Übersicht Krypto II - Signatur

Abkürzungen:

- OWF = Einwegfunktion
- CRH = kollisionsresistente Hashfunktion

Funktionalität	**Sicherheit**	**Annahme**
Signatur *RSA-FDH*	CMA (ROM) $\sigma = H(m)^d$	TD-OWP *RSA-Annahme*
Einwegsignatur fester Länge *Lamport*	CMA $y_{i,j} = f(x_{i,j}), \sigma_i = x_{i,m_i}$	OWF
Einwegsignatur beliebiger Länge *Lamport + Hash&Sign*	CMA *Signiere $H(m)$.*	CRH
Signatur *Merkle*	CMA *Zertifiziere Baumpfad.*	CRH

Konstruktion aus Einwegpermutation

Wir wollen folgende Konstruktionen kurz skizzieren:

1. Einwegpermutation
   $\Rightarrow$ Einwegpermutation + Hardcore-Prädikat.
2. Einwegpermutation + Hardcore-Prädikat
   $\Rightarrow$ Pseudozufallsgenerator mit 1 Bit Expansion.
3. Pseudozufallsgenerator mit 1 Bit Expansion
   $\Rightarrow$ Pseudozufallsgenerator mit polynomieller Expansion.
4. Pseudozufallsgenerator mit polynomieller Expansion
   $\Rightarrow$ Pseudozufallsfunktion.

Einwegperm. $\Rightarrow$ Einwegperm. + Hardcore-Prädikat

Goldreich-Levin Hardcore-Prädikat

Sei $f : \{0,1\}^n \to \{0,1\}^n$ eine Einwegpermutation und

$$g : \{0,1\}^{2n} \to \{0,1\}^{2n}, (x,r) \mapsto (f(x),r).$$

Dann ist $hc(x,r) := \langle x,r \rangle = \sum_{i=1}^{n} x_i r_i$ ein Hardcore-Prädikat für g.

Beweis:

- Offenbar ist g ebenfalls eine Einwegpermutation.
- Sei $\mathcal{A}$ ein polynomieller Angreifer für das Hardcore-Prädikat hc.
- Vereinfachende Annahme: $\mathcal{A}$ besitze Erfolgsws

  $$\mathrm{Ws}_{x,r \in_R \{0,1\}^n}[\mathcal{A}(g(x),r) = hc(x,r)] = 1.$$

  (Der Beweis für Erfolgsws $\frac{1}{2} + \frac{1}{poly(n)}$ ist deutlich komplexer.)
- Konstruieren einen Angreifer $\mathcal{A}'$ zum Invertieren von f.

Goldreich-Levin Hardcore-Prädikat

Algorithmus $\mathcal{A}'$

EINGABE: $1^n, y = f(x) \in \{0,1\}^n$

- For $i = 1$ to n: Setze $x_i := \mathcal{A}(y, e_i)$ für den i-ten Einheitsvektor e_i.

AUSGABE: $x = x_1 \ldots x_n \in \{0,1\}^n$

Korrektheit und Laufzeit:

- Es gilt $hc(y, e_i) = hc(f(x), e_i) = \langle x, e_i \rangle = x_i$.
- Die Laufzeit von $\mathcal{A}'$ ist n-mal die Laufzeit von $\mathcal{A}$.

Konstruktion eines pseudozufälligen Bits

Idee: Gegeben sei $f(x)$. Dann ist $hc(x)$ ein pseudozufälliges Bit.

Satz

Sei $f : \{0,1\}^n \to \{0,1\}^n$ eine Einwegpermutation mit Hardcore-Prädikat hc. Dann ist

$$G : \{0,1\}^n \to \{0,1\}^{n+1}, x \mapsto (f(x), hc(x))$$

ein Pseudozufallsgenerator mit Expansionsfaktor $\ell(n) = n + 1$.

Beweis:

- Sei $\mathcal{A}$ ein Unterscheider für G mit Erfolgsws
$$\epsilon(n) = \mathrm{Ws}_{x \in_R \{0,1\}^n}[\mathcal{A}(G(x) = 1)] - \mathrm{Ws}_{r \in_R \{0,1\}^{n+1}}[\mathcal{A}(r) = 1].$$
- Konstruieren daraus Angreifer $\mathcal{A}'$ auf das Hardcore-Prädikat.

Beweis Pseudozufallsgenerator(1/3)

Algorithmus $\mathcal{A}'$

EINGABE: $1^n, y = f(x)$

1. Wähle $r \in_R \{0,1\}$. Berechne $b \leftarrow \mathcal{A}(y \,||\, r)$.
2. Setze $h(x) = \begin{cases} r & \text{für } b = 1 \\ 1 - r & \text{für } b = 0 \end{cases}$.

AUSGABE: $h(x)$

Fall 1: $r = hc(x)$

- Dann ist die Verteilung identisch zum Generator G, d.h.

$$\text{Ws}[\mathcal{A}'(f(x)) = hc(x) \mid r = hc(x)] = Ws[\mathcal{A}(G(x)) = 1].$$

Beweis Pseudozufallsgenerator (2/3)

Fall 2: $r \neq hc(x)$

- Dann gilt

$$\begin{aligned} \mathrm{Ws}[\mathcal{A}'(f(x)) = hc(x) \mid r \neq hc(x)] &= Ws[\mathcal{A}(f(x)||\overline{hc(x)}) = 0] \\ &= 1 - Ws[\mathcal{A}(f(x)||\overline{hc(x)}) = 1]. \end{aligned}$$

Unter der Annahme, dass hc ein Hardcore-Prädikat ist, gilt

$$\begin{aligned} & \tfrac{1}{2} + \mathrm{negl}(n) \\ \geq\ & \mathrm{Ws}[\mathcal{A}'(f(x)) = hc(x)] \\ =\ & \tfrac{1}{2} \cdot (\mathrm{Ws}[\mathcal{A}'(f(x)) = hc(x) \mid r = hc(x)] + \mathrm{Ws}[\mathcal{A}'(f(x)) = hc(x) \mid r \neq hc(x)]) \\ =\ & \tfrac{1}{2} \cdot \left(Ws[\mathcal{A}(G(x)) = 1] + 1 - Ws[\mathcal{A}(f(x)||\overline{hc(x)}) = 1] \right) \\ =\ & \tfrac{1}{2} + \underbrace{\frac{1}{2} \cdot \left(Ws[\mathcal{A}(G(x)) = 1] - Ws[\mathcal{A}(f(x)||\overline{hc(x)}) = 1] \right)}_{\text{zu zeigen: } \epsilon(n)} \end{aligned}$$

Daraus folgt $\epsilon(n) \leq \mathrm{negl}(n)$ wie gewünscht.

Beweis Pseudozufallsgenerator (3/3)

Lemma

$$\epsilon(n) = \frac{1}{2} \cdot \left(Ws[\mathcal{A}(G(x)) = 1] - Ws[\mathcal{A}(f(x)||\overline{hc(x)}) = 1] \right)$$

Beweis:

- Es gilt $\epsilon(n) = \mathrm{Ws}_{x \in_R \{0,1\}^n}[\mathcal{A}(G(x) = 1)] - \mathrm{Ws}_{r \in_R \{0,1\}^{n+1}}[\mathcal{A}(r) = 1]$.
- Wir schreiben

$$\begin{aligned}
& \mathrm{Ws}_{r \in_R \{0,1\}^{n+1}}[\mathcal{A}(r) = 1] \\
= \; & \mathrm{Ws}_{r \in_R \{0,1\}^{n}, r' \in \{0,1\}}[\mathcal{A}(r||r') = 1] \\
= \; & \mathrm{Ws}_{x \in_R \{0,1\}^{n}, r' \in \{0,1\}}[\mathcal{A}(f(x)||r') = 1] && \text{// } f \text{ ist Permutation.} \\
= \; & \frac{1}{2} \left(\mathrm{Ws}_{x \in_R \{0,1\}^n}[\mathcal{A}(f(x)||hc(x)) = 1] + \mathrm{Ws}_{x \in_R \{0,1\}^n}[\mathcal{A}(f(x)||\overline{hc(x)}) = 1] \right) \\
= \; & \frac{1}{2} \cdot \mathrm{Ws}_{x \in_R \{0,1\}}[\mathcal{A}(G(x)) = 1] + \frac{1}{2} \cdot \mathrm{Ws}_{x \in_R \{0,1\}^n}[\mathcal{A}(f(x)||\overline{hc(x)}) = 1].
\end{aligned}$$

- Daraus folgt die Behauptung des Lemmas.

1 Bit $\Rightarrow$ viele Bits

Satz

Sei $G : \{0,1\}^n \to \{0,1\}^{n+1}$ ein Pseudozufallsgenerator mit 1 Bit Expansion. Dann existiert ein Pseudozufallsgenerator $G' : \{0,1\}^n \to \{0,1\}^{n+poly(n)}$ mit polynomieller Expansion.

Beweisidee: Konstruktion von G'.

- Berechne $x_1 = G(s) \in \{0,1\}^{n+1}$.
- Setze $x_1 = s_1 y_1$ mit neuer Saat $s_1 \in \{0,1\}^n$ und Bit $y_1 \in \{0,1\}$.
- Berechne $x_2 = G(s_1) \in \{0,1\}^{n+1}$.
- Setze $x_2 = s_2 y_2$ mit neuer Saat $s_2 \in \{0,1\}^n$ und Bit $y_2 \in \{0,1\}$.
- Iteriere, Ausgabe nach $m = poly(n)$ Iterationen ist

$$x_m = G(s_{m-1}) y_m \ldots y_1 \in \{0,1\}^{n+m}.$$

Pseudozufallsgenerator $\Rightarrow$ Pseudozufallsfunktion

Satz

Sei $G : \{0,1\}^n \to \{0,1\}^{2n}$ ein Pseudozufallsgenerator. Dann existiert eine Pseudozufallsfunktion $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$.

Beweisidee:

- Wir schreiben $G(s) = G_0(s)||G_1(s)$ mit $G_i(s) \in \{0,1\}^n$.
- Definieren 1-Bit Funktion $F : \{0,1\}^n \times \{0,1\} \to \{0,1\}^n$ mittels
$$F_k(0) = G_0(k) \text{ und } F_k(1) = G_1(k).$$
- Definieren 2-Bit Funktion $F : \{0,1\}^n \times \{0,1\}^2 \to \{0,1\}^n$ mittels
$$F_k(00) = G_0(G_0(k)), F_k(01) = G_1(G_0(k)),$$
$$F_k(10) = G_0(G_1(k)), F_k(11) = G_1(G_1(k)).$$
- Definieren n-Bit Funktion $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ mittels
$$F_k(x) = G_{x_n}(G_{x_{n-1}} \ldots (G_{x_1}(k)) \ldots).$$