

Public keys as identities ?????

# How do I know if I should trust a website?



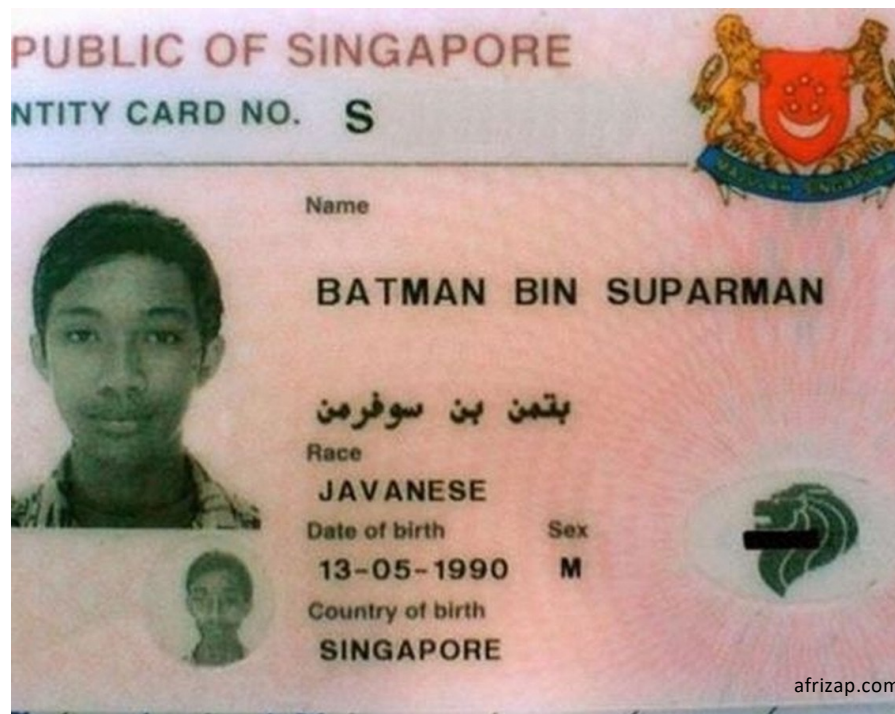
Badis HAMMI

370

# *Authentication: Public keys as identities*

---

- Why we ask to see an identity card ?
- Why we trust an identity card ?



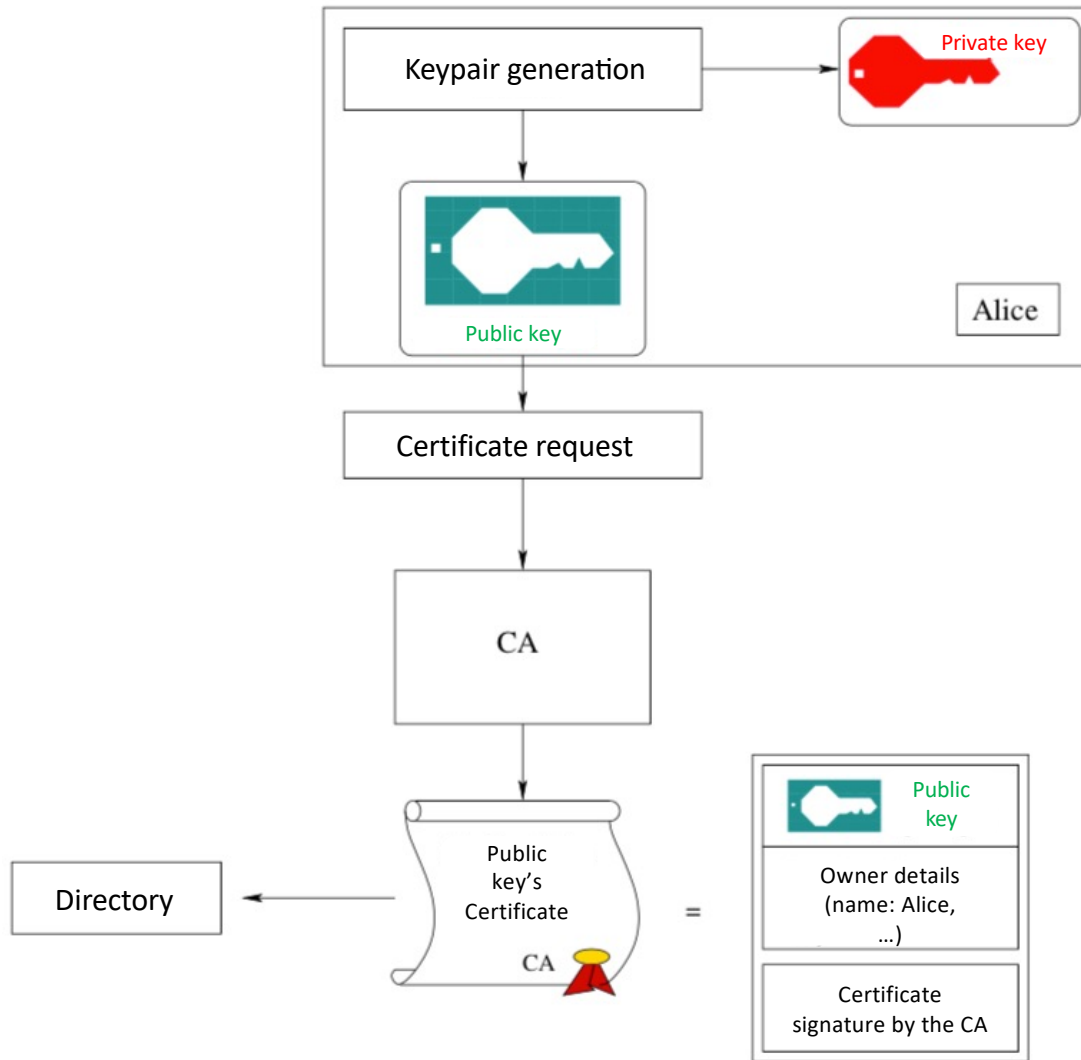
## *Authentication: Public Key Infrastructures*

---

A public key infrastructure (PKI) is a set of authorities, policies, and procedures needed to manage public-key mechanisms. Indeed, PKI is a set of authorities and protocols that binds public keys with respective identities of entities. The binding is established through a process of registration and issuance of certificates. Thus, a PKI creates, manages, distributes, uses, stores, and revokes these defined certificates.

- PKI does not distribute keys but certificates!
- A certificate contains a public key
- It also contains identity data
  - For one person: civil status, address, email ...
  - For a server: domain name, IP address, administrator email, etc.
- A certificate is validated by a trusted third party
  - Certification authority = CA

# Authentication: Public Key Infrastructures



- Alice generates her keys  $K_e$  and  $K_d$ 
  - $K_e$ : public key
  - $K_d$ : private key
- She makes a request to the CA for a  $K_e$  certificate
- CA validates the key, authenticates Alice and generates a certificate
  - the certificate is signed by the CA
  - This signature certifies the origin of the certificate & its integrity.
- The certificate is published in a public directory

# Authentication: Certificate

## Certificate

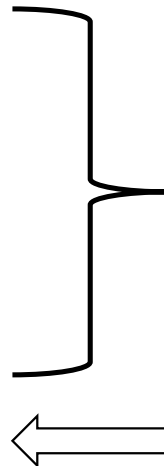
### Information


- Certification authority
  - Certificate owner
  - Email
  - Validity period
  - Public key
  - Algorithm
- 
- Signature

Hash



Encrypt  
with CA's  
private key





 **Safari utilise une connexion chiffrée à [www.amazon.fr](https://www.amazon.fr).**  
Le chiffrement avec un certificat numérique garantit la confidentialité des données lors de l'envoi et la réception depuis le site web <https://www.amazon.fr>.

DigiCert Global Root G2  
↳ DigiCert Global CA G2  
↳ [www.amazon.fr](https://www.amazon.fr)

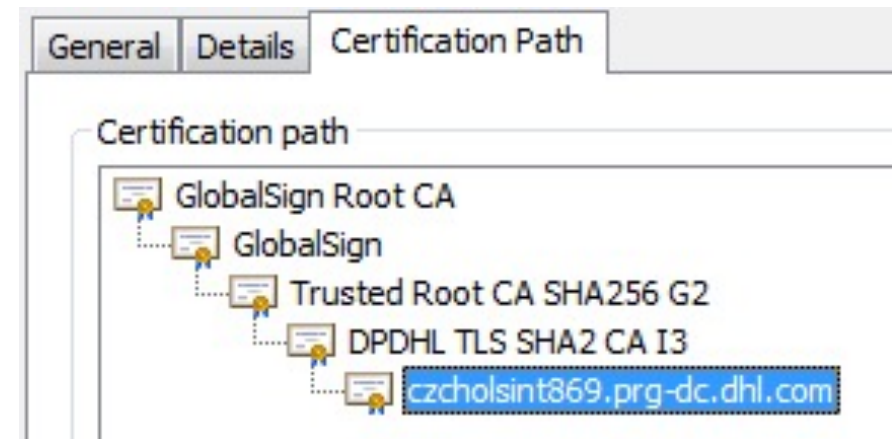
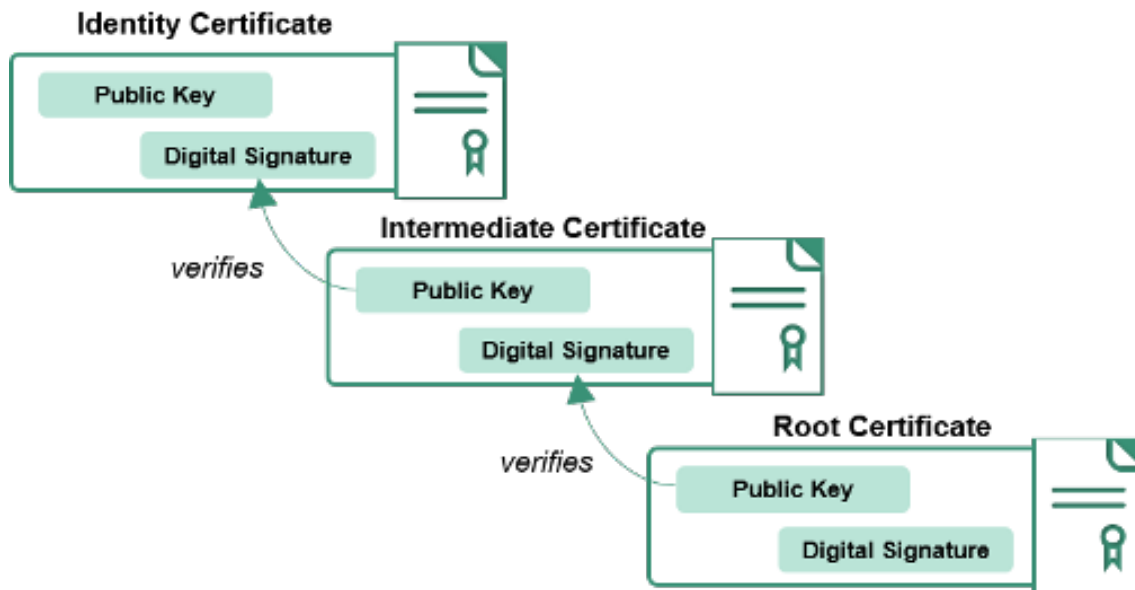
**ID de règles #2 ( 2.23.140.1.2.2 )**

Extension	Points de distribution CRL ( 2.5.29.31 )
Critique	NON
URI	<a href="http://crl3.digicert.com/DigiCertGlobalCAG2.crl">http://crl3.digicert.com/DigiCertGlobalCAG2.crl</a>
URI	<a href="http://crl4.digicert.com/DigiCertGlobalCAG2.crl">http://crl4.digicert.com/DigiCertGlobalCAG2.crl</a>
Extension	Liste d'horodatage de certificat signé et intégré ( 1.3.6.1.4.1.11129.2.4.2 )
Critique	NON
Version SCT	1
ID de clé de l'historique	BB D9 DF BC 1F 8A 71 B5 93 94 23 97 AA 92 7B 47 38 57 95 0A AB 52 E8 1A 90 96 64 36 8E 1E D1 85
Timestamp	samedi 5 mai 2018 à 01:00:45 heure d'été d'Europe centrale
Algorithme de signature	SHA-256 ECDSA
Signature	72 octets : 30 46 02 21 00 C4 28 1A ...
Extension	Accès aux informations de l'autorité du certificat ( 1.3.6.1.5.5.7.1.1 )
Critique	NON
Méthode #1	Protocole du statut du certificat en ligne ( 1.3.6.1.5.5.7.48.1 )
URI	<a href="http://ocsp.digicert.com">http://ocsp.digicert.com</a>
Méthode #2	Emetteurs du CA ( 1.3.6.1.5.5.7.48.2 )
URI	<a href="http://cacerts.digicert.com/DigiCertGlobalCAG2.crt">http://cacerts.digicert.com/DigiCertGlobalCAG2.crt</a>
Empreintes	
SHA-256	02 6F F9 36 DC 5B B6 ED 1D 7D 98 9C 68 31 CA 00 D0 86 E2 A3 84 90 1D 90 7F CB E3 6A 28 AA 66 D2
SHA-1	77 2A 9B B9 C2 6D 51 DD C1 38 16 0F 45 85 17 7D 01 76 A7 0B

 Masquer le certificat 

# Authentication: Public Key Infrastructures

- Each CA has its own certificate
  - The associated private key is used to sign the certificates issued by the CA
  - The CA's certificate is signed by another CA etc ...
  - ⇒ Certificate chain
- The last certificate of the chain is signed by itself
  - We talk about **self-signed** certificate or **root** certificate



# Authentication: Public Key Infrastructures

Gestionnaire de certificats

Vos certificats

Personnes

Serveurs

Autorités

Vous possédez des certificats enregistrés identifiant ces autorités de certification

Nom du certificat	Périphérique de sécurité
Certum Trusted Network CA	Builtin Object Token
Certum Trusted Network CA 2	Builtin Object Token
Certum Global Services CA SHA2	Sécurité personnelle
Yandex CA	Sécurité personnelle
▼ VeriSign, Inc.	
Verisign Class 1 Public Primary Certification A...	Builtin Object Token
Verisign Class 2 Public Primary Certification ...	Builtin Object Token
Verisign Class 3 Public Primary Certification ...	Builtin Object Token
VeriSign Class 3 Public Primary Certification ...	Builtin Object Token
VeriSign Universal Root Certification Authority	Builtin Object Token

Voir...

Modifier la confiance...

Importer...

Exporter...

Supprimer ou ne plus

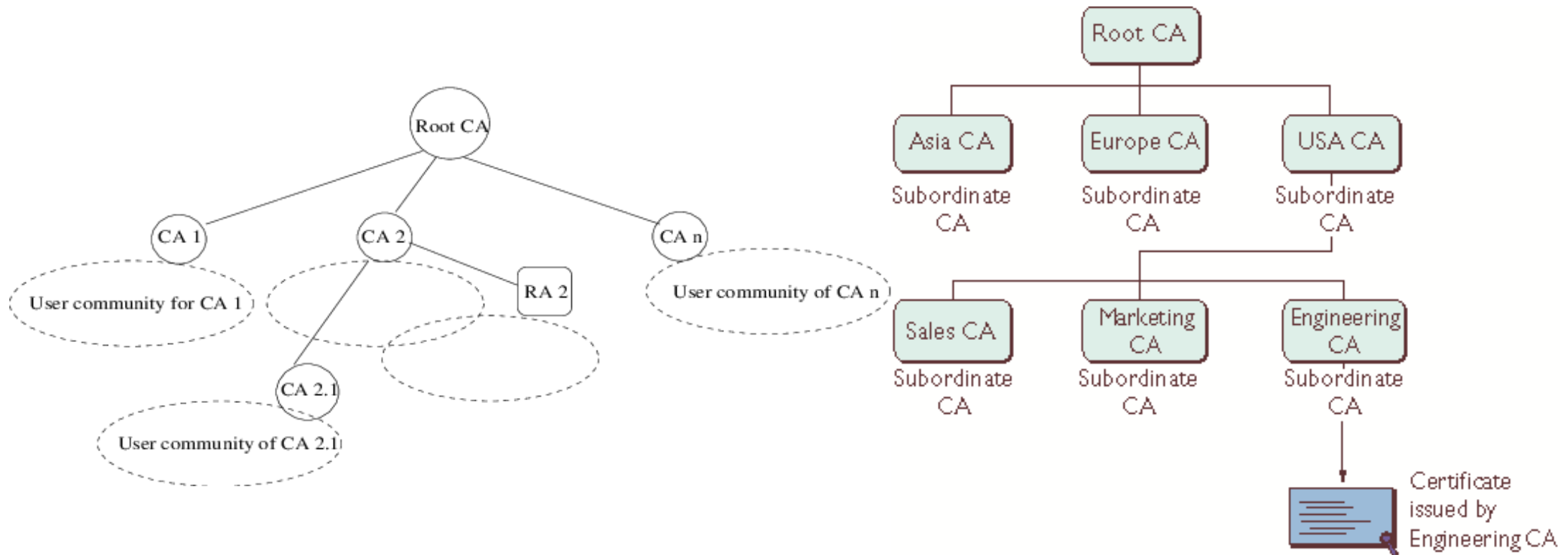


# *Authentication: Public Key Infrastructures*

---

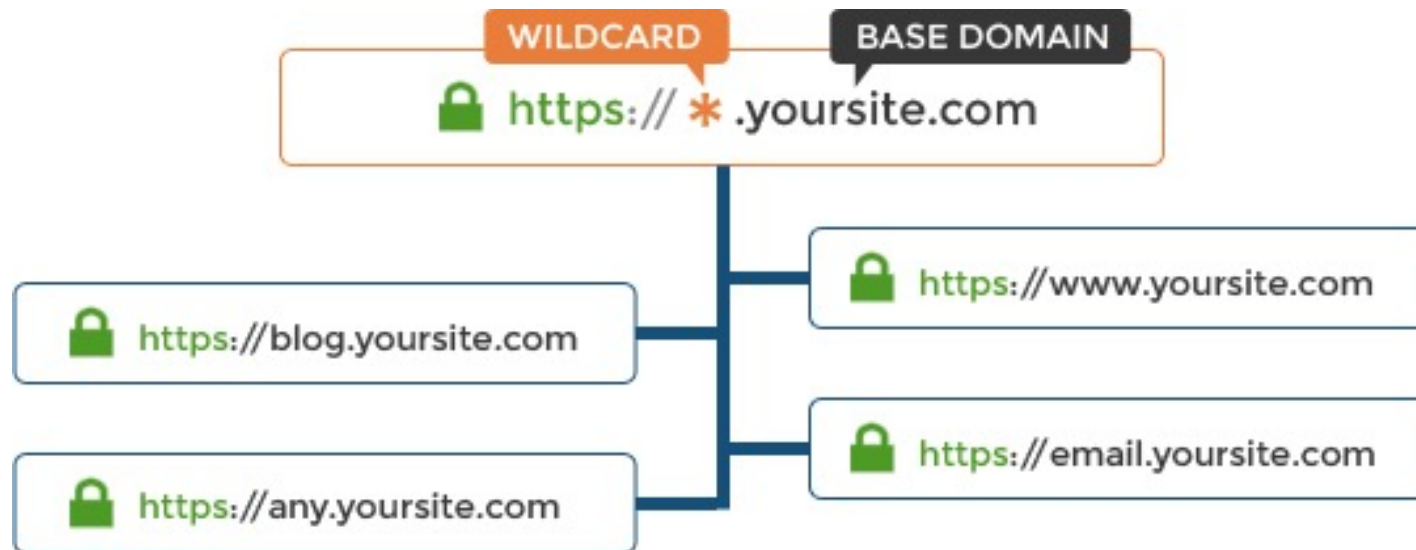
- There are several standards for PKIs
- Two types of infrastructure
  - hierarchical architectures
    - based on different CAs, which are separate from the users.
    - Ex: PKIX (Public Key Infrastructure X.509)
  - non-hierarchical architectures
    - each user has its own CA
    - originally designed to secure PGP and P2P messaging
    - mutual trust between users
    - Ex: SPKI (Simple Public Key Infrastructure, Spooky), SDSII (Simple Distributed Security Infrastructure or Sudsy)

# Authentication: Public Key Infrastructures



# Authentication: Public Key Infrastructures

---



# *Authentication: Public Key Infrastructures*

---

## ***PKI Actors***

### ***1. Certificate holder***

- entity that has a private key
- the digital certificate contains the associated public key
- several type of certificate: client, server, VPN etc ...

### ***2. Certificate user***

- get the certificate
- uses the public key in its transaction with the holder.

### ***3. The Certification Authority (CA)***

- Legal and moral entity of a PKI
- Set of resources defined by a name and a public key which:
  - generates certificates
  - sends and maintains information about CRL
  - publish certificates that have not yet expired
  - maintains the archives of expired / revoked certificates

# *Authentication: Public Key Infrastructures*

---

## **4. Registration Authority (RA)**

- Intermediary between the holder of the key and the CA
- Checks user requests
- Sends requests to the CA
  - The level of verification depends on the security policy
- Each CA has a list of accredited RAs
- An RA is known to a CA by a name and a public key.
- CA verifies the information of the RA by means of its signature

## **5. CRL transmitter**

# *Authentication: Public Key Infrastructures*

---

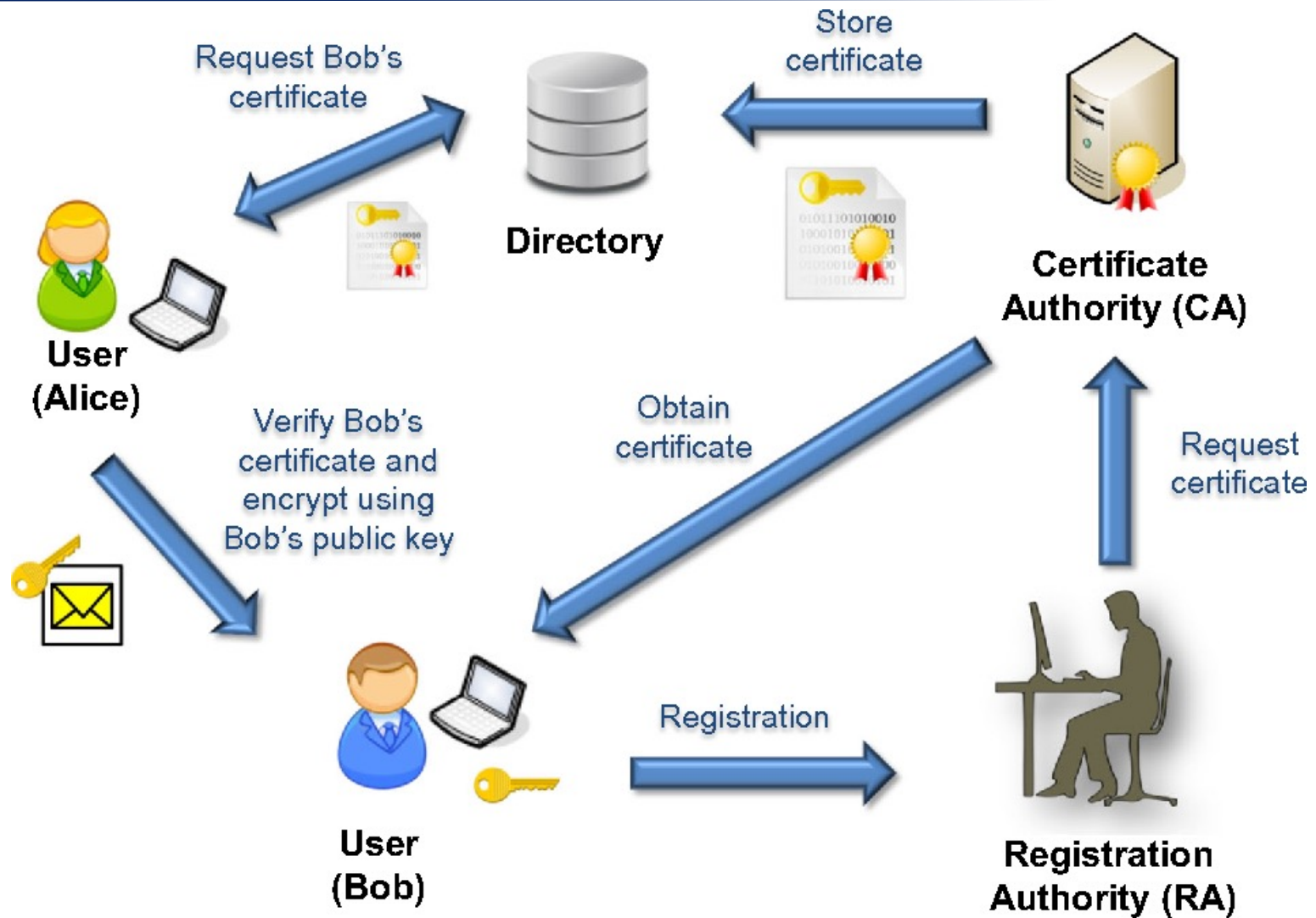
## **6. *Directory (Repository)***

- Distributes certificates and CRLs
- Known by its address and access protocol

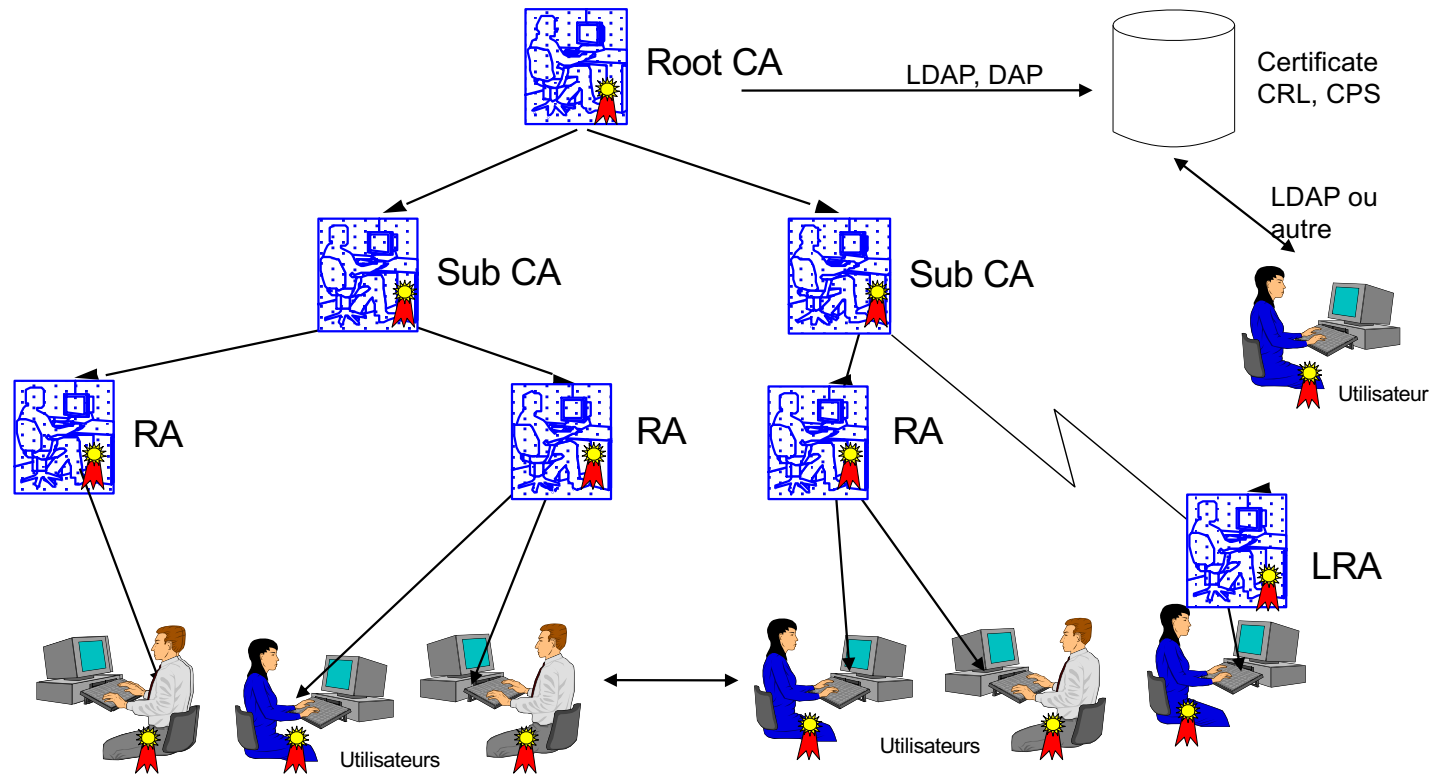
## **7. *Archive***

- long-term storage of information on behalf of a CA.
- allows to settle disputes
  - knowing which certificate was valid at a given time.

# Authentication: Public Key Infrastructures



# Authentication: Public Key Infrastructures





# *Authentication: Public Key Infrastructures*

---

X.509 is a standard defining the format of public key certificates.[1] X.509 certificates are used in many Internet protocols, including TLS/SSL, which is the basis for HTTPS. X.509 also defines certificate revocation lists.

X509 certificate fields:

Version	Indicates which version of X.509 this certificate corresponds to
Serial number	Certificate serial number (specific to each CA)
Signature Algo ID	Identifier of the type of signature used
Issuer Name	Distinguished Name (DN) of the CA issuing the certificate
Validity period	Validity period of the certificate (not before / not after)
Subject Name	Distinguished Name (DN) of the public key holder
Subject pub. key info	Information on the public key of this certificate
Issuer Unique ID	Unique identifier of the issuer of this certificate
Subject Unique ID	Unique identifier of the public key holder
Extensions	Optional generic extensions.
Signature	Digital signature of CA on previous fields

# *Authentication: Public Key Infrastructures*

---

Pgp?

# *Authentication: Public Key Infrastructures*

---

## **Security policy**

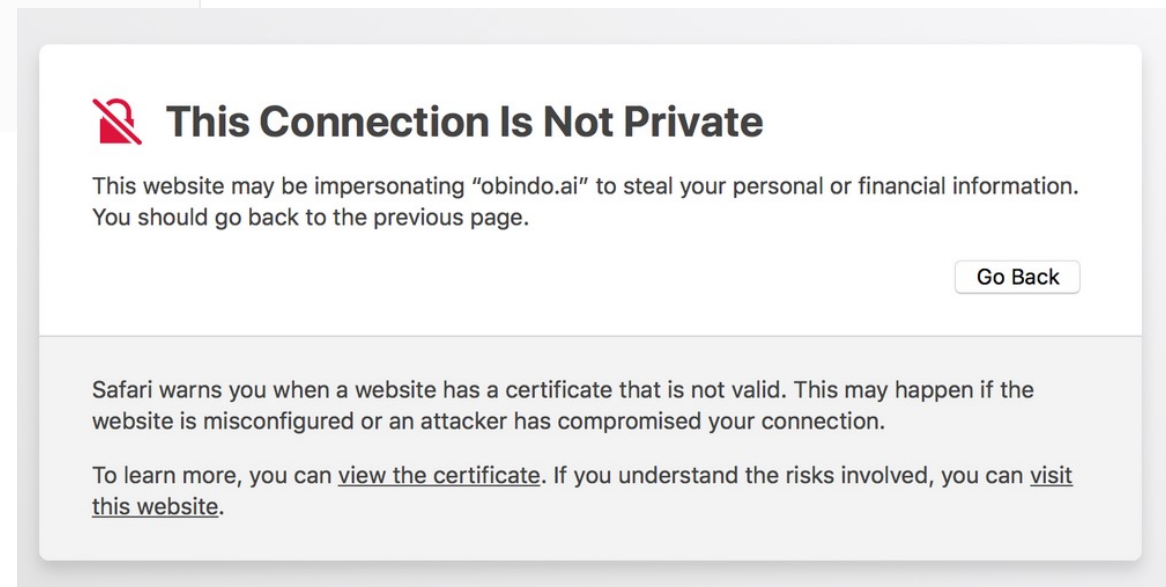
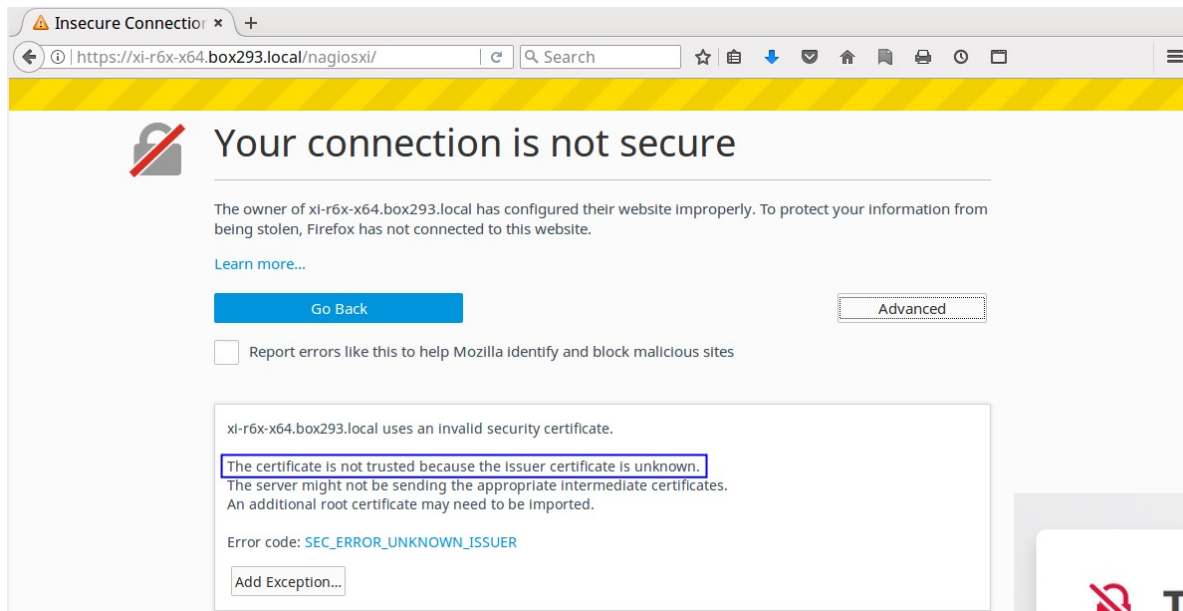
- Defines the overall strategy and responds to threats.
- In particular, it is fundamental to describe:
  - Who is responsible for what (implementation, execution, audit, tests, etc.)
  - What is the basic security policy for the computer network
  - Why everyone should do what they do

## *Authentication: Public Key Infrastructures*

---

How does the certificate verification process work?

# Authentication: Public Key Infrastructures



## *Authentication: Public Key Infrastructures*

---



# *Authentication: Public Key Infrastructures*

---

A certificate revocation list (or CRL) is "a list of digital certificates that have been revoked by the issuing certificate authority (CA) before their scheduled expiration date and should no longer be trusted"

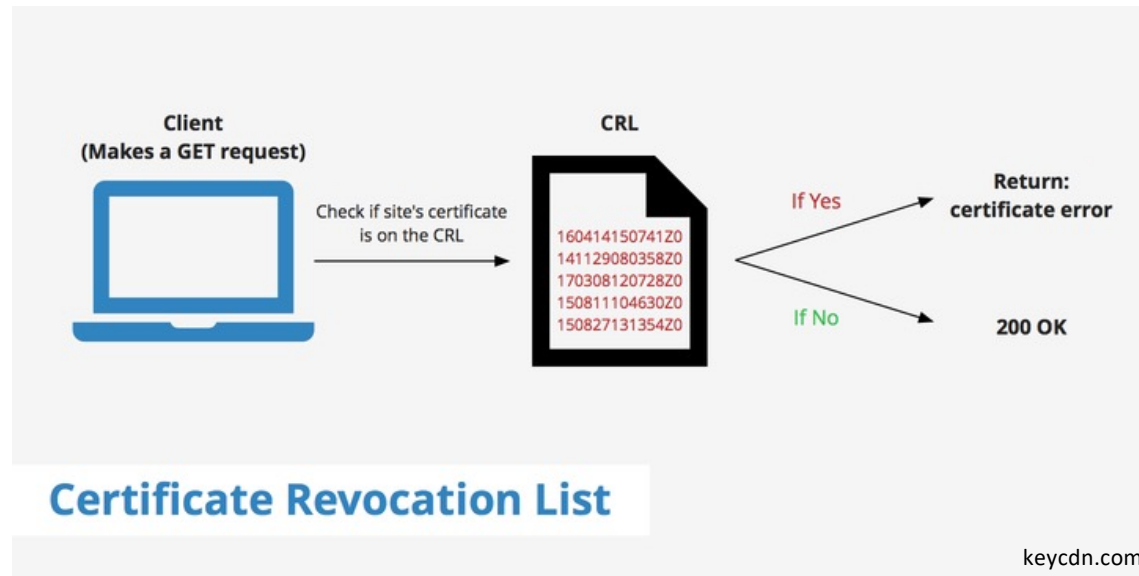
Reasons to revoke a certificate according to RFC 5280 (p69) are:

- unspecified
- keyCompromise
- cACompromise
- affiliationChanged
- superseded
- cessationOfOperation
- certificateHold
- removeFromCRL
- privilegeWithdrawn
- aACompromise

# Authentication: Public Key Infrastructures

## Certificate Revocation list (CRL)

A CRL (Certificate Revocation List) contains the list of revoked certificates, dated and signed by a CA and periodically published. To verify the validity of a certificate, the verifier must send a request to the publication server hosting the corresponding CRL, with as argument the identifier of the CA in charge of the certificate; it then receives the last CRL generated by the CA; he must then verify the signature of the CRL and its period of validity, and then search for the certificate in the CRL (Bekara et al.).





# Authentication: Public Key Infrastructures

---

## Format

```
Certificate Revocation List (CRL):
  Version 2 (0x1)
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: /C=US/O=Google Trust Services/CN=GTS CA 101
  Last Update: Mar 30 04:01:05 2020 GMT
  Next Update: Apr 9 04:01:05 2020 GMT
  CRL extensions:
    X509v3 CRL Number:
      2491
    X509v3 Authority Key Identifier:
      keyid:98:D1:F8:6E:10:EB:CF:9B:EC:60:9F:18:90:1B:A0:EB:7D:09:FD:2B

Revoked Certificates:
  Serial Number: 3304E7A12A4E0F02000000006075D1
  Revocation Date: Mar 29 16:21:50 2020 GMT
  Serial Number: 409A97C6F65A6202000000006075D2
  Revocation Date: Mar 29 16:21:50 2020 GMT
  Serial Number: BF52D1D1D1A31105000000004FDCEF
  Revocation Date: Mar 29 18:25:07 2020 GMT
  Serial Number: C0FB854CD42DEA03000000007E157D
  Revocation Date: Mar 28 17:25:07 2020 GMT
  Serial Number: CC1E8ADAD3EC4303000000007DE919
  Revocation Date: Mar 27 19:55:07 2020 GMT
  ...
  Signature Algorithm: sha256WithRSAEncryption
  5c:3f:4f:e9:18:28:2f:f3:e8:c2:e3:5e:35:0e:1c:95:8a:60:
  96:be:6a:55:3b:33:6a:39:db:22:3a:62:a0:5e:f1:69:5e:cd:
  eb:2a:14:3b:1b:c4:d3:3a:fc:49:8e:bd:1a:40:53:a5:38:24:
  7a:46:7e:a4:18:dd:3d:18:28:30:aa:39:a7:1e:b9:e6:77:09:
  ...
```

# *Authentication: Public Key Infrastructures*

---

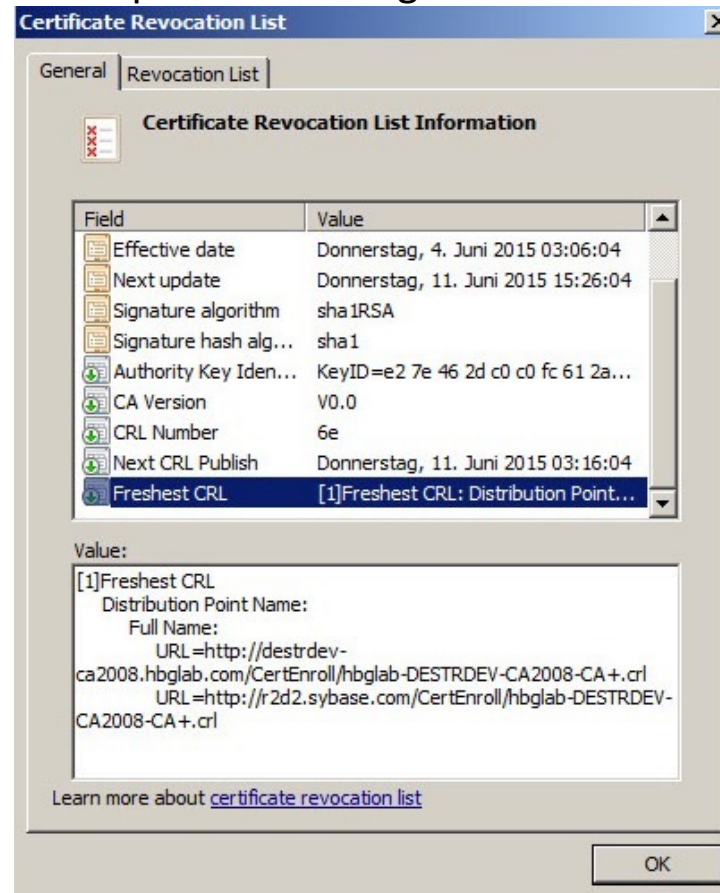
Demo read CRL

```
openssl crl -inform DER -text -noout -in theCrl.crl
```

# Authentication: Public Key Infrastructures

## Delta-CRL ( $\Delta$ -CRL)

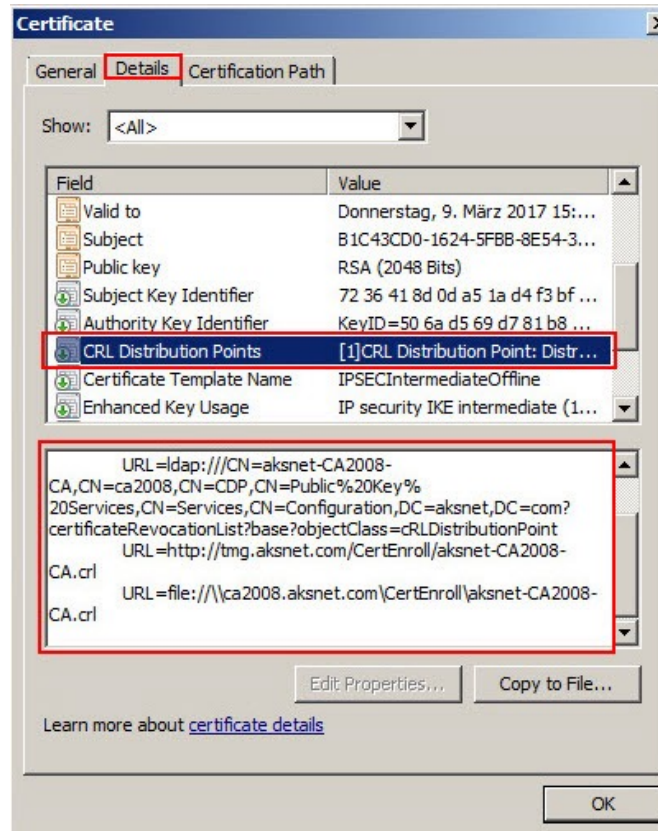
The  $\Delta$ -CRL is a signed list which contains all the certificates revoked since the publication of the last CRL (basic CRL). Thus, the verification of a certificate will require recovering both the basic CRL and the most recent  $\Delta$ -CRL (Bekara et al.).



# Authentication: Public Key Infrastructures

## CRL Distribution Points (CRL DP)

This method consists of dividing a CRL into segments, each containing a subset of the certificates revoked by a CA. Since any certificate contains a pointer to the corresponding segment, a verifier will be able to access the segment concerned directly. Thus the CRL DP method appears more extensible than the CRL.



# *Authentication: Public Key Infrastructures*

---

## ***Certificate Revocation Tree (CRT)***

The 2-3 CRT method uses a 2-3 order tree in which the leaves correspond to revoked certificates and are ordered in ascending order of serial numbers. The value of a tree node is the hash of the values of its children and the root value is signed by the CA to guarantee its authenticity.

A certificate is considered revoked if it appears as a leaf of the tree; a certificate will be considered valid if two certificates correspond to two adjacent leaves of the tree with for one, a serial number higher than the certificate's serial number and for the other a lower number (Bekara et al.).

# *Authentication: Public Key Infrastructures*

---

## **Certificate Revocation Status Directory (CRS Directory):**

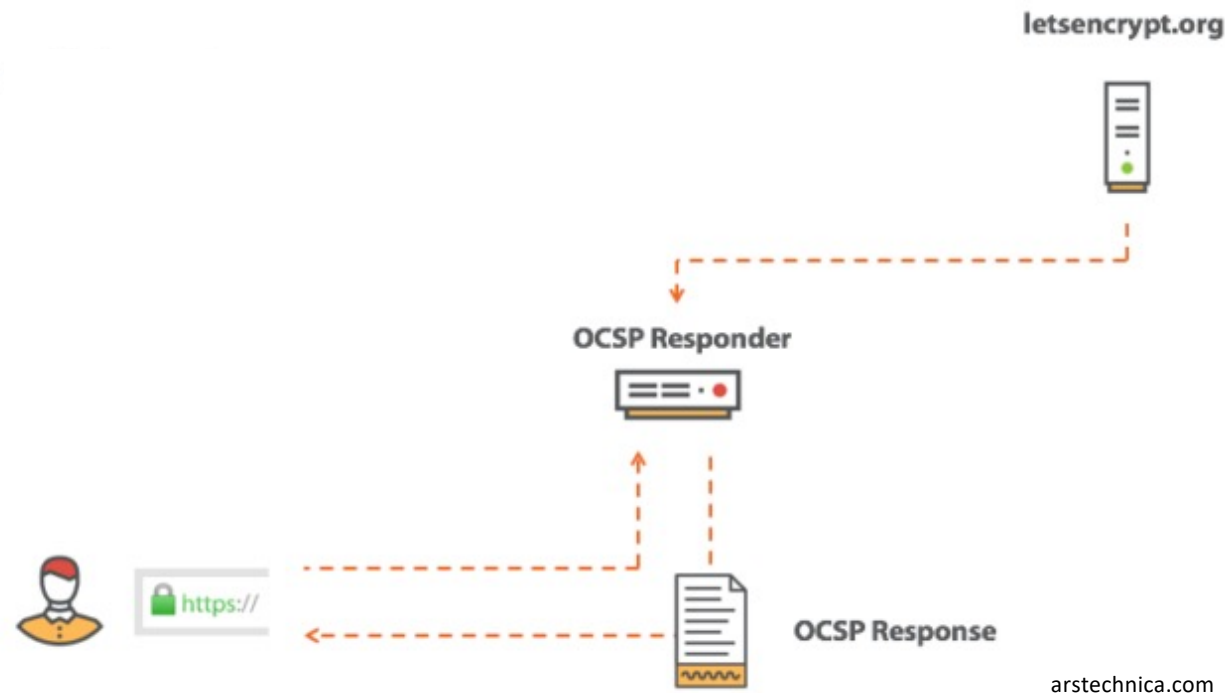
In a system that adopts the CRS technique, the certificate structure is extended with two additional fields of 100 bits each. Every day, the CA sends signed statements to the CRS Directory about the status of single issued certificates (to certificate users). There are signed statements for every non-expired certificate.

When a user inquires about a certificate revocation status, the CRS Directory replies with information which the user can use to verify the requested status. The CRS approach decreases the communication load between the server and end entities which makes it achieve an overall performance gain compared to CRL approach. However, it considerably increases the communication load between the server and the CA.

# Authentication: Public Key Infrastructures

## Online Certificate Status Protocol (OCSP)

OCSP is an IETF standard (RFC2560) - introduces a trusted OCSP server which will be requested by the verifiers to check the validity status of a given certificate ('good', 'revoked', 'unknown' ). The verifier who has knowledge of the server certificate must verify the authenticity of the messages signed and returned by the server. The OCSP servers can be dissociated from the ACs, or even co-located with the AC, in which case the AC and the server share the same directories, which guarantees optimal freshness of the revocation information.



# *Authentication: Public Key Infrastructures*

---

## Online Certificate Status Protocol Drwbacks:

- The approach is **centralized**. Thus, represents a single point of failure
- The OCSP responder verifies the revocation status of a certificate **without checking the validity of its serial number** and if it belongs to the CA. Thus, a malicious user can flood the server with verification requests for certificates that do not belong to the CA, making the server work intensively which can cause its denial of service
- it was proven that OCSP lookups are costly, especially in time, which increases the client side **latency**
- OCSP is an **on-line** scheme which makes it ineffective for offline systems
- OCSP may provide real-time responses to revocation queries, however it is unclear whether the responses actually contain **updated revocation information**. Some OCSP responders may rely on **cached CRLs** on their backend
- the OCSP approach introduces a **privacy risk**. Indeed, the OCSP responders know which certificates are being verified by end users and they can therefore track the sites a user is visiting



# *Authentication: Public Key Infrastructures*

---

## **OCSP stapling:**

In this solution, the web server itself requests OCSP validation which it passes as a response to inquiring clients. Stapling removes the latency involved with OCSP validation because the client does not need an additional round trip to communicate with the OCSP responder to check the certificate's validity. It also addresses the privacy issue since the OCSP responder does not have access to knowledge about a web site's visitors. Nonetheless, it does not resolve the problem of scalability due to the single point of failure.