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 $\Pi =$ 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 (1/2)

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 (2/2)

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
$$c \leftarrow (r^e \bmod N, Enc'_k(m)).$$
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.

Beweisskizze:

- 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.

Jacobi-Symbol

Erinnerung Jacobi-Symbol: Beweise siehe Diskrete Mathematik II

Definition Quadratischer Rest

Sei $N \in \mathbb{N}$. Ein Element $a \in \mathbb{Z}_N$ heißt *quadratischer Rest* in $\mathbb{Z}_N$, falls es ein $b \in \mathbb{Z}_N$ gibt mit $b^2 = a \bmod N$. Wir definieren

$$QR_N = \{a \in \mathbb{Z}_N^* \mid a \text{ ist quadratischer Rest } \} \text{ und } QNR_N = \mathbb{Z}_N^* \setminus QR_N.$$

Lemma Anzahl quadratischer Reste in primen Restklassen

Sei $p > 2$ prim. Dann gilt $|QR_p| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p-1}{2}$.

Beweisidee:

- Quadrieren auf $\mathbb{Z}_p^*$, $x \mapsto x^2$, ist eine 2:1-Abbildung.
- Die verschiedenen Werte $x, (-x)$ werden beide auf x^2 abgebildet.

Legendre-Symbol

Definition Legendre Symbol

Sei $p > 2$ prim und $a \in \mathbb{N}$. Das *Legendre Symbol* ist definiert als

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{falls } p|a \\ 1 & \text{falls } (a \bmod p) \in QR_p \\ -1 & \text{falls } (a \bmod p) \in QNR_p \end{cases} .$$

Satz

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \bmod p$$

Eigenschaften Quadratischer Reste

1. Multiplikativität: $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$
2. $(QR_p, \cdot)$ ist eine multiplikative Gruppe.

Das Jacobi Symbol

Definition Jacobi Symbol

Sei $N = p_1^{e_1} \cdot \ldots \cdot p_k^{e_k} \in \mathbb{N}$ ungerade und $a \in \mathbb{N}$. Dann ist das *Jacobi Symbol* definiert als

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{e_i} \cdot \ldots \cdot \left(\frac{a}{p_k}\right)^{e_k}.$$

- **Warnung:** $\left(\frac{a}{N}\right) = 1$ impliziert nicht, dass $a \in QR_N$ ist.
- Bsp: $\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right) \cdot \left(\frac{2}{5}\right) = (-1)(-1) = 1$.
- D.h. $2 \in QNR_3$ und $2 \in QNR_5$. Damit besitzt $x^2 = 2$ weder Lösungen modulo 3 noch modulo 5.
- Nach CRT besitzt $x^2 = 2 \mod 15$ ebenfalls keine Lösung.

Pseudoquadrate

Berechnung des Jacobi-Symbols: Sei $a \in \mathbb{Z}_N$.

- Berechnung von $\left(\frac{a}{N}\right)$ ist in Zeit $\log^2(N)$ möglich, **ohne** die Faktorisierung von N zu kennen.
- Algorithmus ist ähnlich zum Euklidischen Algorithmus, verwendet das Gaußsche Reziprozitätsgesetz.

Definition Pseudoquadrat

Sei $N \in \mathbb{N}$. Die Menge der *Pseudoquadrate* ist definiert als

$$QNR_N^{+1} = \{a \in \mathbb{Z}_N^* \mid \left(\frac{a}{N}\right) = 1 \text{ und } a \notin QR_N\}.$$

Multiplikation von Resten/Nichtresten

Lemma Multiplikation von Resten/Nichtresten

Sei $N = pq$ ein RSA-Modul. Seien $x, x' \in QR_N$ und $y, y' \in QNR_N^{+1}$.

1. $xx' \in QR_N$
2. $yy' \in QR_N$
3. $xy \in QNR_N^{+1}$

Beweis: für 3 (1+2 folgen analog)

- Nach Chinesischem Restsatz gilt
$$QR_N \simeq QR_p \times QR_q \text{ und } QNR_N^{+1} \simeq QNR_p \times QNR_q.$$
- Aus der Multiplikativität des Legendre-Symbols folgt
$$\left(\frac{xy}{N}\right) = \left(\frac{xy}{p}\right)\left(\frac{xy}{q}\right) = \left(\frac{x}{p}\right)\left(\frac{x}{q}\right)\left(\frac{y}{p}\right)\left(\frac{y}{q}\right) = 1 \cdot 1 \cdot (-1) \cdot (-1) = 1.$$
- Analog gilt
$$\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right) = (-1).$$
- Daraus folgt $xy \in QNR_N^{+1}$.