# IT Security

# Chapter 3: Checksums and Digital Signatures

# Part 2

- Practical aspects of digital signatures
- Components of a PKI
- Certificates (X509, XML)
- Signature Act

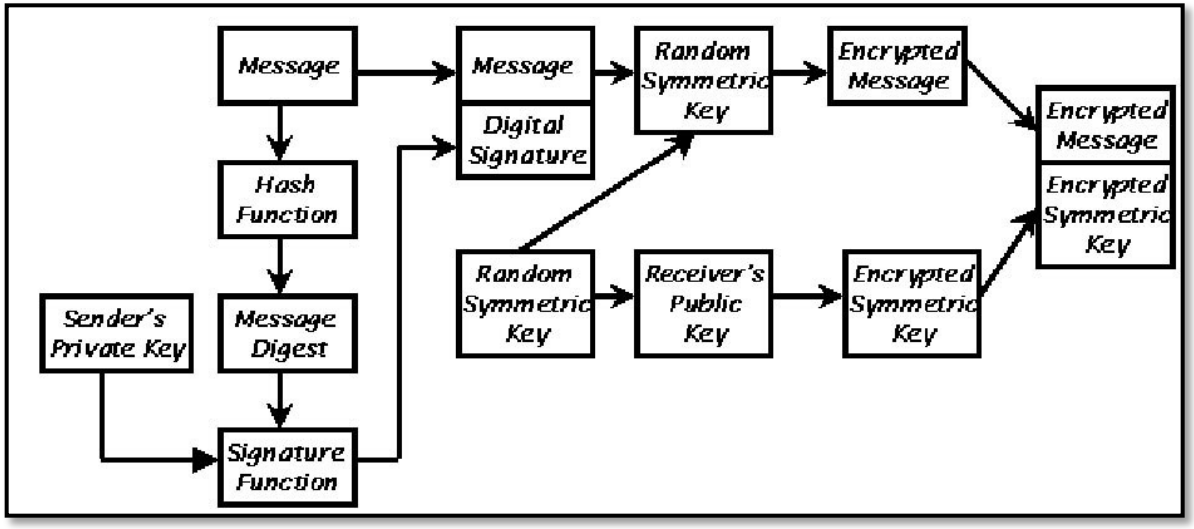
**Prof. Dr. Reiner Hüttl, TH Rosenheim, © 2023, 15.03.23**

# Practical aspects of digital signatures

- Representation problem: With signatures you must see everything you sign
  - **WYSIWYS** (**W**hat **y**ou **s**ee **i**s **w**hat **y**ou **s**ign)
  - There are document formats with content you can't see,
    e.g. macros in Word, JavaScript in web pages.
  - What do I do with such documents?
    - show hidden content
    - eliminate hidden content
    - reformat the document to a format without hidden content
- In combination with encryption
  - first sign then encrypt
  - otherwise, you sign a document that you can not read

# PKCS#7 Signature Standard

- Describes structure of encrypted and signed messages
- Multiple formats: Data, Signed-Data, Enveloped-Data, Signed-and-enveloped-Data.
- Process to create a digital envelope around digitally signed data (Signed-and-enveloped-Data) :

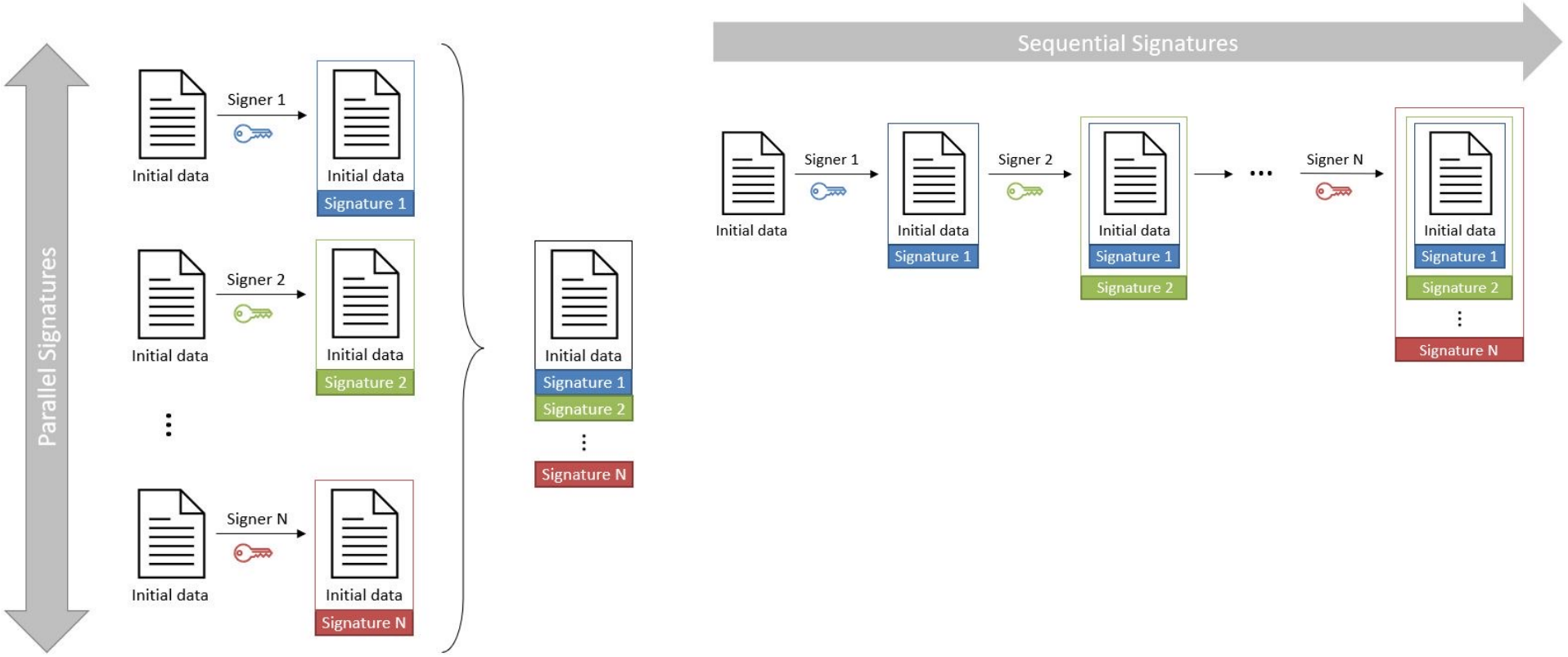



- Further worldwide accepted PKCS published by RSA Laboratories
  - **PKCS#5** (Password-Based Cryptography Standard)
  - **PKCS#10** (Certification Request Syntax Standard)
  - https://en.wikipedia.org/wiki/PKCS

# Example for a PKCS#7 Signature

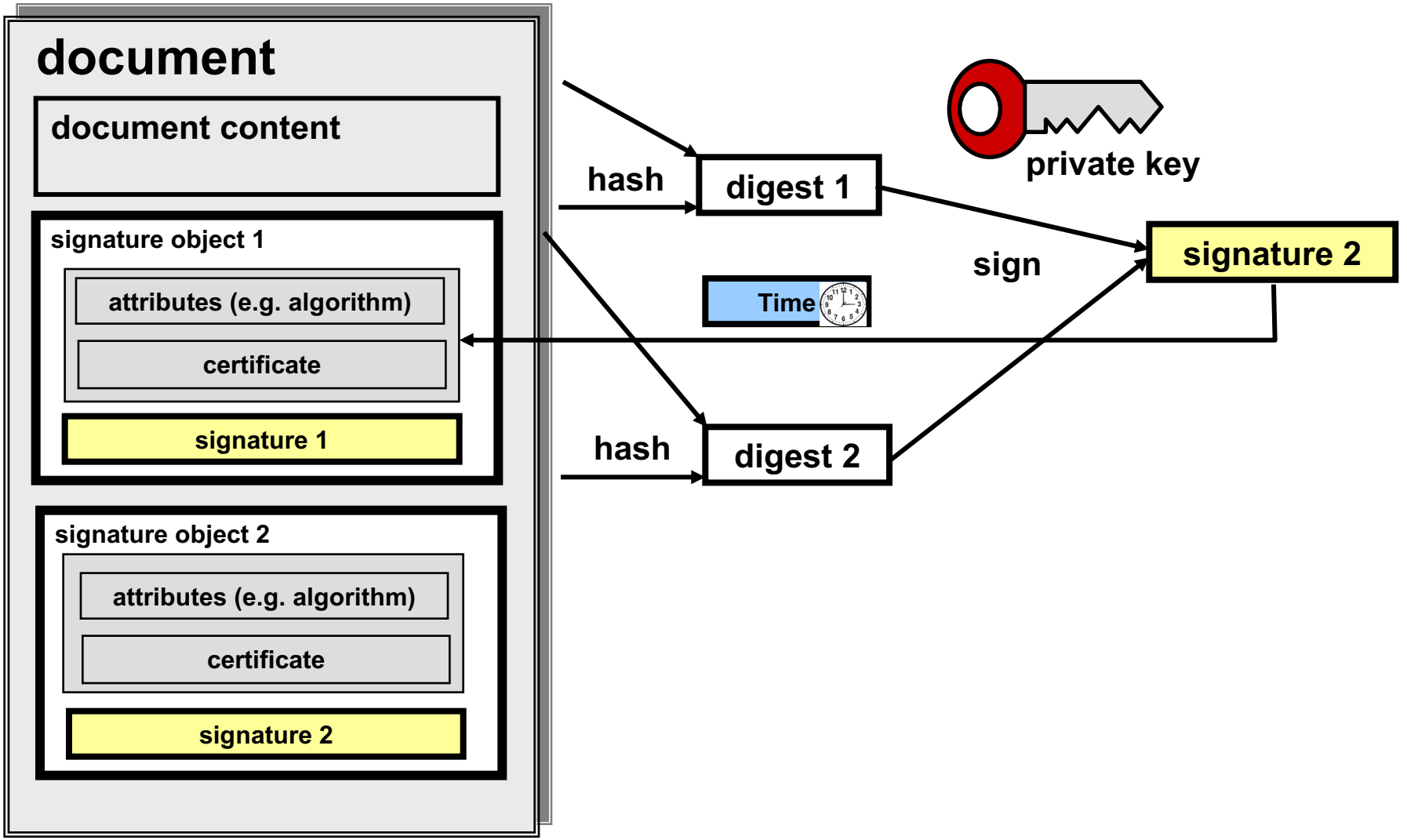
```
SignedData  {
   version           0,
   digestAlgorithms  {
      {1 3 36 3 2 1}, -- OID von RIPEMD-160
      {1 3 14 3 1 18}  -- OID von SHA-1
      },
   encapContentInfo  {
      eContentType  {iso(1) member-body(2) us(840) rsadsi(113549)
                     pkcs(1) pkcs7(7) 1 } -- OID für Data Content
      eContent    [0] "Hello World!"
      },
   signerInfos  {
      {version  1,
       sid      issuerAndSerialNumber {
                   issuer         Alice,
                   serialNumber   3333    -- Zertifikats-Seriennummer
                   },
       digestAlgorithm    {1 3 36 3 2 1}, -- OID von RIPEMD-160
       signatureAlgorithm {1 3 36 3 3 1 2}, -- OID von RSAwithRIPEMD
       signature          'xx..xx' -- RSA-Signatur, 1024 Bit
      },
      {version  2,
       sid      issuerAndSerialNumber {
                   issuer         Alice,
                   serialNumber   4444
                   },
       digestAlgorithm    {1 3 14 3 1 18}  -- OID von SHA-1
       signatureAlgorithm {1 2 840 10045 1}, -- OID von ECDSAwithSHA1
       signature          'yy..yy' -- ECDSA-Signatur, 160 Bit
      }
     }
```

# How do multiple signatures work?

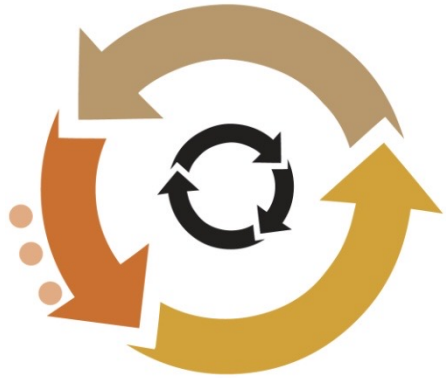



Source: https://dss.nowina.lu/doc/dss-documentation.html#ParallelSignatures

# How does signature renewal work?



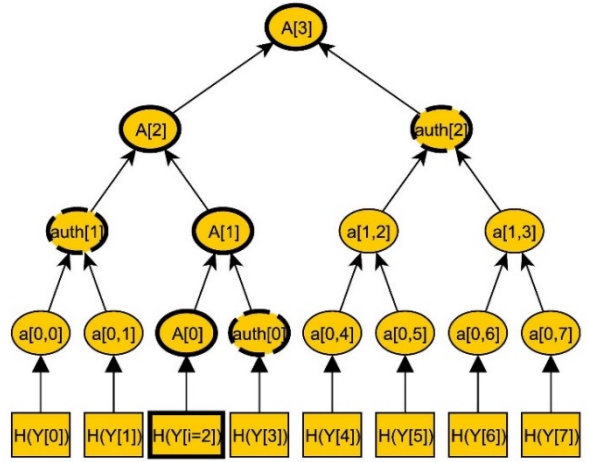
# Signature renewal process



- Cause
  - if certificates are no longer listed in the TrustCenter
  - the procedures in the certificate are insecure (hash, encryption)
  - file formats and signature formats are changing
  - laws require verifiability of the signature for a longer period of time
- Process of re-signing
  - verification of the old signature
  - creation of the new signature
    - New signature must include document data and old signature
    - format change of document and/or signature may be necessary
  - process must take place in secure environment
  - process should be certified to ensure legal certainty

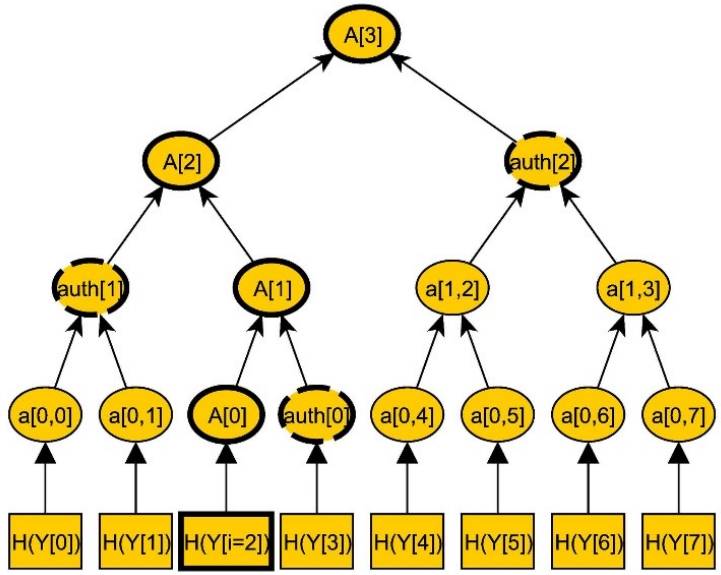
# Merkle signature scheme

Quelle https://en-academic.com/dic.nsf/enwiki/11462918




- Merkle signatures are a signature scheme based on hash trees (Merkle tree).
- The public key is the root of the Merkle tree
- The number of signatures per public key is limited (by the number of leaves, this is a power of 2)
- For the signature, the hash values along the path from a leave to the root are also appended.
- If all leaves are used, a new tree must be taken.
- Merkle signatures are resistant to quantum computing
- Merkle trees are used in blockchains for authentication (e.g. Bitcoin)
- Hash trees for securing integrity are more efficient than hash lists

# Signature and verification with a Merkle signature scheme




Quelle https://en-academic.com/dic.nsf/enwiki/11462918

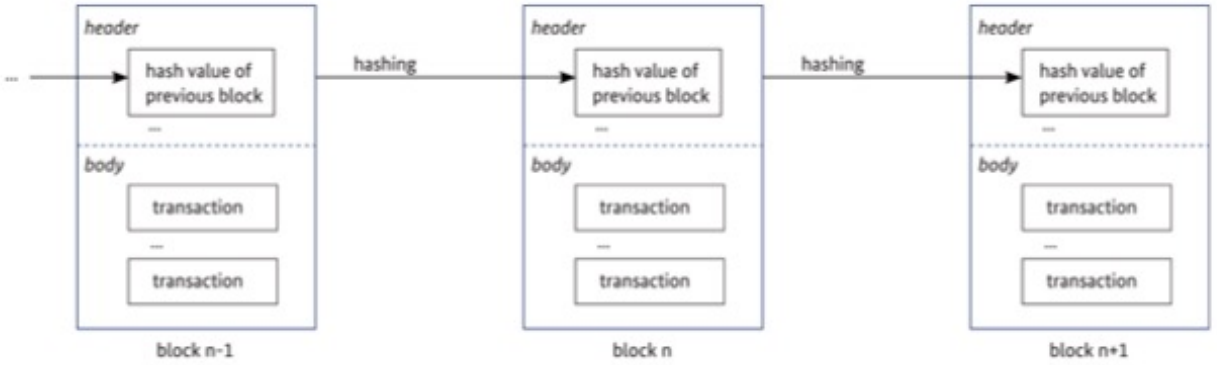
## Signature with a Merkle scheme:

1. Generate n key pairs (Xi, Yi) ,
   Xi is private key, Yi is public key
   in the example is i=8
2. Calculate Merkle tree
3. A[n] is public key of Merkle tree
4. Sign message M with Xi, -> sig‘
5. Calculate path from Yi to the root
   Example for i=2
   A[0] = H(Y2)
   A[1] = H(A[0] || auth[0]) = H(A[0] || H(Y3))
   A[2] = H(A[1] || auth[1]) = H(A[1] || H(a[0,0] || H(a[0,1]) )
   = H(A[1] || H(H(Y0) || H(Y1) )
   A[3] = H(A[2] || auth[2])
6. Signature sig = (sig‘, auth[0], auth[1],auth[2]))

## Verification

1. Verify the signature sig‘ with Y2
2. Calculate A[3] from Y2, auth[0], auth[1],auth[2]
3. Verify if public key of Merkle tree is identical with A[3]

# The blockchain as an example of application of cryptography




Source: Towards Secure Blockchain – Concepts, Requirements, Assessments, BS Bundesamt für Sicherheit in der Informationstechnik, März 2019)
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.html

- The chaining of the blocks is protected against manipulation by **hash values**.
- The algorithm for **consensus** (adding a new block to the chain) is usually based on cryptographic methods.
  - e.g. in Bitcoin the block is only accepted if the **miner** finds a hash value for the block that starts with a given number of zeros. For this purpose, the miner may append any number (nonce) until he has a suitable hash.
- **Signatures** with a public key that is not assigned to an explicit user (**pseudonymization**)

# How are the goals of IT security implemented in the blockchain?

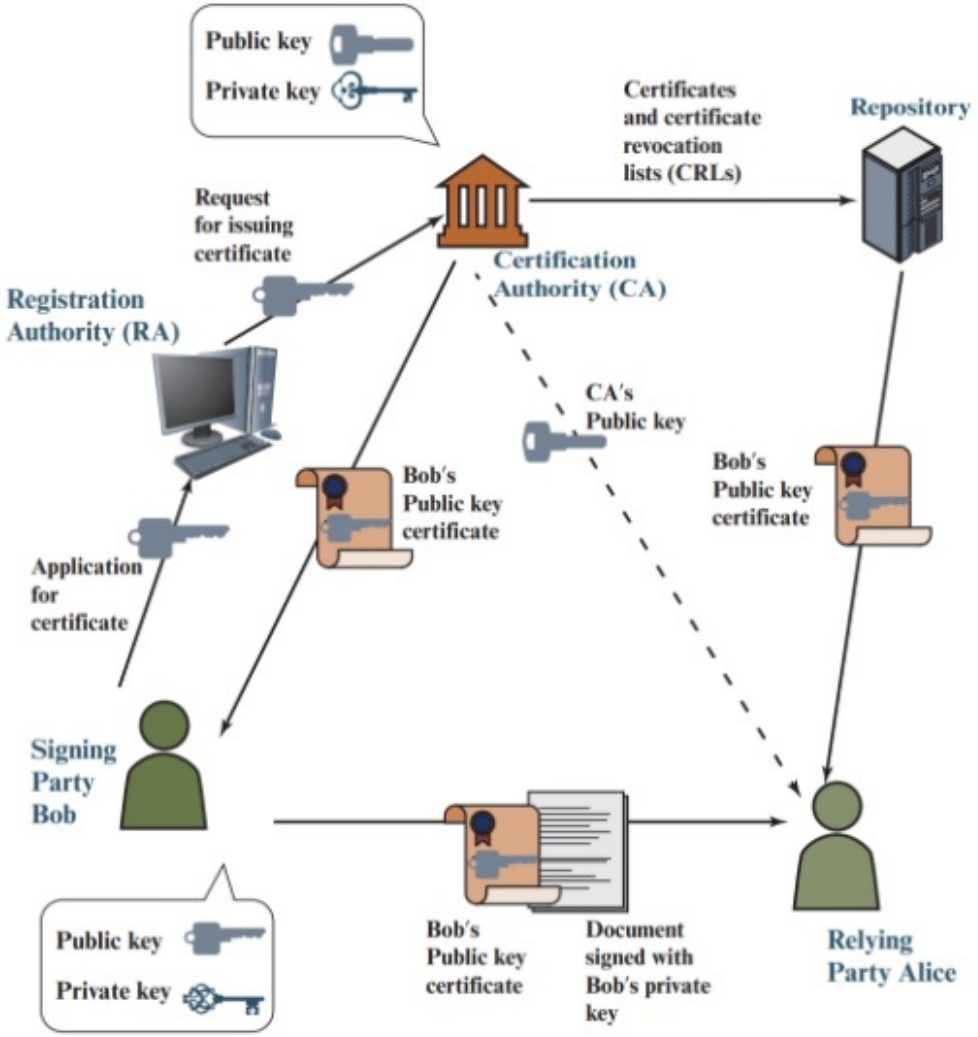
- **Integrity** is based on hash values
- **Availability** is based on decentralization
- **Confidentiality** is difficult to implement and often not desired.
  - store confidential data on external storage
  - use complex procedures (homomorphic encryption, Trusted Execution Environments TEE, secure Multi-Party Computation sMPC)
- **Authenticity** is based on private signature keys
  - for private blockchains, identification of accounts is desired
  - for public blockchains, pseudonymity is desired

Security goals and design goals are sometimes in conflict

Source: Towards Secure Blockchain – Concepts, Requirements, Assessments, BS Bundesamt für Sicherheit in der Informationstechnik, März 2019)
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Secure_Blockchain.html

# Public Key Infrastructure (PKI)

- What is a PKI?
  - It provides a confidential and efficient key and certificate management
  - It is a interface for Trust Services (generation, verification, revocation)




Source: Stallings, William. *Cryptography and Network Securit Principles and Practice, Global Edition*. Available from: VitalSource Bookshelf, (8th Edition). Pearson International Content, 2022.

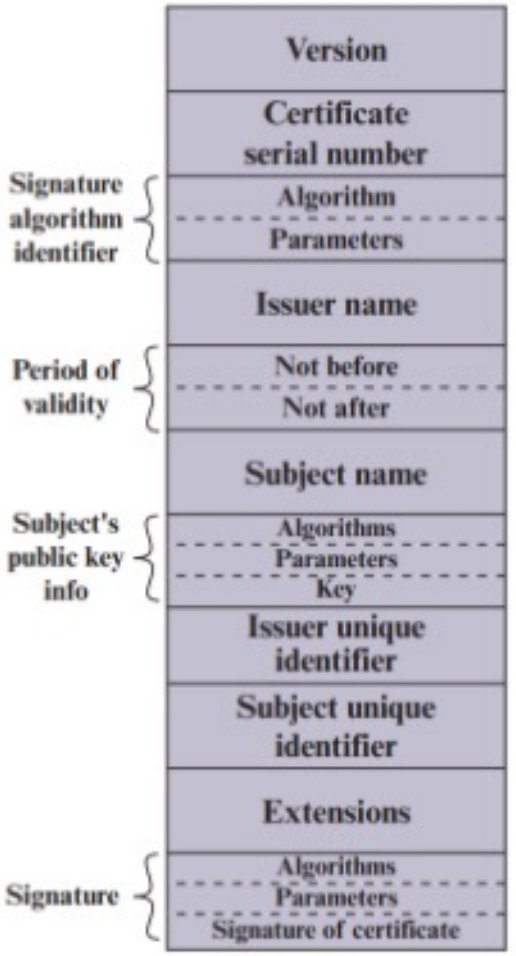
# Components of a PKI

- CA Certification Authority
  - creates certificates
- RA Registration Authority
  - interface between CA and subjects (registration office)
  - handles subject identification
- Directory service / Repository
  - contains list of all issued certificates
  - provides revocation-list
- Client unit
  - contains application (PC, Smart phone, ...)
- Personal Security Environment (PSE)
  - Environment in which key is stored (chip card, hard disk, ...)
- Other optional components
  - Time stamp service TSS
  - Revocation instance (REV)
  - Recovery instance

# Certificates are digital IDs

- **Problem**: Authentic provision of public keys (man in the middle).
- **Solution**: Certification Authorities (Trust Centers) control the identity of the owner and guarantee the authenticity of the keys
- Certificates
  - have limited lifetime
  - can be revoked
  - are used by protocols such as SSL, S/MIME, IPSec

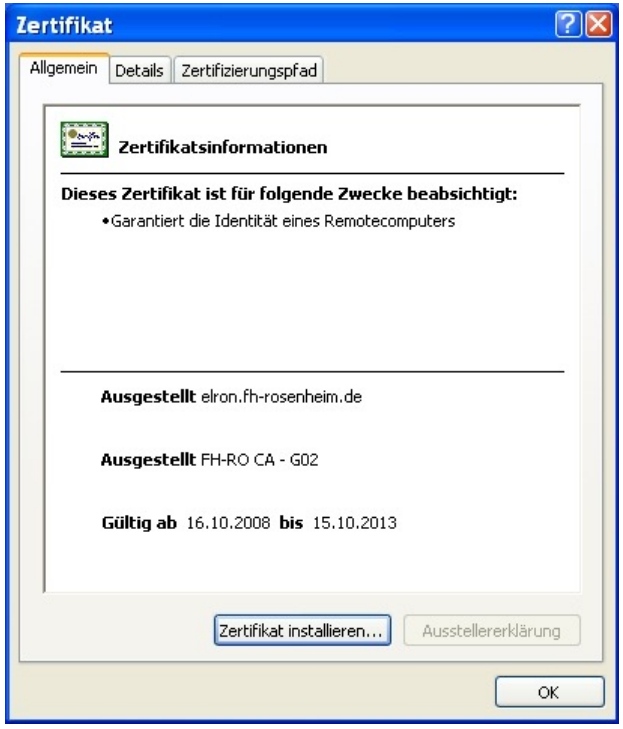
Elements of a X.509 certificate

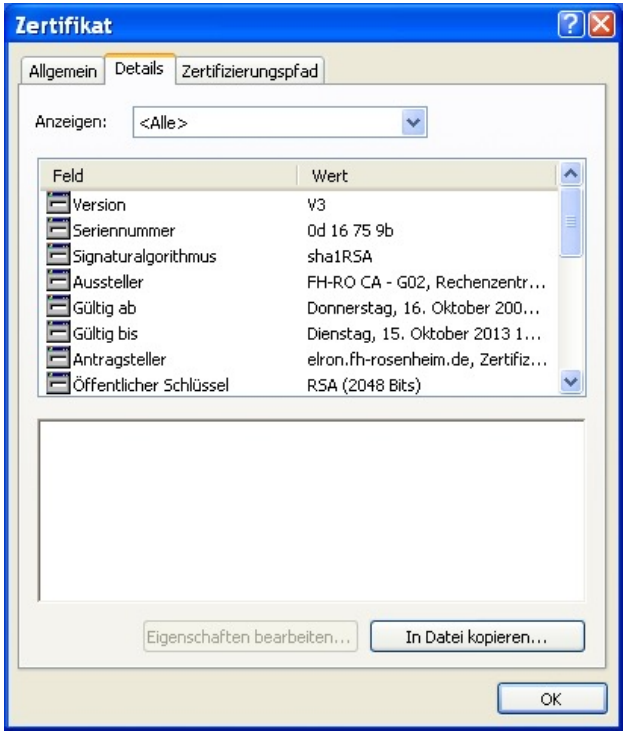


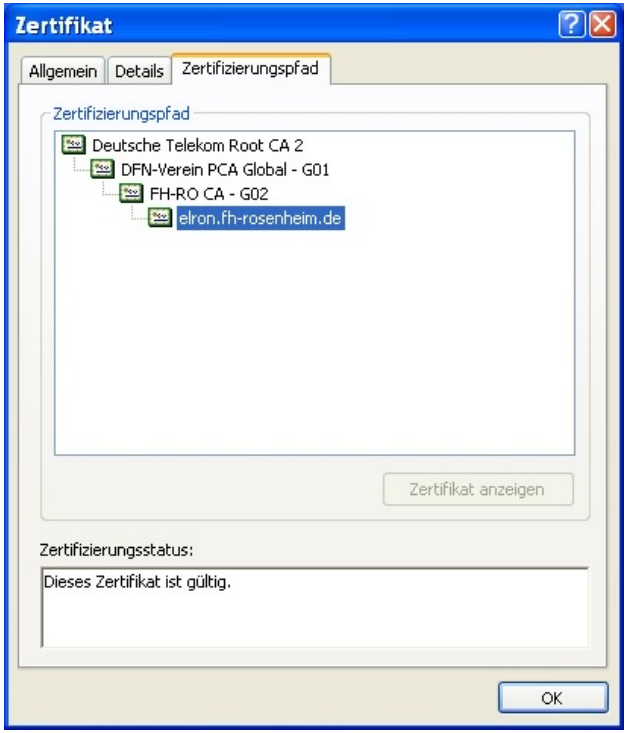

Source: Stallings, William. *Cryptography and Network Security: Principles and Practice, Global Edition*. Available from: VitalSource Bookshelf, (8th Edition). Pearson International Content, 2022.

# Standard for certificate format X509v3

- Description language of certificates is ASN.1 (Abstract Syntax Notation)
- Subject Names: stored in format X.500 Distinguished Name
  - CN = Common Name, O = Organization, OU = Organization Untit, C = Country, S = State
- A lot of other attributes can be stored in a certificate (public key, serial number, issuer, …)



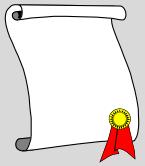
# Key and certificate generation

user generates key pair

**Keystore**

private key

public key



**CSR**

data (attributes) for the certificate

\+

Public Key

Certificate Signing Request

user identifies himself at RA

user generates a CSR and sends it to CA

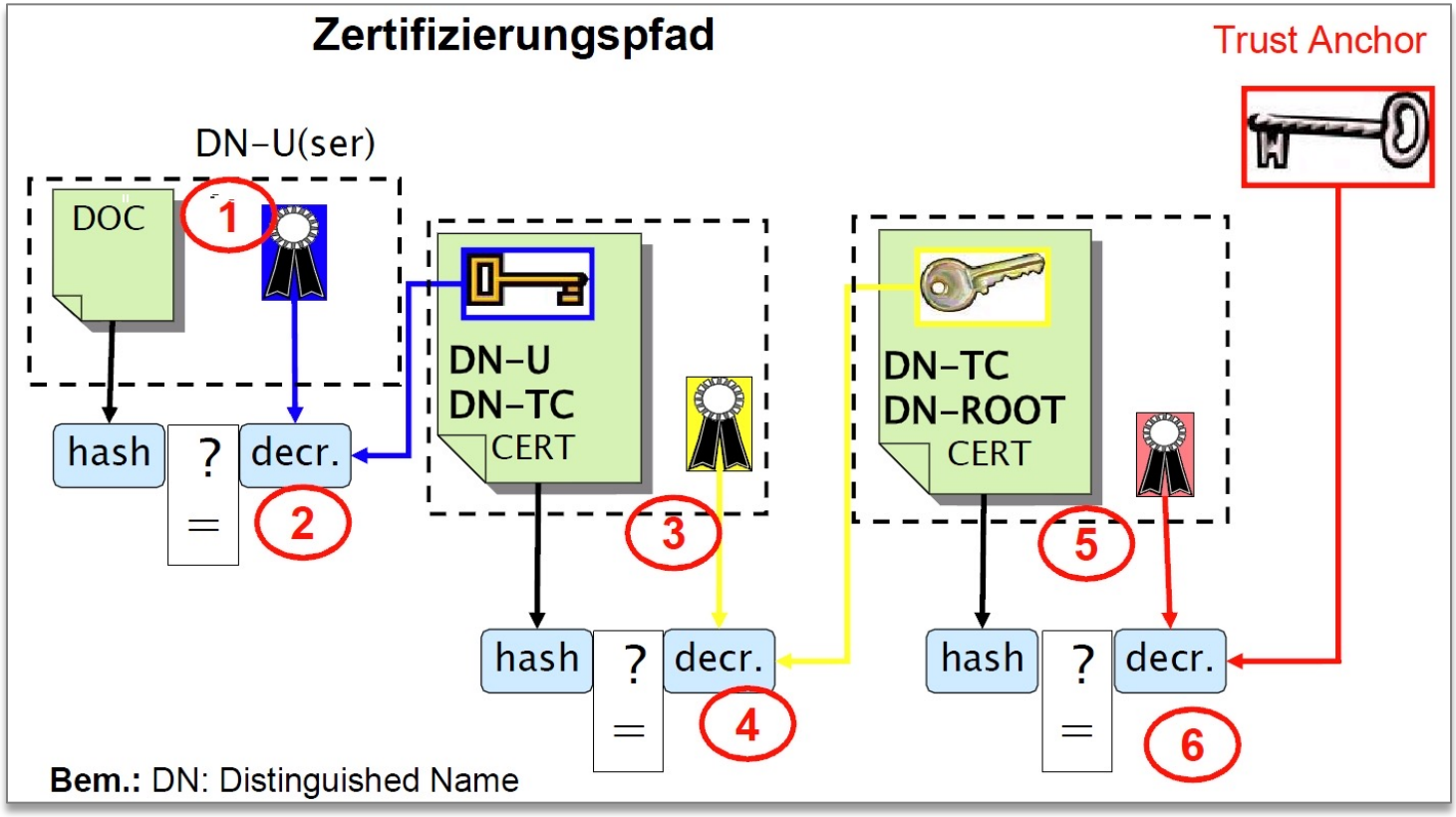
CA sends certificate to applicant

**RA**

RA confirms identity of applicant and public key

**CA**

# Verification of a signature and the certificates.

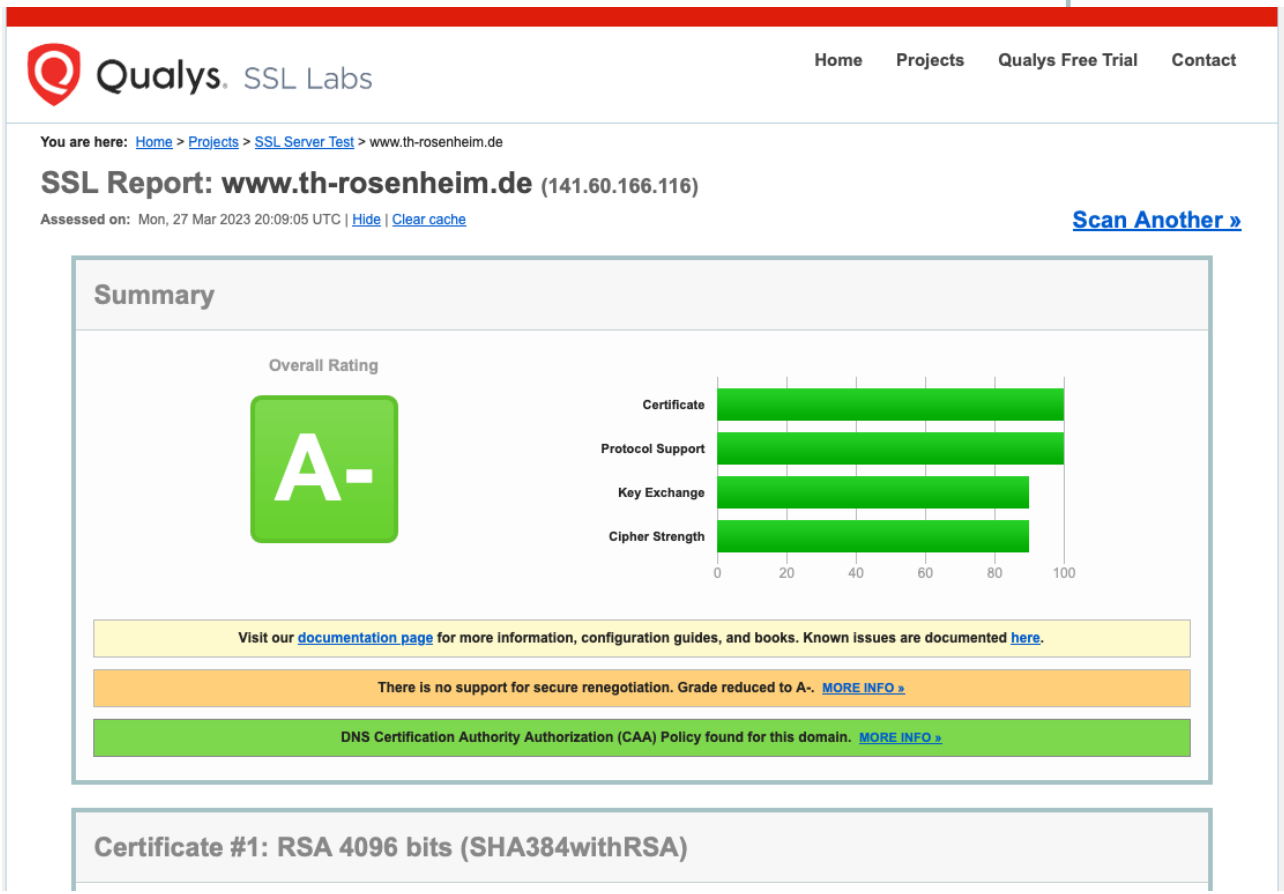


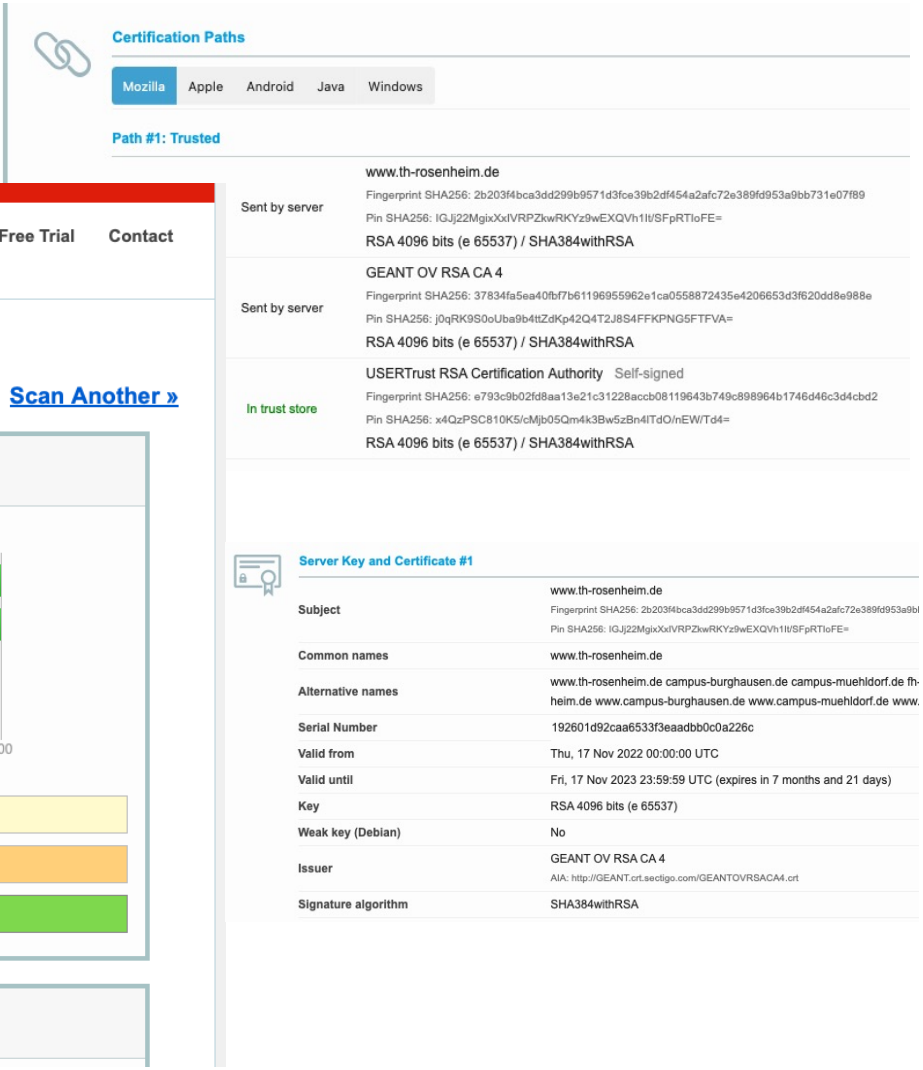

Source: Claudia Eckert, TUM

- Get all certificates of the chain (often via LDAP Lightweight Directory Access Protocoll)
- Check validity period and signatures of the certificates
- Check revocation list (**OCSP** Online Certificate Status Protocol)

# Online tool for verifying certificates.

https://www.ssllabs.com/ssltest/

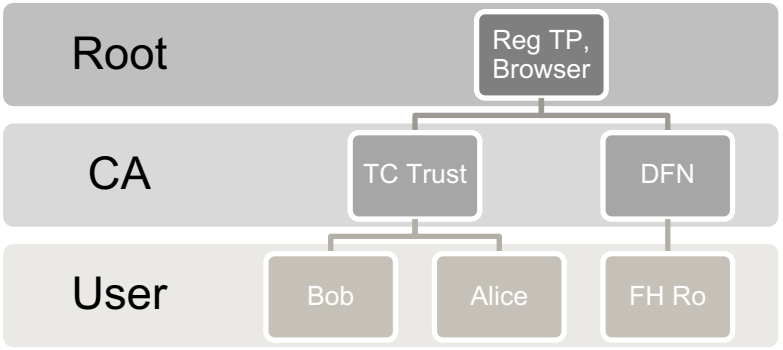


# Certification models

- Web of Trust
  - + simple and flexible in use
  - + many potential certificate chains
  - - no evidential value, or only with difficulty to obtain
  - - finding a trustworthy path is more complex

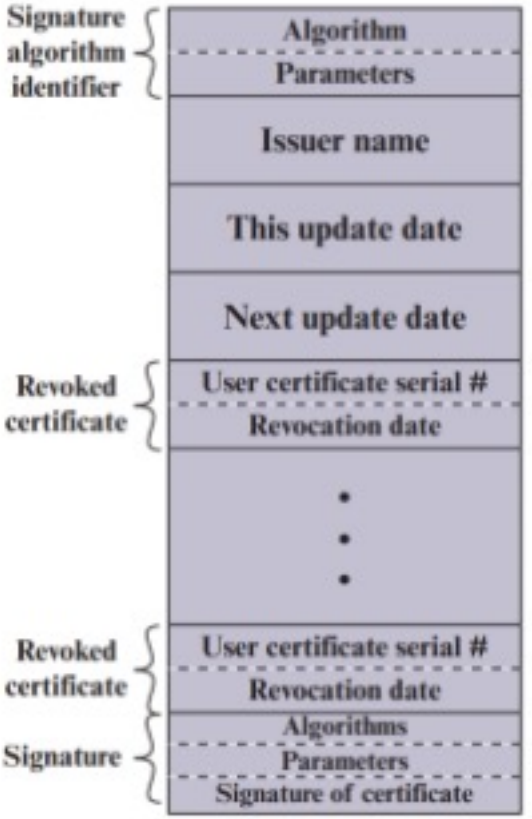
- Hierarchical certification
  - + clear structures and accountability
  - + evidential value in case of dispute
  - - overhead due to organizational structure




- Example for a free certification authority https://letsencrypt.org

# CRL Certificate Revocation List

- Motivation: In case of loss or theft (copy), a key must be blocked to warn other users
- Properties of the CRL
  - list of serial numbers of all revoked certificates
  - IETF standard
  - stored at directory service of CA
  - signed and timestamped by CA
  - frequently updated
- Issues
  - actuality
  - size (can get very large)
  - Distribution to the applications, how can clients access the list?
    - Download CRL of CA form link in certificate
    - Perform Online Status Check (OCSP Online Certificate Status Protocol) via link in certificate



# European eSignature Directive eIDAS

- **E**lectronic **ID**entification **A**uthentication and trust **S**ervices (since 1.7.2016)
- EU regulation on electronic identification, electronic signatures and trust services
  http://eur-lex.europa.eu/eli/reg/2014/910/oj
- Three level of signatures
  - **Simple** electronic signatures
    - Data in electronic form attached to the document
  - **Advanced** electronic signatures (AdES)
    - uniquely linked to and capable of identifying the signatory
    - linked to the document in a way that any subsequent change of the data is detectable.
  - **Qualified** electronic signatures
    - created by a qualified signature creation device (QES)
    - and is based on a qualified certificate for electronic signatures
    - it is equivalent to a handwritten signature.

# Components of the eIDAS regulation

- Electronic identification (identity card with eID functions)
- Trust services (providers of qualified services)
  - support creation and verification of **electronic signatures** (natural persons, declaration of intent), **electronic seals** (legal persons, proof of origin),electronic **time stamps**
  - deliver electronic registered mail
  - issue certificates for web site authentication
  - have to go through certification process themselves
- This enables creation of electronic documents with
  - electronic signature as legitimate proof
  - authentication of the document by electronic seal
  - proof of creation by time stamp
  - confirmation of receipt by electronic delivery services

# Applications of digital signatures


- Digital identity card
- Archiving of documents (with time stamp)
- Paperless invoices, reminders for invoices
- Public authorities (e-government, land registry)
- Electronic tax declaration (Elster)
- Pension account information
- Communication with patent court, patent office
- Digital banking transactions
- Electronic signatures with cell phone or tablet

# Summary checksums and digital signatures

- Cryptographic checksums such as MAC enable the authentication of data
- Digital signatures are a combination of hash value calculation and asymmetric encryption
- During implementation, many aspects must be considered (multiple signatures, signature renewal, canonicalization)
- In practice, digital signatures often require a great effort for hardware, software and process redesign
- A PKI is the basis for certificates and public key management
- A PKI enables digital signatures and confidential communication
- A certificate is a digital identity card that should be issued by a trustworthy trust center.
- Verification of a certificate requires checking the certificate chain, the CRL and the content of the certificate.