Informationssicherheit

4. Verschlüsselung

Prof. Dr. Christoph Skornia
christoph.skornia@oth-regensburg.de

Grundlagen

- Ziele von Verschlüsselung:
  - Vertraulichkeit
  - Integrität
  - Authentizität
  - Verbindlichkeit
- Begriffe:
  - Kryptographie

    Lehre von Methoden zur Ver- und Entschlüsselung
  - Kryptoanalyse

    Wissenschaft von Methoden zur Entschlüsselung
  - Kryptologie

    Lehre von Ver- und Entschlüsselung

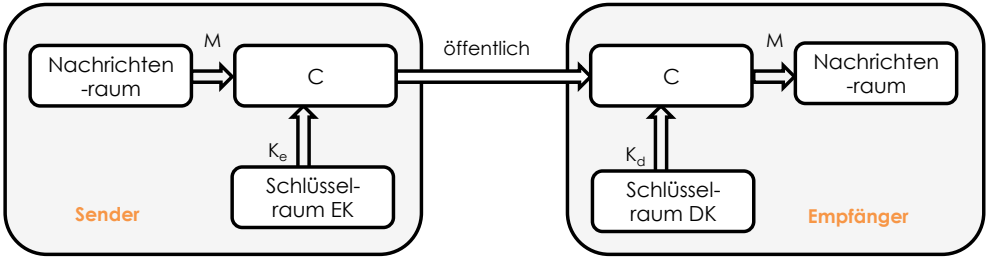
Kryptosystem

Definition: Ein kryptographisches Verfahren ist gegeben durch ein Tupel (M, C, EK, DK, K_E, K_D) mit:

1. M: Menge von Klartextnachrichten über dem Alphabet A_1
2. C: Menge von Kryptonachrichten C über dem Aphabet A_2
3. EK: Menge von Verschlüsselungs-Schlüsseln
4. DK: Menge von Entschlüsselungsschlüsseln und der Abbildung $f : EK \longrightarrow DK$ mit $K_D = f(K_E)$
5. dem injektiven Verschlüsselungsverfahren: $E : A_1^* \times EK \longrightarrow A_2^*$
6. dem Entschlüsselungsverfahren: $D : A_2^* \times DK \longrightarrow A_1^*$ mit $\forall M \in A_1^* : D(E(M, K_E), K_D) = M$

Generell gibt es zwei Klassen von kryptographischen Verfahren:

1. Symmetrische Verfahren: $K_e = K_d$, $f = \mathrm{id}$
2. Asymmetrische Verfahren: $K_e \neq K_d$



Kryptographie

Anforderungen an kryptographische Verfahren:

- Sicherheit darf nicht von Geheimhaltung der Ver- und Entschlüsselungsfunktionen abhängen! Häufiger Verstoß dagegen: Security by Obscurity
- Geheimer Schlüssel darf mit der Kenntnis über die verwendeten Verfahren nicht praktikabel berechenbar sein!
- Stärke des Verfahren darf nur von der Güte des geheimen Schlüssels abhängen! Kerckhoffs-Prinzip

Kryptographie

Anforderungen an kryptographische Verfahren:

- Sicherheit darf nicht von Geheimhaltung der Ver- und Entschlüsselungsfunktionen abhängen! Häufiger Verstoß dagegen: Security by Obscurity
- Geheimer Schlüssel darf mit der Kenntnis über die verwendeten Verfahren nicht praktikabel berechenbar sein!
- Stärke des Verfahren darf nur von der Güte des geheimen Schlüssels abhängen! Kerckhoffs-Prinzip

Bemerkung: Berechnungsaufwand zum Schlüsselknacken ist abhängig von

- der aktuellen Rechner-Technologie (CPU),
- Möglichkeiten zur dezentralen Berechnung (Internet, etc.)
- neuen Rechner-Architekturen, z.B. Quantencomputer?

Konsequenzen:

- Verfahren muss gut konzipiert werden, EK muss sehr groß sein
- Ausprobieren aller Schlüssel (brute force) soll nicht mit praktikablem Aufwand möglich sein (exhaustive Search)
- Beispiel: 56-Bit Schlüssel (u.a. DES): Schlüsselraum $|EK| = 2^{56}$
  - 1998 Deep-Crack-Supercomputer: Kosten ca. 250.000$

    Knacken eines DES-Schlüssels in 56 Stunden!
  - 2006: COPACOBANA (http://www.copacobana.org/):

    Kosten $<$ 10.000 $, durchschnitt. 7 Tage zum Knacken von DES
- Anforderung: (u.a. von Bundesnetzagentur)
  - symmetrische Verfahren: Schlüssel ≥ 128 Bit
  - asymmetrische Verfahren: Schlüssel ≥ 2048 Bit

Kryptographie

Symmetrische Verfahren:

- Ver- und Entschlüsselungs-Schlüssel sind gleich, oder leicht auseinander ableitbar, $K_d = f(K_e), f$ einfach berechenbar
- Nutzung eines gemeinsamen, geheimen Schlüssels (**Secret-Key**)
- Bekannte Repräsentanten:
  - ROT (oder auch Cäsar-Code) (**Substitutionschiffre**)
  - Skytale (**Transpositionschiffre**)
  - DES (Data Encryption Standard)(unsicher): noch immer weit verbreitet
  - AES (Krypto-Standard),
  - RC4(unsicher), A5, IDEA

Kryptographie

Symmetrische Verfahren:

- Ver- und Entschlüsselungs-Schlüssel sind gleich, oder leicht auseinander ableitbar, $K_d = f(K_e), f$ einfach berechenbar
- Nutzung eines gemeinsamen, geheimen Schlüssels (**Secret-Key**)
- Bekannte Repräsentanten:
  - ROT (oder auch Cäsar-Code) (**Substitutionschiffre**)
  - Skytale (**Transpositionschiffre**)
  - DES (Data Encryption Standard)(unsicher): noch immer weit verbreitet
  - AES (Krypto-Standard),
  - RC4(unsicher), A5, IDEA
- Problem: Sicherer Austausch des gemeinsamen Schlüssels K!

Asymmetrische Verfahren:

Ein Schlüsselpaar (K_e, K_d) pro Kommunikationspartner A

Basis: Einweg-Funktion $f : X \longrightarrow Y$

- Eigenschaften von Einweg-Funktionen:
  1. $\forall x \in X$ gilt: $f(x)$ ist effizient berechenbar; und
  2. für fast alle $y \in Y$ gilt, dass es nicht effizient möglich ist, $f^{-1}(y)$ zu berechnen
- Existenz von Einwegfunktionen bis heute aber unbewiesen
- Bewiesen ist: Falls eine Einwegfunktion existiert, dann gilt $P \neq NP$.

Gute Kandidaten:

1. Multiplikation von Primzahlen $\longleftrightarrow$ Primfaktorisierung
   gegeben $n = p \cdot q$ mit p und q prim, gesucht p und q
2. Potenzierung im $\mathbb{Z}/n\mathbb{Z} \longleftrightarrow$ Diskreter Logarithmus
   gegeben p prim und $g, y \leq p$, gesucht k mit $y = g^k \bmod p$

Gute Kandidaten:

1. Multiplikation von Primzahlen $\longleftrightarrow$ Primfaktorisierung
   gegeben $n = p \cdot q$ mit p und q prim, gesucht p und q
2. Potenzierung im $\mathbb{Z}/n\mathbb{Z} \longleftrightarrow$ Diskreter Logarithmus
   gegeben p prim und $g, y \leq p$, gesucht k mit $y = g^k \bmod p$

Für Verschlüsselung nötig: Einweg-Funktion mit Falltür (engl. **trapdoor function**)

- d.h. mit Zusatzinformation (Schlüssel) sind Urbilder effizient berechenbar
- **Bsp.:**
  - gegeben $n = p \cdot q$, und Funktion f mit $f(x) = x^2 \bmod n$
  - Invertierung von f schwierig ohne Kenntnis von p und q
  - Kenntnis von p und q ist „Falltür", mit p, q ist Invertierung effizient berechenbar

Allgemeine Eigenschaften asymmetrischer Verfahren

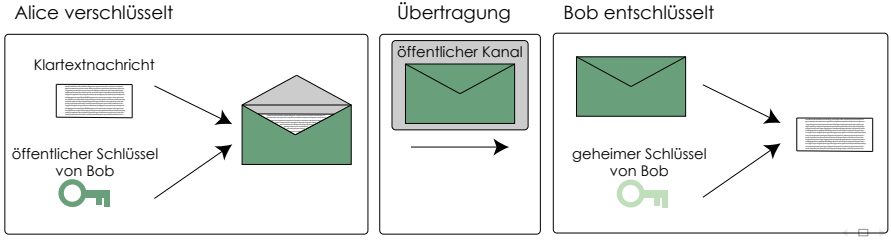
- Die Schlüsselpaare (K_E, K_D) müssen folgende Eigenschaft erfüllen:

$$\forall M \in A_1^* : D(E(M, K_E), K_D) = M$$

- Schlüsselpaare müssen leicht erzeugbar sein
- Ver- und Entschlüsselungen sind effizient durchführbar
- K_D ist aus K_E nicht mit vertretbarem Aufwand berechenbar

- Beispiele für asymmetrische Verfahren:
  - RSA („Quasi-Standard")
  - ElGamal-Verfahren

Asymmetrische Verschlüsselung

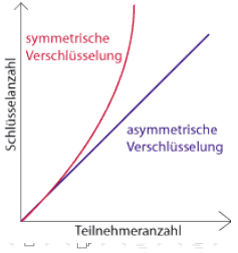


Vergleich:

	symmetrisch	**asymmetrisch**
Rechenzeit:	schnell	ca. Faktor 1000 langsamer
Schlüsselverteilung:	sicherer Kanal nötig	kein sicherer Kanal nötig
Schlüsselsicherheit:	jeder muss den Schlüssel kennen	private Key braucht nur der Eigentümer kennen
Schlüsselanzahl:	quadratisch mit der Anzahl der Partner	linear mit der Anzahl der Partner
Sicherheit	kann gesichert werden	beruht auf mathematisch unbewiesenen Annahmen

Lösungsansatz: Hybride Verschlüsselung:

- Nutzdaten werden symmetrisch verschlüsselt
- dafür nötige Schlüssel werden asymmetrisch verschlüsselt und ausgetauscht



Schlüsselaustausch:

- Hybride Verfahren funktionieren nur, wenn die Kommunikationspartner beide öffentliche und private Schlüssel besitzen (z.B. bei Onlinebanking nicht gegeben)
- ideal wäre es einen geheimen Schlüssel vereinbaren zu können ohne diesen austauschen zu müssen
- Idee (diskreter Logarithmus):
  - wähle eine große Primzahl q (jedem bekannt)
  - wähle einen Wert g, der eine primitive Wurzel von q in der zyklischen Gruppe $\mathbb{Z}/q\mathbb{Z}$ ist

Diffie-Hellman Schlüsselaustausch:

Sei p prim und g eine Primitivwurzel von p

Allice				Bob		
Geheim	Öffentlich	berechnet	sendet	berechnet	Öffentlich	Geheim
$a \in \{1...p-1\}$	p, g		$p, g \rightarrow$			$b \in \{1...p-1\}$
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
$a, \boldsymbol{s}$	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	$b, \boldsymbol{s}$

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = g^{b \cdot a} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p = s$$

Diffie-Hellman Schlüsselaustausch:

Sei p prim und g eine Primitivwurzel von p

Allice				Bob		
Geheim	Öffentlich	berechnet	sendet	berechnet	Öffentlich	Geheim
$a \in \{1...p-1\}$	p, g		$p, g \rightarrow$			$b \in \{1...p-1\}$
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
$a, \boldsymbol{s}$	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	$b, \boldsymbol{s}$

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = g^{b \cdot a} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p = s$$

Beispiel: 1 Alice und Bob einigen sich auf $p = 13$ und $g = 2$

Diffie-Hellman Schlüsselaustausch:

Sei p prim und g eine Primitivwurzel von p

Allice				Bob		
Geheim	Öffentlich	berechnet	sendet	berechnet	Öffentlich	Geheim
$a \in \{1...p-1\}$	p, g		$p, g \rightarrow$			$b \in \{1...p-1\}$
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
$a, \boldsymbol{s}$	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	$b, \boldsymbol{s}$

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = g^{b \cdot a} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p = s$$

Beispiel:

1. Alice und Bob einigen sich auf $p = 13$ und $g = 2$
2. Alice wählt $a = 5$ und Bob wählt $b = 7$

Diffie-Hellman Schlüsselaustausch:

Sei p prim und g eine Primitivwurzel von p

Allice				Bob		
Geheim	Öffentlich	berechnet	sendet	berechnet	Öffentlich	Geheim
$a \in \{1...p-1\}$	p, g		$p, g \rightarrow$			$b \in \{1...p-1\}$
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
$a, \boldsymbol{s}$	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	$b, \boldsymbol{s}$

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = g^{b \cdot a} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p = s$$

Beispiel:

1. Alice und Bob einigen sich auf $p = 13$ und $g = 2$
2. Alice wählt $a = 5$ und Bob wählt $b = 7$
3. Alice berechnet $A = 6$ und Bob $B = 11$

Diffie-Hellman Schlüsselaustausch:

Sei p prim und g eine Primitivwurzel von p

Allice				Bob		
Geheim	Öffentlich	berechnet	sendet	berechnet	Öffentlich	Geheim
$a \in \{1...p-1\}$	p, g		$p, g \rightarrow$			$b \in \{1...p-1\}$
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
$a, \boldsymbol{s}$	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	$b, \boldsymbol{s}$

$$s = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = g^{b \cdot a} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p = s$$

Beispiel:

1. Alice und Bob einigen sich auf $p = 13$ und $g = 2$
2. Alice wählt $a = 5$ und Bob wählt $b = 7$
3. Alice berechnet $A = 6$ und Bob $B = 11$
4. Alice und Bob berechnen $s = 7$

Problem von Diffie-Hellman:

Kein Schutz vor „Man-in-the-Middle"-Angriffen (MitM)

Allice	$\longleftrightarrow$	Mallory	$\longleftrightarrow$	Bob
$\rightarrow$	$A = g^a \bmod p$			
		$\rightarrow$	$Z = g^z \bmod p$	
			$B = g^b \bmod p$	$\leftarrow$
	$Z = g^z \bmod p$	$\leftarrow$		

$\Longrightarrow$ Authentizität der ausgetauschten Information muss sichergestellt werden

Exkurs Hashing

Eine **Hashfunktion** ist eine Abbildung: $H : X^* \longrightarrow X^n$

klar ist: H kann nicht injektiv sein.

- **Ziel:** Integrität von Daten überprüfen
- **Idee:**
  - Austausch von Dokument und Hashwert über unterschiedliche Kanäle
  - Bei Veränderung von Daten muss sich auch ein anderer Hashwert ergeben
  - mögliches Problem: Kollisionen
    (d.h. zwei unterschiedliche Werte ergeben den gleichen Hash-Wert).
- **Beispiele:**
  - DES-CBC: 64-Bit Hashwert, der letzte Block dient als Hash
  - MD4, MD5 mit 128-Bit Hash
  - SHA-1 (Secure Hash-Algorithm) 160-Bit Hash
  - **besser: SHA-2 (SHA-224, SHA-256, SHA-384 und SHA-512) SHA-3 (ebenso 224 - 512 Bit)**

Exkurs Hashing

Anforderungen an eine Hashfunktion:

- $\mathrm{H}(M) = h$ muss einfach zu berechnen sein
- Für ein gegebenes h ist es nicht effizient möglich M mit der Eigenschaft $\mathrm{H}(M) = h$ zu bestimmen
- Für eine gegebenes M ist es nicht effizient möglich ein $M' \neq M$ mit der Eigenschaft $\mathrm{H}(M) = \mathrm{H}(M')$ zu bestimmen
- Sind diese Eigenschaften gegeben, so spricht man von einer *kryptographischen Hashfunktion*

Verschlüsselung und Authentizität

- Verschlüsselung ist ohne Überprüfung der Authentizität in vielen Fällen nicht viel wert (siehe DH)
- Gesucht: Schutz vor *Known-Ciphertext-Attacks*
- **Idee:** Hashfunktionen mit Schlüssel (Message Authentication Code, MAC): $\mathrm{MAC} : A^* \times EK \to A^n$:
  1. Sender A berechnet $h = \mathrm{MAC}(M, K_{AB})$ und sendet h an B
  2. Empfänger B empfängt M‘, h und überprüft h
- Einfachste Variante: $\mathrm{MAC}(M, K_{AB}) = \mathrm{H}(K_{AB}||M)$ **unsicher!!!**
- Seit 1997 üblich: $\mathrm{HMAC}_K(M) = \mathrm{H}\Big((K \oplus opad) \;||\; \mathrm{H}\big((K \oplus ipad) \;||\; M\big)\Big)$
  mit $opad = \underbrace{0\mathrm{x}5\mathrm{C} \ldots 0\mathrm{x}5\mathrm{C}}_{B\text{-mal}}$ und $ipad = \underbrace{0\mathrm{x}36 \ldots 0\mathrm{x}36}_{B\text{-mal}}$

Digitale Signatur

- Eine MAC erlaubt eine verschlüsselte Nachricht sicher dem Besitzer eines Schlüssels zuzuordnen, wenn man diesen Schlüssel selbst besitzt.
- Passt gut für symmetrische Verschlüsselungsverfahren, wo beide Partner den Schlüsse besitzen.

Zusätzlich nötig:

- Einem Besitzer eines privaten Schlüssels eine Nachricht sicher zuzuordnen, wenn man selbst nur den öffentlichen Schlüssel besitzt.

Digitale Signatur

- Eine MAC erlaubt eine verschlüsselte Nachricht sicher dem Besitzer eines Schlüssels zuzuordnen, wenn man diesen Schlüssel selbst besitzt.
- Passt gut für symmetrische Verschlüsselungsverfahren, wo beide Partner den Schlüsse besitzen.

Zusätzlich nötig:

- Einem Besitzer eines privaten Schlüssels eine Nachricht sicher zuzuordnen, wenn man selbst nur den öffentlichen Schlüssel besitzt.

> **Idee:** Verschlüssele den Hash einer Nachricht mit dem privaten Schlüssel eines asymmetrischen Verfahrens $\Rightarrow$ Jeder kann überprüfen, dass der Sender der Nachricht einen bestimmten privaten Schlüssel besitzt.

Digitale Signatur

Digitale Signatur:

Sei H eine kryptographische Hashfunktion sowie E und D die Ver- bzw. Entschlüsselungsfunktion eines asymmetrischen Kryptosystems. M sei eine Nachricht und (K_E^A, K_D^A) das Schlüsselpaar eines Benutzers A bzgl. des o.g. Systems.

Dann heisst:

$$S = \mathrm{E}\left(\mathrm{H}(M), K_D^A\right)$$

eine *Digitale Signatur* von M des Users A

Digitale Signatur

Digitale Signatur:

Sei H eine kryptographische Hashfunktion sowie E und D die Ver- bzw. Entschlüsselungsfunktion eines asymmetrischen Kryptosystems. M sei eine Nachricht und (K_E^A, K_D^A) das Schlüsselpaar eines Benutzers A bzgl. des o.g. Systems.

Dann heisst:

$$S = \mathrm{E}\left(\mathrm{H}(M), K_D^A\right)$$

eine *Digitale Signatur* von M des Users A

Anmerkungen:

- $\mathrm{D}\left(S, K_E^A\right) = \mathrm{H}(M)$
- Nicht jede digitale Signatur beruht auf diesem Prinzip, aber die grundsätzliche Idee ist überall ähnlich.

Digitale Signatur

Was fehlt noch?

- Zuordnung von öffentlichen Schlüsseln zu Personen
- Validierung dieser Zuordnung auf Anfrage

Digitale Signatur

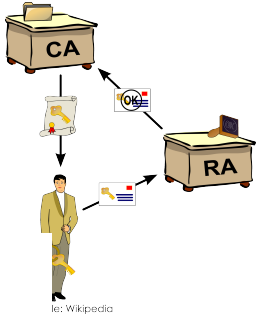
Was fehlt noch?

- Zuordnung von öffentlichen Schlüsseln zu Personen
- Validierung dieser Zuordnung auf Anfrage
- Also eine **P***ublic-***K***ey-***I***nfrastruktur* **(PKI)**

Digitale Signatur

Was fehlt noch?

- Zuordnung von öffentlichen Schlüsseln zu Personen
- Validierung dieser Zuordnung auf Anfrage
- Also eine **P**ublic-**K**ey-**I**nfrastruktur **(PKI)**



Ie: Wikipedia

Komponenten einer PKI:

- ***Certification Authority (CA):***
  - Stellt Zertifikate aus, signiert und veröffentlicht sie
  - Erstellt und veröffentlicht Listen von ungültigen Zertifikaten (CRLs), Certificate Revocation List
- ***Registration Authority (RA):***
  bürgt für die Verbindung zw. öffentlichem Schlüssel und Identitäten/Attributen der Zertifikatsinhaber

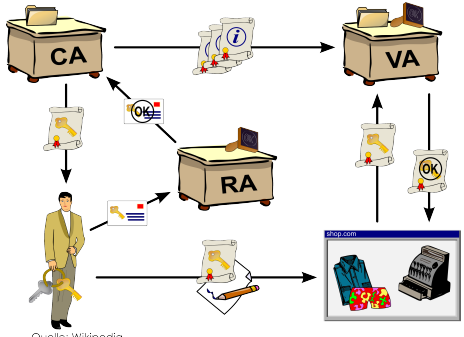
Digitale Signatur

Was fehlt noch?

- Zuordnung von öffentlichen Schlüsseln zu Personen
- Validierung dieser Zuordnung auf Anfrage
- Also eine **P***ublic-***K***ey-***I***nfrastruktur* **(PKI)**

Komponenten einer PKI:

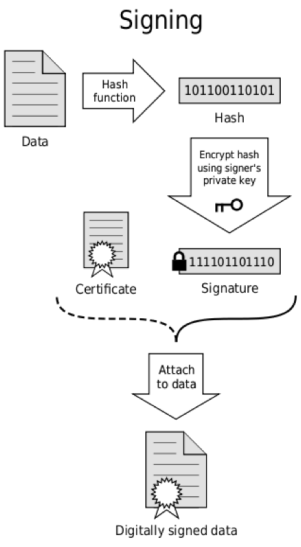
- ***Certification Authority (CA):***
  - Stellt Zertifikate aus, signiert und veröffentlicht sie
  - Erstellt und veröffentlicht Listen von ungültigen Zertifikaten (CRLs), Certificate Revocation List
- ***Registration Authority (RA):***
  bürgt für die Verbindung zw. öffentlichem Schlüssel und Identitäten/Attributen der Zertifikatsinhaber

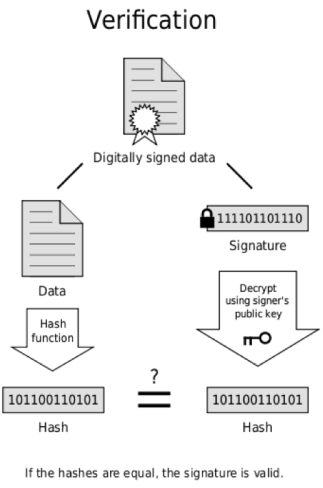


Quelle: Wikipedia

- ***Validation Authority (VA):***
  ermöglicht die Validierung der Zertifikate in Echtzeit (z.B. über CRL-Download oder Online Certificate Status Protocol (OCSP))
- optionaler ***Verzeichnisdienst:***
  Verteilung der Zertifikate und CRLs

Digitale Signatur



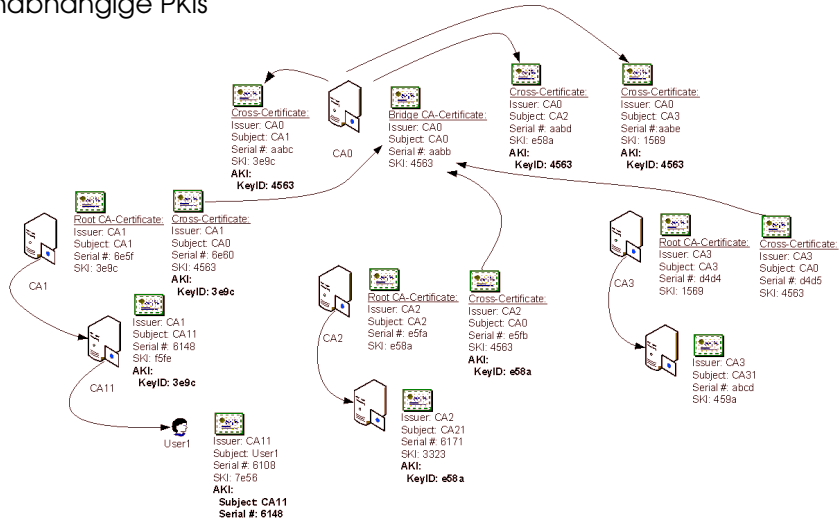


Digitale Signatur

Problem: Viele separate unabhängige PKIs

Lösung:

1. Hierarchien
   - untergeordnete CAs vertrauen den übergeordneten
   - alle Vertrauen der top-level CA (z.B. Bundesnetzagentur)
2. Cross-Zertifikate: CAs zertifizieren sich gegenseitig.



Rechtlicher Rahmen: Deutsches Signaturgesetz (2001)

Drei Arten der digitalen Signatur:

- ***einfache Signatur:***
  - keine speziellen Anforderungen an Zertifikate, Erzeugung etc.
  - nicht der Schriftform gleichgestellt
  - d.h. potentiell Geschädigter muss den Schaden selber nachweisen
- ***fortgeschrittene Signatur***
  - Anforderungen an Zertifikatsaussteller, Signierumgebung und die Verknüpfung von Signatur mit Dateien
  - Signaturanbieter haftet für die Richtigkeit und Vollständigkeit der Zertifikatsangaben
- ***qualifizierte Signatur***
  - fortgescrhrittene Signatur mit qualifiziertem Zertifikat
  - sicherer Signaturerstellungseinheit (CC EAL4)
  - rechtliche Gleichstellung mit eigenhändiger Unterschrift

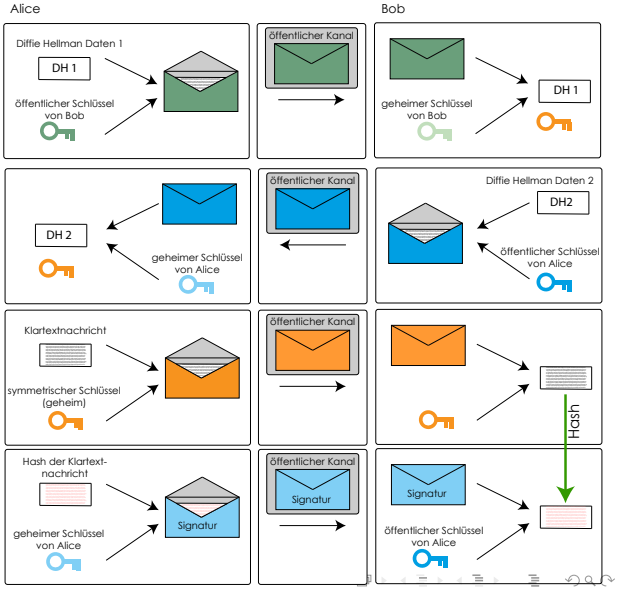
Hybride Verfahren

Und jetzt alle zusammen:

Hybride Verfahren:

- kombinieren die Vorteile der Einzelverfahren
  - So schnell wie symmetrische Verfahren
  - Kein sicherer Kanal für den Schlüsselaustausch nötig
  - Perfect Forward Secrecy
- stellen den aktuellen De-facto-Standard für Verschlüsselung im Internet dar

Anmerkung:
Die hier vorgestellten Verfahren sind schematisch korrekt in der konkreten Implementierung gibt jedoch eine Reihe von Anpassungen.



Kryptographie: Fazit

Einsatzbereiche der Kryptographie:

- **Vertraulichkeit**
  - Schlüsselaustausch
  - Verschlüsselung
- **Integrität**
  - Hashfunktionen
  - MAC
- **Verbindlichkeit**
  - Digitale Signatur
  - Zertifikate
  - PKI
- **Authentizität**
  - MAC
  - Signaturen

Danke für Ihr Interesse

Fortsetzung folgt