Definition Signaturverfahren

Definition Signaturverfahren

Ein *Signaturverfahren* ist ein 3-Tupel $(Gen, Sign, Vrfy)$ von ppt Alg mit

1. **Gen:** $(pk, sk) \leftarrow Gen(1^n)$.
2. **Sign:** $\sigma \leftarrow Sign_{sk}(m)$ für $m \in \{0,1\}^*$.
3. **Vrfy:** $Vrfy_{pk}(m,\sigma) = \begin{cases} 1 & \text{falls } \sigma \text{ gültig für } m \text{ ist.} \\ 0 & \text{sonst} \end{cases}$.

   Es gilt $Vrfy_{pk}(m, Sign_{sk}(m)) = 1$ für alle $m \in \{0,1\}^*$.

Unfälschbarkeit von Signaturen

Spiel CMA-Spiel $Forge_{\mathcal{A},\Pi}(n)$

Sei Π ein Signaturverfahren mit Angreifer $\mathcal{A}$.

1. $(pk, sk) \leftarrow Gen(1^n)$
2. $(m, \sigma) \leftarrow \mathcal{A}^{Sign_{sk}(\cdot)}(pk)$, wobei $Sign_{sk}(\cdot)$ ein Signierorakel für beliebige Nachrichten $m' \neq m$ ist.
3. $$Forge_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } Vrfy_{pk}(m,\sigma) = 1, Sign_{sk}(m) \text{ nicht angefragt} \\ 0 & \text{sonst} \end{cases}.$$

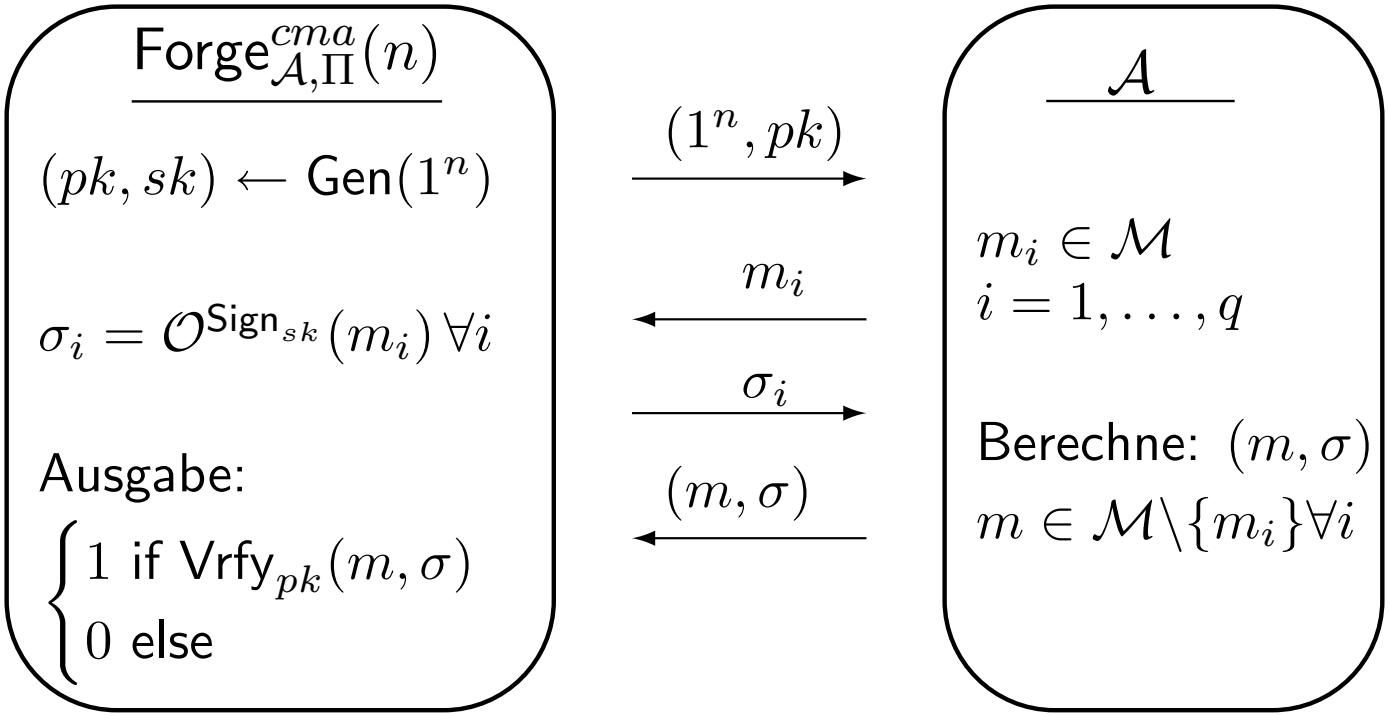
Definition CMA-Sicherheit

Sei Π ein Signaturverfahren. Π heißt *existentiell unfälschbar* unter *Chosen Message Angriffen (CMA)*, falls für alle ppt Angreifer $\mathcal{A}$ gilt

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

Wir bezeichnen Π auch abkürzend als *CMA-sicher*.

CMA Spiel Forge



Unsicherheit von Textbook RSA Signaturen

Algorithmus Textbook RSA Signaturen

1. **Gen:** $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$. Setze $pk = (N, e)$, $sk = (N, d)$.
2. **Sign:** Für $m \in \mathbb{Z}_N$ berechne $\sigma = m^d \bmod N$.
3. **Vrfy:** Für $(m, \sigma) \in \mathbb{Z}_N^2$ Ausgabe 1 gdw $\sigma^e \stackrel{?}{=} m \bmod N$.

Unsicherheit: gegenüber CMA-Angriffen

- Wähle beliebiges $\sigma \in \mathbb{Z}_N$. Berechne $m \leftarrow \sigma^e \bmod N$.
- Offenbar ist σ eine gültige Signatur für m.
- Angreifer besitzt keine Kontrolle über m (existentielle Fälschung).

Fälschen einer Signatur für ein gewähltes $m \in \mathbb{Z}_N$:

- Wähle $m_1 \in_R \mathbb{Z}_N^*$ mit $m_1 \neq m$. Berechne $m_2 = \frac{m}{m_1} \bmod N$.
- Lasse m_1, m_2 vom Orakel $\mathit{Sign}_{sk}(\cdot)$ unterschreiben.
- Seien σ_1, σ_2 die Signaturen. Dann ist $\sigma := \sigma_1 \cdot \sigma_2 = m_1^d \cdot m_2^d = (m_1 m_2)^d = m^d \bmod N$ gültig für m.

Hashfunktionen und Kollisionen

Definition Hashfunktion

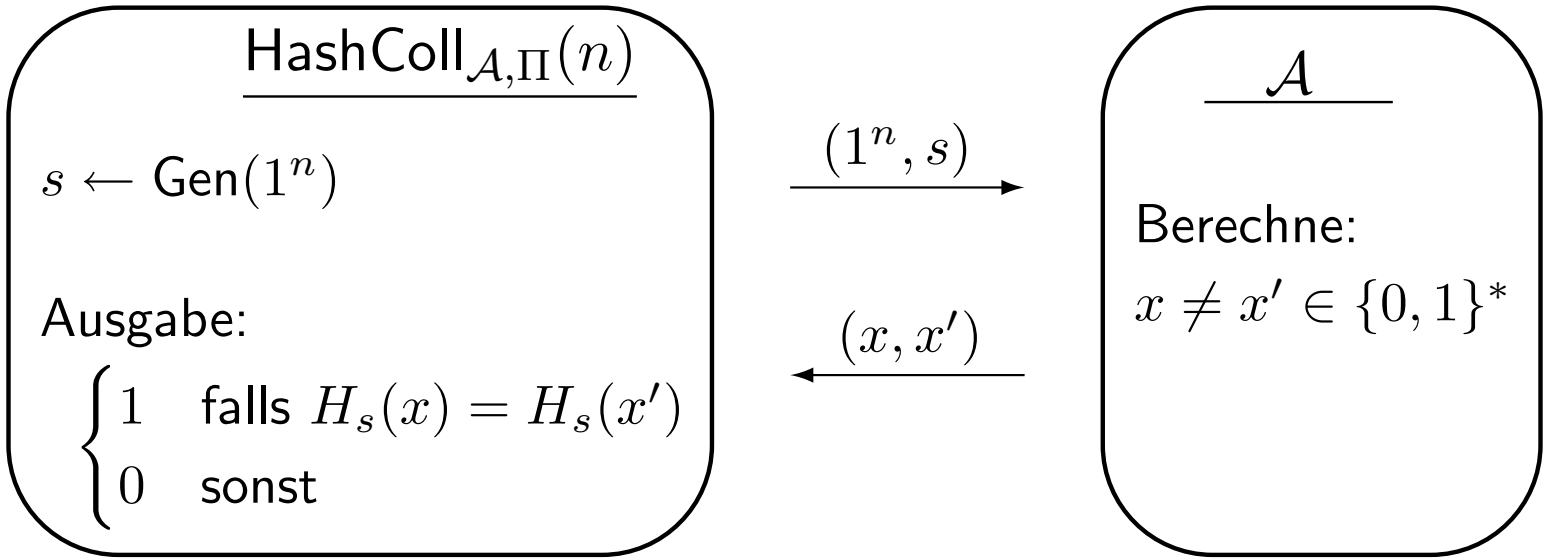
Eine *Hashfunktion* ist ein Paar (Gen, H) von pt Algorithmen mit

1. **Gen:** $s \leftarrow Gen(1^n)$. *Gen* ist probabilistisch.
2. **H:** $\{0,1\}^n \leftarrow H_s(x)$ für alle $x \in \{0,1\}^*$. *H* ist deterministisch.

Spiel $HashColl_{\mathcal{A},\Pi}(n)$

1. $s \leftarrow Gen(1^n)$
2. $(x, x') \leftarrow \mathcal{A}(s)$
3. $$HashColl_{\mathcal{A},\Pi} = \begin{cases} 1 & \text{falls } H_s(x) = H_s(x') \text{ und } x \neq x' \\ 0 & \text{sonst} \end{cases}.$$

Kollisionsresistente Hashfunktionen



Definition Kollisionsresistenz

Eine Hashfunktion Π heißt *kollisionsresistent*, falls für alle ppt $\mathcal{A}$ gilt $\mathrm{Ws}[\mathit{HashColl}_{\mathcal{A},\Pi}(n) = 1] \leq \mathrm{negl}(n)$.

Hashed RSA

Algorithmus Hashed RSA

1. **Gen:** $(N, e, d, H) \leftarrow \mathit{GenHashRSA}(1^n)$ mit $H : \{0,1\}^* \to \mathbb{Z}_N$. Ausgabe $pk = (N, e, H)$, $sk = (N, d, H)$.
2. **Sign:** Für $m \in \{0,1\}^*$ berechne $\sigma = H(m)^d \bmod N$.
3. **Vrfy:** Für $(m, \sigma) \in \mathbb{Z}_N^2$ Ausgabe 1 gdw $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

Einfacher Angriff:

- Sei $m_1 \neq m_2$ eine Kollision für H ist, d.h. $H(m_1) = H(m_2)$.
- Frage (m_1, σ) an. Dann ist (m_2, σ) eine gültige Fälschung.
- D.h. wir benötigen für H Kollisionsresistenz.

Anmerkung: Sicherheit gegen unsere Angriffe für Textbook RSA

1. Wähle $\sigma \in \mathbb{Z}_N$, $m' \leftarrow \sigma^e$. Müssen $m \in H^{-1}(m')$ bestimmen. Übung: Urbildbestimmung ist schwer für kollisionsresistentes H.
2. Für ein $m \in \mathbb{Z}_N^*$ benötigen wir m_1, m_2 mit $H(m) = H(m_1) \cdot H(m_2)$ in $\mathbb{Z}_n$. Scheint Invertierbarkeit von H zu erfordern.

Später: Zeigen CMA-Sicherheit einer Hashed RSA Variante. (im ROM)

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 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

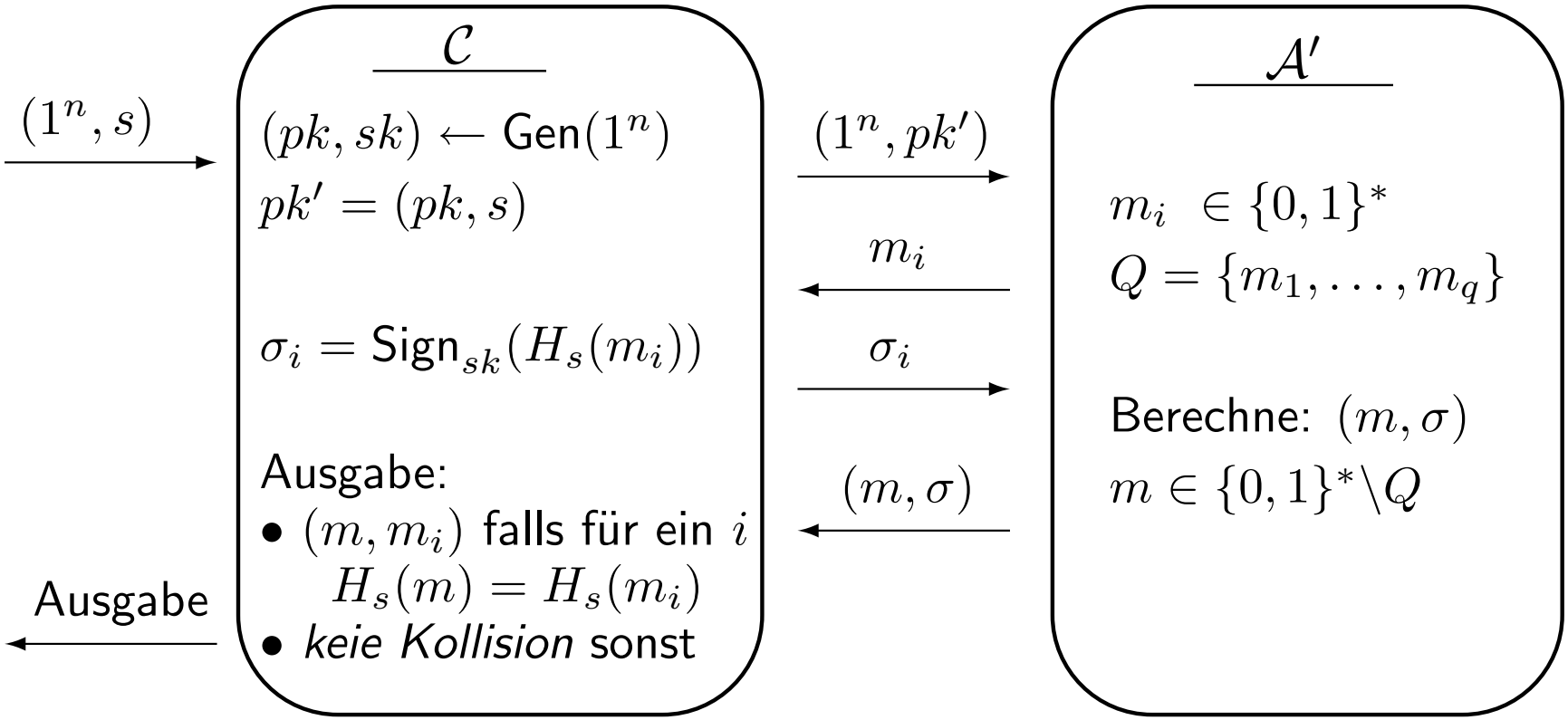
1. Berechne $(pk, sk) \leftarrow Gen(1^n)$. Setze $pk' \leftarrow (pk, s)$.
2. $(m, \sigma) \leftarrow \mathcal{A}'(pk')$. Auf Orakelanfrage $m_i \in \{0,1\}^*$, antworte mit $\sigma_i \leftarrow Sign_{sk}(H_s(m_i))$.

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

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

Algorithmus $\mathcal{A}$

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

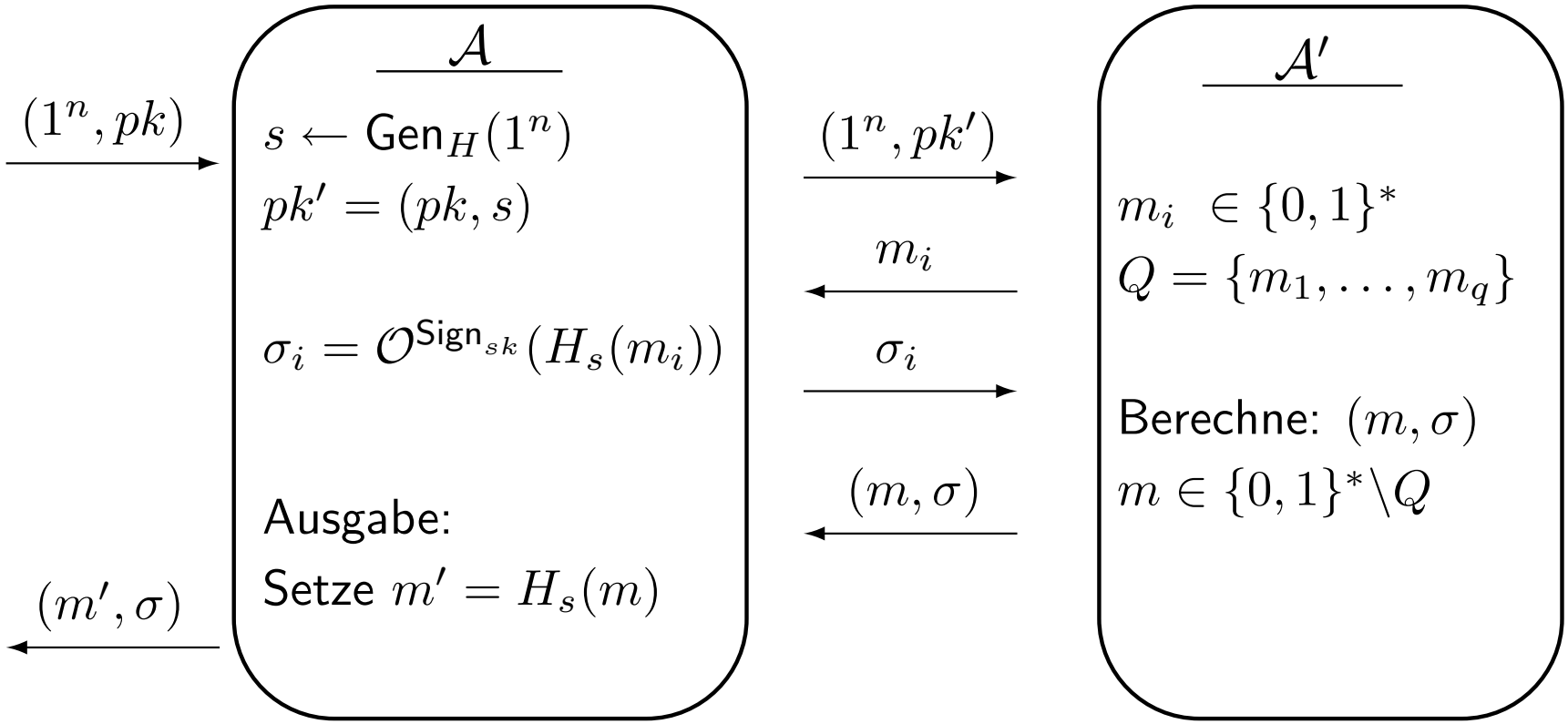
1. Berechne $s \leftarrow Gen_H(1^n)$. Setze $pk' = (pk, s)$.
2. $(m, \sigma) \leftarrow \mathcal{A}'(pk')$. Beantworte Orakelanfrage $m_i \in \{0,1\}^*$ mit Ausgabe $\sigma_i \leftarrow Sign_{sk}(H_s(m_i))$ des Signierorakels.
3. Setze $m' \leftarrow H_s(m)$.

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 $\mathrm{Ws}[Forge_{\mathcal{A}',\Pi'}(n) \wedge \overline{coll}] = \mathrm{Ws}[Forge_{\mathcal{A},\Pi}(n) = 1]$.
- Aus der CMA-Sicherheit von Π folgt

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

Algorithmus $\mathcal{A}$ für Fälschungen



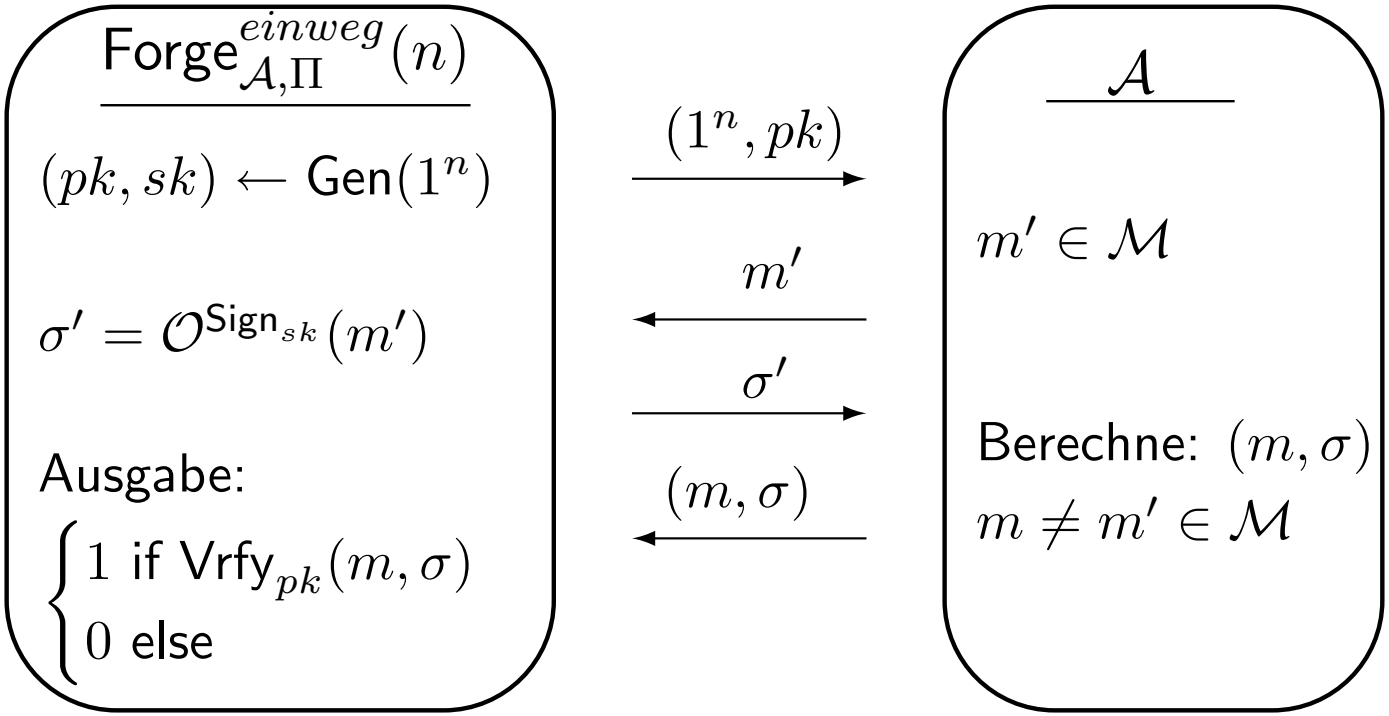
Einwegsignaturen

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 $y \leftarrow f(x_{i,j})$ 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 $i \in [\ell]$.

Sicherheit von Lamport Einwegsignaturen

Satz CMA-Sicherheit von Lamport

Lamport Einwegsignaturen sind CMA-sicher, falls die Nachrichtenlänge ℓ polynomiell in n und f eine Einwegfunktion ist.

Beweis: Sei Π das Lamport Signaturverfahren.

- Sei $\mathcal{A}$ ein Angreifer mit $\epsilon(n) := \mathrm{Ws}[Forge^{einweg}_{\mathcal{A},\Pi}(n) = 1]$.
- Wir konstruieren einen Invertierer für f mittels $\mathcal{A}$.

Algorithmus Invertierer *I*

EINGABE: y

1. Wähle $i \in_R \{0,1\}^\ell$ und $b \in_R \{0,1\}$.
2. Berechne $(pk, sk) \leftarrow Gen_\Pi(1^n)$. Setze $y_{i,b} \leftarrow y$ in pk.
3. $(m, \sigma) \leftarrow \mathcal{A}$. Bei Signaturanfrage für m' antworte mit $\sigma = (\sigma_{1,m'_1}, \ldots, \sigma_{\ell,m'_\ell})$ falls $m'_i \neq b$. Sonst Abbruch.

$$\text{AUSGABE:} = \begin{cases} x_i & \text{falls } m_i \neq m'_i \\ \text{Abbruch} & \text{sonst} \end{cases}.$$

Sicherheit von Lamport Einwegsignaturen

Beweis: Fortsetzung

- Sei (m, σ) eine gültige Signatur mit $m_i = b$.
- Dann ist σ_i ein Urbild von y, d.h. $f(\sigma_i) = y$.
- Wahl von y im Invertier-Spiel erfolgt durch $x \in_R D$ und $y \leftarrow f(x)$.
- D.h. pk ist identisch verteilt zum Lamport Signaturverfahren.

- Benötigen $m'_i \neq b$ und $m'_i \neq m_i$. Es gilt $\mathrm{Ws}[m'_i \neq b] = \frac{1}{2}$.
- Wegen $m \neq m'$ folgt $\mathrm{Ws}[m'_i \neq m_i] \geq \frac{1}{\ell}$.
- Wir erhalten insgesamt $\mathrm{Ws}[\mathit{Invert}_{y,f}(n) = 1]$

$$\begin{aligned} &= \mathrm{Ws}[\mathit{Forge}^{\mathit{einweg}}_{\mathcal{A},\Pi}(n) \wedge (m'_i \neq b) \wedge (m'_i \neq m_i)] \\ &= \mathrm{Ws}[\mathit{Forge}^{\mathit{einweg}}_{\mathcal{A},\Pi}(n)] \cdot \mathrm{Ws}[(m'_i \neq b)] \cdot \mathrm{Ws}[(m'_i \neq m_i)] \geq \epsilon(n) \cdot \frac{1}{2\ell} \end{aligned}$$

- Aufgrund der Einweg-Eigenschaft von f gilt

$$\mathrm{negl}(n) \geq \mathrm{Ws}[\mathit{Invert}_{y,f}(n) = 1].$$

- Daraus folgt $\epsilon(n) \leq 2\ell \cdot \mathrm{negl}(n)$.
- Dies ist vernachlässigbar für polynomielles ℓ.