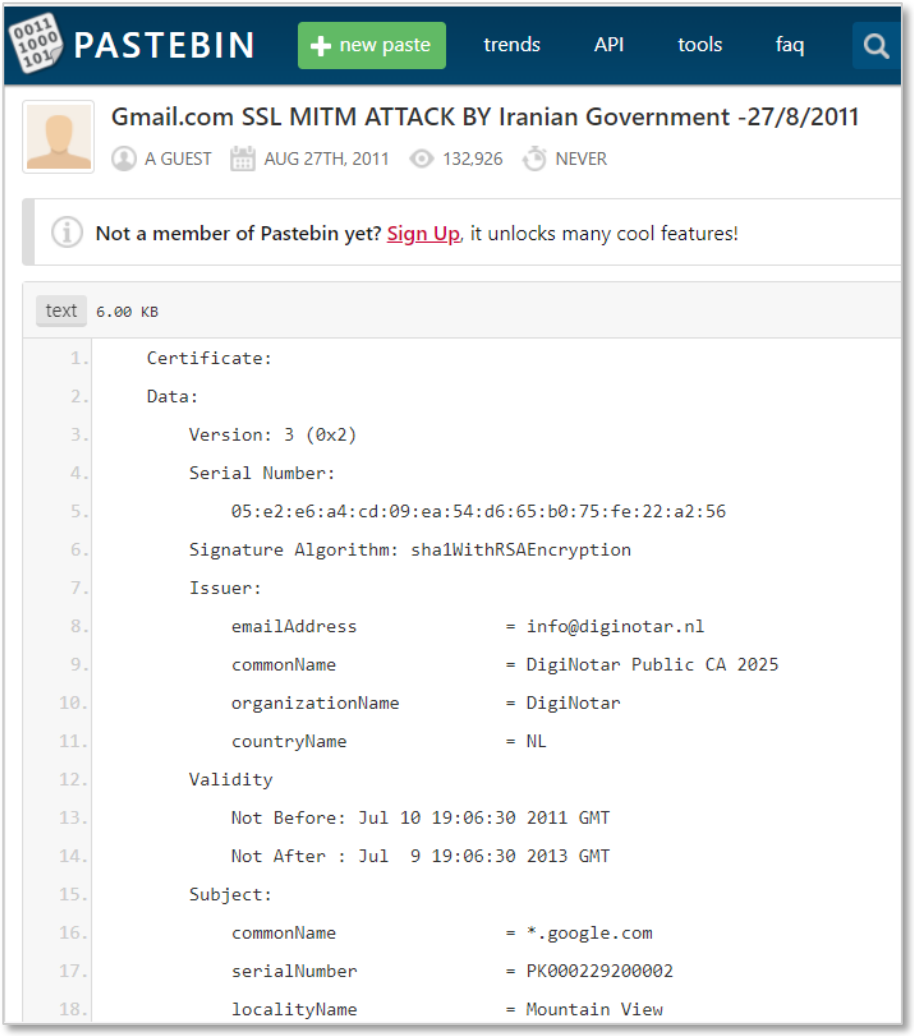


PKI / Certificate Security

Compromised/Sloppy/Rogue Certificate Authority

Compromised Certificate Authority

- CA DigiNotar was hacked in 2011
- A number of illegitimate certificates (e.g. *.google.com) were created by the intruders



Source: https://pastebin.com/ff7Yg663

Compromised Certificate Authority

- CA DigiNotar was hacked in 2011
- A number of illegitimate certificates (e.g. *.google.com) were created by the intruders

KIM ZETTER SECURITY 09.20.11 03:05 PM

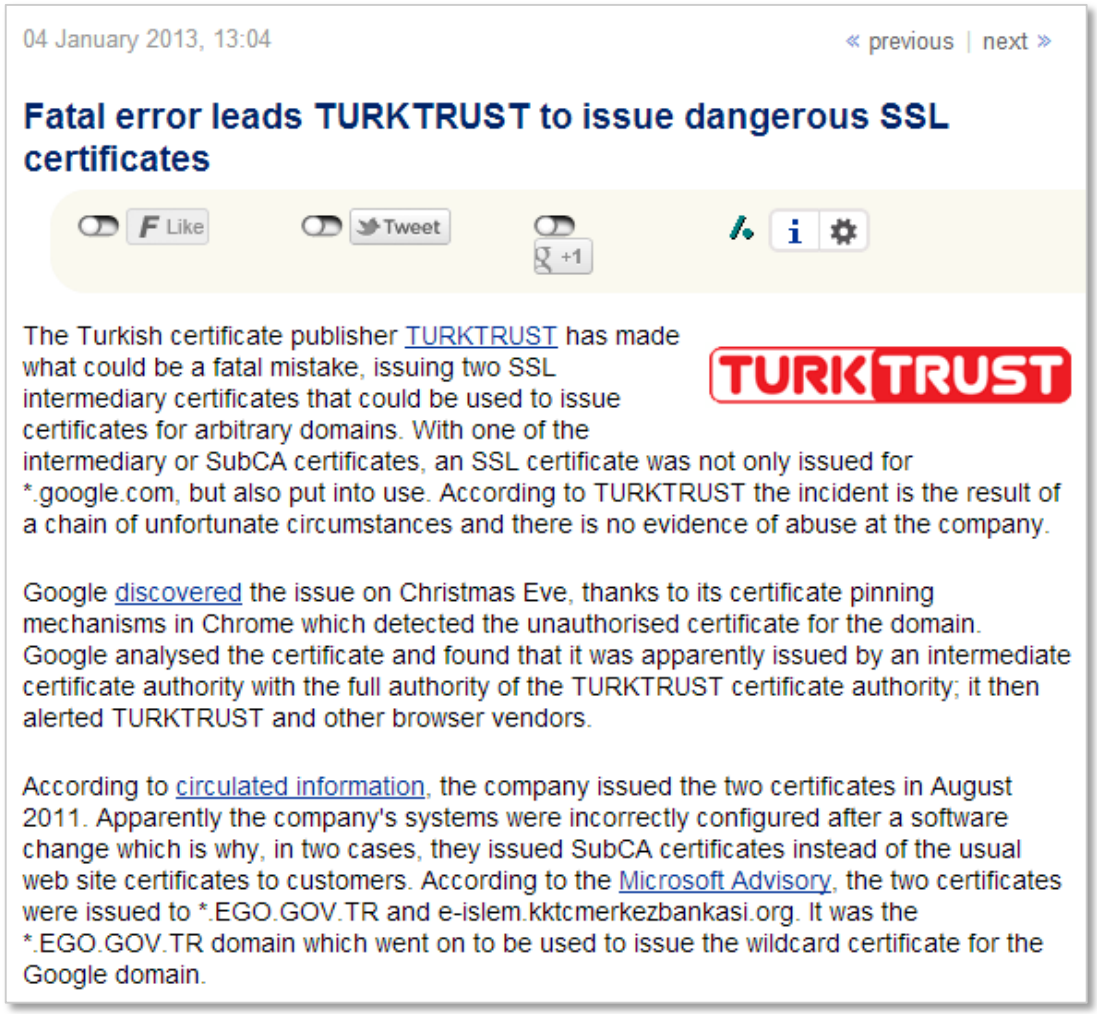
DIGINOTAR FILES FOR BANKRUPTCY IN WAKE OF DEVASTATING HACK

A Dutch certificate authority that suffered a major hack attack this summer has been unable to recover from the blow and filed for bankruptcy this week.

Source: https://www.wired.com/2011/09/diginotar-bankruptcy/

Sloppy Certificate Authority

- CA issued CA certificates to end-entities
- Issue remained undetected for 15 month



04 January 2013, 13:04 « previous | next »

Fatal error leads TURKTRUST to issue dangerous SSL certificates

Like Tweet +1

The Turkish certificate publisher TURKTRUST has made what could be a fatal mistake, issuing two SSL intermediary certificates that could be used to issue certificates for arbitrary domains. With one of the intermediary or SubCA certificates, an SSL certificate was not only issued for *.google.com, but also put into use. According to TURKTRUST the incident is the result of a chain of unfortunate circumstances and there is no evidence of abuse at the company.

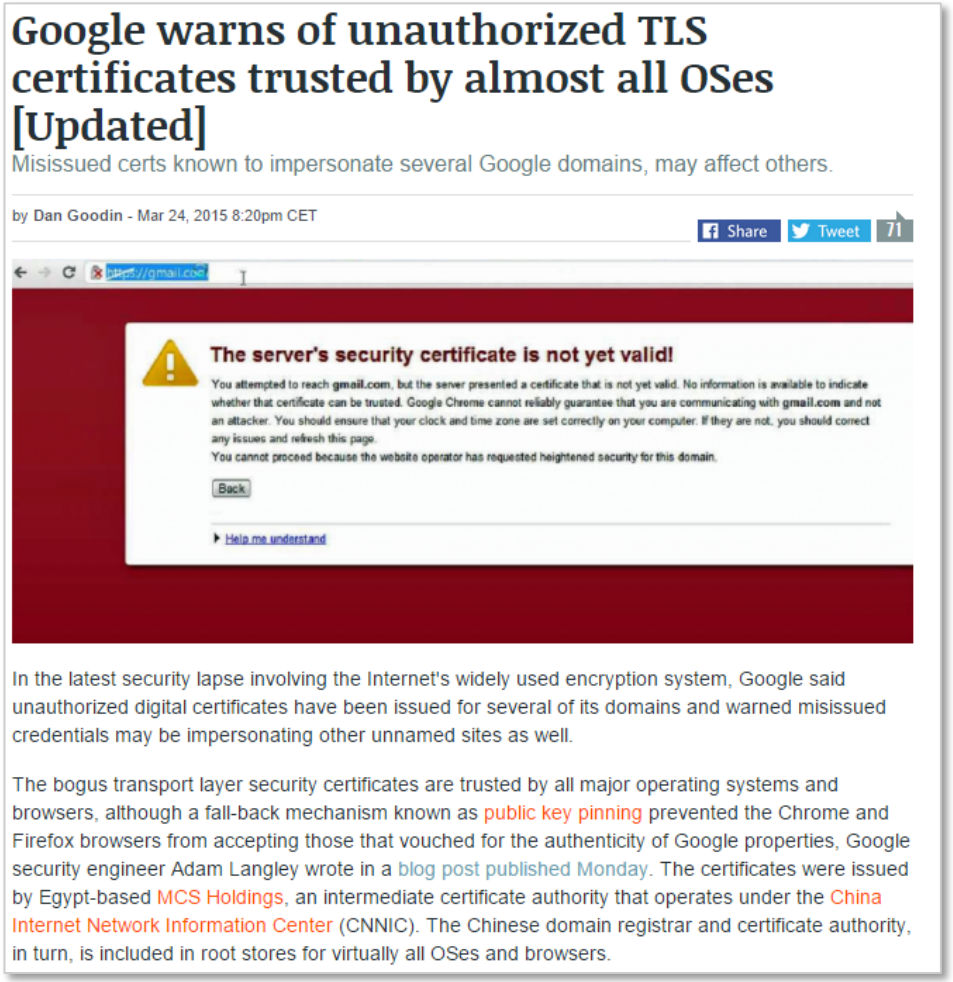
Google discovered the issue on Christmas Eve, thanks to its certificate pinning mechanisms in Chrome which detected the unauthorised certificate for the domain. Google analysed the certificate and found that it was apparently issued by an intermediate certificate authority with the full authority of the TURKTRUST certificate authority; it then alerted TURKTRUST and other browser vendors.

According to circulated information, the company issued the two certificates in August 2011. Apparently the company's systems were incorrectly configured after a software change which is why, in two cases, they issued SubCA certificates instead of the usual web site certificates to customers. According to the Microsoft Advisory, the two certificates were issued to *.EGO.GOV.TR and e-islem.kktcmerkezbankasi.org. It was the *.EGO.GOV.TR domain which went on to be used to issue the wildcard certificate for the Google domain.

Source: http://h-online.com/-1777291

Sloppy (Rogue?) Certificate Authority

- CA issued certificates which were not requested by the domain owner
- These certificates are accepted by all (or most) clients



Google warns of unauthorized TLS certificates trusted by almost all OSes [Updated]

Misissued certs known to impersonate several Google domains, may affect others.

by Dan Goodin - Mar 24, 2015 8:20pm CET

Share Tweet 71

https://gmail.com

The server's security certificate is not yet valid!

You attempted to reach **gmail.com**, but the server presented a certificate that is not yet valid. No information is available to indicate whether that certificate can be trusted. Google Chrome cannot reliably guarantee that you are communicating with **gmail.com** and not an attacker. You should ensure that your clock and time zone are set correctly on your computer. If they are not, you should correct any issues and refresh this page.

You cannot proceed because the website operator has requested heightened security for this domain.

Back

Help me understand

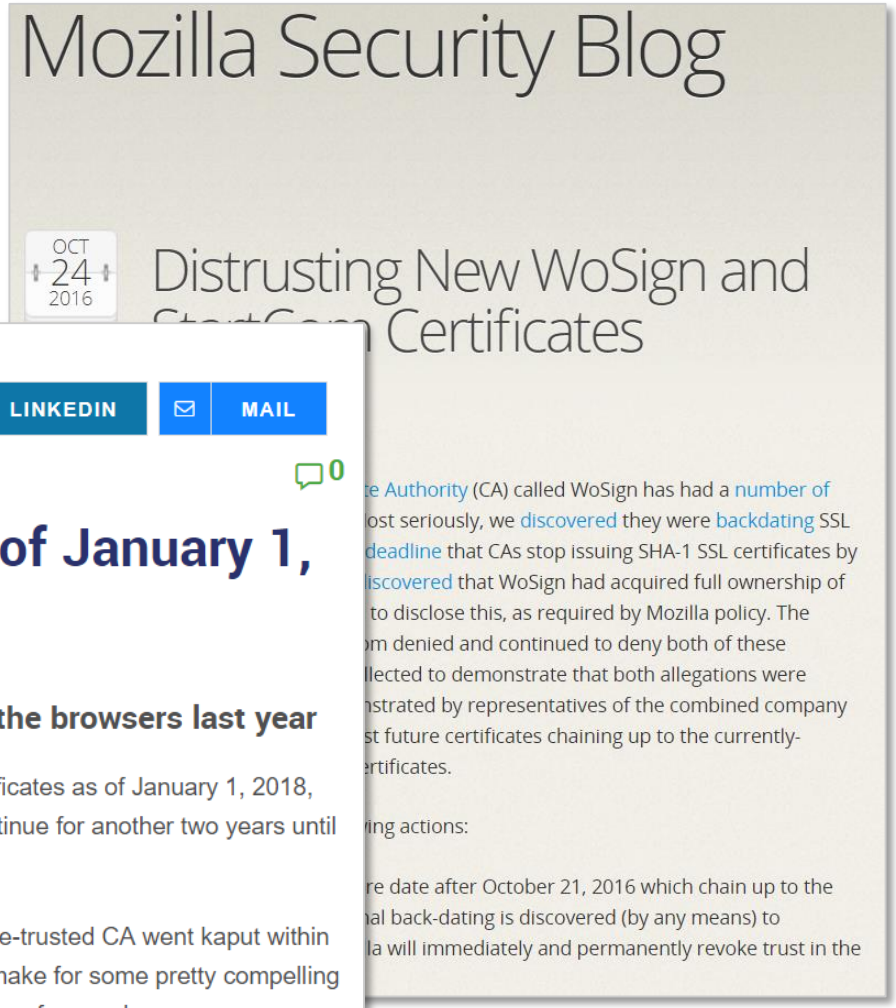
In the latest security lapse involving the Internet's widely used encryption system, Google said unauthorized digital certificates have been issued for several of its domains and warned misissued credentials may be impersonating other unnamed sites as well.

The bogus transport layer security certificates are trusted by all major operating systems and browsers, although a fall-back mechanism known as public key pinning prevented the Chrome and Firefox browsers from accepting those that vouched for the authenticity of Google properties, Google security engineer Adam Langley wrote in a blog post published Monday. The certificates were issued by Egypt-based MCS Holdings, an intermediate certificate authority that operates under the China Internet Network Information Center (CNNIC). The Chinese domain registrar and certificate authority, in turn, is included in root stores for virtually all OSes and browsers.

Source: https://arstechnica.com

Sloppy Certificate Authority

- CA created SHA-1 signed certificates and backdated them



Mozilla Security Blog

OCT 24 2016

Distrusting New WoSign and StartCom Certificates

Authority (CA) called WoSign has had a number of
ost seriously, we discovered they were backdating SSL
deadline that CAs stop issuing SHA-1 SSL certificates by
iscovered that WoSign had acquired full ownership of
to disclose this, as required by Mozilla policy. The
m denied and continued to deny both of these
llected to demonstrate that both allegations were
nstrated by representatives of the combined company
st future certificates chaining up to the currently-
rtificates.

ing actions:

re date after October 21, 2016 which chain up to the
al back-dating is discovered (by any means) to
la will immediately and permanently revoke trust in the

(4 votes, average: 5.00 out of 5)

FACEBOOK TWITTER GOOGLE + LINKEDIN MAIL

November 20, 2017 0

StartCom SSL Shutting Down as of January 1, 2018

StartCom SSL couldn't overcome being distrusted by the browsers last year

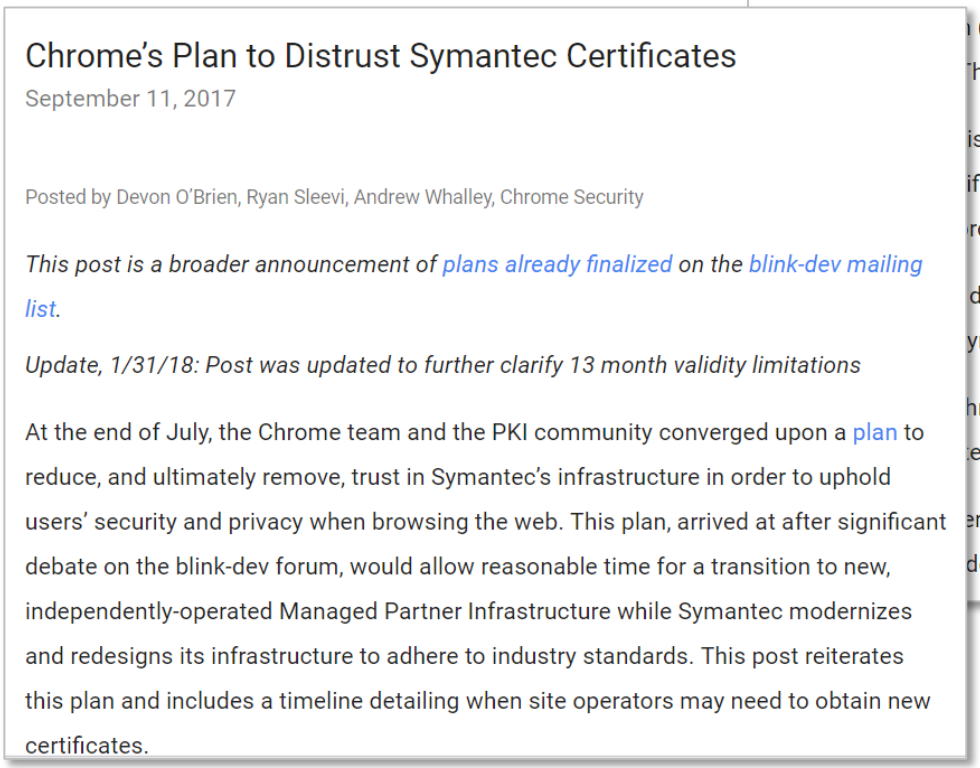
StartCom SSL has announced that it will no longer issue new digital certificates as of January 1, 2018, effectively closing the company, though CRL and OCSP services will continue for another two years until StartCom's three roots expire in 2020.

This marks the end of an odd, perhaps even cautionary tale of how a once-trusted CA went kaput within about a year of the browsers distrusting it. Seriously, this would actually make for some pretty compelling drama because what happened to StartCom feels straight out of the pages of a novel.

Source: https://blog.mozilla.org/security/2016/10/24/distrusting-new-wosign-and-startcom-certificates/
https://www.thesslstore.com/blog/startcom-ssl-shutting-down-2018/

Sloppy CA – The Symantec Case

- CA issued certificates which were not requested by the domain owner



Chrome's Plan to Distrust Symantec Certificates

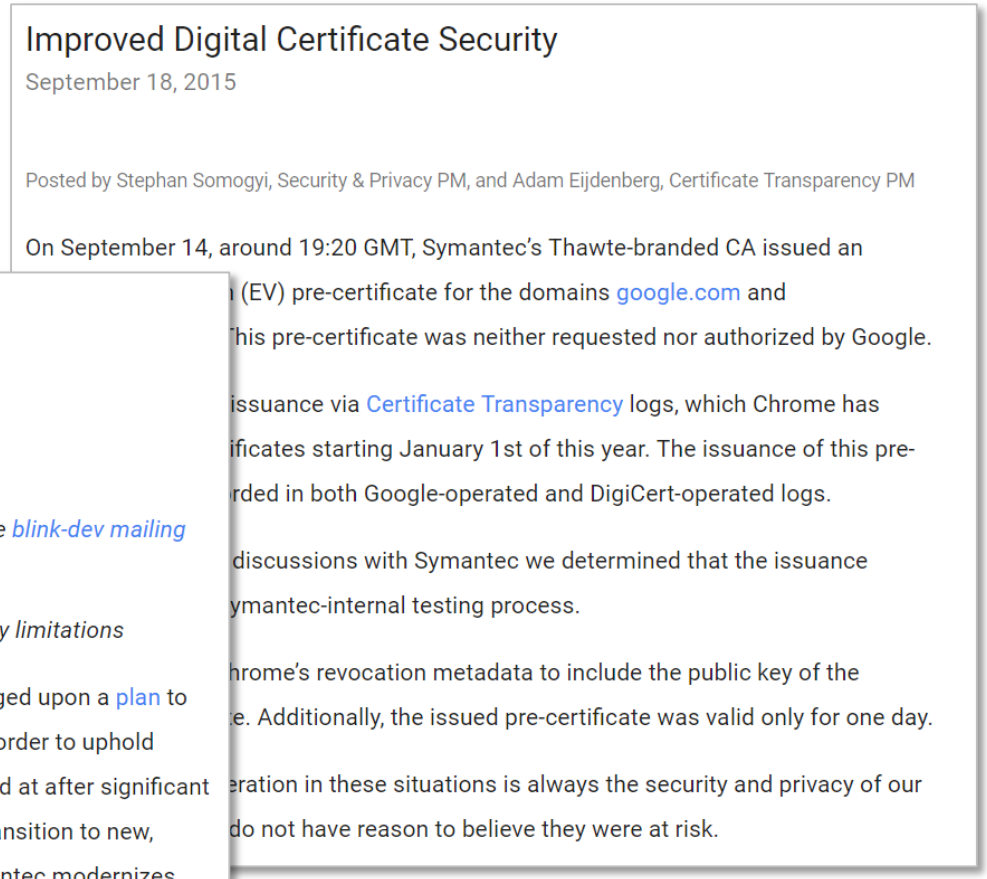
September 11, 2017

Posted by Devon O'Brien, Ryan Sleevi, Andrew Whalley, Chrome Security

This post is a broader announcement of plans already finalized on the blink-dev mailing list.

Update, 1/31/18: Post was updated to further clarify 13 month validity limitations

At the end of July, the Chrome team and the PKI community converged upon a plan to reduce, and ultimately remove, trust in Symantec's infrastructure in order to uphold users' security and privacy when browsing the web. This plan, arrived at after significant debate on the blink-dev forum, would allow reasonable time for a transition to new, independently-operated Managed Partner Infrastructure while Symantec modernizes and redesigns its infrastructure to adhere to industry standards. This post reiterates this plan and includes a timeline detailing when site operators may need to obtain new certificates.



Improved Digital Certificate Security

September 18, 2015

Posted by Stephan Somogyi, Security & Privacy PM, and Adam Eijdenberg, Certificate Transparency PM

On September 14, around 19:20 GMT, Symantec's Thawte-branded CA issued an
(EV) pre-certificate for the domains google.com and
his pre-certificate was neither requested nor authorized by Google.

issuance via Certificate Transparency logs, which Chrome has
ificates starting January 1st of this year. The issuance of this pre-
rded in both Google-operated and DigiCert-operated logs.

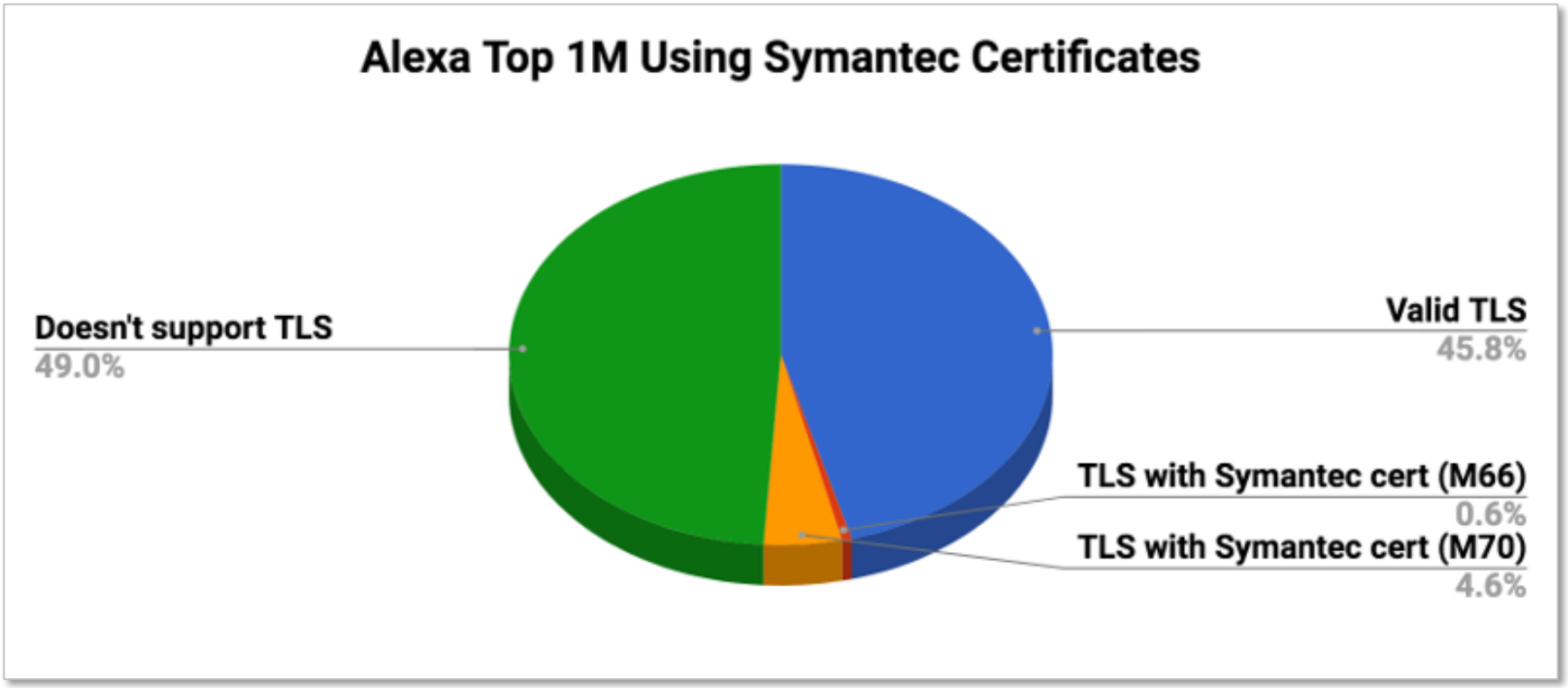
discussions with Symantec we determined that the issuance
ymantec-internal testing process.

hrome's revocation metadata to include the public key of the
e. Additionally, the issued pre-certificate was valid only for one day.

eration in these situations is always the security and privacy of our
do not have reason to believe they were at risk.

Source: https://security.googleblog.com/2015/09/improved-digital-certificate-security.html
https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html#

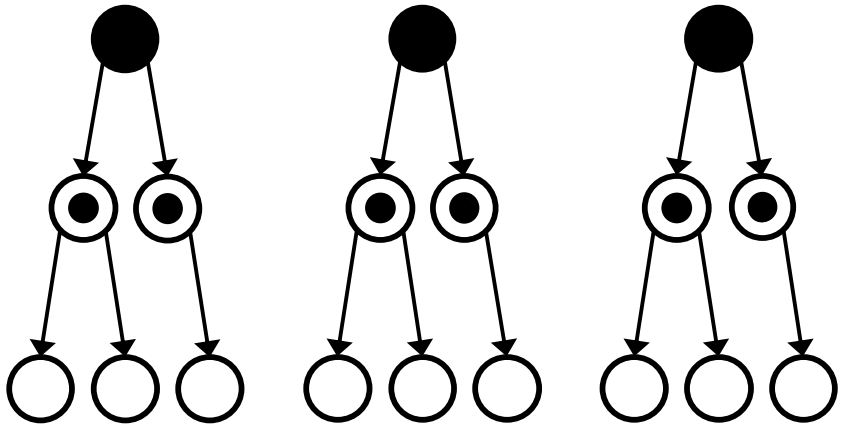
The Symantec Case – Affected Certificates



Source: https://arkadiyt.com/2018/02/04/quantifying-untrusted-symantec-certificates/

Compromised/Sloppy Certificate Authority

- HTTP Public Key Pinning (HPKP)
- DNS-based Authentication of Named Entities (DANE)
- DNS Certification Authority Authorization (CAA)
- Certificate Transparency (CT)



HTTP Public Key Pinning (HPKP)

- HTTPS server can “pin” the public keys for the TLS certificates
- Example (HPKP entry in a HTTP response header):

```
Public-Key-Pins:
  pin-sha256="cUPcTAZWKaASuYWhhneDttWpY3oBAkE3h2+soZS7sWs=";
  pin-sha256="M8HztCzM3elUxkcjR2S5P4hhyBNf6lHkmjAHKhpGPWE=";
  max-age=5184000; includeSubDomains
```

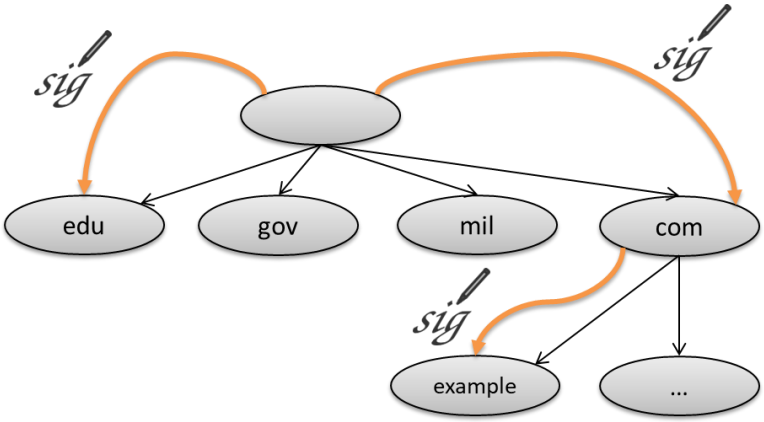
- The pinned key can belong to:
  - root certificate
  - intermediate certificate
  - end-entity certificate
- For the specified duration (here: 2 month) no other CA/certificate is accepted by the browser

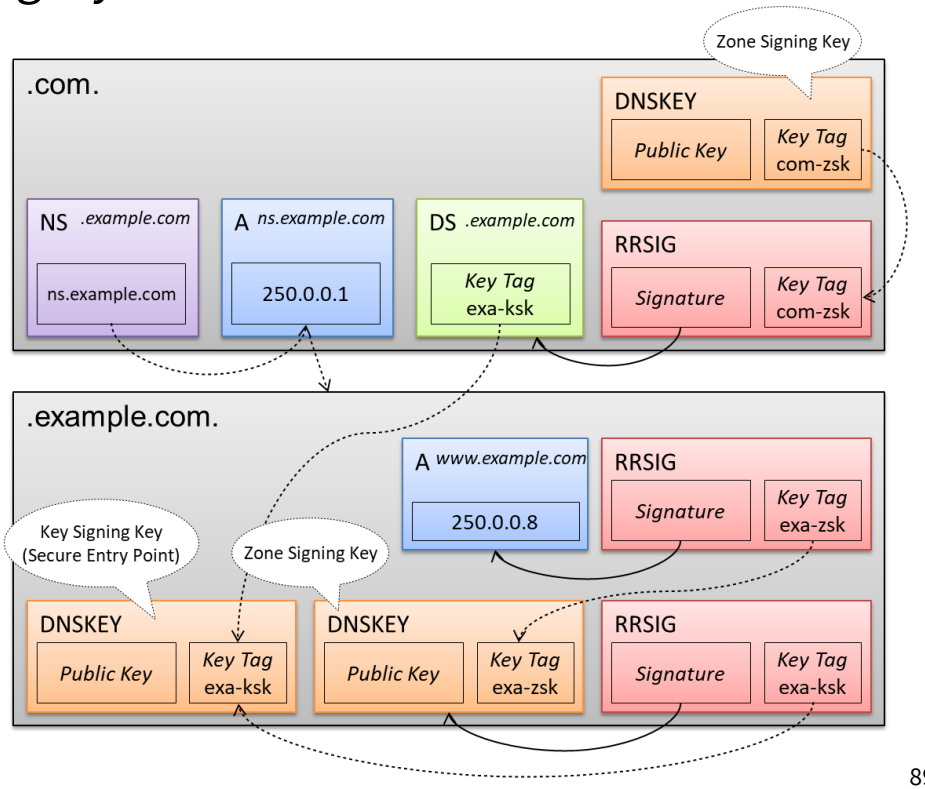
HTTP Public Key Pinning (HPKP)

- Problems:
  - "Trust on first use"
  - For certificate pins: if certificate is changed (e.g. compromised) → no connection
  - For CA pins: if CA goes out of business → certificate from different CA → no connection
  - Error prone server configuration → sites lock out clients
  - Possibility for blackmailing server owner: RansomPKP
  - Used only by very few Web sites
  - Only supported by Chrome browser
- Implication:
  - Will be removed from Chrome (May 2018)

DNS-based Authentication of Named Entities (DANE)

- DNSSEC:
  - Domain Name System Security Extensions
  - Ensures authenticity and integrity of DNS resource records





DNS-based Authentication of Named Entities (DANE)

- DANE adds a new record to the DNSSEC: TLSA
- TLSA record includes one of the following:
  - Trusted certificate
  - Trusted certificate authority (root CA or arbitrary CA)
- DNSSEC → no DNS spoofing possible
- Example:

```
www.example.com.              IN A 192.0.2.1
_443._tcp.www.example.com.    IN TLSA 3 1 1 (
                                    8A9A70596E869BED72C69D97A8895DFA
                                    D86F300A343FECEFF19E89C27C896BC9 )
```

DNS-based Authentication of Named Entities (DANE)

- Advantages:
  - No other certificates / CAs are trusted by the client
  - Works also completely without PKI
- Disadvantages:
  - DNSSEC not very widespread
  - Extreme small DANE dissemination
  - No native browser support

DNS Certification Authority Authorization (CAA)

- Domain owner can add name of used CA into the DNS
- Special DNS resource record: CAA with 3 properties:
  - `issue` / `issuewild`:
    - authorizes the named CA to issue (wildcard) certificates for this (sub-) domain
  - `iodef`:
    - contact information of the domain owner (in case of misuse)
- As of September 2017 CAs must check the CAA record before issuing a certificate (CABForum ballot 187)
- Example

```
> dig google.com CAA
google.com.          21599    IN      CAA     0 issue "pki.goog"
```

CAA – Problems

Domain owner specifies
let's encrypt for this domain

```
> dig cmc.tlsfun.de CAA
tlsfun.de.        3600    IN      CAA     0 iodef "mailto:[illegible]nno@hboeck.de"
tlsfun.de.        3600    IN      CAA     0 iodef "https://int21.de/r/"
tlsfun.de.        3600    IN      CAA     0 issue "letsencrypt.org"
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f5:65:de:3e:35:33:45:ce:da:4a:16:56:58:b8:3a:e1
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 1455)
            commonName                = COMODO RSA Domain Validation Secure Server CA
            organizationName          = COMODO CA Limited
            localityName              = Salford
            stateOrProvinceName       = Greater Manchester
            countryName               = GB
        Validity
            Not Before: Sep  9 00:00:00 2017 GMT
            Not After : Dec  8 23:59:59 2017 GMT
        Subject:
            commonName                = cmc.tlsfun.de
            organizationalUnitName    = Free SSL
            organizationalUnitName    = Domain Control Validated
```

COMODO CA seems
to ignore this entry

CAA – Problems

Lack of CAA checking at Comodo

Hanno Böck via dev-security-policy | Mon, 11 Sep 2017 07:19:47 -0700

Hi,

On saturday I was able to receive a certificate from comodo depsite the subdomain having a CAA record only allowing Let's Encrypt as the CA. Here's the cert:
https://crt.sh/?id=207082245

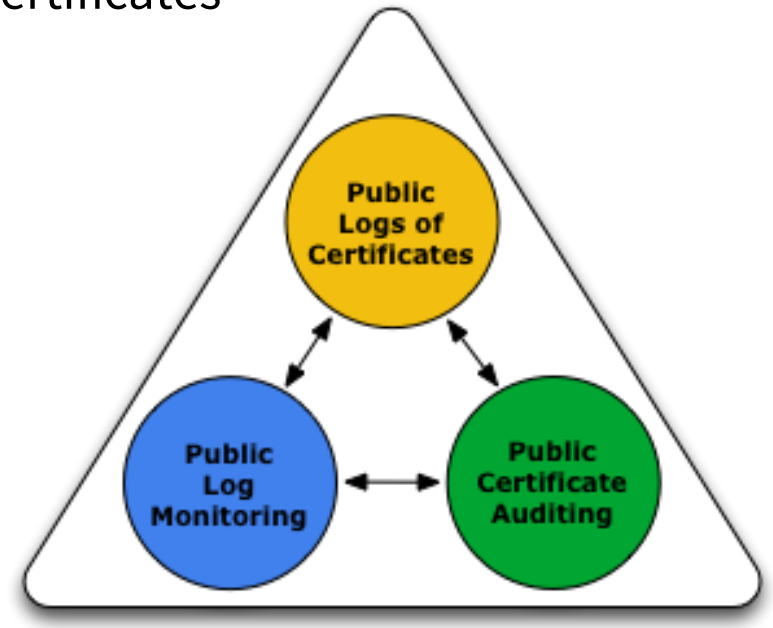
I have by now heard from multiple other people that confirmed the same. Seems right now Comodo isn't checking CAA at all. There's also a bug in the Mozilla bug tracker:
https://bugzilla.mozilla.org/show_bug.cgi?id=1398545

I was originally informed about the lack of CAA checking at Comodo by Michael Kliewe from the mail provider mail.de. However that was before CAA became mandatory. But even back then the Comodo webpage claimed that Comodo would check CAA since at least 12 months:
https://support.comodo.com/index.php?/Knowledgebase/Article/View/1204/1/caa-record---certification-authority-authorization

Source: https://www.mail-archive.com/dev-security-policy@lists.mozilla.org/msg08027.html

Certificate Transparency (CT)

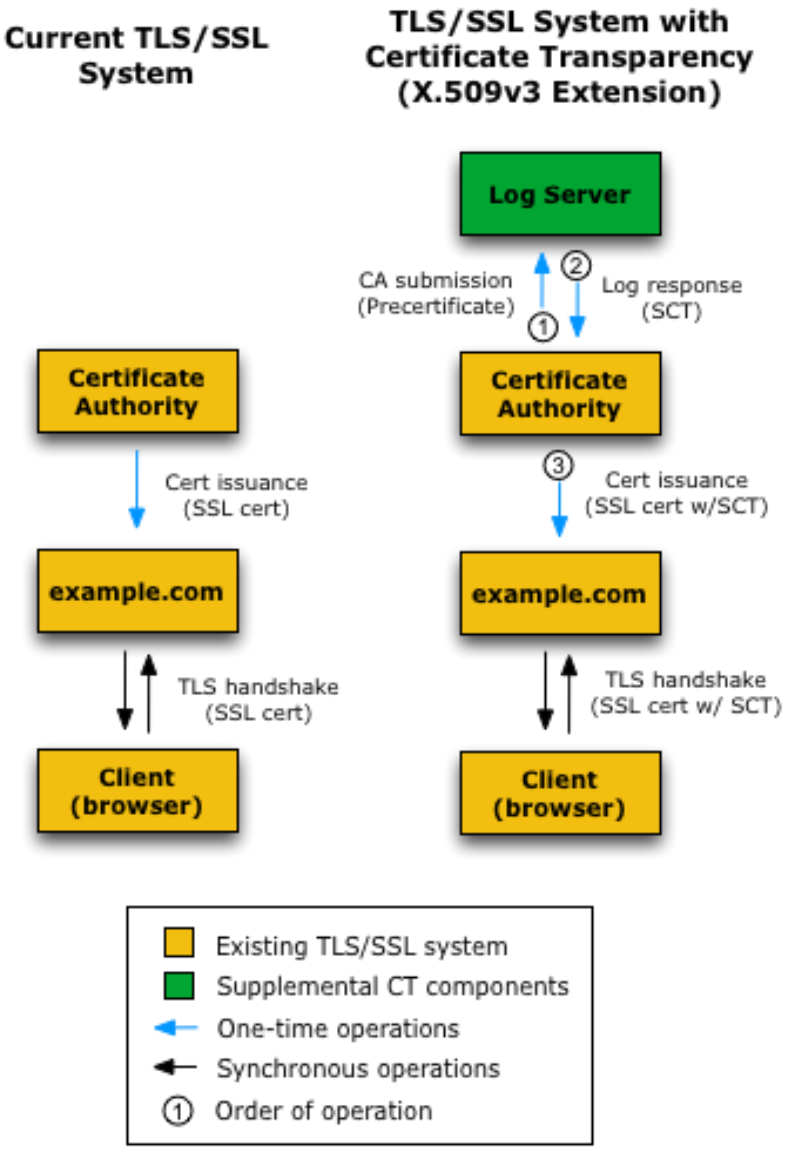
- Idea:
  - All issued certificates are logged into a public append-only log
  - These logs can be monitored and audited by CAs, domain owners and clients
  - Mistakenly or maliciously issued certificates can be detected (not stopped!)



Source: https://www.certificate-transparency.org/

Certificate Transparency

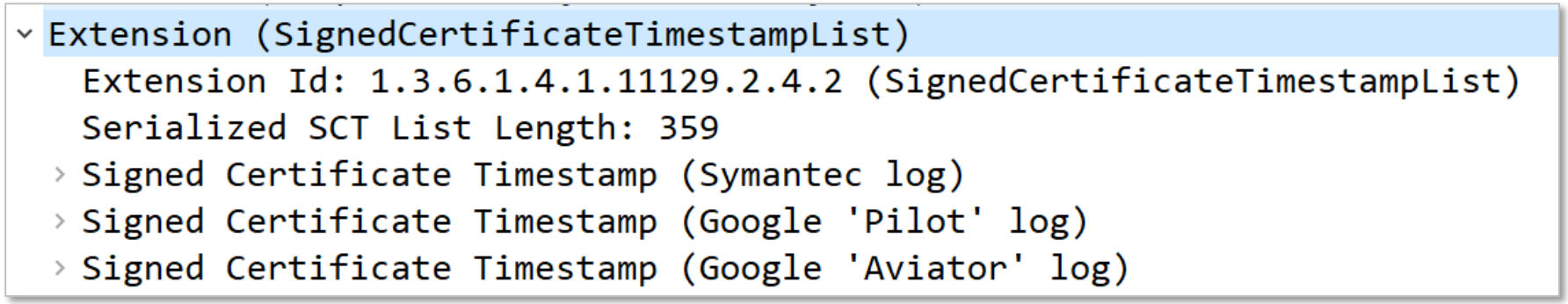
- Typically CAs add newly created certificates to one or more logs
- The log creates a signed certificate timestamp (SCT)
- The SCT can be embedded into the certificate (X.509 extension)
- If the client receives a SCT, he knows that the certificate is included in a CT log



Source: https://www.certificate-transparency.org/

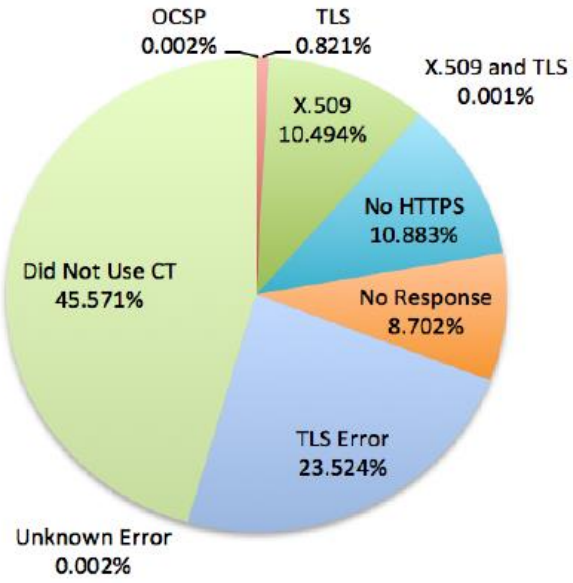
Certificate Transparency

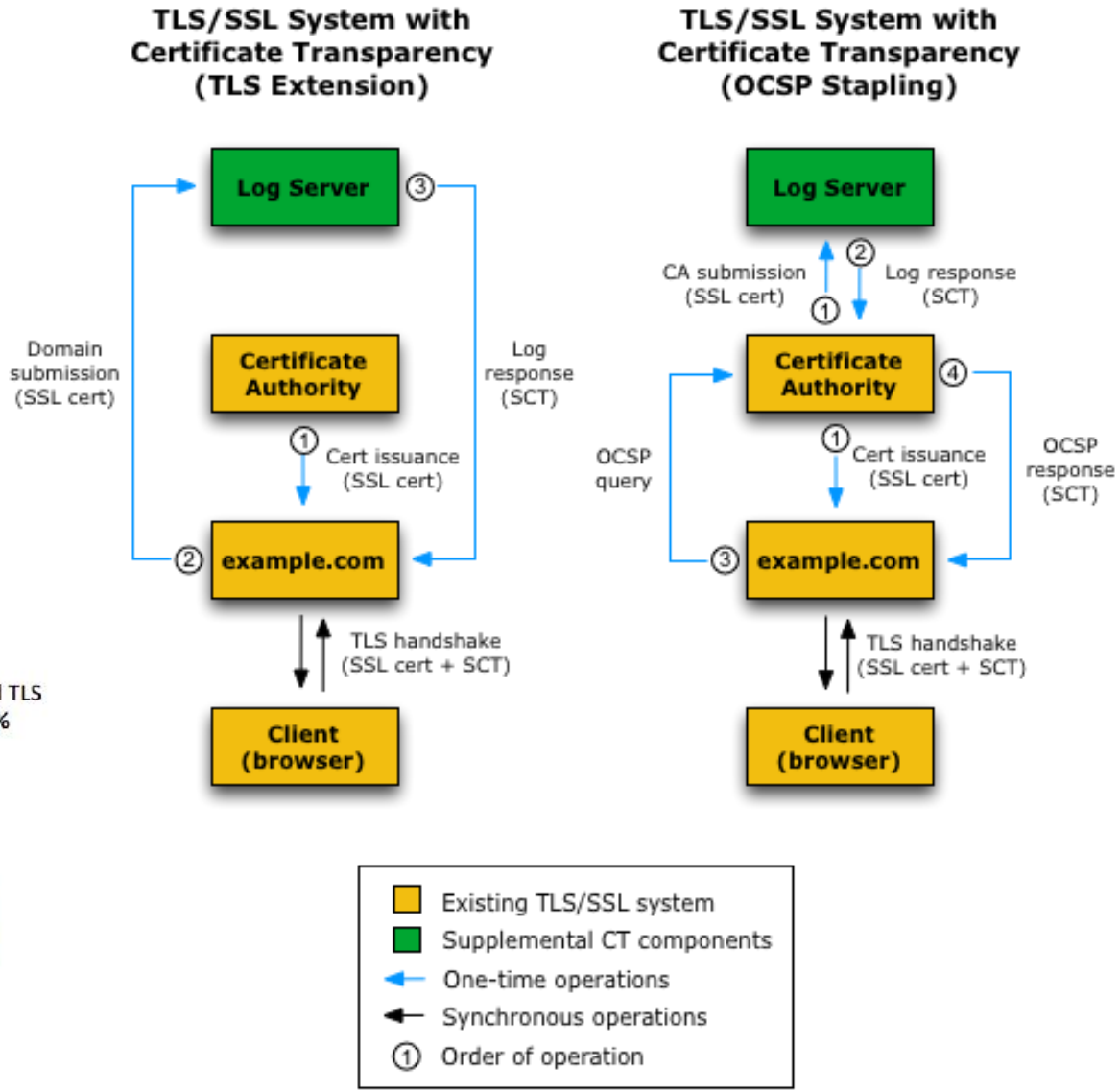
- Example:
  - Signed certificate timestamps from 3 different CT logs inside the X.509 certificate



Certificate Transparency

- Alternative transport options:
  - TLS extension
  - OCSP stabling

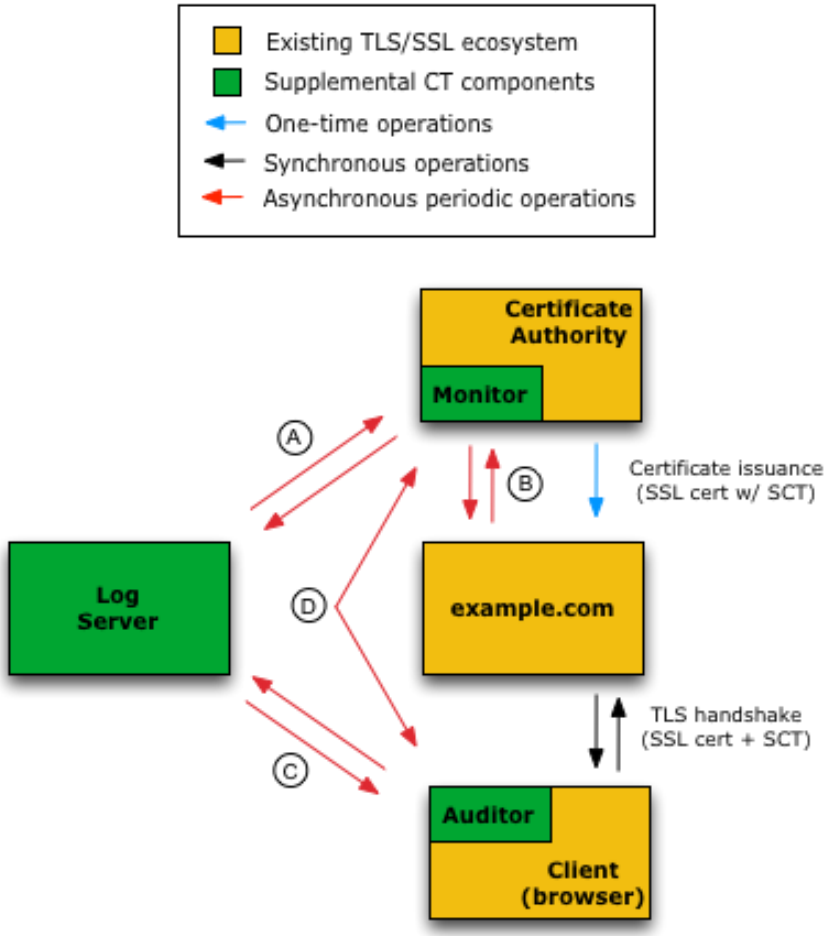




Source: https://www.certificate-transparency.org/
Source: L. Sjöström and C. Nykvist, How Certificate Transparency Impact the Performance. 2017.

Certificate Transparency

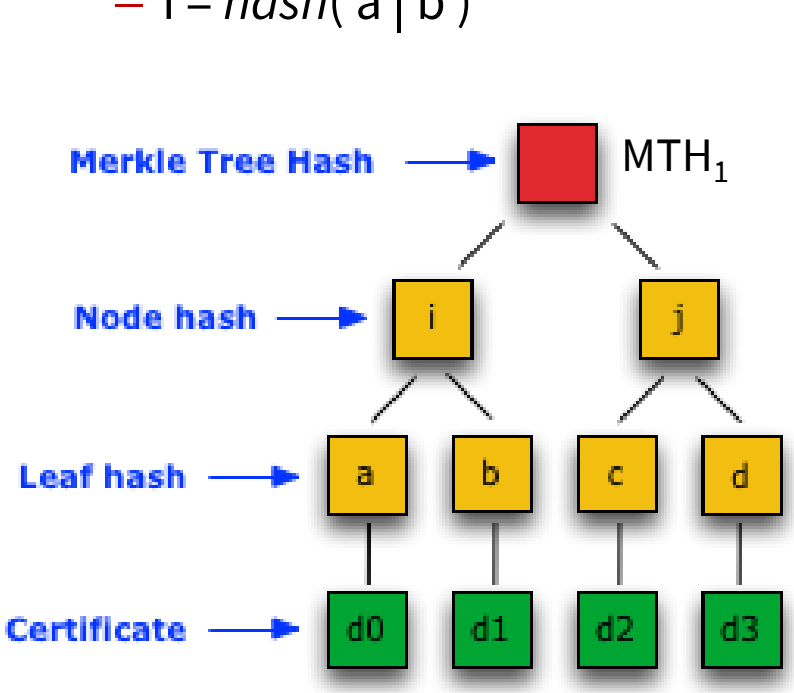
- Sample system configuration
  - A. Monitor watch logs for suspicious certificates
  - B. Certificate owner request logs for their domain
  - C. Auditors verify correct log behaviour
  - D. Monitors and auditor exchange information about logs

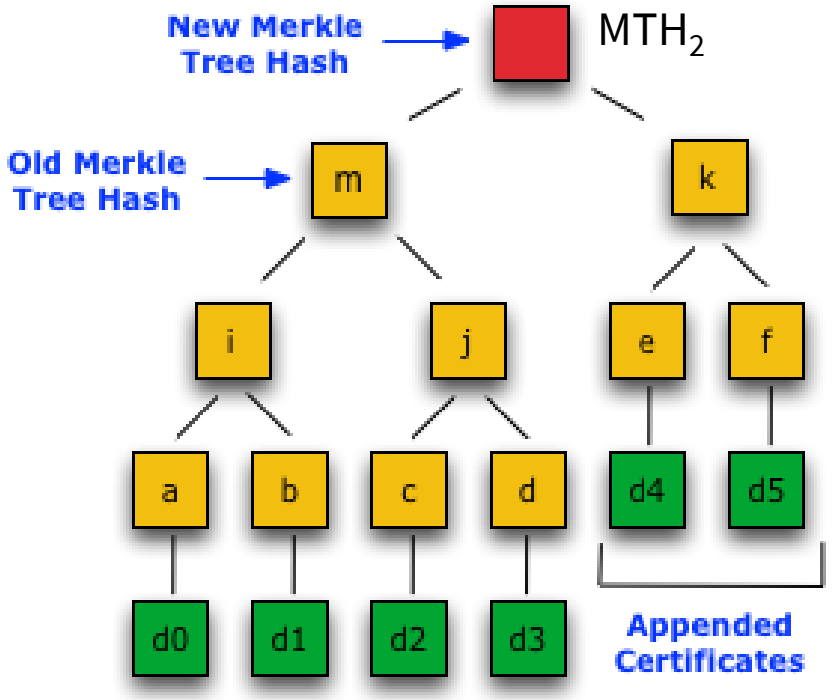


Source: https://www.certificate-transparency.org/

Certificate Transparency

- Certificates are stored at logs in a Merkle tree: every node contains the hash value of its children, e.g.:
  - i = *hash*(a | b)





Source: https://www.certificate-transparency.org/

Certificate Transparency

- The (signed) Merkle tree hash is published by the logs
- Example (Argon 2020):

STH timestamp (UTC)	Tree size	Merkle Tree Hash
2018-04-03 11:55:35	977,001	1qkwHvxlr8591D4cegXlVCu4AzZOzxbChNB1uhV6J2c=
2018-04-03 10:37:03	976,963	+7Vw7lHumD69SpgbHwPvv4UVpGTxCGHExq4WYMG4lGU=
2018-04-03 10:06:33	976,950	ZC1uZQJO8vYUj27rypOmk8MyRoQVNTFyhn98DVSdR/4=

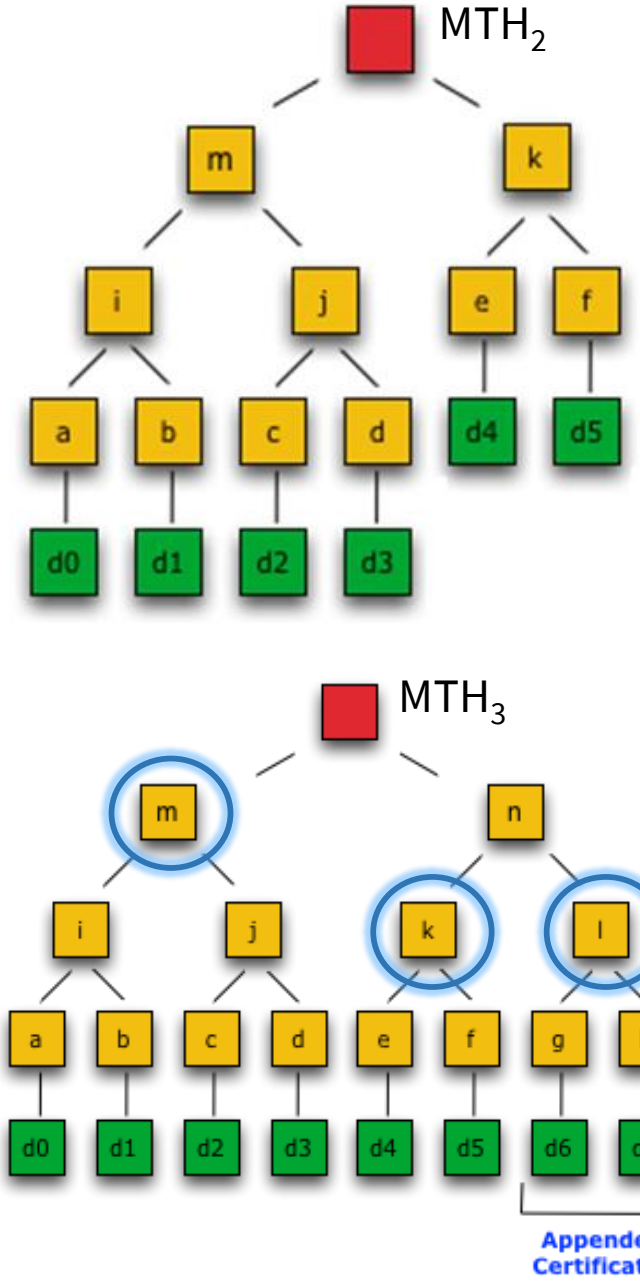
Certificate Transparency

- Logs offer an Web API for accessing its content
- Example (Argon 2020):
  - Request:
    - https://ct.googleapis.com/logs/argon2020/ct/v1/get-sth
  - Response:

```
{
  "tree_size": 977001,
  "timestamp": 1522756535450,
  "sha256_root_hash": "1qkwHvxlr8591D4cegXlVCu4AzZOzxbChNB1uhV6J2c=",
  "tree_head_signature": "BAMASDBGAiEAukAsW4l4EZzDV5t79kQOLpbmoZm2wlBwHda4KNs
                B7DkCIQCNHaltANk7DFOfzIhsu8qtz6ZcB+a0nJ5zPkmx3bty7A=="
}
```

Certificate Transparency

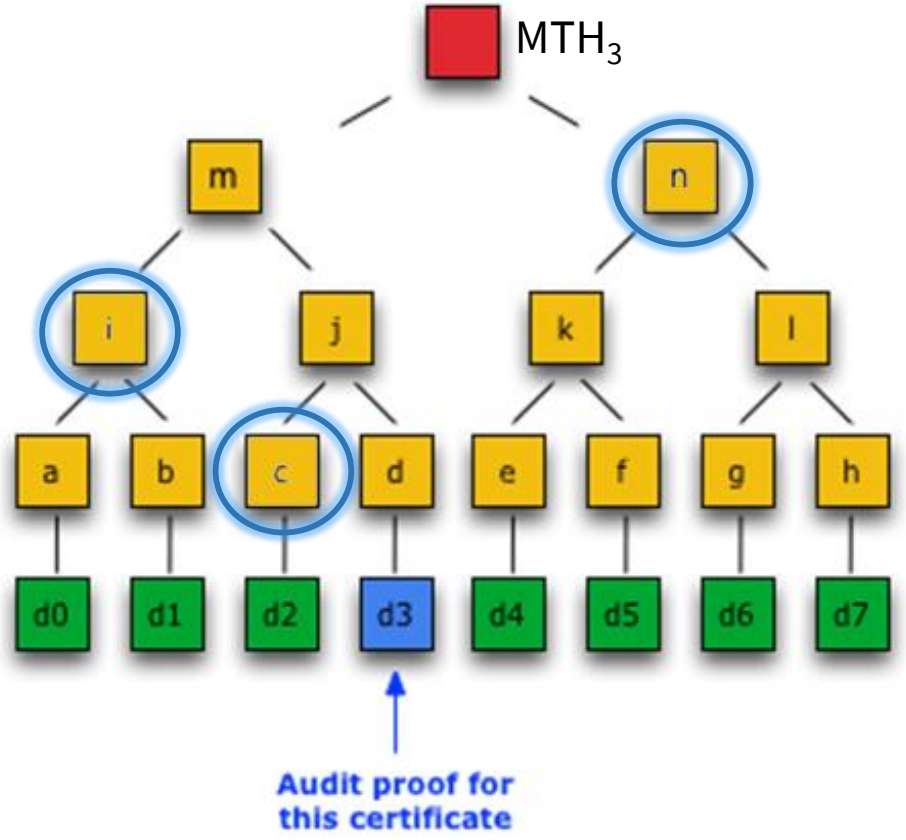
- Merkle consistency proof
  - Is the new tree actually an „extension“ of the old tree?
  - Monitor/Auditor already knows MTH_2 and MTH_3
  - Log sends m, k and l
  - Monitor/Auditor calculates:
    - $MTH_2^* = hash(m, k)$
    - if $MTH_2 = MTH_2^*$ → old tree is unchanged
    - $MTH_3^* = hash(m, hash(k, l))$
    - if $MTH_3 = MTH_3^*$ → new tree is extension of old tree
  - As hash functions are one-way, the proof can not be spoofed by the log



Source: https://www.certificate-transparency.org/

Certificate Transparency

- Merkle audit proof
  - Is d3 actually included in the log?
  - Auditor already knows MTH_3
  - Log sends hashes c, i, n
  - Auditor can calculate d, j, m and MTH_3*
  - Auditor checks if MTH_3* = MTH_3



Source: https://www.certificate-transparency.org/

Example Monitor

Cert Spotter

Centralize your certificate management and monitor for unauthorized certificates using Cert Spotter.

Cert Spotter is watching **1** domain. Edit watch list...

Cert Spotter has discovered **509** unexpired certificates for your domains that were not issued through SSLMate. You have acknowledged every certificate.

There are **896** expired certificates not shown here. Upgrade to a paid plan to view them.

Issuer	Subject	Issue Date	Expiration	
TERENA	*.ezproxy.uio.no ezproxy.uio.no	2018-02-16	2021-02-19	Details Download
TERENA	*.ezproxy-test.uio.no ezproxy-test.uio.no	2018-02-16	2021-02-19	Details Download
TERENA	*.ezproxy.uio.no ezproxy.uio.no	2018-02-15	2021-02-19	Details Download
TERENA	*.ezproxy-test.uio.no ezproxy-test.uio.no	2018-02-15	2021-02-19	Details Download
TERENA	*.ezproxy-test.uio.no ezproxy-test.uio.no	2018-02-15	2021-02-19	Details Download
TERENA	*.ezproxy.uio.no ezproxy.uio.no			
TERENA	wiki-test.uio.no			

Cert Spotter has discovered the following certificate for domain(s) on your watch list:

Issuer: C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
DNS Name: wikitest.ifi.uio.no
Fingerprint: 03272f672916a588240dd7ab1a3ec0663a1e8e67037896543fe1d1c306071f8d
Pubkey hash: 7114b1fdff490eec5bc7604a8f7942e2b0cd27a8841a956994afe53dff58413b
Details: https://sslmate.com/foreign_certs/details/1201696?token=tA8W4jpF7rnudzrlre5k
Download: https://sslmate.com/foreign_certs/download/1201696?token=tA8W4jpF7rnudzrlre5k

Enforcing Certificate Transparency

Google Makes Certificate Transparency Mandatory On Chrome.

/ 1. Announcements / 8. SSL Library / Google Makes Certificate Transparency Mandatory On Chrome.

February 2, 2018 SSL Specialist 1. Announcements/ 8. SSL Library

0 0

Last year Google once again flexed its muscles by announcing the requirement for Certificate Transparency for all new SSL/TLS certificates in October 2017. This has since been pushed back until April 2018.

This requirement means that Chrome will no longer trust new SSL/TLS certificates that are not qualified for Certificate Transparency (CT). CT is

Source: https://www.sslsupportdesk.com/google-makes-certificate-transparency-mandatory-chrome/

Certificate Transparency

- Advantages:
  - Simple overview of all issued certificates
  - Fast detection of mis-issued certificated and sloppy/rogue CAs
  - If one log is not available, other logs can be requested
- Disadvantages:
  - No mechanism for revocation of mis-issued certificates
  - Logs might become large and slow
  - Logs reveal (sub) domain names
  - If the client access a log, the log might learn the users access pattern
  - If the client finds a missing certificate it is supposed to publish the log misbehavior → user's privacy of the user at risk

Summary

- Certificates are essential for TLS and for a “more secure Web”
- A single unreliable or untrustworthy certificate authority can endanger the whole Web PKI
- Still, no secure and practical solution is available
- Also unclear: who is responsible …
- Certificate transparency is the current candidate favored by the browser vendors
- Current research:
  - Certificate revocation for CT logs
  - Efficient log implementation
  - Privacy conserving log management

References

- J. Gustafsson, G. Overier, M. Arlitt, and N. Carlsson, “A First Look at the CT Landscape: Certificate Transparency Logs in Practice,” in *Passive and Active Measurement*, 2017, pp. 87–99.
- Google, “Certificate Transparency,” 2018. [Online]. Available: https://www.certificate-transparency.org/. [Accessed: 22-Feb-2018].
- S. Eskandarian, E. Messeri, J. Bonneau, and D. Boneh, “Certificate Transparency with Privacy,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 329–344, 2017.
- K. Borgolte, T. Fiebig, S. Hao, C. Kruegel, and G. Vigna, “Cloud Strife: Mitigating the Security Risks of Domain-Validated Certificates,” in *Proc. Internet Society Symposium on Network and Distributed System Security (NDSS)*, 2018.
- “Dell Computers Contain CA Root Certificate Vulnerability.” [Online]. Available: https://www.us-cert.gov/ncas/current-activity/2015/11/24/Dell-Computers-Contain-CA-Root-Certifcate-Vulnerability. [Accessed: 25-Feb-2018].
- L. Sjöström and C. Nykvist, *How Certificate Transparency Impact the Performance*. 2017.
- C. Jackson, A. Barth, and J. Hodges, “HTTP Strict Transport Security (HSTS),” 2012. [Online]. Available: https://tools.ietf.org/html/rfc6797. [Accessed: 23-Feb-2018].
- Nettrack, “SSL Issuer Popularity,” *NetTrack - Anonymous Web Statistics*, 2018. [Online]. Available: https://nettrack.info//ssl_certificate_issuers.html. [Accessed: 22-Feb-2018].
- H. Böck, “The Problem with OCSP Stapling and Must Staple and why Certificate Revocation is still broken - Hanno’s blog,” 2017. [Online]. Available: https://blog.hboeck.de/archives/886-The-Problem-with-OCSP-Stapling-and-Must-Staple-and-why-Certificate-Revocation-is-still-broken.html. [Accessed: 22-Feb-2018].
- Google, “Transparency Report,” 2018. [Online]. Available: https://transparencyreport.google.com/. [Accessed: 22-Feb-2018].
- D. Eastlake, “Transport Layer Security (TLS) Extensions: Extension Definitions,” 2011. [Online]. Available: https://tools.ietf.org/html/rfc6066. [Accessed: 23-Feb-2018].
- S. Galperin, S. Santesson, M. Myers, A. Malpani, and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” 2013. [Online]. Available: https://tools.ietf.org/html/rfc6960. [Accessed: 23-Feb-2018].
- P. Hallam-Baker, “X.509v3 Transport Layer Security (TLS) Feature Extension,” 2015. [Online]. Available: https://tools.ietf.org/html/rfc7633. [Accessed: 23-Feb-2018].
- R. Dahlberg, T. Pulls, and R. Peeters, “Efficient Sparse Merkle Trees,” in *Secure IT Systems*, 2016, pp. 199–215.