Asymmetric key management

Asymmetric key management : Goals

- Key pair generation
  - When and how should they be generated
- Exploitation of private keys
  - How can they be kept private
- Distribution of public keys
  - How can them be distributed correctly worldwide
- Lifetime of key pairs
  - Until when should they be used
  - How can one check the obsoleteness of a key pair

Generation of key pairs: Design principles

- Good random generators for producing secrets
  - Bernoulli ½ generator
    - Memoryless generator, unpredictability is crucial!!
    - P(b=1) = P(b=0) = 1/2
- Facilitate without compromising security
  - Efficient RSA public keys
    - Few bits, typically 2^k+1 values (3, 17, $65537 = 2^{16} + 1$)
    - Accelerates operations with public keys
    - No security issues
- Self-generation of private keys
  - To maximize privacy
  - This principle can be relaxed when not involving signatures

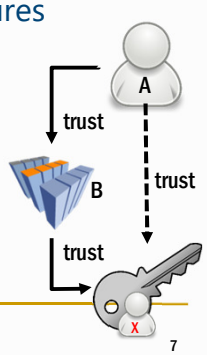
Exploitation of private keys

- Correctness
  - The private key represents a subject
    - Its compromise must be minimized
    - Physically secure backup copies can exist in some cases
  - The access path to the private key must be controlled
    - Access protection with password or PIN
    - Correctness of applications
- Confinement
  - Protection of the private key inside a (reduced) security domain (ex. cryptographic token)
    - The token generates key pairs
    - The token exports the public key but never the private key
    - The token internally encrypts/decrypts with the private key

Distribution of public keys

- Distribution to all **senders** of confidential data
  - Manual
  - Using a shared secret
  - Ad-hoc using digital certificates
- Distribution to all **receivers** of digital signatures
  - Ad-hoc using digital certificates
- Trustworthy dissemination of public keys
  - Transitive trust paths / graphs

    If entity A trusts entity B and B trust in K_X^+, then A trusts in K_X^+
  - Certification hierarchies / graphs



Public key (digital) certificates

- Documents issued by a Certification Authority (CA)
  - Bind a public key to an entity
    - Person, server or service
  - Are public documents
    - Do not contain private information, only public one
  - Are cryptographically secure
    - Digitally signed by the issuer, cannot be changed
- Can be used to distribute public keys in a trustworthy way
  - A certificate receiver can validate it
    - With the CA's public key
  - If the signer (CA) public key is trusted, and the signature is correct, then the receiver can trust the (certified) public key
    - As the CA trust the public key, if the receiver trusts on the CA public key, the receiver can trust on the public key

Public key (digital) certificates

- X.509v3 standard
  - Mandatory fields
    - Version
    - Subject
    - Public key
    - Dates (issuing, deadline)
    - Issuer
    - Signature
    - etc.
  - Extensions
    - Critical or non-critical
- PKCS #6
  - Extended-Certificate Syntax Standard
- Binary formats
  - ASN.1 (Abstract Syntax Notation)
    - DER, CER, BER, etc.
  - PKCS #7
    - Cryptographic Message Syntax Standard
  - PKCS #12
    - Personal Information Exchange Syntax Standard
- Other formats
  - PEM (Privacy Enhanced Mail)
  - base64 encodings of X.509

Key pair usage

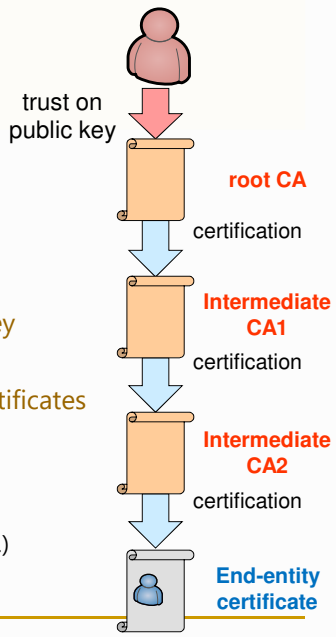
- A key pair is bound to a usage profile by its public key certificate
  - Public keys are seldom multi-purpose
- Typical usages
  - Authentication / key distribution
    - Digital signature, Key encipherment, Data encipherment, Key agreement
  - Document signing
    - Digital signature, Non-repudiation
  - Certificate issuing
    - Certificate signing, CRL signing
- Public key certificates have an extension for this
  - Key usage (critical)

Certification Authorities (CA)

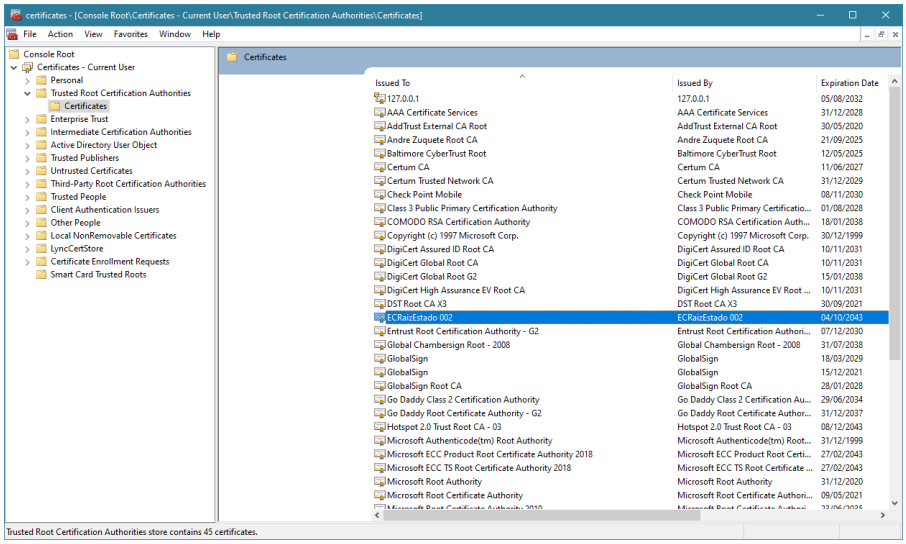
- Organizations that manage public key certificates
- Define policies and mechanisms for
  - Issuing certificates
  - Revoking certificates
  - Distributing certificates
  - Issuing and distributing the corresponding private keys
- Manage certificate revocation lists
  - Lists of revoked certificates

CA types

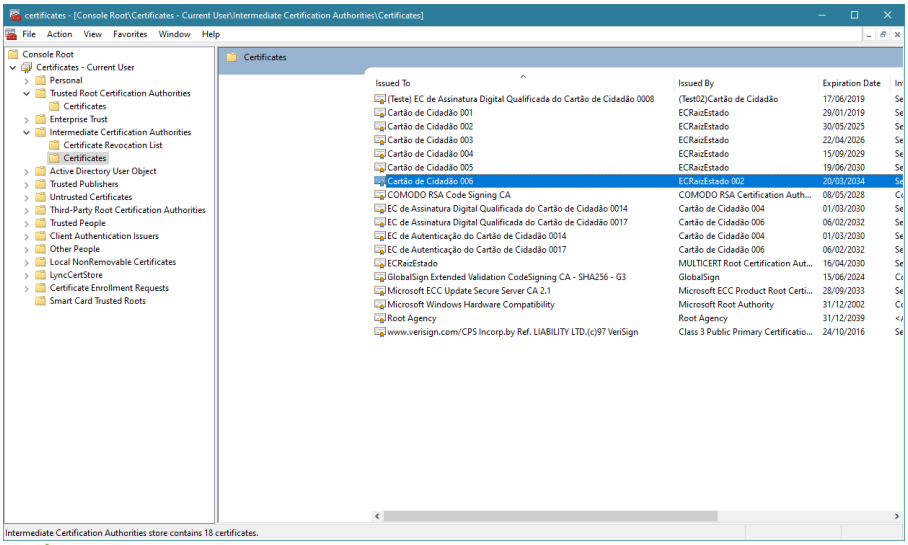
- Intermediate CAs
  - CAs certified by other Cas
- Root CAs
  - CAs for which one has a trusted public key
  - Trust anchor
  - Usually implemented by self-certified certificates
    - Issuer = Subject
    - Self-certification is not a reason for trusting!
  - Manual distribution
    - Tools' repositories (Firefox, Thunderbird, etc.)
    - Operating systems' repositories



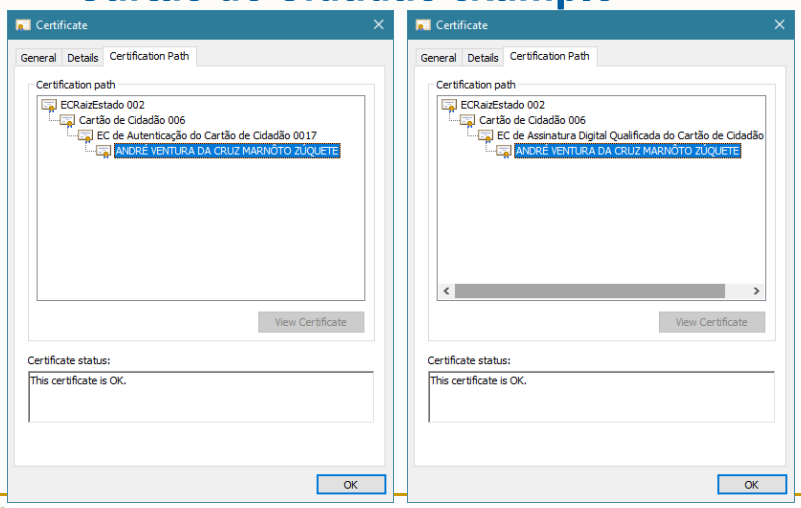
Certificates of Root CAs: Windows 10



Certs. of Intermediate CAs: Windows 10



Certification hierarchies (or chains, paths): Cartão de Cidadão example



Certification hierarchies: PEM (Privacy Enhanced Mail) model

- Distribution of certificates for PEM (secure e-mail)
  - Worldwide hierarchy (monopoly)
  - Single root (IPRA)
  - Several PCA (Policy Creation Authorities) bellow the root
  - Several CA below each PCA
    - Possibly belonging to organizations or companies
- Never implemented
  - Forest of hierarchies
    - Each with its independent root CA
    - Oligarchy
  - Each root CA negotiates the distribution of its public key along with some applications or operating systems
    - ex. Browsers, Windows

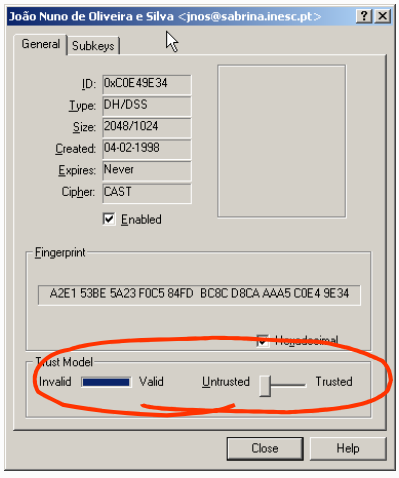
Certification hierarchies: PGP (Pretty Good Privacy) model

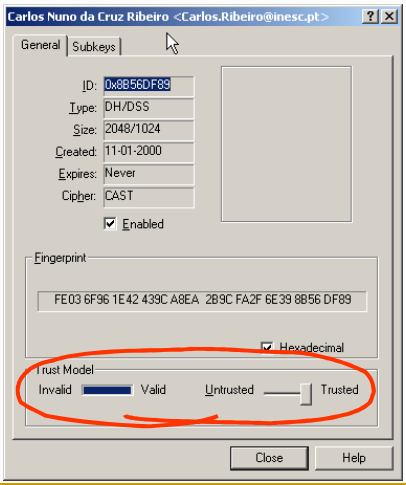
- Web of trust
  - No central trustworthy authorities
    - Each person is a potential certifier
    - Can certify a public key (issue a certificate) and publish it
  - People uses 2 kinds of trust
    - Trust in the keys they know
      - Validated using any means (FAX, telephone, etc.)
    - Trust in the behavior of certifiers
      - Assumption that they know what they are doing when issuing a certificate
- Transitive trust
  - If

    Alice trusts Bob is a correct certifier; and
    Bob certified the public key of Carl,
  - then

    Alice trusts the public key belongs to Carl

PGP public key certificates: Validity vs. trust





Refreshing of asymmetric key pairs

- Key pairs should have a limited lifetime
  - Because private keys can be lost or discovered
  - To implement a regular update policy
- Problem
  - Certificates can be freely copied and distributed
  - The universe of certificate holders is unknown!
    - Thus, cannot be told to eliminate specific certificates
- Solutions
  - Certificates with a validity period
  - Certificate revocation lists
    - To revoke certificates before expiring their validity

Certificate revocation lists (CRL)

- Base or delta
  - Complete / differences
- Signed list of identifyers of prematurely invalidated certificates
  - Must be regurlarly fetched by verifiers
    - e.g. once a day
  - OCSP protocol for single certificate check
    - RFC 2560
  - Can tell the revocation reason →

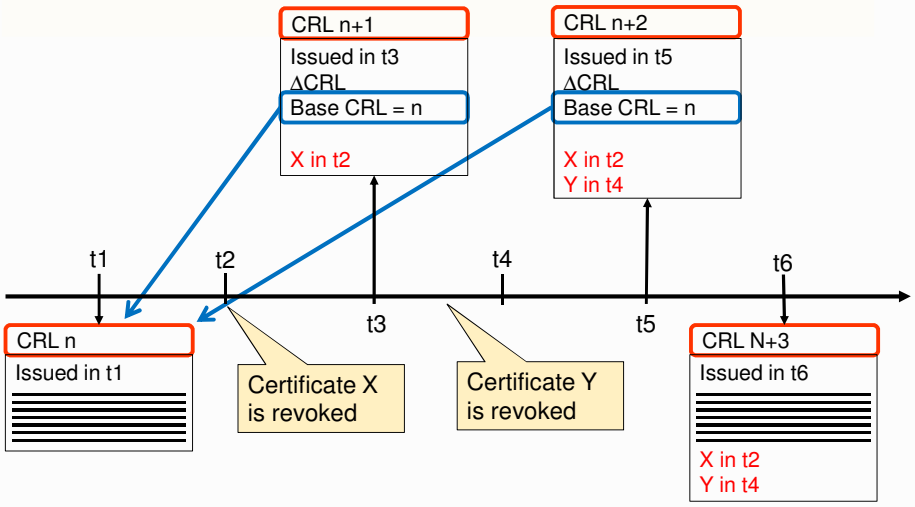
RFC 3280

unspecified (0)
keyCompromise (1)
CACompromise (2)
affiliationChanged (3)
superseded (4)
cessationOfOperation (5)
certificateHold (6)

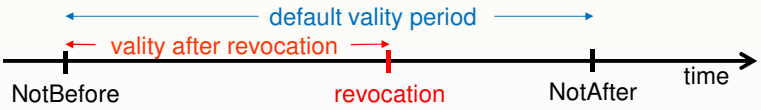
removeFromCRL (8)
privilegeWithdrawn (9)
AACompromise (10)

- Publication and distribution of CRLs
  - Each CA keeps its CRL and allows public access to it
  - CAs exchange CRLs to facilitate their widespreading

CRL and Delta CRL



Validity of signatures



- A signature is **valid** if it was generated during the **validity period** of the corresponding pub key certificate
  - The validity period starts on the certificate's **NotBefore** date field
  - By default, the validity ends on the **NotAfter** date field
    - Unless revoked
- A private key can be used out of that period
  - But the signature it produces is invalid
- A public key certificate can be used anytime
  - Namely, after the validity period to check past signatures

Distribution of public key certificates

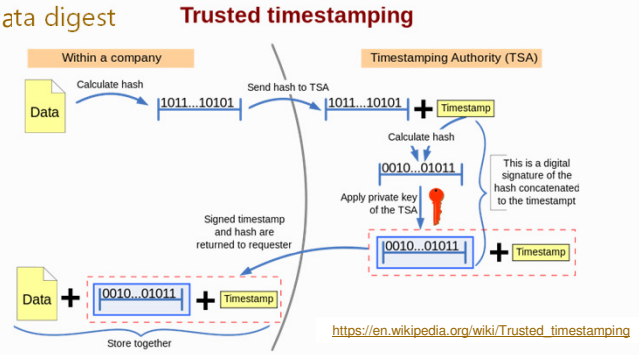
- Integrated with systems or applications
- Directory systems
  - Large scale
    - ex. X.500 through LDAP
  - Organizational
    - ex. Windows 2000 Active Directory (AD)
- Together with signatures
  - Within protocols using certificates for peer authentication
    - e.g. secure communication protocols (SSL, IPSec, etc.)
  - As part of document signatures
    - PDF/Word/XML, etc. documents, MIME mail messages

Distribution of public key certificates

- Explicit (voluntarily triggered by users)
- User request to a service for getting a required certificate
  - e.g. request sent by e-mail
  - e.g. access to a personal HTTP page
- Useful for creating certification chains for frequently used terminal certificates
  - e.g. certificate chains for authenticating with the Cartão de Cidadão

Time Stamping Authority (TSA)

- A service that provides signatures over a timestamp
  - Linked with a data digest



- This is useful for adding trust to a data signature date
  - The signature date becomes linked to the signed data

PKI (Public Key Infrastructure)

- Infrastructure for enabling the use of keys pairs and certificates
  - Creation of asymmetric key pairs for each enrolled entity
    - Enrolment policies
    - Key pair generation policies
  - Creation and distribution of public key certificates
    - Enrolment policies
    - Definition of certificate attributes
  - Definition and use of certification chains (or paths)
    - Insertion in a certification hierarchy
    - Certification of other CAs
  - Update, publication and consultation of CRLs
    - Policies for revoking certificates
    - Online CRL distribution services
    - Online OCSP services
  - Use of data structures and protocols enabling inter-operation among components / services / people

PKI:

Example: Cartão de Cidadão policies

- Enrollment
  - In loco, personal enrolment
- Multiple key pairs per person
  - One for authentication
  - One for signing data
  - Generated in smartcard, not exportable
  - Require a PIN in each operation
- Certificate usage (authorized)
  - Authentication
    - SSL Client Certificate, Email (Netscape cert. type)
    - Signing, Key Agreement (key usage)
  - Signature
    - Email (Netscape cert. type)
    - Non-repudiation (key usage)
- Certification path
  - PT root CA below global root (before 2020)
  - PT root CA (after 2020)
  - CC root CA below PT root CA
  - CC Authentication CA and CC signature CA below CC root CA
- CRLs
  - Signature certificate revoked by default
    - Removed if owner explicitly requires the usage of signatures
  - Certificates revoked upon a owner request
    - Requires a revocation PIN
  - CRL distribution points explicitly mentioned in each certificate

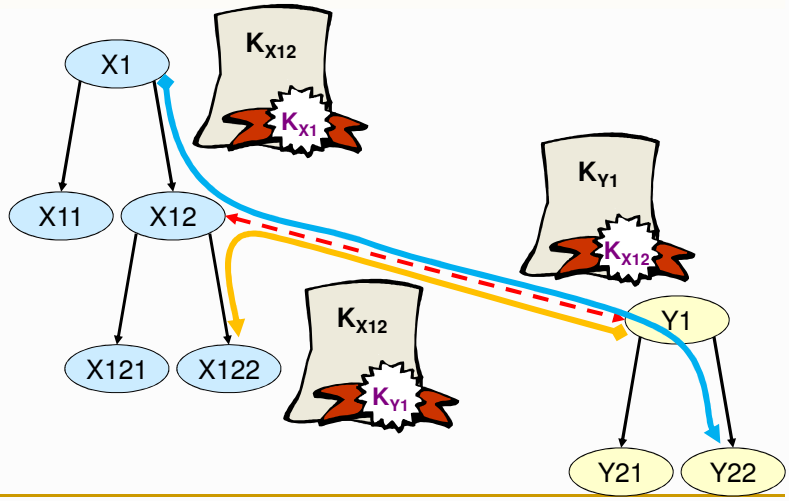
PKI:

Trust relationships

- A PKI defines trust relationships in two different ways
  - By issuing certificates for the public key of other CAs
    - Hierarchically below; or
    - Not hierarchically related
  - By requiring the certification of its public key by another CA
    - Above in the hierarchy; or
    - Not hierarchically related
- Usual trust relationships
  - Hierarchical
  - Crossed (A certifies B and vice-versa)
  - Ad-hoc (mesh)
    - More or less complex certification graphs

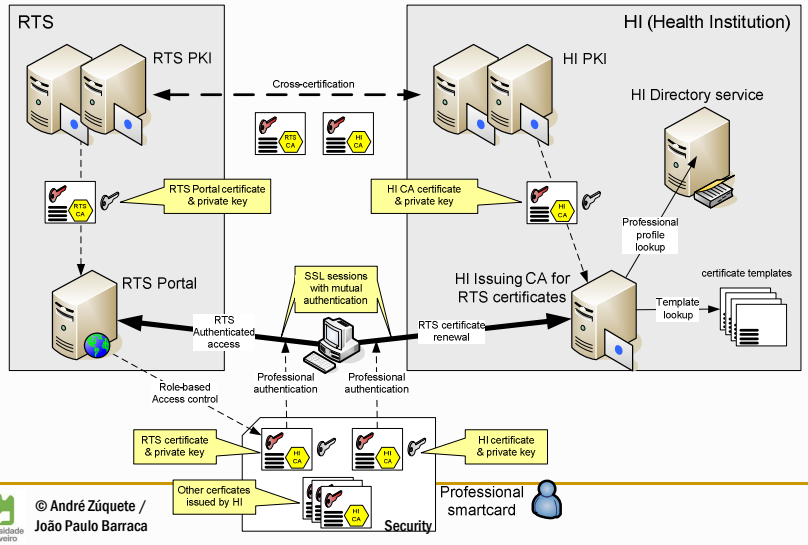
PKI:

Hierarchical and crossed certifications



Cross-certification of PKIs:

A practical example



Additional documentation

- [RFC 3280] Internet X.509 Public Key Infrastructure: Certificate and CRL Profile
- Other RFCs
  - **[RFC 4210]** Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)
  - **[RFC 4211]** Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)
  - **[RFC 3494]** Lightweight Directory Access Protocol version 2 (LDAPv2) to Historic Status
  - **[RFC 6960]** X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP
  - **[RFC 2585]** Internet X.509 PKI Operational Protocols: FTP and HTTP
  - **[RFC 2587]** Internet X.509 PKI LDAPv2 Schema
  - **[RFC 3029]** Internet X.509 PKI Data Validation and Certification Server Protocols
  - **[RFC 3161]** Internet X.509 PKI Time-Stamp Protocol (TSP)
  - **[RFC 3279]** Algorithms and Identifiers for the Internet X.509 PKI Certificate and Certificate Revocation List (CRL) Profile
  - **[RFC 3281]** An Internet Attribute Certificate Profile for Authorization
  - **[RFC 3647]** Internet X.509 PKI Certificate Policy and Certification Practices Framework
  - **[RFC 3709]** Internet X.509 PKI: Logotypes in X.509 Certificates
  - **[RFC 3739]** Internet X.509 PKI: Qualified Certificates Profile
  - **[RFC 3779]** X.509 Extensions for IP Addresses and AS Identifiers
  - **[RFC 3820]** Internet X.509 PKI Proxy Certificate Profile