

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

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

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

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

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

Einwegsignaturen

Ziel: Einwegsignaturen

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

Spiel $Forge_{\mathcal{A}, \Pi}^{\text{einweg}}(n)$

- 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 $Forge_{\mathcal{A}, \Pi}^{\text{einweg}}(n) = \begin{cases} 1 & \text{falls } Vrfy_{pk}(m, \sigma) = 1 \\ 0 & \text{sonst} \end{cases}$.

Definition CMA-sichere Einwegsignaturen

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

Beispiel von Lamports Einwegsignaturen

Illustration: Signieren einer 3-Bit Nachricht

- Verwende Einwegfunktion $f : D \rightarrow R$.
- 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 $i \in [\ell], j \in \{0, 1\}$.

Setze $sk = \begin{pmatrix} x_{1,0} & \dots & x_{\ell,0} \\ x_{1,1} & \dots & x_{\ell,1} \end{pmatrix}$ 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 $\epsilon(n) := \text{Ws}[Forge_{\mathcal{A}, \Pi}^{\text{einweg}}(n) = 1]$.
- Wir konstruieren einen Invertierer für f mittels \mathcal{A} .

Algorithmus Invertierer /

EINGABE: y

- 1 Wähle $i \in_R \{0, 1\}^\ell$ und $b \in_R \{0, 1\}$.
- 2 Berechne $(pk, sk) \leftarrow \text{Gen}_\Pi(1^n)$. Setze $y_{i,b} \leftarrow y$ in pk .
- 3 $(m, \sigma) \leftarrow \mathcal{A}$. Bei Signaturanfrage für m' antworte mit $\sigma = (\sigma_1, m'_1, \dots, \sigma_\ell, m'_\ell)$ falls $m'_i \neq b$. Sonst Abbruch.

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 $\text{Ws}[m'_i \neq b] = \frac{1}{2}$.
- Wegen $m \neq m'$ folgt $\text{Ws}[m'_i \neq m_i] \geq \frac{1}{\ell}$.
- Wir erhalten insgesamt $\text{Ws}[Invert_{y,f}(n) = 1]$
 - $= \text{Ws}[Forge_{\mathcal{A},\Pi}^{einweg}(n) \wedge (m'_i \neq b) \wedge (m'_i \neq m_i)]$
 - $= \text{Ws}[Forge_{\mathcal{A},\Pi}^{einweg}(n)] \cdot \text{Ws}[(m'_i \neq b)] \cdot \text{Ws}[(m'_i \neq m_i)] \geq \epsilon(n) \cdot \frac{1}{2\ell}$
- Aufgrund der Einweg-Eigenschaft von f gilt
$$\text{negl}(n) \geq \text{Ws}[Invert_{y,f}(n) = 1].$$
- Daraus folgt $\epsilon(n) \leq 2\ell \cdot \text{negl}(n)$.
- Dies ist vernachlässigbar für polynomielles ℓ .