Management of Asymmetric key pairs

SIO

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

André Zúquete

Problems to solve

Ensure proper and correct use of asymmetric key pairs

- Privacy of private keys
  - To ensure confidentiality (when used for decryption)
  - To prevent the repudiation of digital signatures (when used for signature issuing)
- Correct distribution of public keys
  - To ensure confidentiality (when used for encryption)
  - To ensure the correctness of digital signatures (when used for signature validation)

Problems to solve

Temporal evolution of (entity, key pair) mappings

- To tackle catastrophic occurrences
  - Loss of private keys
- To tackle normal exploitation requirements
  - Renewal of key pairs for reducing discovery risks
  - End of the bound between entity and key pair (e.g. professional relationship)

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

- Key pair generation
  - When and how should they be generated
- Handling of private keys
  - How do I use them, while maintaining them private
- Distribution of public keys
  - How are they correctly distributed worldwide
- Lifetime of key pairs
  - 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

Large, complex passwords for protecting secrets

- When randomly-generated secrets are stored in password-protected readable repositories

- When secrets are deterministically computed from a password

Generation of key pairs: design principles

Facilitate without compromising security

- Efficient RSA public keys
  - Few 1 bits, typically 2k+1 prime values (3, 17, 65537)
  - Accelerates operations with public keys
  - Cost is proportional to the number of 1 bits
  - No security issues

Generation of key pairs: design principles

Self-generation of private keys

- Maximizes privacy as no other party ever knew the private key
  - Only the owner has the key
  - Even better: The owner doesn’t know the key, but may use the key
- Principle can be relaxed when not involving signature generation
  - Where there are no issues related with non-repudiation
  - In confidential communications it allows to maintain the readability of encrypted messages

Handling of private keys

Correctness

- The private key represents a subject
  - e.g., a citizen, a service
  - 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 get their value

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 decrypts/signs with the private key

- Example: SmartCards, FIDO2 tokens
  - We ask the SmartCard to decrypt/sign 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

Certification concept

- Transitive trust
  - If A trusts K_X^+, and B trusts A, then B trusts K_X^+
  - Trust paths / graphs
- 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

Public key (digital) certificates

Can be used to distribute public keys in a trustworthy way

- A certificate receiver must validate it in many ways
  - With the CA's public key
  - Can also validate the identification
  - Validate the validity
  - Validate if the corresponding key pair 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 Alice
    - If the receiver trusts the CA
    - Then it will trust that the public key in the certificate belongs to Alice

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
- Textual encodings
  - PEM (Privacy Enhanced Mail)
  - base64 encoding of X.509

Key pair usage

- The public certificate binds the key pair to a usage profile
  - Private 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 (exclusively for CAs)
    - Certificate signing, CRL signing
  - Timestamping (exclusively for TSAs)

- 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
  - Have the task of validating the relation between key and identity

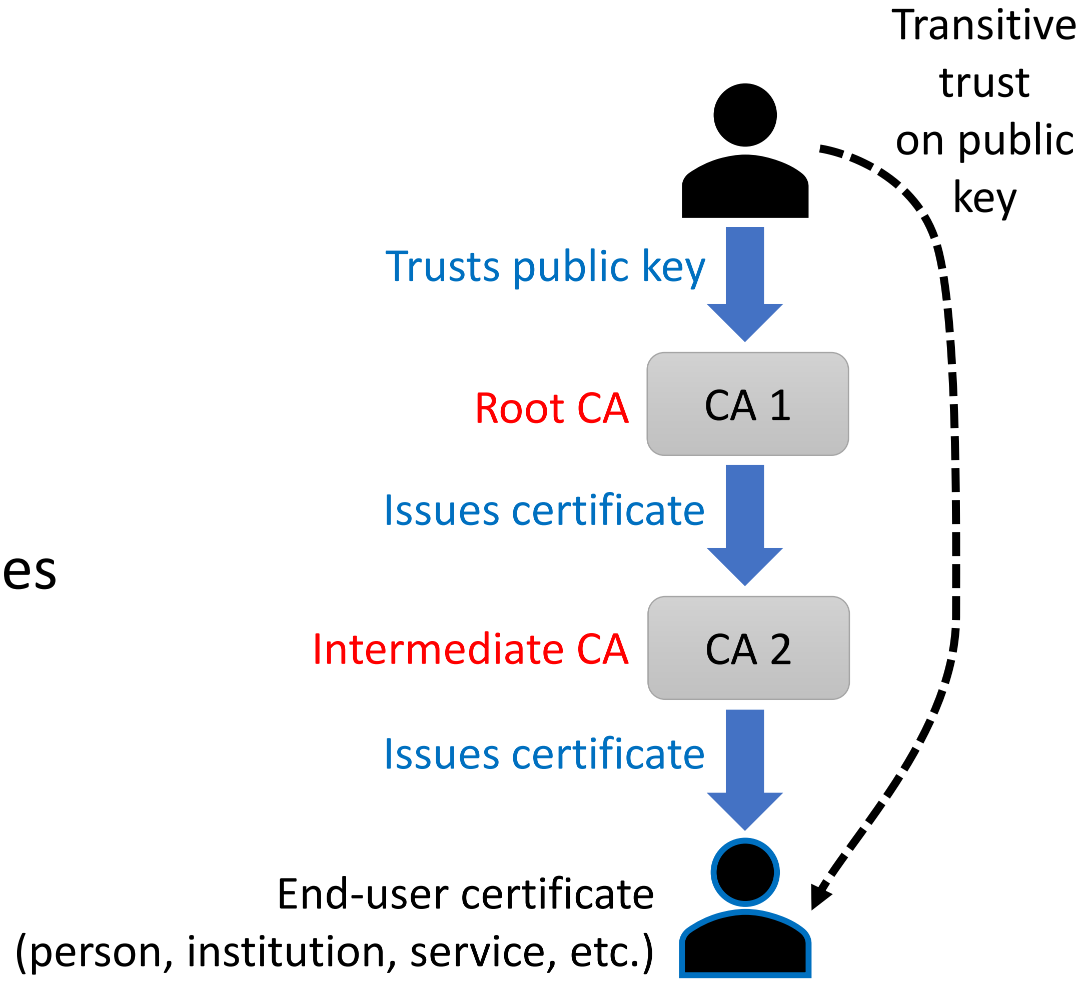
- Define policies and mechanisms for:
  - Issuing certificates
  - Revoking certificates
  - Distributing certificates
  - Issuing and distributing the corresponding private keys

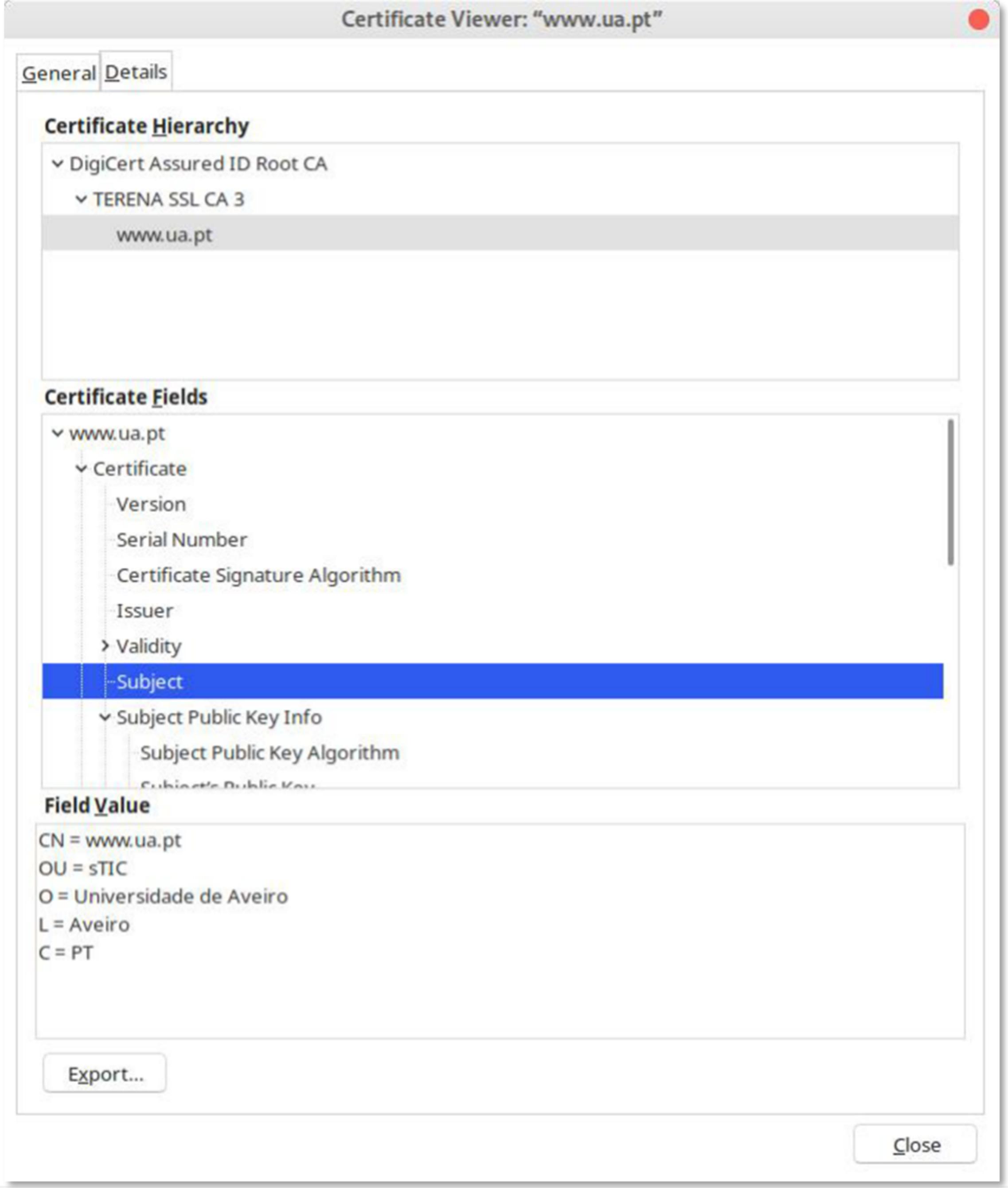
- Manage certificate revocation lists
  - Lists of revoked certificates
  - Programmatic interfaces to verify the current state of a certificate

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





Certificate Hierarchy

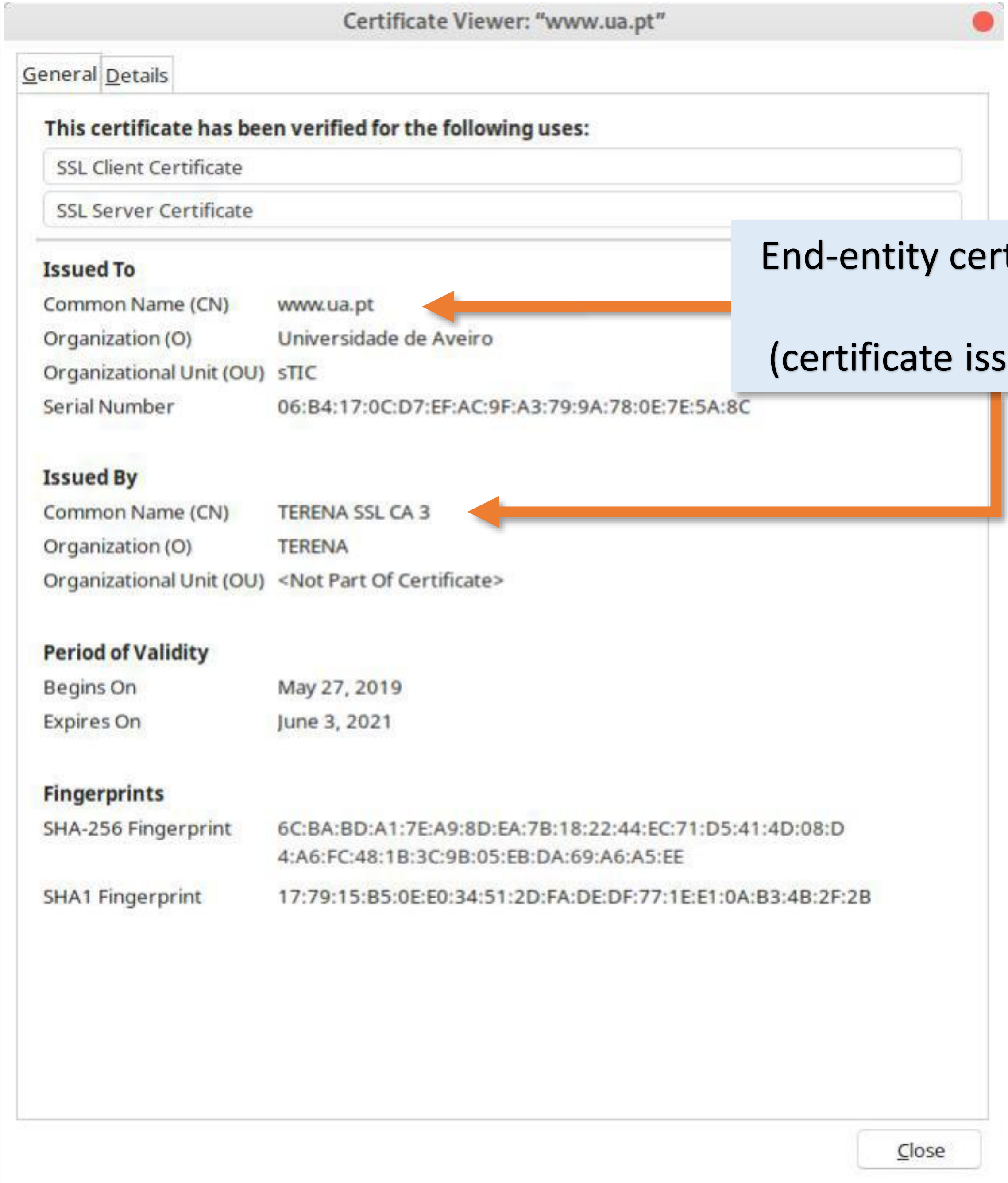
- DigiCert Assured ID Root CA
  - TERENA SSL CA 3
    - www.ua.pt

Certificate Fields

- www.ua.pt
  - Certificate
    - Version
    - Serial Number
    - Certificate Signature Algorithm
    - Issuer
    - Validity
    - Subject
    - Subject Public Key Info
      - Subject Public Key Algorithm
      - Subject's Public Key

Field Value

CN = www.ua.pt
OU = sTIC
O = Universidade de Aveiro
L = Aveiro
C = PT



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

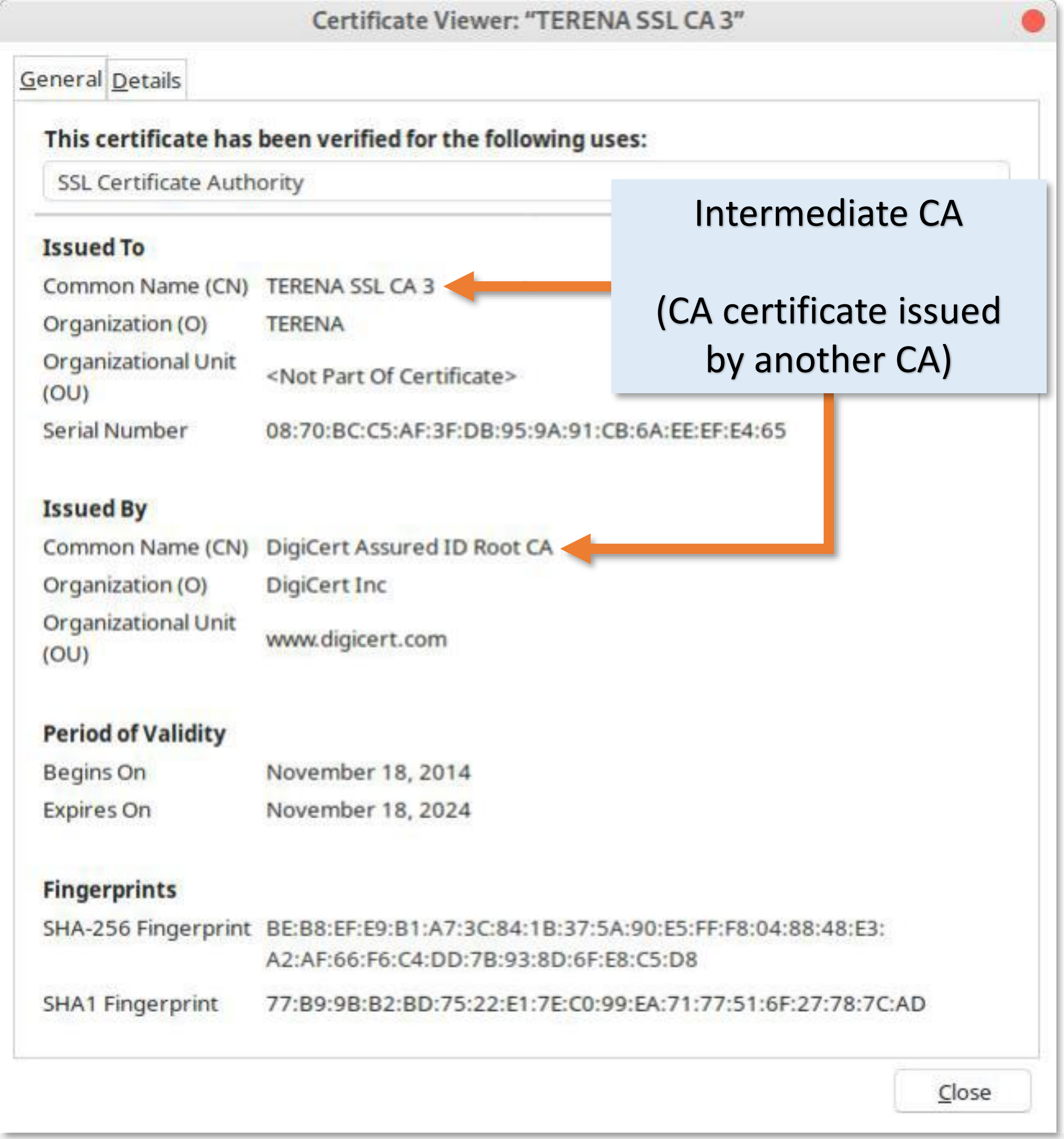
Begins On	May 27, 2019
Expires On	June 3, 2021

Fingerprints

SHA-256 Fingerprint	6C:BA:BD:A1:7E:A9:8D:EA:7B:18:22:44:EC:71:D5:41:4D:08:D 4: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

End-entity certificate (host)

(certificate issued by a CA)



Certificate Viewer: "TERENA SSL CA 3"

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

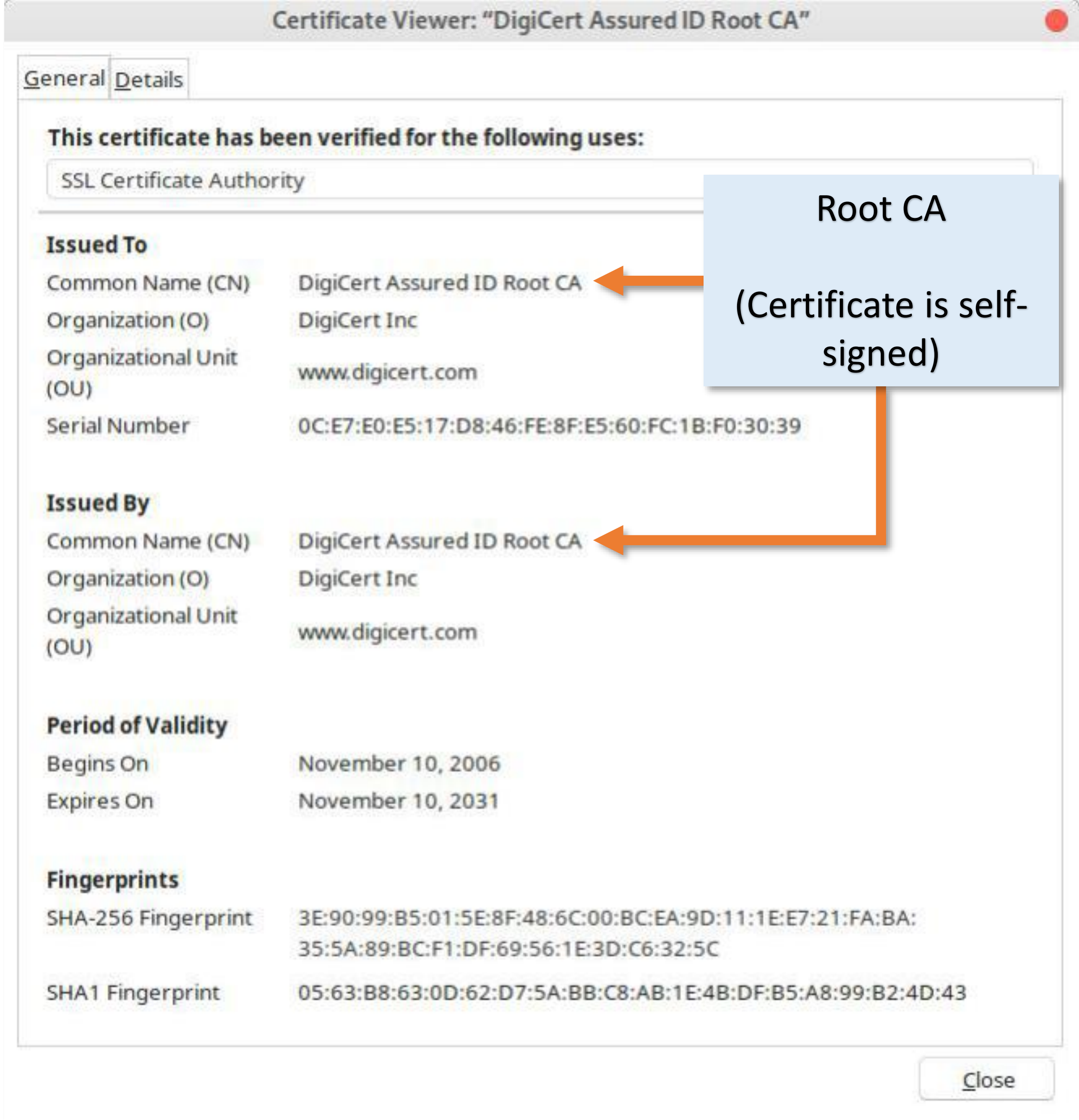
Begins On	November 18, 2014
Expires On	November 18, 2024

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

Intermediate CA

(CA certificate issued by another CA)



Certificate Viewer: "DigiCert Assured ID Root CA"

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

Root CA

(Certificate is self-signed)

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)
  - Voluntary use of certificate revocation lists
    - To revoke certificates before expiring their validity

Certificate revocation lists (CRL)

- Base or delta
  - Complete / differences

- Signed lists of certificates (identifiers) prematurely invalidated
  - Must be regularly consulted by certificate receivers
  - OCSP protocol for single certificate validation
    - RFC 6960
  - Can tell the revocation reason

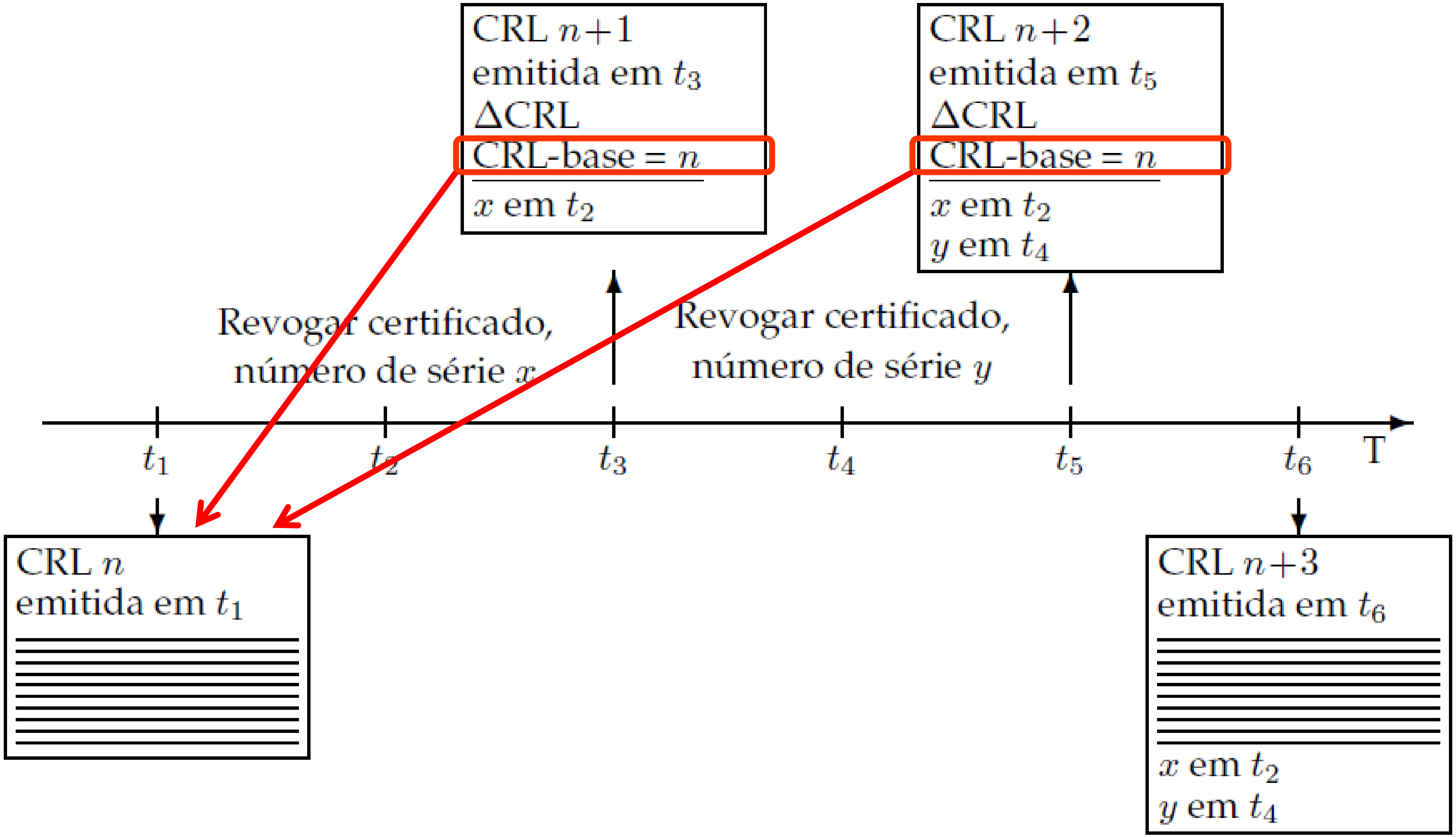
RFC 5280

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

Base 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

- Requires lower bandwidth to clients
  - One check per certificate instead of a bulk download of the CRL

- Involves higher computational overhead to CAs
  - One check per certificate
  - Privacy issues as the CA will know that a certificate is being used

OCSP stapling

- Add a recent OCSP response to certificate sent by a server
  - Reduces verification delay and load on CA
  - Avoids privacy issues

- Very useful in some specific scenarios
  - e.g. Wi-Fi network authentication

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
  - CRL issuing policies and distribution services
  - OCSP services

- Use of data structures and protocols enabling inter-operation among components / services / people

PKI Example: Portuguese 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
    - Self-Certified PT root CA
  - 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

Certificate Pinning

- If attacker has access to a 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)

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

Certification Transparency (RFC 9162)

- Problems
  - CAs can be compromised (e.g., DigiNotar)
    - By 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: a 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