

---

# **Security**

## **- LV 4121 und 4241 -**

**Asymmetrische Kryptosysteme**

- El-Gamal Kryptosystem
- Asymmetrische kryptographische Verfahren sowie RSA-Algorithmus
- Das Rabin-Verschlüsselungsverfahren
- El-Gamal-Signaturverfahren
- Das Drei-Wege-protokoll nach X.509

## Kap. 6: Asymmetrische Kryptosysteme

### Teil 1: Asymmetrische Verschlüsselung

- Das ElGamal-Kryptosystem
- Wahl der Systemparameter
- Ver- und Entschlüsselungsalgorithmus
- Sicherheit des Verfahrens

- 
- Das ElGamal-Verschlüsselungsverfahren wurde 1985 von dem Kryptologen **Taher Elgamal** entwickelt.
  - Es zählt zu den Public-Key-Verschlüsselungsverfahren und verwendet für jeden **Teilnehmer T** einen **öffentlichen Schlüssel  $PK_T$**  sowie einen **geheimen Schlüssel  $SK_T$** .
  - Der öffentliche Schlüssel kann veröffentlicht werden und dient der Verschlüsselung, während der geheime Schlüssel (nur dem Empfänger der Nachricht bekannt) bei der Entschlüsselung angewandt wird.
  - Im folgenden wird davon ausgegangen, dass ein **Sender A** eine **Nachricht m** an einen **Empfänger B** senden möchte.
  - Sender A und Empfänger B verfügen jeweils über ein **Schlüsselpaar**, welches einmalig erzeugt werden muss.
-

### Systemparameter:

1. eine **Primzahl**  $p$ .
2. eine **multiplikative Einheitengruppe (Körper)**  $\mathbf{G}$  über  $\mathbf{Z}_p^*$  mit:

$$\mathbf{Z}_n^* := \{a \in \mathbf{Z}_n \setminus \{0\} \mid \text{ggT}(a, n) = 1\}$$

3. einen **Erzeuger**  $g$ .

### Anmerkung:

Die Parameter  $(\mathbf{G}, g)$  werden öffentlich gemacht und die Sicherheit des Kryptosystems verlangt eine **möglichst große Ordnung** der Gruppe  $\mathbf{G}$ . Daher wird  $p$  so gewählt, dass

$$p - 1 = 2q,$$

wobei  $q$  wiederum eine Primzahl ist (vgl. Sophie-Germain-Primzahlen).

### Schlüsselerzeugung:

Das Schlüsselpaar  $(PK_B, SK_B)$  des **Empfängers B** wird folgendermaßen erzeugt:

1. **B** wählt zufällig eine Zahl  $b \in \{1, \dots, p-1\}$  mit  $\text{ggT}(b, p) = 1$ .

2. **B** berechnet das Gruppenelement:

$$B = g^b \in G(\mathbb{Z}_p^*)$$

3. **B** setzt den Entschlüsselungsschlüssel (**geheim**)  $SK_B$ :

$$SK_B := b$$

4. **B** setzt den Verschlüsselungsschlüssel (**öffentlich**)  $PK_B$ :

$$PK_B := B$$

### Verschlüsselungsvorschrift:

Um die **Nachricht m**  $\in \mathbf{G}(\mathbf{Z}_p^*)$  zu versenden, verfährt der **Sender A** folgendermaßen:

1. **A** wählt zufällig eine Zahl  $r \in \{1, \dots, p-1\}$  mit  $\text{ggT}(r, p) = 1$ .

2. **A** berechnet das Gruppenelement:

$$R = g^r \in \mathbf{G}(\mathbf{Z}_p^*)$$

3. **A** berechnet das Chiffre (mit dem öffentlichen Schlüssel des Empfängers B):

$$c := \text{PK}_B^r \cdot \mathbf{m} \in \mathbf{G}(\mathbf{Z}_p^*)$$

4. **A** versendet:

$$(R, c)$$

als verschlüsselte **Nachricht m** an den Empfänger B.

## Entschlüsselungsvorschrift:

Um die verschlüsselte Nachricht  $(R, c)$  zu entschlüsseln, verfährt der **Empfänger B** folgendermaßen:

1. **B** berechnet das Gruppenelement (mit seinem **geheimen** Schlüssel  $SK_B$ ):

$$R^{-SK_B} \cdot c \equiv R^{p-1-SK_B} \cdot c \in G(\mathbb{Z}_p^*)$$

bzw. unter Verwendung des Satzes von Fermat<sup>1)</sup>:

$$R^{p-1-SK_B} \cdot c \in G(\mathbb{Z}_p^*)$$

-----  
1)

$$a^{p-1} \bmod p = 1 ; (a \neq 0)$$



## Entschlüsselungsvorschrift (Fortsetzung):

2. **B** verwendet das zuvor berechnete Gruppenelement als **Nachricht m**, denn es gilt:

$$\begin{aligned} R^{-SK_B} \cdot c &= (g^r)^{-SK_B} \cdot PK_B^r \cdot m = (g)^{-r \cdot SK_B} \cdot (g^b)^r \cdot m \\ &= (g)^{-r \cdot b} \cdot (g^b)^r \cdot m \\ &\quad \underbrace{\hspace{1.5cm}}_{= 1} \\ &= m \end{aligned} \quad \square$$

---

### Anmerkung:

$$R := g^r ; c := PK_B^r \cdot m ; PK_B := B ; B = g^b ; SK_B := b$$

### Aufgabenstellung:

Wir betrachten das ElGamal-Verschlüsselungsverfahren über der Gruppe  $G(\mathbb{Z}_{29}^*)$  mit dem Erzeuger  $g = 2$  und verschlüsseln die Nachricht  $m = 10$  mit dem öffentlichen Schlüssel  $PK_B = 5$  des Empfängers B sowie der Zufallszahl  $r = 8$ .

#### 1. Verschlüsselung:

$$\Rightarrow p = 29 \ (\forall \text{ mod } p!)$$

- Gruppenelement  $R = g^r = 2^8 \text{ mod } 29 = 24$
- Chiffre  $c := PK_B^r \cdot m = (5^8 \cdot 10) \text{ mod } 29 = 8$
- **Sender A** versendet  $(R, c) = (24, 8)$  als **verschlüsselte** Nachricht  $m$  an den **Empfänger B**.

### 2. Entschlüsselung:

$$\Rightarrow p = 29 (\forall \text{ mod } p!)$$

Der Zusammenhang zwischen dem öffentlichen und dem geheimen Schlüssel ergibt sich beim ElGamal-Verfahren aus:

$$PK_B := B ; B = g^b ; SK_B := b \Rightarrow PK_B := g^b = g^{SK_B} \Rightarrow PK_B \equiv g^{SK_B} \text{ mod } p$$

$\Rightarrow$

- Geheimer Schlüssel aus  $5 \equiv 2^{SK_B} \text{ mod } 29 \Rightarrow \mathbf{SK_B = 22 \text{ (DL-Problem!)}}$
- Gruppenelement  $R^{p-1-SK_B} \cdot c = (24^{29-1-22} \cdot 8) \text{ mod } 29$   
 $= (24^6 \cdot 8) \text{ mod } 29 = 10$
- D. h. die gesendete Nachricht ist  $m = 10$ . □

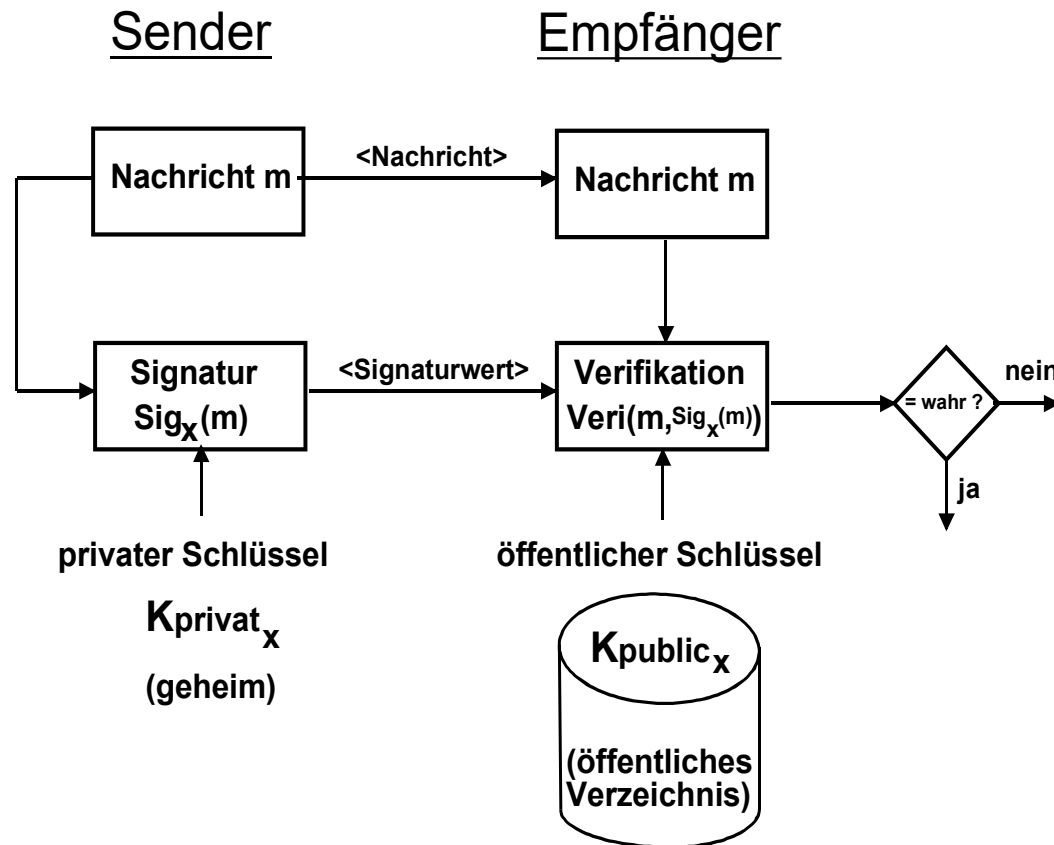
## Sicherheit des Verfahrens:

- Das ElGamal-Verschlüsselungsverfahren baut auf der Idee des Diffie-Hellman-Verfahrens auf.
- Es ist beweisbar sicher unter der Annahme, dass das sogenannte Diskret-Log-Problem in der zugrundeliegenden Gruppe schwierig ist.
- Durch schlechte Parameterwahl oder Implementierungsfehler können jedoch Spezialfälle konstruiert werden, die unsicher sind.
- In jedem Fall ist bei der Wahl der Gruppenstruktur  $G$  deren Zerfall in kleine Untergruppen zu vermeiden.
- Durch Mehrfachnutzung der gleichen Zufallszahl  $r$  sind Known-Plaintext-Angriffe möglich.

## Kap. 6: Asymmetrische Kryptosysteme

### Teil 2: Digitale Signaturen

- Ablaufskizze
- RSA-Signaturen
- Angriff auf RSA-Signatursysteme



### Basisprinzip:

- Privater (geheimer) und öffentlicher Schlüssel
- Signaturwert mit privatem Schlüssel

### Eigenschaften:

- Nachweisbarkeit
- Nicht Abstreitbarkeit
- Authentizität
- Echtheit
- Identitätsnachweis

Signiervorschrift:

$$\text{sig}(m) := h(m)^{S_k} \bmod n$$

wobei  $n = p \cdot q$ ; (Modul)

$h$  = Hashfunktion

$S_k$  = geheimer Signaturschlüssel

Schlüsselerzeugung:

$$S_k \cdot P_k \bmod \phi(n) = 1 \text{ sowie } \text{ggT}(S_k, \phi(n)) = 1$$

mit  $\phi(n) = (p - 1)(q - 1)$

Verifizierungsvorschrift:

$$h(m) \stackrel{?}{=} \text{sig}(m)^{P_k} \bmod n$$

wobei  $P_k$  = öffentlicher Verifizierschlüssel

Sicherheit:

$$S_k = P_k^{-1} \bmod \phi(n) \text{ schwierig zu berechnen!} \Rightarrow n = \{2^{768} \dots 2^{4096}\}$$

# RSA-Verfahren

## Zahlenbeispiel

	<i>Vorgang</i>	<i>Erklärung</i>
	↓	↓
Wähle:	$p = 13$ $q = 17$	Primzahl, ist geheim zu halten Primzahl, ist geheim zu halten
Berechne:	$n = 13 \cdot 17 = 221$ $\phi(n) = (13 - 1) (17 - 1) = 192$	$n = p \cdot q$ (Modul, öffentlich) $\phi(n) = (p - 1) (q - 1) ;$
Wähle (zufällig):	$P_k = 101$	(öffentl. Verifizierschlüssel)
Nun ermittle $S_k$ aus	$S_k \cdot 101 \text{ mod } 192 = 1$	$S_k \cdot P_k \text{ mod } \phi(n) = 1$ d. h. $S_k \cdot 101 = z \cdot 192 + 1$
	$\Rightarrow \underline{\underline{S_k = 173}}$ (für $z = 91$ )	(geheimer Signaturschlüssel)



*Vorgang*



*Erklärung*



Annahme:  $h(m) := 50$

zu signierende Nachricht (Text)  
 $50 = 110010_2$

signieren:

$$\text{sig}(m) = 50^{173} \bmod 221 = \underline{\underline{33}}$$

$\text{sig}(m) := h(m)^{S_k} \bmod n$   
Die berechnete Signatur ist  
 $33 = 100001_2$

verifizieren:

$$33^{101} \bmod 221 = \underline{\underline{50 := h(m)}}$$

$\text{sig}(m)^{P_k} \bmod n \stackrel{?}{=} h(m)$   
Die berechnete Signatur ist  
korrekt!

Signatur:  $\text{sig}(m) := m^{\text{Sk}} \bmod n = \mathbf{a} \underbrace{(m^{\text{Sk}} \bmod p)}_{\text{sig}_1} + \mathbf{b} \underbrace{(m^{\text{Sk}} \bmod q)}_{\text{sig}_2}$  wobei  
 $\mathbf{a} \bmod q(p) = 0(1)$   $\mathbf{b} \bmod q(p) = 1(0)$

$$(m^{\text{Sk}} \bmod n) \bmod p = \mathbf{a} \bmod p (m^{\text{Sk}} \bmod p) + \mathbf{b} \bmod p (m^{\text{Sk}} \bmod q)$$

$$m^{\text{Sk}} \bmod p = \mathbf{1} (m^{\text{Sk}} \bmod p) \quad \square$$

Analog mod q:  $\Rightarrow m^{\text{Sk}} \bmod q = \mathbf{1} (m^{\text{Sk}} \bmod q) \quad \square$

Fehlerbetrachtung (Differential Fault Analysis):

Voraussetzung: Fehler bei der Berechnung von **sig**<sub>1</sub> oder **sig**<sub>2</sub>

Annahme: Fehler bei der Berechnung **sig**<sub>1</sub>; Berechnung **sig**<sub>2</sub> sei o.k.!

**sig**<sub>err</sub> ist das Ergebnis der fehlerhaften Berechnung von sig(m)

$$\text{sig}(m) = \mathbf{a} \text{sig}_1(m) + \mathbf{b} \text{sig}_2(m) \quad \Rightarrow \text{korrekte Signatur}$$

$$\text{sig}_{\text{err}}(m) = \mathbf{a} \text{sig}_{\text{err}}(m) + \mathbf{b} \text{sig}_2(m) \quad \Rightarrow \text{fehlerhafte Signatur}$$

---

$$\text{sig}(m) - \text{sig}_{\text{err}}(m) = \mathbf{a} (\text{sig}_1(m) - \text{sig}_{\text{err}}(m)) \quad \text{Differential Fault Analysis (DFA)}$$

---

$$(\text{sig}(m) - \text{sig}_{\text{err}}(m)) = a (\text{sig}_1(m) - \text{sig}_{\text{err}}(m)) \mid \text{mod } q$$

$$(\text{sig}(m) - \text{sig}_{\text{err}}(m)) \text{ mod } q = 0, \text{ weil } a \text{ mod } q = 0$$

dann gilt:  $\text{sig}(m) \equiv \text{sig}_{\text{err}} \text{ mod } q$  aber  $\text{sig}(m) \neq \text{sig}_{\text{err}} \text{ mod } p$

also ist:  $q$  ein Teiler von  $(m - \text{sig}_{\text{err}}^{P_k})$  und  $p$  kein Teiler von  $(m - \text{sig}_{\text{err}}^{P_k})$

Ein Faktor von  $n = p \cdot q$  kann dann ermittelt werden aus:

$$\text{ggT}(\text{sig}(m)^{P_k} - \text{sig}_{\text{err}}^{P_k}, n) = \text{ggT}(m - \text{sig}_{\text{err}}^{P_k}, n) = q$$

Und der zweite Faktor aus:

$$p = n / q$$

Geheimer Signaturschlüssel aus:  $S_k \cdot P_k \text{ mod } \phi(n) = 1$  mit  $\phi(n) = (p - 1)(q - 1)$

$$\Rightarrow S_k = P_k^{-1} \text{ mod } ((p - 1)(q - 1))$$

# RSA-Signatur

## Zahlenbeispiel

Öffentlich bekannt:  $Pk = 101$  (öffentlicher Verifizierschlüssel)  
 $n = 221$  (Modul)

Korrektes Signaturergebnis:

$$\text{sig}(m) = [a \mathbf{sig}_1 + b \mathbf{sig}_2] \mod n = 33$$

Fehlerhafte Signaturberechnung:

$$\mathbf{sig}_{\text{err}} = [a \mathbf{sig}_{\text{err}1} + b \mathbf{sig}_2] \mod n = 84$$

Analyse:

$$\begin{aligned} &\rightarrow \text{sig}(m) - \mathbf{sig}_{\text{err}} = -51 \\ &\rightarrow \text{ggT}(-51, n = 221) = \mathbf{17} =: \mathbf{q} \quad \Rightarrow \quad p = 221 / 17 = 13 \\ &\quad \downarrow \quad \quad \downarrow \\ &\quad -3 \text{ mal } 17 \quad 13 \text{ mal } 17 \\ &\rightarrow \phi(n) = (13 - 1)(17 - 1) = 192 \end{aligned}$$

Geheimer Signaturschlüssel aus:

$$\Rightarrow \mathbf{Sk} = 101^{-1} \mod 192 = \underline{\underline{173}} \rightarrow \text{Signaturssystem gebrochen!}$$

## Kap. 6: Asymmetrische Kryptosysteme

### Teil 3: Das Rabin-Verschlüsselungsverfahren

- Rabin-Modul und quadratische Reste
- Ver- und Entschlüsselung
- Sicherheit des Verfahrens

### Asymmetrische Kryptographie:

#### Erfinder:

Michael O. Rabin

- 1979 von Michael O. Rabin (isr. Inform.) veröffentlicht
- Asymmetrisches Kryptosystem
- Öffentlicher und privater Schlüssel
- Verwandt mit RSA-Verfahren



Michael O. Rabin

### Ablauf:

Sei  $T$  der Klartext,  $G$  der verschlüsselte Geheimtext und  $n = p \cdot q$  (Rabin-Modul) sowie  $p \equiv q \equiv 3 \pmod{4}$  mit  $p, q \in \mathbb{P}$ .

- Wähle zwei Primzahlen  $p$  und  $q$  mit  $p \equiv q \equiv 3 \pmod{4}$ . Das Paar  $(p, q)$  ergibt den **geheimen** Schlüssel.
  - Als Produkt der beiden Primzahlen ergibt sich der Rabin-Modul  $n = p \cdot q$ , welcher den **öffentlichen** Schlüssel darstellt.
  - Der Sender verschlüsselt seine Nachricht mithilfe des öffentlichen Schlüssels  $n$  wie folgt:  $G = T^2 \pmod{n}$ .
  - Der Empfänger entschlüsselt die Nachricht, indem er mithilfe des geheimen Schlüssels  $(p, q)$  die **vier** Zahlen  $\pm r$  und  $\pm s$  berechnet durch
-

$$r = (y_p \cdot p \cdot T_q + y_q \cdot q \cdot T_p) \bmod n, \quad s = (y_p \cdot p \cdot T_q - y_q \cdot q \cdot T_p) \bmod n$$

mit

$$y_p \cdot p + y_q \cdot q = 1, \text{ sodass gilt: } T \in \{\pm r \bmod n, \pm s \bmod n\}.$$

Die Sicherheit des Verfahrens bemisst sich aus kryptologischer Sicht an:

$$G = T^2 \bmod n \text{ schwierig nach } T \text{ aufzulösen!}$$

$$\Rightarrow n = \{2^{400} \dots 2^{1024}\}$$

### Anmerkung:

Man nennt **y** in der Gleichung des Typs  $y \equiv x^2 \bmod n$  auch als **quadratischen Rest** bezüglich des Moduls n, falls Lösungen für x bzw. Quadratwurzeln von y mod n existieren. Sonst quadratischer Nichtrest.

---



### Faktorisierungsalgorithmus:

Bestimme Zufallszahl  $r \in \mathbb{Z}_n^*$

Berechne  $x = \text{RABINDECRYPT}(r^2 \bmod n)$ , d. h.  $x = r^2 \bmod n$

**if** (  $x \equiv + r \pmod{n}$  ||  $x \equiv - r \pmod{n}$  ) // ODER-Beziehung

**dann** Fehlschlag (wg. trivialer Quadratwurzel)

→ neues  $r$  bestimmen

**sonst** faktorisieren:  $p = \text{ggT}(x + r, n)$  sowie  $q = n / p$ .

Der Algorithmus faktorisiert  $n$  mit einer Erfolgswahrscheinlichkeit von  $1/2$ .

---

(Anm.:  $-r = n - r$ )

---

## Kap. 6: Asymmetrische Kryptosysteme

### Teil 4: Signaturen und Authentifizierung

- ElGamal-Signaturen über  $\mathbb{Z}_p^*$
- Das Drei-Wege-Protokoll nach X.509

---

Der **ElGamal-Algorithmus** ist eine Verallgemeinerung des Diffie-Hellman-Verfahrens und beruht auch auf der Basis des **diskreten Logarithmusproblems**. Mit den gleichen Bezeichnungen wie auf Folie 4 ergibt sich die gleiche Prozedur bis zum Austausch der beiden öffentlichen Schlüssel.

- **A** und **B** einigen sich auf eine große **Primzahl** **p** und eine geeignete **Primitivwurzel** **g**. Die beiden Zahlen dürfen öffentlich bekannt sein.
- **A** wählt eine Zufallszahl **SK<sub>A</sub>** und sendet **PK<sub>A</sub>** =  $g^{\text{SK}_A} \bmod p$  an **B**.  
(**PK<sub>A</sub>**, g, p) ist der **öffentliche**, (**SK<sub>A</sub>**, g, p) der **geheime** Schlüssel von **A**.
- **B** wählt eine Zufallszahl **SK<sub>B</sub>** und sendet **PK<sub>B</sub>** =  $g^{\text{SK}_B} \bmod p$  an **A**.  
(**PK<sub>B</sub>**, g, p) ist der öffentliche, (**SK<sub>B</sub>**, g, p) der geheime Schlüssel von **B**.

## Signaturerstellung:

Wir gehen davon aus, dass der Teilnehmer **A** dem Teilnehmer **B** eine signierte Nachricht  **$h(m)$**  übermitteln will. Die zu  **$h(m)$**  gehörige **digitale Signatur** werde durch ein Paar  **$\text{sig}(h(m)) = (r, s)$**  repräsentiert.

- **A** wählt eine zu  $p - 1$  teilerfremde Zahl  **$k$** . // Wegen Voraus.  $\exists k^{-1}$
- **A** berechnet  $r = g^k \bmod p$ .
- **A** löst die Kongruenz  $h(m) = (SK_A \cdot r + k \cdot s) \bmod (p - 1)$ . Der unbekannte Wert  $s$  ergibt sich durch  **$s = k^{-1} \cdot (h(m) - SK_A \cdot r) \bmod (p - 1)$** .
- **A** schickt  **$h(m)$**  sowie  **$\text{sig}(h(m))$**  d. h.  **$h(m)$**  und  **$(r, s)$**  an **B**.

## Signaturprüfung:

Es ergibt sich mit

$$h(m) = (SK_A \cdot r + k \cdot s) \bmod (p - 1)$$

die Beziehung:

$$g^{h(m)} = g^{SK_A \cdot r + k \cdot s} = g^{SK_A \cdot r} \cdot g^{k \cdot s} = PK_A^r \cdot r^s \bmod p.$$

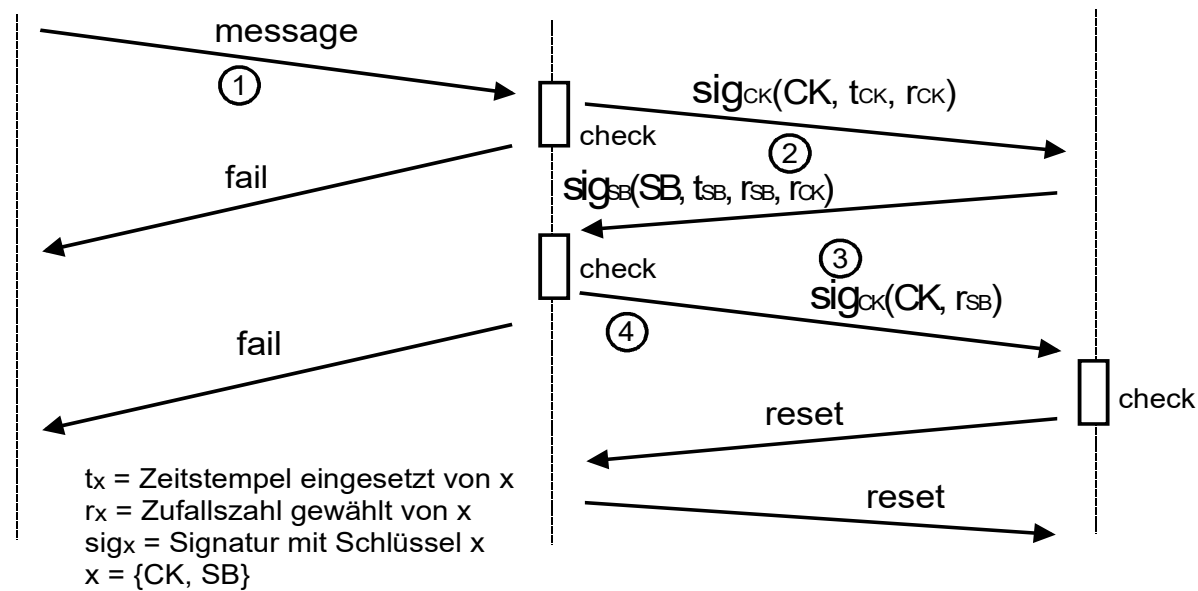
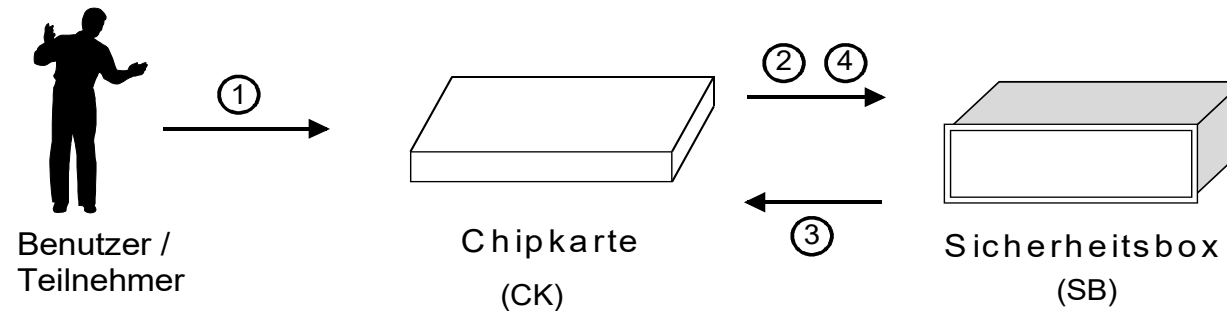
Damit ist jeder Teilnehmer, insbesondere der Empfänger **B**, in der Lage zu verifizieren, ob **h(m)** tatsächlich von **A** signiert wurde.

$$\text{Verify}(h(m), (r, s), PK_A) = \text{true} \iff g^{h(m)} \bmod p = PK_A^r \cdot r^s \bmod p$$

Im Gegensatz zu RSA ist hier eine Signatur  $(r, s)$  etwa doppelt so lang.

## Sicherheit des Verfahrens:

- In der Praxis wählt man auch hier die (sichere) Primzahlen  $p \in \mathbf{P}$  (sogenannte *Sophie-Germain-Primzahl*) gemäß  $p = 2q + 1$ , wobei  $q \in \mathbf{P}$  ebenfalls eine Primzahl ist.
- Dann besitzen alle Untergruppen die **Ordnung**  $q$ .
- Die Größe von  $q$  bestimmt die Sicherheit des Verfahrens.
- Die Parameterwahl  $p$ ,  $q$  und  $g$  kann für alle Teilnehmer gemeinsam getroffen werden.
- Ferner wird eine **kollisionsresistente** Hashfunktion  $h$  benötigt.
- Alle  $SK_T$  der Teilnehmer  $T$  müssen selbstverständlich **geheim** gehalten werden.



# Dreiwegeprotokoll

X.509

Initiator **A**

Responder **B**

$A, \{B\}, R_a, \text{SecNeg}_a, \{\text{Cert}_a\}$

Flow1-3WE  $\longrightarrow$

$A, B, \text{SecNeg}_b, \{\text{Cert}_b\}, \{R_a, R_b, \{\text{Enc}_{K_a}(\text{ConfPar}_b)\},$   
 $\text{Sig}_{K_b}(\text{hash}(A, B, R_a, R_b, \text{SecNeg}_a, \text{SecNeg}_b, \{\text{ConfPar}_b\}))\}$

$\longleftarrow$  Flow2-3WE

$\{A, B, R_b, \{\text{Enc}_{K_b}(\text{ConfPar}_a)\}, \text{Sig}_{K_a}(\text{hash}(A, B, R_b, \{\text{ConfPar}_a\}))\}$

Flow3-3WE  $\longrightarrow$

**SecNeg**<sub>\_\_</sub> = Security Negotiation

**Cert**<sub>\_\_</sub> = Zertifikat / CRL

**Enc**<sub>\_\_</sub> = Verschlüsselung

**ConfPara**<sub>\_\_</sub> = Confidential Parameters

**Sig**<sub>\_\_</sub> = Signatur

**hash** = Hashfunktion



---

## Kap. 6: Asymmetrische Kryptosysteme

### Zusammenfassung:

- In diesem Kapitel wurden asymmetrische Kryptoverfahren, die auf dem Faktorisierungsproblem, dem modularen Wurzelziehen und dem diskreten Logarithmusproblem beruhen, vorgestellt.
- Dabei wurden zunächst das auf dem Faktorisierungsproblem basierende Signaturverfahren von Rivest, Shamir und Adleman (**RSA**-Verfahren) und anschließend das auf quadratischen Resten beruhende Verschlüsselungsverfahren von **Rabin** präsentiert.
- Dann folgte das auf dem DLP basierende Schlüsselaustauschprotokoll von von **ElGamal** (**EG**-Verfahren).