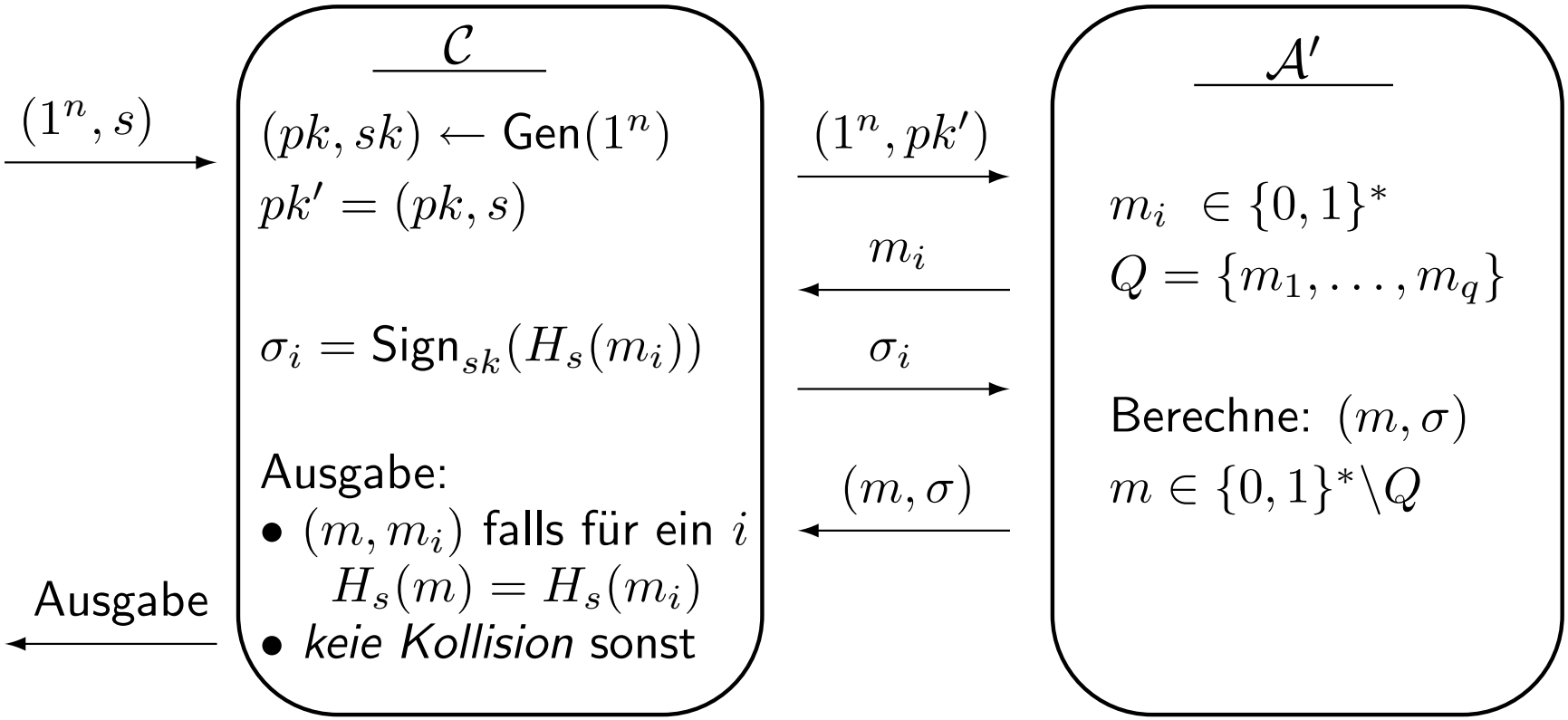
Algorithmus $\mathcal{C}$ für Kollisionen



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 $Sign_{sk}(\cdot)$

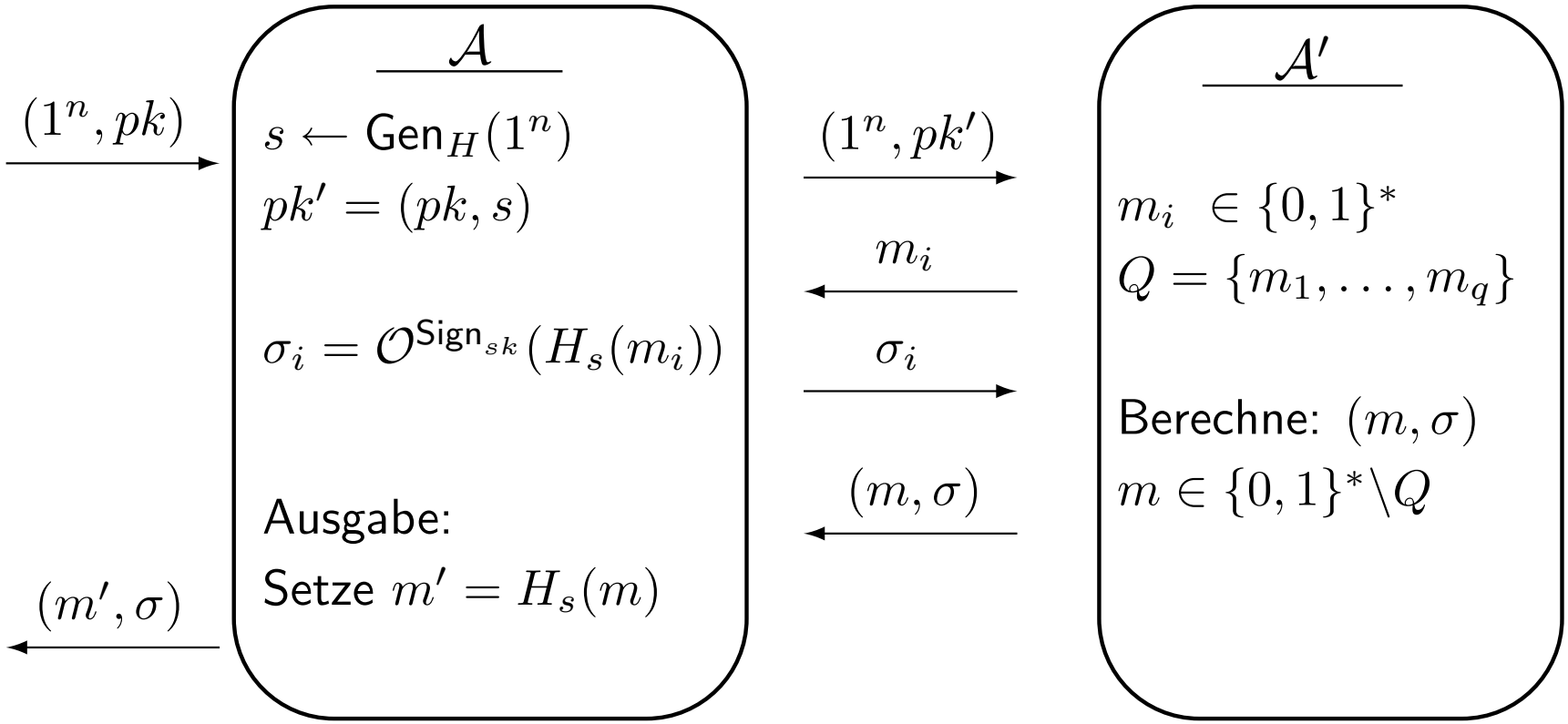
1. Berechne $s \leftarrow Gen_H(1^n)$. Setze $pk' = (pk, s)$.
2. $(m, \sigma) \leftarrow \mathcal{A}'(pk')$. Beantworte Orakelanfrage $m_i \in \{0,1\}^*$ mit Ausgabe $\sigma_i \leftarrow Sign_{sk}(H_s(m_i))$ des Signierorakels.
3. Setze $m' \leftarrow H_s(m)$.

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



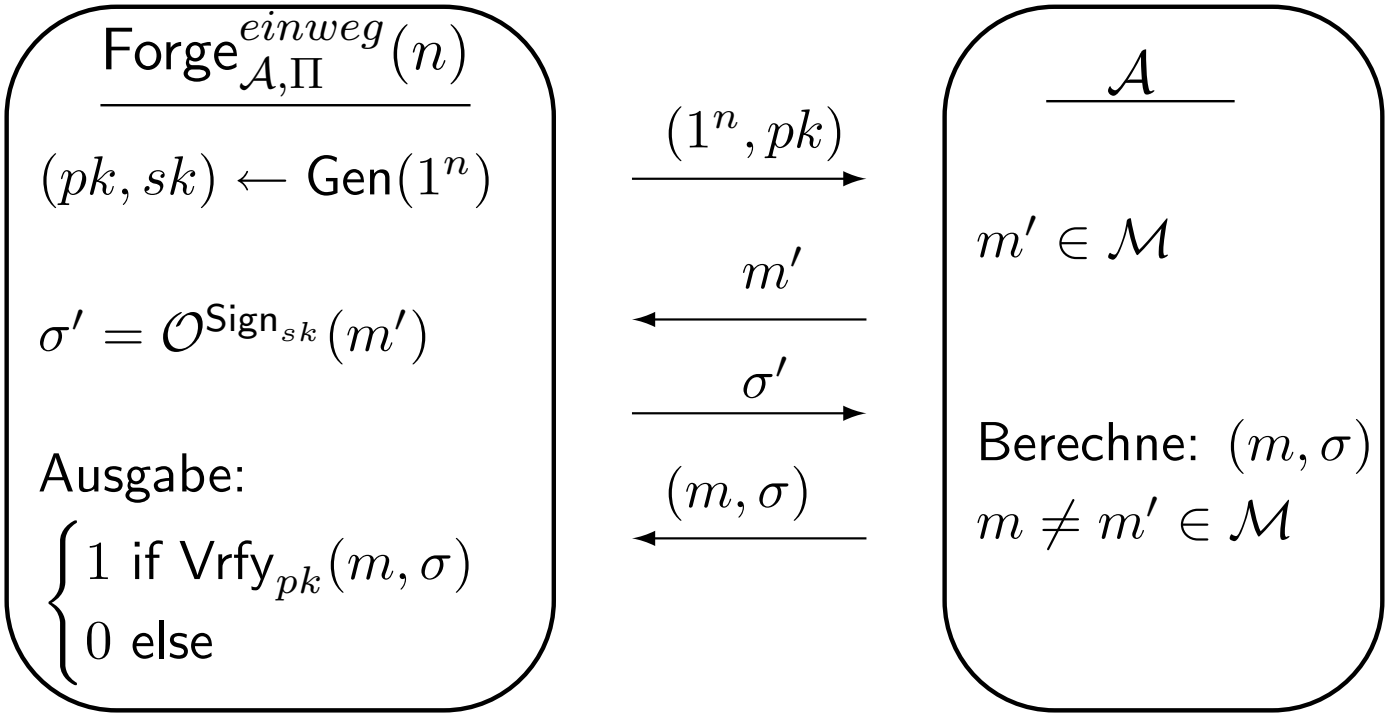
Einwegsignaturen

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.

Einwegsignaturen für Nachrichten beliebiger Länge

Satz Einwegsignaturen für Nachrichten beliebiger Länge

Unter der Annahme kollisionsresistenter Hashfunktionen existiert ein CMA-sicheres Einwegsignatur-Verfahren für Nachrichten beliebiger Länge.

Beweisskizze:

- Aus der Existenz von kollisionsresistenten Hashfunktionen folgt die Existenz von Einwegfunktionen. (Übung)
- Nutzen Einwegfunktion f zur Konstruktion von CMA-sicheren Lamport-Signaturen der Nachrichtenlänge ℓ.
- Nutzen Hash-and-Sign Paradigma für beliebige Nachrichtenlänge. Hier verwenden wir erneut kollisionsresistente Hashfunktionen.

Einfache k-wegsignaturen

k-wegsignaturen

- Definiere mittels k-maliger Anwendung von $Gen_{\text{Lamport}}(1^n)$

$$pk = (pk_1, \ldots, pk_k) \text{ und } sk = (sk_1, \ldots, sk_k).$$

- Unterzeichnen die i-te Nachricht m mittels sk_i als (m, σ).
- Man bezeichnet i auch als *Zustand* im Signaturverfahren.
- (m, σ) gilt als gültig, falls (m, σ) für ein pk_i gültig ist.

Nachteile:

- Anzahl k muss bei Schlüsselgenerierung feststehen.
- Länge von pk und sk hängen beide von k ab.

Idee:

- Konstruiere neues Schlüsselpaar nur bei Bedarf.
- Validiere (pk_{i+1}, sk_{i+1}) mittels (pk_i, sk_i).

Zwei Signaturen mit einem öffentlichen Schlüssel

- Sei $\Pi = (Gen, Sign, Vrfy)$ ein Einwegsignaturverfahren.
- Konstruieren $\Pi' = (Gen', Sign', Vrfy)$ für zwei Signaturen.

Gen':

- Erzeuge $(pk_1, sk_1) \leftarrow Gen(1^n)$.

Sign' und Vrfy' von m_1:

- Erzeuge (pk_2, sk_2). Berechne $\sigma_1' \leftarrow Sign_{sk_1}(m_1||pk_2)$.
- Ausgabe der Signatur $\sigma_1 = (pk_2, \sigma_1')$. Merke (m_1, σ_1, sk_2).
- Verifikation von $(m_1, \sigma_1) = (m_1, pk_2, Sign_{sk_1}(m_1||pk_2))$:

$$\text{Überprüfe } Vrfy_{pk_1}(m_1||pk_2, \sigma_1') \stackrel{?}{=} 1.$$

Sign' und Vrfy' von m_2:

- Erzeuge (pk_3, sk_3). Berechne $\sigma_2' \leftarrow Sign_{sk_2}(m_2||pk_3)$.
- Ausgabe $\sigma_2 = (m_1, \sigma_1, pk_3, \sigma_2')$. Merke (m_2, σ_2, sk_3)
- Verifikation von $(m_2, \sigma_2) = (m_2, m_1, pk_2, \sigma_1', pk_3, \sigma_2')$:

$$\text{Überprüfe } Vrfy_{pk_1}(m_1||pk_2, \sigma_1') \stackrel{?}{=} 1 \textbf{ und } Vrfy_{pk_2}(m_2||pk_3, \sigma_2') \stackrel{?}{=} 1.$$