MASTER OF SCIENCE
IN ENGINEERING

TSM_SecIndOpT

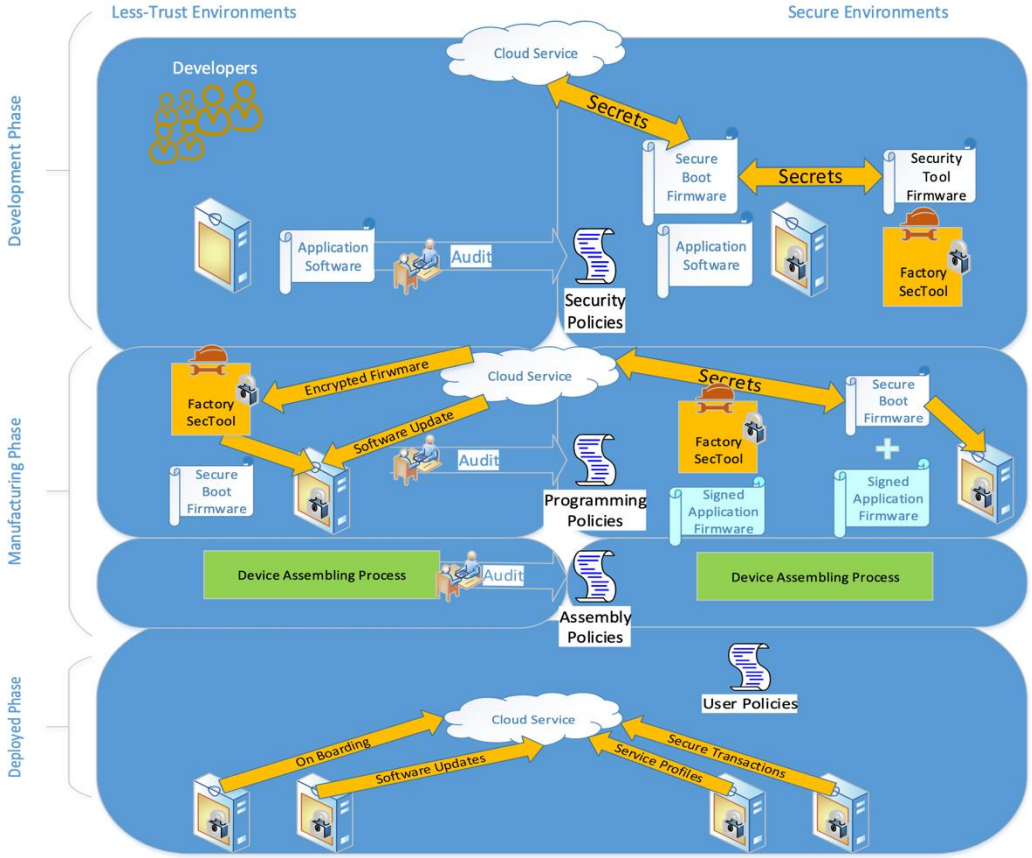
Public Key Infrastructure (PKI)

Version: 1.2

Luca Haab | 30.03.2026 | Cours MSE

How to distribute keys?

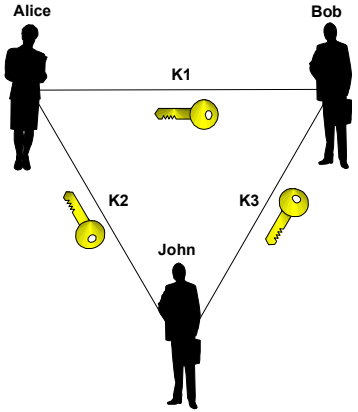
(I)IoT & Secure boot

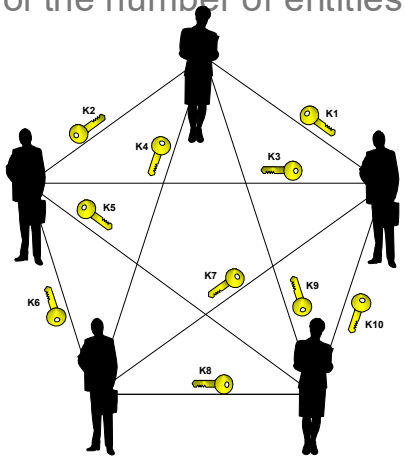


Source: https://www.nxp.com/docs/en/supporting-information/Designing-Secure-IoT-Devices-Starts-with-a-Secure-Boot.pdf

Symmetric Key Distribution

- A symmetric encryption method bases the confidentiality of the message on a secret key
- The number of keys needed evolves with the square of the number of entities





- One of the practical issues of cryptography concerns the distribution of keys!


Digital
Signatur

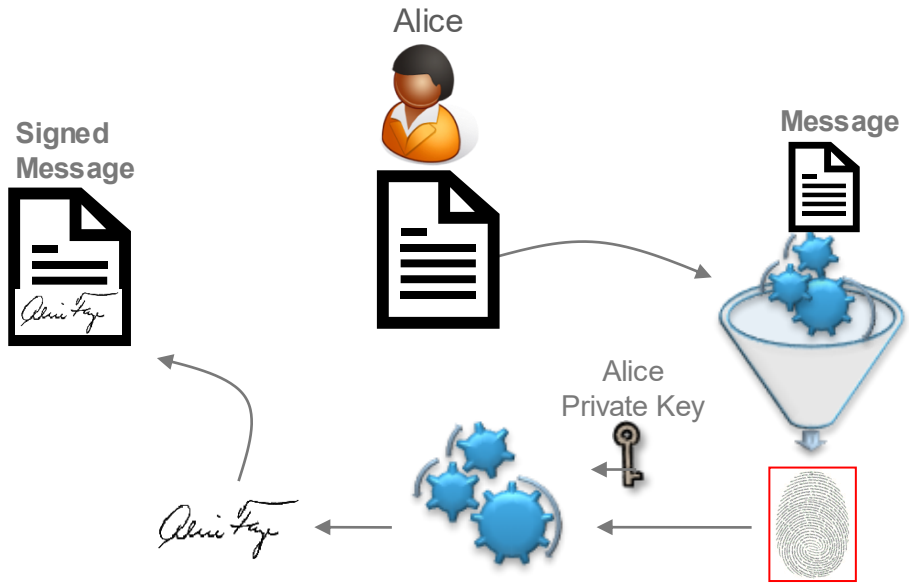
Electronic Signature (I)

- The electronic signature of a document (or a message) involves two successive operations
  - Extracting a fingerprint by a hash algorithm
  - Encryption of the previous fingerprint using a public key algorithm

- The objectives of a signature process are:
  - Guarantee the integrity of the transmitted data
  - Guarantee the authenticity of the document

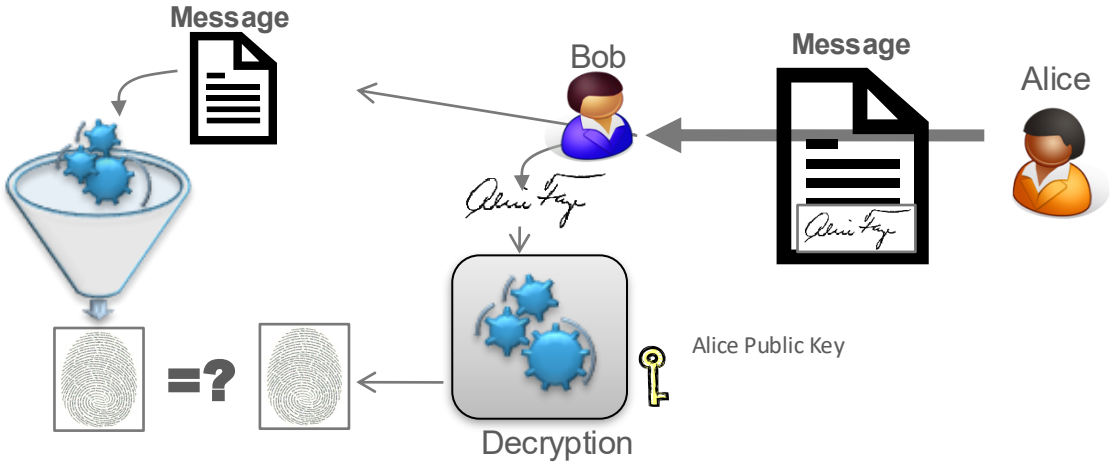
Electronic Signature (II)

- An electronic signature is associated with the signed message



Electronic Signature (III)

- To verify a signature:
  - Bob uses Alice's public key to decrypt the signature of the message that Alice created with her private key
  - Bob calculates the hash function of the message
  - Bob compares the two results and validates the origin of the message



X.509 Certificates

Certificat X.509 (v3 or later) – Brief overview

- A digital certificate is a document that allows to authenticate an entity and to encrypt messages. It is composed of various administrative information associated with a public key, all signed by a trusted third party who attests to the link between the identity and the certificate.
- It is a secure link between an entity and a public key.
- An entity can be a person (John Smith), an Internet server (www.mybank.secure), a software, etc.
- The international standard X.509, published by the ITU, specifies the format of a certificate as well as the methods to validate it.

Digital Certificates

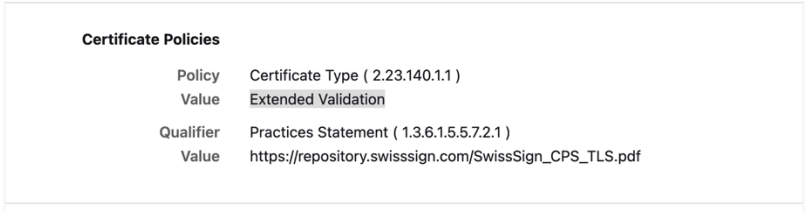
A certificate can be compared to a passport/ID card as it represents a secure link between an entity and a public key.

- Passport number
- Name of the authority
- Validity
- Identity of the person
- Signature
- Description
- Signature of authority

Source: https://www.fedpol.admin.ch/fedpol/en/home/pass-–identitaetskarte/pass/fotos-idk.html

Certificate Content (I)

- A certificate contains a lot of informative or imperative information.
- Example of the certificate delivered to www.zkb.ch :



Certificate Policies

Policy	Certificate Type (2.23.140.1.1)
Value	Extended Validation
Qualifier	Practices Statement (1.3.6.1.5.5.7.2.1)
Value	https://repository.swisssign.com/SwissSign_CPS_TLS.pdf

- Classe 3: a or multiple representatives of the Zürcher Kantonal Bank underwent a secure identification process

Certificate Content (II)

Subject Name

Inc. Country	CH
Inc. State/Province	Zürich
Business Category	Private Organization
Country	CH
State/Province	ZH
Locality	Zurich
	8001
	Bahnhofstrasse 9
Organization	Zürcher Kantonalbank
Serial Number	CHE-108.954.607
Common Name	zkb.ch

- The "Common Name" field specifies the name of the entity. It is unique
  - Can be "wildcarded" e.g. *.zkb.ch
  - The "alternative Name" option is preferred
- In a robust authentication, the client checks that the "Common Name" matches the DNS host name.
  - Depends on the implementation – Be aware

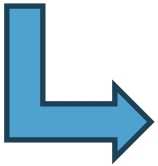
Subject Alt Names

DNS Name	zkb.ch
DNS Name	www.zkb.ch
DNS Name	unternehmensprofil.zkb.ch
DNS Name	bank.zuerich
DNS Name	www.bank.zuerich

Certificate Content (III)

Issuer Name

Country	CH
Organization	SwissSign AG
Common Name	SwissSign RSA TLS EV ICA 2022 - 1



Issuer Name

Country	CH
Organization	SwissSign AG
Common Name	SwissSign RSA TLS Root CA 2022 - 1

Issuer Name

Country	CH
Organization	SwissSign AG
Common Name	SwissSign Gold CA - G2

Certificate Content (IV)

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	A3:00:38:E0:C9:BB:B0:A1:F8:95:78:27:1B:3D:69:35:73:99:12:7A:1C:07:E7:...

Miscellaneous

Serial Number	49:94:97:F5:B6:67:DE:13:81:50:36:3A:C6:2C:20:61:4A:61:65:A0
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	PEM (cert) PEM (chain)

Fingerprints

SHA-256	D3:DF:78:AE:3B:B6:8F:39:23:4A:34:A5:14:9E:3D:7B:4A:17:15:A6:7A:B0:54...
SHA-1	35:7F:B3:B7:3E:54:CE:1B:04:AE:64:71:B6:14:F0:B6:C9:19:07:56

Key Usages

Purposes	Digital Signature, Key Encipherment

- The most important part of the certificate:
  - Entity's public Key
  - CA signature

Certificate Content (V)

Extended Key Usages

Purposes	Server Authentication, Client Authentication

Subject Key ID

Key ID	6F:76:92:B0:01:90:3B:18:92:E6:1A:79:5A:AF:7F:09:5A:78:FC:0A

Authority Key ID

Key ID	49:52:DF:30:86:92:59:5F:34:9C:25:48:24:AB:C0:EB:D1:06:F2:D6

CRL Endpoints

Distribution Point	http://crl.swisssign.ch/cdp-9fdd910e-b9ff-4b2f-be38-2e93708c1b36

Authority Info (AIA)

Location	http://aia.swisssign.ch/air-20350159-813d-4532-b988-8519eca57650
Method	CA Issuers
Location	http://ocsp.swisssign.ch/sign/ocs-aaccced5-66e8-4069-9b1b-fd29ab73efec
Method	Online Certificate Status Protocol (OCSP)

Certificate Policies

Policy	Certificate Type (2.23.140.1.1)
Value	Extended Validation
Qualifier	Practices Statement (1.3.6.1.5.5.7.2.1)
Value	https://repository.swisssign.com/SwissSign_CPS_TLS.pdf

- Different extensions allow specification of certificate usage as well as other entity names

Full requirements list:
https://cabforum.org/working-groups/server/baseline-requirements/requirements/

X.509v3 Extensions

- The following extensions are foreseen:
  - Authority key identifier: identifies the public key to be used to verify the signature on this certificate or CRL.
  - Subject key identifier: a subject may have multiple key pairs and, correspondingly, different certificates for different purposes (e.g.: digital signature, …)
  - Key usage: may indicate one or more of the following possibilities:

X.509 Format

Uses Abstract Syntax Notation 1 – ASN.1

```
Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }
   CertificateSerialNumber  ::=  INTEGER
   Validity ::= SEQUENCE {
        notBefore      CertificateValidityDate,
        notAfter       CertificateValidityDate }
   CertificateValidityDate ::= CHOICE {
        utcTime        UTCTime,
        generalTime    GeneralizedTime }
   UniqueIdentifier  ::=  BIT STRING
   SubjectPublicKeyInfo  ::=  SEQUENCE  {
        algorithm            AlgorithmIdentifier,
        subjectPublicKey     BIT STRING  }
   Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension
   Extension  ::=  SEQUENCE  {
        extnID      OBJECT IDENTIFIER,
        critical    BOOLEAN DEFAULT FALSE,
        extnValue   OCTET STRING  }
```

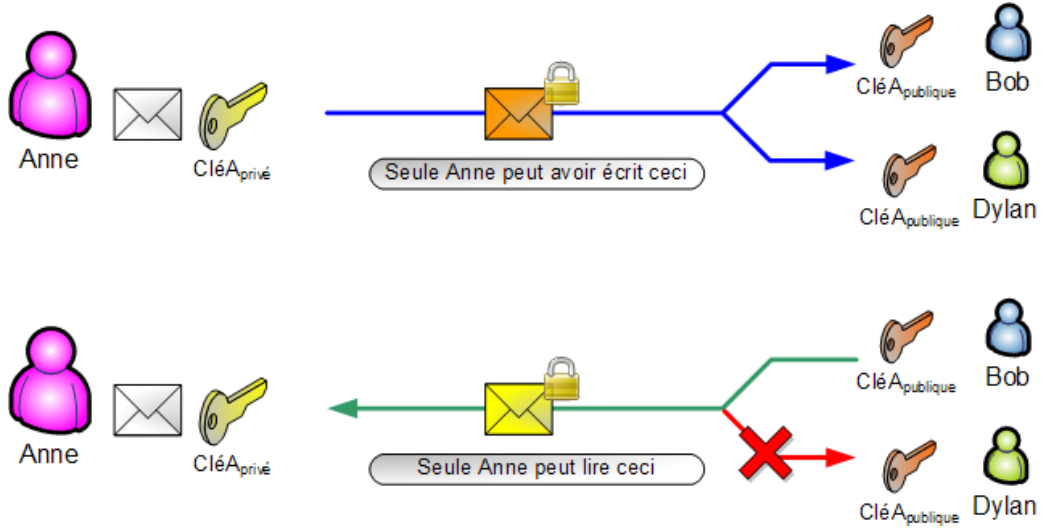
```
Certificate  ::=  SEQUENCE  {
       tbsCertificate       TBSCertificate,
       signatureAlgorithm   AlgorithmIdentifier,
       signature            BIT STRING  }
 TBSCertificate  ::=  SEQUENCE  {
       version         [0]  EXPLICIT Version DEFAULT v1,
       serialNumber         CertificateSerialNumber,
       signature            AlgorithmIdentifier,
       issuer               Name,
       validity             Validity,
       subject              Name,
       subjectPublicKeyInfo SubjectPublicKeyInfo,
       issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                            -- If present, version must be v2 or v3
       subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                            -- If present, version must be v2 or v3
       extensions      [3]  EXPLICIT Extensions OPTIONAL
                            -- If present, version must be v3
       }
```

Encoding of an X.509 certificate (ASN.1)

- With ASN.1 the content is described
- Numerous binary representations exist:
  - PEM (Privacy-enhanced Electronic Mail), Base 64, Préfix ---BEGIN…)
    - -----BEGIN CERTIFICATE-----
    - MIIDrTCCAxagAwIBAgIBBDANBgkqhkiG9w0BAQQFADCBijELMAkGA1UEBhMCQ0gx
    - RWNvbGUgSW5nZW5pZXVycyBkZSBGcmlib3VyZzEdMBsGA1UECxMUU2VjdGlvbiBJ
    - ……..
    - DpQb4VJNsZkybWtDvQLUtHYb52ikwGm1WJPwAWC//Y1Q
    - -----END CERTIFICATE-----
  - DER (Distinguished Encoding Rules)
  - PKCS developed by RSA laboratories
    - PKCS#1: RSA encryption standard
    - PKCS#5: Password-Based Encryption Standard
    - PKCS#7: Cryptographic Message Syntax Standard
    - PKCS#8: Private-key Information Syntax Standard
    - PKCS#10: Certification Request Syntax Standard
    - PKCS#12: Personal Information Exchange Syntax Standard

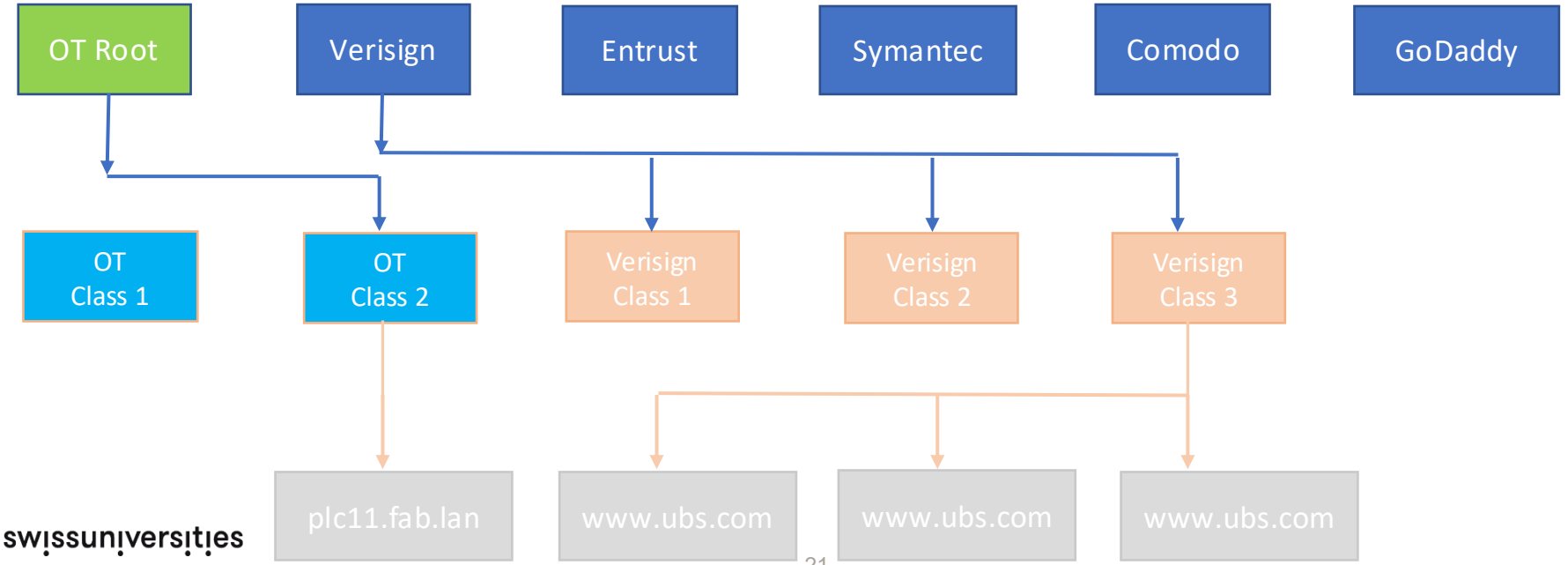
Use of Public Key

- Decrypt a signature (encrypted hash) and validate the identity of the signatory who used this private key
- Encrypt a content that only the owner of the private key can decrypt



Certification Hierarchy (I)

- A X.509 Public Key Infrastructure (PKI) is a hierarchical structure – other models exist



Certification Hierarchy (II)

- The certificates at the top of the hierarchy are called Root Certificates.
- They are self-signed and usually have an extended validity period (10-30 years)
- They are stored in web browsers or operating systems
- They are used to sign intermediate certificates (one or more levels), which in turn are used to sign individual certificates.

swissuniversities

Public Key Infrastructure

PKI In A Glimpse

- A PKI (Public Key Infrastructure) is a set of hardware, software, policies and procedures to create, manage, store, distribute and revoke digital certificates.
- The main mechanisms of a PKI are:
  - Registration and publication of certificates
  - Storage and distribution of certificates
  - Certificate revocation and status checking

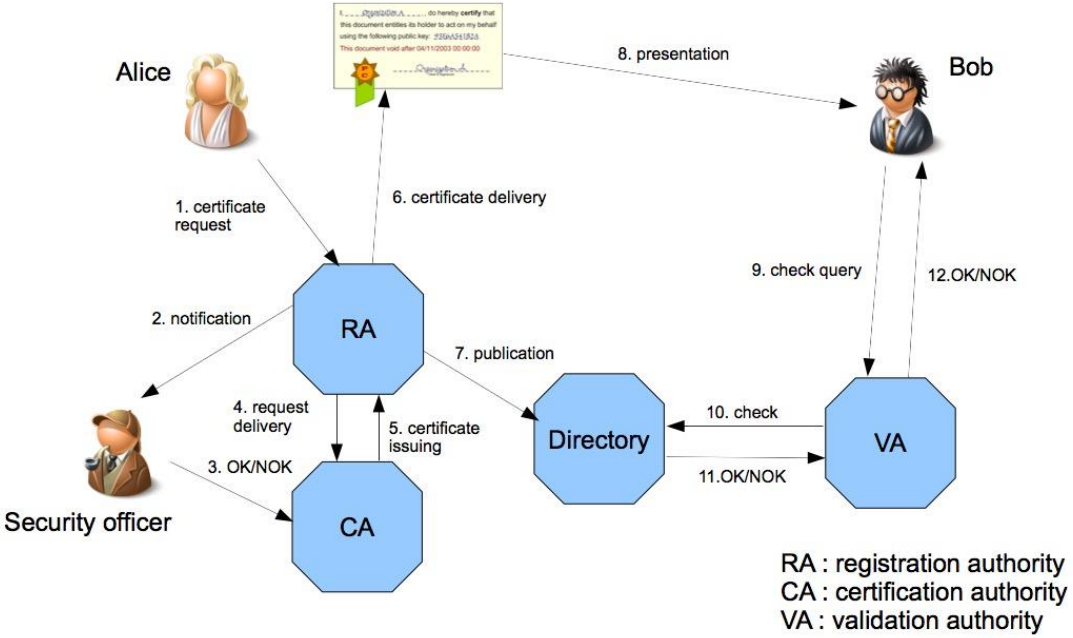
- Key generation is done by the requesting entity itself because the authority does not need to know the private part of the certified keys

PKI Objectives

- Symmetric and asymmetric private keys must be confidential and authentic
- Public asymmetric keys must be authentic

- An X.509 PKI aims to solve the challenge of public key through certificates
  - Is based on the notion of digital certificates
  - Uses a Trusted Third Party (TTP) to establish the trust relationship between entities

PKI Process


Alice
Bob
8. presentation
1. certificate request
6. certificate delivery
9. check query
12.OK/NOK
2. notification
RA
7. publication
4. request delivery
5. certificate issuing
Directory
10. check
VA
3. OK/NOK
11.OK/NOK
Security officer
CA
RA : registration authority
CA : certification authority
VA : validation authority

PKI: Certificate Authority (CA)

- The role of a CA is to publish and renew certificates as well as to create revocation lists (Certificate Revocation List : CRL)
- The CA, in a PKI concept, has a capital importance because the entire trust is based on it.
- Anyone can become a CA. In practice, public CAs are defined by root certificates contained in most browsers.
- Each certificate has a limited validity period. It is however possible to revoke a certificate in case, for example, a private key has been compromised.
- The revocation list contains the list of serial numbers of the revoked certificates, it is signed by the CA
- The generation of the keys is done by the requesting entity because the certification authority does not need to know the private key.

PKI: Registration Authority (RA)

- The role of an RA is to manage certificate requests. This includes:
  - Storing them
  - Verification of credentials
  - Communicating with requesting entities to verify compliance with certificate policies
  - Publishing revocation lists

- In practice, a bank, an administration, a telecommunication operator can provide partial RA services.

PKI: Validation Authority (VA)

- The role of a CA is to provide a certificate verification service, e.g. to verify that a submitted certificate is valid and has not been revoked.
- An example of a verification protocol is the Online Certificate Status Protocol (OCSP), described in the IETF standard RFC 2560.
- Nowadays: a browser does this 😉

Certification Practice

- A CA must describe the process used to issue a certificate in a document called a Certification Practice Statement

- The more the CA invests in verifying the identity of the entity, the more secure (and expensive) the certificate will be

- Extended Validation (EV) is currently the most stringent model
  - Shown in “green” in the browsers (well, not any longer but you can still see the difference if the certificate is analyzed)

- Rules for obtaining an EV certificate
  - Verification of the legal entity as well as the physical and operational presence of the entity.
  - Verification that the entity has the rights on the required domain.
  - Notarized confirmation of the entity and its rights.

Check a public site (MSE site in this example):
https://www.ssllabs.com/ssltest/analyze.html?d=www.msengineering.ch&latest

Certification Practice

- **March 15, 2026**: maximum TLS certificate lifespan shrinks to 200 days. This accommodates a six-month renewal cadence. The Domain Control Validation (DCV) reuse period reduces to 200 days.
- **March 15, 2027**: maximum TLS certificate lifespan shrinks to 100 days. This accommodates a three-month renewal cadence. The DCV reuse period reduces to 100 days.
- **March 15, 2029**: maximum TLS certificate lifespan shrinks to 47 days. This accommodates a one-month renewal cadence. The DCV reuse period reduces to 10 days.

- **Enhanced security:** Shorter certificate renewals protect private keys from being compromised by limiting the time they are exposed to potential threats, ultimately reducing the risk of man-in-the-middle attacks and data breaches.
- **Encouraging automation**: Reducing certificate lifespans encourages automation and the adoption of practices that drive the ecosystem away from baroque, time-consuming, and error-prone issuance processes. The result enables faster adoption of emerging security capabilities, changes in cryptographic algorithms, and general best practices.
- **Preparing for quantum challenges:** In an era of promoting quantum preparedness, shorter certificate lifespans foster crypto agility by accelerating the adoption of stronger algorithms and ensure compliance with evolving security standards.

Sources:
https://cabforum.org/working-groups/server/baseline-requirements/requirements/#7123-technically-constrained-non-tls-subordinate-ca-certificate-profile
https://knowledge.digicert.com/alerts/domain-validation-reuse-changes-in-2026#:~:text=This%20change%20does%20not%20affect,Learn%20more%20about%20Ballot%20SC081v3.

OT Certification Practice (I)

- Embed the CRL Distribution Point in your certificates
- Host CRL (Certificate Revocation Lists) internally
  - point to an internal HTTP server (e.g.http://pki.yourplant.local/crl/intermediate.crl)
  - use HTTP for the CRL endpoint — otherwise you get a chicken-and-egg problem
- Automate CRL generation and publishing
  - generate a new CRL every 24-36 hours
  - set the CRL's "next update" field to something like 4–5 times the generation period (gives you a buffer if publishing fails)
- Decide on hard-fail vs soft-fail
  - a device cannot verify revocation status, it rejects the connection
  - if revocation status cannot be checked, the connection proceeds anyway. Less secure, but prioritizes availability
- Plan for the air-gapped or heavily segmented case
- Monitor and alert on revocation infrastructure health
  - alert if the CRL has not been regenerated
  - alert if the CRL HTTP server is unreachable from key network zones
  - log when devices report CRL check failures

OT Certification Practice (II)

- Many OT protocols (Modbus, OPC UA, DNP3 with TLS) have varying levels of support for certificate validation and revocation checking.
  Test thoroughly with your actual devices — some may ignore CRLs entirely.
- IEC 62351 (security for power systems), in section 62351-9, has specific guidance on certificate management in OT that may be relevant depending on your industry.
- Change management in OT is more rigid. Plan CRL updates and certificate renewals around maintenance windows and make sure your validity periods give you enough runway to handle delayed updates.