CPA-sicheres effizientes RSA im ROM

Verschlüsselung ROM-RSA

Sei $H : \mathbb{Z}_N^* \to \{0,1\}^{\ell(n)}$ ein Random Oracle.

1. **Gen:** $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$ mit $pk = (N, e), sk = (N, d)$.
2. **Enc:** Für $m \in \{0,1\}^{\ell(n)}$, wähle $r \in_R \mathbb{Z}_N^*$. Berechne
$$c \leftarrow (r^e \bmod N, H(r) \oplus m).$$
3. **Dec:** Für $c = (c_1, c_2)$ berechne
$$r \leftarrow c_1^d \bmod N \text{ und } m \leftarrow H(r) \oplus c_2.$$

Sicherheit von RSA im Random Oracle Modell

Satz CPA-Sicherheit von ROM-RSA

Unter der RSA-Annahme und für ein Random Oracle H ist ROM-RSA CPA-sicher.

Beweis:

- Sei Π = ROM-RSA und $\epsilon = \mathrm{Ws}_{\mathcal{A},\Pi}[PubK^{cpa}_{\mathcal{A},\Pi}(n) = 1]$.
- Angreifer $\mathcal{A}$ darf Orakelanfragen an H stellen, sowohl vor Ausgabe von (m_0, m_1) als auch nach Erhalt von $Enc(m_b)$.
- Definiere *Success* : Ereignis $PubK^{cpa}_{\mathcal{A},\Pi}(n) = 1$.
- Definiere *Query* : Ereignis $\mathcal{A}$ stellt Anfrage $r = c_1^d \bmod N$ an H.
- Es gilt

$$\begin{aligned} \mathrm{Ws}[Success] &= \mathrm{Ws}[Success \wedge \overline{Query}] + \mathrm{Ws}[Success \wedge Query] \\ &\leq \mathrm{Ws}[Success \wedge \overline{Query}] + \mathrm{Ws}[Query]. \end{aligned}$$

- Zeigen $\mathrm{Ws}[Success \wedge \overline{Query}] \leq \frac{1}{2}$ und $\mathrm{Ws}[Query] \leq \mathrm{negl}(n)$.
- Daraus folgt $\mathrm{Ws}[Success] = \epsilon(n) \leq \frac{1}{2} + \mathrm{negl}(n)$.

Beweis der CPA-Sicherheit von ROM-RSA

Beweis: $\mathrm{Ws}[\mathit{Success} \wedge \overline{\mathit{Query}}] \leq \frac{1}{2}$

- Falls r nicht an H angefragt wird, ist $H(r) \oplus m$ nach Eigenschaft des Random Oracles ein perfektes One-Time Pad für m.
- Daraus folgt $\mathrm{Ws}[\mathit{Success} \mid \overline{\mathit{Query}}] = \frac{1}{2}$. Damit gilt

$$\begin{aligned} \mathrm{Ws}[\mathit{Success} \wedge \overline{\mathit{Query}}] &= \mathrm{Ws}[\mathit{Success} \mid \overline{\mathit{Query}}] \cdot \mathrm{Ws}[\overline{\mathit{Query}}] \\ &\leq \mathrm{Ws}[\mathit{Success} \mid \overline{\mathit{Query}}] = \frac{1}{2}. \end{aligned}$$

Beweis der CPA-Sicherheit von ROM-RSA

Beweis: $\mathrm{Ws}[\mathit{Query}] \leq \mathrm{negl}(n)$

- Idee: Verwende Anfragen von $\mathcal{A}$, um e-te Wurzeln zu berechnen.

Algorithmus RSA-Invertierer $\mathcal{A}'$

EINGABE: $N, e, c_1 = r^e \bmod N$

1. Wähle $k \in_R \{0,1\}^{\ell(n)}$. (Wir setzen $H(r) = k$, ohne r zu kennen.)
2. $(m_0, m_1) \leftarrow \mathcal{A}(N, e)$, beantworte Orakelanfragen r_i an $H(\cdot)$

$$\text{konsistent mit } \begin{cases} k_i = k & \text{für } r_i^e = c_1 \bmod N \\ k_i \in_R \{0,1\}^{\ell(n)} & \text{sonst} \end{cases}.$$

3. Berechne $c \leftarrow (c_1, k \oplus m_b)$ für ein $b \in_R \{0,1\}$.
4. $b' \leftarrow \mathcal{A}(c)$, beantworte Anfragen von $\mathcal{A}$ an $H(\cdot)$ wie zuvor.
5. Falls $r_i^e = c_1 \bmod N$ für eine der Orakelanfragen, setze $r \leftarrow r_i$.

AUSGABE: r

- Es gilt $Ws[\mathit{Query}] = Ws[\mathcal{A}'(N, e, r^e) = r] \leq \mathrm{negl}(n)$.

Sicherheit gegenüber CCA

Idee:

- Ersetze One-Time Pad durch CCA-sicheres Secret Key Verfahren.
- Konstruktion von CCA-sicherem Secret Key Verfahren mittels sogenannter Pseudozufallsfunktionen und MACs möglich.

Verschlüsselung ROM-RSA-2

Sei $H : \{0,1\}^{\ell(n)} \to \mathbb{Z}_N^*$ ein Random Oracle, $\Pi' = (Gen', Enc', Dec')$ ein CCA-sicheres Secret Key Verschlüsselungsverfahren.

1. **Gen:** $(N, e, d) \leftarrow GenRSA(1^n)$ mit $pk = (N, e)$, $sk = (N, d)$.
2. **Enc:** Für $m \in \{0,1\}^{\ell(n)}$, wähle $r \in_R \mathbb{Z}_N^*$. Berechne $k = H(r)$ und
$$c \leftarrow (r^e \bmod N, Enc'_k \oplus m).$$
3. **Dec:** Für $c = (c_1, c_2)$ berechne
$$r \leftarrow c_1^d \bmod N,\ k \leftarrow H(r) \text{ und } m \leftarrow Dec'_k(c_2).$$

Sicherheit von ROM-RSA-2

Satz Sicherheit von ROM-RSA-2

Unter der RSA-Annahme, für ein Random Oracle H und ein CCA-sicheres Π' liefert ROM-RSA-2 CCA-sichere Verschlüsselung.

Anmerkungen:

- Wir werden den Satz hier nicht formal beweisen.
- Der Beweis verläuft großteils analog zum vorigen Beweis.
- Problem: Müssen Orakel $Dec_{sk}(\cdot)$ simulieren, ohne sk zu kennen.
- Verwende dazu geschicktes Simulieren des Random Oracles $H(\cdot)$.
- Bsp. für geschicktes Simulieren: s. folgender Beweis zu RSA-FDH.

RSA Full Domain Hash (RSA-FDH) Signaturen

Signatur RSA-FDH

Sei $H : \{0,1\}^* \to \mathbb{Z}_N^*$ ein Random-Oracle.

1. $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$ mit $pk = (N, e)$ und $sk = (N, d)$.
2. Für eine Nachricht $m \in \{0,1\}^*$ berechne $\sigma \leftarrow H(m)^d \bmod N$.
3. Für (m, σ) überprüfe $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

Anmerkung:

- RSA-FDH entspricht Hashed-RSA mit einem Random Oracle als Hashfunktion.

CMA-Sicherheit von RSA-FDH

Satz CMA-Sicherheit von RSA-FDH

Unter der RSA-Annahme und für ein Random-Oracle H ist RSA-FDH ein CMA-sicheres Signaturverfahren.

Beweis:

- Sei $\Pi =$ RSA-FDH und $\epsilon = \mathrm{Ws}[\mathit{Forge}_{\mathcal{A},\Pi}(n) = 1]$.
- OBdA gelten folgende Annahmen für die Orakelanfragen von $\mathcal{A}$:
  1. $\mathcal{A}$ fragt verschiedene $x_1, \ldots, x_q$ an $H(\cdot)$.
  2. Bevor $\mathcal{A}$ Anfrage m an $\mathit{Sign}_{sk}(\cdot)$ stellt, fragt er $H(m)$ an.
  3. Für eine Fälschung (m, σ) hat $\mathcal{A}$ zuvor Anfrage $H(m)$ gestellt.
- Konstruieren RSA-Invertierer $\mathcal{A}'$ mittels $\mathcal{A}$.

Beweis der CMA-Sicherheit von RSA-FDH

Algorithmus RSA-Invertierer $\mathcal{A}'$

EINGABE: $N, e, y = x^e \bmod N$

1. Wähle $j \in_R \{1, \ldots, q\}$.
2. $(m, \sigma) \leftarrow \mathcal{A}^{Sign_{sk}(\cdot)}(N, e)$.
   - Beantworte Orakelanfragen m_i an $H(\cdot)$ konsistent mit
     $$y_i = \begin{cases} y & \text{für } i = j \\ \sigma_i^e \bmod N \text{ für ein selbst gewähltes } \sigma \in_R \mathbb{Z}_N^* & \text{sonst} \end{cases}.$$
   - Beantworte Orakelanfragen m_i an $Sign_{sk}(\cdot)$ mit σ_i für $i \neq j$. Bei Orakelanfrage $Sign_{sk}(m_j)$, Abbruch.
3. Falls $m = m_j$ und $\sigma^e = y \bmod N$, setze $x \leftarrow \sigma$.

AUSGABE: x

- Unter der RSA-Annahme gilt $\mathrm{negl}(n) \geq Ws[\mathcal{A}'(N, e, x^e) = x]$
  $$= Ws[m = m_j] \cdot \mathrm{Ws}[Forge_{\mathcal{A},\Pi}(n) = 1] = \frac{\epsilon(n)}{q}.$$
- Damit ist $\epsilon(n) \leq q \cdot \mathrm{negl}(n)$ vernachlässigbar für polynomielles q.