Security

- LV 4120 und 7240 -

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

ElGamal-Verschlüsselung

- 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 $\mathbf{PK_T}$ sowie einen **geheimen** Schlüssel $\mathbf{SK_T}$.
- 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.

ElGamal-Verschlüsselung

Systemparameter:

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

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

3. einen Erzeuger g.

Anmerkung:

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

$$p - 1 = 2\,q\,,$$

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

ElGamal-Verschlüsselung

Schlüsselerzeugung:

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

1. **B** wählt zufällig eine Zahl $b \in \{1, ..., p - 1\}$ mit $ggT(b, p) = 1$.
2. **B** berechnet das Gruppenelement:

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

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

$$SK_B := b$$

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

$$PK_B := B$$

ElGamal-Verschlüsselung

Verschlüsselungsvorschrift:

Um die Nachricht m $\in$ **G**(**Z**p*) zu versenden, verfährt der **SenderA** folgendermaßen:

1. **A** wählt zufällig eine Zahl $r \in \{1, ..., p - 1\}$ mit $ggT(r, p) = 1$.
2. **A** berechnet das Gruppenelement:

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

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

$$c := PK_B{}^r \cdot m \in \mathbf{G}(\mathbf{Z}p^*)$$

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):

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

bzw. unter Verwendung des Satzes von Fermat [1)]:

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

---

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:

$$R^{-SK_B} \cdot c = (g^r)^{-SK_B} \cdot PK_B^{\,r} \cdot m = (g)^{-r \cdot SK_B} \cdot (g^b)^r \cdot m$$

$$= \underbrace{(g)^{-r \cdot b} \cdot (g^b)^r}_{=1} \cdot m$$

$$= \mathbf{m} \qquad \square$$

Anmerkung:

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

ElGamal-Verschlüsselung

Aufgabenstellung:

Wir betrachten das ElGamal-Verschlüsselungsverfahren über der Gruppe **G(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 ($\forall$ mod p!)

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

2. Entschlüsselung:

$\Rightarrow$ p = 29 ($\forall$ mod p!)

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

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

$\Rightarrow$

- Geheimer Schlüssel aus $5 \equiv 2^{SK_B} \bmod 29 \ \Rightarrow$ **SK_B = 22 (DL-Problem!)**
- Gruppenelement $R^{p-1-SK_B} \cdot c \ = (24^{29-1-22} \cdot 8) \bmod 29$

  $= (24^{6} \cdot 8) \bmod 29 = 10$
- D. h. die gesendete Nachricht ist m = 10. □

Sicherheit des Verfahrens:

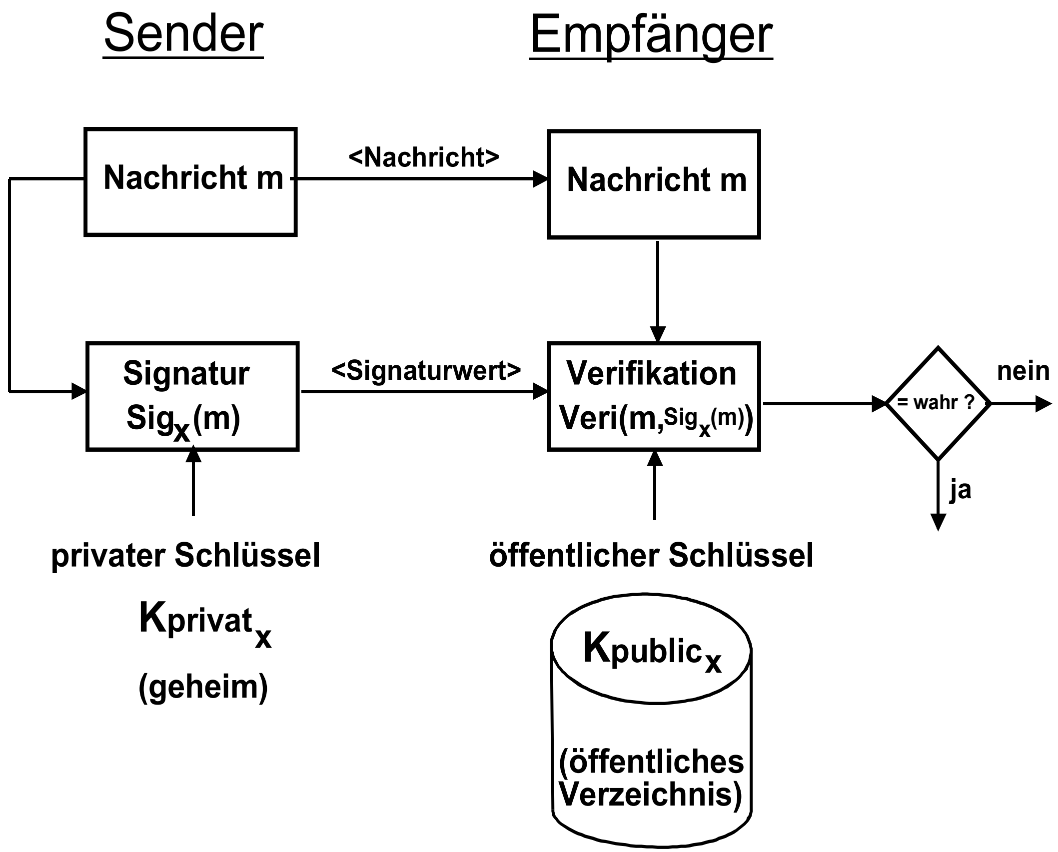
- Das ElGamal-Verschlüsselungsverfahren baut auf der Idee des Diffie-Hellman-Verfahrens auf.
- Es ist beweisbar sicher unter der Annahme, dass das sogenannte Diskret-Log-Problem in der zugrundeliegenden Gruppe schwierig ist.
- Durch schlechte Parameterwahl oder Implementierungsfehler können jedoch Spezialfälle 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

Digitale Signatur



Basisprinzip:

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

Eigenschaften:

- Nachweisbarkeit
- Nicht Abstreitbarkeit
- Authentizität
- Echtheit
- Identitätsnachweis

Signiervorschrift:

$$\boxed{\mathbf{sig(m) := h(m)^{Sk}}\ \boldsymbol{mod}\ \mathbf{n}}$$

wobei	n	=	p . q; (Modul)
	h	=	Hashfunktion
	Sk	=	geheimer Signaturschlüssel

Schlüsselerzeugung:

$Sk \cdot Pk \ mod\ \phi(n) = 1$ sowie $ggT(Sk, \phi(n)) = 1$

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

Verifiziervorschrift:

$$\boxed{\mathbf{h(m)\ {}^{?}{=}^{?}\ sig(m)^{Pk}}\ \boldsymbol{mod}\ \mathbf{n}}$$

wobei	Pk	=	öffentlicher Verifizierschlüssel

Sicherheit:

$Sk = Pk^{-1}\ mod\ \phi(n)$ schwierig zu berechnen! $\Rightarrow$ $n = \{2^{768} \dots 2^{4096}\}$

RSA-Verfahren

	Vorgang ↓	*Erklärung* ↓
Wähle:	$p = 13$	Primzahl, ist geheim zu halten
	$q = 17$	Primzahl, ist geheim zu halten
Berechne:	$n = 13 \cdot 17 = 221$	$n = p \cdot q$ (Modul, öffentlich)
	$\phi(n) = (13 - 1)(17 - 1) = 192$	$\phi(n) = (p - 1)(q - 1)$;
Wähle (zufällig):		
	Pk = 101	(öffentl. Verifizierschlüssel)
Nun ermittle **Sk** aus		
	Sk $\cdot$ 101 *mod* 192 = 1	Sk $\cdot$ Pk *mod* $\phi(n) = 1$
		d. h.
		Sk $\cdot 101 = z \cdot 192 + 1$
	$\Rightarrow$ **Sk = 173** (für $z = 91$)	(geheimer Signaturschlüssel)

	Vorgang ↓	***Erklärung*** ↓
Annahme:	h(m) := **50**	zu signierende Nachricht (Text) $50 = 110010_2$
signieren:		
	$\text{sig(m)} = 50^{173}\ mod\ 221 = \underline{\underline{\mathbf{33}}}$	$\text{sig(m)} := \text{h(m)}^{Sk}\ mod\ \text{n}$ Die berechnete Signatur ist $33 = 100001_2$
verifizieren:		
	$33^{101}\ mod\ 221 = \underline{\underline{\mathbf{50 := h(m)}}}$	$\text{sig(m)}^{Pk}\ mod\ \text{n} \ ?=?\ \text{h(m)}$ Die berechnete Signatur ist korrekt!

Signatur: $\mathrm{sig(m)} := \mathrm{m}^{\mathrm{Sk}} \bmod n = \mathbf{a}\,\underbrace{(\mathrm{m}^{\mathrm{Sk}} \bmod p)}_{\mathbf{sig}_{\mathbf{1}}} + \mathbf{b}\,\underbrace{(\mathrm{m}^{\mathrm{Sk}} \bmod q)}_{\mathbf{sig}_{\mathbf{2}}}$ wobei

$\mathrm{a} \bmod q(p) = 0(1)$ $\qquad$ $\mathrm{b} \bmod q(p) = 1(0)$

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

$$\mathrm{m}^{\mathrm{Sk}} \bmod p = \mathbf{1}\,(\mathrm{m}^{\mathrm{Sk}} \bmod p) \qquad \square$$

Analog mod q: $\Rightarrow$ $\mathrm{m}^{\mathrm{Sk}} \bmod q = \mathbf{1}\,(\mathrm{m}^{\mathrm{Sk}} \bmod q) \qquad \square$

Fehlerbetrachtung (Differential Fault Analysis):

Voraussetzung: Fehler bei der Berechnung von $\mathbf{sig}_{\mathbf{1}}$ oder $\mathbf{sig}_{\mathbf{2}}$

Annahme: Fehler bei der Berechnung $\mathbf{sig}_{\mathbf{1}}$; Berechnung $\mathbf{sig}_{\mathbf{2}}$ sei o.k.!

$\mathbf{sig}_{\mathbf{err}}$ ist das Ergebnis der fehlerhaften Berechnung von sig(m)

$$\mathrm{sig(m)} = \mathbf{a}\ \mathbf{sig}_{\mathbf{1}}(\mathrm{m}) + \mathbf{b}\ \mathbf{sig}_{\mathbf{2}}(\mathrm{m}) \quad \Rightarrow \text{korrekte Signatur}$$

$$\mathrm{sig}_{\mathbf{err}}(\mathrm{m}) = \mathbf{a}\ \mathbf{sig}_{\mathbf{err}}(\mathrm{m}) + \mathbf{b}\ \mathbf{sig}_{\mathbf{2}}(\mathrm{m}) \quad \Rightarrow \text{fehlerhafte Signatur}$$

$$\mathrm{sig(m)} - \mathrm{sig}_{\mathbf{err}}(\mathrm{m}) = \mathbf{a}\,(\mathbf{sig}_{\mathbf{1}}(\mathrm{m}) - \mathbf{sig}_{\mathbf{err}}(\mathrm{m}))$$ Differential Fault Analysis (DFA)

$$(\mathrm{sig}(m) - \mathrm{sig}_{\mathbf{err}}(m)) = \mathbf{a}\ (\mathbf{sig}_{\mathbf{1}}(m) - \mathbf{sig}_{\mathbf{err}}(m)) \quad |\ \mathrm{mod}\ q$$

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

dann gilt: $\mathrm{sig}(m) \equiv \mathbf{sig}_{\mathbf{err}} \bmod q$ aber $\mathrm{sig}(m) \neq \mathbf{sig}_{\mathbf{err}} \bmod p$

also ist: $\mathbf{q}$ ein Teiler von $(m - \mathbf{sig}_{\mathbf{err}}{}^{Pk})$ und $\mathbf{p}$ <u>kein</u> Teiler von $(m - \mathbf{sig}_{\mathbf{err}}{}^{Pk})$

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

$$\mathrm{ggT}(\mathrm{sig}(m)^{Pk} - \mathbf{sig}_{\mathbf{err}}{}^{Pk}, n) = \mathrm{ggT}(m - \mathbf{sig}_{\mathbf{err}}{}^{Pk}, n) = \mathbf{q}$$

Und der zweite Faktor aus:

$$\mathbf{p} = n / q$$

Geheimer Signaturschlüssel aus: $Sk \cdot Pk\ \mathit{mod}\ \phi(n) = 1$ mit $\phi(n) = (\mathbf{p} - 1)\ (\mathbf{q} - 1)$

$$\boxed{\Rightarrow \mathbf{Sk = Pk^{-1}}\ \boldsymbol{mod}\ \mathbf{((p - 1)\ (q - 1))}}$$

RSA-Signatur

Zahlenbeispiel

Öffentlich bekannt: Pk = 101 (öffentlicher Verifizierschlüssel)

n = 221 (Modul)

Korrektes Signaturergebnis:

$$sig(m) = [a\ \mathbf{sig}_{\mathbf{1}} + b\ \mathbf{sig}_{\mathbf{2}}] \quad mod\ n = 33$$

Fehlerhafte Signaturberechnung:

$$\mathbf{sig}_{\mathbf{err}} = [a\ \mathbf{sig}_{\mathbf{err\ 1}} + b\ \mathbf{sig}_{\mathbf{2}}] \quad mod\ n = 84$$

Analyse:

$$\rightarrow sig(m) - \mathbf{sig}_{\mathbf{err}} = -51$$

$$\rightarrow ggT(\underbrace{-51}_{\textbf{-3}\text{ mal }\textbf{17}},\ n = \underbrace{221}_{\textbf{13}\text{ mal }\textbf{17}}) = \mathbf{17 =: q} \quad \Rightarrow \quad p = 221 / 17 = 13$$

$$\rightarrow \phi(n) = (13 - 1)(17 - 1) = 192$$

Geheimer Signaturschlüssel aus:

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

Kap. 6: Asymmetrische Kryptosysteme

Teil 3: Das Rabin-Verschlüsselungsverfahren

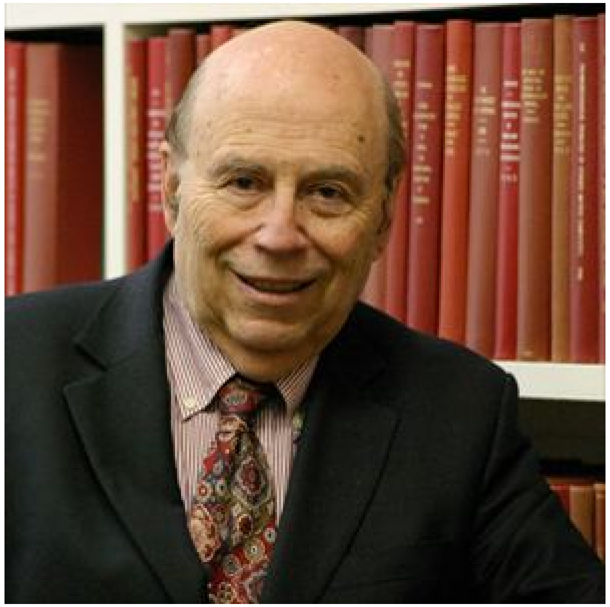
- Rabin-Modul und quadratische Reste
- Ver- und Entschlüsselung
- Sicherheit des Verfahrens

Asymmetrische Kryptographie:

Erfinder:

Michael O. Rabin

- 1979 von Michael O. Rabin (isr. Inform.) veröffentlicht
- Asymmetrisches Krypto-system
- 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 $p \equiv q \equiv 3 \bmod 4$ mit $p, q \in P$.

- Wähle zwei Primzahlen p und q mit $p \equiv q \equiv 3 \bmod 4$. 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: $G = T^2 \bmod n$.
- Der Empfänger entschlüsselt die Nachricht, indem er mithilfe des geheimen Schlüssels (p, q) die **vier** Zahlen ± r und ± s berechnet durch

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

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:

$G = T^2$ *mod* n schwierig nach T aufzulösen!

$$\Rightarrow \quad n = \{2^{400} \ldots 2^{1024}\}$$

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 $r \in Z_n^*$

Berechne x = RABINDECRYPT(r^2 mod 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)

$\rightarrow$ neues r bestimmen

sonst faktorisieren: $p = ggT(x + r, n)$ 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 Zp*
- Das Drei-Wege-Protokoll nach X.509

ElGamal-Signatur

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 $\mathbf{SK_A}$ und sendet $\mathbf{PK_A} = g^{\mathbf{SK_A}} \bmod p$ an **B**. ($\mathbf{PK_A}$, g, p) ist der öffentliche, ($\mathbf{SK_A}$, g, p) der geheime Schlüssel von **A**.
- **B** wählt eine Zufallszahl $\mathbf{SK_B}$ und sendet $\mathbf{PK_B} = g^{\mathbf{SK_B}} \bmod p$ an **A**. ($\mathbf{PK_B}$, g, p) ist der öffentliche, ($\mathbf{SK_B}$, g, p) der geheime Schlüssel von **B**.

ElGamal-Signatur

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 sig(**h(m)**) = (r, s) repräsentiert.

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

Signaturprüfung:

Es ergibt sich mit

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

die Beziehung:

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

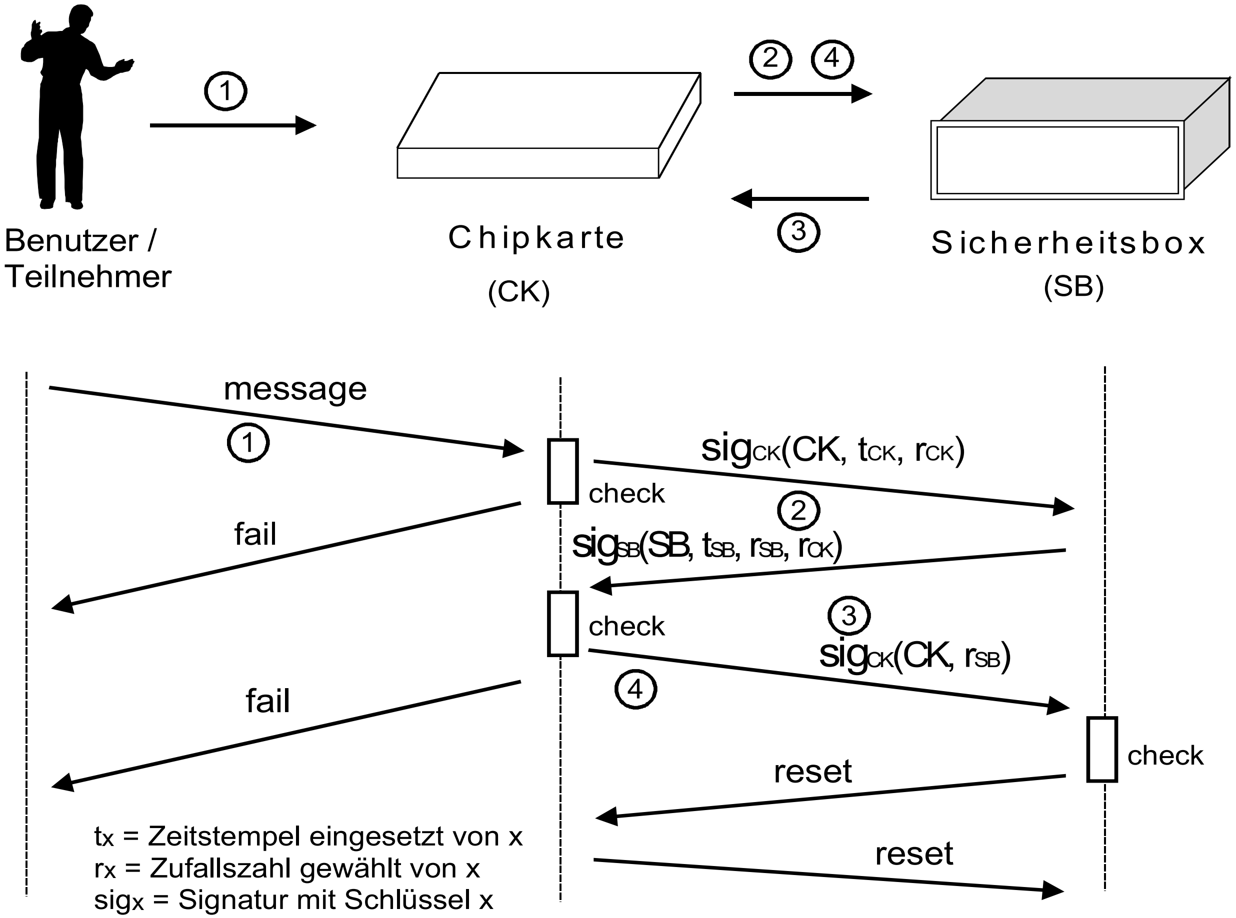
Damit ist jeder Teilnehmer, insbesondere der Empfänger **B**, in der Lage zu verifizieren, ob **h(m)** tatsächlich von **A** signiert wurde.

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

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.



t_x = Zeitstempel eingesetzt von x
r_x = Zufallszahl gewählt von x
sig_x = Signatur mit Schlüssel x
x = {CK, SB}

Dreiwegeprotokoll

Initiator **A** Responder **B**

$A, \{B\}, R_a, SecNeg_a, \{Cert_a\}$

Flow1-3WE ——————————————————————————>

$A, B, SecNeg_b, \{Cert_b\}, \{R_a, R_b, \{Enc_{Ka}(ConfPar_b)\},$
$Sig_{Kb}(hash(A, B, R_a, R_b, SecNeg_a, SecNeg_b, \{ConfPar_b\}))\}$

<—————————————————————————— Flow2-3WE

$\{A, B, R_b, \{Enc_{Kb}(ConfPar_a)\}, Sig_{Ka}(hash(A, B, R_b, \{ConfPar_a\}))\}$

Flow3-3WE ——————————————————————————>

SecNeg_	=	Security Negotiation	**ConfPara_**	=	Confidential Parameters
Cert_	=	Zertifikat / CRL	**Sig_**	=	Signatur
Enc_	=	Verschlüsselung	**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).