

Hybride Verschlüsselungsverfahren

Ziel: Flexibilität von asym. Verfahren und Effizienz von sym. Verfahren.

Szenario:

- Sei $\Pi = (Gen, Enc, Dec)$ ein PK-Verschlüsselungsverfahren und $\Pi' = (Gen', Enc', Dec')$ ein SK-Verschlüsselungsverfahren.
- Berechne $(pk, sk) \leftarrow Gen(1^n)$.

Algorithmus Hybride Verschlüsselung

Eingabe: m, pk

- 1 Wähle $k \in_R \{0, 1\}^n$.
- 2 Verschlüssele $c_1 \leftarrow Enc_{pk}(k)$ mit asym. Verschlüsselung.
- 3 Verschlüssele $c_2 \leftarrow Enc'_k(m)$ mit sym. Verschlüsselung.

Ausgabe: Chiffretext $c = (c_1, c_2)$

Hybride Entschlüsselung

Algorithmus Hybride Entschlüsselung

Eingabe: $c = (c_1, c_2)$, sk

- 1 Entschlüssele $k \leftarrow Dec_{sk}(c_1)$.
- 2 Entschlüssele $m \leftarrow Dec'_k(c_2)$.

Ausgabe: Klartext m

- Effizienzgewinn für $|m| \gg n$, sofern Π' effizienter ist als Π .
- **Frage:** Ist hybride Verschlüsselung sicher, falls Π, Π' sicher?

Sicherheit von hybrider Verschlüsselung

Satz Sicherheit hybrider Verschlüsselung

Sei Π ein CPA-sicheres PK-Verschlüsselungsverfahren und Π' ein KPA-sicheres SK-Verschlüsselungsverfahren.
Dann ist das hybride Verfahren Π^{hy} CPA-sicher.

Beweisskizze:

- **Notation** $X \equiv Y$: Kein ppt Angreifer kann X und Y unterscheiden.
- Sicherheit von Π' : $Enc_k(m_0) \equiv Enc_k(m_1)$ für $k \in_R \{0, 1\}^n$.
- Sicherheit von Π^{hy} : Müssen zeigen dass

$$(Enc_{pk}(k), Enc'_k(m_0)) \equiv (Enc_{pk}(k), Enc'_k(m_1)).$$

- Problem: Erstes Argument könnte beim Unterscheiden des zweiten Arguments helfen.

Beweis Sicherheit hybrider Verschlüsselung

Beweisskizze: Zeigen die folgenden 3 Schritte

- ① Sicherheit von Π liefert

$$(Enc_{pk}(k), Enc'_k(m_0)) \equiv (Enc_{pk}(0^n), Enc'_k(m_0)).$$

Gilt sogar falls \mathcal{A} die Werte $k, 0^n$ kennt.

Dass das zweite Argument k beinhaltet ist daher kein Problem.

- ② Sicherheit von Π' liefert

$$(Enc_{pk}(0^n), Enc'_k(m_0)) \equiv (Enc_{pk}(0^n), Enc'_k(m_1)).$$

Kein Problem mehr, da 2. Argument nicht vom ersten abhängt.

- ③ Sicherheit von Π liefert

$$(Enc_{pk}(0^n), Enc'_k(m_1)) \equiv (Enc_{pk}(k), Enc'_k(m_1)).$$

Transitivität der 3 Ergebnisse liefert schließlich wie gewünscht

$$(Enc_{pk}(k), Enc'_k(m_0)) \equiv (Enc_{pk}(k), Enc'_k(m_1)).$$

Textbook RSA

Algorithmus Schlüsselerzeugung *GenRSA*

Eingabe: 1^n

- 1 $(N, p, q) \leftarrow \text{GenModulus}(1^n)$ mit primen n -Bit p, q und $N = pq$.
- 2 $\phi(N) \leftarrow (p - 1)(q - 1)$
- 3 Wähle $e \in \mathbb{Z}_{\phi(N)}^*$.
- 4 Berechne $d \leftarrow e^{-1} \bmod \phi(N)$.

Ausgabe: (N, e, d) mit $pk = (N, e)$ und $sk = (N, d)$.

Definition Textbook RSA Verschlüsselungsverfahren

Sei n ein Sicherheitsparameter.

- 1 **Gen** : $(N, e, d) \leftarrow \text{GenRSA}(1^n)$
- 2 **Enc** : $c \leftarrow m^e \bmod N$ für eine Nachricht $m \in \mathbb{Z}_N$.
- 3 **Dec** : $m \leftarrow c^d \bmod N$

RSA Problem und Sicherheit von RSA

Definition RSA Problem

Das *RSA Problem* ist hart bezüglich $GenRSA(1^n)$, falls für alle ppt Algorithmen \mathcal{A} gilt $Ws[\mathcal{A}(N, e, m^e \bmod N) = m] \leq \text{negl}(n)$.

Wsraum: Wahl $m \in_R \mathbb{Z}_N$ und interne Münzwürfe von \mathcal{A} , $GenRSA$.

RSA Annahme: Das RSA Problem ist hart bezüglich $GenRSA$.

Anmerkungen:

- Falls N effizient faktorisiert werden kann, ist das RSA Problem nicht hart. (Warum?)
- Berechnen von d ist so schwer wie Faktorisieren von N .
- **Offenes Problem:** Impliziert eine Lösung des RSA Problem eine Lösung des Faktorisierungsproblems?
- *Enc* ist deterministisch, d.h. Textbook RSA ist nicht CPA-sicher.
- Unter der RSA-Annahme: Kein ppt Angreifer kann für zufällige $m \in \mathbb{Z}_N^*$ aus $(N, e, m^e \bmod N)$ die ganze Nachricht m berechnen.

Angriffe auf Textbook RSA

Verschlüsseln von kurzen Nachrichten mit kleinem e

- Sei $m < N^{\frac{1}{e}}$. Dann gilt $c = m^e < N$. D.h. $c^{\frac{1}{e}}$ über \mathbb{Z} liefert m .
- Realistisch bei hybrider Verschlüsselung: N 1024-Bit und $e = 3$, m ist 128-Bit Schlüssel für symmetrische Verschlüsselung.

Hastad Angriff auf RSA

- **Szenario:** Verschlüsselung desselben m unter verschiedenen pk .
- Sei $pk_1 = (N_1, 3)$, $pk_2 = (N_2, 3)$, $pk_3 = (N_3, 3)$. Angreifer erhält $c_1 = m^3 \bmod N_1$, $c_2 = m^3 \bmod N_2$ und $c_3 = m^3 \bmod N_3$.
- Berechne mittels Chinesischem Restsatz eind. $c \in \mathbb{Z}_{N_1 N_2 N_3}$ mit

$$\begin{cases} c = c_1 \bmod N_1 \\ c = c_2 \bmod N_2 \\ c = c_3 \bmod N_3 \end{cases}.$$

- Es gilt $c = m^3 < (\min\{N_1, N_2, N_3\})^3 < N_1 N_2 N_3$, d.h. $m = c^{\frac{1}{3}}$.

Padded RSA

Definition Padded RSA Verschlüsselungsverfahren

Sei n ein Sicherheitsparameter und ℓ eine Fkt. mit $\ell(n) \leq 2n - 2$.

- 1 **Gen** : $(N, e, d) \leftarrow \text{GenRSA}(1^n)$
- 2 **Enc** : Für $m \in \{0, 1\}^{\ell(n)}$ und $r \in_R \{0, 1\}^{|N| - \ell(n) - 1}$ berechne
$$c \leftarrow (r||m)^e \bmod N.$$
- 3 **Dec** : $r||m := c^d \bmod N$. Gib die untersten $\ell(n)$ Bits aus.

Anmerkungen

- Für $\ell(n) = 2n - \mathcal{O}(\log n)$ kann r in polyn. Zeit geraten werden.
- Für $\ell(n) = cn$, konstantes $c < 2$, ist kein CPA Angriff bekannt.
- Für $\ell(n) = \mathcal{O}(\log n)$ kann CPA-Sicherheit gezeigt werden.
- Weitverbreitete standardisierte Variante von Padded RSA: PKCS #1 version 1.5 mit $c := (0^8||0^610||r||0^8||m)^e \bmod N$.

Einfache **symmetrische** Verschlüsselung

Algorithmus ONE-TIME GRUPPENELEMENT

Sei n ein Sicherheitsparameter.

- 1 **Gen:** Schlüsselerzeugung $(G, g) \leftarrow \mathcal{G}(1^n)$, wobei G eine Gruppe und $g \in_R G$ ein zufälliger gemeinsamer geheimer Schlüssel ist.
- 2 **Enc:** Verschlüssele $m \in G$ als $c \leftarrow m \cdot g$.
- 3 **Dec:** Entschlüssele $c \in G$ als $m := c \cdot g^{-1}$.

Anmerkung:

- Das One-Time Pad erhalten wir als Spezialfall mit $(G, \cdot) = (\mathbb{F}_2^n, +)$.

Perfekte Sicherheit von ONE-TIME ELEMENT

Satz Perfekte Sicherheit von ONE-TIME ELEMENT (Krypto I)

ONE-TIME ELEMENT Π ist eine perfekt sichere **symmetrisches** Verschlüsselung, d.h. für alle Angreifer \mathcal{A} gilt $W_s[\text{PrivK}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}$.

Beweis:

- Klartext-, Chiffretext und Schlüsselraum sind identisch

$$\mathcal{M} = \mathcal{C} = \mathcal{K} = G.$$

- Jeder Schlüssel $g \in G$ wird gleichverteilt mit $W_s \frac{1}{|G|}$ gewählt.
- Für jedes $m, c \in G$ existiert genau ein Schlüssel $g = c \cdot m^{-1}$ mit

$$c = \text{Enc}_g(m).$$

- Nach dem Satz von Shannon ist Π damit perfekt sicher. D.h.

$$W_s[\text{PrivK}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}.$$

Anmerkungen:

- Geheimer Schlüssel $sk = g$ muss stets neu gewählt werden.
- Idee für PK-Verfahren: Ersetze das zufällige g durch ein stets neu gewähltes “pseudozufälliges” Gruppenelement.

ElGamal Verschlüsselungsverfahren (1984)

Definition ElGamal Verschlüsselungsverfahren

Sei n ein Sicherheitsparameter.

- Gen** : $(q, g) \leftarrow \mathcal{G}(1^n)$, wobei g eine Gruppe G der Ordnung q generiert. Wähle $x \in_R \mathbb{Z}_q$ und berechne $h \leftarrow g^x$.
Schlüssel: $pk = (q, g, h)$, $sk = (q, g, x)$
- Enc** : Für eine Nachricht $m \in G$ wähle ein $y \in_R \mathbb{Z}_q$ und berechne
 $c \leftarrow (g^y, h^y \cdot m)$.
- Dec** : Für einen Chiffretext $c = (c_1, c_2)$ berechne $m := \frac{c_2}{c_1}$.

- Korrektheit**: $\frac{c_2}{c_1^x} = \frac{h^y \cdot m}{(g^y)^x} = \frac{(g^x)^y \cdot m}{g^{xy}} = m$.
- c_2 ist ein Analog von *Enc* bei ONE-TIME GRUPPENELEMENT mit einem DH-Schlüssel g^{xy} als "pseudozufälligem" Gruppenelement.

Anmerkung:

- Parameter q, g können global für alle Teilnehmer gewählt werden.