Kapitel 4
Public-Key-Infrastrukturen (PKI)

1. Vertrauensmodelle
2. Zertifikate
3. PKI-Organisation
4. PKI-Anwendungen

Praktische Probleme bei der Anwendung von Public Key-Verfahren

- **Authentizität der Schlüssel**
  - Einem Schlüssel ist nicht anzusehen, wem er gehört (→ „Man-in-the-Middle Angriffe“)
- **Verbindlichkeit von Schlüsseln**
  - Es ist nicht beweisbar, wem ein Schlüssel gehört
- **Sperrung von Schlüsseln**
  - z.B. nach Diebstahl eines privaten Schlüssels
- **Vereinbarung organisatorischer Richtlinien**
  - Verbreitung von öffentlichen Schlüsseln, Schlüssellängen, zeitliche Gültigkeit von Schlüsseln, …

Vertrauensmodell: „Direct Trust“

- **Persönliche Kommunikation** zum direkten Austausch der öffentlichen Schlüssel (z.B. auf USB-Stick)
- Problemlösung:
  - Authentizität: gegeben
  - Verbindlichkeit: gering (öffentlicher Schlüssel ist nachträglich abstreitbar)
  - Sperrung: durch persönliche Information möglich
  - Organisatorische Richtlinien: schwer durchsetzbar
- Insgesamt: nur beschränkt einsetzbar (nur für private Nutzung)

Vertrauensmodell: „Web of Trust“

- Persönliche Kommunikation zum direkten Austausch der öffentlichen Schlüssel und zusätzlich **gegenseitige signierte Weitergabe von öffentlichen Schlüsseln** (wechselseitige „Bürgschaft“)
- Problemlösung:
  - Authentizität: gegeben
  - Verbindlichkeit: hoch (aufgrund der „Bürgen“)
  - Sperrung: sehr schwer durchführbar
  - Organisatorische Richtlinien: schwer durchsetzbar
- Insgesamt: nur beschränkt einsetzbar (nur für private Nutzung)

Vertrauensmodell: „Hierarchical Trust“

- **Eine** unabhängige, vertrauenswürdige Instanz übernimmt „Bürgschaften“ und Verteilung öffentlicher Schlüssel
- Bezeichnungen: „Certification Authority“ (CA) oder „Zertifizierungsstelle“
- Problemlösung:
  - Authentizität: gegeben
  - Verbindlichkeit: hoch (Bürgschaft durch zentrale Instanz)
  - Sperrung: durch zentrale Instanz gut durchführbar
  - Organisatorische Richtlinien: durch zentrale Instanz gut durchsetzbar
- Insgesamt: „Hierarchical Trust“ ist Basis der meisten „Public-Key-Infrastrukturen“

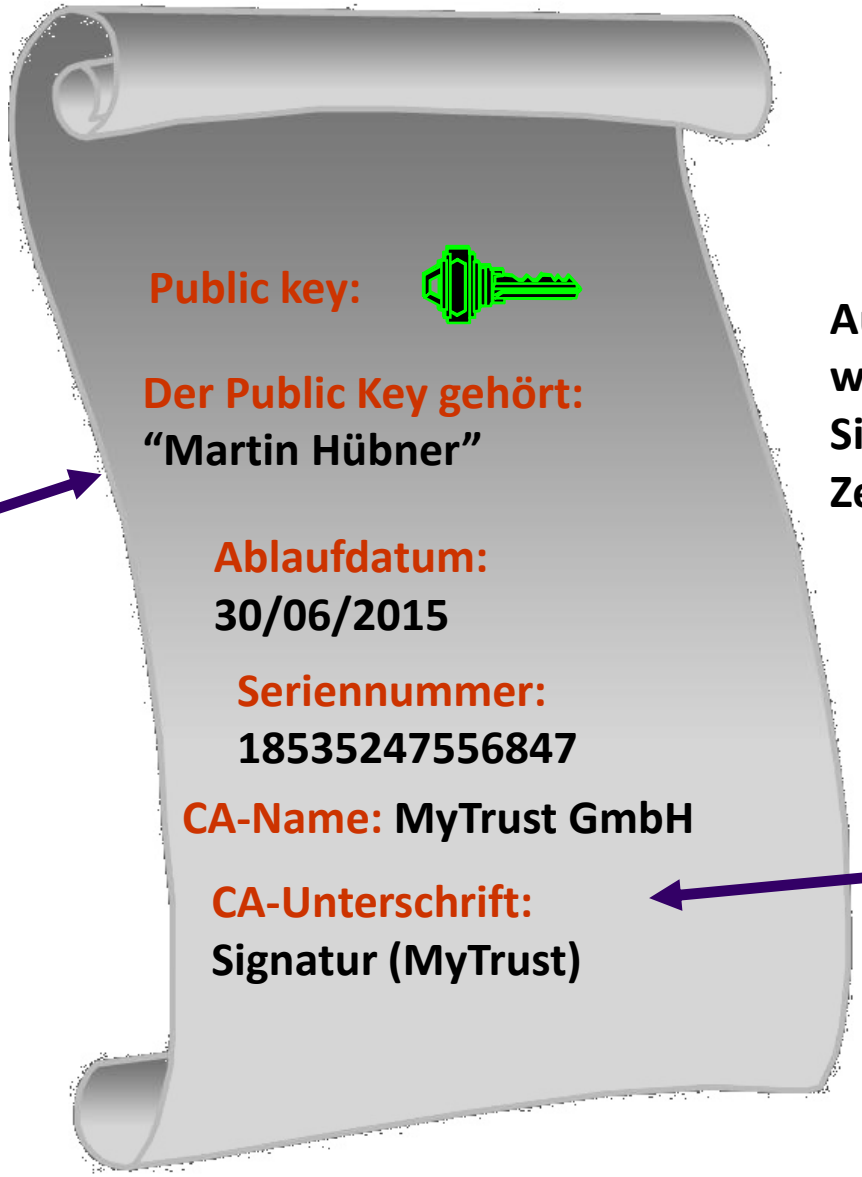
Kapitel 4
Public-Key-Infrastrukturen (PKI)

1. Vertrauensmodelle
2. Zertifikate
3. PKI-Organisation
4. PKI-Anwendungen

Zertifikate

- Ein Zertifikat wird von einer vertrauenswürdigen Stelle ausgestellt, um eine bestimmte Eigenschaft einer Entität zu beglaubigen
- Mit einem Digitalen Zertifikat wird die Zugehörigkeit eines öffentlichen Schlüssels zu einem Inhaber beglaubigt
- Certification Authorities (CAs) stellen digitale Zertifikate aus, um öffentliche Schlüssel zu beglaubigen.
- Die meisten heute im Einsatz stehenden Digitalen Zertifikate sind konform zu ITU-T X.509 Version 3 [RFC 2459]

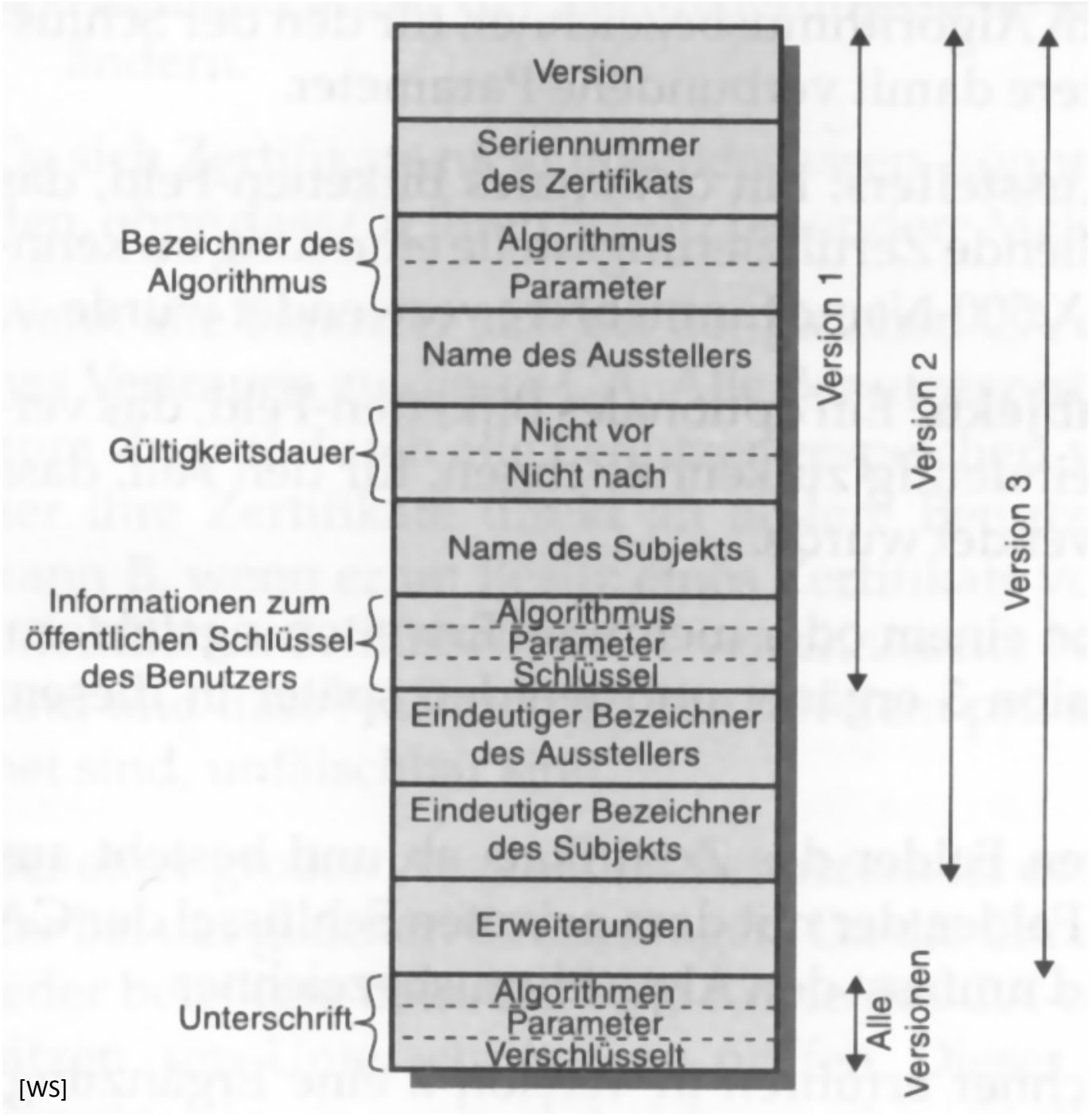
Inhalte von Zertifikaten nach X.509



Name des Inhabers wird an den öffentlichen Schlüssel (Public Key) gebunden

(eindeutiger Name innerhalb einer CA)

Authentizität des Zertifikats wird garantiert durch die Signatur der Zertifizierungsstelle (CA)



X.509 - Formate Version 1-3

[WS]

Exkurs: ASN.1 (Abstract Syntax Notation 1)

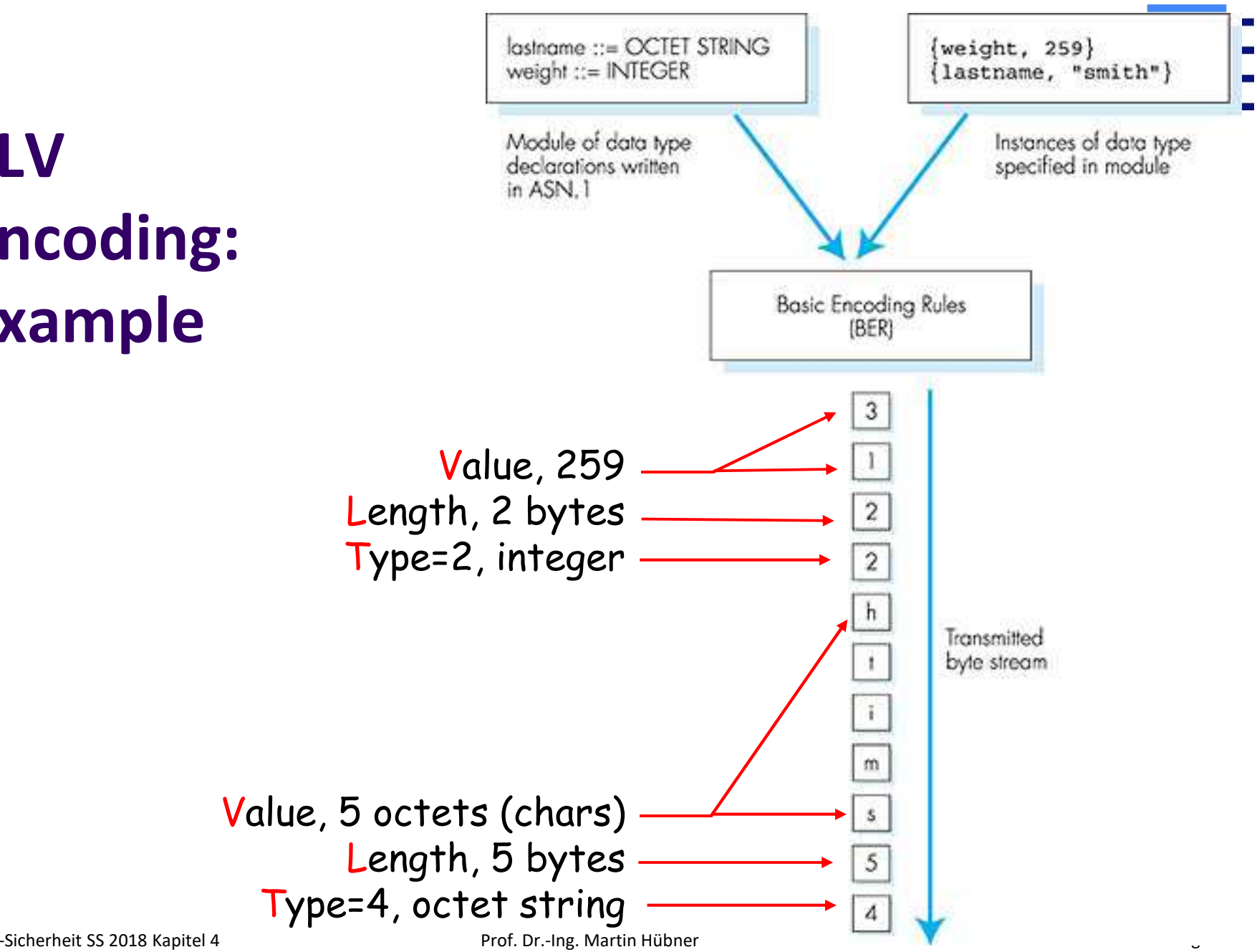
- Ziel: Effiziente, maschinenlesbare und plattformunabhängige Spezifikation von Datentypen
- ISO Standard X.680
- Alternativen: EDIFACT, XML, JSON, …
- Basisdatentypen:

Typcode	Typ	
1	Boolean	wahr/falsch
2	Integer	beliebig groß
3	Bitstring	ein oder mehrere Bits
4	Octet string	ein oder mehrere Bytes
5	Null	kein Wert
6	Object Identifier (OID)	Objekt im ISO-Namensbaum (siehe http://www.oid-info.com)
9	Real	Gleitkommawert

ASN.1-Anwendung: Encoding Rules

- Basic Encoding Rules („BER")
  Idee: Übertragene Daten sollen selbstbeschreibend sein
  ➔ TLV-Codierung jeden Wertes
  - T ("Type"): Datentyp (ASN.1-Typcode)
  - L ("Length"): Länge des Datenwerts in Byte
  - V ("Value"): Datenwert (gemäß ASN.1 – Regeln codiert)
- Distinguished Encoding Rules („DER")
  BER ohne Implementierungsoptionen
  → notwendig für Verschlüsselung/Hashing
- Packed Encoding Rules („PER")
  Effizienz-Optimierung: (L)V – Codierung
- XML Encoding Rules („XER")

TLV encoding: example



X.509v3-Spezifikation (ASN.1-Syntax) [RFC 2459]

```
Certificate  ::=  SEQUENCE  {
     tbsCertificate          TBSCertificate,
     signatureAlgorithm      AlgorithmIdentifier,  (OID)
     signatureValue          BIT STRING  }
TBSCertificate  ::=  SEQUENCE  {
     version                 EXPLICIT Version DEFAULT v1,
     serialNumber            CertificateSerialNumber,
     signature               AlgorithmIdentifier,  (OID)
     issuer                  Name,
     validity                Validity,
     subject                 Name,
     subjectPublicKeyInfo    SubjectPublicKeyInfo,
     issuerUniqueID          IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version shall be v2 or v3
     subjectUniqueID         IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version shall be v2 or v3
     extensions              EXPLICIT Extensions OPTIONAL
                        -- If present, version shall be v3
```

X.509v3-Spezifikation (Forts.) [RFC 2459]

```
Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }
CertificateSerialNumber  ::=  INTEGER
Validity ::= SEQUENCE {
        notBefore       Time,
        notAfter        Time }
Time ::= CHOICE {
        utcTime         UTCTime,
        generalTime     GeneralizedTime }
UniqueIdentifier  ::=  BIT STRING
SubjectPublicKeyInfo  ::=  SEQUENCE  {
        algorithm           AlgorithmIdentifier,
        subjectPublicKey    BIT STRING  }
AlgorithmIdentifier  ::=  SEQUENCE  {
        algorithm           OBJECT IDENTIFIER,
        parameters          ANY DEFINED BY algorithm OPTIONAL
   }
```

Kapitel 4
Public-Key-Infrastrukturen (PKI)

1. Vertrauensmodelle
2. Zertifikate
3. PKI-Organisation
4. PKI-Anwendungen

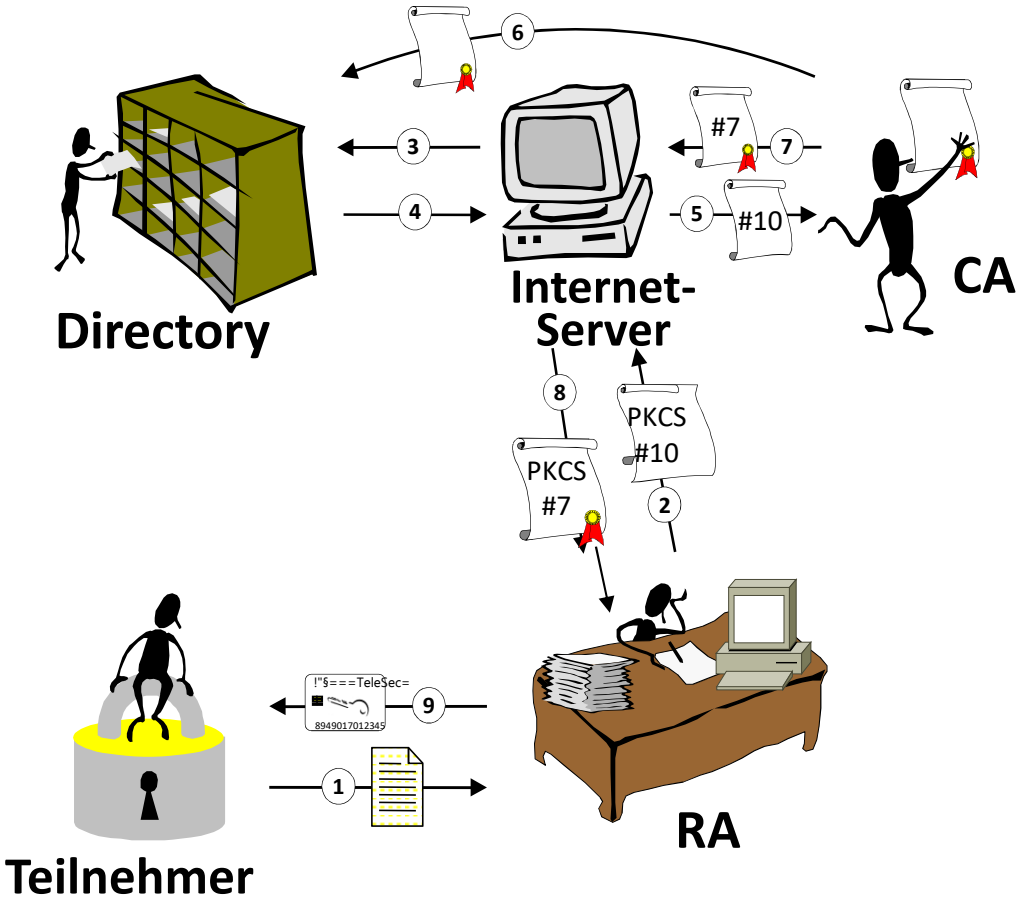
Was ist eine PKI?

- Eine Infrastruktur für die Verwaltung von Digitalen Zertifikaten mittels eines **Trust-Centers** aufgrund des Vertrauensmodells „Hierarchical Trust“
- Aufgaben eines **Trust Centers**
  - Registrierung von PKI-Teilnehmern und Erzeugung / Erneuerung von Digitalen Zertifikaten als Zertifizierungsstelle (CA)
  - Veröffentlichung von Digitalen Zertifikaten (Verzeichnisdienst)
  - Rücknahme / Sperrung von Digitalen Zertifikaten
  - Überprüfung von Digitalen Zertifikaten (optional)
  - Einhaltung und Durchsetzung von organisatorischen Richtlinien („Policy“)

PKI-Komponenten

- **Innerhalb** eines **Trust Centers**:
  - Zertifizierungsstelle (CA)
    - Erzeugt und verwaltet Digitale Zertifikate
    - Schlüssel werden dezentral erzeugt
  - Registrierungsstelle (RA)
    - Anmeldestelle, überprüft die Identität des Teilnehmers
  - Verzeichnisdienst (Directory Service)
    - Veröffentlicht Zertifikate (meist: LDAP / X.500 – basiert)
- **Außerhalb** eines **Trust Centers**:
  - ggf. lokale Registrierungsstelle (RA)
  - Endeinheit („End entity“)
    - Realisiert eine PKI-Anwendung (PC, Web-Server, Smartphone, ..)
  - Personal Security Environment (PSE)
    - Speicherort des privaten Schlüssels (Festplatte, Chip-Karte,..)

Beispiel: Registrierung und Zertifikatsausgabe („Enrollment“) mit lokaler RA



- Teilnehmer stellt Papierantrag
- RA überprüft Daten, erstellt Schlüsselpaar und beantragt Zertifikat
- Server überprüft, ob Eintrag im Directory vorhanden
- CA erstellt Zertifikat und sendet es zur RA und zum Directory
- RA stellt Teilnehmer PSE (Chipkarte, PKCS#12-File) aus

Quelle: TESTA-CA

Zertifikatsklassen

- Die meisten Trust Center bieten Zertifikate unterschiedlicher Vertrauenswürdigkeit an
  → Abhängig vom Registrierungsverfahren
- Typische Klasseneinteilung
  - Class 0: für Testzwecke → keinerlei Identitätsprüfung!
  - Class 1: Prüfung durch Erreichbarkeitstest der Mailadresse
  - Class 2: Prüfung einer Ausweiskopie und Unterschrift
  - Class 3: Prüfung durch persönliches Erscheinen

Sperrung von Zertifikaten

- Mögliche Gründe für eine Sperrung:
  - Schlüsselkompromittierung
  - CA-Kompromittierung
  - Änderung des Zertifikatsinhalts
  - Kündigung durch Teilnehmer
  - Ablauf der Gültigkeit
- *Unterscheidung nötig:*
  - *Authentifikation: Ist das Zertifikat jetzt gültig?*
  - *Elektronische Unterschrift: War das Zertifikat zum Zeitpunkt der Signierung gültig?*

Technische Realisierung von Zertifikats-Sperrungen

- Veröffentlichung von Sperrlisten
  („Certificate Revokation Lists“ **CRL**)
  mit ungültigen Zertifikats-Seriennummern
  - komplette Sperrlisten
  - Teilsperrlisten an **CDP**s („CRL Distribution Points“)
  - Liste mit Neuzugängen seit letzter Aktualisierung („Delta-CRL“)

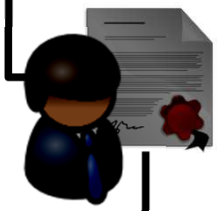
- Online-Sperrprüfung
  - **OCSP** („Online Certificate Status Protocol“) [RFC2560]
  - **SCVP** („Simple Certificate Validation Protocol“)
    - Ermöglicht zusätzlich die komplette Überprüfung eines Zertifikats

Zusammenarbeit von Trustcentern / CAs

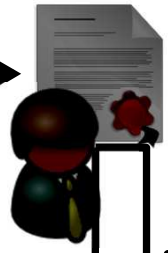
- **Web-Modell: Unabhängige Koexistenz**
  - Jeder Benutzer muss die öffentlichen Schlüssel (Zertifikate) aller CAs kennen (und ihnen vertrauen), um Benutzer-Zertifikate prüfen zu können
- **Cross-Zertifizierung**
  - Gegenseitige Zertifizierung von CAs
- **Zertifikats-Hierarchien**
  - Eine **„Root"-CA** zertifiziert „Sub-CAs" in mehreren Stufen → Baumstruktur → „Zertifizierungsketten"
  - Eine übergeordnete CA heißt Policy-CA (**PCA**), wenn sie verbindliche Richtlinien (eine „Policy") vorschreibt!

Beispiel einer Zertifikatshierarchie

Deutsche Telekom Root CA2
(selbstsigniertes Root-Zertifikat)
bereits durch Hersteller in Browsern installiert



DFN-Verein PCA Global – G01
durch Benutzer oder Roll-Out installiert



HAW Hamburg CA – G02
CA für jede teilnehmende Einrichtung

Web-Server Zertifikat
z.B.: **`www.haw-hamburg.de`**

Benutzer-Zertifikat
z.B.: **`Martin Hübner`**

Prüfen von Zertifikaten durch Endeinheiten

- Szenario: Ein Zertifikat Z_S eines Senders, ausgestellt von CA-0, wird empfangen *(z.B. das Zertifikat eines Web-Servers beim Aufbau einer SSL/TLS-Verbindung)*
- Ziel: Überprüfung der Vertrauenswürdigkeit des Zertifikats Z_S
- Mögliche Prüfschritte:
  - Im Zertifikatsspeicher der vertrauenswürdigen CAs der Endeinheit *(z.B. PC)* ist ein gültiges Zertifikat $Z_{CA\text{-}0}$ der ausstellenden CA CA-0 vorhanden:
    - Prüfen der Sperrliste (CRL) oder Online-Prüfung (OCSP) der Zertifikate Z_S und $Z_{CA\text{-}0}$ bzgl. Sperrung
    - Wenn nicht gesperrt und Signatur von Z_S korrekt verifiziert: OK!

Prüfen von Zertifikaten (2)

- Mögliche Prüfschritte (Fortsetzung):
  - CA-0 ist unbekannt, kein Zertifikat Z_{CA-0} vorhanden
    - *Problem: Kein vertrauenswürdiger öffentlicher Schlüssel der CA zur Überprüfung des Endzertifikats Z_S verfügbar!*

    1. Möglichkeit: Manuelle Prüfung des Zertifikats Z_S mit persönlicher Entscheidung über Annahme / Ablehnung
    2. Möglichkeit:
       - Ermittlung des CA-Zertifikats Z_{CA-0} über Verzeichnisdienstanfrage (z.B. LDAP)
       - Wenn Z_{CA-0} durch CA-1 zertifiziert wurde → Prüfung von Z_{CA-1} (rekursives Durchlaufen der Zertifizierungskette, bis bekanntes Zertifikat oder Wurzelzertifikat gefunden wurde → gültiger „Zertifizierungspfad“!)

Kapitel 4
Public-Key-Infrastrukturen (PKI)

1. Vertrauensmodelle
2. Zertifikate
3. PKI-Organisation
4. PKI-Anwendungen

PKI-Standards

- **PKIX** (*„Public Key Infrastructure X.509“*) [IETF] **RFC 5280**
  - Basis: Hierarchical Trust, X.509v3-Zertifikate
  - Umfassende PKI-Regelungen
- **Common PKI** *(ehem. ISIS „Industrial Signature Interoperability Specification“*) [BSI, deutsche Trust Center]
  - Regelung spezifischer Anforderungen des deutschen Signaturgesetzes
  - Modifikation/Erweiterung des PKIX bzgl. Signaturen

- **OpenPGP** (*„Pretty Good Privacy“*) [IETF] **RFC 4880**
  - Basis: Web of Trust, PGP-Formate
  - Kommerziell nicht relevant

Beispiele für PKI-Anwendungen I

- Verschlüsseln einer Webserver-Verbindung (z.B. für E-Commerce)
  - Server-Authentifizierung und Verschlüsselung mit SSL/TLS
- Virtuelle Private Netze (VPN)
  - Betrieb eines virtuellen privaten Netzes über das Internet mit IPSec
- Benutzerauthentifikation
  - Digitale Signatur statt Passwort (mit privatem Schlüssel auf Chipkarte)
- E-Mail-Verschlüsselung und –Signierung
  - PEM (veraltet), S/MIME
- Datei-Verschlüsselung
  - Festplatten-Verschlüsselung, Cloud-Dateiablage
- Code-Signing
  - Signieren von mobilem Code (z.B. JAVA-Applets) / Treibern

Beispiele für PKI-Anwendungen II

- Digital Rights Management („DRM“)
  - Verschlüsseln von Multimedia-Inhalten
- Elektronischer Rechtsverkehr
  - Digitale Signatur als elektronische Unterschrift
- Online-Banking
  - Authentifizierung und Verschlüsselung
- Online-Bezahlsysteme
  - PayPal, T-Pay, …
- Elektronische Ausweise
  - E-Reisepass, E-Personalausweis, E-Gesundheitskarte, …
- …

Rechtliche Rahmenbedingungen

- Deutsches Signaturgesetz von 1997 / 2001
  - Rechtliche Gleichstellung von eigenhändiger Unterschrift und Digitaler Signatur gegeben, falls ein qualifiziertes Zertifikat für die elektronische Signatur *(„qualifizierte Signatur“)* vorliegt
  - Definition der Anforderungen an eine qualifizierte Signatur
- eIDAS-Verordnung der EU *(electronic IDentification, Authentication and trust Services)* ab 2016
  - Definiert Rahmenbedingungen für elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen
  - Unterscheidet *qualifizierte Signatur* (für natürliche Person) und *qualifiziertes Siegel* (für juristische Person)
- Vertrauensdienstegesetz *(löste am 29.07.2017 das Signaturgesetz ab)*
  - Kein eigenständiges Gesetz → Basiert auf der eIDAS-Verordnung
  - Regelt die eIDAS-Durchführung in Deutschland, insbesondere die Zuständigkeiten der Bundesnetzagentur und des BSI

TrustCenter (Root-CAs)

- Kommerziell
  - D-Trust (Bundesdruckerei)
  - T-Systems (Deutsche Telekom)
  - DigiCert (US-Firma)
  - Verisign (US-Firma)
  - Comodo *(kostenlose Class 1-Zertifikate unter https://www.comodo.com/home/email-security/free-email-certificate.php)*
  - *...*
- Nicht-kommerziell
  - Let's Encrypt *(https://letsencrypt.org – mit automatischer Zertifikatsverwaltung für Webserver)*
  - CACERT *(www.cacert.org - keine Browser-Unterstützung!)*
  - OpenSSL *(https://www.openssl.org - Schlüssel und Zertifikate selbst erzeugen!)*

Root-CA-Zertifikate werden bei der Browser-Installation i.d.R. mit installiert
→ Server-Authentifikation bei TLS durch Zertifikatsprüfung möglich!

Ende des 4. Kapitels:
Was haben wir geschafft?

Public-Key-Infrastrukturen (PKI)

1. Vertrauensmodelle
2. Zertifikate
3. PKI-Organisation
4. PKI-Anwendungen