Certificate Validation in TLS: Challenges and Emerging Trends

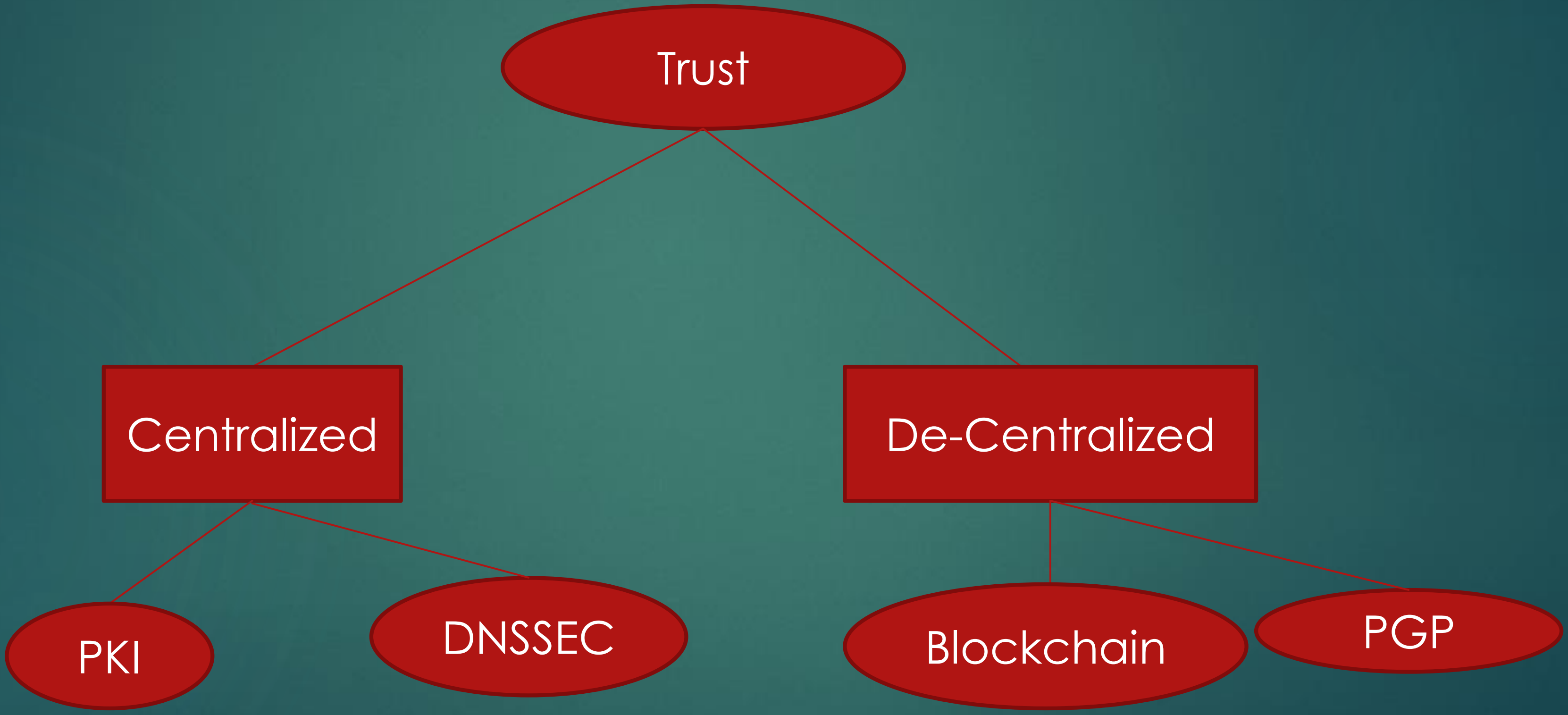
DR. BALAJI RAJENDRAN
PRINCIPAL TECHNICAL OFFICER
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING
NO.68, ELECTRONICS CITY, BANGALORE

IIESOC CONNECTIONS @ INFOSYS, BANGALORE
8TH NOVEMBER 2017

Agenda

- Electronic Trust Models
- CA, CRL, Types of Certificates in TLS
- Validation for Certificate Issuance
- RFCs – ACME
- Certificate Validation Algorithm
- Certificate Transparency and RFCs
- Summary

Approaches to Establish Electronic Trust



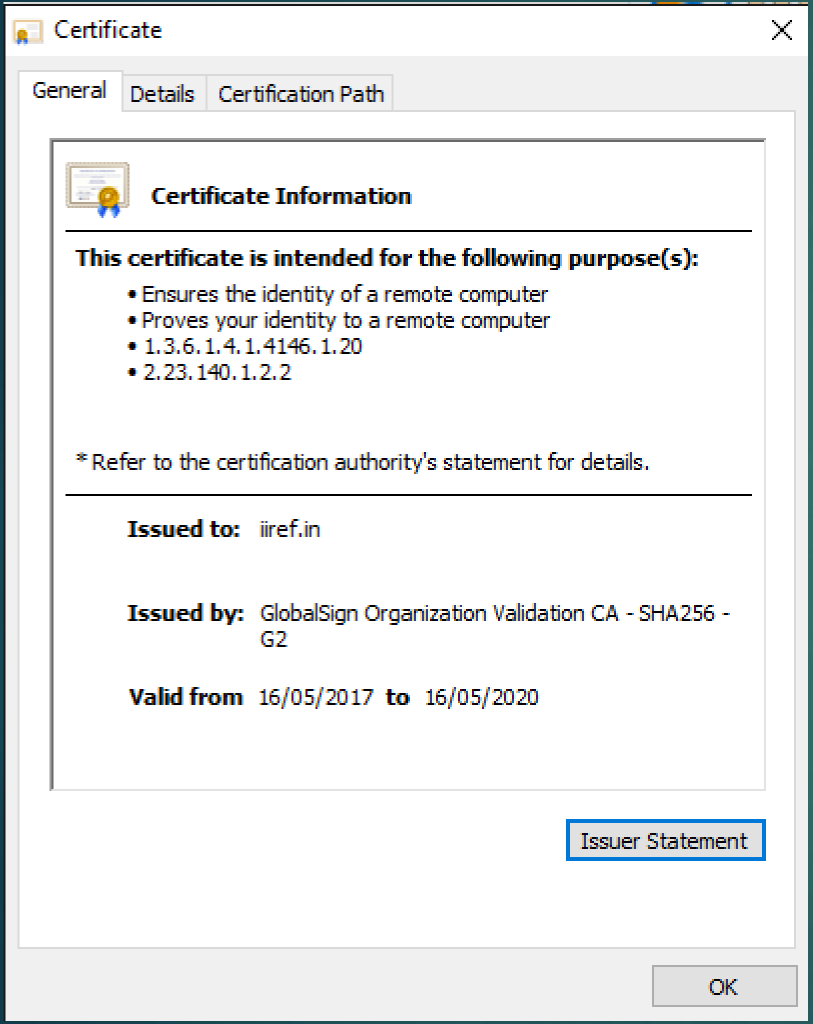
Certifying Authority (CA)

- Certifying authority is an entity which issues Digital Signature Certificate **(DSC)**
- It is a **trusted third party**
- CA's are the important components of Public Key Infrastructure (PKI)

Responsibilities of CA

- Verify the credentials of the entity requesting for the certificate (RA's responsibility)
- Issue certificates
- Revoke certificate
- Generate and upload CRL

Sample Certificate



Certificate Information

This certificate is intended for the following purpose(s):

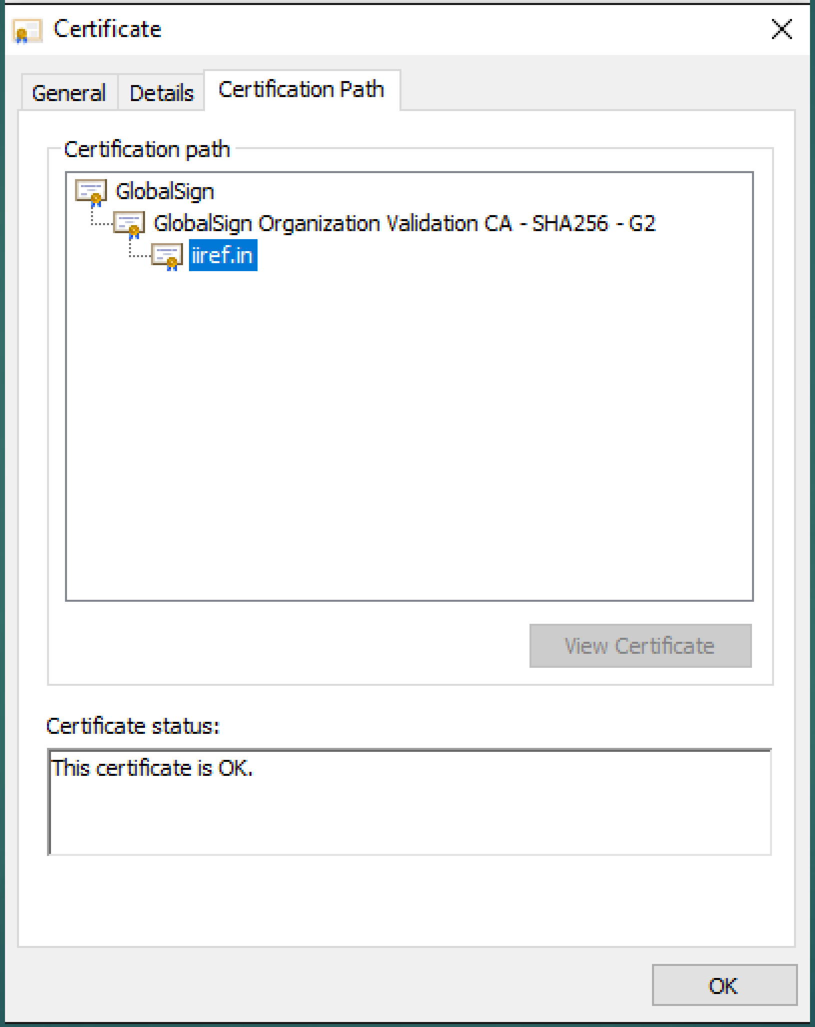
- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 1.3.6.1.4.1.4146.1.20
- 2.23.140.1.2.2

* Refer to the certification authority's statement for details.

Issued to: iiref.in

Issued by: GlobalSign Organization Validation CA - SHA256 - G2

Valid from 16/05/2017 **to** 16/05/2020

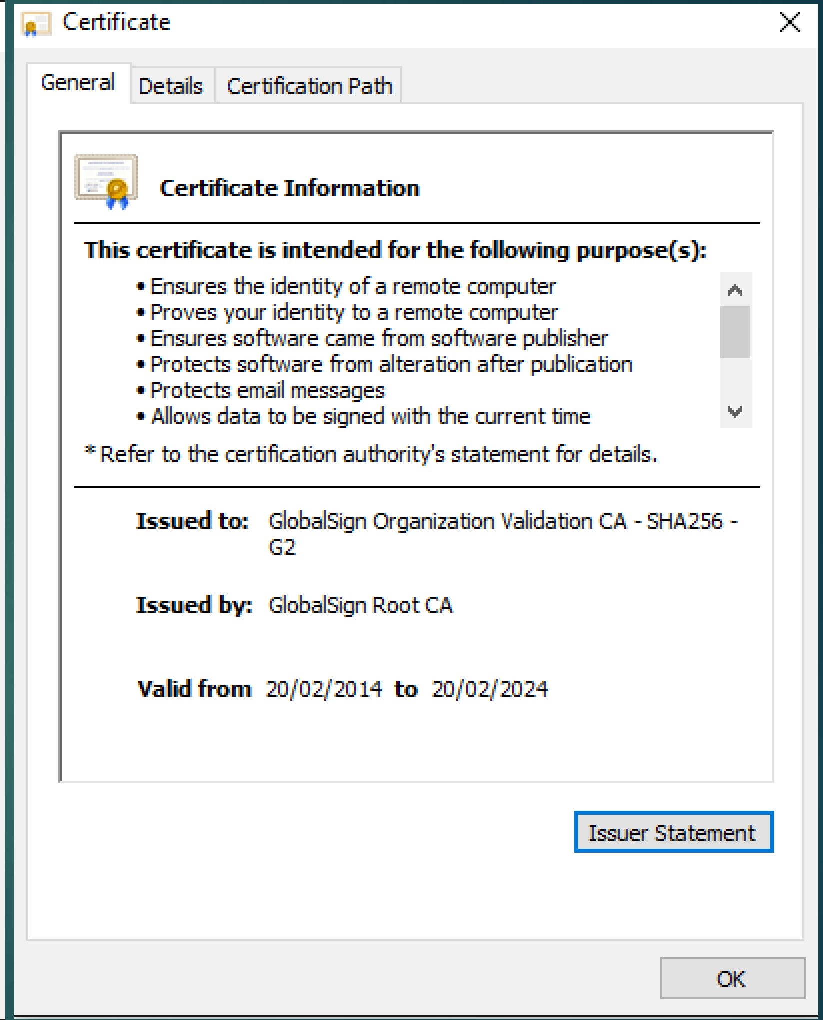


Certification path

- GlobalSign
  - GlobalSign Organization Validation CA - SHA256 - G2
    - iiref.in

Certificate status:

This certificate is OK.



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- Ensures software came from software publisher
- Protects software from alteration after publication
- Protects email messages
- Allows data to be signed with the current time

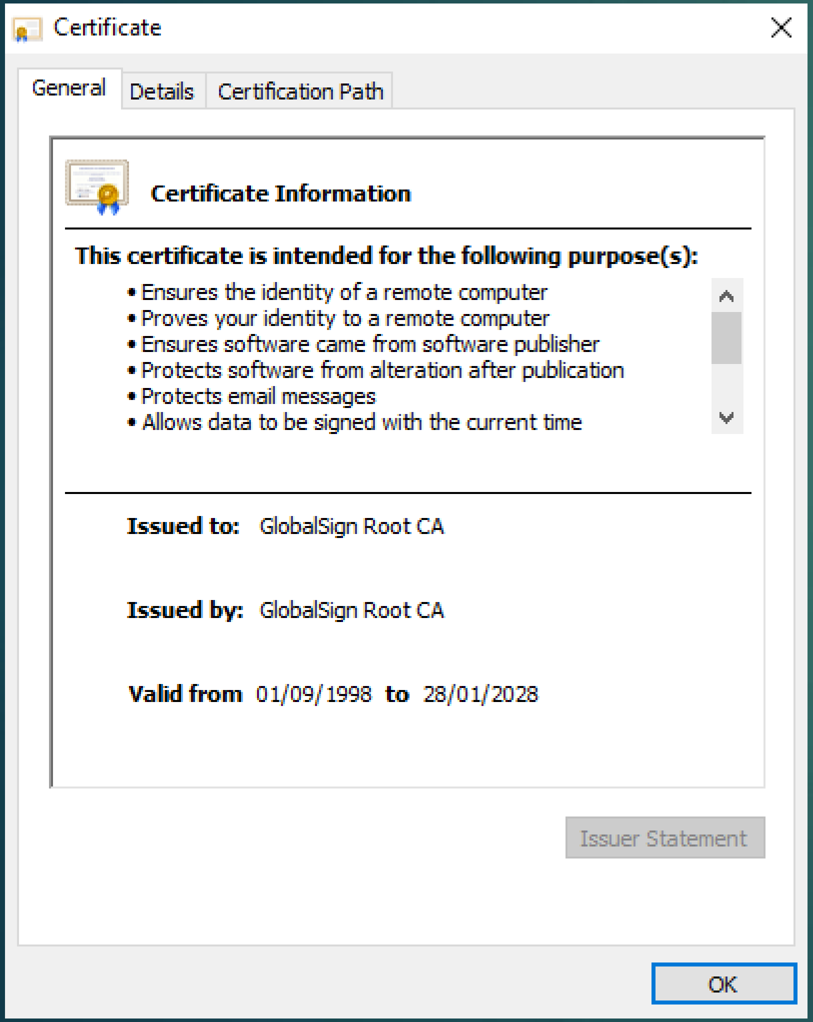
* Refer to the certification authority's statement for details.

Issued to: GlobalSign Organization Validation CA - SHA256 - G2

Issued by: GlobalSign Root CA

Valid from 20/02/2014 **to** 20/02/2024

Sample Certificate and CRL



Certificate Information

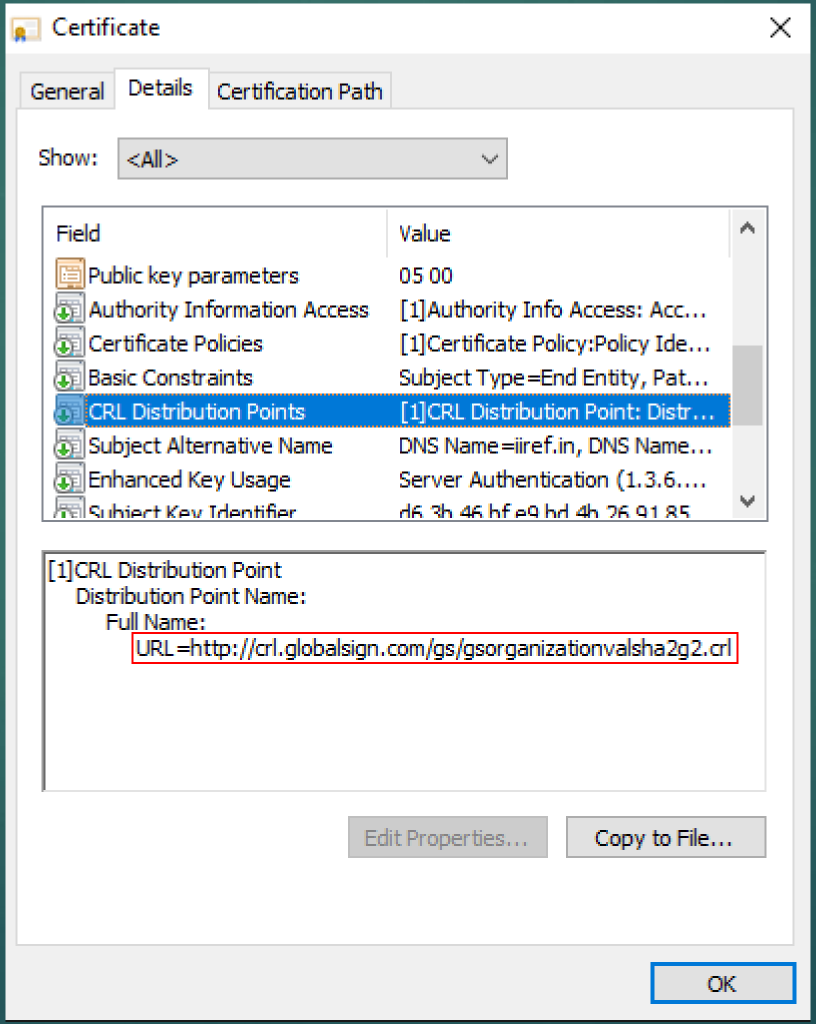
This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- Ensures software came from software publisher
- Protects software from alteration after publication
- Protects email messages
- Allows data to be signed with the current time

Issued to: GlobalSign Root CA

Issued by: GlobalSign Root CA

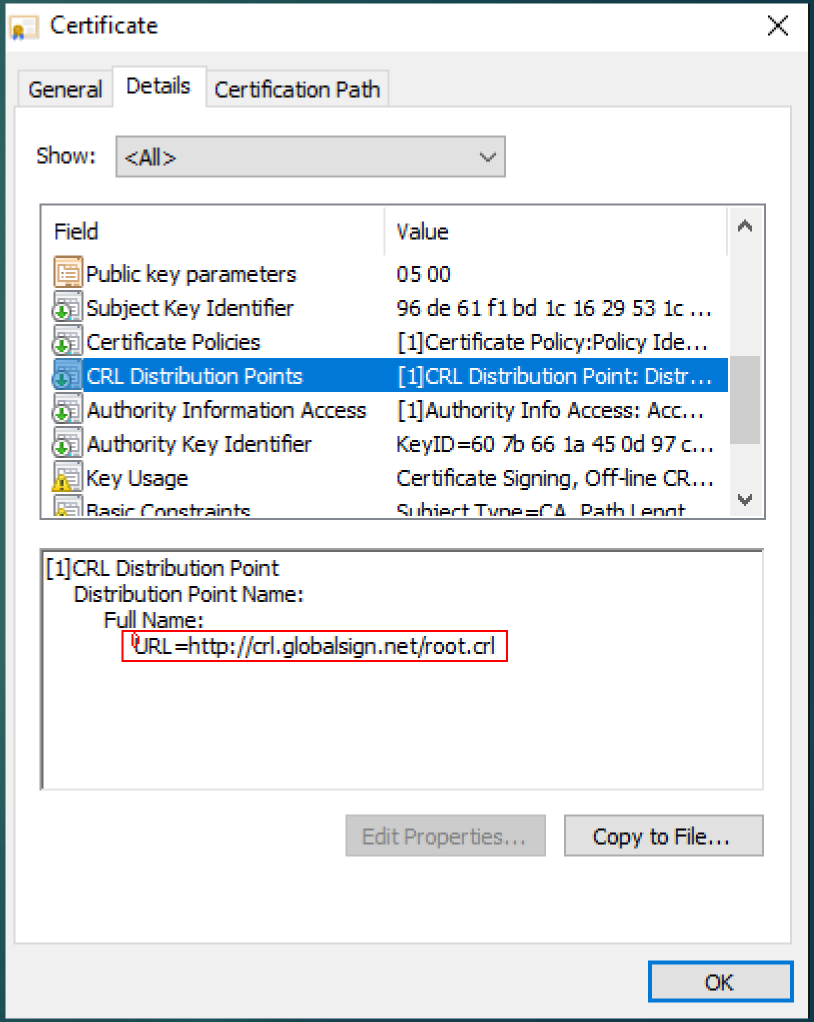
Valid from 01/09/1998 **to** 28/01/2028



Show: <All>

Field	Value
Public key parameters	05 00
Authority Information Access	[1]Authority Info Access: Acc...
Certificate Policies	[1]Certificate Policy:Policy Ide...
Basic Constraints	Subject Type=End Entity, Pat...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Subject Alternative Name	DNS Name=iiref.in, DNS Name...
Enhanced Key Usage	Server Authentication (1.3.6....
Subject Key Identifier	d6 3b 46 bf e9 bd 4b 26 91 85

```
[1]CRL Distribution Point
     Distribution Point Name:
          Full Name:
               URL=http://crl.globalsign.com/gs/gsorganizationvalsha2g2.crl
```



Show: <All>

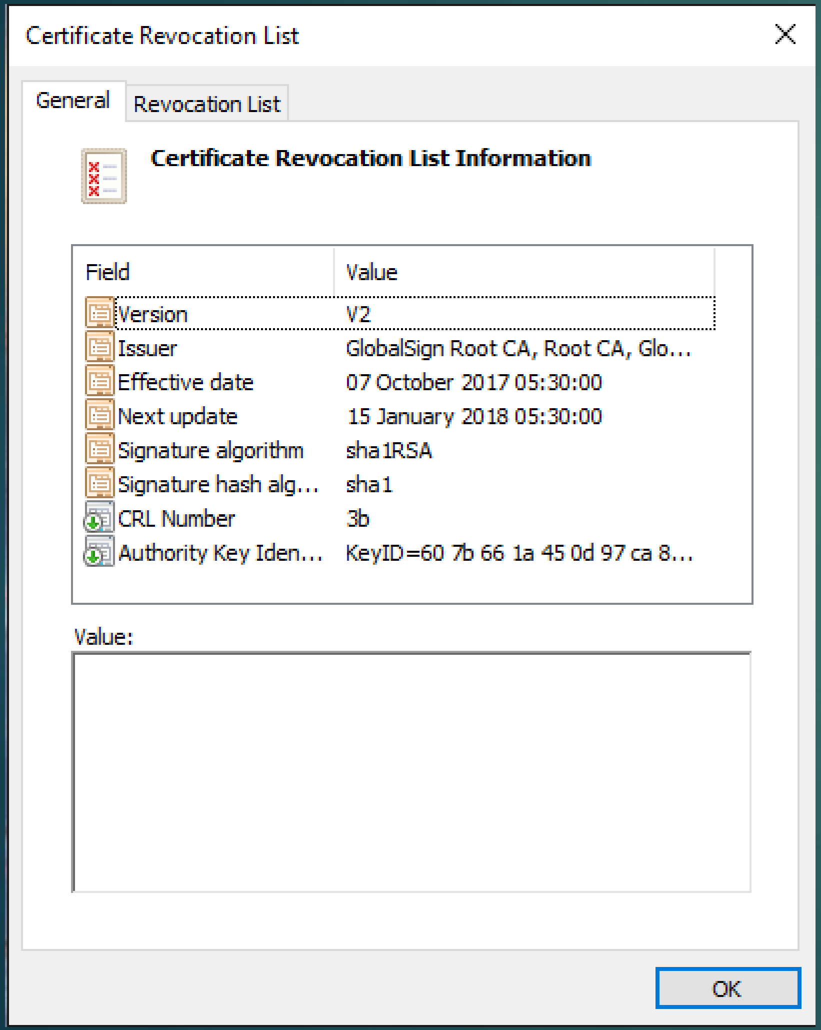
Field	Value
Public key parameters	05 00
Subject Key Identifier	96 de 61 f1 bd 1c 16 29 53 1c ...
Certificate Policies	[1]Certificate Policy:Policy Ide...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Information Access	[1]Authority Info Access: Acc...
Authority Key Identifier	KeyID=60 7b 66 1a 45 0d 97 c...
Key Usage	Certificate Signing, Off-line CR...
Basic Constraints	Subject Type=CA, Path Lengt

```
[1]CRL Distribution Point
     Distribution Point Name:
          Full Name:
               URL=http://crl.globalsign.net/root.crl
```

CRL – Certificate Revocation List

- A list containing the serial number of those certificates that have been revoked by a particular CA
  - CRL is digitally signed by CA;
  - Maintained by the CA's
- Why they have been revoked?
  - If keys are compromised and users reports to the CA
  - If CA discovers, false information being used to obtain the certificate
- How frequently the CRL is updated ?
  - Generally twice a day; based on CA's policies
- Is there any automated system in place for accessing the CRL?
  - OCSP

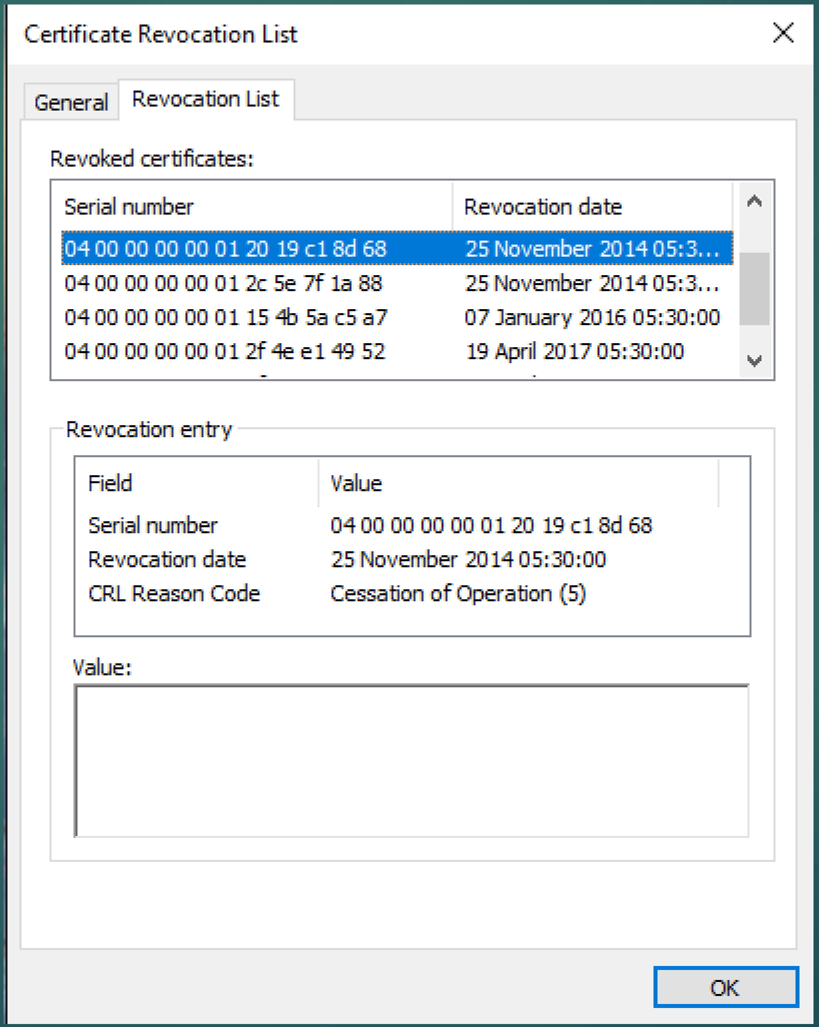
CRL



Certificate Revocation List Information

Field	Value
Version	V2
Issuer	GlobalSign Root CA, Root CA, Glo...
Effective date	07 October 2017 05:30:00
Next update	15 January 2018 05:30:00
Signature algorithm	sha1RSA
Signature hash alg...	sha1
CRL Number	3b
Authority Key Iden...	KeyID=60 7b 66 1a 45 0d 97 ca 8...

Value:



Revoked certificates:

Serial number	Revocation date
04 00 00 00 00 01 20 19 c1 8d 68	25 November 2014 05:3...
04 00 00 00 00 01 2c 5e 7f 1a 88	25 November 2014 05:3...
04 00 00 00 00 01 15 4b 5a c5 a7	07 January 2016 05:30:00
04 00 00 00 00 01 2f 4e e1 49 52	19 April 2017 05:30:00

Revocation entry

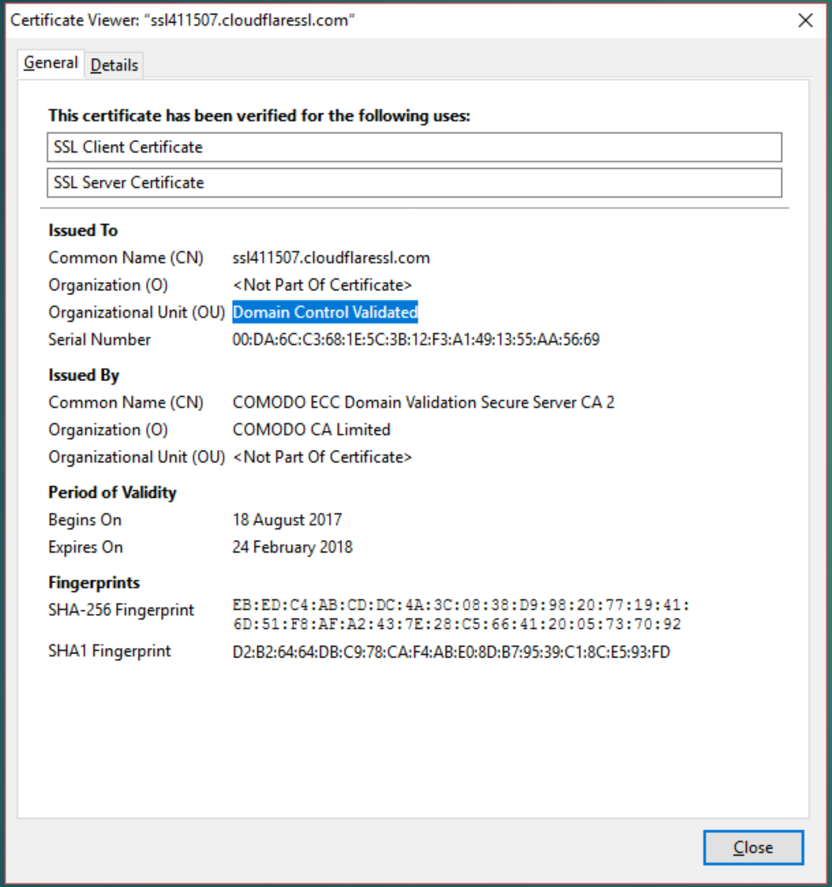
Field	Value
Serial number	04 00 00 00 00 01 20 19 c1 8d 68
Revocation date	25 November 2014 05:30:00
CRL Reason Code	Cessation of Operation (5)

Value:

Types of Certificates

- Based on Business requirement
  - Multi-domain Certificate
  - Wild Card Certificate
- Based on Validation
  - Domain Validated (DV) Certificates
  - Organization Validated (OV) Certificates
  - Extended Validation (EV) Certificates

Sample DV Certificate



Certificate Viewer: "ssl411507.cloudflaressl.com"

This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	ssl411507.cloudflaressl.com
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	Domain Control Validated
Serial Number	00:DA:6C:C3:68:1E:5C:3B:12:F3:A1:49:13:55:AA:56:69

Issued By

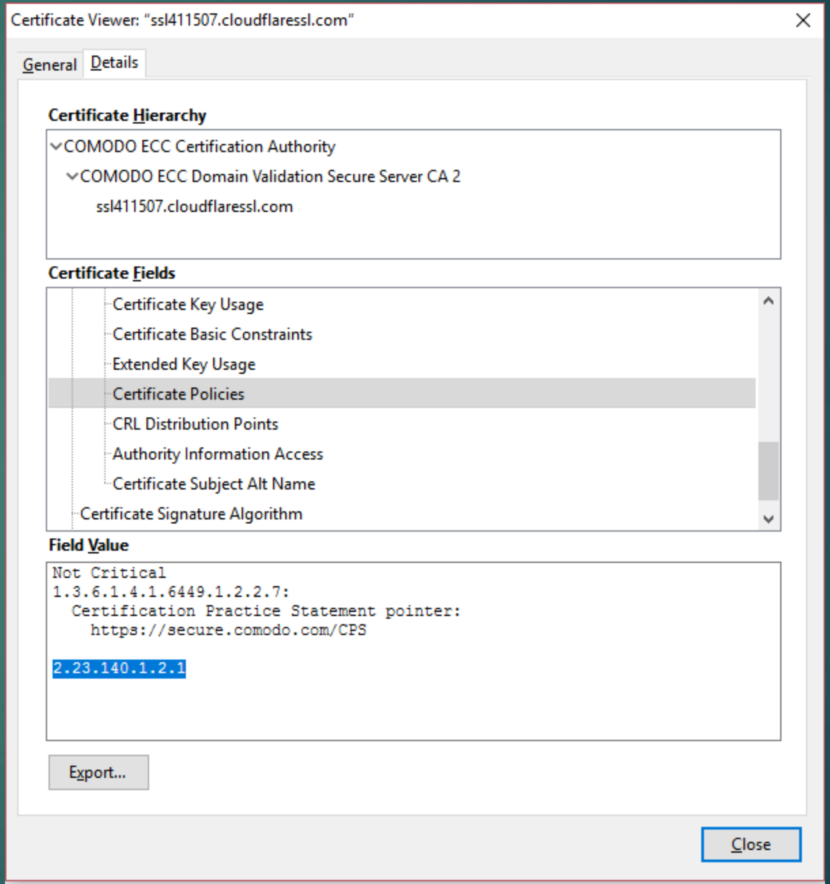
Common Name (CN)	COMODO ECC Domain Validation Secure Server CA 2
Organization (O)	COMODO CA Limited
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	18 August 2017
Expires On	24 February 2018

Fingerprints

SHA-256 Fingerprint	EB:ED:C4:AB:CD:DC:4A:3C:08:38:D9:98:20:77:19:41: 6D:51:F8:AF:A2:43:7E:28:C5:66:41:20:05:73:70:92
SHA1 Fingerprint	D2:B2:64:64:DB:C9:78:CA:F4:AB:E0:8D:B7:95:39:C1:8C:E5:93:FD



Certificate Viewer: "ssl411507.cloudflaressl.com"

Certificate Hierarchy

- COMODO ECC Certification Authority
  - COMODO ECC Domain Validation Secure Server CA 2
    - ssl411507.cloudflaressl.com

Certificate Fields

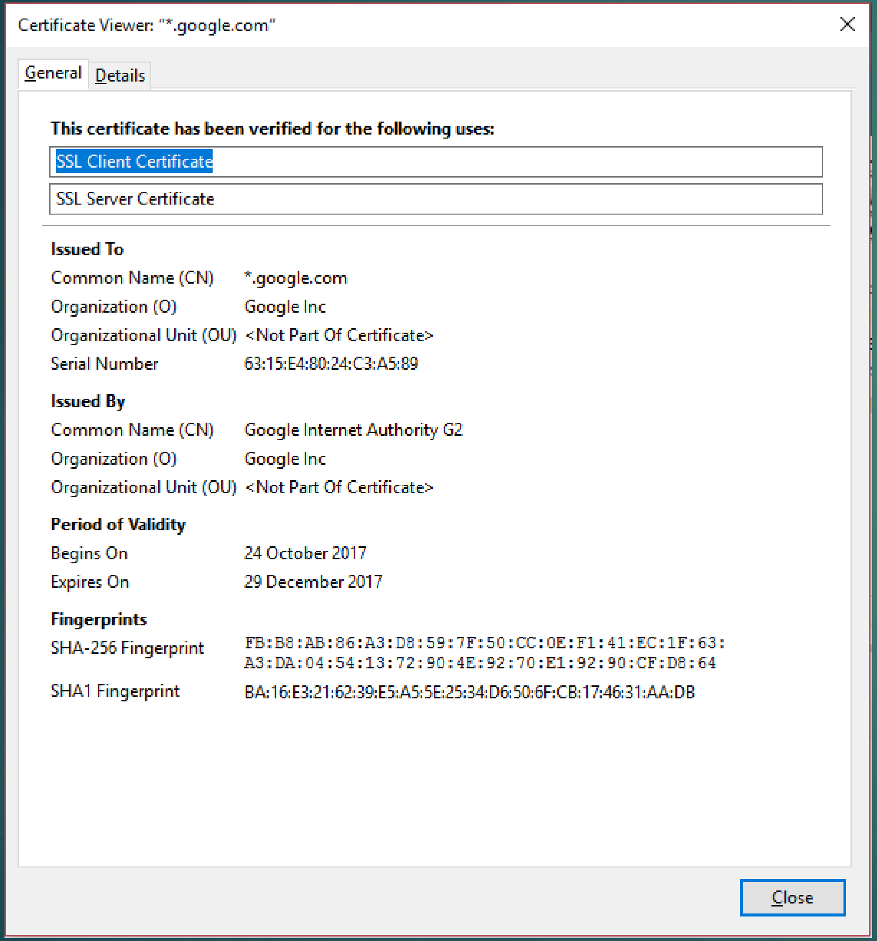
- Certificate Key Usage
- Certificate Basic Constraints
- Extended Key Usage
- Certificate Policies
- CRL Distribution Points
- Authority Information Access
- Certificate Subject Alt Name
- Certificate Signature Algorithm

Field Value

```
Not Critical
1.3.6.1.4.1.6449.1.2.2.7:
  Certification Practice Statement pointer:
    https://secure.comodo.com/CPS

2.23.140.1.2.1
```

Sample OV Certificate



Certificate Viewer: "*.google.com"

This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	*.google.com
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	63:15:E4:80:24:C3:A5:89

Issued By

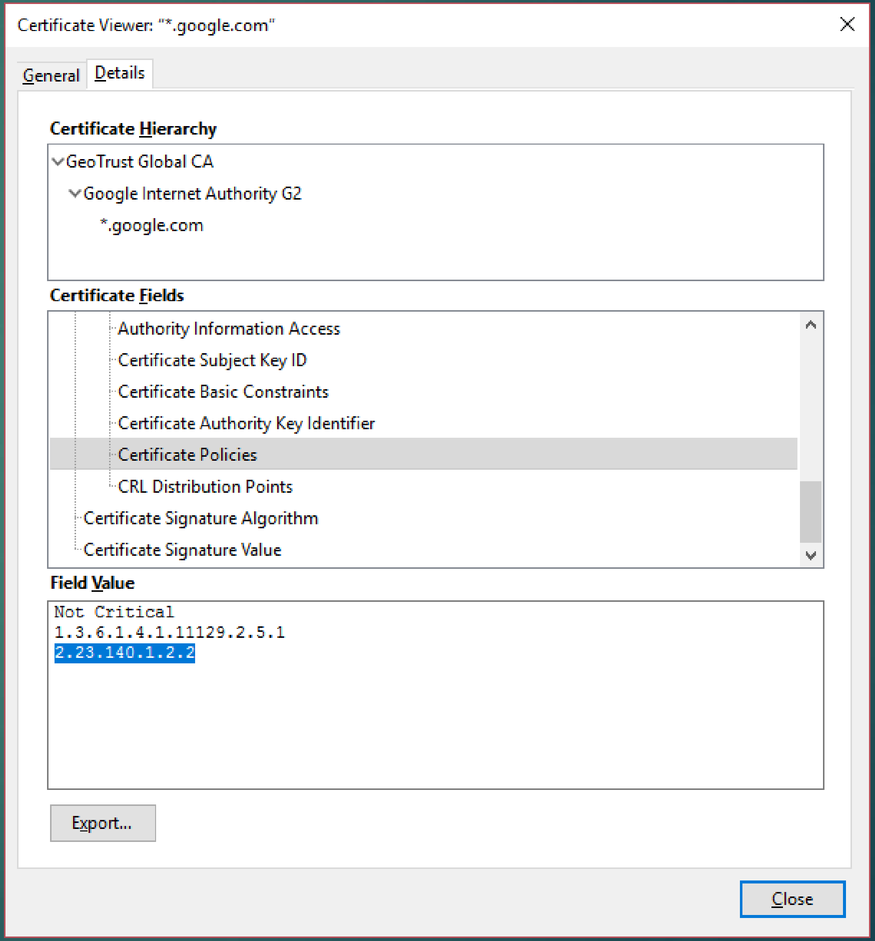
Common Name (CN)	Google Internet Authority G2
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	24 October 2017
Expires On	29 December 2017

Fingerprints

SHA-256 Fingerprint	FB:B8:AB:86:A3:D8:59:7F:50:CC:0E:F1:41:EC:1F:63: A3:DA:04:54:13:72:90:4E:92:70:E1:92:90:CF:D8:64
SHA1 Fingerprint	BA:16:E3:21:62:39:E5:A5:5E:25:34:D6:50:6F:CB:17:46:31:AA:DB



Certificate Viewer: "*.google.com"

Certificate Hierarchy

- GeoTrust Global CA
  - Google Internet Authority G2
    - *.google.com

Certificate Fields

- Authority Information Access
- Certificate Subject Key ID
- Certificate Basic Constraints
- Certificate Authority Key Identifier
- Certificate Policies
- CRL Distribution Points
- Certificate Signature Algorithm
- Certificate Signature Value

Field Value

```
Not Critical
1.3.6.1.4.1.11129.2.5.1
2.23.140.1.2.2
```

Process of Obtaining DV Certificate

- Generate a PKCS#10 [RFC2986] Certificate Signing Request (CSR).
- Cut-and-paste the CSR into a CA web page.
- Prove ownership of the domain by one of the following methods:
  - Put a CA-provided challenge at a specific place on the web server.
  - Put a CA-provided challenge at a DNS location corresponding to the target domain.
  - Receive CA challenge at a (hopefully) administrator-controlled email address corresponding to the domain and then respond to it on the CA's web page.
- Download the issued certificate and install it on their Web Server.

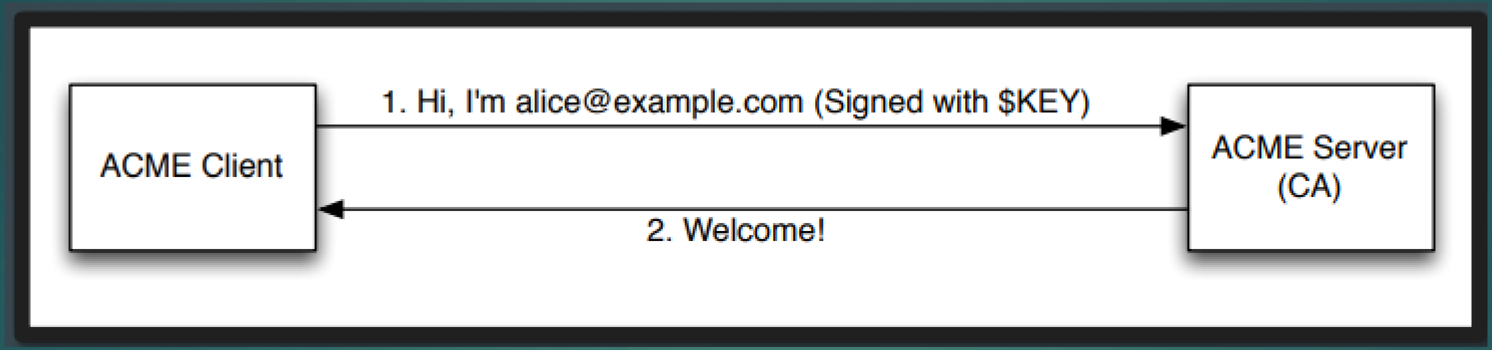
Few incidents …

- Comodo
  - Exploiting the CA process to issue bogus certificates
  - March 2011, 9 bogus certificates were issued based on request coming from Iran
- DigiNotar
  - A dutch CA had to close its business owing to exploitation of its infrastructure
    - July 2011, an attacker issued a Wildcard Certificate for Google!
    - Around 500 fake Diginotar certificates were found to be issued
    - All browsers started to remove DigiNotar from their trust stores
    - As a result of this, one of the sub-CA of DigiNotar that was issuing certificates to Dutch Government also was affected

Automatic Certificate Management Environment (ACME)

- Working Group: ACME
- draft-ietf-acme-acme-07
  - Authors: Richard Barnes, Hoffman-Andrews, Kasten
- Proposes to automate the process of verification of domain names (as given by applicant) by the CA for DV Certificates
  - Also proposes to automate the process of Certificate Issuance and Revocation
- Designed as a REST application

ACME – Explained …

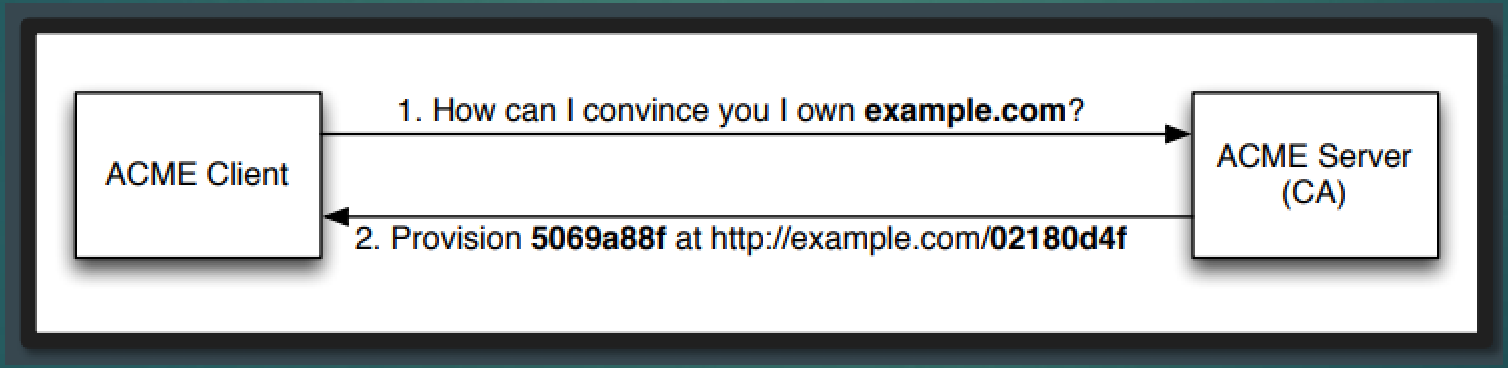


ACME Client

1. Hi, I'm alice@example.com (Signed with $KEY)

ACME Server (CA)

2. Welcome!



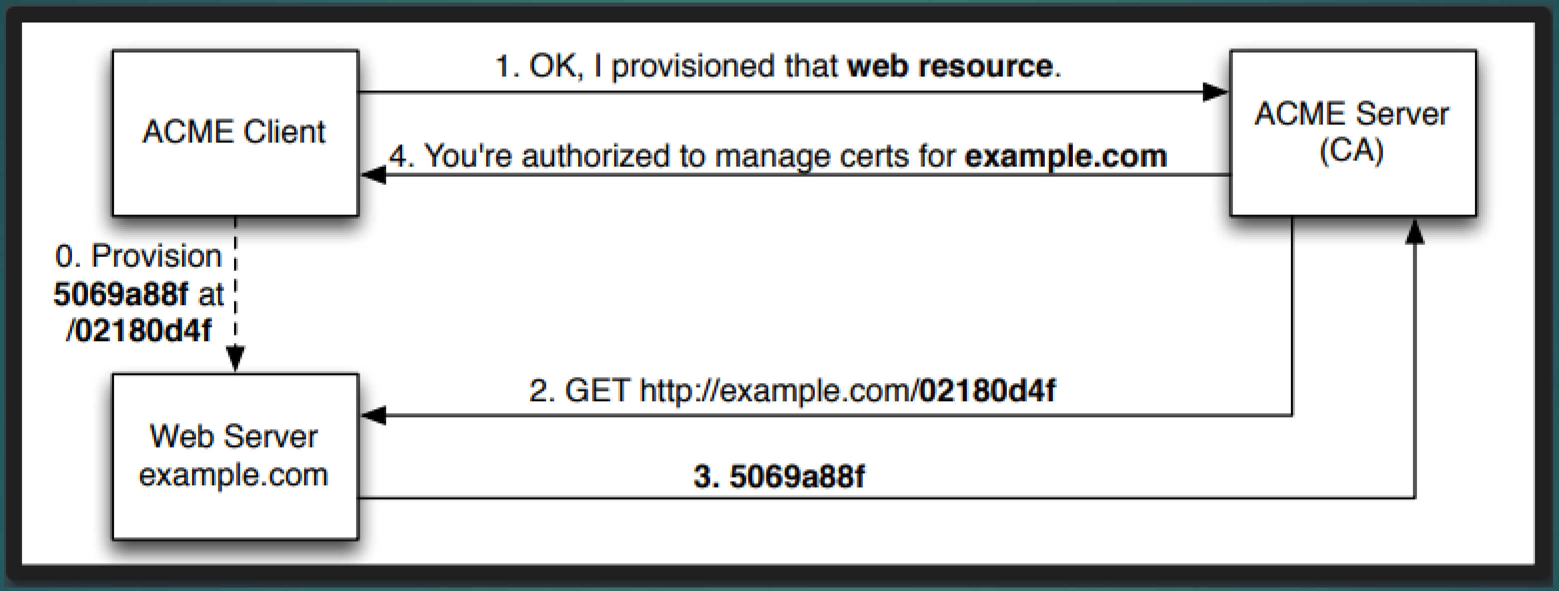
ACME Client

1. How can I convince you I own **example.com**?

ACME Server (CA)

2. Provision **5069a88f** at http://example.com/**02180d4f**

ACME - Verification



ACME Client

1. OK, I provisioned that **web resource**.

ACME Server (CA)

4. You're authorized to manage certs for **example.com**

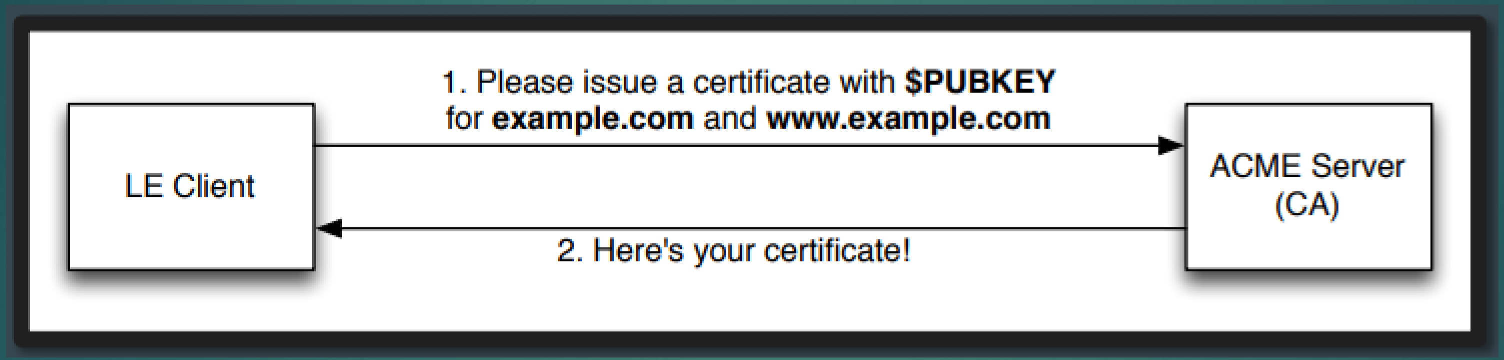
0. Provision **5069a88f** at **/02180d4f**

Web Server example.com

2. GET http://example.com/**02180d4f**

3. 5069a88f

ACME – Issuance and Revocation

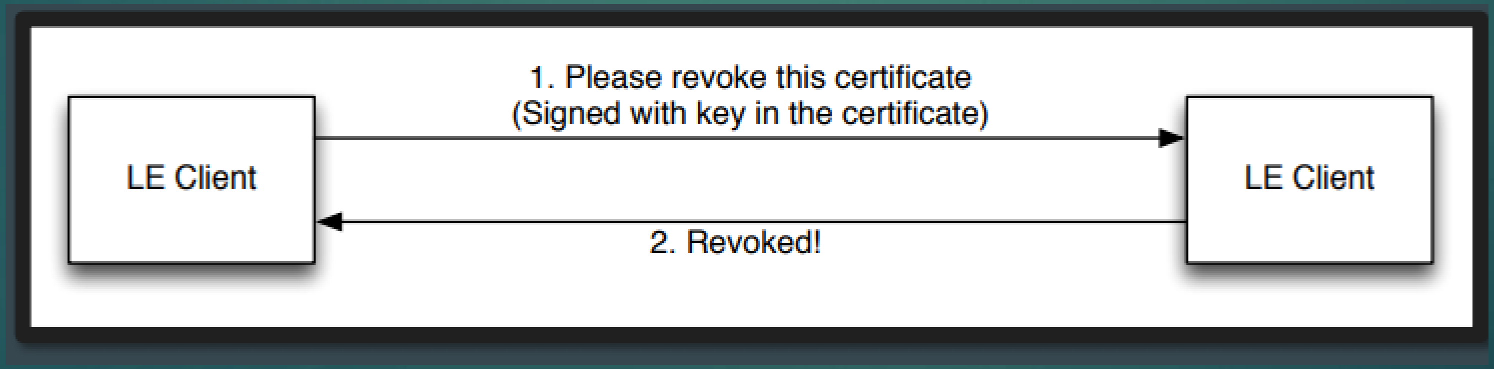


LE Client

1. Please issue a certificate with **$PUBKEY** for **example.com** and **www.example.com**

ACME Server (CA)

2. Here's your certificate!



LE Client

1. Please revoke this certificate (Signed with key in the certificate)

LE Client

2. Revoked!

ACME - Protocol

- The ACME client prompts the operator for the intended domain name(s) that the web server is to stand for
- The ACME client presents the operator with a list of CAs from which it could get a certificate.
- The operator selects a CA.
- In the background, the ACME client contacts the CA and requests that it issue a certificate for the intended domain name(s).
- The CA verifies that the client controls the requested domain name(s).
- Once the CA is satisfied, the certificate is issued and the ACME client automatically downloads and installs it, potentially notifying the operator via email, SMS, etc.
- The ACME client periodically contacts the CA to get updated certificates, stapled OCSP responses
- To request that a certificate be revoked, the client sends a POST request to the ACME server's revoke-cert URL.

OCSP

- Online Certificate Status Protocol
  - A request is **made by the browser** to the CA about the validity of a specific TLS Certificate
    - CA runs a OCSP Responder that checks and tells whether the certificate is valid or revoked
  - Response returned by the CA is digitally signed by it;
  - Defined in RFC 2560 and RFC 5019

OCSP Stapling

- An alternative to OCSP
- **Web server sends** a query to CA Server (OCSP Responder)
- OCSP Responder responds with status of certificate and digitally signs the response and timestamps it
- Web server caches the response received and staples with TLS Certificate and sends it to client during SSL handshake
- Defined in RFC 6066
  - The word stapling is not used; but “status_request” is used

OCSP Stapling

- Advantage:
  - Eliminates the need for browser to contact the CA
- Disadvantage:
  - Most TLS certificates are signed by intermediate CA's which are signed by a root CA
  - Validity of both certificates need to be verified; however OCSP stapling allows only one certificate status to be sent

Certificate Validation

- Validating Chain of Trust - A recursive program!
  - As you go several levels deeper, complexity increases and potential of risk increases!
- Implemented by PKI enabled Application (Eg: Browsers)
- The validation process performs following checks
  - Format
  - Signature Validation - Digital signature of the issuer (CA)
  - Time (Validity of the certificate)
  - Revocation (CRL verification)
  - **Trust (Public Key verification) till root level**

Certificate Validation Failures – Typical Cases

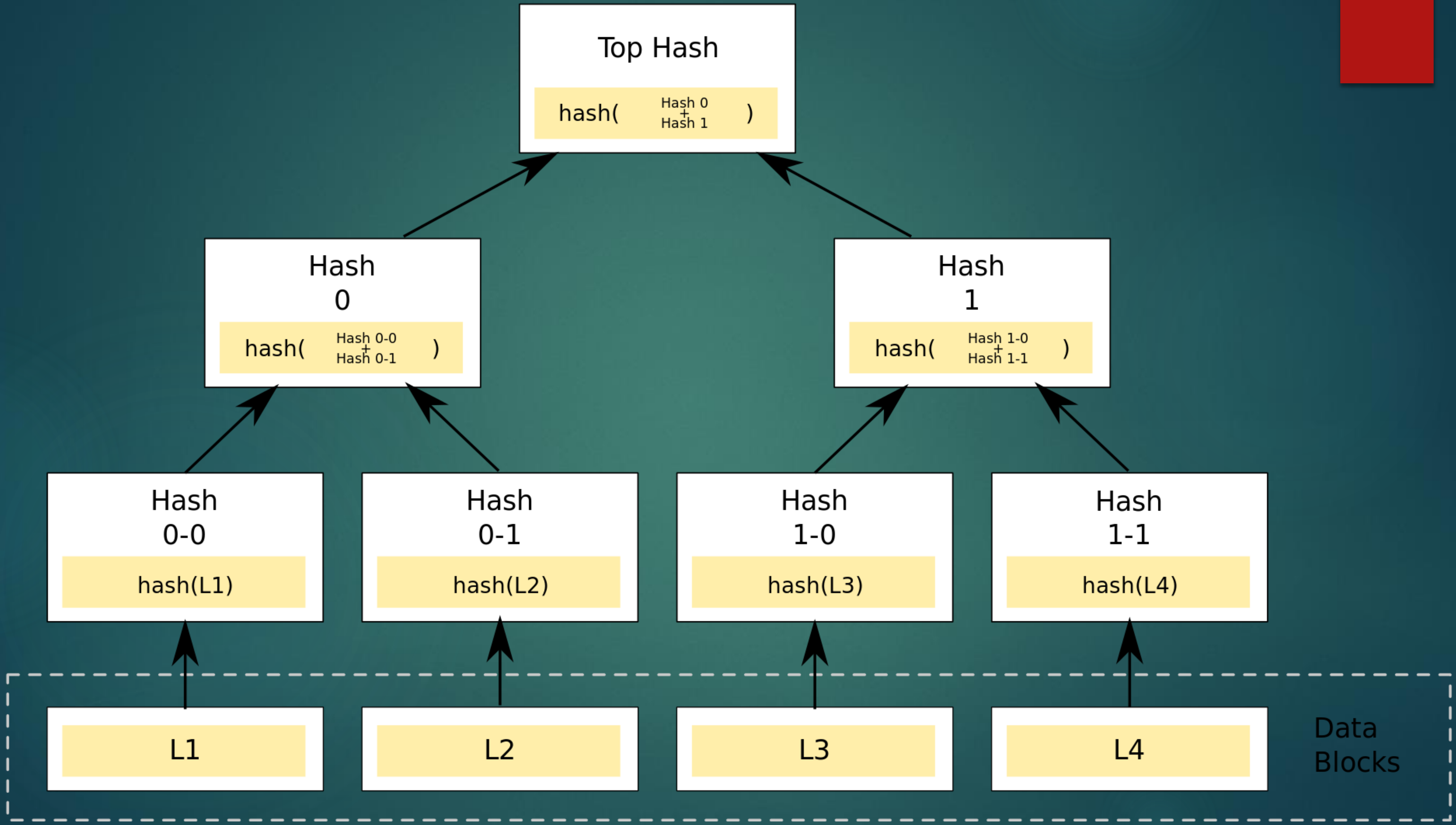
- Domain Mismatch
- Certificate Expired
- Could not find path to certificate

Certificate Validation Algorithm

- Algorithm in Brief
    1. Check for Validity (Time, CRL (except for root), Format) of Certificate
    2. Check and Validate the Signature in the Certificate using the issuer's certificate (which contains the public key) – including the CPS (Policy)
    3. If the issuer's certificate is not a self-signed certificate, then continue with this certificate from Step 1
    4. If it is a self-signed certificate,
        - Check if the Certificate is present in trust stores (Trusted Root CA)
            - If present, trust it and exit (allow user to proceed further)
            - If not prompt the user to take a decision to trust it or leave the site

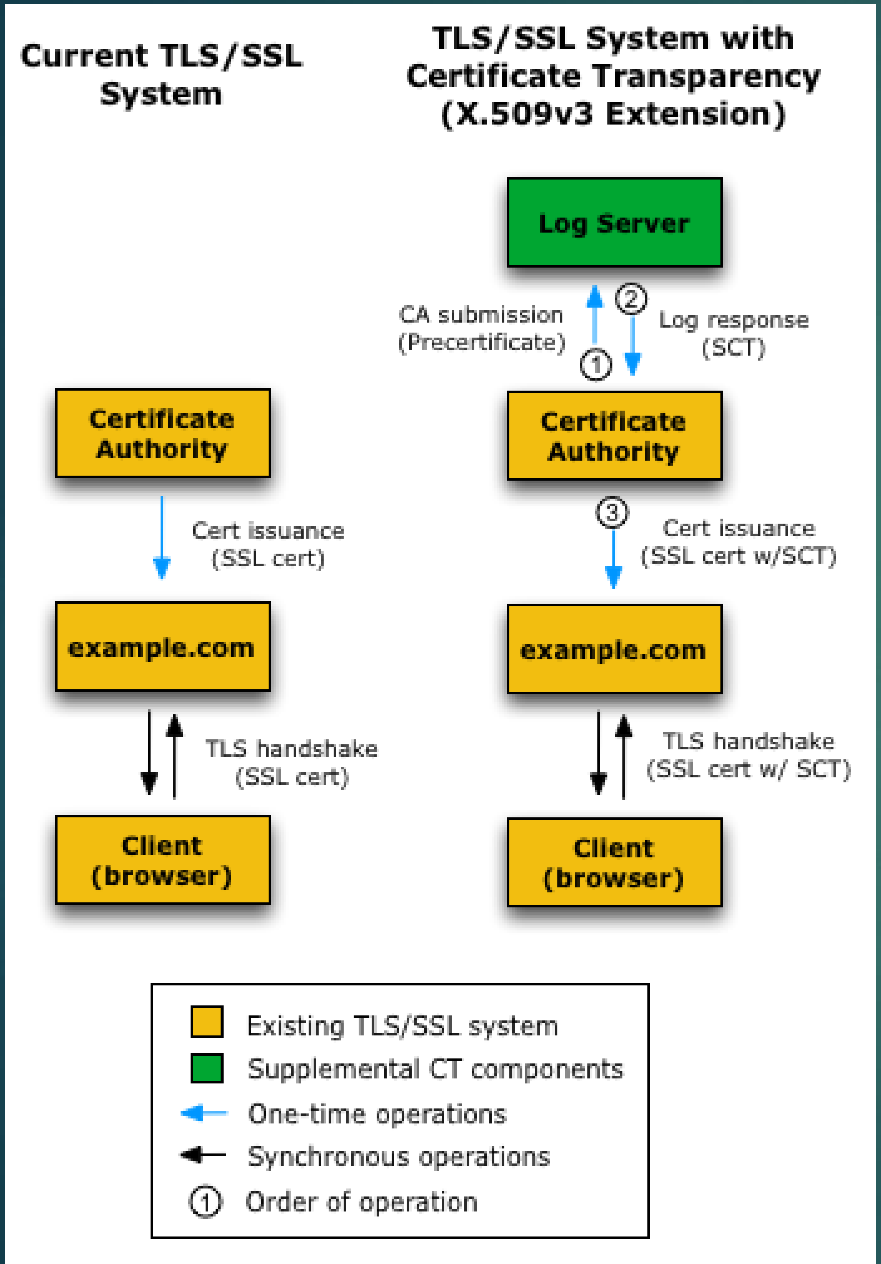
Certificate Transparency

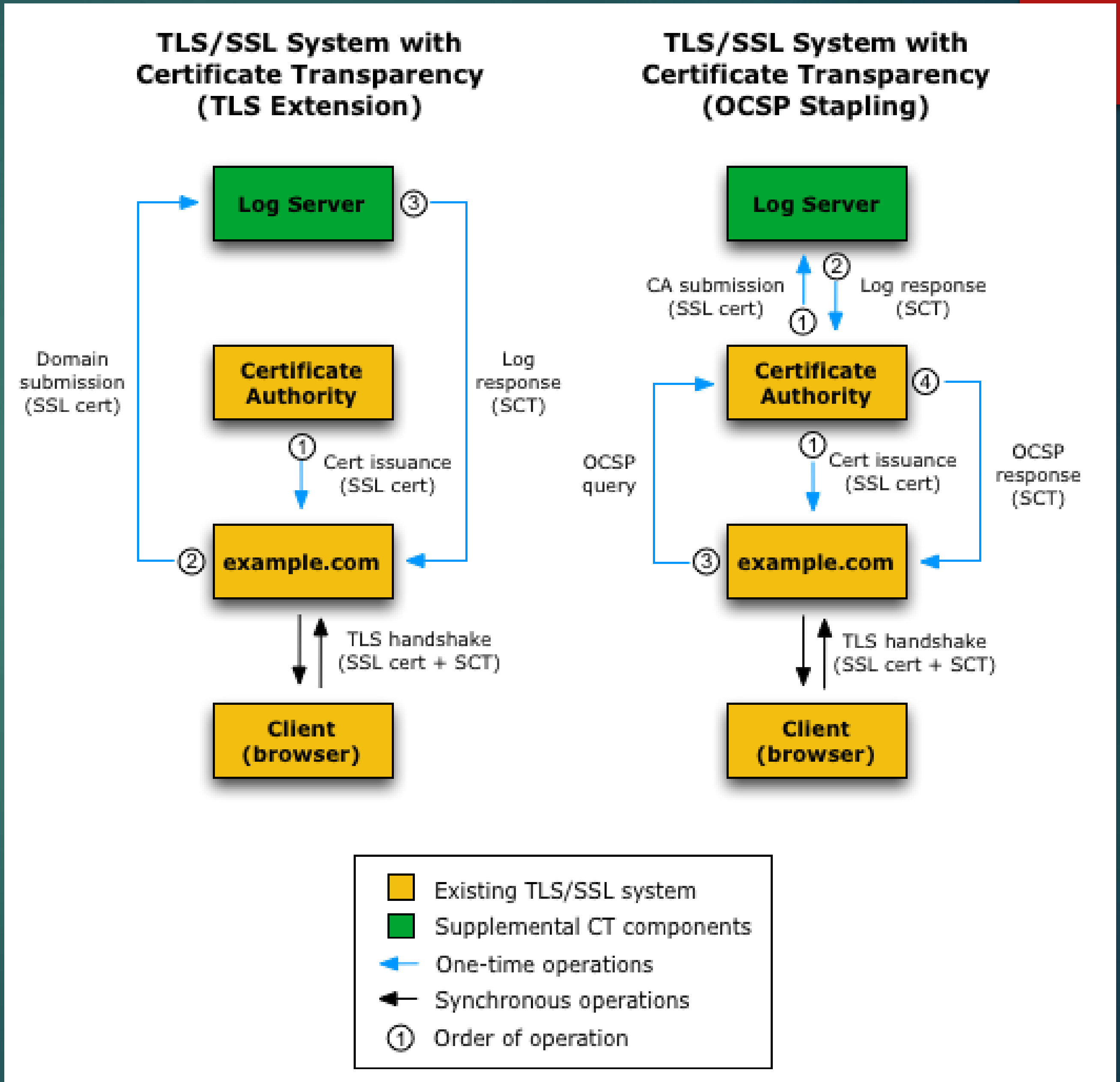
- Certificate Logs
  - Append-only, Cryptographically-assured, publicly auditable;
  - Operated as a network service
  - Few copies of logs (around 10) accessible across the world is sufficient
    - Each log can operate independently of other logs
  - Each certificate log must publicly advertise its URL and its public key
- Log Operations
  - Any one can submit a certificate to a log
  - Log server validates it, and respond with a signed certificate timestamp (SCT)
    - SCT is the maximum time period (MMD) required to add the certificate to the log
    - SCT accompanies certificate throughout the certificate lifetime


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

Delivering SCT with a Certificate

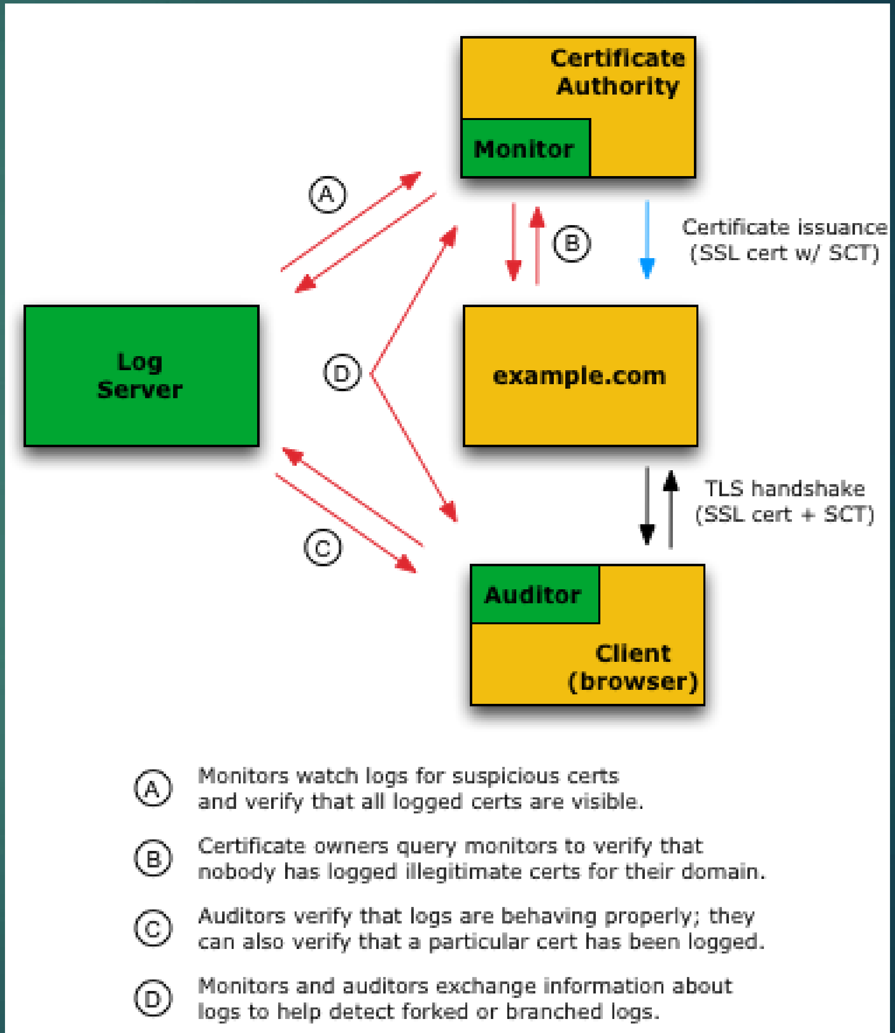
- X.509 v3 Extension
  - CA's add SCT to certificate using an X.509 V3 extension
- TLS Extension
  - Server operators add SCT using special TLS Extension
    - In this case, server operator submits the certificate to log instead of CA
    - *signed_certificate_timestamp* TLS Extension is used
- OCSP Stapling
  - CA simultaneously issues certificate to log server and server operator
  - Server makes OCSP query to CA, CA responds with SCT
  - SCT is added to the OCSP extension


Current TLS/SSL System
TLS/SSL System with Certificate Transparency (X.509v3 Extension)
Log Server
CA submission (Precertificate)
1
2
Log response (SCT)
Certificate Authority
Cert issuance (SSL cert)
Certificate Authority
3
Cert issuance (SSL cert w/SCT)
example.com
example.com
TLS handshake (SSL cert)
TLS handshake (SSL cert w/ SCT)
Client (browser)
Client (browser)
🟨 Existing TLS/SSL system
🟩 Supplemental CT components
One-time operations
Synchronous operations
① Order of operation


TLS/SSL System with Certificate Transparency (TLS Extension)
TLS/SSL System with Certificate Transparency (OCSP Stapling)
Log Server
3
Domain submission (SSL cert)
Log response (SCT)
Certificate Authority
1
Cert issuance (SSL cert)
2
example.com
TLS handshake (SSL cert + SCT)
Client (browser)
Log Server
CA submission (SSL cert)
1
2
Log response (SCT)
Certificate Authority
4
OCSP query
OCSP response (SCT)
1
Cert issuance (SSL cert)
3
example.com
TLS handshake (SSL cert + SCT)
Client (browser)
🟨 Existing TLS/SSL system
🟩 Supplemental CT components
One-time operations
Synchronous operations
① Order of operation

Auditing and Monitoring Services

- Monitors are programs that watch for:
  - Suspicious certificates in logs, such as:
    - illegitimate or unauthorized certificates
    - Unusual certificate extensions
    - Certificates with strange permissions
  - Typically run by CAs
- Auditors verify overall integrity of logs
  - Programs that compute Merkle Proofs
  - Typically run by browsers



Ⓐ 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.

Summary

- Ever-increasing use of TLS Certificates, thanks to Cloud and IoT
- Ever-increasing attacks and bugs!
- Mechanisms to increase Transparency, and block-chain inspired solutions springing up !

References

- Automated Certificate Management Environment (ACME)
  - https://tools.ietf.org/html/draft-ietf-acme-acme-07
- Automated Certificate Management – ACME + Let's Encrypt by Richard Barnes
  - https://ripe71.ripe.net/presentations/32-Automated-Certificate-Management.pdf
- Certificate Transparency – RFC 6962
  - https://www.certificate-transparency.org/how-ct-works
- Polygora - https://polygora.tech/

Thank you