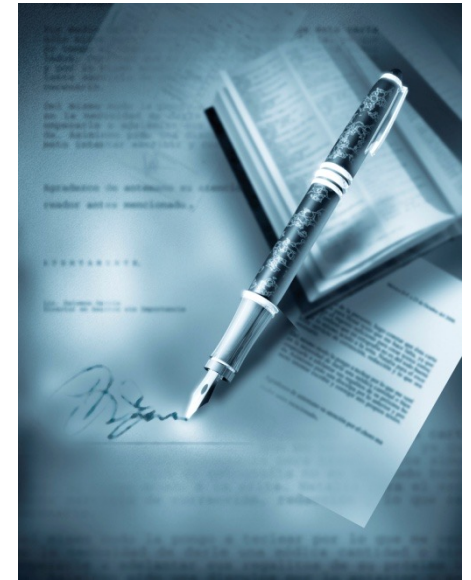


IT-Sicherheit



Kapitel 3: Prüfsummen und Digitale Signaturen

Teil 2

- ▶ Praktische Aspekte bei digitalen Signaturen
- ▶ Komponenten einer PKI
- ▶ Zertifikate (X509, XML)
- ▶ Signaturgesetz

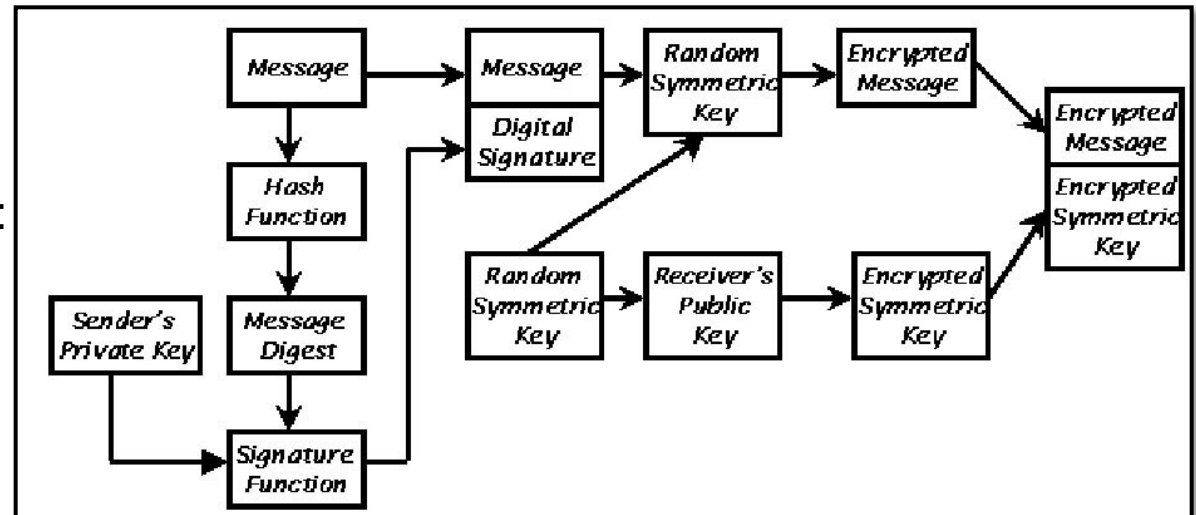


Praktische Aspekte bei Digitalen Signaturen

- ▶ Darstellungsproblem: Bei Signaturen muss man alles sehen was man unterschreibt
 - ▶ **WYSIWYS** (**W**hat **y**ou **s**ee **i**s **w**hat **y**ou **s**ign)
 - ▶ Es gibt Dokumentformaten mit Inhalte die man nicht sieht, z.B. Makros in Word, Javascript in Webseiten
 - ▶ Was mache ich mit solchen Dokumenten?
 - ▶ Versteckte Inhalte anzeigen oder eliminieren oder Dokument umformatieren
- ▶ Bei Kombination mit Verschlüsselung
 - ▶ Zuerst signieren dann verschlüsseln
 - ▶ Sonst unterschreibt man ein Dokument das man nicht lesen kann

▶ PKCS#7 Signatur Standard

- ▶ Beschreibt Aufbau von verschlüsselten und signierten Nachrichten
- ▶ Mehrere Formate: Data, Signed-Data, Enveloped-Data, Signed-and-enveloped-Data
- ▶ Prozess um einen digitalen Umschlag (envelop) um digital signierte Daten zu erzeugen (**Signed-and-enveloped-Data**) :



- ▶ Weitere weltweit akzeptierte PKCS-Standards der Firma RSA Laboratories (EMC2) z.B.
PKCS #5 (Password-Based Cryptography Standard),
PKCS #10 (Certification Request Syntax Standard) findet man unter
https://de.wikipedia.org/wiki/Public-Key_Cryptography_Standards

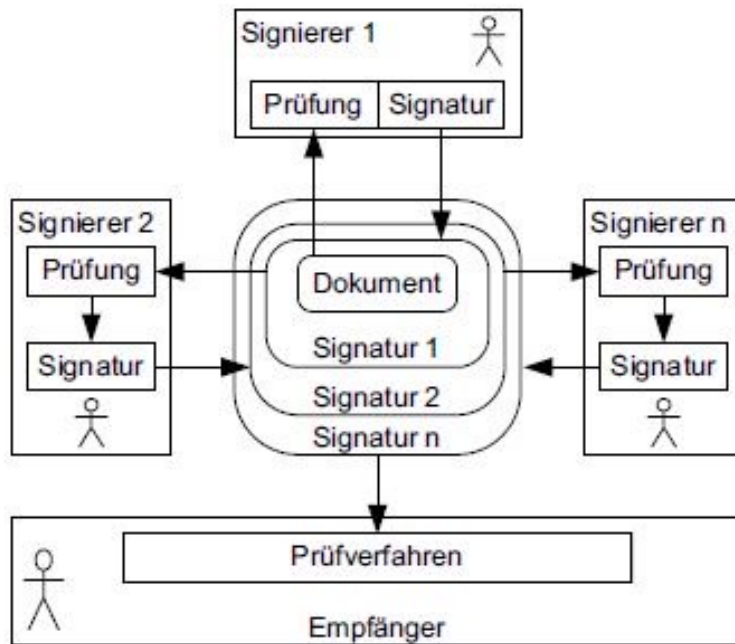


Beispiel für PKCS#7 Signatur

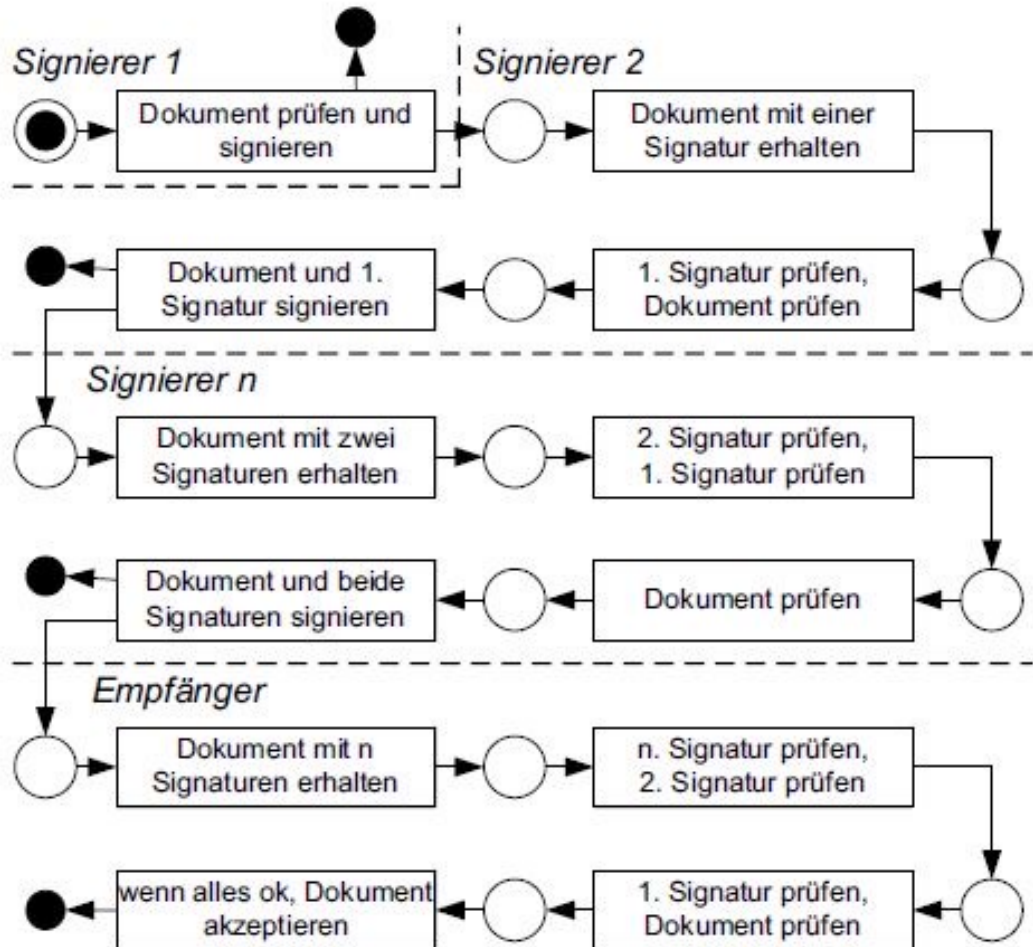
```
SignedData {
  version          0,
  digestAlgorithms {
    {1 3 36 3 2 1}, -- OID von RIPEMD-160
    {1 3 14 3 1 18} -- OID von SHA-1
  },
  encapContentInfo {
    eContentType {iso(1) member-body(2) us(840) rsadsi(113549)
                  pkcs(1) pkcs7(7) 1 } -- OID für Data Content
    eContent     [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 RSAsWithRIPEMD
      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
    }
  }
}
```

Wie funktionieren mehrfache Signaturen?

a) Struktur



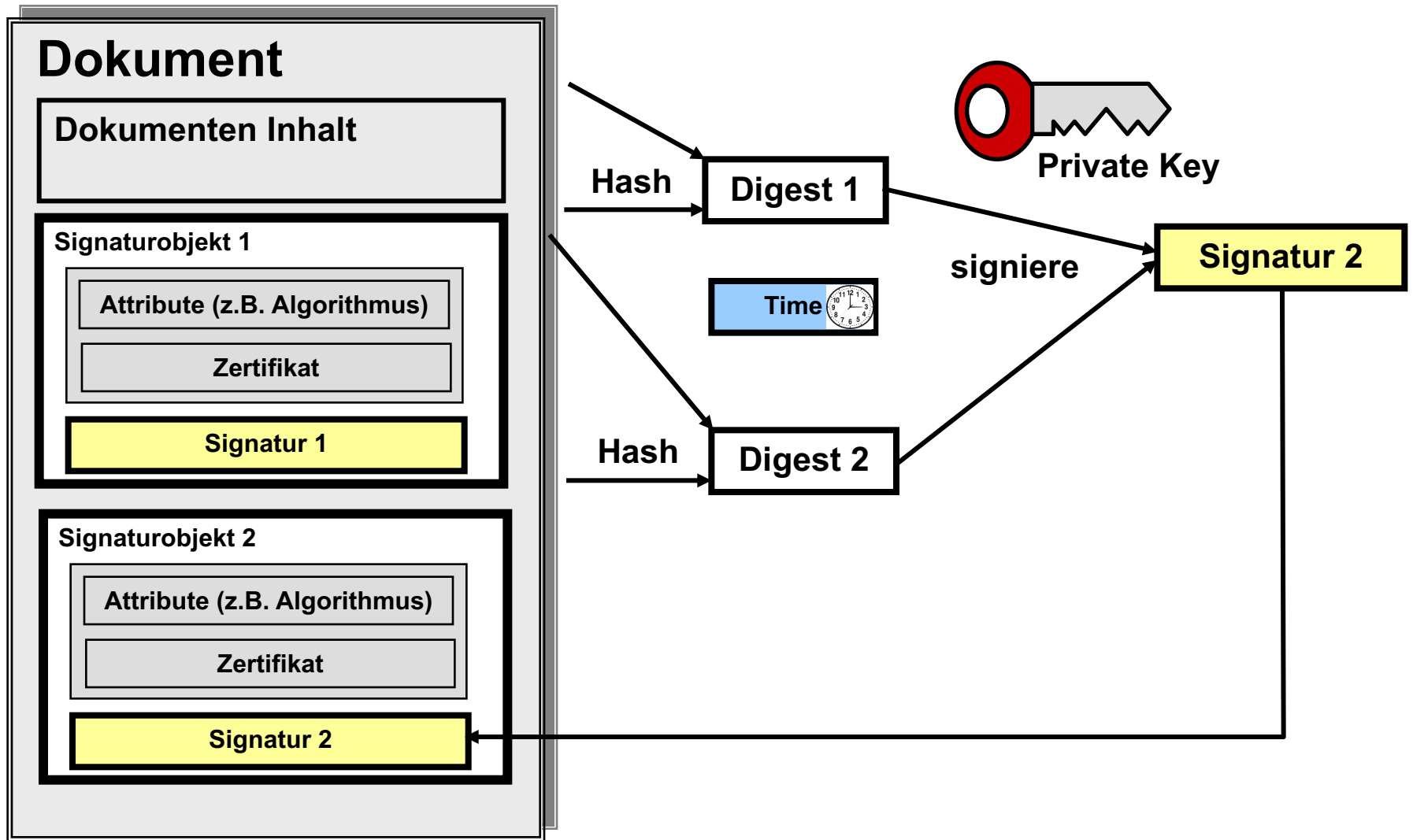
b) Ablauf



Quelle: Elektronische Signaturen in modernen Geschäftsprozesse, Gruhn et al, Vieweg, 2007 (eBook in Bibliothek)



Signaturerneuerung





Signaturerneuerungsprozess



▶ Ursache

- ▶ Wenn Zertifikate nicht mehr im TrustCenter gelistet werden oder die Verfahren im Zertifikat unsicher sind
- ▶ Sicherheit der Verfahren (Hash, Verschlüsselung)
- ▶ Dateiformate und Signaturformate ändern sich
- ▶ Gesetze schreiben Nachprüfbarkeit für längeren Zeitraum vor

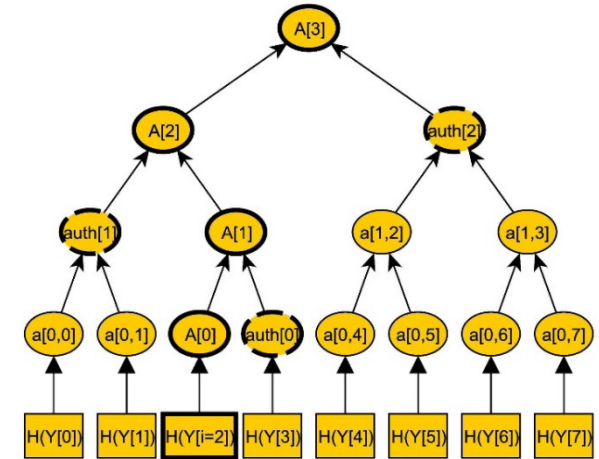
▶ Prozess der Neusignatur

- ▶ Verifikation der alten Signatur
- ▶ Erstellung der neuen Signatur
 - ▶ muss Daten und die alte Signatur umfassen
 - ▶ eventuell Formatwechsel von Dokument notwendig
- ▶ Vorgang muss in sicherer Umgebung stattfinden
- ▶ Verfahren sollte zertifiziert werden um Rechtssicherheit zu gewährleisten



Merkle Signaturen

- ▶ Merkle Signaturen sind ein Signaturschema basierend auf Hash-Bäumen (Merkle-Baum)
- ▶ Der öffentliche Schlüssel ist die Wurzel des Merkle-Baums
- ▶ Die Anzahl der Signaturen pro öffentlichen Schlüssel ist begrenzt (durch eine Potenz von 2, die Anzahl der Blätter)
- ▶ Bei der Signatur werden zusätzlich noch die Hashwerte entlang des Pfades zur Wurzel angehängt.
- ▶ Sind alle Blätter verbraucht muss ein neuer Baum verwendet werden
- ▶ Merkle Signaturen sind resistent gegen **Quantencomputer**
- ▶ Merkle-Bäume werden in Blockchains zur Authentisierung eingesetzt (z.B. Bitcoin): Hash Bäume zur Sicherung der Integrität sind effizienter als Hash-Listen

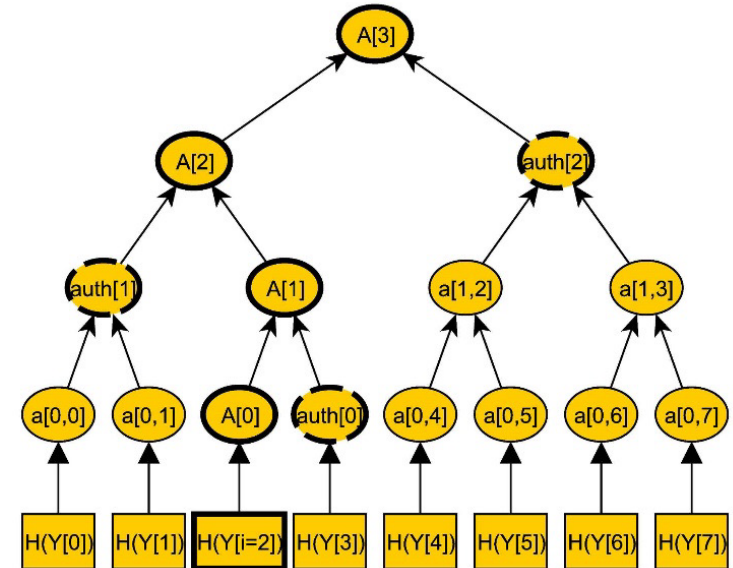




Signatur und Verifikation mit einem Merkle Signaturverfahren

Signatur mit Merkle Schema:

1. Generiere n Schlüsselpaare (X_i, Y_i) , X_i ist private Key, Y_i ist public Key
im Beispiel ist $i=8$
2. Berechne Merkle Baum
3. $A[n]$ ist Public Key des Merkle Baums
4. Signiere Nachricht M mit X_i , $\rightarrow \text{sig}'$
5. Berechne Pfad von Y_i bis zur Wurzel
Bsp für $i=2$
 $A[0] = H(Y_2)$
 $A[1] = H(A[0] \parallel \text{auth}[0]) = H(A[0] \parallel H(Y_3))$
 $A[2] = H(A[1] \parallel \text{auth}[1]) = H(A[1] \parallel H(a[0,0] \parallel H(a[0,1])))$
 $\quad = H(A[1] \parallel H(H(Y_0) \parallel H(Y_1)))$
 $A[3] = H(A[2] \parallel \text{auth}[2])$
6. Signatur $\text{sig} = (\text{sig}', \text{auth}[0], \text{auth}[1], \text{auth}[2])$



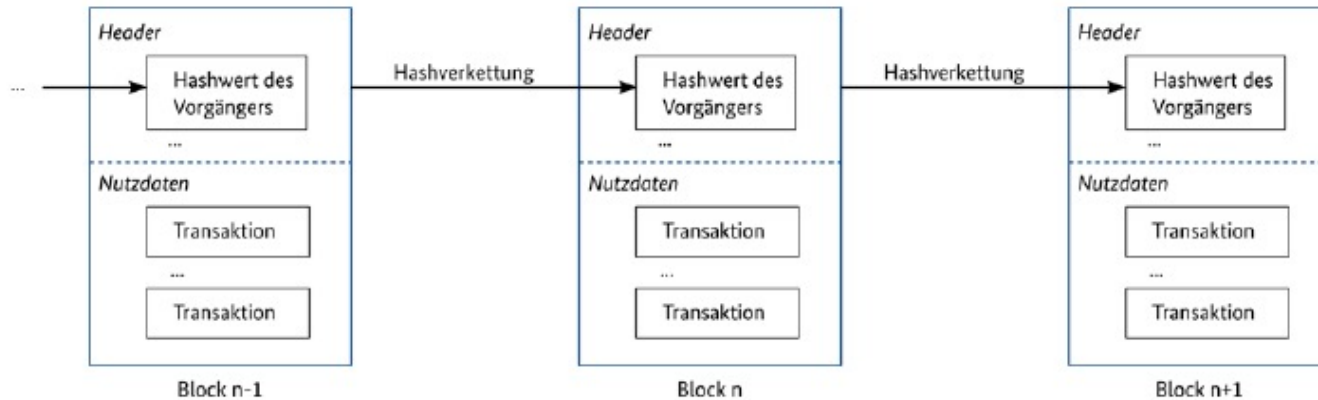
Quelle <https://deacademic.com/dic.nsf/dewiki/2506511>

Verifikation

1. Verifiziere die Signatur sig' mit Y_2
2. Berechne $A[3]$ aus Y_2 , $\text{auth}[0]$, $\text{auth}[1]$, $\text{auth}[2]$
3. Überprüfen ob Public Key von Merkle Baum identisch $A[3]$ ist



Die Blockchain als Beispiel für Anwendung von Kryptographie



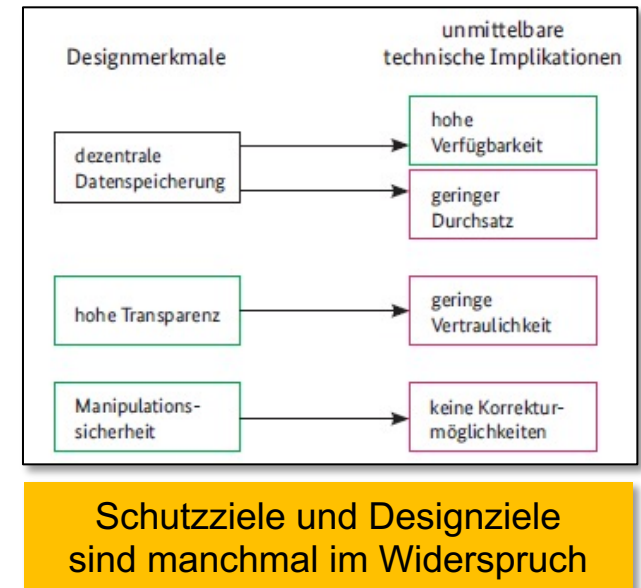
Quelle: Blockchain sicher gestalten, BS Bundesamt für Sicherheit in der Informationstechnik, März 2019)

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.html

- ▶ Die Verkettung der Blöcke wird durch **Hashwerte** vor Manipulationen geschützt.
- ▶ Der Algorithmus zum **Konsens** (einen neuen Block in die Kette hängen) basiert meist auf kryptographische Verfahren
 - ▶ z.B. bei **Bitcoin** wird der Block nur akzeptiert wenn der **Miner** einen Hashwert für den Block findet der mit einer vorgegebene Anzahl von Nullen beginnt. Dazu darf der Miner eine beliebige Zahl (Nonce) anhängen bis er einen geeigneten Hash hat.
- ▶ Signaturen mit einem public key der keinem expliziten Benutzer zugeordnet ist (**Pseudonymisierung**)

▶ Die Ziele der IT-Sicherheit und die Blockchain

- ▶ Die **Integrität** basiert auf Hashwerte
- ▶ Die **Verfügbarkeit** erfolgt durch Dezentralität
- ▶ Die **Vertraulichkeit** ist schwierig umzusetzen und oft nicht gewünscht.
 - ▶ Externe Speicherung vertraulicher Daten
 - ▶ Komplexe Verfahren (homomorphe Verschlüsselung, Trusted Execution Environments TEE, secure Multi-Party Computation sMPC)
- ▶ **Authentizität** basiert auf private Signaturschlüsseln
 - ▶ Bei privaten Blockchains Identifizierung der Konten erwünscht
 - ▶ Bei öffentlichen Blockchains liegt Pseudonymität vor

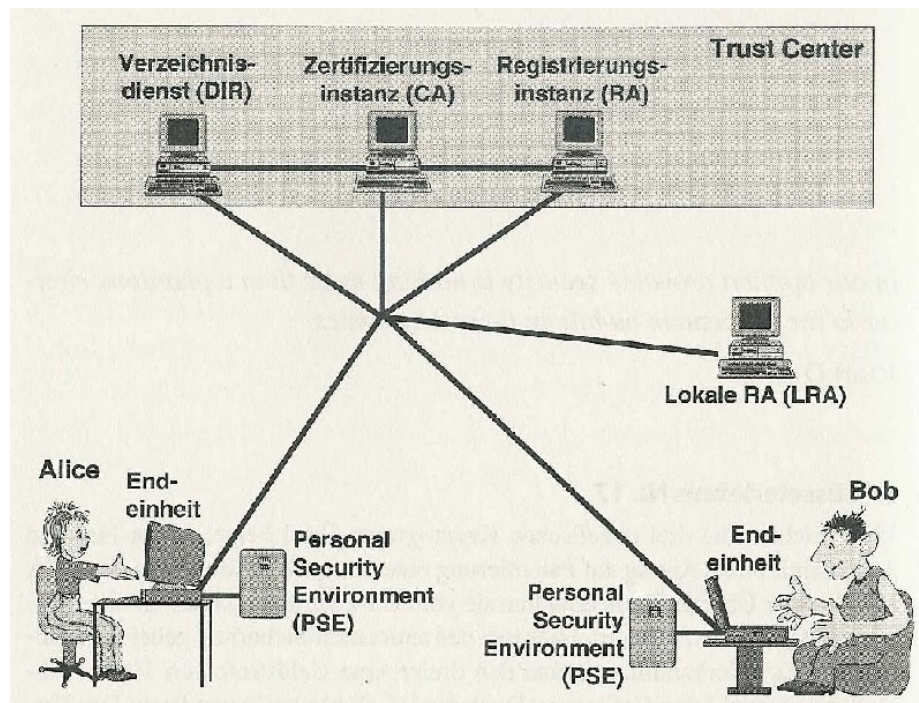


Quelle: Blockchain sicher gestalten, BS Bundesamt für Sicherheit in der Informationstechnik, März 2019)
https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Krypto/Blockchain_Analyse.html



Public Key Infrastruktur (PKI)

- ▶ Was ist eine PKI
 - ▶ Bringt vertrauliches und effizientes Key- und Zertifikats-Management
 - ▶ Interface für Trust Services
 - ▶ Trust Services decken Lebenszyklus von Zertifikaten ab (Generierung, Verifikation, Revozierung)



Quelle: Klaus Schmech: Kryptografie, dpunkt.verlag



Komponenten einer PKI

- ▶ CA Certification Authority
 - ▶ Erstellt Zertifikate
- ▶ RA Registration Authority
 - ▶ Schnittstelle zwischen CA und Subjekten (Anmeldestelle)
 - ▶ Subjekt Identifikation
- ▶ Verzeichnisdienst
 - ▶ Enthält Liste aller ausgestellten Zertifikate
 - ▶ Revocation-List
- ▶ Endeinheit
 - ▶ Realisiert Anwendung (PC, Handy, ...)
- ▶ Personal Security Environment (PSE)
 - ▶ Umgebung in der Schlüssel gespeichert ist (Chip Karte, Festplatte, ...)
- ▶ Weitere optionale Komponenten
 - ▶ Zeitstempeldienst TSS
 - ▶ Sperr-Instanz (REV)
 - ▶ Recovery-Instanz



Zertifikate sind digitale Ausweise

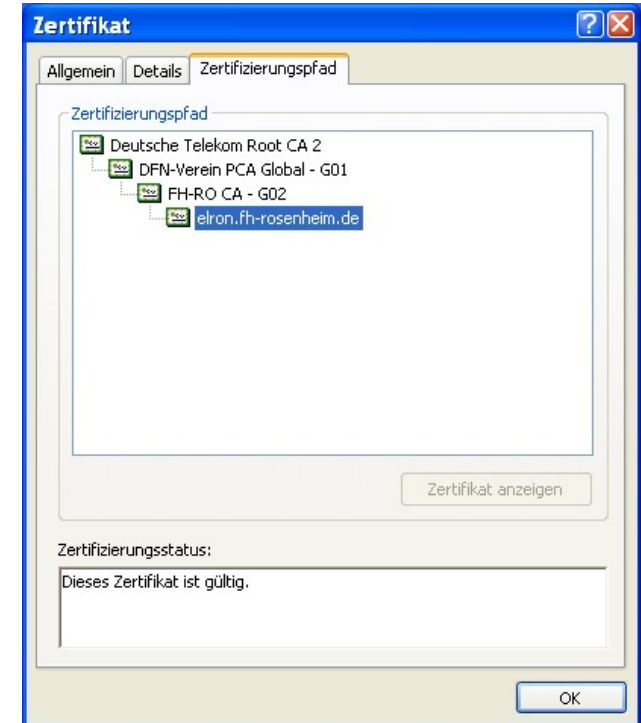
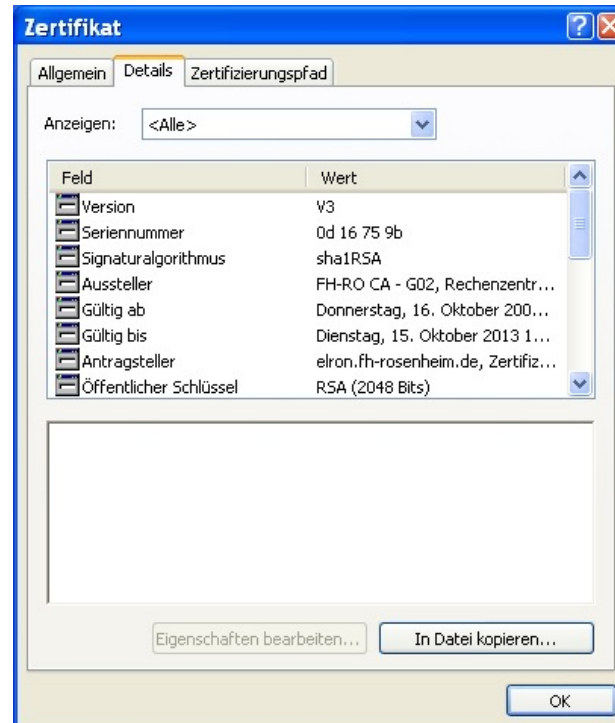
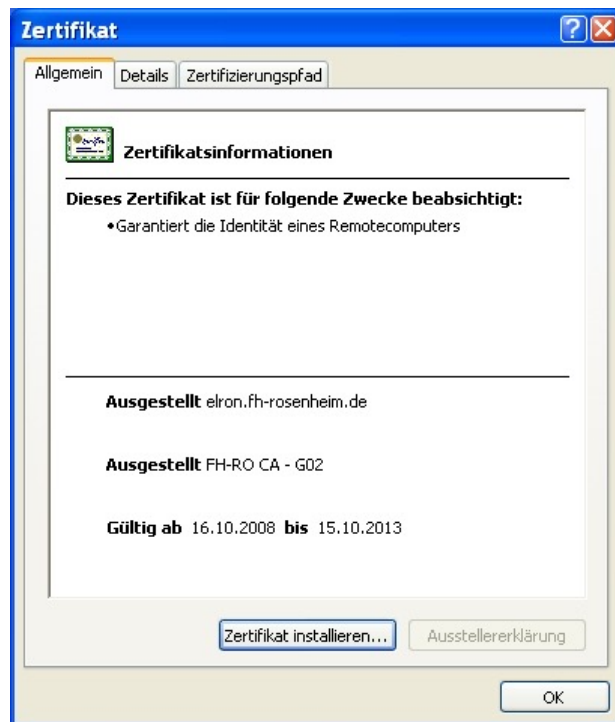


- ▶ Problem: Authentische Bereitstellung von öffentlichen Schlüsseln (man in the middle)
- ▶ Lösung: Certification Authorities (Trust Center) kontrollieren die Identität des Inhabers und garantieren die Echtheit der Schlüssel
- ▶ Bestandteile eines Zertifikats
 - ▶ Laufende Nummer
 - ▶ Bezeichnung der Algorithmen zur Benutzung des Schlüssels
 - ▶ Namen und Daten des Ausstellers (Certification Authority)
 - ▶ Gültigkeitszeitraum
 - ▶ Öffentlichen Schlüssel des Inhabers
 - ▶ Persönliche Daten des Inhabers
 - ▶ Signatur des Zertifikats (durch Aussteller)
- ▶ Haben begrenzte Lebensdauer
- ▶ Können revoziert werden
- ▶ Werden von Protokollen wie SSL, S/MIME, IPSec verwendet

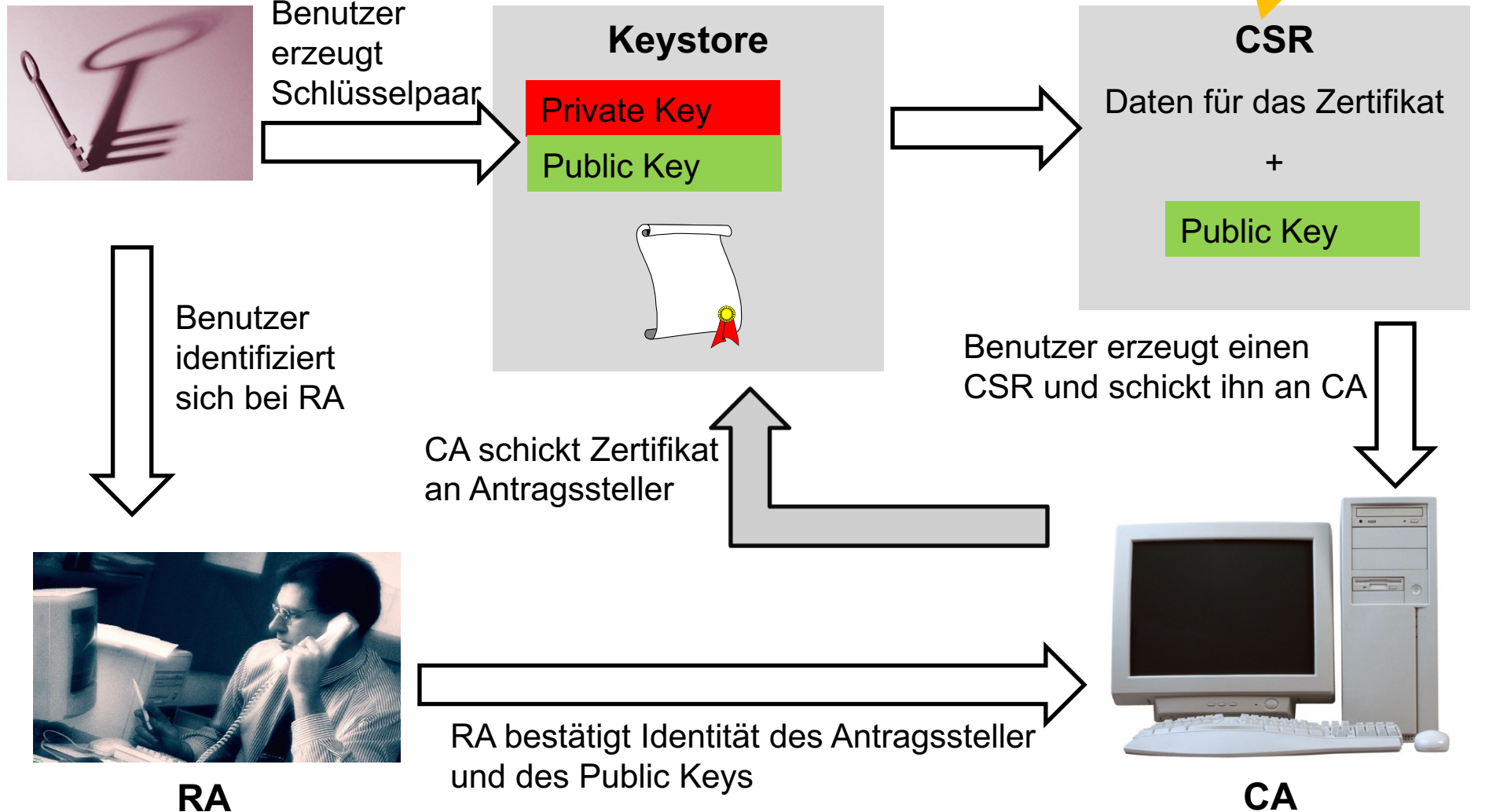


Standard für Zertifikatsformat X509v3

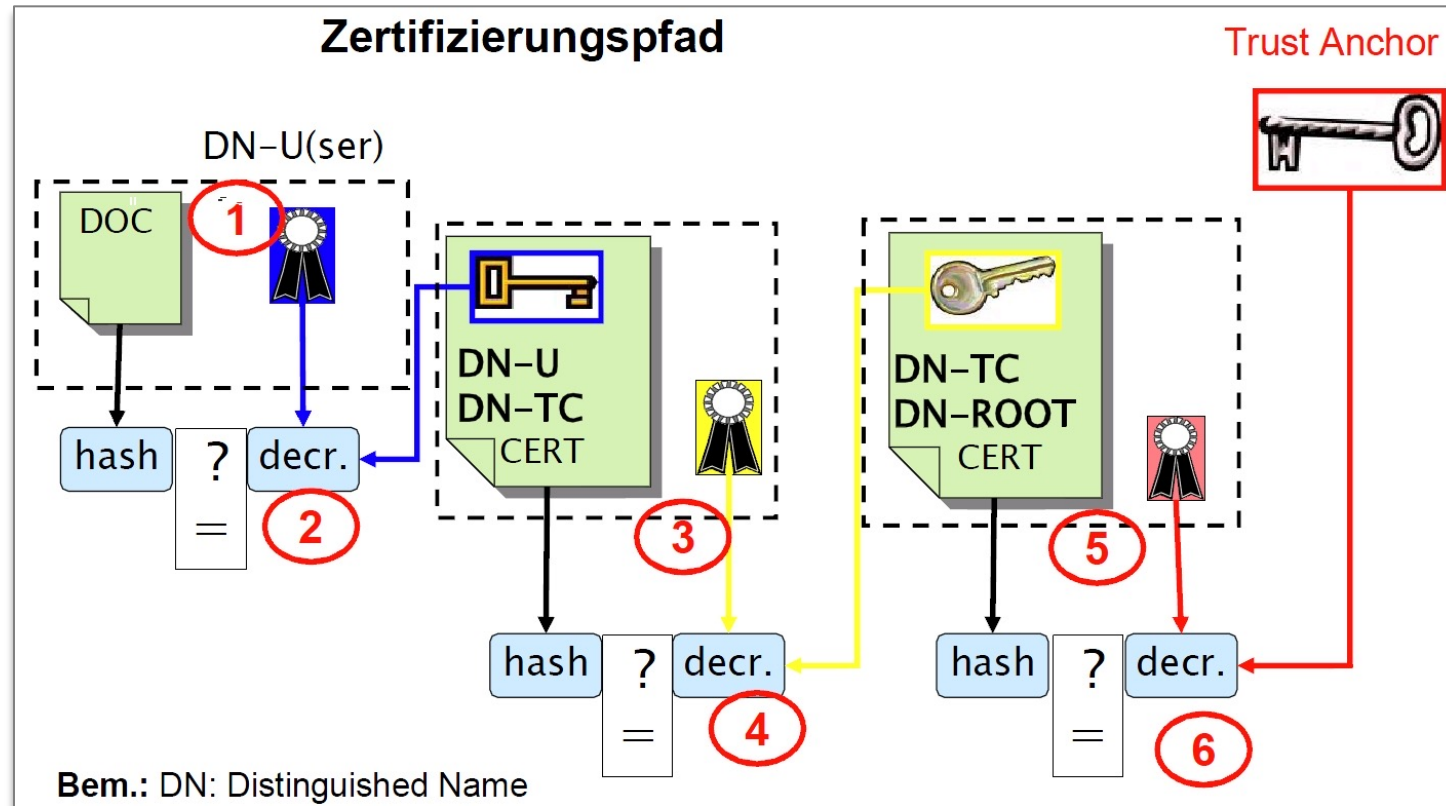
- ▶ Format der Zertifikate ist ASN.1
- ▶ Subject Names: Distinguished name DN – X500
- ▶ Gespeichert DER codiert (Distinguished Encoding Rules), Base 64 codiert oder im PKCS#7 Format



▶ Key- und Zertifikats-Generierung



▶ Verifikation einer Signatur und der Zertifikate


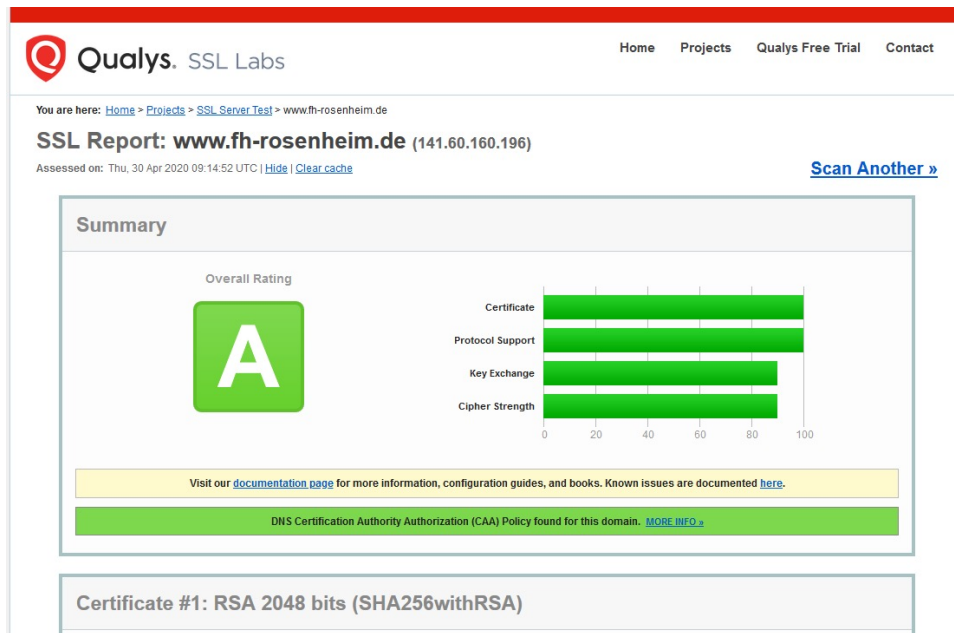


Quelle: Claudia Eckert, TUM

- ▶ Alle Zertifikate der Kette holen (oft per LDAP Lightweight Directory Access Protocol)
- ▶ Gültigkeitsperiode der Zertifikate Checken
- ▶ Revozierungsliste checken (z.B. OCSP Online Certificate Statusprotocol)

Onlinetool zum Überprüfen von Zertifikaten

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



Zertifikat

Allgemein Details Zertifizierungspfad

Zertifikatsinformationen

Dieses Zertifikat ist für folgende Zwecke beabsichtigt:

- Garantiert die Identität eines Remotecomputers
- Garantiert dem Remotecomputer Ihre Identität
- 1.3.6.1.4.1.22177.300.1.1.4
- 1.3.6.1.4.1.22177.300.30

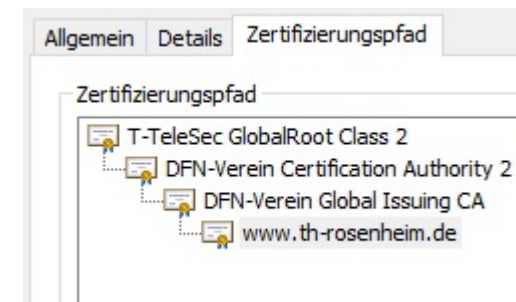
Ausgestellt für: www.th-rosenheim.de

Ausgestellt von: DFN-Verein Global Issuing CA

Gültig ab: 25.09.2018 **bis:** 27.12.2020

[Ausstellereklärung](#)

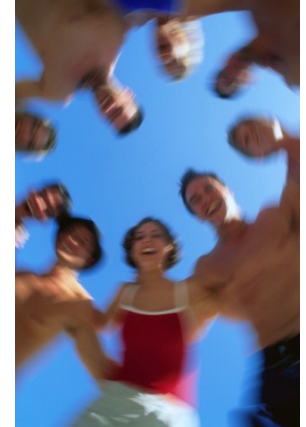
OK



▶ Zertifizierungsmodelle

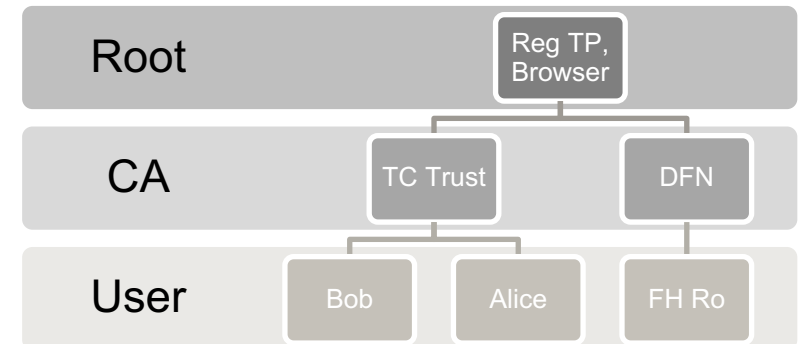
▶ Web of Trust

- ▶ + Einfache flexible Nutzung
- ▶ + Viele potentielle Zertifikatsketten
- ▶ - keine oder nur schwer erreichbare Beweiskraft
- ▶ - Finden eines vertrauenswürdigen Pfades aufwendiger



▶ Hierarchische Zertifizierung

- ▶ + Klare Strukturen und Zurechenbarkeiten
- ▶ + Beweiskraft im Streitfall
- ▶ - Overhead durch Organisationsstruktur



▶ Beispiel für eine freie Zertifizierungsstelle <https://letsencrypt.org/de/>



CRL Certificate Revocation List

- ▶ Motivation: Bei Verlust oder Diebstahl (Kopie) muss ein Schlüssel gesperrt werden
- ▶ Eigenschaften der CRL
 - ▶ Liste der Seriennummern aller revozierten Zertifikate
 - ▶ IETF Standard
 - ▶ Gespeichert bei Verzeichnisdienst der CA
 - ▶ Signiert und mit Zeitstempel versehen durch CA
 - ▶ Häufig aktualisiert
- ▶ Probleme
 - ▶ Aktualität, Größe
 - ▶ Distribution, Wie greifen Clients zu
 - ▶ Polling
 - ▶ Push
 - ▶ Online Status Check (z.B. OCSP Online Certificate Status Protocol)
 - ▶ Delta Sperrlisten
 - ▶ Verteilte Sperrlisten, Sperrbäume



Europäische Signaturverordnung eIDAS-VO TODO



- ▶ Electronic **ID**entification **A**uthentication and trust **S**ervices (seit 1.7.2016)
- ▶ Verordnung über elektronische Identifizierung und Vertrauensdienste
<http://eur-lex.europa.eu/eli/reg/2014/910/oj>
- ▶ EU-Richtlinie über elektronische Signaturen
- ▶ Löst das Deutsche Signaturgesetz (SigG) und die Verordnung zur elektronischen Signatur (SigV) von 2001 ab



Bestandteile des eIDAS-Verfahren



- ▶ Elektronische Identifizierung (Personalausweis mit eID-Funktionen)
- ▶ Vertrauensdienste (Anbieter qualifizierter Dienste)
 - ▶ Erstellung und Überprüfung von **elektronischen Signaturen** (natürliche Personen, Willenserklärung), **elektronische Siegel** (juristische Personen, Herkunftsnachweis), elektronische Zeitstempel
 - ▶ Zustellung elektronischer Einschreiben
 - ▶ Zertifikate für Webseiten Authentifizierung
 - ▶ Müssen selbst Zertifizierungsprozess durchlaufen
- ▶ Das ermöglicht Erstellung von elektronischen Dokumenten mit
 - ▶ Elektronischer Signatur als legitimer Nachweis
 - ▶ Authentifizierung des Dokument durch elektronisches Siegel
 - ▶ Nachweis der Erstellung durch Zeitstempel
 - ▶ Empfangsbestätigung durch elektronische Zustellservices



Anwendungen

- ▶ Digitaler Personalausweis
- ▶ Archivierung von Dokumenten (Zeitstempel)
- ▶ Papierlose Rechnungen, Mahnungen
- ▶ Behörden (E-Government 2.0, Grundbuchamt)
- ▶ Elektronische Steuererklärung (Elster)
- ▶ Abfrage vom Rentenkonto
- ▶ Patentgericht, Patentamt
- ▶ Digitale Bankgeschäfte (z.B. Kreditabschluss)
- ▶ Elektronische Unterschriften mit Handy oder Tablet



Zusammenfassung Prüfsummen und Digitale Signaturen



- ▶ Kryptographische Prüfsummen wie MAC ermöglichen die Authentisierung von Daten
- ▶ Digitale Signaturen sind eine Kombination aus Hash-Wert Berechnung und asymmetrischer Verschlüsselung
- ▶ Bei der Umsetzung sind viele Aspekte zu berücksichtigen (Mehrfachsignaturen, Signaturerneuerung, Kanonisierung)
- ▶ In der Praxis erfordern digitale Signaturen oft hohe Aufwände für Hardware, Software und Neugestaltung der Prozesse
- ▶ Eine PKI ist die Basis für Zertifikate und die Verwaltung öffentlicher Schlüssel
- ▶ Eine PKI ermöglicht digitale Signaturen und vertrauliche Kommunikation
- ▶ Ein Zertifikat ist ein digitaler Personalausweis der von einem vertrauenswürdigen Trust Center ausgestellt werden sollte
- ▶ Die Verifikation eines Zertifikats erfordert die Überprüfung der Zertifikatskette, der CRL und des Inhalts des Zertifikats