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 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 SK-Verschlüsselungsverfahren mit ununterscheidbaren Chiffretexten gegenüber passiven Angreifern.
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
$$(Enc_{pk}(0^n), Enc'_k(m_1)) \equiv (Enc_{pk}(k), Enc'_k(m_1)).$$

Transitititivität 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. $(N, p, q) \leftarrow \mathit{GenModulus}(1^n)$ mit primen $n/2$-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** : $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$
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 $\mathrm{Ws}[\mathcal{A}(N, e, m^e \bmod N) = m] \leq \mathrm{negl}(n)$.
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 Textbuch 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

$$\left| \begin{array}{l} c = c_1 \bmod N_1 \\ c = c_2 \bmod N_2 \\ c = c_3 \bmod N_3 \end{array} \right| .$$

- 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** : $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$
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** : $r||m \leftarrow c^d \bmod N$. 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 $c := (0^8||0^6 10||r||0^8||m)^e \bmod N$.

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 $m \leftarrow c \cdot g^{-1}$.

Perfekte Sicherheit von ONE-TIME GRUPPENELEMENT

Satz Perfekte Sicherheit von ONE-TIME GRUPPENELEMENT

ONE-TIME GRUPPENELEMENT ist ein perfekt sicheres **symmetrisches** Verschlüsselungsverfahren, d.h. für alle Angreifer $\mathcal{A}$ gilt

$$\mathrm{Ws}[\mathcal{A}(G, c) = m] = \frac{1}{|G|}.$$

Beweis:

- Sei $g' \in G$ beliebig. Da g ein zufälliges Gruppenelement ist, gilt
$$\mathrm{Ws}[c = g'] = \mathrm{Ws}[m \cdot g = g'] = \frac{1}{|G|}.$$
- Die Wsverteilung auf den Chiffretexten ist die Gleichverteilung.
- Insbesondere ist die Verteilung unabhängig von der Nachricht m.

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.

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

- **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:

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

Sicherheit von ElGamal

Satz CPA-Sicherheit ElGamal

Falls DDH schwer ist bezüglich $\mathcal{G}$, besitzt ElGamal ununterscheidbare Chiffretexte unter CPA.

Beweis-Skizze:

- Sei $\mathcal{A}$ ein Angreifer auf das ElGamal-Protokoll Π mit Erfolgsws
$$\epsilon(n) := \mathrm{Ws}[PubK^{cpa}_{\mathcal{A},\Pi}(n) = 1].$$
- Betrachten modifiziertes Verschlüsselungsverfahren Π' mit
$$c' = (c'_1, c'_2) = (g^y, g^z \cdot m) \text{ mit } y, z \in_R \mathbb{Z}_q.$$
- c' ist unabhängig gleichverteilt in G^2, d.h. unabhängig von m.
- Daher gilt $\mathrm{Ws}[PubK^{cpa}_{\mathcal{A},\Pi'}(n) = 1] = \frac{1}{2}$.
- **Idee:** Lösen von DDH durch Unterscheiden von Π und Π'.
- DDH-Instanz: (G, q, g, g^x, g^y, g') mit $g' = g^{xy}$ oder $g' = g^z$.