Management of Asymmetric keys

Problems to solve

Ensure proper and correct use of asymmetric key pairs

Privacy of private keys
- To ensure authenticity
- To prevent the repudiation of digital signatures

Correct distribution of public keys
- To ensure confidentiality
- To ensure the correct validation of digital signatures

Problems to solve

Temporal evolution of entity <-> key pair mappings

To tackle catastrophic occurrences

- e.g. loss of private keys

To tackle normal exploitation requirements

- e.g. refresh of key pairs for reducing impersonation risks

Problems to solve

Ensure a proper generation of key pairs

Random generation of secret values
- So that they cannot be easily predicted

Increase efficiency without reducing security
- Make security mechanisms more useful
- Increase performance

Goals

1. **Key pair generation**
   - When and how should they be generated

2. **Handling of private keys**
   - How do I maintain them private

3. **Distribution of public keys**
   - How are they correctly distributed worldwide

4. **Lifetime of key pairs**
   1. When will they expire
   - Until when should they be used
   - How can I check the obsolesce of a key pair

Generation of key pairs: Design principles

Good random generators for producing secrets

Result is indistinguishable from noise

- All values have equal probability
- No patterns resulting from the iteration number or previous values

Example: Bernoulli ½ generator

- Memoryless generator
- $P(b=1) = P(b=0) = ½$
- Coin toss

Generation of key pairs: Design principles

Facilitate without compromising security

Efficient public keys

- Few bits, typically 2^k+1 values (3, 17, 65537)
- Accelerates operations with public keys
- No security issues

Generation of key pairs: Design principles

Self-generation of private keys

Maximizes privacy as no other party will be able to use a given private key

- Only the owner has the key
- Even better: The owner doesn’t have the key, but may use the key

Principle can be relaxed when not involving signature generation

- Where there are not issues related with non repudiation

Handling of private keys

Correctness

The private key represents a subject

- E.g. a citizen
- 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 that use it
- Authentication in the applications that allow to use it

Handling of private keys

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

Example: SmartCards

- We ask the SmartCard to cipher/decipher something
- The private key never leaves the SmartCard

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
- Manual
- Ad-hoc using digital certificates

Distribution of public keys

Problem: How to ensure the correctness of the public key?

Trustworthy dissemination of public keys

- Trust paths / graphs

If A trusts K_X^+, and B trusts A, then B trusts K_X^+

- Certification hierarchies / graphs
  - With the trust relations expressed between entities
  - Certification is unidirectional!

Public key (digital) certificates

Digital Document issued by a Certification Authority (CA)

Binds a public key to an entity
- Person, server or service

Are public documents
- Do not contain private information, only public one
- Can have additional binding information (URL, Name, email, etc..)

Are cryptographically secure
- Digitally signed by the issuer, cannot be changed
- Have a cryptographic fingerprint for fast validation

Public key (digital) certificates

Can be used to distribute public keys in a trustworthy way

A certificate receiver can validate it

- With the CA's public key
- Can also validate the identification
- Validate the validity
- Validate is the key is being properly used

A certificate receiver trusts the behavior of the CA

- Therefore will trust the documents they sign
- When a CA associates a certificate to A. If the receiver trusts the CA, it will trust that the association of A is correct

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 encoding of X.509

Key pair usage

The public certificate binds the key pair to a usage profile

- Public keys are seldom multi-purpose

Typical usage profiles

- 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

- Companies, not for profit organizations or governmental
- With the task of validating the relation between key and identity

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
- Programmatic interfaces to verify the current state of a certificate

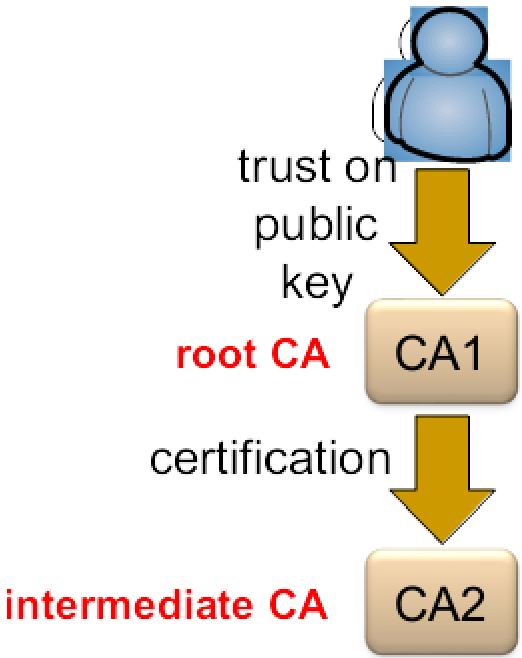
Trusted Certification Authorities

Intermediate CAs: CAs certified by other trusted CAs

- Using a certificate
- Enable the creation of certification hierarchies

Trusted anchor (or certification root) : One has a trusted public key

- Usually implemented by self-certified certificates
  - Issuer = Subject
- Manual distribution
  - e.g. within browsers code (Firefox, Chrome, etc.), OS, distribution...



Certificate Viewer: "www.ua.pt"

General | Details

This certificate has been verified for the following uses:

SSL Client Certificate

SSL Server Certificate

Issued To

Common Name (CN)	www.ua.pt
Organization (O)	Universidade de Aveiro
Organizational Unit (OU)	sTIC
Serial Number	06:B4:17:0C:D7:EF:AC:9F:A3:79:9A:78:0E:7E:5A:8C

Issued By

Common Name (CN)	TERENA SSL CA 3
Organization (O)	TERENA
Organizational Unit (OU)	<Not Part Of Certificate>

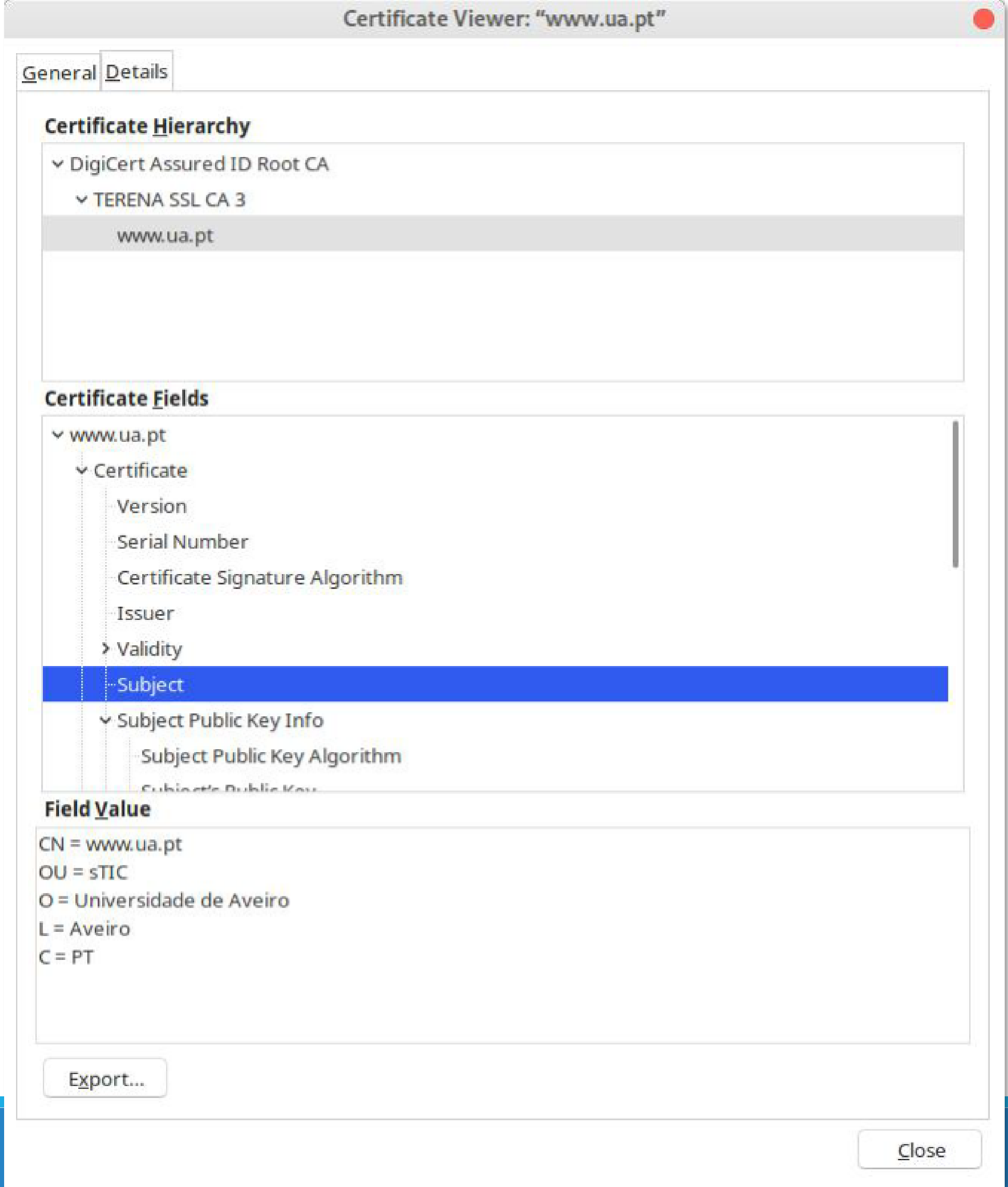
Period of Validity

Begins On	May 27, 2019
Expires On	June 3, 2021

Fingerprints

SHA-256 Fingerprint	6C:BA:BD:A1:7E:A9:8D:EA:7B:18:22:44:EC:71:D5:41:4D:08:D4:A6:FC:48:1B:3C:9B:05:EB:DA:69:A6:A5:EE
SHA1 Fingerprint	17:79:15:B5:0E:E0:34:51:2D:FA:DE:DF:77:1E:E1:0A:B3:4B:2F:2B

Close


Certificate Viewer: "www.ua.pt"
General
Details
Certificate Hierarchy
DigiCert Assured ID Root CA
TERENA SSL CA 3
www.ua.pt
Certificate Fields
www.ua.pt
Certificate
Version
Serial Number
Certificate Signature Algorithm
Issuer
Validity
Subject
Subject Public Key Info
Subject Public Key Algorithm
Field Value
CN = www.ua.pt
OU = sTIC
O = Universidade de Aveiro
L = Aveiro
C = PT
Export...
Close

Certificate Viewer: "TERENA SSL CA 3"

General | Details

This certificate has been verified for the following uses:

SSL Certificate Authority

Issued To

Common Name (CN)	TERENA SSL CA 3
Organization (O)	TERENA
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	08:70:BC:C5:AF:3F:DB:95:9A:91:CB:6A:EE:EF:E4:65

Issued By

Common Name (CN)	DigiCert Assured ID Root CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

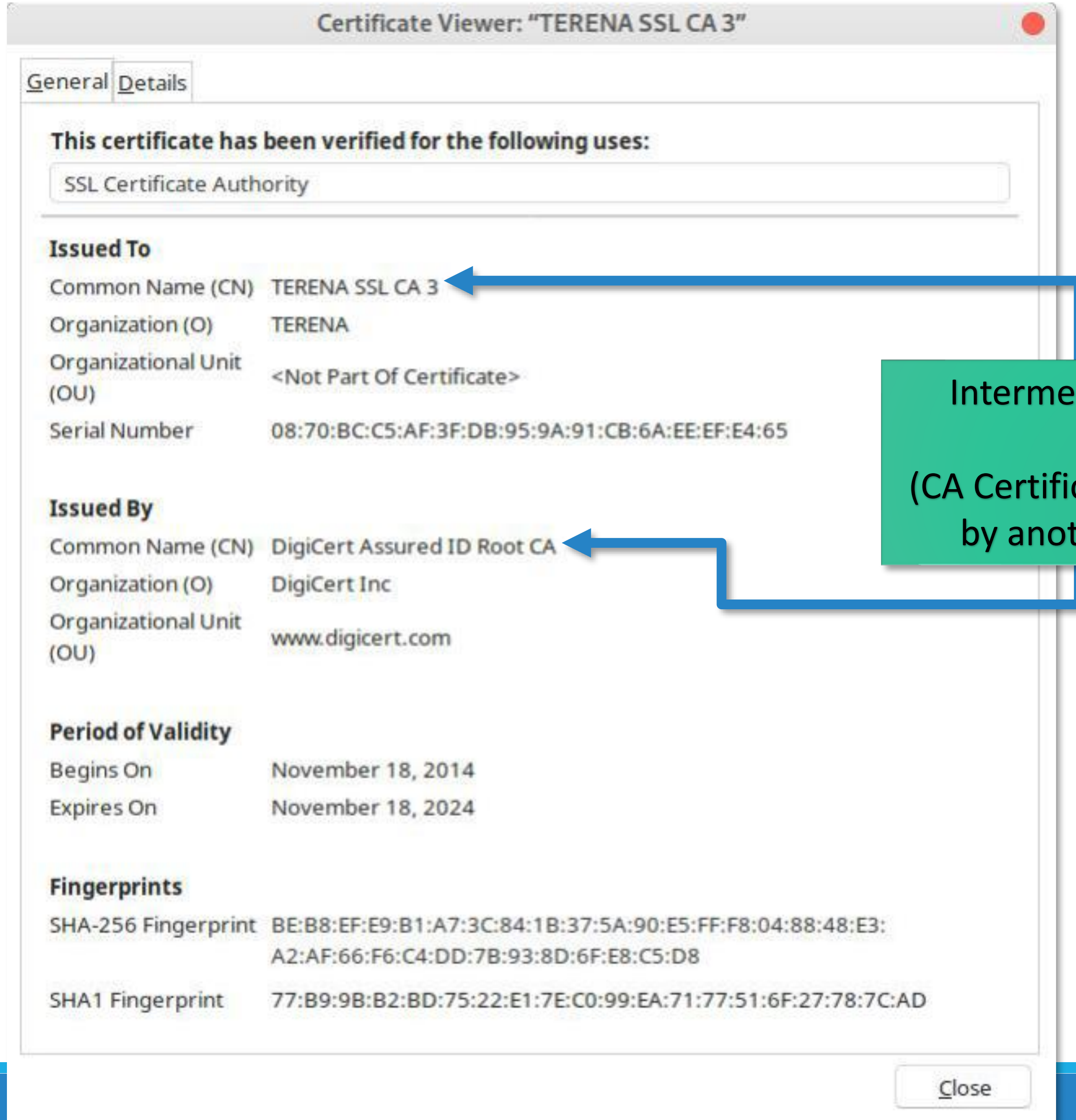
Period of Validity

Begins On	November 18, 2014
Expires On	November 18, 2024

Fingerprints

SHA-256 Fingerprint	BE:B8:EF:E9:B1:A7:3C:84:1B:37:5A:90:E5:FF:F8:04:88:48:E3:A2:AF:66:F6:C4:DD:7B:93:8D:6F:E8:C5:D8
SHA1 Fingerprint	77:B9:9B:B2:BD:75:22:E1:7E:C0:99:EA:71:77:51:6F:27:78:7C:AD

Close



Intermediate CA

(CA Certificate signed by another CA)

Certificate Viewer: "DigiCert Assured ID Root CA"

General | Details

This certificate has been verified for the following uses:

SSL Certificate Authority

Issued To

Common Name (CN)	DigiCert Assured ID Root CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com
Serial Number	0C:E7:E0:E5:17:D8:46:FE:8F:E5:60:FC:1B:F0:30:39

Issued By

Common Name (CN)	DigiCert Assured ID Root CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

Period of Validity

Begins On	November 10, 2006
Expires On	November 10, 2031

Fingerprints

SHA-256 Fingerprint	3E:90:99:B5:01:5E:8F:48:6C:00:BC:EA:9D:11:1E:E7:21:FA:BA:35:5A:89:BC:F1:DF:69:56:1E:3D:C6:32:5C
SHA1 Fingerprint	05:63:B8:63:0D:62:D7:5A:BB:C8:AB:1E:4B:DF:B5:A8:99:B2:4D:43

Close

Root CA

(Certificate is self signed)

Certification hierarchies: PEM (Privacy Enhanced Mail) model

Distribution of certificates for Privacy-enhanced Electronic Mail

IETF Proposed Standard in 1993 (RFC1421-1423)

Worldwide hierarchy (monopoly model)

- Single root IPRA (Internet Policy Registration Authority)
- Several PCA (Policy Creation Authorities) bellow the root
- Several CA below each PCA
  - Possibly belonging to organizations or companies
- Certification paths

Certification hierarchies: PEM (Privacy Enhanced Mail) model

Model was never actively deployed

- Except for a small number of implementations (90s)

Forest of hierarchies below CAs without a root PCA

- Independent hierarchies with an independent root CA
- Oligarchy

Each root CA negotiates the distribution of its public key along with some applications or operating systems

- ex. Browsers, Operating Systems

Certification hierarchies: PGP (Pretty Good Privacy) Model

Web of trust model

- And not a forest model

No central trustworthy authorities

- Each person is a potential certifier
- Anyone can certify a public key (issue a certificate) and publish the signature for others

People uses two kinds of trust

- Trust in the keys they know
  - Validated using any means (FAX, telephone, direct meeting, etc.)
- Trust in the correct behavior of certifiers
  - Assuming they know what they are doing when issuing a certificate

Certification hierarchies: PGP (Pretty Good Privacy) Model

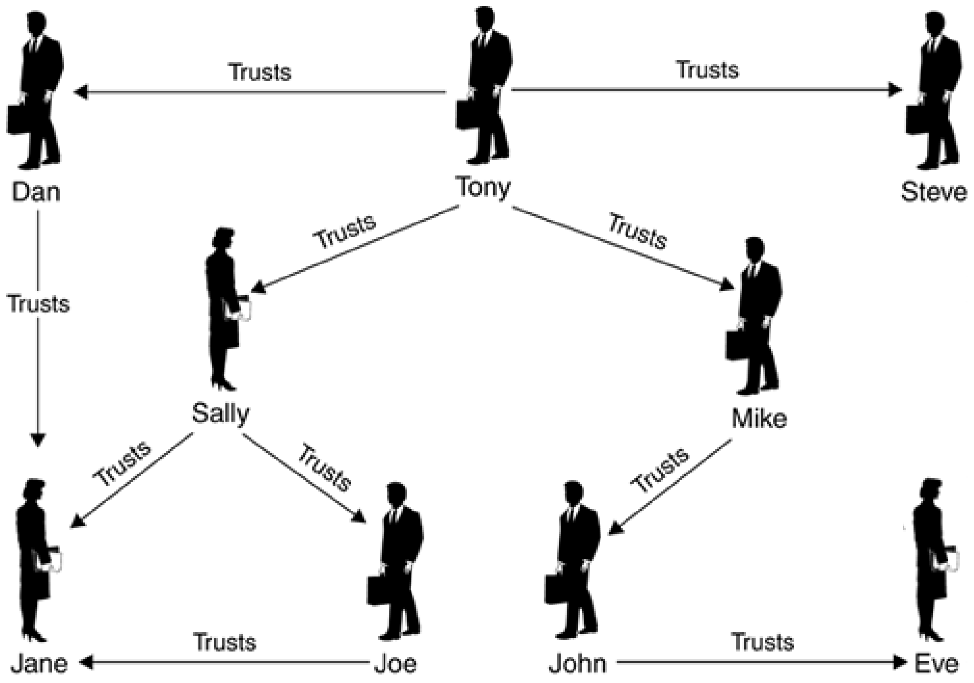
Transitive trust

If

Mike trusts John is a correct certifier;
and
John certified the public key of Eve,

Then

Mike trusts Eve's public key



PGP public key certificates: Validity vs. trust

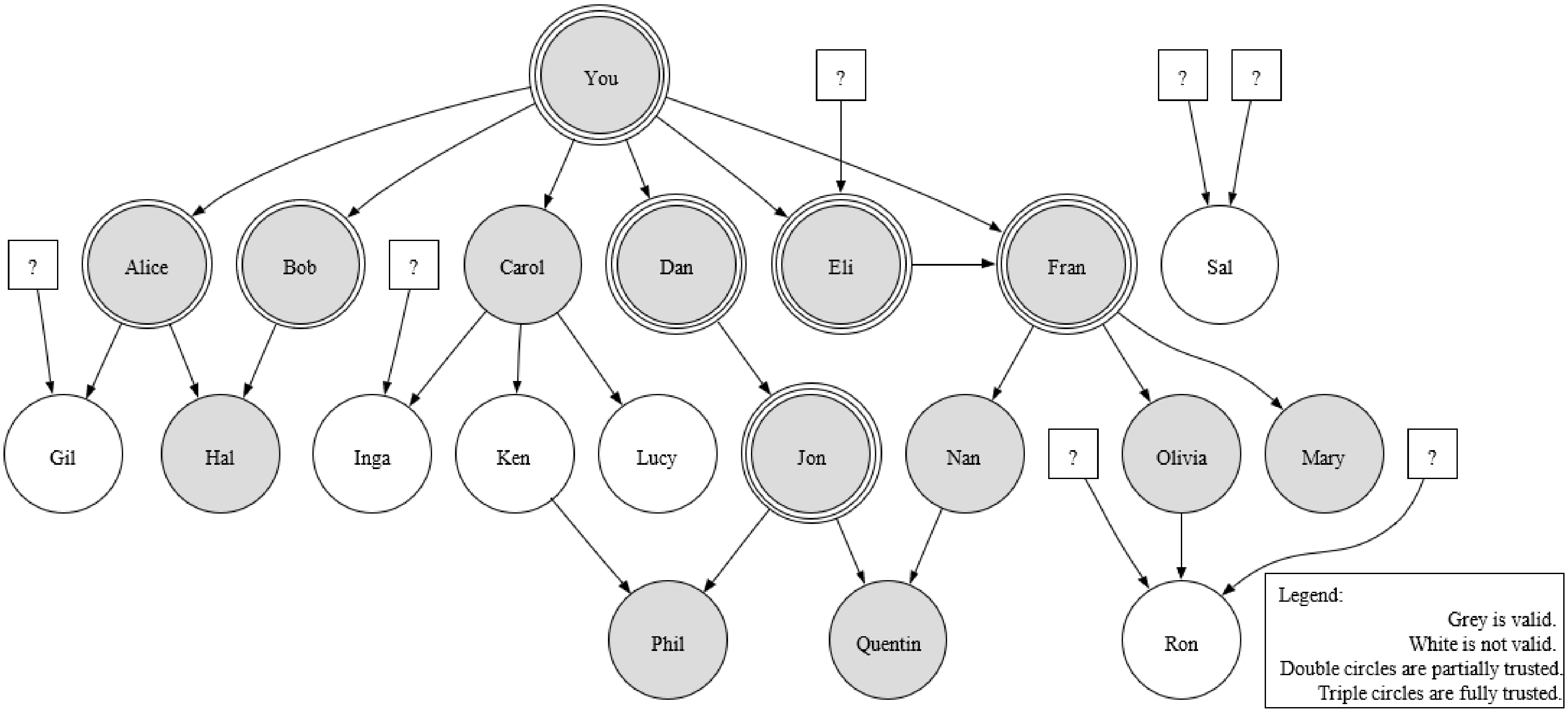
Trust: How much one trusts the other person

- Trust is unidirectional, personal and subjective
- Levels:
  - Ultimate (our own keys, we have the private key)
  - Complete trust
  - Marginal trust
  - Notrust (or Untrusted)

Validity: How much verification this key has (eg, A regarding user E)

- **Valid:** A completely trusts B, or marginally trusts C and D; D or B and C signed E key
- **Marginally valid**: A marginally trusts B and B signed E key
- **Invalid**: No path

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 holders of certificates is unknown
  - Therefore we cannot contact them to eliminate specific certificates

Solutions

- Certificates with a validity period (not before, not after)
- Certificate revocation lists
  - To revoke certificates before expiring their validity

Certificate revocation lists (CRL)

Base or delta

- Complete / differences

Signed lists of certificate (identifiers) prematurely invalidated

- Must be regularly consulted by certificate holders
- OCSP protocol for single certificate validation
  - RFC 2560
- Can tell the revocation reason

RFC 3280

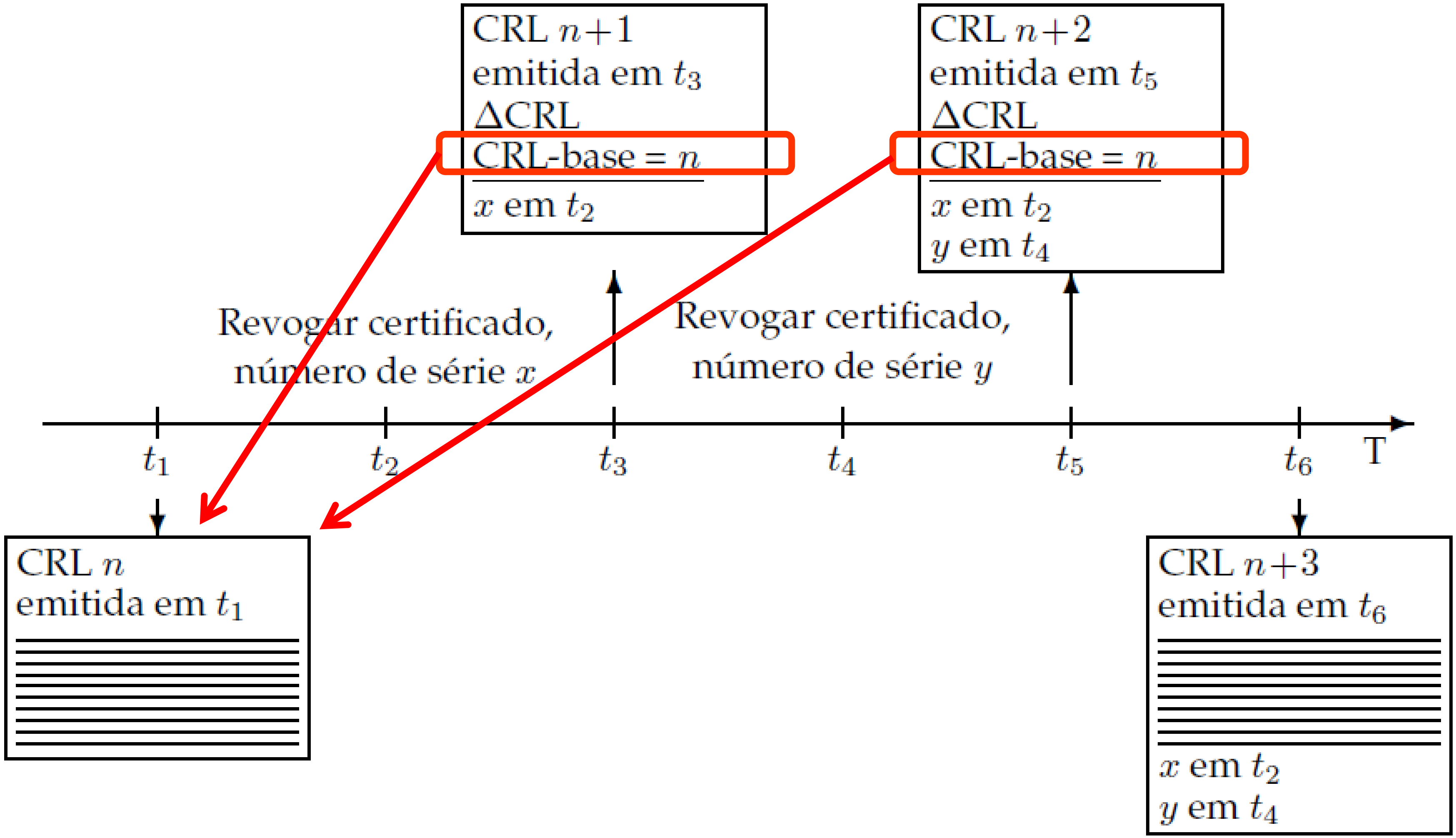
unspecified (0)
keyCompromise (1)
CACompromise (2)
affiliationChanged (3)
superseded (4)
cessationOfOperation (5)
certificateHold (6)

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



Online Certificate Status Protocol

HTTP based protocol to assert certificate status

- Request includes the certificate serial number
- Response states if the certificate is revoked
  - Response is signed by the CA and has a validity
- One check per certificate

Request lower bandwidth to clients

- One check per certificate instead of a bulk download of the CRL

Involves higher bandwidth to CAs

- One check per certificate
- Privacy issues as the CA will know that a system is accessing a service

OCSP Stapling

- Including a recently signed timestamp in the server response to assert validity
- Reduces verification delay and load on CA

Distribution of public key certificates

Transparent (integrated with systems or applications)

- Directory systems
  - Large scale (ex. X.500 through LDAP)
  - Organizational (ex. Windows 2000 Active Directory (AD), Manually (UA IDP))
- On-line: within protocols using certificates for peer authentication
  - eg. secure communication protocols (TLS, IPSec, etc.)
  - eg. digital signatures within MIME mail messages or within documents

Explicit (voluntarily triggered by users)

- User request to a service for getting a required certificate
  - eg. request sent by e-mail
  - eg. access to a personal HTTP page

PKI (Public Key Infrastructure) (1/2)

Infrastructure for enabling a proper use of asymmetric keys and public key 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

PKI (Public Key Infrastructure) (2/2)

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: Citizen Card

Enrollment
- In loco, personal enrolment

Multiple key pairs per person
- One for authentication
- One for signing data
- Both generated inside smartcard, not exportable
- Both require a PIN to be used 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
- Uses a well-known, widely distributed root certificate
  - GTE Cyber Trust Global Root
- PT root CA below GTE
- CC root CA below PT root CA
- CC Authentication CA and CC signature CA below CC root CA

CRLs
- Signature certificate revoked by default
  - Revocation is removed if the CC owner explicitly requires the usage of CC digital signatures
- All certificates are revoked upon a owner request
  - Requires a revocation PIN
- CRL distribution points explicitly mentioned in each certificate

PKI Trust relationships

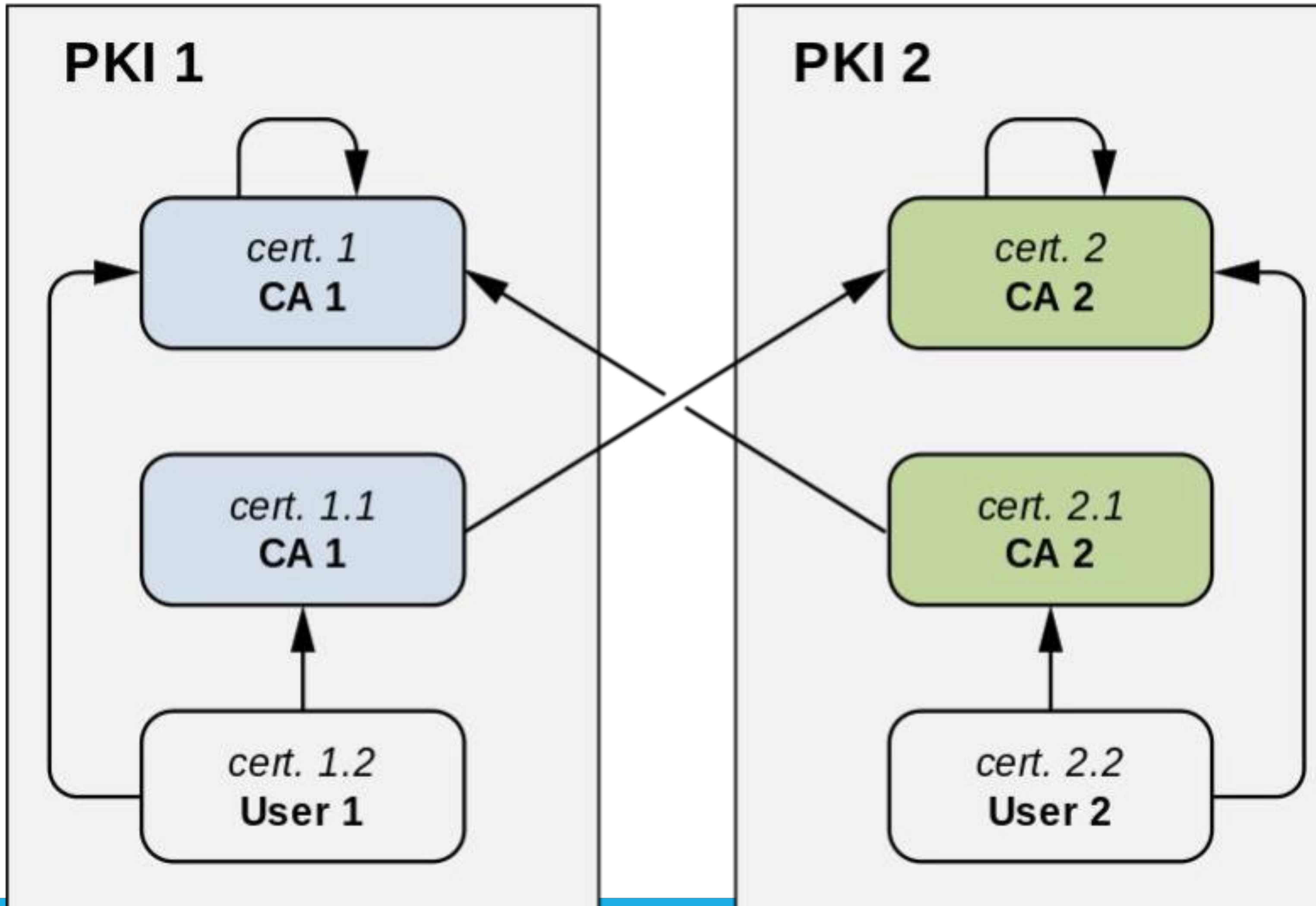
A PKI defines trust relationships in two different ways

- By issuing certificates for the public key of other CAs
  - Hierarchically below; or
  - Not hierarchically related
- By requiring the certification of its root 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



Certificate Pinning

If attacker has access to trusted Root, it can impersonate every entity

- Manipulate a trusted CA into issuing certificate (unlikely)
- Inject custom CA certificates in the victim's database (likely)

Certificate Pinning: add the fingerprint of the PubK to the source code

- Fingerprint is a hash (e.g. SHA256)
- Also store the URL to access

Validation process:

- Certificate must be valid according to local rules
- Certificate must have a public they with the given fingerprint

Certification Transparency (RFC 6962)

Problems

- CAs can be compromised (e.g., DigiNotar)
  - By malicious attackers
  - By governments, etc…
- Compromise is difficult to detect
  - Result in the change of assumptions associated to the behavior of the CA
  - Owner will selfdom know

Definition: Global system records all public certificates created

- Ensure that only a single certificate has the correct roots
- Stores the entire certification chain of each certificate
- Presents this information for auditing
  - Organizations or ad-hoc by the end users