Cybersecurity: Certificates, Certification Authorities and Public-Key Infrastructures

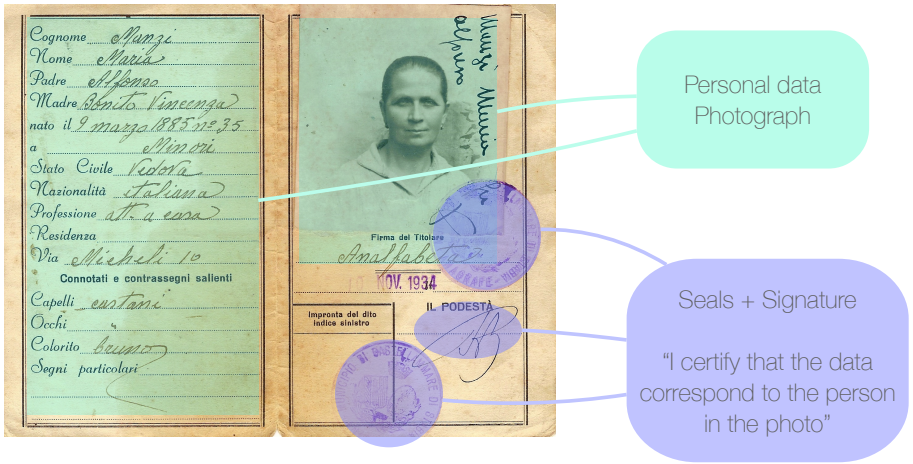
Ozalp Babaoglu

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

(Digital) Certificates

- We need to be sure that the public key used to encrypt a message indeed belongs to the destination of the message
- Distributing public keys in a naive manner is subject to possible *man-in-the-middle attacks*
- Distributing public keys using digital *certificates* prevents an intruder from impersonating someone else by substituting their public key

Certificates in the physical world



Certificates in the digital world

- Today, we have many examples of *digital certificates* such as the Covid-19 Green Pass
- The autograph signature of the trusted issuer is replaced with their *digital signature*
- Use of the certificate requires that it be *validated* by making sure that it has not expired and that the signature belongs to an entity that is considered and *authority*

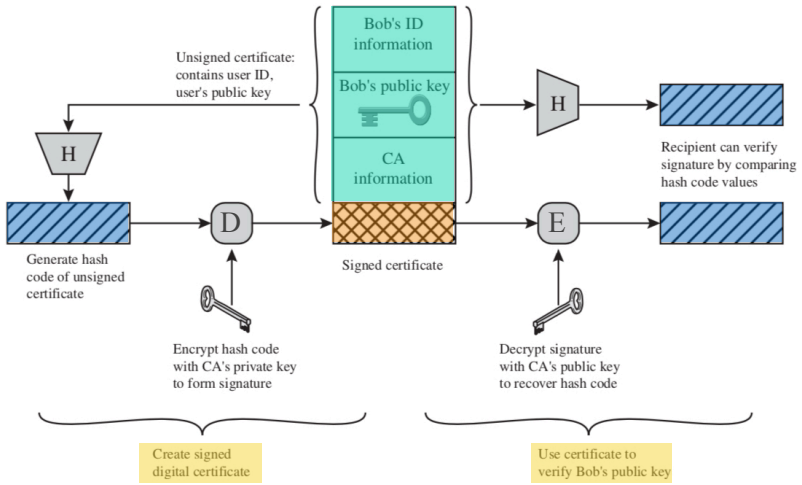
Certificates in the digital world

- Fake certificates like this one were produced by hacking a French health authority and stealing their private key

Digital Certificates

- In asymmetric cryptography, a digital *certificate* is the form in which public keys are communicated
- It is a ***binding*** between a ***public key*** and ***identity information*** about a subject
- It is ***signed*** by a trusted issuer (CA: ***certification authority***)
- Functions much like a physical certificate
- Avoids *man-in-the-middle* attacks

Digital Certificate use



Digital Certificate properties

- Any participant can ***read*** a certificate to determine the name and public key of the certificate's owner
- Any participant can ***validate*** a certificate to determine that it originated from a certification authority, that it is not a counterfeit and that it has not expired
- Only the certification authority can ***create*** or ***update*** certificates

X.509 Certificates

X.509 is a standard that specifies digital certificates with the following fields:

Subject: Distinguished Name, Public Key

Issuer: Distinguished Name, Signature

Validity: Not Before Date, Not After Date

Administrative Info: Version, Serial Number

Extended Info: …

X.509 Certificates

- *Distinguished Name Fields* as defined by X.509 Standard

Common Name	CN=Kenneth Lay
Organization or Company	O=Enron
Organizational Unit	OU=Management
City/Locality	L=Houston
State/Province	ST=Texas
Country (ISO Code)	C=US

Certification Authority

- *Certification Authority* (CA) is responsible for ***certification***, ***validation*** and ***revocation*** of certificates
- Many different types of CAs exist: commercial, government, free, etc.
- Examples of CAs: VeriSign, Symantec, GoDaddy, Geotrust, Visa, Actalis, Comodo, Thawte, Taiwan GRCA

Public Key Infrastructure

- The collection of hardware, software, people, policies, and protocols needed to create, manage, store, distribute, and revoke digital certificates constitutes a ***Public Key Infrastructure*** (PKI)

PKI – Certification

- The *subject* generates a (private, public) key pair
- Asks a CA that the (subject_ID, public_key) be certified and transformed into a *certificate*
- The CA ***authenticates*** the subject by verifying that the ID indeed belongs to her
- CA generates the signature for (subject_ID, public_key) using CA's private key
- CA attaches the signature to (subject_ID, public_key) to create the certificate
- CA returns the certificate to the subject (and anyone else who needs it)

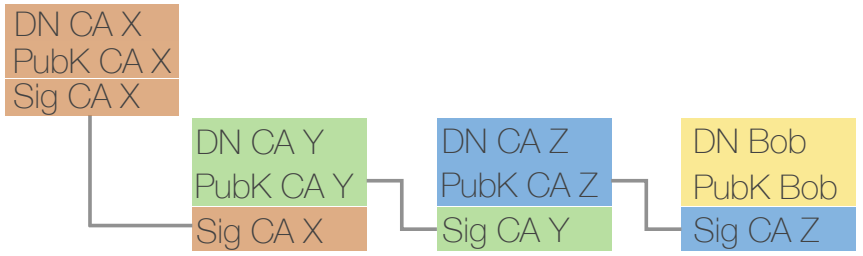
PKI – Authentication

- Out-of-band authentication:
  - performed using traditional methods, such as mail, fax, telephone or face-to-face meeting
- In-band authentication:
  - performed using the PKI itself
  - possible only for certain types of certificates where the ***identity information*** (e.g. email addresses) can indeed be verified

PKI – Certification Authorities

- The certification process is based on trust
  - Users trust the issuing authority to issue only certificates that correctly associate subjects to their public keys
- Only one CA for the entire world?
  - No — would be impractical
- Instead:
  - Most PKI allow one CA to certify another CA
  - One CA is telling its users that they can trust what a second CA says in its certificates

PKI – Certificate Chains

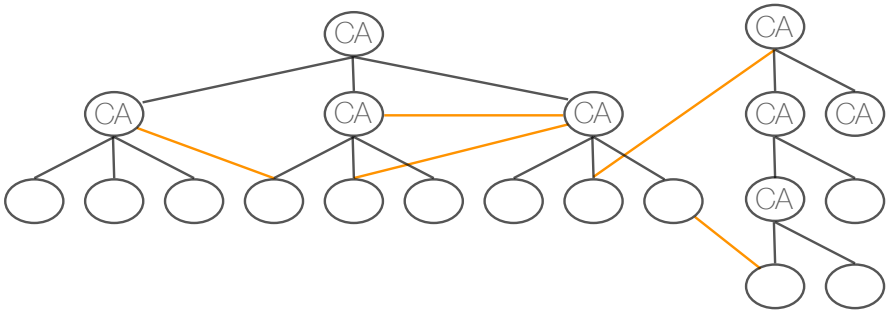


PKI – Certificate Chains

- Certificate chains can be of arbitrary length
- Each certificate in the chain is ***validated*** by the preceding one until the root certificate (which is self validated) is reached
- Different certificates:
  - "Leaf" certificates (end-user)
  - "Intermediate" certificates
  - "Root" certificates

PKI – CA Hierarchies

- CAs can be organized
  - as a rooted tree (X.509)
  - as a general graph (PGP)



Hierarchical Trust (X.509)

- Based on chains of trust forming a rooted tree among entities that are reputed to be CAs
- The (blind) trust we place on root-level CAs must be acquired through reputation, experience, operational competence and other non-technical aspects
- Anyone claiming to be a CA must be a trusted entity and we must believe that it is secure and correct

Web of Trust (PGP)

- In PGP, any user can act as a CA and sign the public key of another user
- A public key is considered valid only if a sufficient number of trusted users have signed it
- As the system evolves, complex trust relations emerge to create a dynamic "web"
- Trust need not be *symmetric* or *transitive*
- (more on PGP later)

PKI – Validation

- ***Validation*** — need to control that the certificate
  - Is ***current*** — within "not before" and "not after" dates,
  - Has been ***signed*** by a root CA or there is a chain that leads to a root CA,
  - Has not been ***tampered*** with
  - Has not been ***revoked***
- Checking *currency, signature* and *tampering* can be done locally and off-line by the certificate user (like in the *VerificaC19* App for the Green Pass)
- Checking if the certificate has been *revoked* is more complicated

PKI – Revocation

- ***Revocation*** — the process of breaking the binding between a public key and a subject for various reasons
  - subject's private key becomes compromised
  - subject identifier information changes (name, URL, email address)
- Since certificates are handed out to clients, it is impossible to physically recall them back
- Revocation can only insert the certificate to be revoked in a *list* of revoked certificates

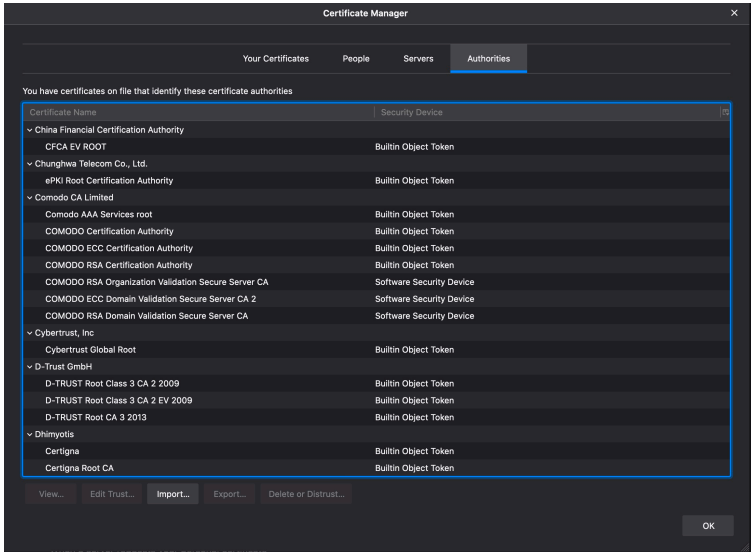
PKI – Revocation

- Controlling if a certificate has been revoked during validation can be performed either
  - On-line
    - consult the centralized database of all revoked certificates
    - *Online Certificate Status Protocol* (OCSP) of X.509 describes how to check validity and revoke certificates
  - Off-line
    - consult a "local copy" of the revoked certificates database
    - since the local copy can be "out-of-date" (missing some recent revocations), there is the risk of validating a certificate (while it is current) that has been revoked

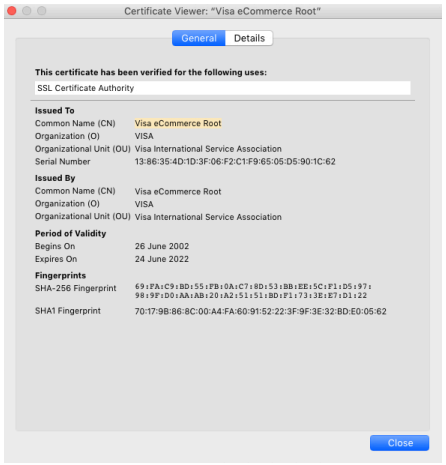
PKI – Revocation

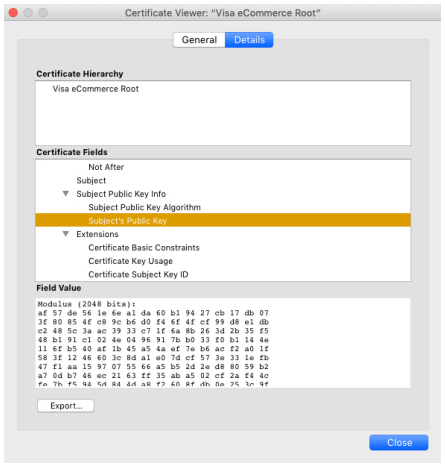
- Certificate Revocation List (CRL)
  - a list of revoked certificate ids constructed, signed and periodically distributed by the CA
  - user must check the latest CRL during validation to make sure that a certificate has not been revoked
  - X.509 includes a CRL profile, describing the format of CRLs
- CRL Problems
  - CRL time-granularity problem — how often to issue CRLs?
  - CRL size — incremental versus bulk

Certificates in Practice: Firefox



Certificates in Practice: Firefox





Certificates in Practice: Firefox