# Security Architecture / Exam questions

**What is the major issue solved by Public Key Infrastructure ?**

Have Authenticity of the public key ← link an id with a public key
→ link : $pub_K \leftrightarrow ID$ — we can prevent Man-in-the-middle
also don't have KDC with all key (non repudiation)

**What is a X.509 certificate and what are the information contained in it ?**

Standard format for certificate in a PKIX (Public Key Infrastructure
information in it :
- ID = serial number → a unique identifier assigned by the CA to distinguish each certificate
- $pub_K$ = subject
- validity period
- signature by the CA = n'a pas l'air de représenté l'issuer (mais on va dire = issuer.

**Why publish electronic certificates in a repository ?**

Must be a public data.

**Why publish revocation lists in a repository ?**

Must be able to check the validity or the revocation status for a key.

**What is the main usage of Certificate Revocation Lists and what are the contents of a CRL published by a PKIX-like Hierarchical PKI ?**

Check the revokation status of a certificat ← sign by the CA
List of serial number

Cert ∈ ? { issuer, serialNumber, ... }

$\sigma_{CA}$ ← sign by the CA

- issuer
- date / Next update
- [ serial, Key compromise, Affiliation changed, Superseded, ... serial ] *

* unique combinaison serial / CA

**What is the effect of revocation on the lifetime of certificates ?**

Revocation is a pain, but with that we can have long validity periods.
However we must have short TTL.

Drawback of CRL : daily broadcast of (potentially large) amount of data
→ distribution point

**What is a Δ-CRL ?**

List of all certificates revoked since (the previous) base CRL # ← possible to have multiple Δ-CRL
Δ-CRL is not enough (Not complete as the CRL) must check the original CRL.

**What is an indirect CRL**

Not handle directly by the CA but by the CRL issuer (= something given by the CA).
The revocation authority signs the CRL itself, not the CA that issued the certificates. ← scalability / Separation of responsabilities.

**What are the differences between DV (Domain validation), OV (Organisation Validation) and EV (Extended Validation), certificate ?**

Can add information (like the owner) on the certificate.

**What are a Certification Authority (CA), a self signed certificate and a trust anchor.**

Certification Authority : a trusted entity that issues and manage certificates within a PKI. Its primary role is to ensure the authenticity of public keys and their associated identity. Main responsabilities : ① Certification issuance ② Digital signature ③ Revocation Management *

A self signed certificate : a certificate signed by the same entity that issued it (a CA could be self signed).

A Trust Anchor : a trusted entity at the top of the certification chain / Point de départ de la confiance / si la trust anchor est compromise, toute la chaine devient invalide. ↓

* : the CA publishes Certificate Revocation Lists (CRL) or maintain OCSP servers to indicate whether a certificate is still valid.
4. Maintaining the chain of trust

## What is a Certificate Signing Request (CSR)?

The data that you want the CA include in the certificate.

## What is RHtps?

http transport + SSL ← for securing communication

----> TLS = way to secure exchanges ← cipher using asymmetric cipher

web server presents a certificate to the server.

## What are the Online Certificate Status Protocol (OCSP) and a stapled OCSP?

(Server) OCSP: pour vérifier l'état de validité des certificats numérique ← permet de ne pas télécharger un fichier (CRL) ← permet une vérification rapide et ciblée.

stapledOCSP: Dans un échange, il peut être intéressant que ce soit B qui fasse la demande de statut de son propre certifat, et qu'il joigne (agrafe) ensuite le resultat avec ses message à A.

## Pourquoi Signer (SiG) / Comment Signer?

Authentication / data integrity / non repudiation / Trust in certificates

Sign with the private key } asymmetric encryption

Check with the public key

## Pourquoi faire/utiliser un MAC (Message Authentication Code) / Comment faire?

Authentication

data integrity + protection against attacks.

Use a shared secret key between the sender and the receiver ← symmetric encryption

protect against man-in-the-middle

De quoi ça protège? SiG ⟷ garantir que la clé est authentique et provient d'une entité fiable

vérifier l'identité des parties prenantes impliquées dans la communication

MAC → garantir l'intregrité des données (données non modifiées ou corrompues durant le transfert)

↘ l'authenticité des données: s'assurer qu'elles proviennent bien de l'expediteur attendu.

protect against identity missbinding.

## What is the cross certification?

Process in which 2 or more certificate Authorities (CA) mutually trust each other by signing each other's certificate. This allow users to trust certificates issued by the other CA, even if they are not directly linked within the same trust chain.

## mutual cross certification?

When CA signed the certificate of the others ← both of us, the cross certification can be unidirectional ← this is the difference with the mutual cross certification.

## What is a blockchain → 3 components

Block → each block contains a set of transactions, these blocks are linked together in a chain

Distributed Network → the blockchain operates on a network of nodes (computers), each holding a copy of the entire blockchain

Consensus Mecanism → ensures that all nodes on the network agree on the current state of the blockchain. The 2 most common meanisms are: blocks

- PoW = Proof of work: nodes must solve complex computational problems to validate
- PoS = Proof of stake: Nodes are chosen to validate blocks based on the amount of crypto currency they hold and are willing to stake

## IBE?

Type of encryption where the public key of a user is derived from their identity. Relies on 3 elements

- Private key generation
- Central Authority (PKG) Public Key generator → CA responsible for managing/generating the user's private keys
- Encryption - Decryption: encypted with the public key

adv: simplicity / Reduce key management costs cons: Trust in PKG Authority / Scalability & perfo → can become a bottleneck issues

## All Steps that Alice performs to recover and verify Bob public key:

lecture 9.5 → ex 3.2