Public Key Certificates

Levente Buttyán

CrySyS Lab, BME

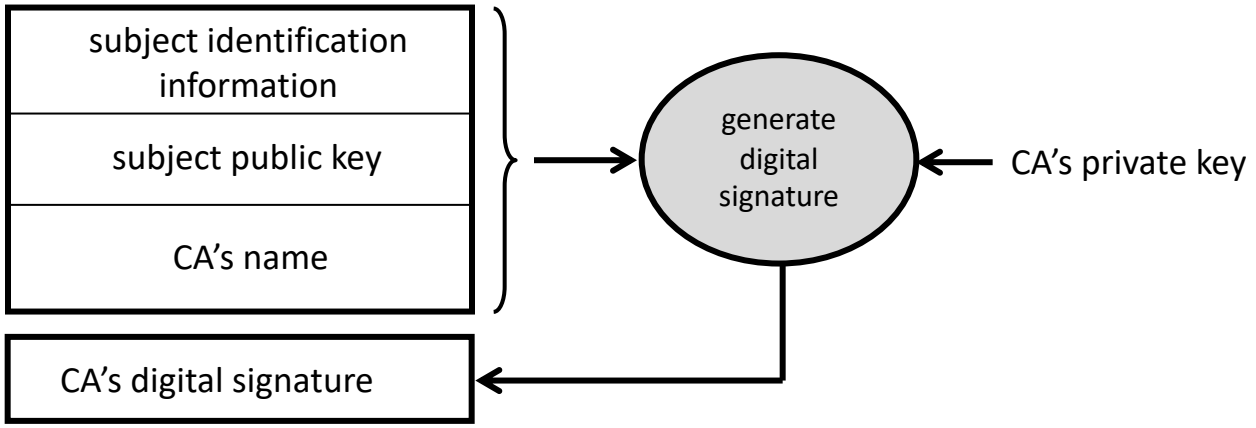
buttyan@crysys.hu

Distribution of public keys

- security requirements for public keys:
  - confidentiality is not needed (they are public, anyway)
  - authenticity (and integrity protection) is indispensable (why?)
- how to authenticate public keys?
  - physical meeting with key owner (out-of-band channel)
  - download public key and call the key owner to check its hash value via phone (out-of-band channel)
  - these solutions are not always practical and they don't scale
- a scalable approach for public key authentication is provided by public key certificates and related PKI mechanisms

Basic idea of certificates

- key owner name and public key are linked together by the digital signature of a **trusted entity** called **certification authority (CA)**



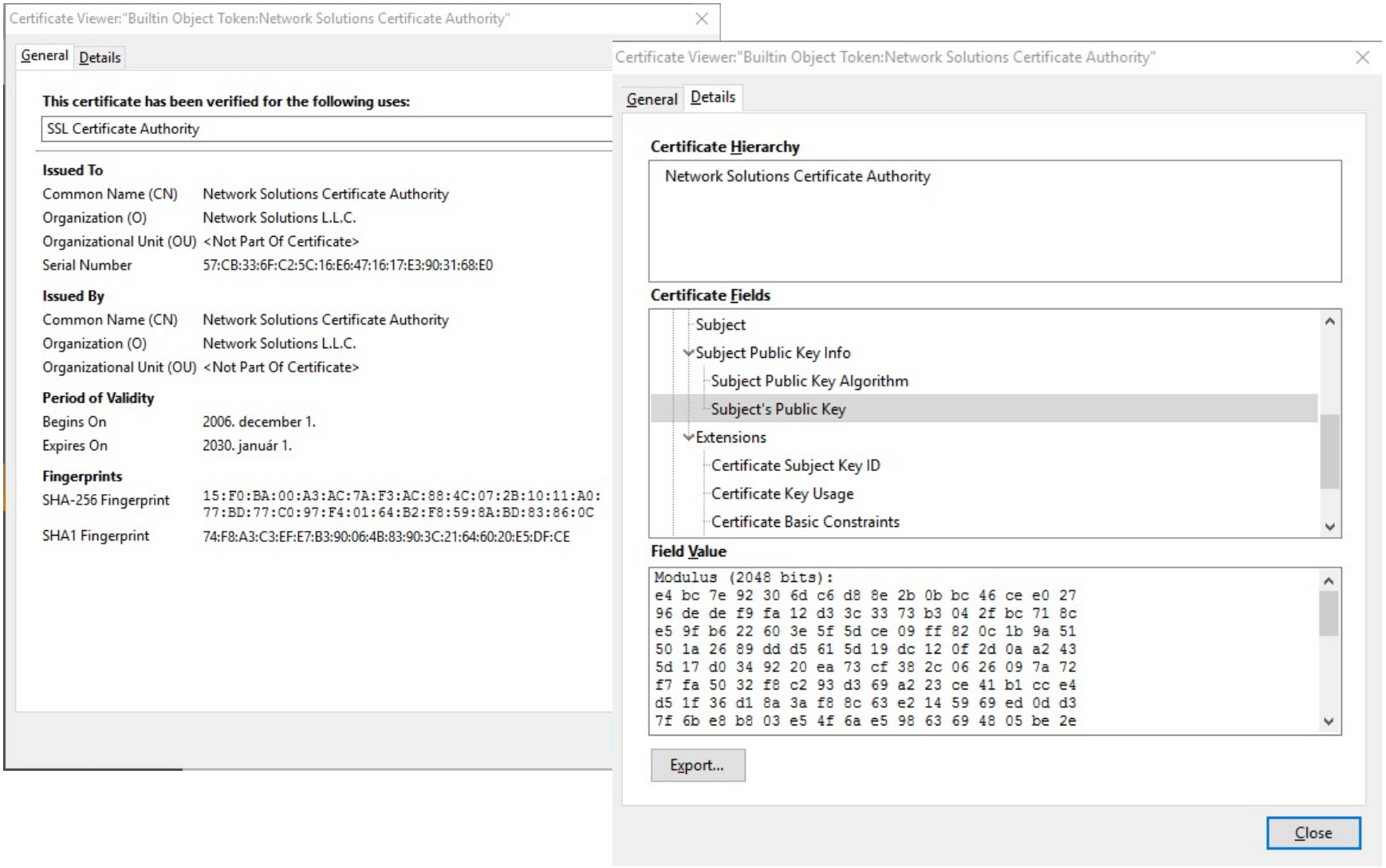
- in order to verify a certificate you need to have an authentic copy of the public key of the CA
- advantages: only the CA's public key need to be distributed via out-of-band channels (scales better)

Validity periods and revocation

- for security reasons, key-pairs shouldn't be valid forever
  - certificates have a scheduled validity period (valid_from, expires_on)
  - a certificate should not be accepted outside its validity period, but …
  - an expired certificate can still be used to verify signatures on documents that were generated within the certificate's validity period

- if a private key is (suspected to be) compromised, then the corresponding certificate needs to be revoked immediately
  - revocation is requested (from the CA)
  - certificate (and key inside) is invalidated (by the CA)
  - revocation information is published (by the CA)
  - revocation status must be be checked (by any relying party) before using a certificate

Certificates illustrated



Certification Authority (CA)

- collection of hardware, software, and staff (people)
- main functions:
  - issues certificates for users or other CAs
  - maintains certificate revocation information
  - publishes currently valid certificates and certificate revocation lists (CRL)
  - maintains archives
- must comply with strict security requirements related to the protection and usage of its private keys (basis of trust)
  - uses tamper resistant Hardware Security Modules that enforce security policies (access and usage control)
  - defines and publishes its certificate issuing policies
  - complies with laws and regulations
  - is subject to regular control (by national supervising authority)

Certificate life cycle

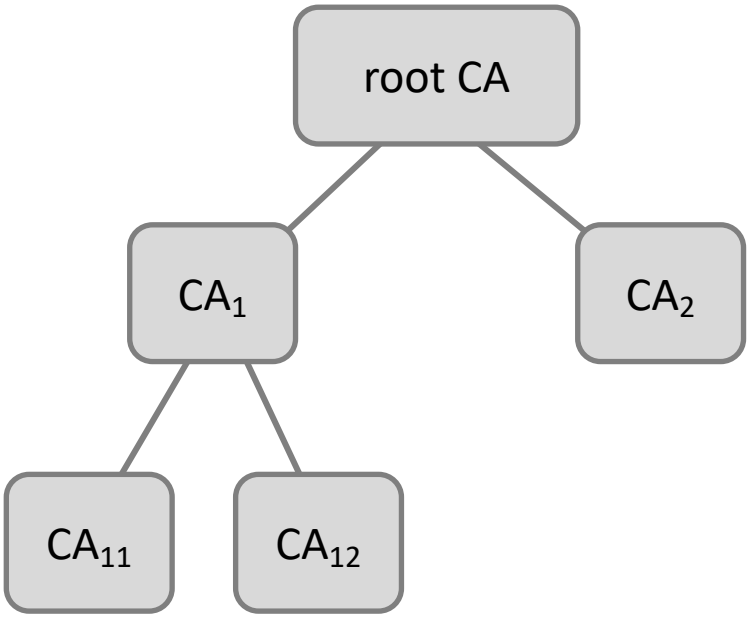
- subscriber registers with the CA and requests a certificate
- CA validates information provided by the subscriber
  - subscriber authentication
  - public-key verification (if supplied by the subscriber)
- key-pair generation
  - possibly in the hardware (e.g., smart card) of the subscriber where the private key will be stored later
  - extreme care is required if the key-pair is generated on the CA's system and the private key needs to be transferred to the subscriber's system
- issuance of the certificate
  - certificate is signed by the CA and transferred to the subscriber
  - copies are placed in repositories and archives
  - event is logged in secure audit trail
- certificate is used
- certificate may be revoked if needed
- certificate expires

Private-key protection

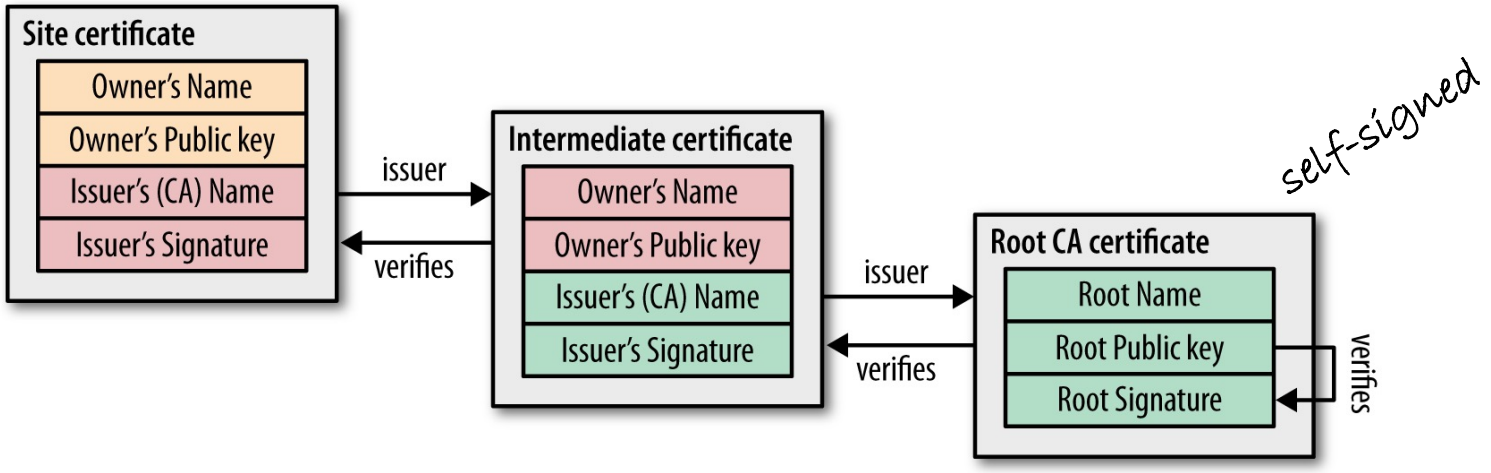
- protection of private keys from unauthorized access is of paramount importance (especially those of the CA)
- the CA's private key is stored in a tamper resistant hardware module
  - detection of tamper events and active response (deletion of keys)
- the private key of the subscriber is typically stored in
  - a tamper resistant hardware token (e.g., smart card)
  - an encrypted file on regular data storage media (e.g., USB key)
- access to the private key needs to be protected via one or more authentication mechanisms
  - typically, passwords and PINs ...
    - » can be used directly in case of hardware tokens
    - » encryption keys can be derived from them in case of encrypted files
  - ... and also biometric checks

Hierarchical PKI

- CAs are typically organized into a hierarchy where the key of a subordinate CA is certified by another, higher level CA
- this can be modelled as a (directed) tree
  - nodes are CAs
  - edges are certificates
- in practice, there exist multiple trees with different roots
- some roots may cross-certifiy each other to connect their trees



Certificate chains



- every end-user must have an authentic copy of the public key of the root CA (obtained out-of-band)
- every end-user certificate can be verified by verifying a chain of certificates that
  - starts with the root CA's self-signed certificate
  - ends with the end-user certificate
  - contains certificates of intermediate CAs on the path from the root to the user

Certificate revocation

- sometimes certificates need to be revoked before their expiration time
  - detected or suspected key compromise
  - change of data contained by the certificate (e.g., name, e-mail)
- revoked certificates are usually put on a Certificate Revocation List (CRL) published by the CA
- when verifying a certificate, one needs to
  - verify the CA's signature on the certificate
  - check the CRL in order to make sure that the certificate is still valid (not revoked)
- if you verify a certificate chain, then revocation status must be checked for every certificate in the chain!

Control questions

- Why do we need public key certificates?
- What essential elements does a public key certificate contain?
- Why should certificates have an expiration date?
- What could be reasons for revoking a certificate before it expires?
- What is a CA? What are its functions? Why do we trust CAs?
- What are the steps of the certificate life cycle?
- How are private keys stored and protected?
- How a hierarchical PKI operates?
- What is a certificate chain? How is it verified?
- Why and how are certificates revoked?