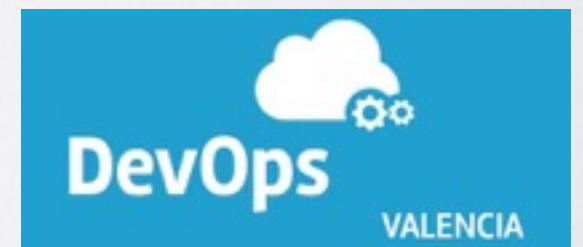




SSL/TLS and Digital Certificates



cesar@cesarsaez.es

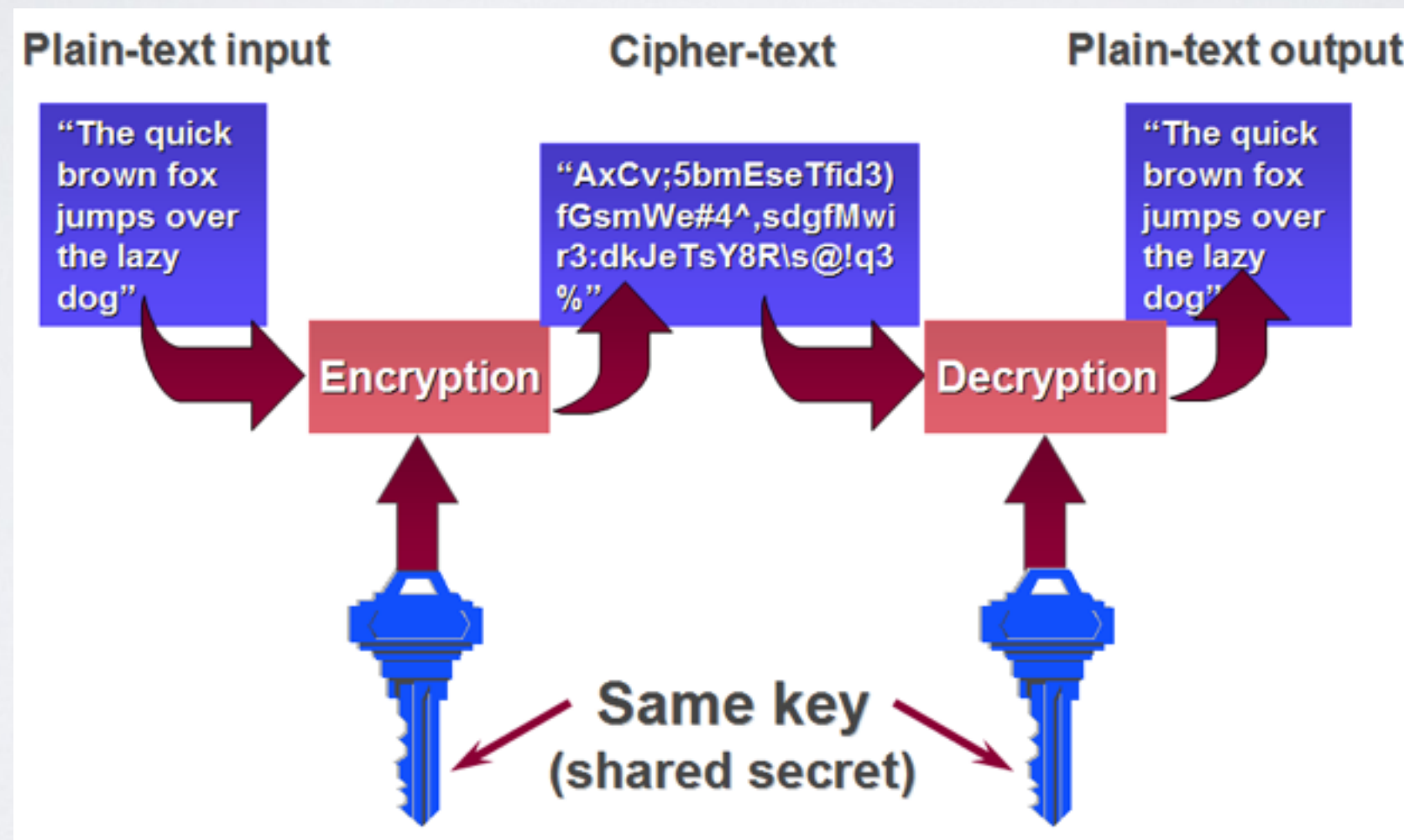
<http://valenciadevops.me>

TOPICS

- Encryption types
- What are SSL and TLS
- Digital Certificates
- TLS usage
- What is a PKI
- Certificate revocation
- Certificate stores
- Certificate file types
- TLS Handshake
- TLS performance
- Certificate security issues

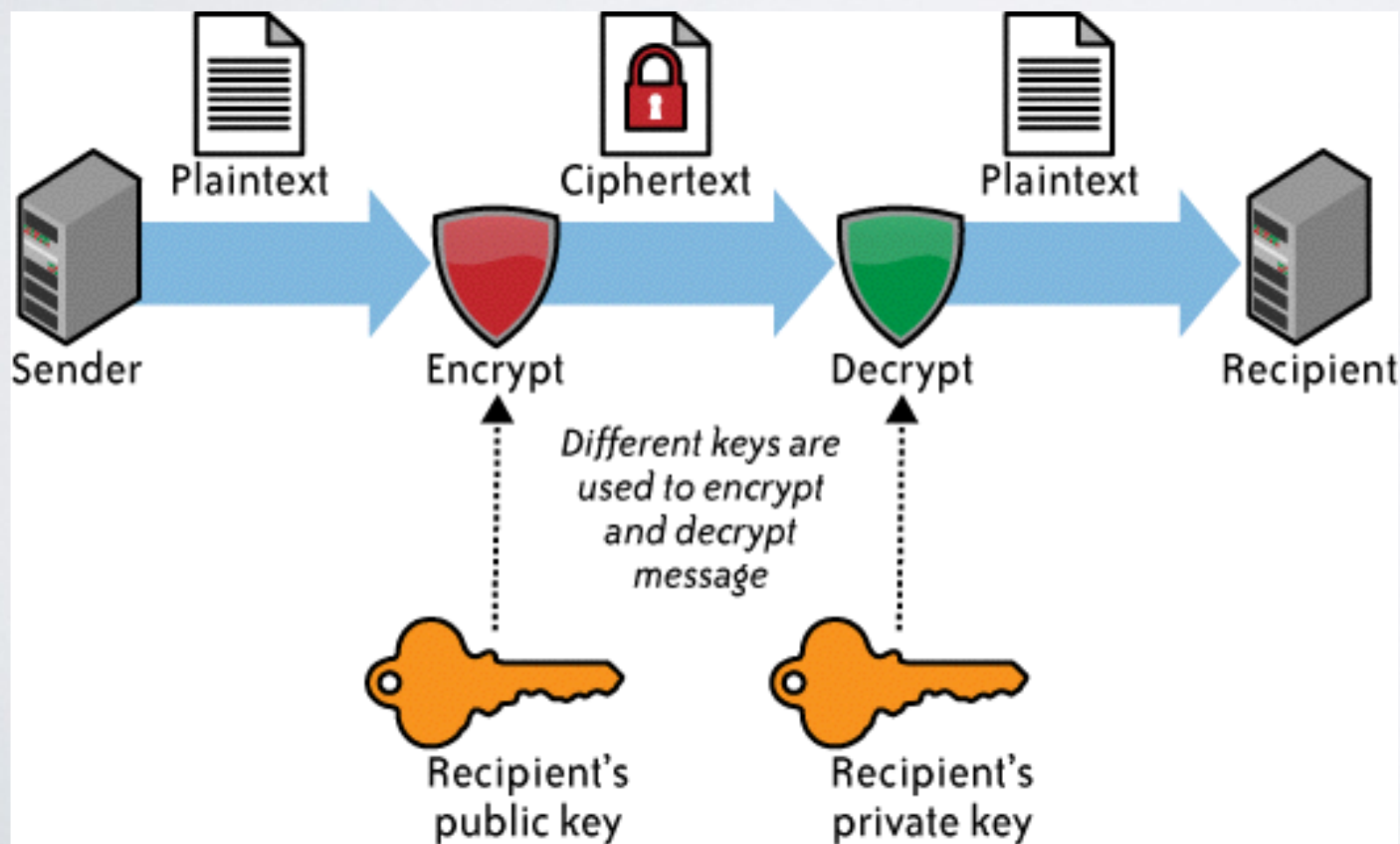
ENCRYPTION TYPES

- **Symmetric encryption:** The same key is used for encryption and decryption. The key must be exchanged so that both the data sender and the recipient can access the plaintext data.



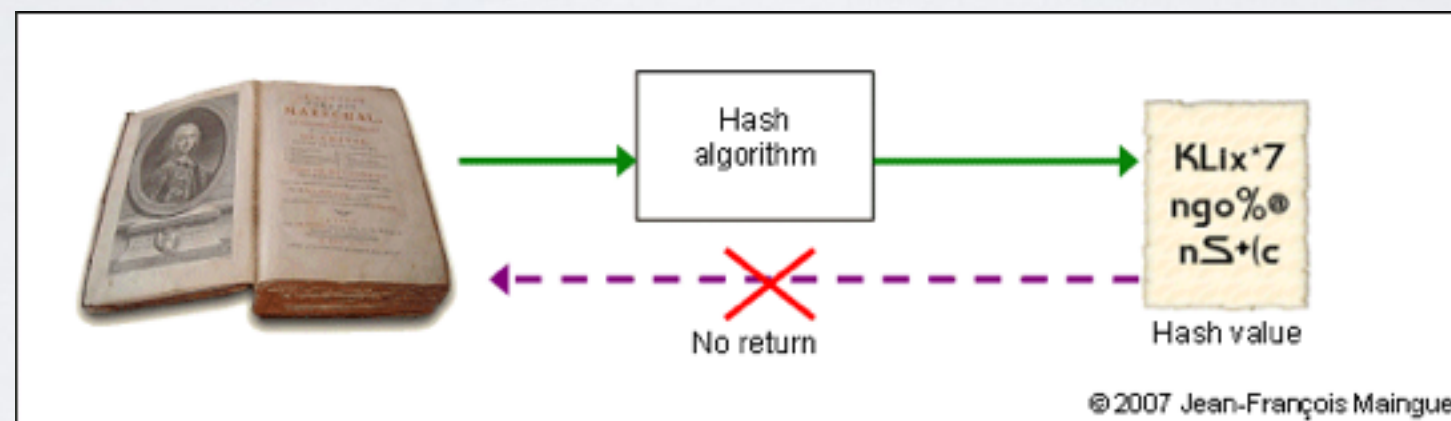
ENCRYPTION TYPES

- **Asymmetric encryption:** Two mathematically related keys, a key pair consisting of a public key and a private key, are used in the encryption and decryption processes.
 - If the public key is used for encryption, the associated private key is used for decryption.
 - If the private key is used for encryption, the associated public key is used for decryption.
- **Private key must not be shared. Public key can be shared freely**



ENCRYPTION TYPES

- One-way **hash algorithms** like MD5, SHA-1, SHA-256 convert any input message to a long unique and irreversible string of numbers and letters that **serve as a cryptographic fingerprint** for that message.



- A **cipher suite** is a **combination** of **authentication, encryption, message authentication code and key exchange algorithms** used to negotiate the security settings for a network connection using the TLS / SSL.

ENCRYPTION TYPES

- **Data** is encrypted **with** a **symmetric** algorithm.
- Only the **symmetric encryption key** is encrypted **with** the **asymmetric** algorithm.
- It is very rare for an application to use only an asymmetric encryption algorithm.

WHAT ARE SSL AND TLS

- **SSL/TLS** are protocols to **provide privacy and data integrity** between two communicating computer applications.
- TLS (Transport Layer Security) is the evolution of SSL (Secure Socket Layer).
- TLS is a IETF standard based on the earlier SSL specifications developed by Netscape Communications.

WHAT ARE SSL AND TLS

- The connection is private because symmetric cryptography is used to **encrypt** the **data** transmitted.
- The **keys** for this symmetric encryption are **generated uniquely for each connection** and are based on a secret negotiated at the start of the session.
- The connection is reliable because each message transmitted includes a message integrity check.

WHAT ARE SSL AND TLS

- There are five protocols in the SSL/TLS family: SSL v2, SSL v3, TLS v1.0, TLS v1.1, and TLS v1.2. Of these:
- SSL v2 is insecure and must not be used.
- SSL v3 is insecure when used with HTTP and weak when used with other protocols. It's also obsolete (since June 2015), which is why it shouldn't be used.
- TLS v1.0 is largely still secure. When used with HTTP, it can almost be made secure with careful configuration.
- **TLS v1.1 and v1.2 are without known security issues.**
- In order **to support older clients**, you need to continue to support **TLS v1.0 and TLS v1.1** for the time being.
- **TLS v1.2 should be your main protocol.**

DIGITAL CERTIFICATES

- In cryptography, a **digital certificate** (also known as a public key certificate) is an **electronic document used to prove ownership** of a public key.
- Includes information about the key, its **owner's identity**, and the **digital signature** of an entity **that has verified** the certificate's contents are correct.

DIGITAL CERTIFICATES

- In a typical public-key infrastructure (**PKI**) scheme, the **signer is a Certificate Authority** (CA).
- Certificates are an important component of TLS.
- They prevent identity spoof attacks for a secure server.
- Used as well in other important applications, such as email encryption, email signature and code signing, document signature, etc...

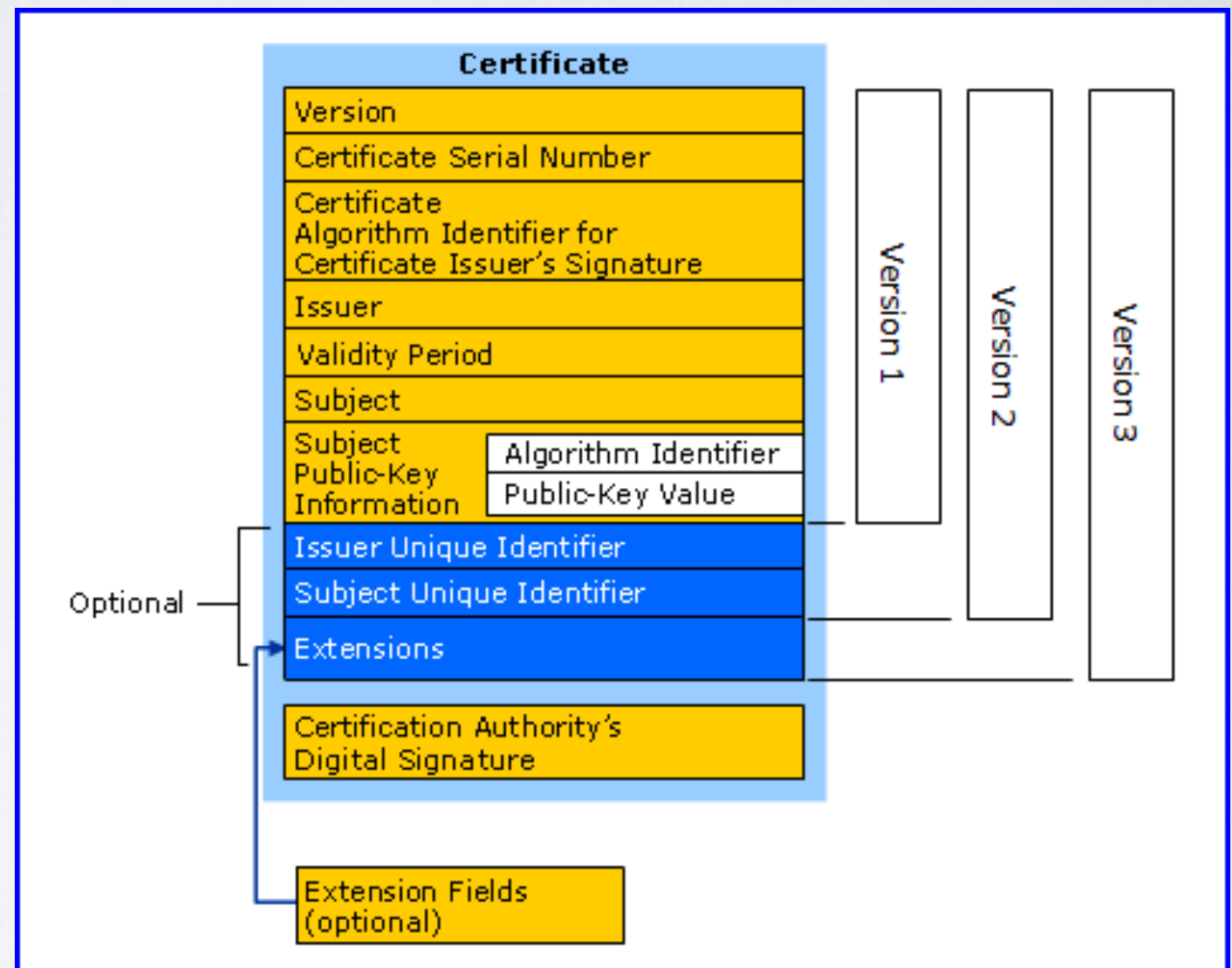
DIGITAL CERTIFICATES

- Standard X.509 v3 is based RFC5280 (<http://tools.ietf.org/html/rfc5280#section-4.2.1.3>)
- **Key Usage** extensions define the purpose of the public key contained in a certificate.
- **Extended Key Usage** refines Key Usage extension
- Can be combined to allow or restrict for as many operations as needed

DIGITAL CERTIFICATES

- The structure of an X.509 v3 digital certificate

- Certificate
 - Version Number
 - Serial Number
 - Signature Algorithm ID
 - Issuer Name
 - Validity period
 - Not Before
 - Not After
 - Subject name
 - 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



DIGITAL CERTIFICATES



Safari is using an encrypted connection to en.wikipedia.org.

Encryption with a digital certificate keeps information private as it's sent to or from the https website en.wikipedia.org.



GlobalSign Root CA



GlobalSign Organization Validation CA - SHA256 - G2



*.wikipedia.org



***.wikipedia.org**

Issued by: GlobalSign Organization Validation CA - SHA256 - G2

Expires: Sunday 19 February 2017 at 13 h 00 min 00 s Central European Standard Time

✓ This certificate is valid

► **Trust**

▼ **Details**

Subject Name	
Country	US
State/Province	California
Locality	San Francisco
Organization	Wikimedia Foundation, Inc.
Common Name	*.wikipedia.org
Issuer Name	
Country	BE
Organization	GlobalSign nv-sa
Common Name	GlobalSign Organization Validation CA - SHA256 - G2



Hide Certificate

OK

DIGITAL CERTIFICATES

Serial Number	11 21 E7 DF D9 CF 1C 5E 9A D5 9F 41 5F 6D A9 1F E2 4B
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	none
Not Valid Before	Tuesday 23 June 2015 at 20 h 37 min 07 s Central European Summer Time
Not Valid After	Sunday 19 February 2017 at 13 h 00 min 00 s Central European Standard Time
Public Key Info	
Algorithm	Elliptic Curve Public Key (1.2.840.10045.2.1)
Parameters	Elliptic Curve secp256r1 (1.2.840.10045.3.1.7)
Public Key	65 bytes : 04 6B 3F AD 07 E8 95 CF ...
Key Size	256 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 03 75 7B ED A6 35 70 0F ...

DIGITAL CERTIFICATES

Extension Key Usage (2.5.29.15)
Critical YES
Usage Digital Signature, Key Encipherment

Extension Basic Constraints (2.5.29.19)
Critical NO
Certificate Authority NO

Extension Extended Key Usage (2.5.29.37)
Critical NO
Purpose #1 Server Authentication (1.3.6.1.5.5.7.3.1)
Purpose #2 Client Authentication (1.3.6.1.5.5.7.3.2)

Extension Subject Key Identifier (2.5.29.14)
Critical NO
Key ID 08 BF CD A9 EE 0F EA 30 D3 32 6B 2D DF FF 64 A4 CC 8B 23 F7

Extension Authority Key Identifier (2.5.29.35)
Critical NO
Key ID 96 DE 61 F1 BD 1C 16 29 53 1C C0 CC 7D 3B 83 00 40 E6 1A 7C

Extension Subject Alternative Name (2.5.29.17)
Critical NO
DNS Name *.wikipedia.org
DNS Name *.mediawiki.org
DNS Name *.wikibooks.org

DIGITAL CERTIFICATES

Extension	Certificate Policies (2.5.29.32)
Critical	NO
Policy ID #1	(2.23.140.1.2.2)
Qualifier ID #1	Certification Practice Statement (1.3.6.1.5.5.7.2.1)
CPS URI	https://www.globalsign.com/repository/
Extension	CRL Distribution Points (2.5.29.31)
Critical	NO
URI	http://crl.globalsign.com/gc/gsorganizationvalsha2g2.crl
Extension	Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)
Critical	NO
Method #1	CA Issuers (1.3.6.1.5.5.7.48.2)
URI	http://secure.globalsign.com/cacert/gsorganizationvalsha2g2r1.crt
Method #2	Online Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
URI	http://ocsp2.globalsign.com/gsorganizationvalsha2g2
Fingerprints	
SHA1	11 D5 D2 0A 9A F8 D9 FC 23 6E 5C 5C 30 EC AF 68 F5 68 FB A3
MD5	38 B6 AF AB E0 D8 7D 3B ED 8B 32 D5 5E 2B F5 30



Hide Certificate

OK

DIGITAL CERTIFICATES

```
openssl s_client -connect www.wikipedia.org:443 2>/dev/null | openssl x509 -noout -text
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      11:21:97:2e:32:a5:e5:b2:e2:9d:47:2d:fe:db:72:d6:27:6e
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Organization Validation CA - SHA256 - G2
    Validity
      Not Before: Dec 16 21:24:03 2014 GMT
      Not After : Feb 19 12:00:00 2017 GMT
    Subject: C=US, ST=California, L=San Francisco, O=Wikimedia Foundation, Inc., CN=*.wikipedia.org
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
        Modulus (2048 bit):
          00:c1:f5:b3:91:51:90:ea:df:ee:9d:df:40:da:aa:
          29:ba:de:d0:9f:fb:c3:78:ac:60:3b:7f:b5:16:70:
          e1:b1:e6:ee:62:3b:dd:8e:77:df:98:b0:ba:5b:eb:
          06:6b:2a:f7:74:52:30:78:b3:2f:1f:09:d1:17:64:
          29:c7:e2:b5:64:bf:a3:49:08:28:fd:70:7a:10:f3:
          1c:e7:ab:1f:ab:35:7b:b6:77:8f:fb:88:ea:63:33:
          0d:93:2c:11:60:e7:3c:0d:b5:93:0a:ad:5c:42:30:
          60:40:23:fa:d4:54:f3:7d:a3:dc:27:89:de:93:02:
          13:be:6e:13:e9:67:37:48:55:24:85:dd:d9:cd:3f:
          86:73:d1:01:64:d5:10:d5:7a:15:01:82:cb:5c:f0:
          a3:09:67:76:f1:53:00:b4:3e:e0:b4:82:36:d6:67:
          c1:5b:52:79:c8:0a:54:f0:0e:fd:69:0b:32:4d:8e:
          74:b0:7a:50:84:d3:d7:03:cd:88:fd:aa:ce:6d:ae:
          15:67:ad:3b:f3:33:37:aa:5a:3a:24:b6:0e:a5:c5:
          a7:8a:fb:29:94:d2:34:e9:77:09:39:fc:32:e7:ca:
          18:fa:80:d1:a8:c9:24:32:d2:5f:4f:ca:0e:60:b5:
          4b:0e:a7:74:17:ab:52:e4:43:49:a0:de:12:ad:88:
          f0:73
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
      X509v3 Certificate Policies:
        Policy: 2.23.140.1.2.2
        CPS: https://www.globalsign.com/repository/

      X509v3 Subject Alternative Name:
        DNS:*.wikipedia.org, DNS:*.mediawiki.org, DNS:*.wikibooks.org, DNS:*.wikidata.org, DNS:*.wiki
        NS:*.wikiquote.org, DNS:*.wikisource.org, DNS:*.wikiversity.org, DNS:*.wikivoyage.org, DNS:*.wiktionary.org,
        .org, DNS:*.m.wikidata.org, DNS:*.m.wikimedia.org, DNS:*.m.wikimediafoundation.org, DNS:*.m.wikinews.org, DNS
```

DIGITAL CERTIFICATES

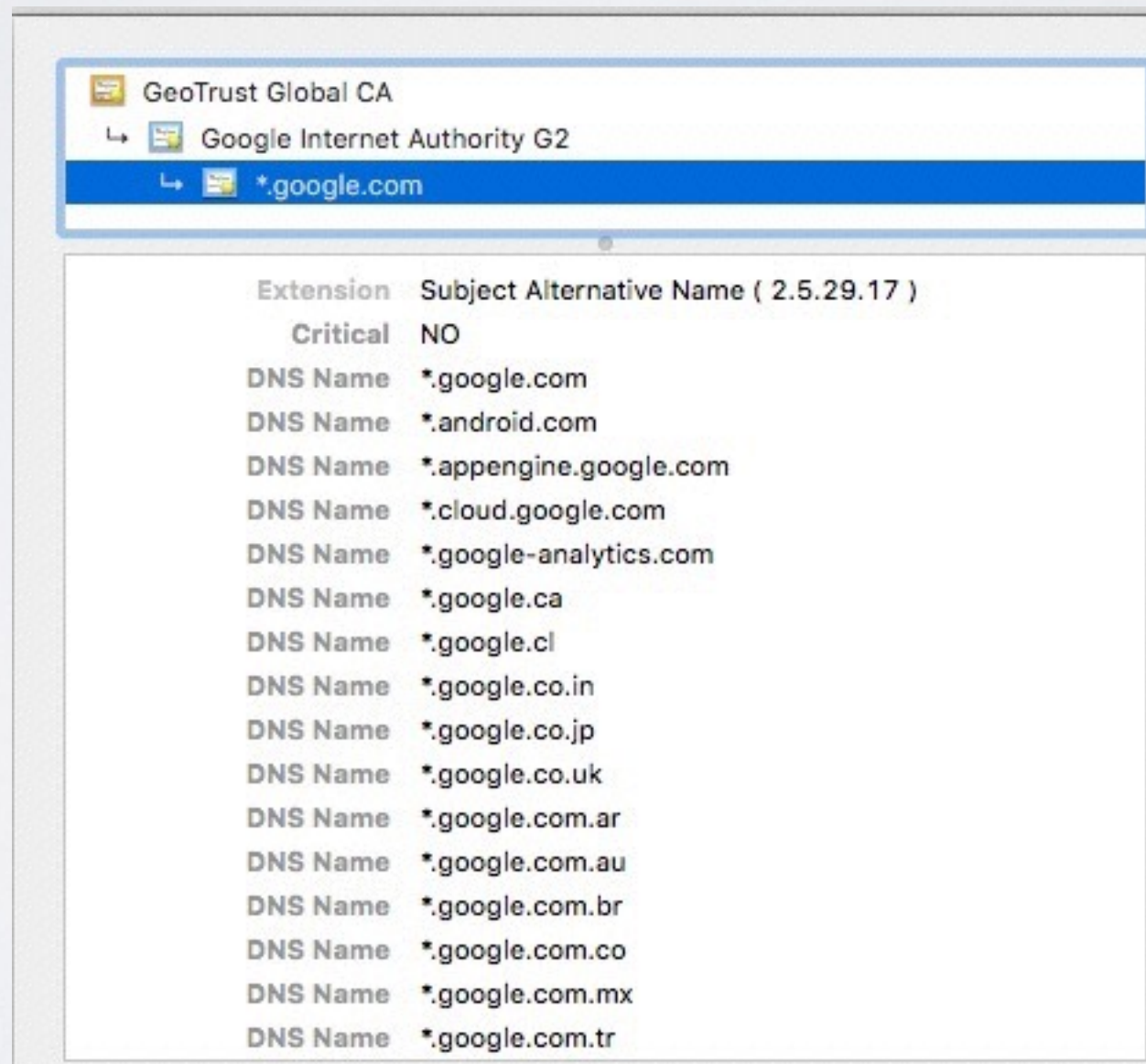
Key usage extension	Description
Digital signature	Use when the public key is used with a digital signature mechanism to support security services other than non-repudiation, certificate signing, or CRL signing. A digital signature is often used for entity authentication and data origin authentication with integrity.
Non-repudiation	Use when the public key is used to verify digital signatures used to provide a non-repudiation service. Non-repudiation protects against the signing entity falsely denying some action (excluding certificate or CRL signing).
Key encipherment	Use when a certificate will be used with a protocol that encrypts keys. An example is S/MIME enveloping, where a fast (symmetric) key is encrypted with the public key from the certificate. SSL protocol also performs key encipherment.
Data encipherment	Use when the public key is used for encrypting user data, other than cryptographic keys.
Key agreement	Use when the sender and receiver of the public key need to derive the key without using encryption. This key can then be used to encrypt messages between the sender and receiver. Key agreement is typically used with Diffie-Hellman ciphers.
Certificate signing	Use when the subject public key is used to verify a signature on certificates. This extension can be used only in CA certificates

DIGITAL CERTIFICATES

Key usage extension	Description
CRL signing	Use when the subject public key is to verify a signature on revocation information, such as a CRL.
Encipher only	Use only when key agreement is also enabled.This enables the public key to be used only for enciphering data while performing key agreement.
Decipher only	Use only when key agreement is also enabled.This enables the public key to be used only for deciphering data while performing key agreement

DIGITAL CERTIFICATES

- Single Common Name
- SAN
- Wildcard

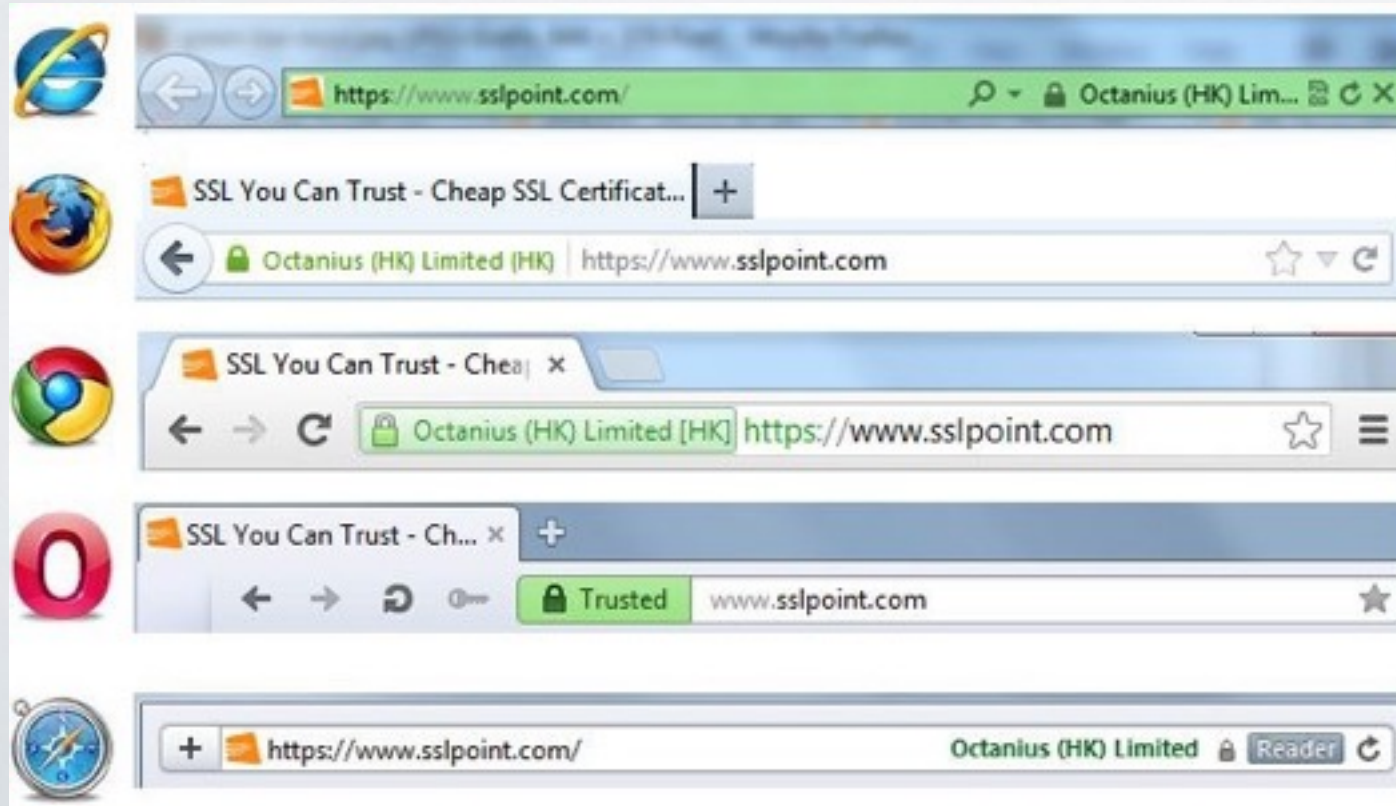


The screenshot shows a browser window with the address bar displaying "GeoTrust Global CA", "Google Internet Authority G2", and "*.google.com". Below the address bar, a table lists the certificate's extensions.

Extension	Subject Alternative Name (2.5.29.17)
Critical	NO
DNS Name	*.google.com
DNS Name	*.android.com
DNS Name	*.appengine.google.com
DNS Name	*.cloud.google.com
DNS Name	*.google-analytics.com
DNS Name	*.google.ca
DNS Name	*.google.cl
DNS Name	*.google.co.in
DNS Name	*.google.co.jp
DNS Name	*.google.co.uk
DNS Name	*.google.com.ar
DNS Name	*.google.com.au
DNS Name	*.google.com.br
DNS Name	*.google.com.co
DNS Name	*.google.com.mx
DNS Name	*.google.com.tr

DIGITAL CERTIFICATES - EV

- Extended Validation Certificate (EV) certificate issued according to a specific set of identity verification criteria
- Require extensive verification of the requesting entity's identity by the CA
- Contain a subject with OIDs with legal information.
- Issuing EV certificates criteria defined by CA/Browser Forum, CAs issuing EV require independent qualified audit review
- Polemics: small business access, effectivity against phishing, Domain Validation problem.



SNI

- Server Name Indication (SNI) is a TLS extension.
- Client indicates which hostname it is attempting to connect to at the start of the TLS handshaking.
- Allows a server to present multiple certificates on the same IP address and TCP port.
- Vastly supported at current web browsers and web servers
- Years ago, with no SNI a HTTPS site required a dedicated IP or use SAN.
- Not supported at Internet Explorer on Windows XP!!!

PKI - DESCRIPTION

- A public key infrastructure (PKI) is a **set of hardware, software, people, policies, and procedures** needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. **PKI and certificates are time based services.**
- The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository and revokes (CRL) them if needed.
- A PKI consists of:
 - A certificate authority (CA) that both issues and signs the digital certificates.
 - A registration authority (RA) which verifies the identity of users requesting information from the CA.
 - A central directory (Directory)—i.e., a secure location in which to store and index keys.
 - A certificate management system.
 - A certificate policy.

PKI - PRIVATE AND PUBLIC

- **Private PKI:**

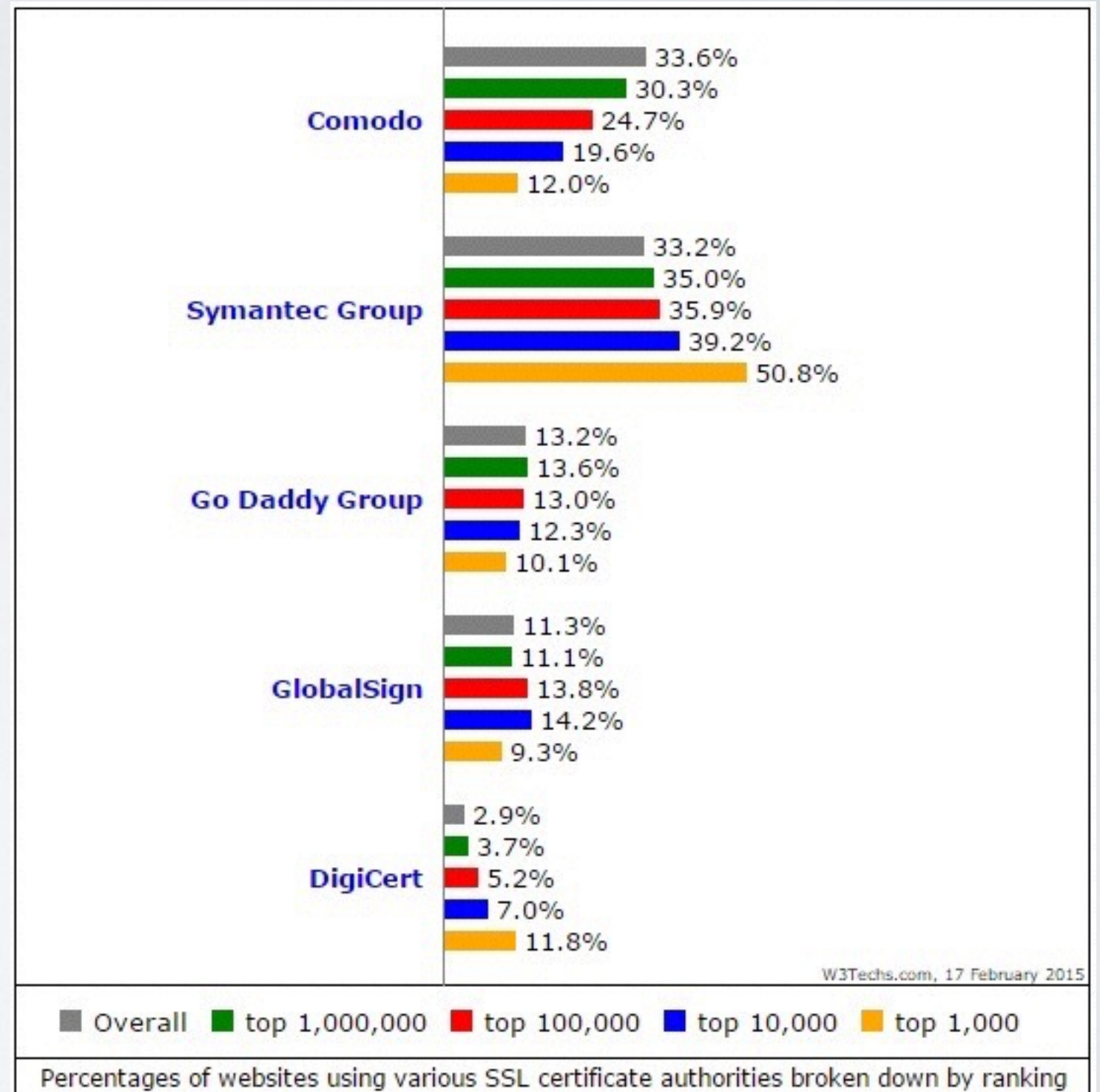
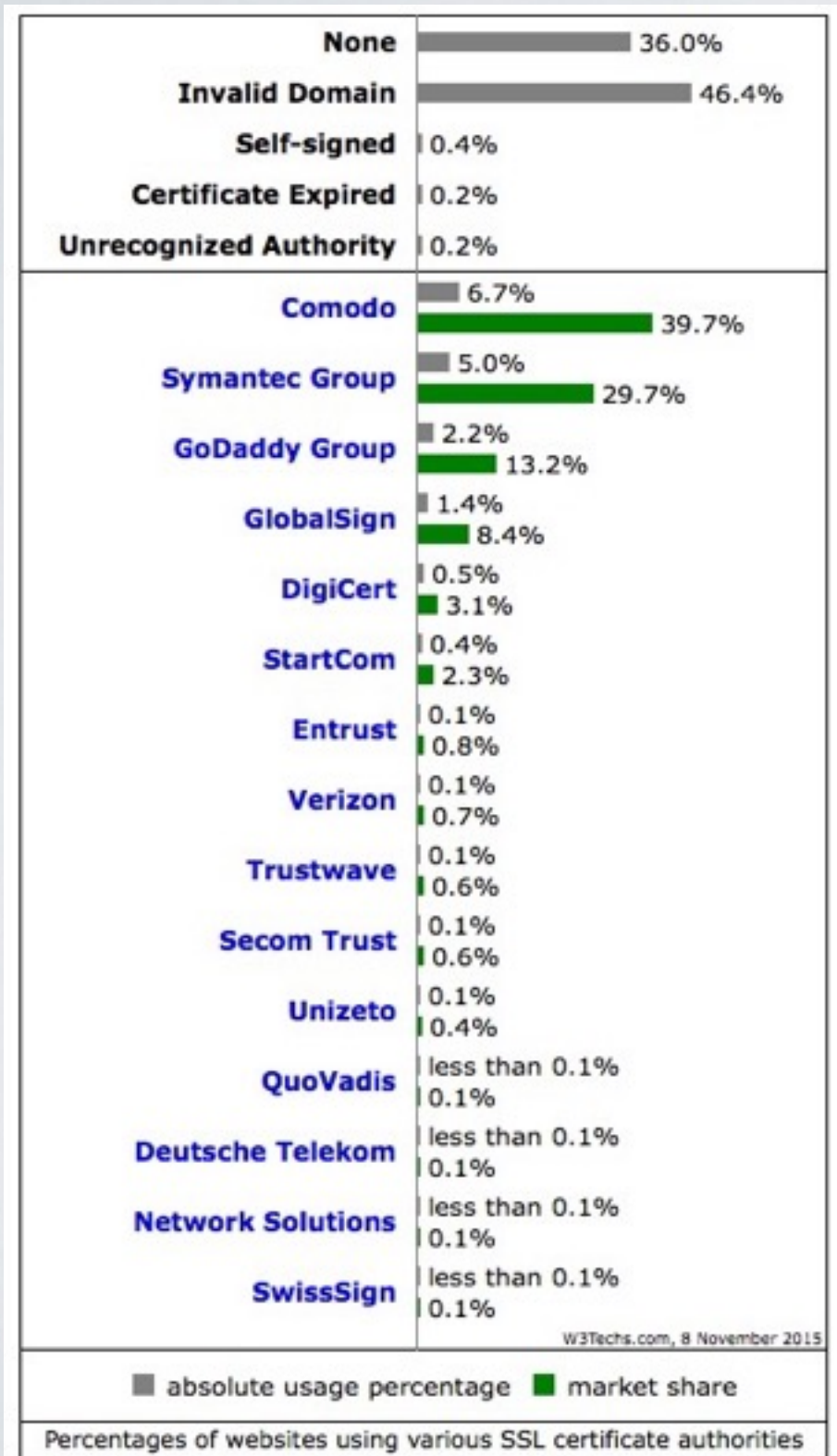
- Certificates are trusted by default only inside the organisation.
- Certificate chain distribution must be designed. Automatic methods are available.
- Economic cost per certificate is less than Public PKI.
- Requires good initial design to avoid future problems.
- Management cost must be considered.
- Depends on the organisation requirements.

PKI - PRIVATE AND PUBLIC

- **Public PKI:**

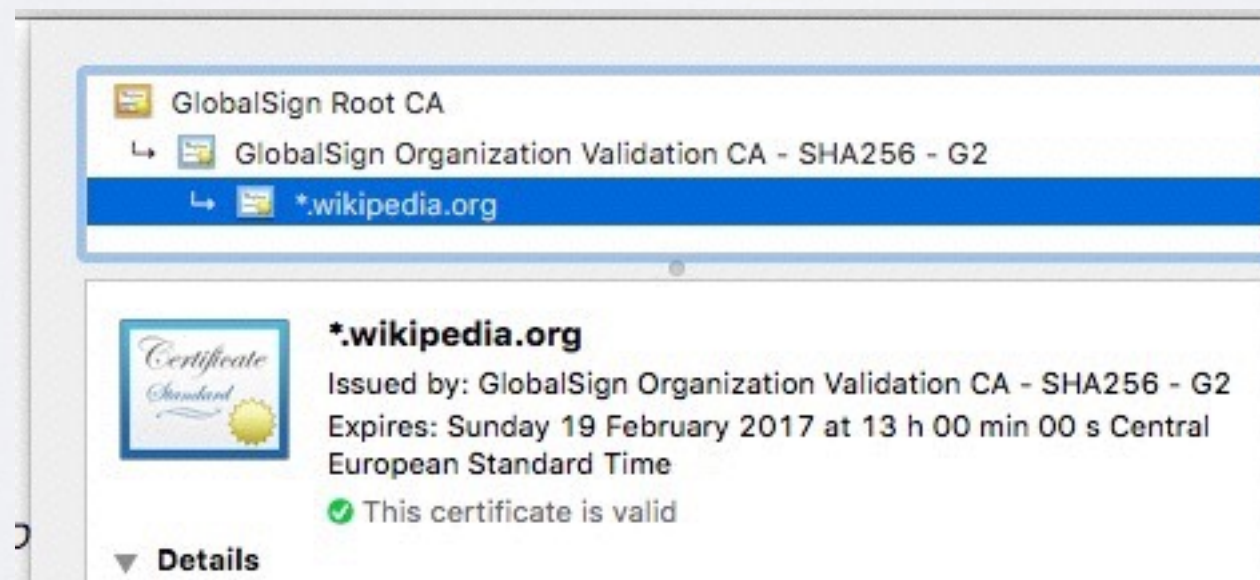
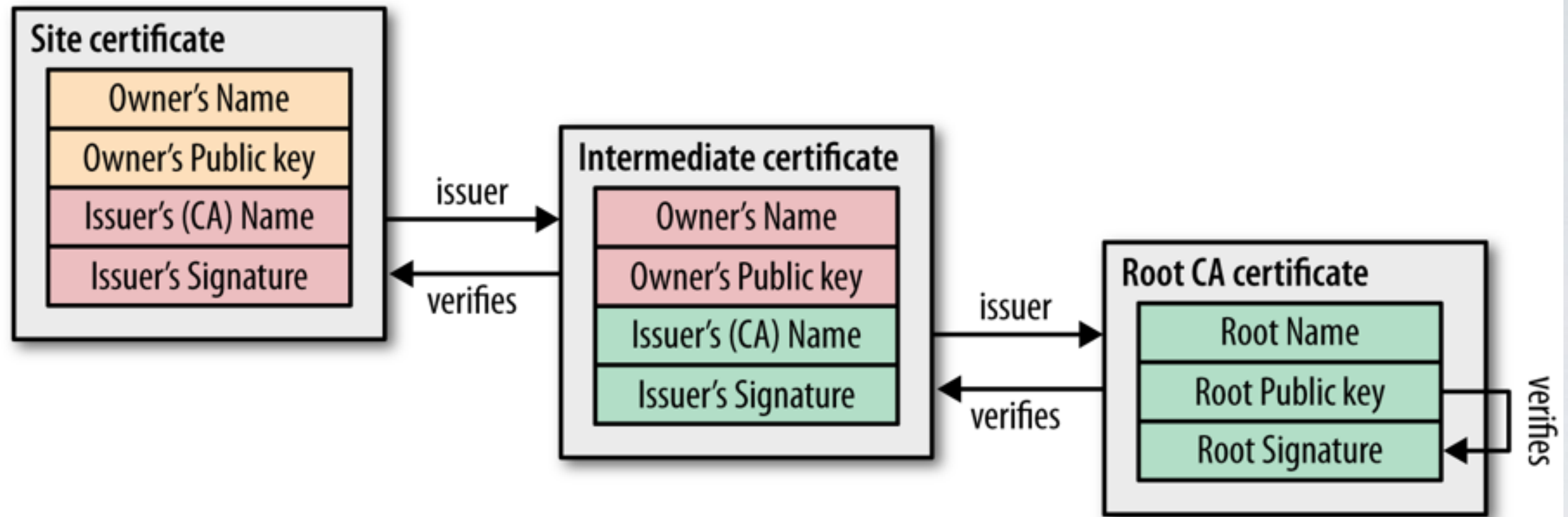
- Due agreements among public PKI vendors and major OS and Browsers, certificates are trusted by default.
- Certificate chain distribution is already done.
- Issued certificates have an economic cost.
- Management cost must be considered.
- Depends on the organisation requirements
- They are audited regularly to be able to operate
- Focussed on PKI business.
- Several services (CRL, OCSP, signature and key algorithm options) and management tools

PKI - PRIVATE AND PUBLIC



source <http://w3techs.com>

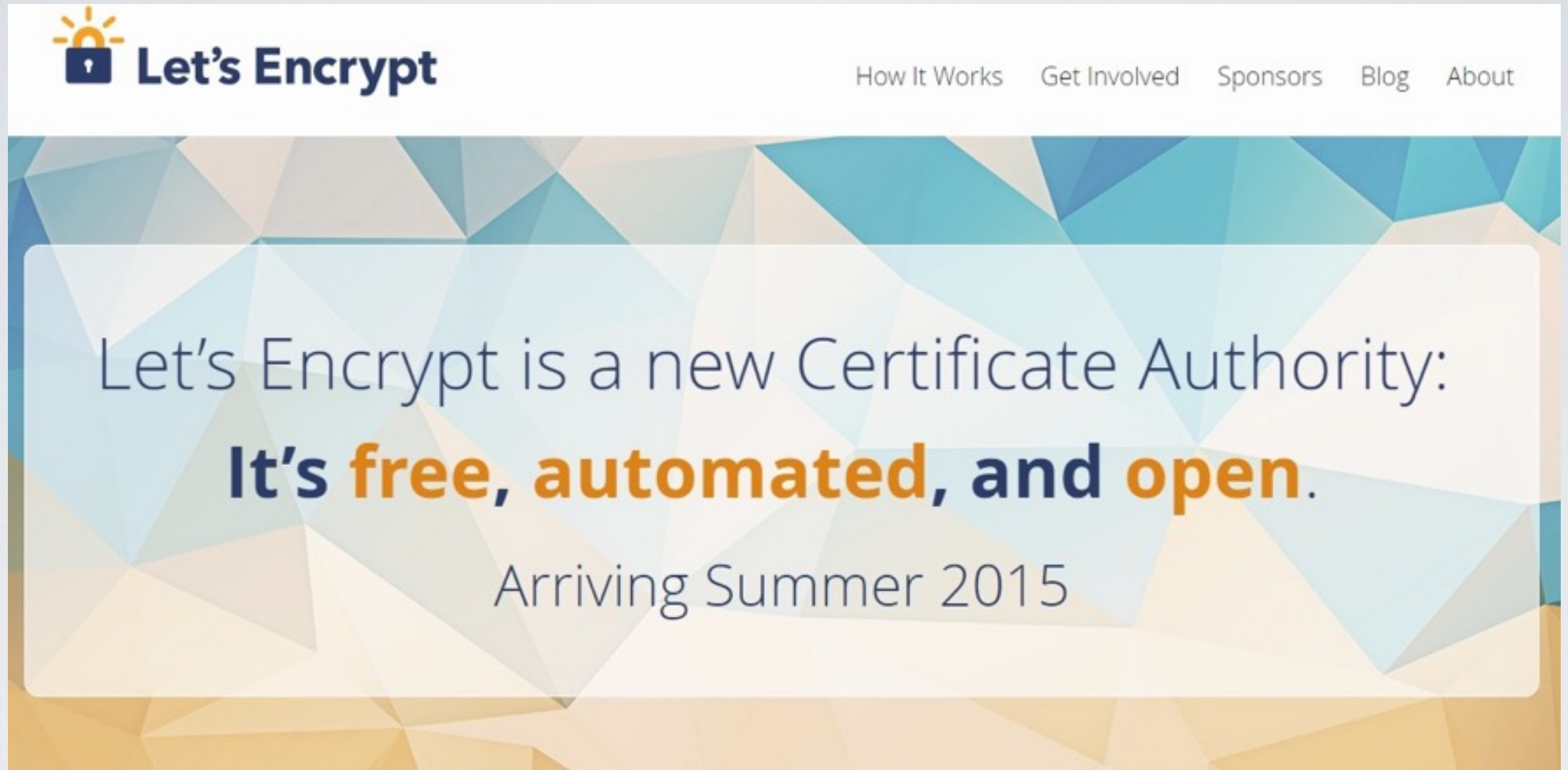
PKI - CERTIFICATE CHAIN



PKI - SOFTWARE

- AD CS
- EJBCA
- OpenSSL
- OpenCA
- SimpleAuthority
- gnoMint
- TinyCA
- XiPKI
- pkIRISGrid

PKI - LET'S ENCRYPT



<https://letsencrypt.org>

PKI - LET'S ENCRYPT

Let's Encrypt is a free, automated, and open certificate authority (CA), run for the public's benefit. Let's Encrypt is a service provided by the Internet Security Research Group (ISRG).

The key principles behind Let's Encrypt are:

- **Free:** Anyone who owns a domain name can use Let's Encrypt to obtain a trusted certificate at zero cost.
- **Automatic:** Software running on a web server can interact with Let's Encrypt to painlessly obtain a certificate, securely configure it for use, and automatically take care of renewal.
- **Secure:** Let's Encrypt will serve as a platform for advancing TLS security best practices, both on the CA side and by helping site operators properly secure their servers.
- **Transparent:** All certificates issued or revoked will be publicly recorded and available for anyone to inspect.
- **Open:** The automatic issuance and renewal protocol will be published as an open standard that others can adopt.
- **Cooperative:** Much like the underlying Internet protocols themselves, Let's Encrypt is a joint effort to benefit the community, beyond the control of any one organization.

PKI - LET'S ENCRYPT

Client tool <https://github.com/letsencrypt/letsencrypt>

Current Features

- Supports multiple web servers:
 - apache/2.x (tested and working on Ubuntu Linux)
 - nginx/0.8.48+ (under development)
 - standalone (runs its own simple webserver to prove you control a domain)
- The private key is generated locally on your system.
- Can talk to the Let's Encrypt (demo) CA or optionally to other ACME compliant services.
- Can get domain-validated (DV) certificates.
- Can revoke certificates.
- Adjustable RSA key bit-length (2048 (default), 4096, ...).
- Can optionally install a http -> https redirect, so your site effectively runs https only (Apache only)
- Fully automated.
- Configuration changes are logged and can be reverted.
- Text and ncurses UI.
- Free and Open Source Software, made with Python.

PKI - LET'S ENCRYPT

Public Beta on December 3, 2015

All you need to do to sign a single domain is:

```
user@www:~$ sudo letsencrypt -d www.example.org auth
```

For multiple domains (SAN) use:

```
user@www:~$ sudo letsencrypt -d www.example.org -d example.org auth
```

and if you have a compatible web server (Apache or Nginx), Let's Encrypt can not only get a new certificate, but also deploy it and configure your server automatically!:

```
user@www:~$ sudo letsencrypt -d www.example.org run
```

Involved parties

Internet Security Research Group (ISRG)
Electronic Frontier Foundation (EFF)
Mozilla Foundation
Akamai
Cisco Systems

IdenTrust
University of Michigan
Stanford Law School
Linux Foundation
...and some independent advisories

CERTIFICATE REVOCATION

- Certificates **can be in revoked meanwhile** they are **not expired**.
- **Every certificate should be checked** online against it's issuing CA every time is used: **revocation status and expiration date**.
- Revocation operations are **time consuming**. OS can cache status.
- **Revocation reasons:** unspecified, keyCompromise, CACompromise, affiliationChanged, superseded, cessationOfOperation, certificateHold, removeFromCRL, privilegeWithdrawn, AACompromise

CERTIFICATE REVOCATION

- **Certificate Revocation List (CRL)** is a simple mechanism to check the status of every certificate: each certificate authority maintains and **periodically** publishes a **list of revoked certificate serial numbers**.
- Anyone attempting to verify a certificate is then able to download the revocation list and check the presence of the serial number
- The CRL file itself can be published periodically via HTTP, or any other file transfer protocol. The CRL is signed by the CA.
- The growing number of revocations means that the CRL list will get larger (only not expired), and each client must retrieve the entire list of serial numbers.
- There is no mechanism for instant notification of certificate revocation—if the CRL was cached by the client before the certificate was revoked, then the CRL will be considered the revoked certificate valid until the cache expires.

CERTIFICATE REVOCATION

- **Online Certificate Status Protocol (OCSP)** provides a mechanism to perform a **real-time check for status** of the certificate
- Allows the verifier to query the certificate database directly for just the serial number in question while validating the certificate chain.
- Consume much less bandwidth and is able to provide real-time validation
- The CA must be able to **handle the load** of the real-time queries.
- The CA must ensure that the **service** is up and **globally available** at all times.
- The client must block on OCSP requests before proceeding with the navigation.

CERTIFICATE REVOCATION

- In practice, CRL and OCSP mechanisms are complementary, and most certificates will provide instructions and endpoints for both.

The screenshot shows a certificate details window for 'GlobalSign Root CA'. The breadcrumb trail indicates the path: GlobalSign Root CA > GlobalSign Organization Validation CA - SHA256 - G2 > *.wikipedia.org. The main content area lists three extensions:

Extension	Certificate Policies (2.5.29.32)
Critical	NO
Policy ID #1	(2.23.140.1.2.2)
Qualifier ID #1	Certification Practice Statement (1.3.6.1.5.5.7.2.1)
CPS URI	https://www.globalsign.com/repository/
Extension	CRL Distribution Points (2.5.29.31)
Critical	NO
URI	http://crl.globalsign.com/gc/gcorganizationvalsha2g2.crl
Extension	Certificate Authority Information Access (1.3.6.1.5.5.7.1.1)
Critical	NO
Method #1	CA Issuers (1.3.6.1.5.5.7.48.2)
URI	http://secure.globalsign.com/cacert/gcorganizationvalsha2g2r1.crt
Method #2	Online Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
URI	http://ocsp2.globalsign.com/gcorganizationvalsha2g2

At the bottom, the 'Fingerprint' section is partially visible.

Handwritten annotations in orange include 'CRL' with a vertical bar pointing to the CRL Distribution Points extension, and 'OCSP' with a vertical bar pointing to the Online Certificate Status Protocol method entry.

CERTIFICATE STORES

- Certificate Store can be thought of as a **logical container** in that holds one or more certificates.
- Contains **list of trusted Root CAs** and Intermediate CAs and manually trusted certificates.
- Can be **OS based or application based** (firefox, java)

CERTIFICATE STORES

Keychain Access

Click to unlock the System Roots keychain.

Search

Keychains

- login
- iCloud
- System
- System Roots**

Category

- All Items
- Passwords
- Secure Notes
- My Certificates
- Keys
- Certificates**

AAA Certificate Services
Root certificate authority
Expires: Monday 1 January 2029 at 00 h 59 min 59 s Central European Standard Time
This certificate is valid

Name	Kind	Expires	Keychain
AAA Certificate Services	certificate	01 Jan 2029 00:59:59	System Roots
Actalis Authentication Root CA	certificate	22 Sep 2030 13:22:02	System Roots
AddTrust Class 1 CA Root	certificate	30 May 2020 12:38:31	System Roots
AddTrust External CA Root	certificate	30 May 2020 12:48:38	System Roots
AddTrust Public CA Root	certificate	30 May 2020 12:41:50	System Roots
AddTrust Qualified CA Root	certificate	30 May 2020 12:44:50	System Roots
Admin-Root-CA	certificate	10 Nov 2021 08:51:07	System Roots
AdminCA-CD-T01	certificate	25 Jan 2016 13:36:19	System Roots
AffirmTrust Commercial	certificate	31 Dec 2030 15:06:06	System Roots
AffirmTrust Networking	certificate	31 Dec 2030 15:08:24	System Roots
AffirmTrust Premium	certificate	31 Dec 2040 15:10:36	System Roots
AffirmTrust Premium ECC	certificate	31 Dec 2040 15:20:24	System Roots
ANF Global Root CA	certificate	05 Jun 2033 19:45:38	System Roots
Apple Root CA	certificate	09 Feb 2035 22:40:36	System Roots
Apple Root CA - G2	certificate	30 Apr 2039 20:10:09	System Roots
Apple Root CA - G3	certificate	30 Apr 2039 20:19:06	System Roots
Apple Root Certificate Authority	certificate	10 Feb 2025 01:18:14	System Roots
Application CA G2	certificate	31 Mar 2016 16:59:59	System Roots
ApplicationCA	certificate	12 Dec 2017 16:00:00	System Roots
ApplicationCA2 Root	certificate	12 Mar 2033 16:00:00	System Roots
Autoridad de Certificacion Firmaprofesional CIF A62634068	certificate	31 Dec 2030 09:38:15	System Roots
Autoridad de Certificacion Raiz del Estado Venezolano	certificate	18 Dec 2030 00:59:59	System Roots
Baltimore CyberTrust Root	certificate	13 May 2025 01:59:00	System Roots
Belgium Root CA2	certificate	15 Dec 2021 09:00:00	System Roots
Buypass Class 2 CA 1	certificate	13 Oct 2016 12:25:09	System Roots
Buypass Class 2 Root CA	certificate	26 Oct 2040 10:38:03	System Roots
Buypass Class 3 Root CA	certificate	26 Oct 2040 10:28:58	System Roots

181 items

CERTIFICATE STORES

Console1 - [Console Root\Certificates (Local Computer)\Trusted Root Certification Authorities\Certificates]

File Action View Favorites Window Help

Console Root

- Certificates (Local Computer)
 - Personal
 - Certificates
 - Trusted Root Certification Authorities
 - Certificates
 - Enterprise Trust
 - Intermediate Certification Authorities
 - Certificate Revocation List
 - Certificates
 - Trusted Publishers
 - Untrusted Certificates
 - Third-Party Root Certification Authorities
 - Trusted People
 - Remote Desktop
 - Certificates
 - Certificate Enrollment Requests
 - Smart Card Trusted Roots
 - Trusted Devices

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name
AddTrust External CA Root	AddTrust External CA Root	30/05/2020	Server Authenticatio...	USERTrust
Baltimore CyberTrust Root	Baltimore CyberTrust Root	13/05/2025	Server Authenticatio...	Baltimore Cyb
Certum CA	Certum CA	11/06/2027	Server Authenticatio...	Certum
Class 3 Public Primary Certifica...	Class 3 Public Primary Certification A...	02/08/2028	Secure Email, Client ...	VeriSign Class
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	31/12/1999	Time Stamping	Microsoft Time
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	10/11/2031	Server Authenticatio...	DigiCert
DigiCert Global Root CA	DigiCert Global Root CA	10/11/2031	Server Authenticatio...	DigiCert
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root CA	10/11/2031	Server Authenticatio...	DigiCert
Entrust Root Certification Auth...	Entrust Root Certification Authority	27/11/2026	Server Authenticatio...	Entrust
Entrust.net Certification Autho...	Entrust.net Certification Authority (...)	24/07/2029	Server Authenticatio...	Entrust (2048
Equifax Secure Certificate Aut...	Equifax Secure Certificate Authority	22/08/2018	Secure Email, Server...	GeoTrust
GeoTrust Global CA	GeoTrust Global CA	21/05/2022	Server Authenticatio...	GeoTrust Glob
GeoTrust Primary Certification ...	GeoTrust Primary Certification Autho...	17/07/2036	Server Authenticatio...	GeoTrust
GlobalSign	GlobalSign	15/12/2021	Server Authenticatio...	GlobalSign
GlobalSign Root CA	GlobalSign Root CA	28/01/2028	Server Authenticatio...	GlobalSign
Go Daddy Class 2 Certification ...	Go Daddy Class 2 Certification Auth...	29/06/2034	Server Authenticatio...	Go Daddy Cla
Go Daddy Root Certificate Aut...	Go Daddy Root Certificate Authority...	01/01/2038	Server Authenticatio...	Go Daddy Ro
GTE CyberTrust Global Root	GTE CyberTrust Global Root	14/08/2018	Secure Email, Client ...	GTE CyberTru
macario-CA	macario-CA	14/06/2019	<All>	<None>
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root Au...	01/01/2000	Secure Email, Code S...	Microsoft Aut
Microsoft Root Authority	Microsoft Root Authority	31/12/2020	<All>	Microsoft Roo
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authority	10/05/2021	<All>	Microsoft Roo
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authority ...	24/06/2035	<All>	Microsoft Roo
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authority ...	23/03/2036	<All>	Microsoft Roo
NO LIABILITY ACCEPTED, (c)9...	NO LIABILITY ACCEPTED, (c)97 Veri...	08/01/2004	Time Stamping	VeriSign Time
Starfield Class 2 Certification A...	Starfield Class 2 Certification Authority	29/06/2034	Server Authenticatio...	Starfield Clas
Starfield Root Certificate Auth...	Starfield Root Certificate Authority -...	01/01/2038	Server Authenticatio...	Starfield Root
StartCom Certification Authority	StartCom Certification Authority	17/09/2036	Server Authenticatio...	StartCom Cer
Thawte Premium Server CA	Thawte Premium Server CA	01/01/2021	Server Authenticatio...	thawte
thawte Primary Root CA	thawte Primary Root CA	17/07/2036	Server Authenticatio...	thawte
thawte Primary Root CA - G3	thawte Primary Root CA - G3	02/12/2037	Server Authenticatio...	thawte Primar
Thawte Timestamping CA	Thawte Timestamping CA	01/01/2021	Time Stamping	Thawte Times
UTN - DATACorp SGC	UTN - DATACorp SGC	24/06/2019	Server Authentication	USERTrust
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Certifi...	17/07/2036	Code Signing, Server...	VeriSign

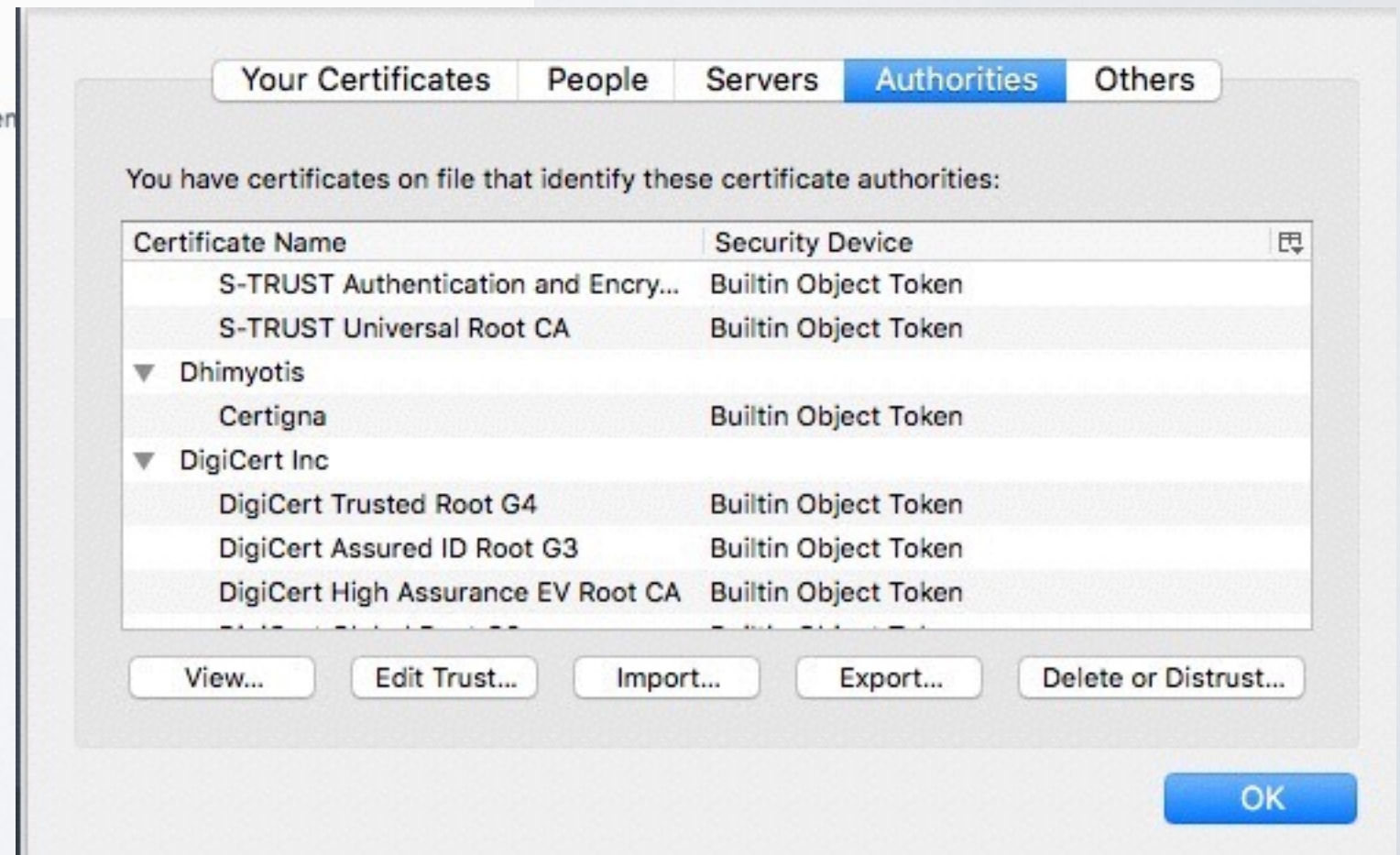
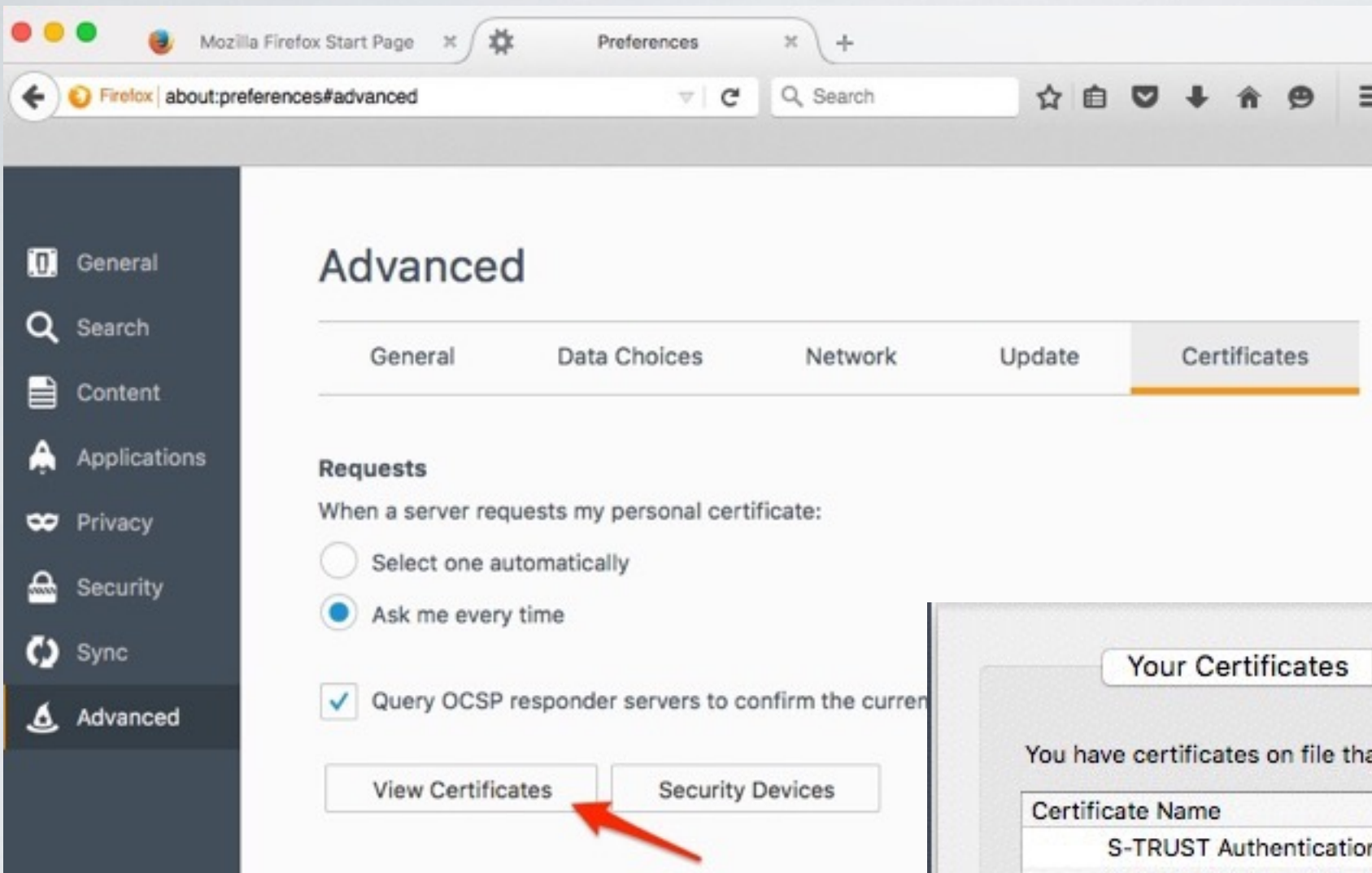
Trusted Root Certification Authorities store contains 35 certificates.

CERTIFICATE STORES

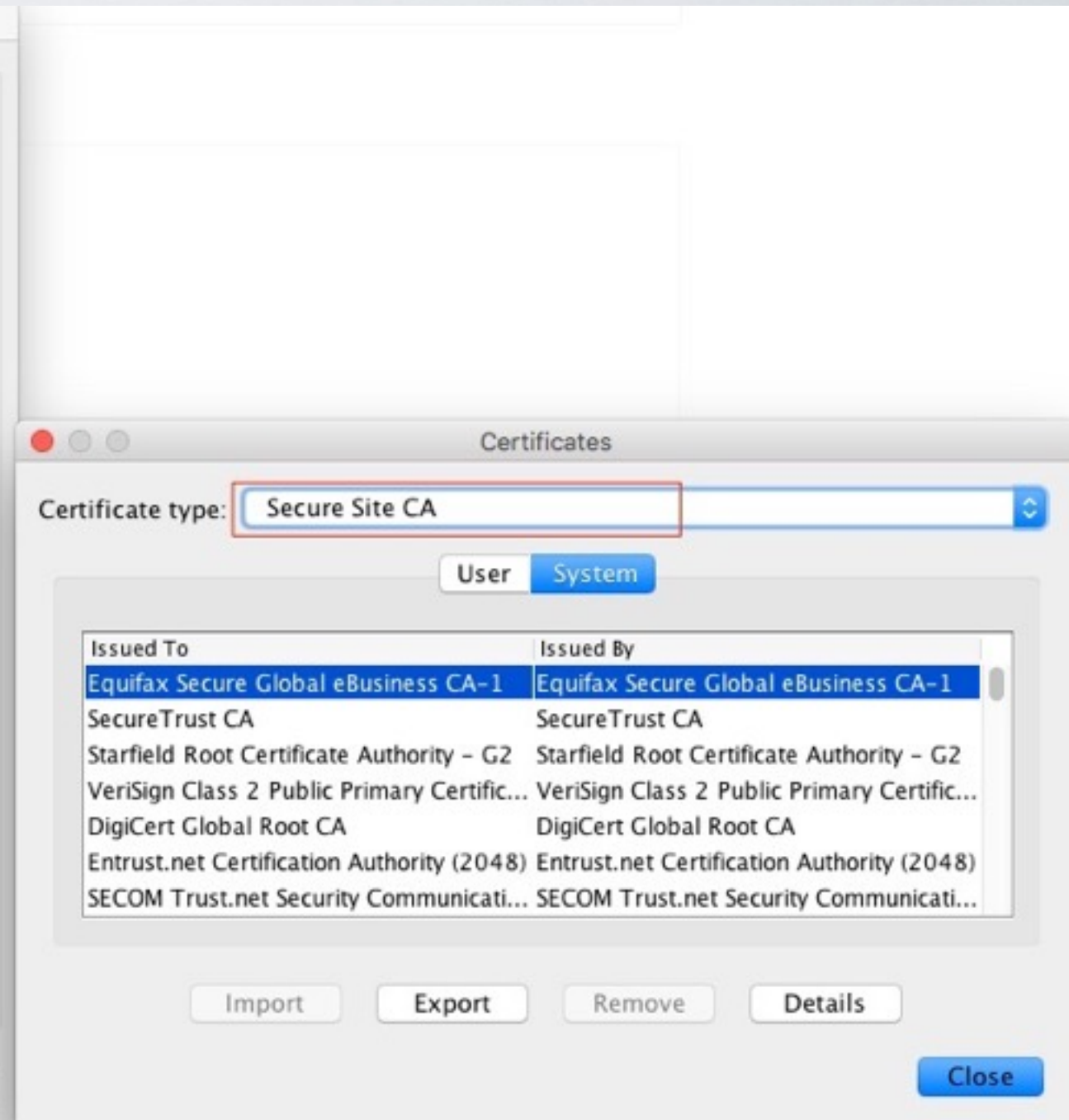
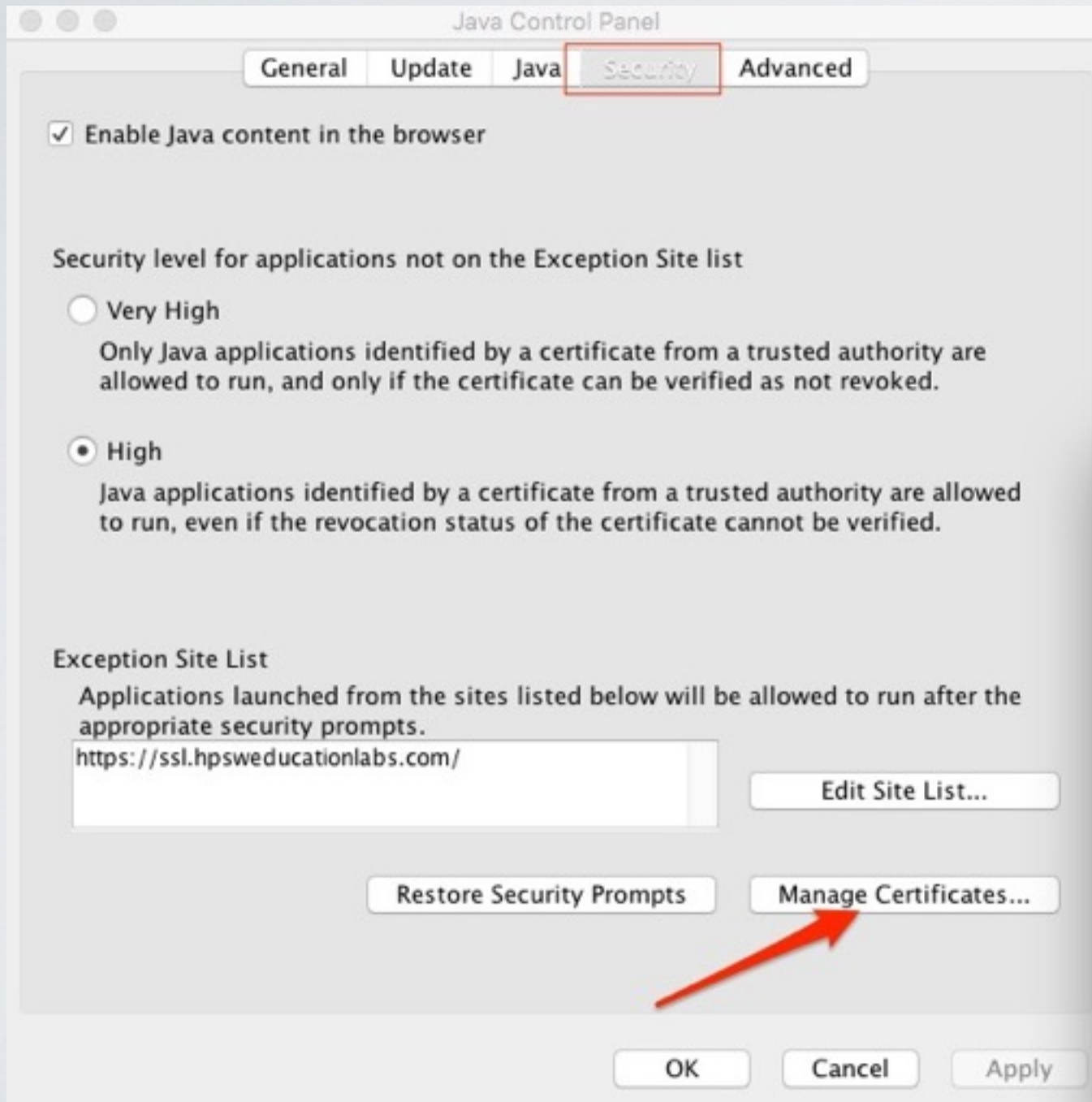
```
tree /etc/pki/tls/
/etc/pki/tls/
|-- cert.pem -> certs/ca-bundle.crt
|-- certs
|   |-- ca-bundle.crt
|   |-- ca-bundle.trust.crt
|   |-- localhost.crt
|   |-- make-dummy-cert
|   |-- Makefile
|   `-- renew-dummy-cert
-- misc
|   |-- CA
|   |-- c_hash
|   |-- c_info
|   |-- c_issuer
|   `-- c_name
-- openssl.cnf
-- private
    |-- 
    `--
```

/etc/pki/tls/certs/ holds the trusted certificates, issued by Private PKI certificates can be copied there to be trusted.
ca-bundle.crt holds certs regular public certificates
ca-bundle.trust.crt holds certs with "extended validation".

CERTIFICATE STORES



CERTIFICATE STORES



CERTIFICATE FILE TYPES

- **Base64-encoded X.509 Certificate** (.cer or .crt)
 - The Base64 format supports storage of a single certificate. This format does not support storage of the private key or certification path. They are Base64 encoded ASCII files. The encoded string is enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----".
- **DER-encoded binary X.509 Certificate** (.cer, .der or .crt)
 - The Distinguished Encoding Rules (DER) format supports storage of a single certificate. This format does not support storage of the private key or certification path.
- **Cryptographic Message Syntax Standard (PKCS#7) Certificate** (.p7b, .p7r or .spc)
 - PKCS #7 format supports storage of certificates and all certificates in the certification path. A PKCS #7 file typically has a .p7b file name extension, but this is not always the case. This again doesn't support storage of private keys. It is generally used by the CA to provide certificate chain to clients.

CERTIFICATE FILE TYPES

- **Personal Information Exchange Format (PKCS#12) Certificate (.pfx or .p12)**

- The Personal Information Exchange format (PFX, also called PKCS #12) defines a file format that can be used for secure storage of certificates (containing both private and public keys), and all certificates in a certification path, protected with a password-based symmetric key. PFX is a predecessor to PKCS#12.
- The PKCS #12 format is the only file format that can be used to export a certificate and its private key.

- **Privacy-enhanced Electronic Mail (.pem)**

- PEM format is a refinement of base64 encoding. This file format is typically used by OpenSSL to make Private Key available from a .pfx/.p12 file. So this is more widely used in the UNIX/LINUX world and not much in Windows.
- It is very similar to the PFX file format and can contain any/all information in one single file
- A single PEM file can also be split into multiple PEM files each containing a part of the original PEM file.

- **Private Key(.key)**

- Contains the private key of the certificate. On Windows there is no mechanism available to extract only the private key from the certificate, as it is not required. However, OpenSSL allows only the Private Key to be extracted from the certificate. IS Base-64 encoded string enclosed between "-----BEGIN RSA PRIVATE KEY-----" and "-----END RSA PRIVATE KEY-----".

CERTIFICATE FILE TYPES

- **Certificate Signing Request (.csr)**

- Used by applications to submit requests to the Certification Authority or CA. The request can be base64 encoded as shown below and is enclosed between "-----BEGIN NEW CERTIFICATE REQUEST-----" and "-----END NEW CERTIFICATE REQUEST-----".

The screenshot displays the OpenSSL CSR Wizard interface, which is divided into two main sections: "Certificate Details" and "Information".

Certificate Details: This section contains several input fields for specifying the certificate information:

- Common Name:** `www.cesarsaez.es`
- Organization:** `cesar saez`
- Department:** `acme`
- City:** `Valencia`
- State / Province:** `Valencia`
- Country:** `Spain` (selected from a dropdown menu)
- Key Size:** `RSA 2048 (recommended)` (selected from a dropdown menu)

A **Generate** button is located at the bottom right of the "Certificate Details" section.

Information: This section provides instructions and the command to generate the CSR:

Now just copy and paste this command into a terminal session on your server. Your CSR will be written to `www_cesarsaez_es.csr`.

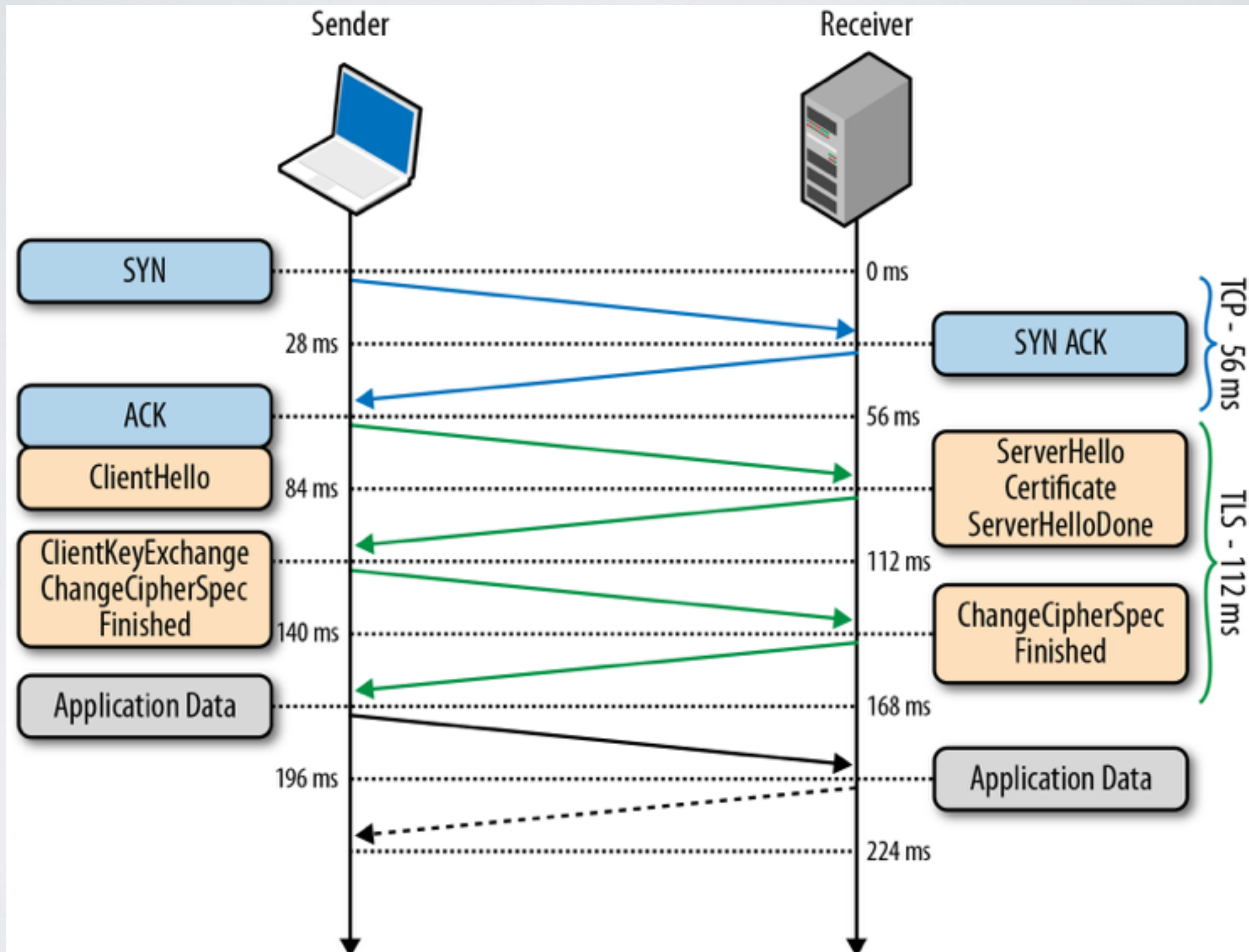
```
openssl req -new -newkey rsa:2048 -nodes -out www_cesarsaez_es.csr -keyout www_cesarsaez_es.key -subj "/C=ES/ST=Valencia/L=Valencia/O=cesar saez/OU=acme/CN=www.cesarsaez.es"
```

OpenSSL CSR Wizard <https://www.digicert.com/easy-csr/openssl.htm>

TLS HANDSHAKE

- Before the client and the server can begin exchanging application data over TLS, the encrypted tunnel must be negotiated
- The client and the server must agree on the version of the TLS protocol.
- Choose the cipher suite.
- Verify certificates.
- Each of these steps requires new packet roundtrips between the client and the server, which adds startup latency to all TLS connections.

TLS HANDSHAKE



Example assumes the same 28 millisecond one-way "light in fiber" delay between New York and London

TLS HANDSHAKE OPTIMIZING

- New TLS connections require two roundtrips for a "full handshake" and CPU resources to verify and compute the parameters for the session.
- Techniques available to avoid repeat the "full handshake" in every case:
 - **TLS Session Resumption:** If the client has previously communicated with the server, an "abbreviated handshake" can be used, which requires one roundtrip and allows the client and server to reduce the CPU overhead by reusing the previously negotiated parameters for the secure session
 - **False Start:** is an optional TLS protocol extension that allows the client and server to start transmitting encrypted application data when the handshake is only partially complete—i.e. once ChangeCipherSpec and Finished messages are sent, but without waiting for the other side to do the same. This optimization reduces new handshake overhead to one roundtrip.
- For best results, both optimizations should be used together to provide a single roundtrip handshake for new and returning visitors.

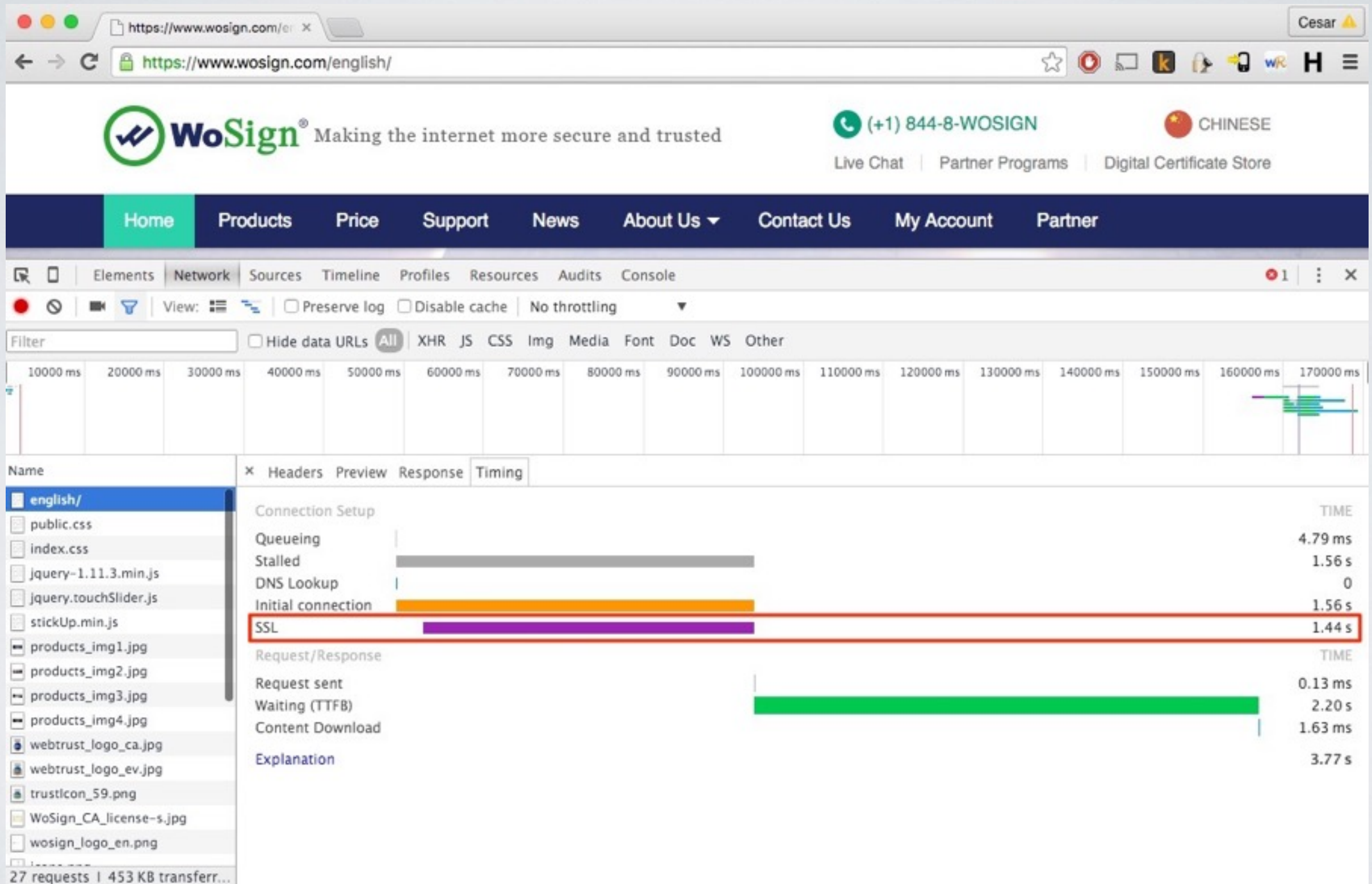
TLS PERFORMANCE

- Online available book “High Performance Browser Networking” chapter dedicated to “Transport Layer Security (TLS)”
1. Get best performance from TCP. Terminate TLS sessions closer to the user to minimize roundtrip latencies.
 2. Use dynamic TLS record sizing to optimize latency and throughput.
 3. Upgrade TLS libraries to latest release, and (re)build servers against them.
 4. Enable and configure session caching and stateless resumption.
 5. Configure forward secrecy ciphers to enable TLS False Start.
 6. Remove unnecessary certificates from your chain; minimize the depth.
 7. Configure OCSP stapling on your server.
 8. Disable TLS compression on your server.
 9. Do not use too much security (4096?)
 10. Use persistent connections (HTTP 1.1)
 11. Application design (avoid mixed content)

TLS PERFORMANCE

- Follow Mozilla's Server Side TLS Guidelines.
- Use “Mozilla SSL Configuration Generator” to config your webserver
- Monitor **frequently** your site with Qualys SSL Test Server <https://www.ssllabs.com/ssltest/>

TLS PERFORMANCE



TLS PERFORMANCE

- Check SSL handshake with curl
- `time_appconnect` : The time, in seconds, it took from the start until the SSL/SSH/etc connect/handshake to the remote host was completed. (Added in 7.19.0)

```
$ curl -w "@curl-format.txt" -o /dev/null -s https://www.wosign.com/
      time_namelookup:  0.140
        time_connect:  0.700
      time_appconnect:  5.411
    time_pretransfer:  5.411
      time_redirect:   0.000
time_starttransfer:   6.846
      -----
            time_total:  6.846
```


CERTIFICATE SECURITY CONCERNS

- End users awareness about what SSL/TLS meanings.
- X.509 specification suffers from being over-functional and underspecified, information is spread across many documents from different standardization bodies.
- CAs are subject to the legal jurisdiction and may be legally compelled to compromise the interests of their customers and their users.
- CAs cannot technically restrict subordinate CAs from issuing certificates outside a limited namespaces or attribute set.
- Certificate private key safeguarding (export allowed).
- CA compromise.

CERTIFICATE SECURITY CONCERNS

- SSLv2 and SSLv3 protocols are deprecated, current versions will eventually be deprecated
- Ciphersuites deprecated (see Mozilla's Security/Server Side TLS guidelines)
- Certificate signature hash weakness (MD5, SHA-1)

CERTIFICATE SECURITY CONCERNS

- If 2 files are discovered to produce the same string from one-way hash like SHA-1, that's a "collision". Collisions are always theoretically possible, but they're supposed to be so rare as to be practically impossible.

war-and-peace.txt	(text/plain) - 3365836 bytes
MD5	78765f4f116bfe59fc52e3f7b0eee0d0
SHA1	baeb2c3a70c85d44947c1b92b448655273ce22bb
SHA256	ac44f7eb6f2a0199f2109ec441f34a706a300fb3f528e36b538bd60ce7d94cbe

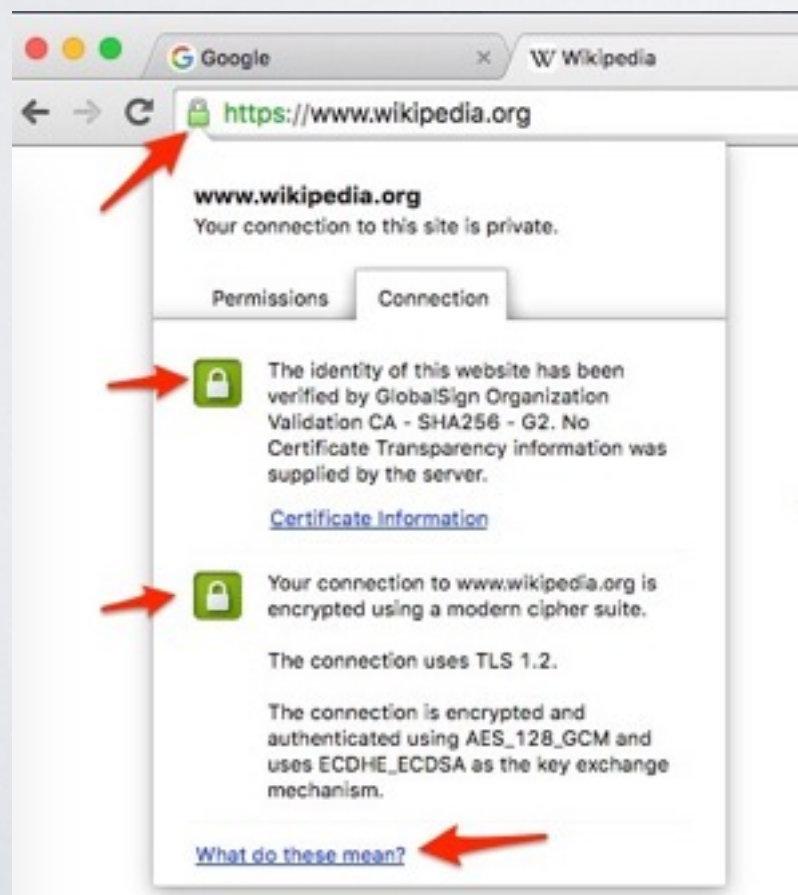
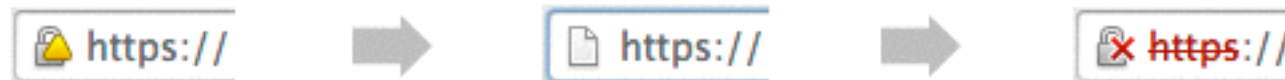
- Browser checks a certificate, calculate the SHA-1 for that certificate's information itself, and then compare it to the signed SHA-1 that the certificate offered as proof.
- **SHA-1** signature hash is weak, all major software developers agreed to **deprecate it on 1st January 2017**

CERTIFICATE SECURITY CONCERNS

- In 2005, cryptographers proved that SHA-1 could be cracked 2,000 times faster than predicted.
- The possibility to reproduce SHA-1 signature from major CA means that an attacker can sign fake CA certificates.
- In 2012 was revealed that Flame malware used same concept, sign code with faked certificate (using MD5 hash)
- **Update to new certificate using SHA-256** is the solution. Not late, but not forget to do it.

CERTIFICATE SECURITY CONCERNS

- All CA offer new certificate with SHA-256 hash by default.
- Windows XP SP3 and Windows 2003 R2 SP2 support SHA2 with some restrictions.
- SHA-256 Compatibility status (updated end Oct) <https://support.globalsign.com/customer/portal/articles/1499561-sha-256-compatibility>



USEFUL DOCS

- Mozilla's Server Side TLS Guidelines https://wiki.mozilla.org/Security/Server_Side_TLS
- Mozilla SSL Configuration Generator <https://mozilla.github.io/server-side-tls/ssl-config-generator/>
- Online book “High Performance Browser Networking” from Ilya Grigorik <http://chimera.labs.oreilly.com/books/12300000000545/index.html>
- Book “Bulletproof SSL and TLS” from Ivan Ristić
- Book “Windows Server 2008 PKI” from Brian Kumar
- Wikipedia - Transport Layer Security https://en.wikipedia.org/wiki/Transport_Layer_Security
- Wikipedia - X.509 <https://en.wikipedia.org/wiki/X.509>
- Linkedin group PKI Professionals <https://www.linkedin.com/grp/home?gid=79441>

- A version of this presentation (with some extra slides) can be found at <http://valenciadevops.me>
- Feedback is welcome cesar@cesarsaez.es

Thanks!



<http://valenciadevops.me>