DI-FCT-UNL
Segurança de Redes e Sistemas de Computadores
Network and Computer Systems Security

Mestrado Integrado em Engenharia Informática
MSc Course: Informatics Engineering
1º Sem., 2020/2021

X509 Authentication

- X509 Certificates and PKI (Public Key Infrastructure)

Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- **PKI - Public Key Infrastructure**
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- X509 Certificates
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- PKI - Public Key Infrastructure
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

X509 Authentication

- Based on Algorithms and Constructions for Digital Signatures of Identity Claims (Asymmetric or Public-Key Cryptography) and Trusted X509 Certificates and Certification Chains)

- Supported in Authentication Protocols involving:
  - Authentication claimants of digital identities:
    - SIGNERS if Authentication Proofs
  - Authentication Validation
    - Authenticators, using verification of signed identity authentication proofs

  - Typical use of digital signatures: standardized constructions (as studied before)

X509 Authentication: Signers as authenticated identity claimants

Signer (as the **authentication of claimant of digital identity claim**)

- Digital identity as unique identifier (UID)
- Control of the keypair generation process
- Must keep Private Key (related to exhibited X509 certificates) w/ required security assumptions
- Need that correspondent public-key must be known by the verifier (as the Authenticator peer): certified in X509 certificates
  - Certificates can be publicly exhibited: in a protocol, signers can send their public-key certificates
  - Certificates are issued by Trusted Certification Authorities

X509 Authentication: Identity Authenticators (as verifiers)

Authenticator (as the verifier of the claimed identity signatures):

- Need to know/obtain public key of the claimant UID in a trusted way, to verify the signed authentication claim
  - Can obtain from X509 public-key certificates
- For X509 Authentication, trust assumptions are based on obtaining and managing X509 certificates (as trusted public key certificates)
  - Role provided by Trusted Certification Authorities as issuers of X509 certificates
    - Also trusted by the verifiers

Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- **PKI - Public Key Infrastructure**
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

X.509 Standardization

X509: a standard framework, part of the ITU-T X500 standardization effort, initially targeted for:

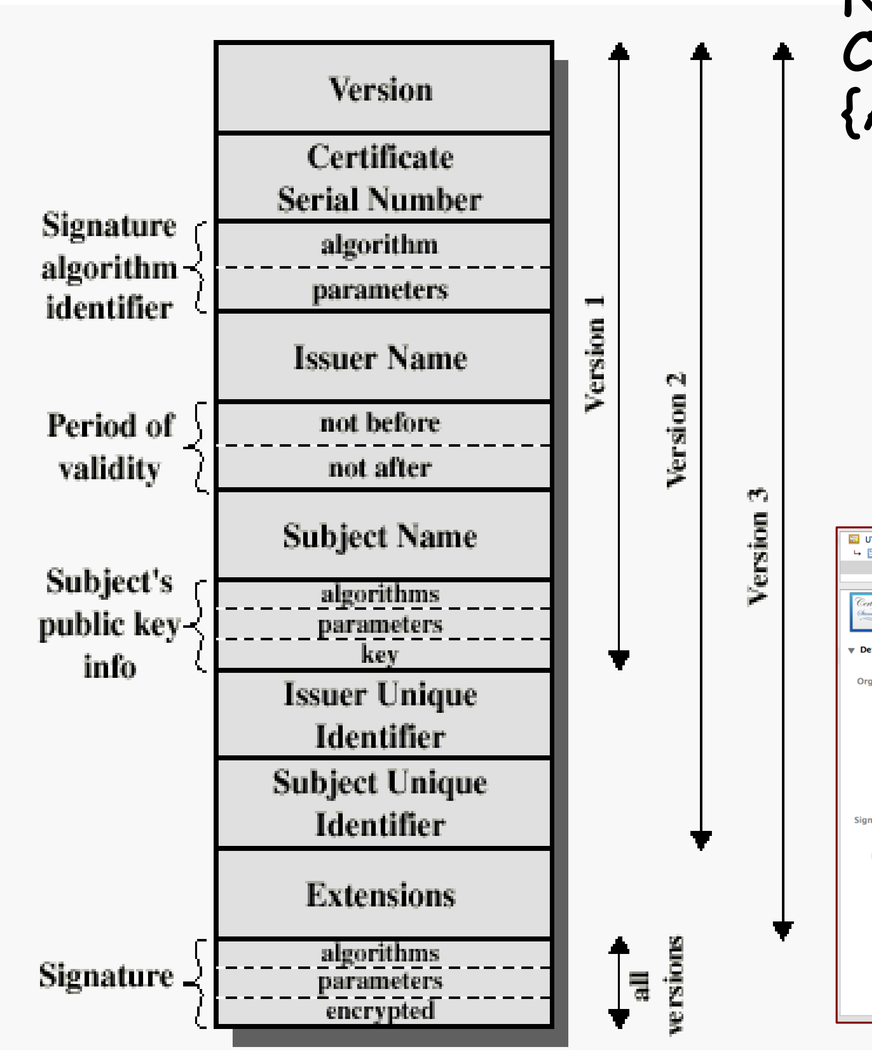
- Provision of **authentication services by X500 directory service**
- Standard representation of keys and public key certificates (formats and their attributes and data representation types), as well as recommended cryptography (algorithms and parameters)
  - Currently: X509v3 Certificates and X509v3 EV (Extended Validation) Certificates
  - Canonical Encoding Standardization
- Framework to address PKI systems (Public **K**ey **I**nfrastructures)
  - Processes, entity roles, interfaces)
  - Life cycle management of certificates: generation, enrollment, certification requests, certificate issuing, validation, revocation

---

Standardization: 1988, 1993 (v1), 1995 (v2), 2000 (v3), …

IETF RFC 2459 (Jan 1999) …… RFC 8399 (May/2018)

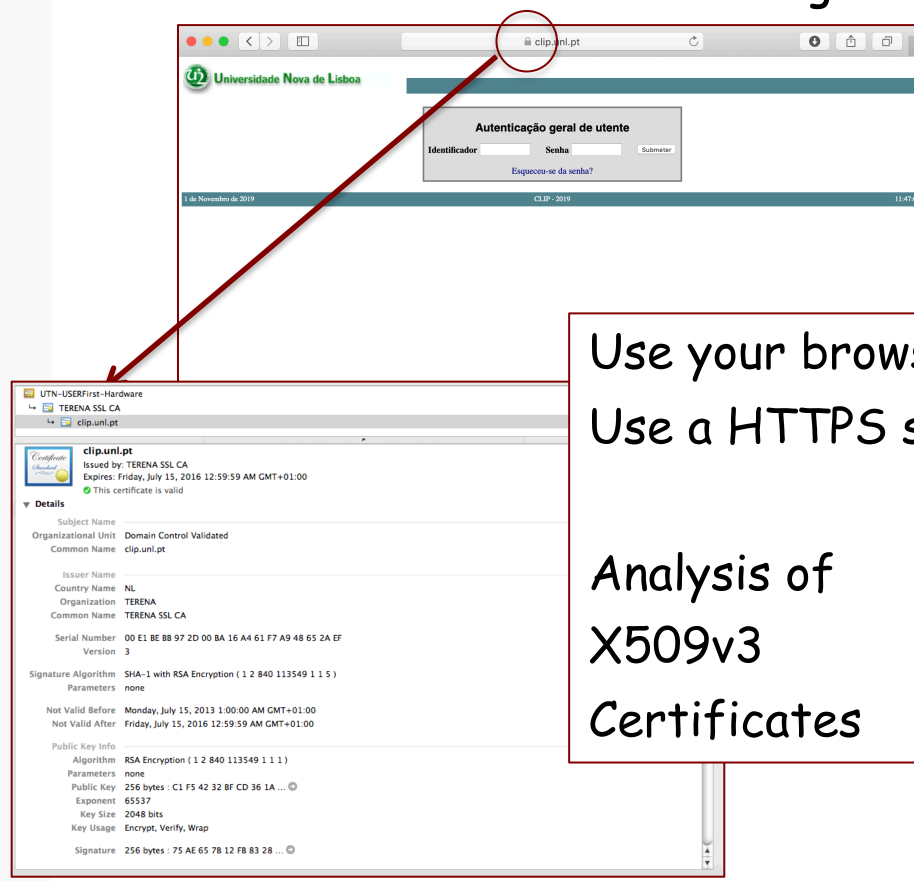
X.509 v3 Certificate: Structure, Attributes, Extensions, Classifiers



Notation:

CA <<A>> =

{A, V, SN, AI, CA, TA, KpubA}$_{SigCA}$



Use your browser

Use a HTTPS site

Analysis of X509v3 Certificates

X509 certificate (Extended attributes: improved in different versions)

X.509 Certificates

Each certificate contains:

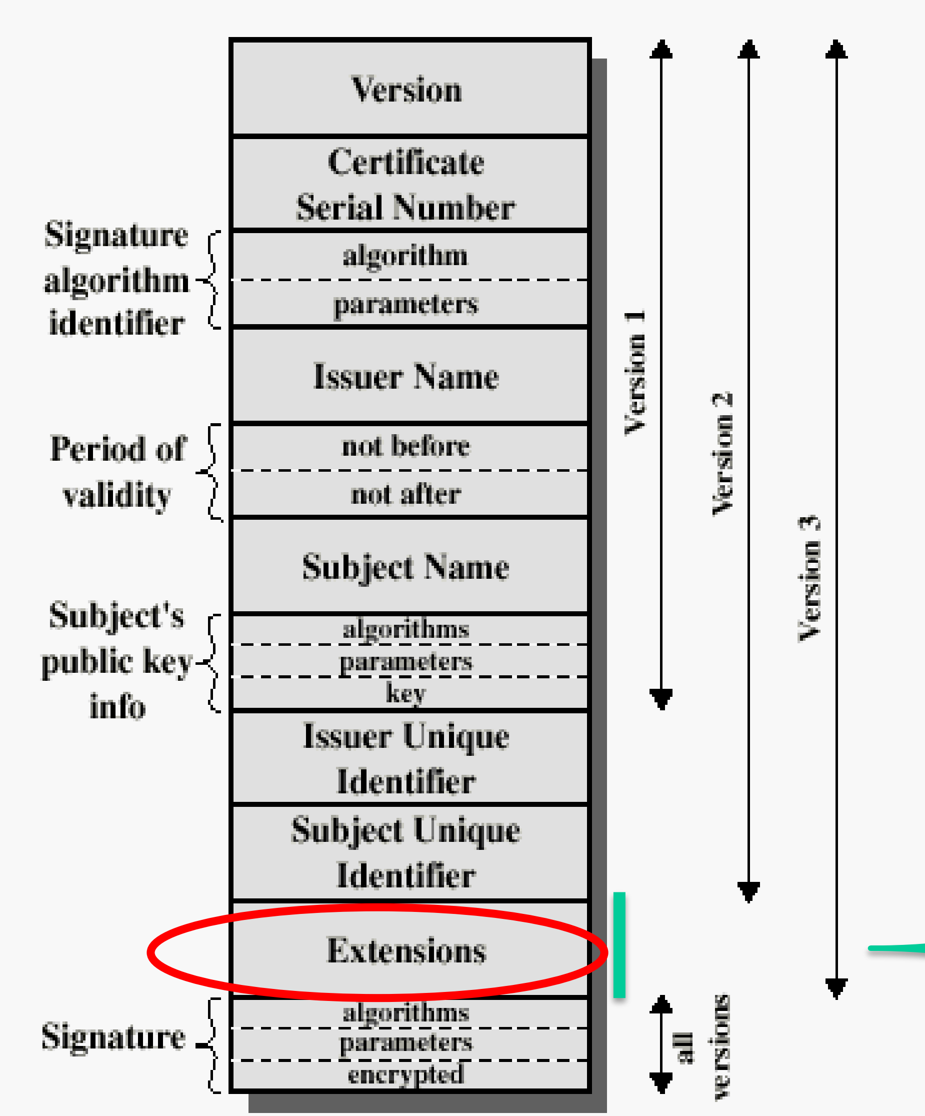
- The public key of a distinguished subject name (principal, user)
  - Subject name, Subject's public key information fields
- Other attributes with additional information as a list of other (field, value) pairs
  - Issuer UID, serial number, version, validity information, relevant information of cipher-suites used, verification control information, several extensions and fingerprints
- Signed with the private key of a CA.
  - Digital signature covering all the other fields
    - Hash of fields, signed with the CA private key

Discussion: see the different fields, policies and extended attributes in current X509v3 Certificates

Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- **PKI - Public Key Infrastructure**
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

X.509 Certificate and CRL Formats



X509 certificate (versions and attributes)

A set of one or more Extension Fields:

- Key Usage
- Constraints
- Extended Key Usage
- Subject Key Identifier
- Authority Key Identifier
- Subject Alt. Names
- Certificate Policies
- CRL Dist. Endppoints
- ESCT List
- Certificate Authority Information ACcess

X509v3 Validation

Other validation issues of certificates for specific validation requirements

- **Subject Name** (fields and attributes)
  - Not only abstract UIDs, URIs, URLs, eMail addresses, …
  - Extended with X500 distinguished name attributes and classification categories as well as alternative names
- **Issuer name**
  - Issuer/CA Distinguished names with X500 attributes
- **Certif. policies, policy mappings and key policies**
  - Allowing for specific validation to a given policy
  - Setting constraints for limitation/contention of the damage from faulty or malicious Cas

X509v3

Other validation issues of certificates for specific validation requirements

- Inclusion of KeyIDs for Subject and Authority,as Key Selectors
- Information on CRL distribution points or for OnLine Status verification points (OCSP) from CA issuers
- Gradual adoption of OID standardization

- Fingerprints with Dual Secure Hashing Functions for Integrity:
  - Current use of SHA-256 and SHA-1

Extended validation (EV) Certificates

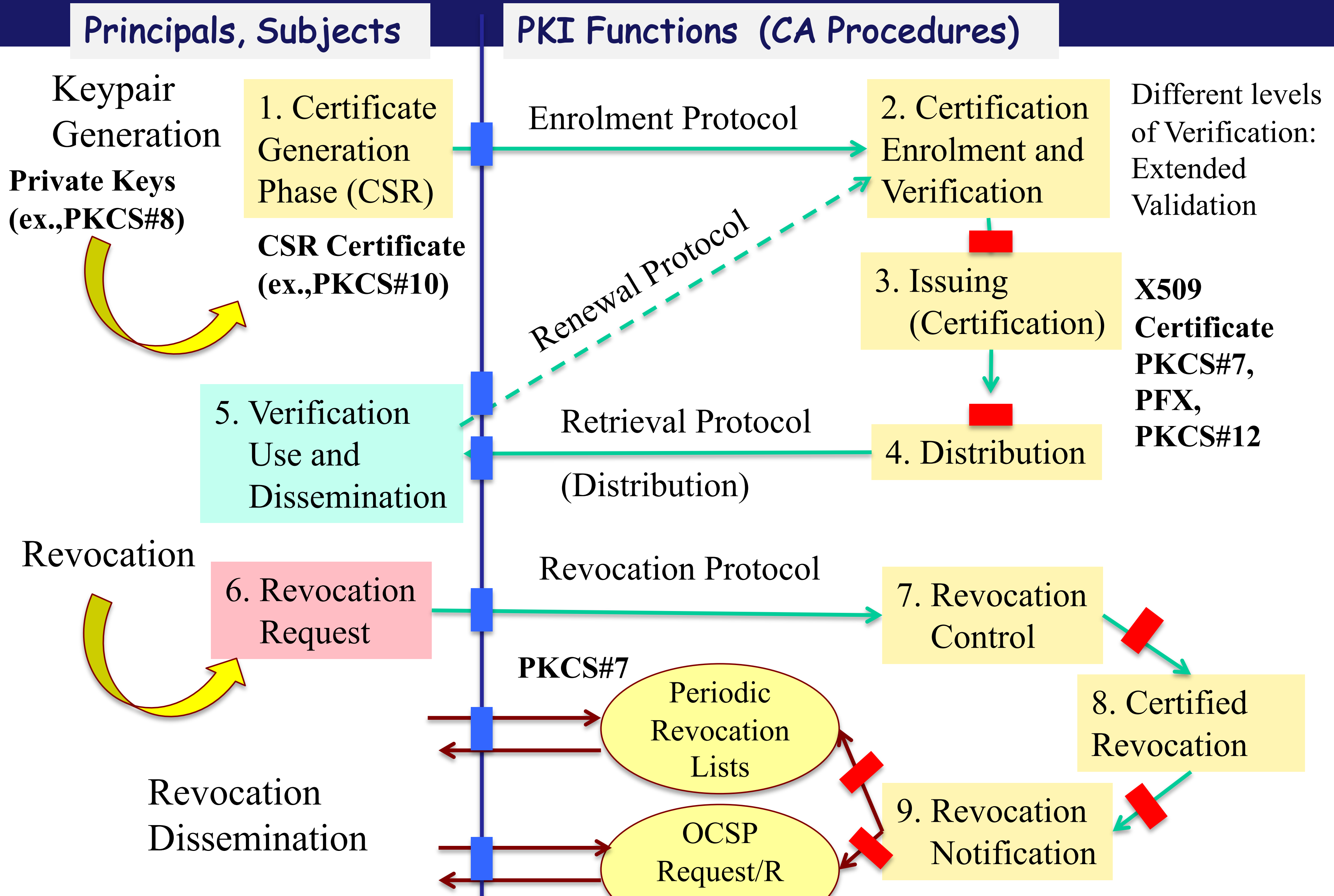
Relevance of
The CA/RA
Delegation Models

- Introduced by the CA/Browser forum
  - http://www.cabforum.org/, http://en.wikipedia.org/wiki/Extended_Validation_Certificate
  - CAs + Relying Party Application Software Suppliers
- Objective: inclusion of standardized procedures for verifying and expressing awareness of the certificate holder and validity (initially motivated by SSL - TLS certificates)
- Additional layer of protection: promotion of good practice, guidelines, accurate verification processes for issuing X509v3 SSL certificates
  - **Verifying the legal, physical and operational existence of the entity**
  - **Verifying that the identity of the entity matches official records**
  - **Verifying that the entity has exclusive right to use the domain specified in the EV Certificate**
  - **Verifying that the entity has properly authorized the issuance of the EV Certificate**

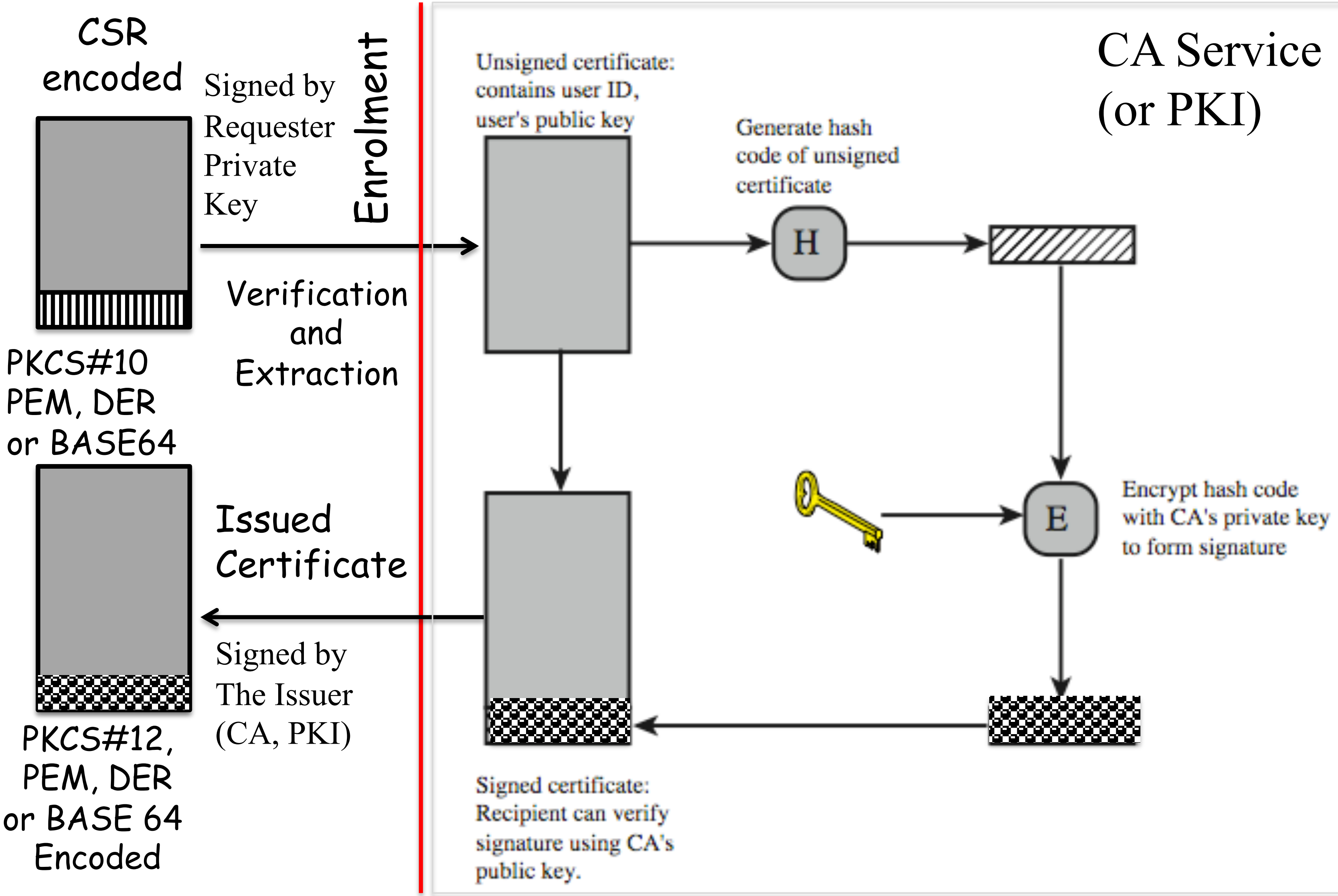
Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- **PKI - Public Key Infrastructure**
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Remember the X509 Life Cycle Management



CSR Validation, extraction and signing (Issuing of Certificates)



Obtaining a User's Certificate

- Certificates: issued by CAs (Functions on PKIs)
  - Any user with access to the public key of the CA can recover and validate the certified user public key
  - Users can exchange certificates and certification chains for verification
    - Can use direct or reverse chains for verification

  - Certificates are public and unforgeable (signed by the issuer CA).
    - Possible to send/distribute/disseminate them in protocols or placed in public directories or repositories
    - Note: having a certificate is not a proof of authentication
      - Need a digital signature, exhibiting the public key certificate to validate the signature

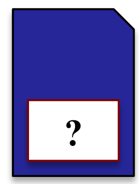
Typical life cycle management

Principals (Sujects):

Generate Keypairs (RSA, DSA)

Generation of Self-Signed Pub-Key Certificate

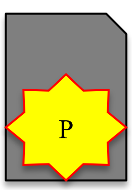
Only usable by principals accepting it (in their trusted cert stores)



Generation of CSR Certificate

Secure storage & management Of Keypair and Private Key !

Enrolment Process for certification



Receives their Issued X509v3 certificates

X509v3 issued certificate

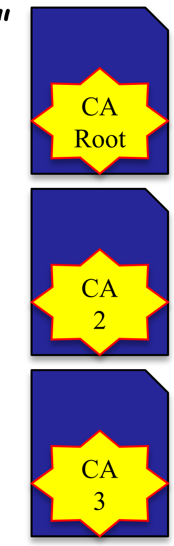
Ready for use when presented in their certification chain

Certification Authority (or PKI solution)

Has a "well-known" disseminated Root Public key Certificate

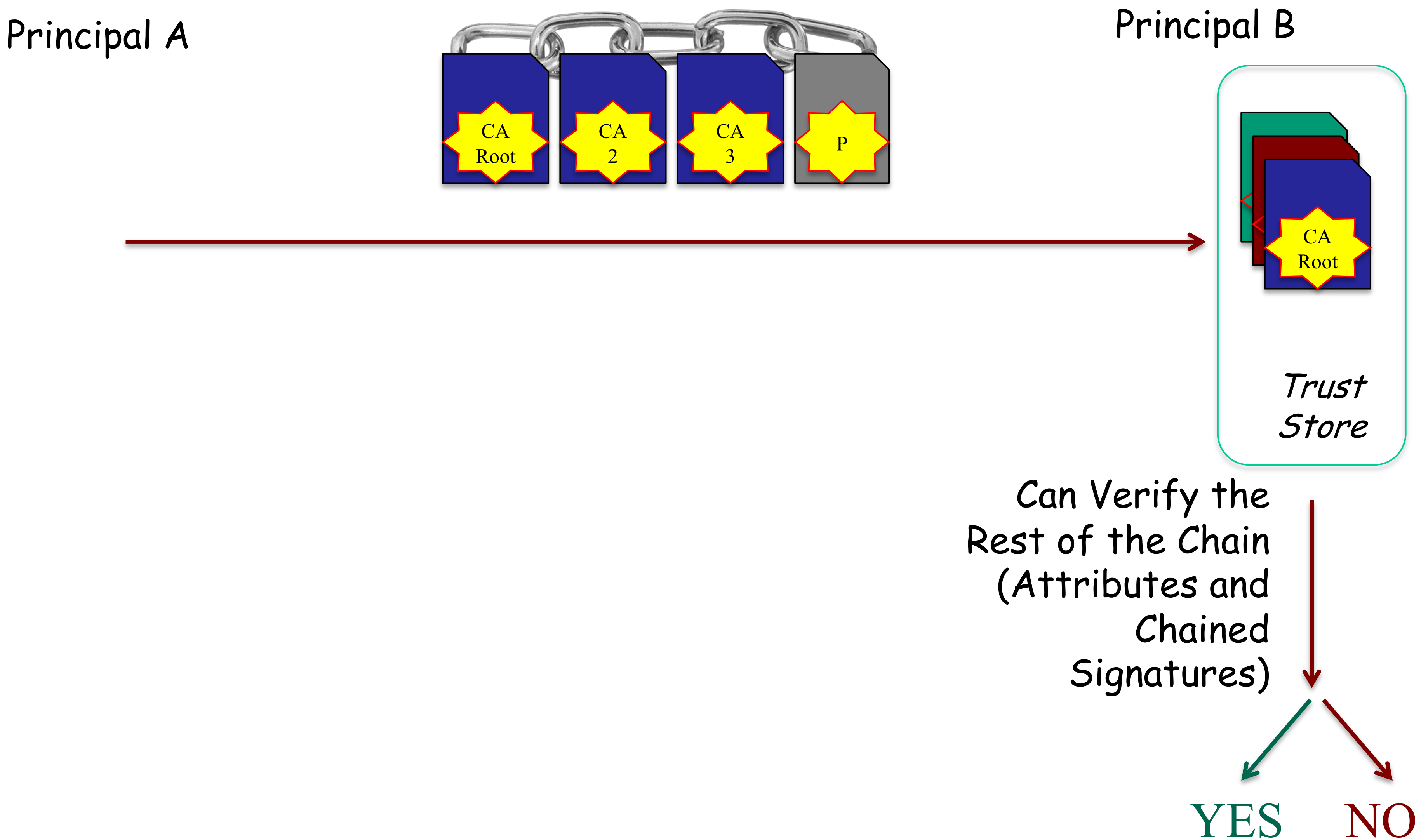
Has Issued Intermediate CA certificates (in a chain)

Validation of enrolment and CSR Certificates



Issues generated certified (signed) X509v3 certificates (in a certain chain)

Certification Chains

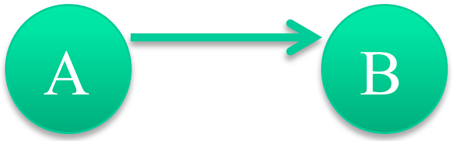


Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- PKI - Public Key Infrastructure
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Summary of Base Authentication Procedures

One-way authentication and Key dist.

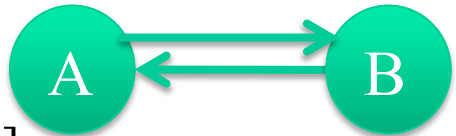


A[{ta, ra, IdB}Kab, Sig_{KprivA} (signData), $\{Kab\}_{KpubB}$]

Two-way (mutual) authentication and Key dist.

A [{ta, ra, IdB}Kab, Sig_{KprivA} (signData), $\{Kab\}_{KpubB}$]

B [{tb, rb, IdA}Kba, Sig_{KprivB} (signData), {Kba}KpubA]

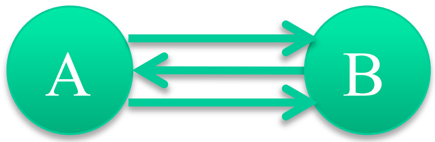


Three-way (Mutual) authentication and Key Dist.

A[{ta, ra, IdB}Kab, Sig_{KprivA} (signData), {Kab}KpubB]

B[{tb, rb, IdA}Kba, Sig_{KprivB} (signData), {Kba}KpubA]

A{rb}



One-Way Authentication

- 1st message (A->B) used to establish:
  - The authenticated identity of A and that message is from A
  - That the message was intended for B
  - Integrity & originality of message
- Message must include timestamp, *nonce*, B's identity and is signed by A
- May include additional info for B
  - Eg., session key, for implicit key-establishment (session key-envelope)
    - Allows the concatenation of additional confidential content or messaging

Two-Way Authentication

- 2 messages (A->B, B->A) establishes in addition:
  - The identity of B and that reply is from B
  - That reply is intended for A
  - Integrity & originality of reply
- Reply includes original nonce from A, also timestamp and a *nonce* from B
- May include additional info for A
  - May establish "half-duplex" session symmetric keys
  - May establish "full-duplex" session symmetric keys (generated from pre-master keys or exchanged seed-material)

Three-Way Authentication

- 3 messages (A->B, B->A, A->B), adding a final round to mutual authentication
  - Enables above authentication **without no need of synchronized clocks**

- Has reply from A back to B containing signed copy of nonce iterated from B
  - Means that timestamps need not be checked or relied upon, preserving anyway message-freshness and ordering (protocol termination) control

Authentication Procedures
Example of concretizations

Autenticação one-way model:

Ex., One-Way TLS Authentication, S/MIME or PGP Message Authentication

Autenticação two-way (mutual)

Ex., Two-Way TLS Authentication, SET Protocol

Autenticação three-way (mutual)

Ex., Two-Way TLS Authentication and Key-Session Generation and Agreement

Practical protocols

Two forms of management of chain trust

Certificates pre-cached (and managed orthogonally) in trusted certificate stores

Ex., JAVA, keystores

> Advantages ? Drawbacks ?

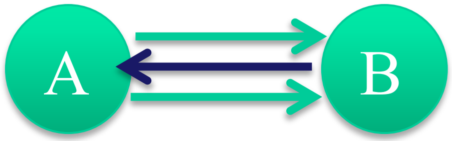
"On the Fly" validation of trust chains

- Only need "root" certificate pre-cached in trusted stores
- Send certification chains in the authentication handshake

> Advantages ? Drawbacks ?

Base Authentication variants (Variant 1)

One-way authentication and Key dist.



A: I am A

B: Authentication challenge **Cb** for the claimer

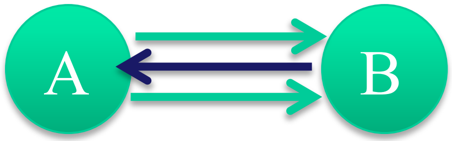
A[{ta, ra, **Cbr**, IdB}Kab, Sig_{KprivA}(signData), $\{Kab\}_{KpubB}$]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

Base Authentication variants (Ex., Variants 1)

One-way authentication and Key dist.



A: I am A, <my ciphersuite proposal>

B: Challenge **Cb**, <my ciphersuite choice>

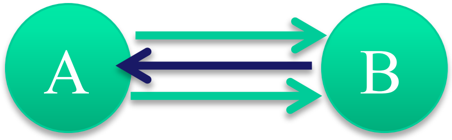
A [{ta, ra, **Cbr**, IdB}Kab, Sig_{KprivA}(signData), {Kab}KpubB]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

Base Authentication Variants (Ex., Variants 2)

One-way authentication and Key dist.



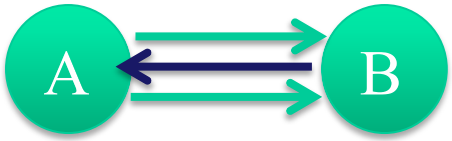
A: I am A, <my ciphersuite proposal>, $CERT_A$
B: Challenge **Cb**, <my ciphersuite choice>, $CERT_B$
A[{ta, ra, **Cbr**, IdB}Kab, Sig_{KprivA}(signData), {Kab}KpubB]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

Base Authentication Procedures (Ex., Variants 3)

One-way authentication and Key dist.



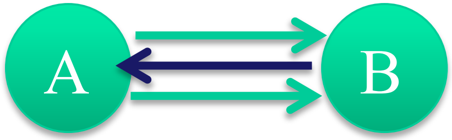
A: I am A, <my ciphersuite proposal>, <Certification Chain>
B[**Cb** Challenge, <my ciphersuite choice>, <Certification Chain>]
A[{ta, ra, **Cbr**, IdB}Kab, Sig_{KprivA}(signData), {Kab}KpubB]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

Base Authentication Procedures (Ex., Variants 4)

One-way authentication and Key dist.



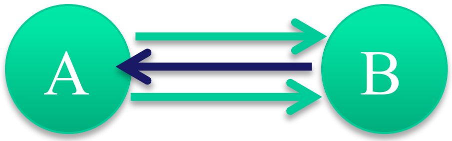
A: I am A, <my ciphersuite proposal>, <Certification Chain>
B: **Cb** Challenge, <my ciphersuite choice>, Sig_{KprivB}(signData), <Cert Chain>
A[{ta, ra, **Cbr**, IdB}Kab, Sig_{KprivA}(signData), $\{Kab\}_{KpubB}$]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

Base Authentication Procedures (Ex., Variants 5)

One-way authentication and Key dist.



A: I am A, <my ciphersuite proposal>, <Certification Chain>

B: **Cb**, <my ciphersuite choice>, Sig_{KprivB}(DHpubB, SignData), <Cert Chain>

A[{ta, ra, **Cbr**, IdB}Ks, Sig_{KprivA}(DHpubA, signData)]

Two-way (mutual) authentication and Key dist.

Three-way (Mutual) authentication and Key Dist.

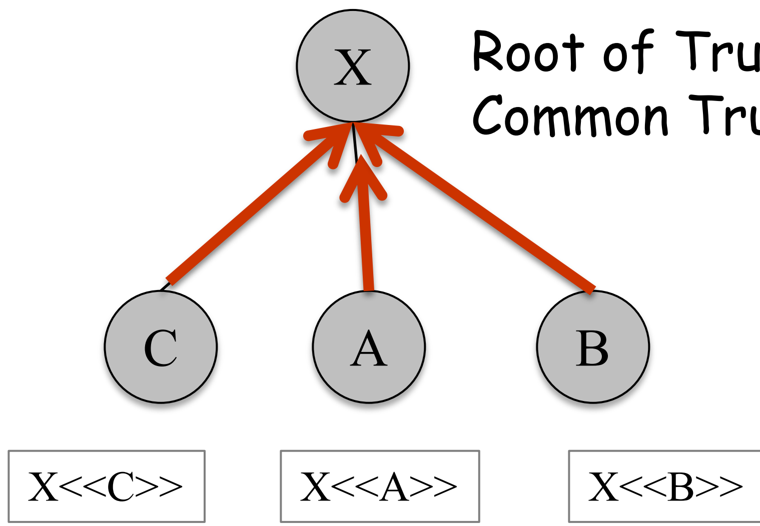
Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- PKI - Public Key Infrastructure
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Trust and Validation Chains

Common trust based validation

- When all users subscribe to the same **Root Of Trust** X
- Ex., Model for a small community of users (non-scalable, centralized-root trust)
- Any user A transmits directly the certificate to any other (B, C)



Root of Trust:
Common Trust (ex., Common CA)

What if we have more than one RooT (or CA)

No common trust verification conditions

- Model for a large community of users (scalable model)
- Users need to have Public Keys of all the CAs ?
- It may be more practical to consider that
  - There will be several Roots of Trust (CAs),
  - But each of which securely provides its public key to some fraction of the users
  - Then we can use cross-certification links in a certification hierarchy

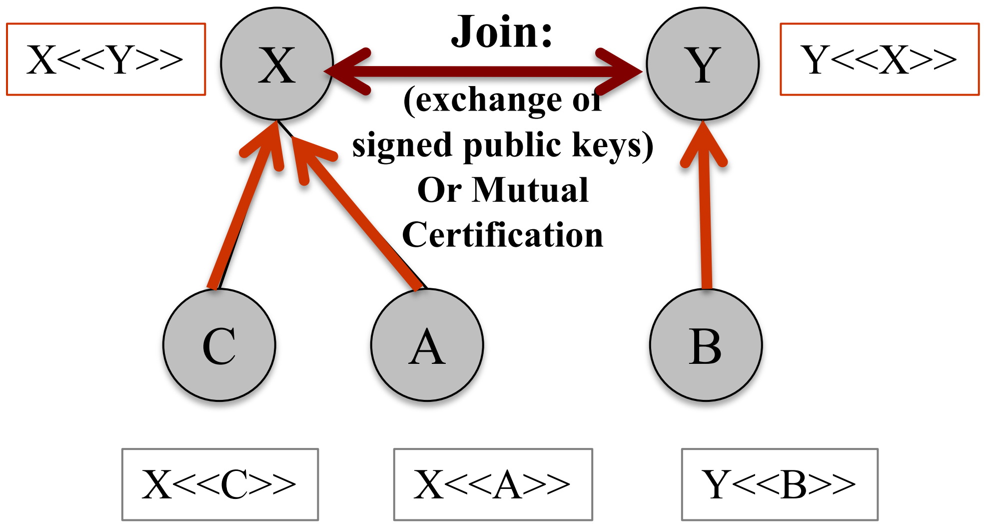
Notation for a Public Key Certificate:

CA <<A>> = {A, V, SN, AI, CA, TA, KpubA}$_{SigCA}$

Y<<X>> means: Certificate of entity **X** issued by **Y**

Verification of certificates => imply that the verifiers previously obtained, in a trusted way, the CA public key

Solution for no Common Trust: Peering



- A obtains X<<Y>> from a directory
- A obtains Y<<B>> from a directory (or directly from B)
- A uses the chain Y <<B>>, X<<Y>>
  B can use the chain: X<<A>> Y<<X>>

  or reverse chain X<<A>> X<<Y>

- Possible generalization for long paths (when joins are at higher levels)

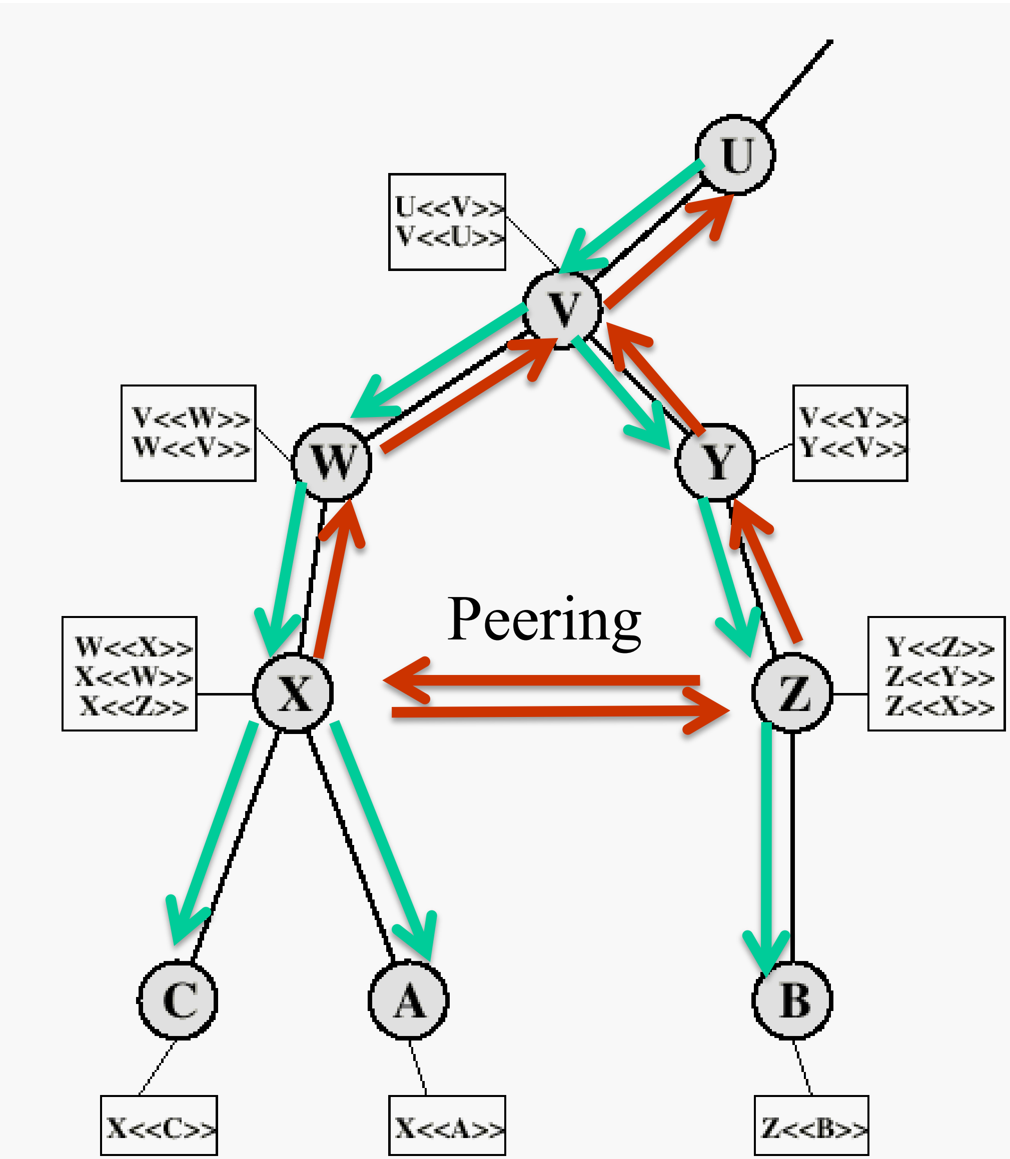
X.509 CA Hierarchy and Chains

- Forward certificates

Forward Chain Validation

- Reverse certificates

Reverse Chain Validation



See a X509v3 Direct Certification Chain in a TLS (HTTPS) connection

- In general the more common is to have Root CA Public Key certificates in local trusted stores
  - the authentication processing supported with a direct certification chain validation

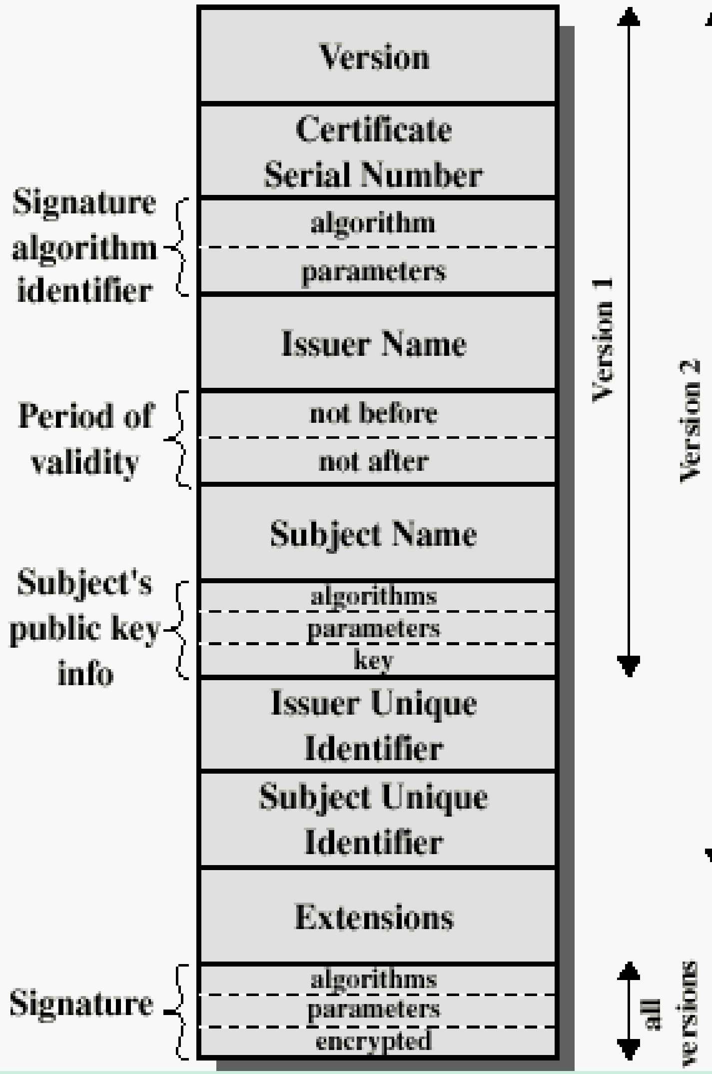
- Ex., see the CA's Root Certificates in your Java installation
  - Find **cacerts** in your /……/**jre/lib/security hierarchy**

- See the certification chain in a TLS (HTTPS) connection:
  - Can use your Browser
  - Or can use openssl
    - `openssl s_client -connect www.feistyduck.com:443`

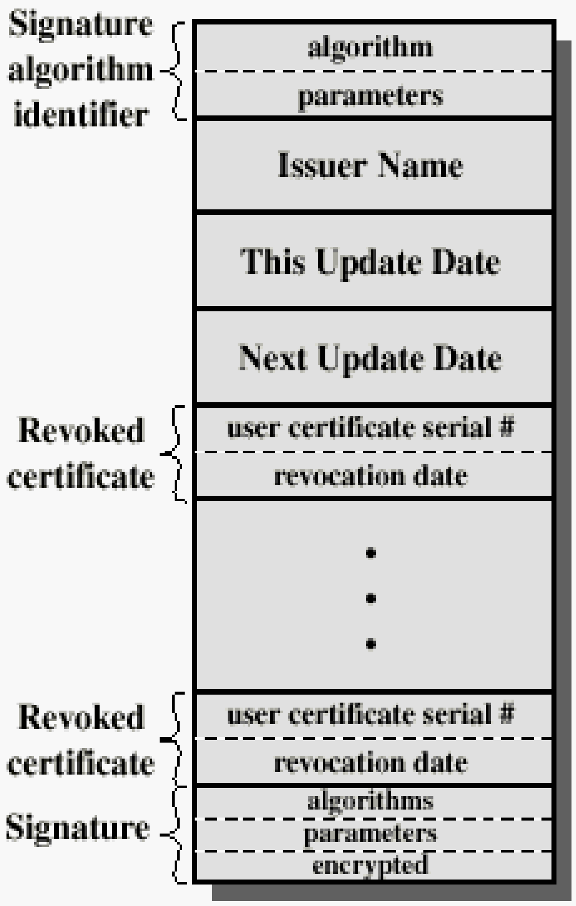
Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- PKI - Public Key Infrastructure
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

X509 Certificates and CRLs



X509 certificate (fields in different versions)



X509 Certificate Revocation List

Revocation of Certificates: Why, When, How

- **Reasons for revocation:**
  - User's private key is assumed to be compromised.
  - User is no longer certified by this CA.
  - CA's certificate is assumed to be compromised.
  - CA's private keys compromised
- **Certificates should not be validated**
  - After the expiration
    - Requires the issuing of a new certificate just before the expiration of the old one
    - The new certificate can be issued by a different CA
  - If the end use is not according with the content (specific attributes, policies, extensions)
  - If it is in a "current" certification revocation list (CRL) issued by the CA that issued the certificate
  - If not validated by synchronous "on line" verification process
    - Via OCSP Protocol

Management of CRLs

- Maintained by each CA (or CRL issuers' end-points)
- Usually provided in DER or PEM Formats
  - A list of revoked (not expired) certificates issued by that CA, including
    - End-user certificates
    - Possible reverse certificates
- CRLs must be managed by final users (user responsibility)
  - Checked from a directory, every time a certificate is received
    - CRL endpoints (in issued X509 certificates)
- Checked from a local cache, periodically updated (ex., Incremental, Time-Controlled, Serial Number Controlled)
    - **Black Lists: CRLs**
    - **Full-Lists vs. Incremental Lists**
    - **Time-controlled vs. Version-Controlled**
    - Also possible: White Lists as White CRLs

See a CRL, as usually issued by CAs

- Download the current CRL from the CRL endpoint of a given (issued) certificate
- Inspect the CRL (example w/ keytool and openssl):

  keytool -printcrl –file <obtainedcrl>

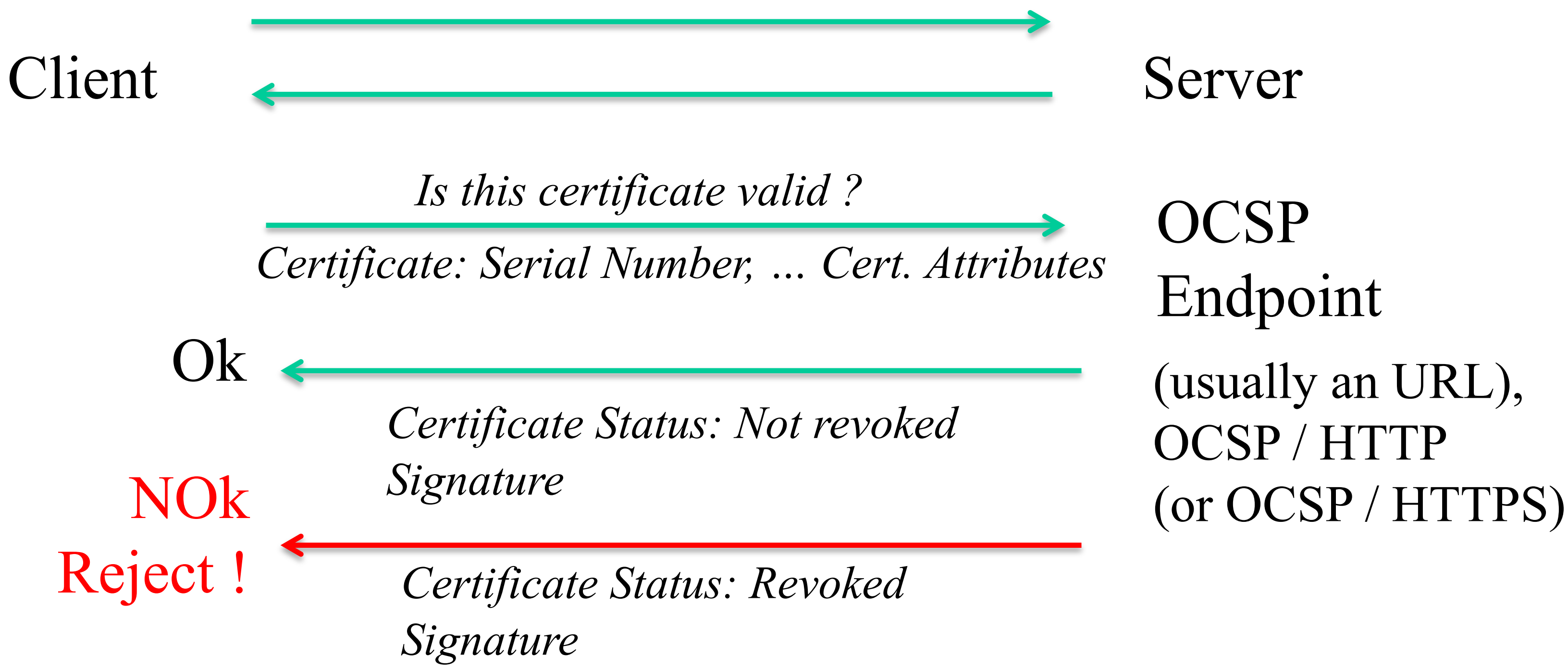
  or

  openssl crl –inform DER –text –noout –in <obtainedcrl>

  To see the CRL in ASN.1 Syntax format specification

Revocation control w/ the OCSP Protocol

- OCSP – On Line Certificate Status Protocol
  - Client/Server Request/Reply Protocol
  - OCSP Endpoints provided by CAs
    - OCSP Endpoint Attribute in issued X509 Certificates



OCSP (example with openssl)

- Given a certificate (ex.): certificte.pem as a chained certificate
- Verify the OCSP endpoint attribute (typically a given URL)
- Verification of all certificates in the chain
- Use of openssl:

Issuer in the chain

openssl ocsp -issuer certificate.pem -cert sslcert.pem -url <http://OCSP-URL> -text -CAfile CAchainfile.pem

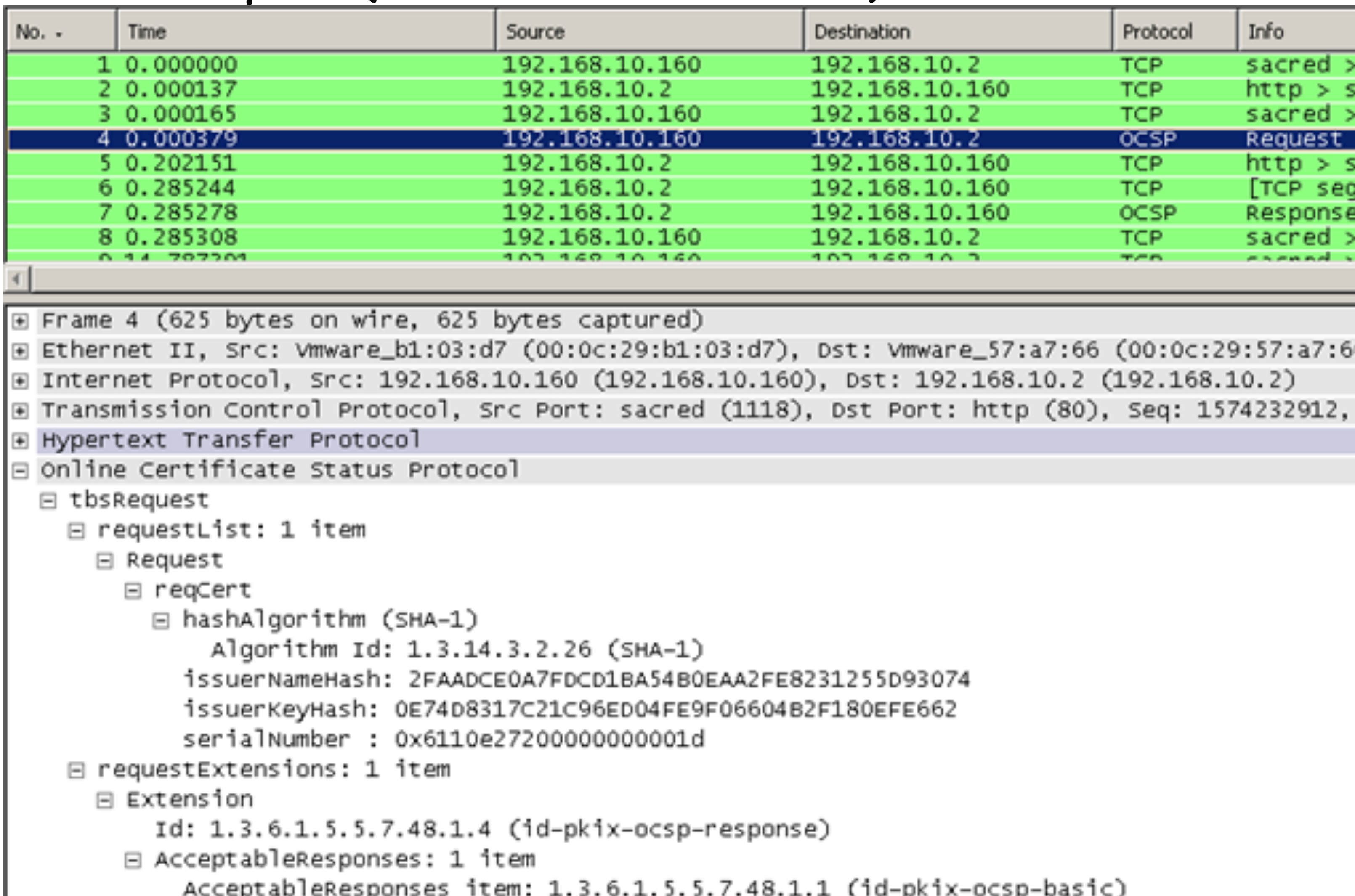
Result …

```
WARNING: no nonce in response
Response verify OK
sslcert.pem: good
	This Update: Mar 13 17:13:19 2012 GMT
	Next Update: Mar 20 17:13:19 2012 GMT
```

```
WARNING: no nonce in response
Response verify OK
sslcert.pem: revoked
	This Update: Mar 16 16:18:11 2012 GMT
	Next Update: Jun 11 00:52:47 2012 GMT
	Reason: keyCompromise
	Revocation Time: Mar 16 16:16:56 2012 GMT
```

OCSP - Online Certificate Status Protocol

- A Request/Response Protocol, usually supported in HTTP
  - **OCSP Request (with the wireshark tool)**



OCSP - Online Certificate Status Protocol

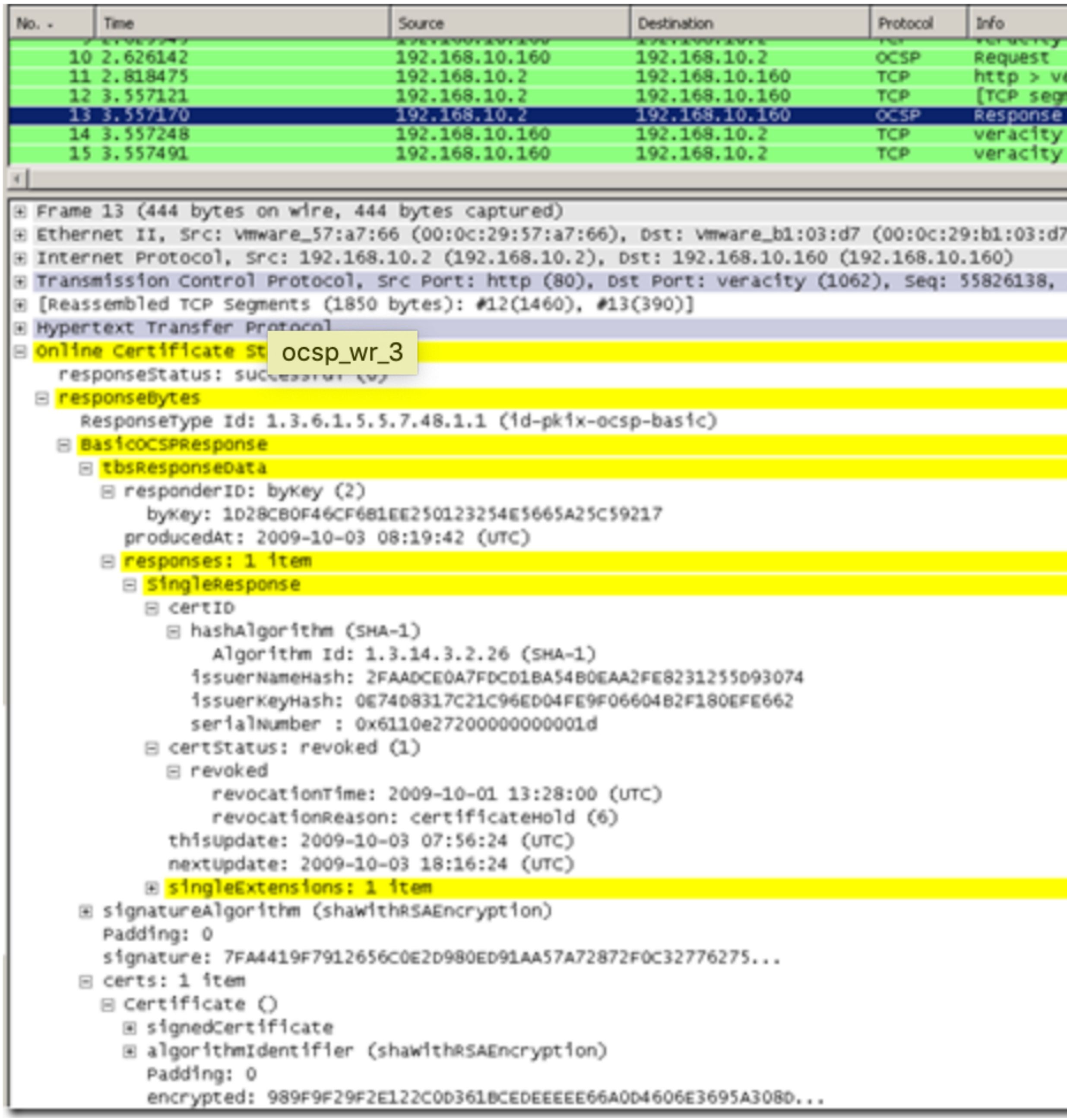
- **OCSP Response (with the wireshark tool)**

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.10.160	192.168.10.2	TCP	sacred >
2	0.000137	192.168.10.2	192.168.10.160	TCP	http > sa
3	0.000165	192.168.10.160	192.168.10.2	TCP	sacred >
4	0.000379	192.168.10.160	192.168.10.2	OCSP	Request
5	0.202151	192.168.10.2	192.168.10.160	TCP	http > sa
6	0.285244	192.168.10.2	192.168.10.160	TCP	[TCP segm
7	0.285278	192.168.10.2	192.168.10.160	OCSP	Response
8	0.285308	192.168.10.160	192.168.10.2	TCP	sacred >

```
⊞ Frame 7 (367 bytes on wire, 367 bytes captured)
⊞ Ethernet II, Src: Vmware_57:a7:66 (00:0c:29:57:a7:66), Dst: Vmware_b1:03:d7 (00:0c:29:b1:03:d7
⊞ Internet Protocol, Src: 192.168.10.2 (192.168.10.2), Dst: 192.168.10.160 (192.168.10.160)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: sacred (1118), Seq: 2186065053,
⊞ [Reassembled TCP Segments (1773 bytes): #6(1460), #7(313)]
⊞ Hypertext Transfer Protocol
⊟ Online Certificate Status Protocol
    responseStatus: successful (0)
  ⊟ responseBytes
      ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkix-ocsp-basic)
    ⊟ BasicOCSPResponse
      ⊞ tbsResponseData
      ⊞ signatureAlgorithm (shaWithRSAEncryption)
        Padding: 0
        signature: 0E5230CC19E6370E39F1F3FA90A797E100D1DC7B5201F82B...
      ⊞ certs: 1 item
```

OCSP - Online Certificate Status Protocol

- **OCSP Response**



Outline

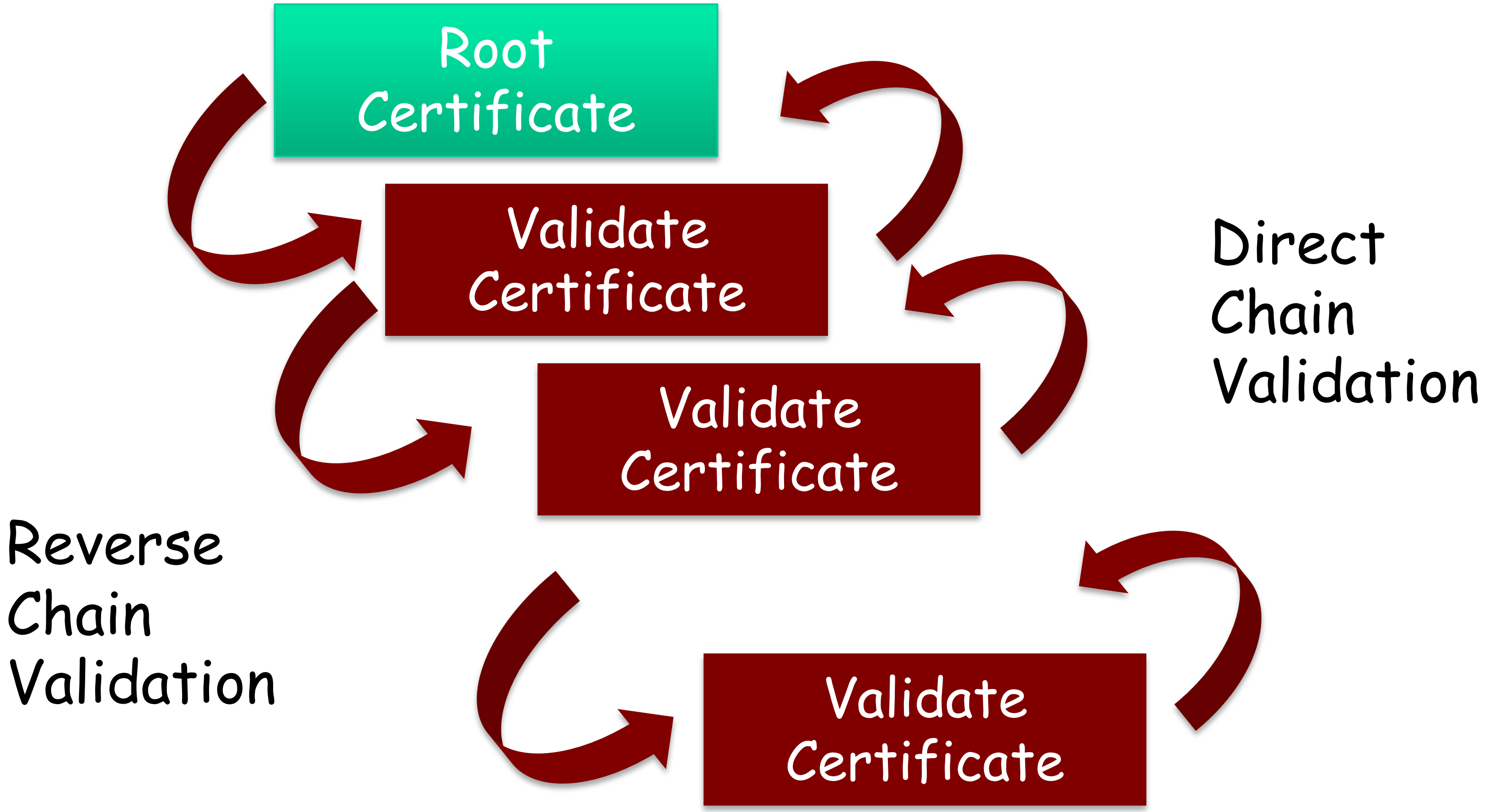
- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- PKI - Public Key Infrastructure
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Validation can be complex, in a long tail

Validate a Certificate

- Validation of different attributes
  - Subject Name Attributes:
    - Names, DNS names
  - Issuer Name Attributes
    - O, OU, Cname, … Validity
- Validation of critical fields and attributes
  - Keysizes, Key usage, …
  - Extensions: critical attributes and other possible required attributes
    - key usage policy
    - Verification of selected extensions
    - Timestamping
    - CRL endpoints => Look to the more recent issued CRL
    - OCSP endpoints => Possibly validate on the OCSP endpoint
    - …
    - Integrity Fingerprints
- Basic constraints
  - Certificate authority
- Validation of signatures

Chain Validation can be more complex yet in a more long tail (direct and/or reverse)



Programming support: ex., JAVA PKI API
http://docs.oracle.com/javase/8/docs/technotes/guides/security/certpath/CertPathProgGuide.html

Complexity management issues (and usually flaws)

- Architectural weaknesses
- Errors and issues involving certificate authorities and/or management of PKIs
  - Ex., Verification problems in enrolment processes
- Implementation issues
- Cryptographic weaknesses

SW Certificates/Certification/Validation weaknesses

- Incorrect verification
- Incomplete verification or limited chain levels
- Implementation Bugs

X509v3, Chains, CRL and OCSP in practice

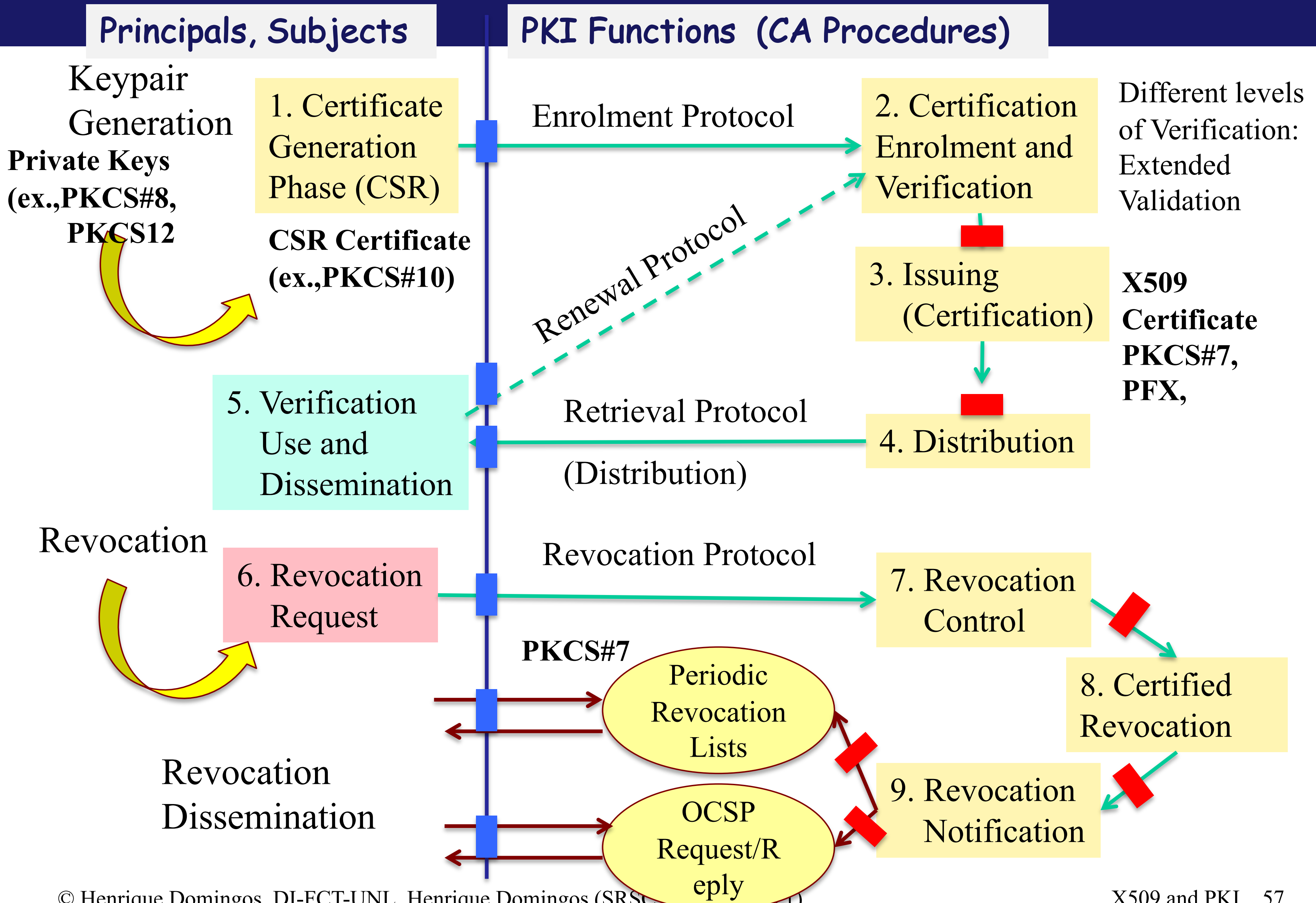
- See in labs the exerices and demos (Lab 7)
  - Use of openssl or keytool (java) tools
  - Generation of Certificates in the Java programming environment
    - Kava Keystores to manage private vs. public keys
    - Generation of public key certificates exported from keystores of keypairs
    - Trusted Certificate Stores: keystores containing imported certificates
  - Use of tools for the verification of certificates
    - CRLs
    - Use of OCSP protocol handlers

  - Management of standard formats for certificates
  - Manipulation of certificates in Java programs

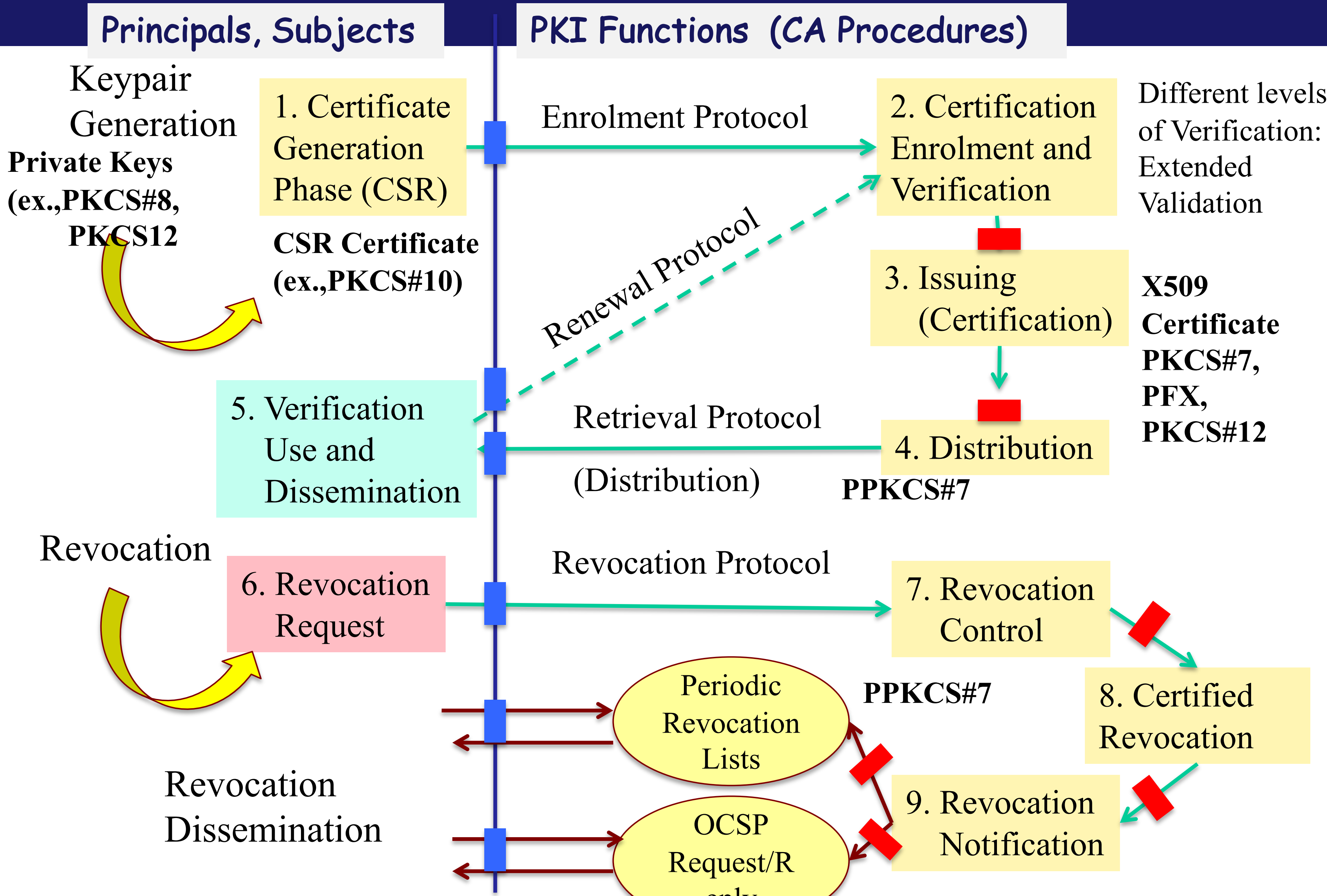
Outline

- **X509 Authentication**
  - X509 Authentication and Key Management Issues
- **X509 Certificates**
  - X509 and X509 v3 Certificates
  - X509 v3 Extensions
  - Life-Cycle Management of X509 Certificates
  - Authentication procedures
  - Forward and reverse certification chains
  - Revocation
  - The possible long tail of certification chains
- **PKI - Public Key Infrastructure**
  - PKI Standardization and PKIX Management
- Complementary: Key Management Issues

Remember the X509 Life Cycle Management



Remember the X509 Life Cycle Management



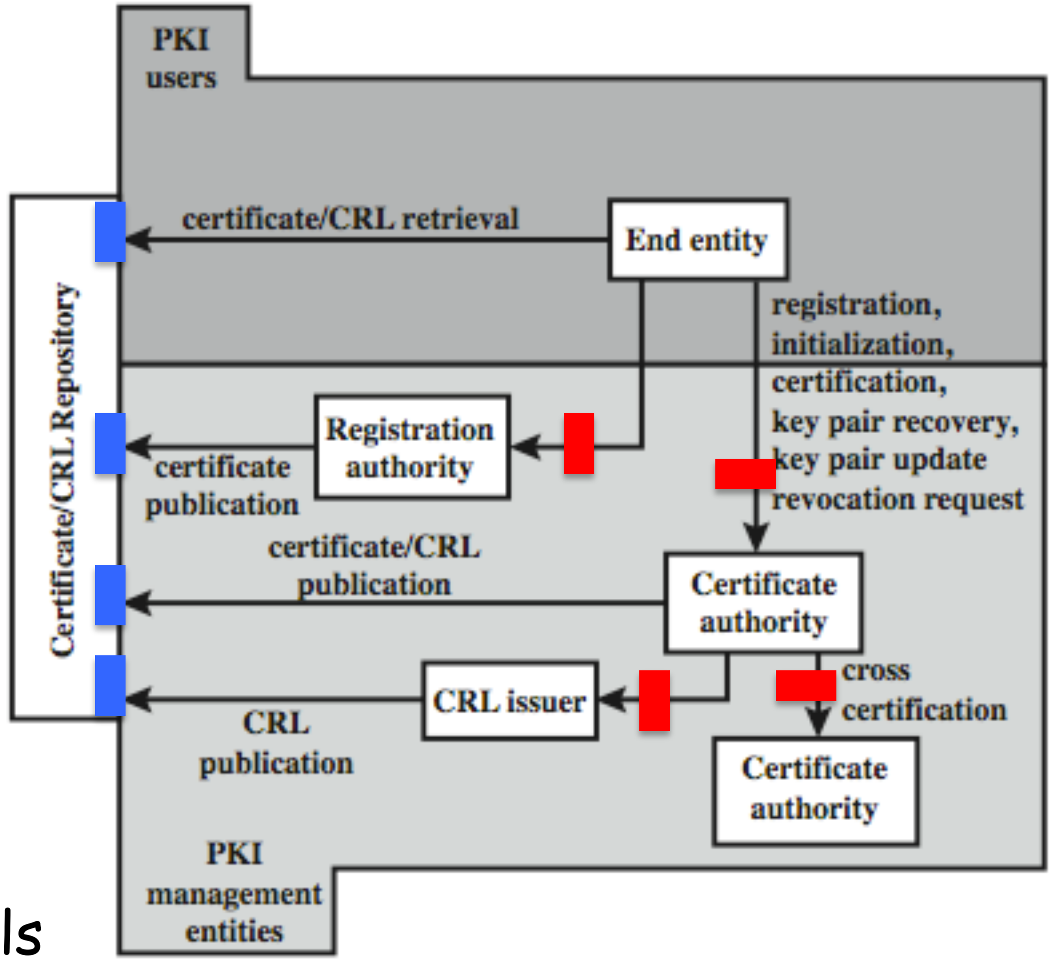
PKI - Public Key Infrastructure

- A Standard Framework Model
  - a set of: HW, SW, People, Rules, Procedures, Policies and Protocols, needed to create, manage, store, distribute and revoke digital certificates
- Objective: enable secure, convenient and efficient acquisition of public keys, promoting strict and well-known specifications
- Coordination by the IETF X509 (PKIX) WG
- Standardized base for compatibility purposes on the above issues in building PKI Platforms
  - Solutions that can also be used by CAs (Certification Authorities) and Ras (Registration Authorities or CA Registrars)

PKIX Architectural model and framework

Key Elements

- Management Functions (APIs):
  - Registration
  - Initialization
  - Certification
  - Key-Recovering
  - Key-Update
  - Revocation Request
  - Cross Certification
- Management Protocols



PKIX Management Functions

- **Registration**
  - Enrollments from users to CAs (directly or through RAs)
  - Offline and Online procedures for mutual authentication
- **Initialization**
  - Initialization and installation of trusted CA certificates
- **Certification**
  - Registration of CSRs (PKCS#10) to obtain CA issued Certificates in standard formats (ex., PKCS#12, PKCS#7)
- **Key Pair Recovery**
  - Restoring encryption/decryption keys
- **Key Pair Update**
  - Regular updates and issuing of new certificates
- **Revocation request**
  - Regular updates and issuing of new certificates
- **Cross certification**
  - Exchanged signed CA public keys, between CAs

Usually, Interoperability Using PEM and DER representations

Scale and more extensible trust model

- Different entities involved, acting with different roles in a distributed way: **CAs, RAs, CRL Issuers, CRs**
  - Difference between:
    - **CA**: Certification authorities (Cert. ISSUING)
      - Different level CAs: aggregated in a direct certification chain
        - » Root CA, Level 2 CA, Level 3 CA, etc
        - » Model practically used in "well-known CA companies" or "CA delegation companies"
    - **R**: Registration authorities (REGISTRATION, ENROLLMENT DELEGATION)
    - **CRL Issuers**: (Issuers of CRLs)
    - **CRs or Certification Repositories** (DISTRIBUTION, for on demand REQUEST-REPLY

PKIX Management Protocols

- Standard protocols between PKIX entities supporting PKIX management functions

  Ex:

  - **OCSP**: X509 Internet Public Key Infrastructure – Online certification status protocol (OCSP) RFC 6960
    - Update for previous RFC 5912, Obsoletes: RFCs 2560, 6277
  - **CMP** - Certificate Management Protocol: RFC 4210 (2015)
  - **CMC** – Certificate Management Messages over CMS:
    - RFC 5272 > updated by recent RFC 6402 proposal
  - **CMS** – Cryptographic Message Syntax: RFC 5652 (obs. 3852)

See the standardization process from the X509 PKIX IETF WG, …
http://datatracker.ietf.org/wg/pkix/

Formats

Certificates has been encoded and/or digitally signed in different formats (defined in RFC 5280 - PKIX) .

See also, for ex: https://en.wikipedia.org/wiki/X.509

Encodings:

- PKCS#10 CSR: Certificate Signed Request format
- PKCS#7 format: Certificates and CRLs - Certificate Revocation Lists dissemination

Certificates and interoperable formats:

DER (binary encoding)

PEM (base64 encoding)

More on Formats

- Encoding Conventions vs. file extensions:
- .pem – (Privacy-enhanced Electronic Mail) Base64 encoded DER certificate, enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"
- .cer, .crt, .der – usually in binary DER form, but Base64-encoded certificates are common too (see .pem above)
- .p7b, .p7c – PKCS#7 SignedData structure without data, just certificate(s) or CRL(s)
- .p12 – PKCS#12, may contain certificate(s) (public) and private keys (password protected)
- .pfx – PFX, predecessor of PKCS#12

Conversions / Management of Formats

Conversions available in some existent tools

See: openssl and keytool:-)))

Example w/ openssl:

- openssl x509 -outform der -in certificate.pem -out certificate.der
- openssl crl2pkcs7 -nocrl -certfile certificate.cer -out certificate.p7b -certfile CACert.cer
- openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in certificate.crt -certfile CACert.crt
- openssl x509 -inform der -in certificate.cer -out certificate.pem
- openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
- openssl pkcs7 -print_certs -in certificate.p7b -out certificate.cer
- openssl pkcs12 -export -in certificate.cer -inkey privateKey.key -out certificate.pfx -certfile CACert.cer
- openssl pkcs12 -in certificate.pfx -out certificate.cer –nodes

Conversions / Management of Formats

Management of CRLs in Java and with Java keytool

- Download and verification
- Can use keytool, KeyStoreExplorer or openssl tools
- Programatically (ex., JAVA, CRL Class, X509CRL SubClass)

https://docs.oracle.com/javase/7/docs/api/java/security/cert/CRL.html

Suggested Readings

Suggested Readings:

W. Stallings, Network Security Essentials – Applications and Standards, Chap 4., sections 4.5 – X509 and 4.6 – PKI

Other references on slides