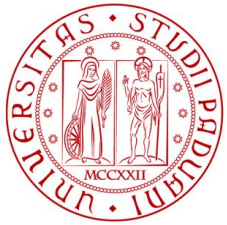
Mobile Security

Dr. Eleonora Losiouk
Department of Mathematics
University of Padua
elosiouk@math.unipd.it
https://www.math.unipd.it/~elosiouk/



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

DIPARTIMENTO
MATEMATICA

App Signing

- Each app is signed with a certificate
- A certificate is
  - the public key of a public/private key pair
  - some other metadata identifying the owner of the key
- The owner of the certificate holds the corresponding private key

Primer on cryptographic signatures

- Developer generates a public/private key pair:
  - private key: PRIV
  - public key: PUB
- The developer keeps PRIV secret
- PUB, as the name suggests, is public

What is Digital Certificate ?

- A certificate is an electronic document that **is used to identify an individual**, a server, a company, or some other entity
- Certificates use **public key cryptography** to address the problem of **impersonation**
- Files with **limited validity** used to guarantee an identity
- **Certificate authorities (CAs)** are entities that validate identities and issue certificates

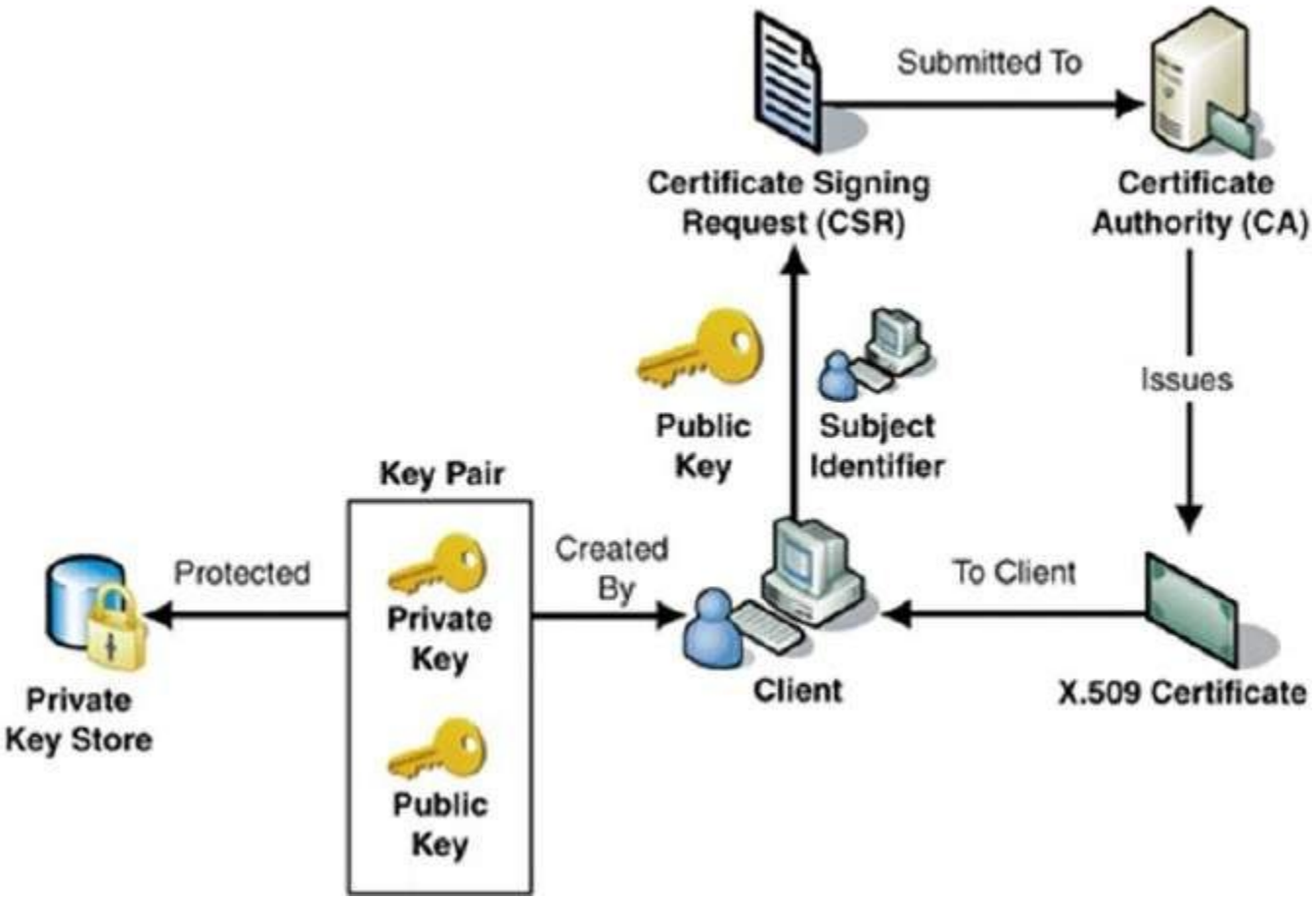
Digital Certificate

- There are many standards for creating certificates, the most used is X509.

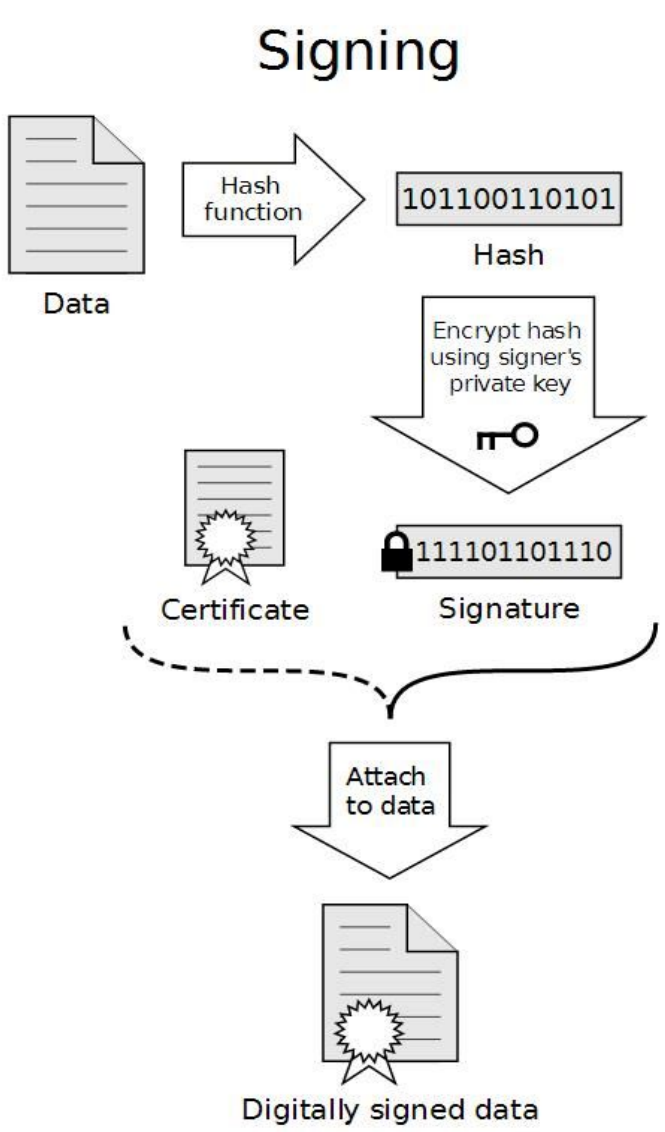
The CA digital signature allows the certificate to function as a ***letter of introduction for users who know and trust the CA****,* ***but do not know the entity that is identified by the certificate***

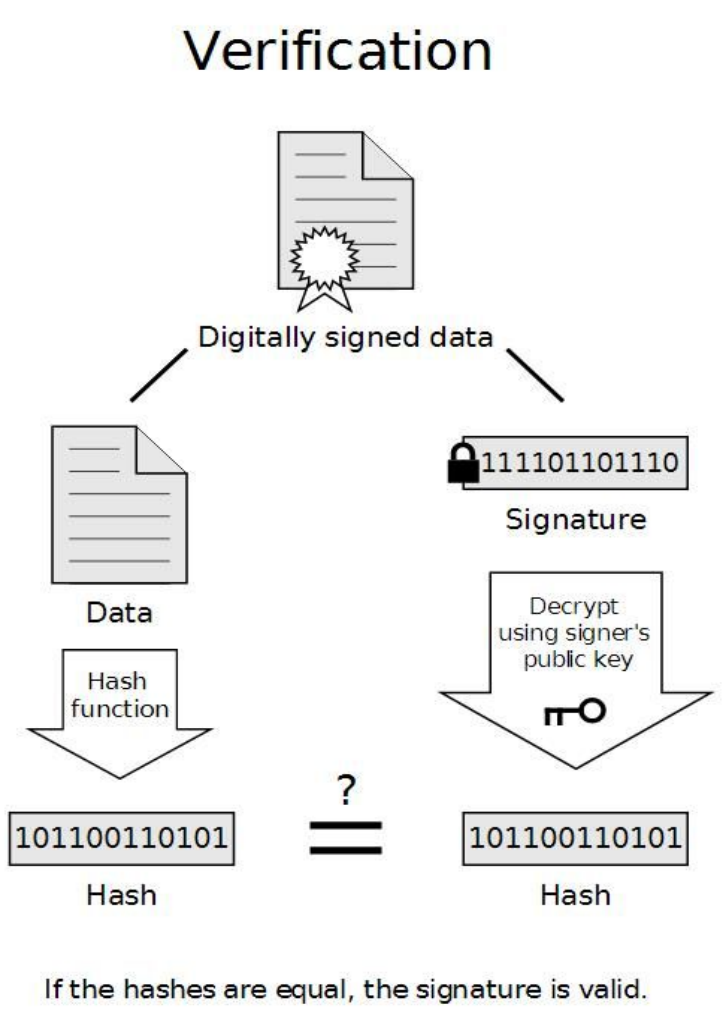
- A certificate is composed by:
  - Version, Serial number, Validity
  - Owner public key
  - Owner information
  - Public key expiration time
  - The CA that released the certificate
  - The CA digital signature

Certificate-Signing Request



Sign and Verify

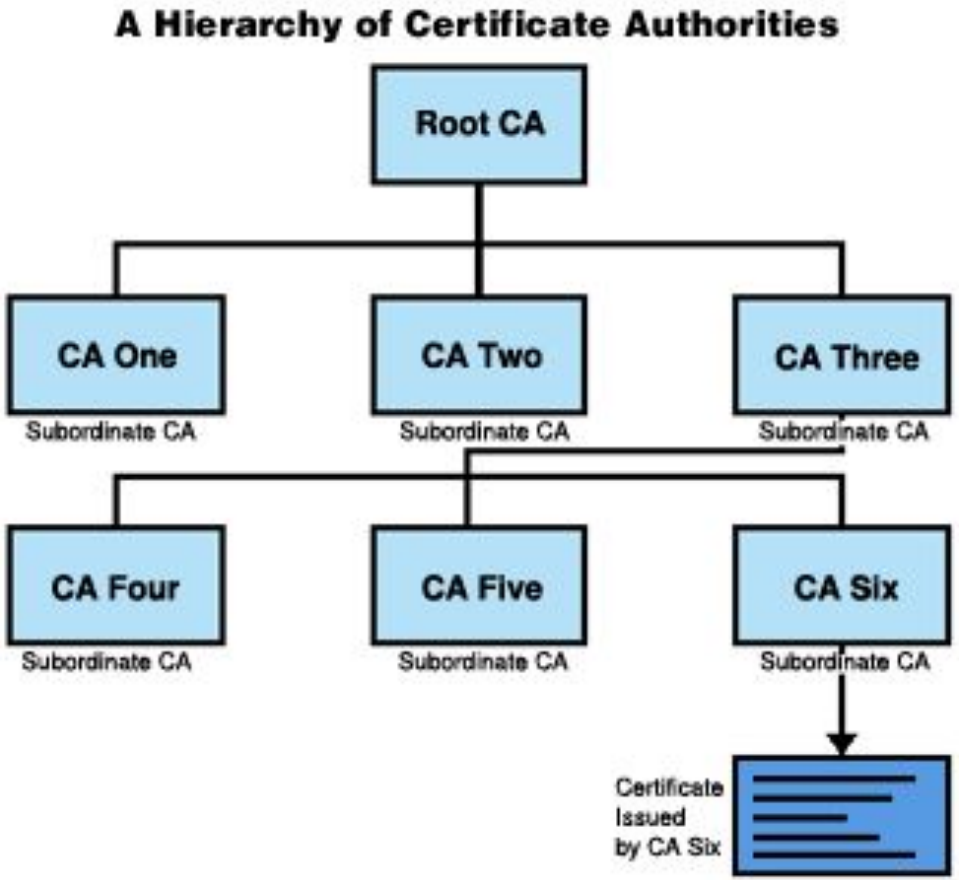




Certification Authority

- CA signature guarantees the device - key bond
- The list of valid and expired certificates is public
- Each CA establishes its own procedure necessary for the client to get the certification
- The structure of the CAs is hierarchical
- There are two types of CA:
  - Public (trusted)
  - Private (untrusted)

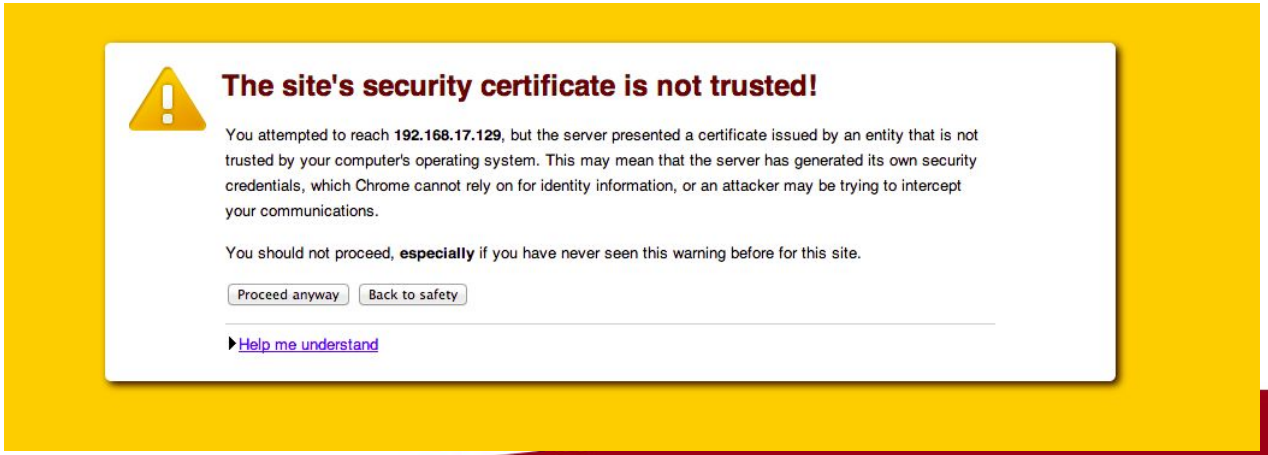
Hierarchical Structure of CA



- Root CA - is self signed
- Subordinate CA's are signed by root CA
- CAs under the subordinate CAs have their CA certificates signed by the higher-level subordinate CAs

Self-Signed certificates

- Certificate issued by the same entity it certifies
- Signed with CA's the private key
- Certificates at the roots of the CA tree are self signed
- Trust problem: the CA can emit as many certificates as it wants



Android App Certificate

- Certificate doesn't need to be signed by cert. authority
  - Apps' certificates can be self-signed
  - Major difference with SSL certificates
    - SSL certificates can be self-signed, but they are not trusted by default
- Purpose: distinguish app authors, NOT identify them
- You can distinguish "system" vs "normal" apps
  - System apps are those that are signed with a "system" certificate
  - That's how the system deals with "signature" permissions

Android App Certificate

- You can determine that a given Facebook app has the same developer as a given Messenger app
- But by only checking an app's certificate...
- ...you CANNOT determine whether the Facebook app you have is the legitimate/official one!

App's certificate vs. SSL certificate

- Your browser knows that the "facebook.com" website you are talking to IS the real, legitimate one

- That certificate is signed by another certificate, which is signed by another certificate, which is signed by ....

- ... which is signed by a certificate that your browser trusts

Signing apps

$ keytool -genkey -v -keystore debug.keystore -alias androiddebugkey -keyalg DSA -sigalg SHA1withDSA -keysize 1024 -validity 10000

$ jarsigner -keystore <path to debug.keystore> -verbose -storepass android -keypass android -sigalg SHA1withDSA -digestalg SHA1 app.apk androiddebugkey