RSA Full Domain Hash (RSA-FDH) Signaturen

Signatur RSA-FDH

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

1. **Gen:** $(N, e, d) \leftarrow GenRSA(1^n)$ mit $pk = (N, e)$ und $sk = (N, d)$.
2. **Sign:** Für eine Nachricht $m \in \{0,1\}^*$ berechne
$$\sigma \leftarrow H(m)^d \bmod N.$$
3. **Vrfy:** 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
     $$H(m_i) = \begin{cases} \sigma_i^e \bmod N \text{ für ein selbst gewähltes } \sigma \in_R \mathbb{Z}_N^* & \text{für } i \neq j \\ y & \text{sonst} \end{cases}.$$
   - Beantworte Orakelanfragen m_i an $Sign_{sk}(\cdot)$ mit
     $$Sign_{sk}(H(m_i)) = \begin{cases} \sigma_i & \text{für } i \neq j \\ \text{Abbruch} & \text{sonst} \end{cases}.$$
3. Falls $m = m_j$ und $\sigma^e = y \bmod N$, setze $x \leftarrow \sigma$.

AUSGABE: x

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

Hash-and-Sign Paradigma

Ziel: Signaturen für Nachrichten beliebiger Länge

- Starten mit Signaturverfahren Π für $m \in \{0,1\}^n$.
- Verwenden Hashfunktion $H : \{0,1\}^* \to \{0,1\}^n$.
- Unterschreiben Hashwerte statt der Nachrichten.

Definition Hash-and-Sign Paradigma

Sei $\Pi = (Gen, Sign, Vrfy)$ und $\Pi_H = (Gen_H, H)$ eine Hashfunktion.

1. **Gen':** $(pk, sk) \leftarrow Gen(1^n)$, $s \leftarrow Gen_H(1^n)$.
   Ausgabe $pk' = (pk, s)$ und $sk' = (sk, s)$.
2. **Sign':** Für eine Nachricht $m \in \{0,1\}^*$ berechne
   $$\sigma \leftarrow Sign_{sk}(H_s(m)).$$
3. **Vrfy':** Für eine Nachricht $m \in \{0,1\}^*$ mit Signatur σ prüfe
   $$Vrfy_{pk}(H_s(m), \sigma) \stackrel{?}{=} 1.$$

Intuition: Fälschung impliziert Fälschung in Π oder Kollision in H.

Sicherheit von Hash-and-Sign

Satz Sicherheit des Hash-and-Sign Paradigmas

Sei Π CMA-sicher und Π_H kollisionsresistent. Dann ist das Hash-and-Sign Signaturverfahren Π' CMA-sicher.

Beweis:

- Sei $\mathcal{A}$ ein Angreifer für Hash-and-Sign Π' mit Ausgabe (m, σ).
- Sei $Q = \{m_1, \ldots, m_q\}$ die Menge der von $\mathcal{A}$ an das Signierorakel $Sign_{sk}(\cdot)$ gestellten Anfragen. Es gilt $m \notin Q$.
- Sei *coll* das Ereignis, dass $m_i \in Q$ mit $H_s(m_i) = H_s(m)$.
- Dann gilt $\mathrm{Ws}[Forge_{\mathcal{A},\Pi'}(n) = 1]$

$$\begin{aligned} &= \mathrm{Ws}[Forge_{\mathcal{A},\Pi'}(n) = 1 \wedge coll] + \mathrm{Ws}[Forge_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{coll}] \\ &\leq \mathrm{Ws}[coll] + \mathrm{Ws}[Forge_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{coll}] \end{aligned}$$

- Wir zeigen nun, dass beide Summanden vernachlässigbar sind.

Algorithmus für Kollisionen

Beweis: $Ws[coll] \leq \text{negl}(n)$

- Konstruieren mittels $\mathcal{A}$ einen Algorithmus C für Kollisionen.

Algorithmus C

EINGABE: s

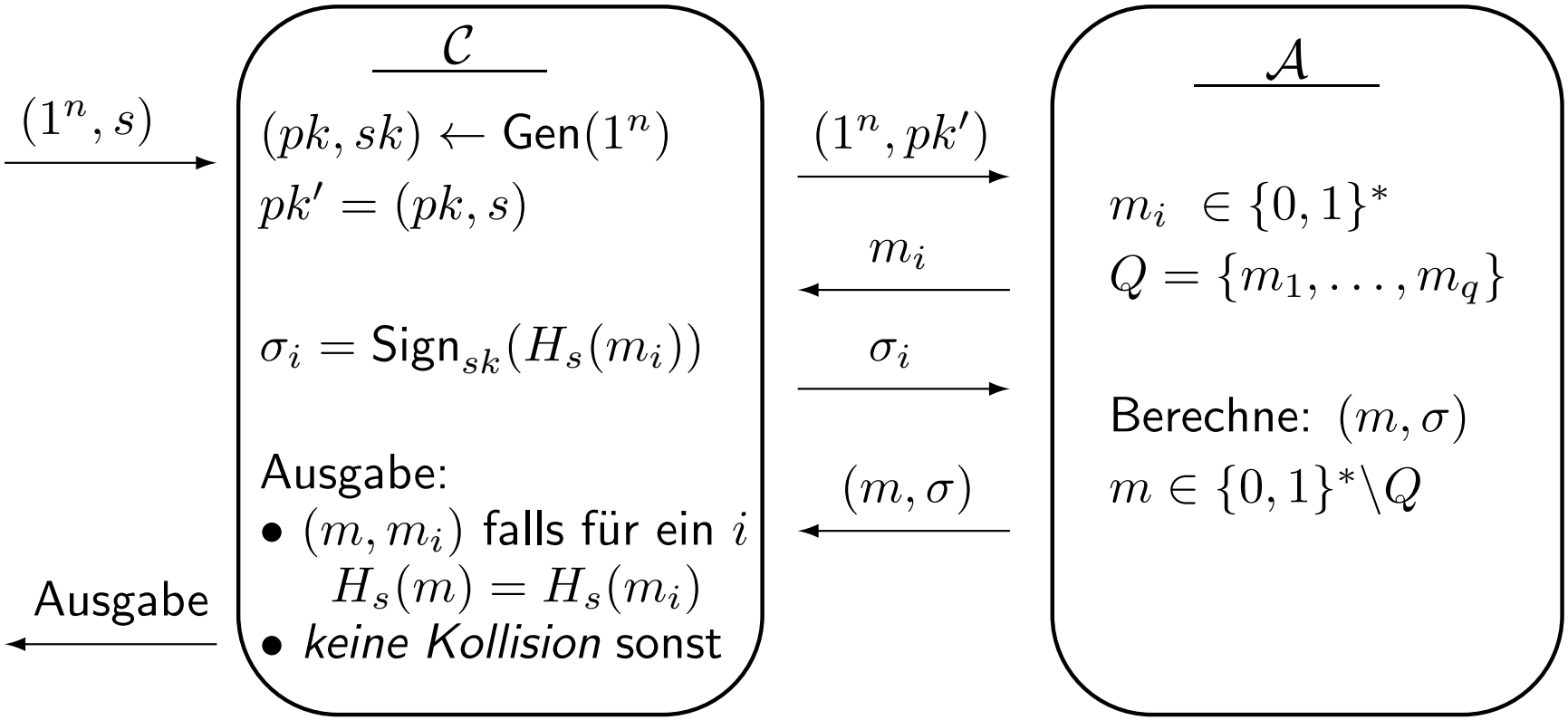
1. Berechne $(pk, sk) \leftarrow Gen(1^n)$. Setze $pk' \leftarrow (pk, s)$.
2. $(m, \sigma) \leftarrow \mathcal{A}(pk')$. Auf Orakelanfrage $m_i \in \{0,1\}^*$, antworte mit $\sigma_i \leftarrow Sign_{sk}(H_s(m_i))$.

AUSGABE: $\begin{cases} (m, m_i) & \text{falls } H_s(m) = H_s(m_i) \text{ für ein } m_i \\ \text{keine Kollision} & \text{sonst} \end{cases}$.

- Es gilt $\text{Ws}[coll] = \text{Ws}[HashColl_{C,\Pi_H}(n) = 1]$.
- Aus der Kollisionsresistenz von H folgt

$$\text{Ws}[HashColl_{C,\Pi_H}(n) = 1] \leq \text{negl}(n).$$

Algorithmus $\mathcal{C}$ für Kollisionen



Fälschen von Signaturen in Π

Beweis: $Ws[Forge_{\mathcal{A},\Pi'}(n) = 1 \wedge \overline{coll}] \leq \text{negl}(n)$

- Konstruieren mittels $\mathcal{A}$ einen Angreifer $\mathcal{A}'$ für Π.

Algorithmus $\mathcal{A}'$

EINGABE: pk, Zugriff auf Signierorakel $Sign_{sk}(\cdot)$

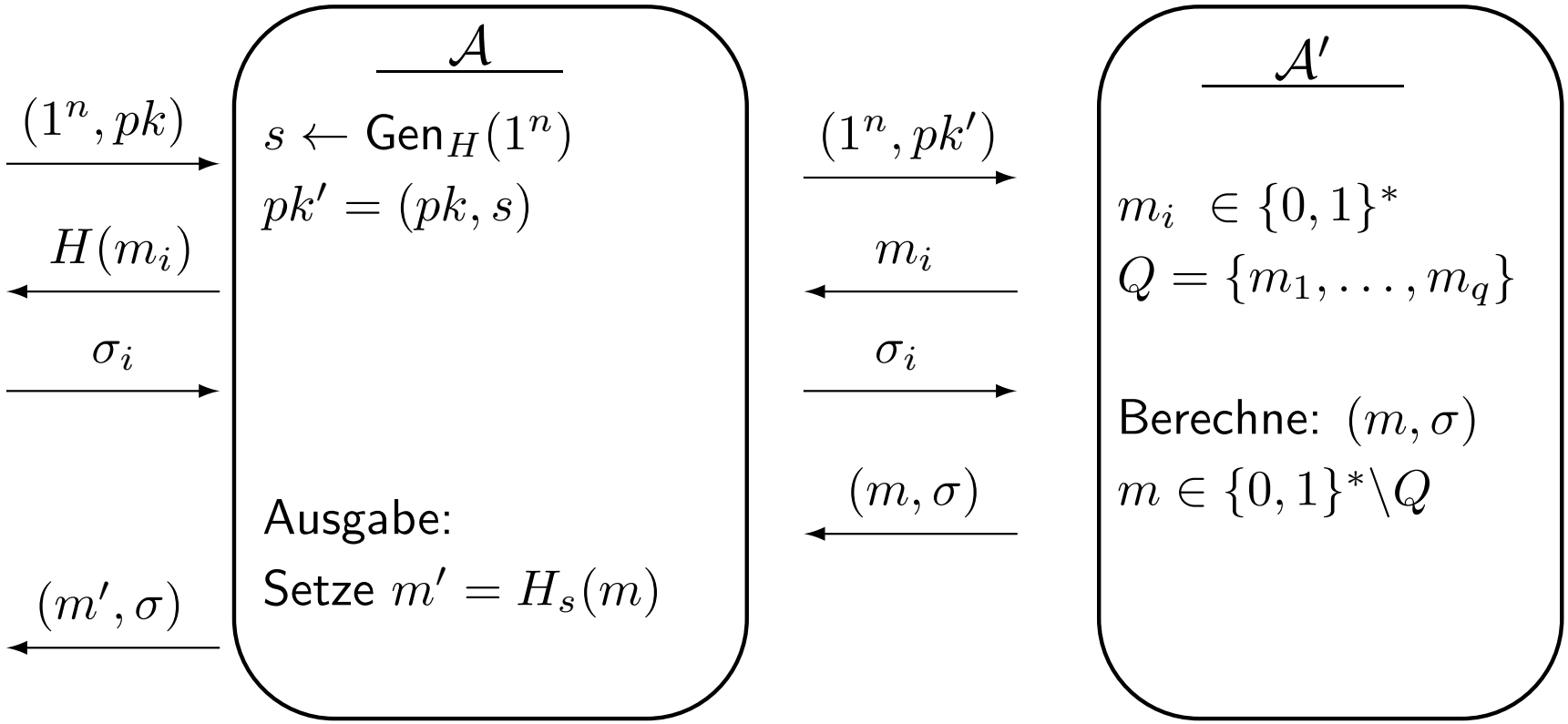
1. Berechne $s \leftarrow Gen_H(1^n)$. Setze $pk' = (pk, s)$.
2. $(m, \sigma) \leftarrow \mathcal{A}(pk')$. Beantworte Orakelanfrage $m_i \in \{0,1\}^*$ mit Ausgabe $\sigma_i \leftarrow Sign_{sk}(H_s(m_i))$ des Signierorakels.
3. Setze $m' \leftarrow H_s(m)$.

AUSGABE: (m', σ)

- Falls (m, σ) gültig ist für Π′, so ist $(m', \sigma) = (H_s(m), \sigma)$ gültig für Π.
- Ereignis $\overline{coll}$ bedeutet, dass $m' \neq H_s(m_i)$ für alle Anfragen $H_s(m_i)$.
- Damit gilt $\text{Ws}[Forge_{\mathcal{A},\Pi'}(n) \wedge \overline{coll}] = \text{Ws}[Forge_{\mathcal{A}',\Pi}(n) = 1]$.
- Aus der CMA-Sicherheit von Π folgt

$$\text{Ws}[Forge_{\mathcal{A}',\Pi}(n) = 1] \leq \text{negl}(n).$$

Algorithmus $\mathcal{A}$ für Fälschungen



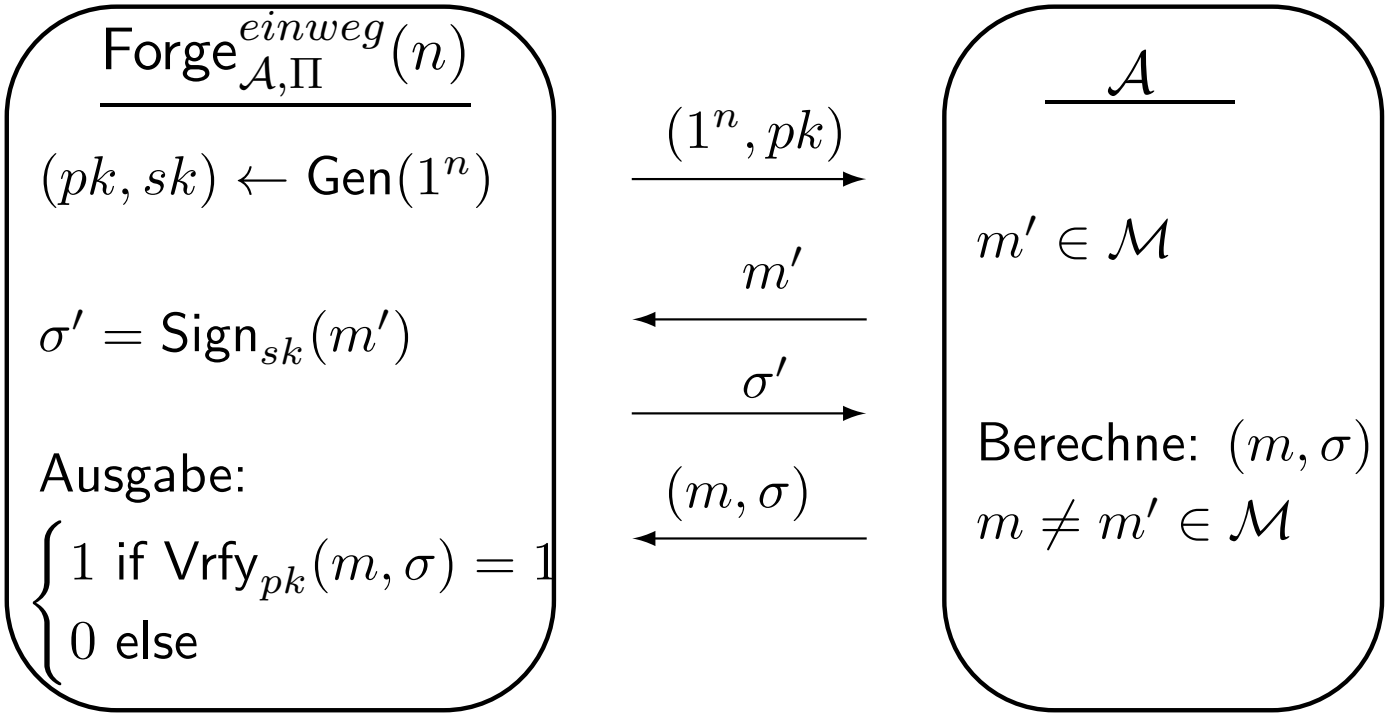
Einwegsignaturen

Ziel: Einwegsignaturen

- Konstruieren Verfahren zum sicheren Signieren *einer* Nachricht.
- Konstruktion mittels kollisionsresistenter Hashfunktionen.

Spiel $Forge^{einweg}_{\mathcal{A},\Pi}(n)$

1. $(pk, sk) \leftarrow Gen(1^n)$
2. $(m, \sigma) \leftarrow \mathcal{A}^{Sign_{sk}(\cdot)}(pk)$, wobei $\mathcal{A}$ **eine** Nachricht $m' \neq m$ an $Sign_{sk}(\cdot)$ anfragen darf.
3. $Forge^{einweg}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } Vrfy_{pk}(m, \sigma) = 1 \\ 0 & \text{sonst} \end{cases}$.



Definition CMA-sichere Einwegsignaturen

Ein Signaturverfahren Π heißt *CMA-sichere Einwegsignatur*, falls für alle ppt $\mathcal{A}$ gilt $\Pr[\mathit{Forge}^{\mathit{einweg}}_{\mathcal{A},\Pi}(n) = 1] \leq \mathrm{negl}(n)$.

Beispiel von Lamports Einwegsignaturen

Illustration: Signieren einer 3-Bit Nachricht

- Verwende Einwegfunktion $f : D \to R$.
- Wähle als geheimen Schlüssel 6 Element $x_{i,j}$ zufällig aus D. Setze

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ x_{1,1} & x_{2,1} & x_{3,1} \end{pmatrix}.$$

- Für alle $x_{i,j}$ berechne $y_{i,j} = f(x_{i,j})$. Dies liefert

$$pk = \begin{pmatrix} y_{1,0} & y_{2,0} & y_{3,0} \\ y_{1,1} & y_{2,1} & y_{3,1} \end{pmatrix}.$$

- Unterschreibe $m = m_1 m_2 m_3 \in \{0,1\}^3$ mit $\sigma = x_{1,m_1} x_{2,m_2} x_{3,m_3}$.
- Verifikation von (m, σ): Überprüfe $f(\sigma_i) = y_{i,m_i}$ für $i = 1, 2, 3$.

Lamports Einwegsignaturen

Definition Lamport Einwegsignaturen

Sei f eine Einwegfunktion. Konstruieren Signaturen für $m \in \{0,1\}^\ell$.

- **Gen:** Bei Eingabe 1^n:
  Wähle $x_{i,j} \in_R \{0,1\}^n$, berechne $y := f(x_{i,j})$ für $i \in [\ell], j \in \{0,1\}$.

$$\text{Setze } sk = \begin{pmatrix} x_{1,0} & \dots & x_{\ell,0} \\ x_{1,1} & \dots & x_{\ell,1} \end{pmatrix} \text{ und } pk = \begin{pmatrix} y_{1,0} & \dots & y_{\ell,0} \\ y_{1,1} & \dots & y_{\ell,1} \end{pmatrix}.$$

- **Sign:** Für $m_1 \dots m_\ell \in \{0,1\}^\ell$, Ausgabe (m, σ) mit
$$\sigma = (x_{1,m_1}, \dots, x_{\ell,m_\ell}).$$
- **Vrfy:** Für (m, σ) überprüfe $f(\sigma_i) \stackrel{?}{=} y_{i,m_i}$ für $i \in [\ell]$.