Existenz von Einwegfunktionen

Satz Einweg-Eigenschaft von f_{FO}

Unter der Faktorisierungsannahme ist f_{FO} eine Einwegfunktion.

Beweis:

- f_{FO} ist mittels FACTOR-ONEWAY effizient berechenbar.
- z.z.: Invertierer $\mathcal{A}$ von f_{FO} impliziert Faktorisierer $\mathcal{A}'$.
- Sei $\mathcal{A}$ ein Invertierer für f_{FO} mit Erfolgsws $\mathrm{Ws}[\mathit{Invert}_{\mathcal{A},f_{FO}}(N) = 1]$.
- Sei $x' \leftarrow \mathcal{A}(N)$ mit $f(x') = N$.
- Berechne die Faktorisierung $(N, p, q) \leftarrow \mathit{GenModulus}(1^n, x')$.
- Unter der Faktorisierungsannahme gilt

$$\mathrm{negl} \geq \mathrm{Ws}[\mathit{Factor}_{A',\mathit{GenModulus}}(n) = 1] = \mathrm{Ws}[\mathit{Invert}_{\mathcal{A},f_{FO}}(n) = 1].$$

Trapdoor-Permutationsfamilie

Definition Permutationsfamilie

Eine *Permutationsfamilie* $\Pi_f = (Gen, Samp, f)$ besteht aus 3 ppt Alg:

1. $I \leftarrow Gen(1^n)$, wobei I eine Urbildmenge D für f definiert.
2. $x \leftarrow Samp(I)$, wobei $x \in_R D$.
3. $y \leftarrow f(I, x)$ mit $y := f(x) \in D$ und $f : D \leftarrow D$ ist bijektiv.

Definition Trapdoor-Permutationsfamilie

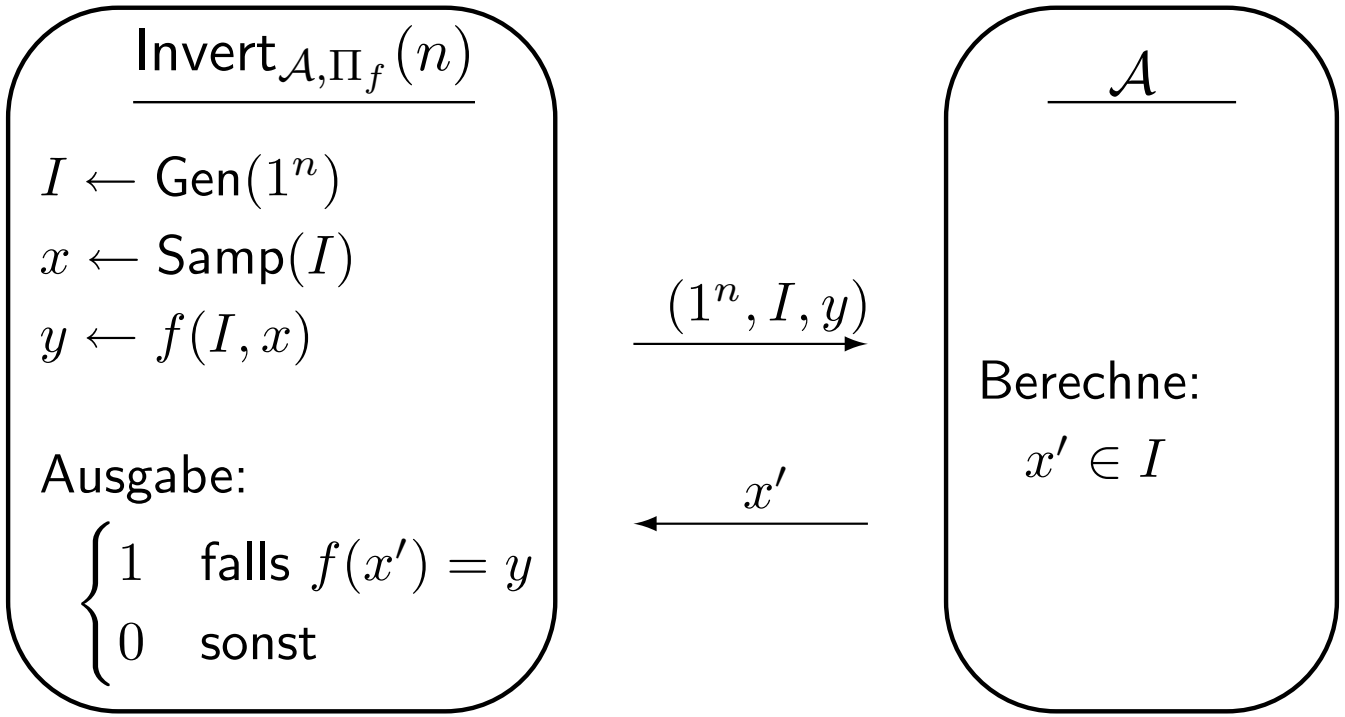
Trapdoor-Permutationsfamilie $\Pi_f = (Gen, Samp, f, Inv)$ besteht aus

1. $(I, td) \leftarrow Gen(1^n)$ mit td als Trapdoor-Information
2. $x \leftarrow Samp(I)$ wie zuvor
3. $y \leftarrow f(I, x)$ wie zuvor
4. $x \leftarrow Inv(td, y)$ mit $Inv_{td}(f(x)) = x$ für alle $x \in D$.

Spiel Invertieren einer Permutation $Invert_{\mathcal{A},\Pi_f}(n)$

Sei $\mathcal{A}$ ein Invertierer für die Familie Π_f.

1. $I \leftarrow Gen(1^n)$, $x \leftarrow Samp(I)$ und $y \leftarrow f(I,x)$.
2. $x' \leftarrow \mathcal{A}(I,y)$.
3. $$Invert_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } f(x') = y \\ 0 & \text{sonst} \end{cases}.$$



Konstruktion einer Trapdoor-Einwegpermutation

Definition Einweg-Permutation

Eine (Trapdoor-)Permutationsfamilie heißt *(Td-)Einwegpermutation* falls für alle ppt Algorithmen $\mathcal{A}$ gilt $\mathrm{Ws}[\mathit{Invert}_{\mathcal{A},\Pi_f}(n) = 1] \leq \mathrm{negl}(n)$.

Bsp: Trapdoor-Einwegpermutation unter RSA-Annahme

- ***Gen*(1^n)**:
  $(N, e, d) \leftarrow \mathit{GenRSA}(1^n)$, Ausgabe $I = (N, e)$ und $td = (N, d)$.
- ***Samp*(I)**:
  Wähle $x \in_R \mathbb{Z}_N$.
- ***f*(I, x)**:
  Berechne $y \leftarrow x^e \bmod N$.
- ***Inv*(td, y)**:
  Berechne $x \leftarrow y^d \bmod N$.

Hardcore-Prädikat

Ziel: Destilliere Komplexität des Invertierens auf ein Bit.

Definition Hardcore-Prädikat

Sei Π_f eine Einwegpermutation. Sei *hc* ein deterministischer pt Alg mit Ausgabe eines Bits $hc(x)$ bei Eingabe $x \in D$. *hc* heißt *Hardcore-Prädikat* für *f* falls für alle ppt Algorithmen $\mathcal{A}$ gilt:

$$\mathrm{Ws}[\mathcal{A}(f(x)) = hc(x)]] \leq \tfrac{1}{2} + \mathrm{negl}(n).$$

Intuition: Bild $f(x)$ hilft nicht beim Berechnen von $hc(x)$.

Bsp: Goldreich-Levin Hardcore-Prädikat (ohne Beweis)

- Sei *f* eine Einwegpermutation mit Definitionsbereich $\{0,1\}^n$.
- Sei $x = x_1 \dots x_n \in \{0,1\}^n$. Konstruiere
$$g(x,r) := (f(x), r) \text{ mit } r \in_R \{0,1\}^n.$$
- Offenbar ist *g* ebenfalls eine Einwegpermutation.
- Wir konstruieren ein Hardcore-Prädikat *hc* für *g* mittels
$$hc(x,r) = \langle x, r\rangle = \textstyle\sum_{i=1}^{n} x_i r_i \bmod 2.$$
- Beweis der Hardcore-Eigenschaft ist nicht-trivial.

Verschlüsselung aus Trapdoor-Einwegpermutation

Algorithmus VERSCHLÜSSELUNG$_{\Pi_f}$

Sei Π_f eine Td-Einwegpermutation mit Hardcore-Prädikat hc.

1. **Gen:** $(I, td) \leftarrow Gen(1^n)$. Ausgabe $pk = I$ und $sk = td$.
2. **Enc:** Für $m \in \{0,1\}$ wähle $x \in_R D$ und berechne
$$c \leftarrow (f(x), hc(x) \oplus m).$$
3. **Dec:** Für Chiffretext $c = (c_1, c_2)$ berechne $x \leftarrow Inv_{td}(c_1)$ und
$$m \leftarrow c_2 \oplus hc(x).$$

Intuition:

- $hc(x)$ ist "pseudozufällig" gegeben $f(x)$.
- D.h. $hc(x) \oplus m$ ist ununterscheidbar von 1-Bit One-Time Pad.

CPA-Sicherheit unserer Konstruktion

Satz CPA-Sicherheit von VERSCHLÜSSELUNG$_\Pi$

Sei Π_f eine Trapdoor-Einwegpermutation mit Hardcore-Prädikat *hc*. Dann ist VERSCHLÜSSELUNG$_\Pi$ CPA-sicher.

Beweis:

- Sei $\mathcal{A}$ ein Angreifer mit Erfolgsws $\epsilon(n) = \mathrm{Ws}[PubK^{cpa}_{\mathcal{A},\Pi_f}(n) = 1]$.
- OBdA $(m_0, m_1) \leftarrow \mathcal{A}(pk)$ mit $\{m_0, m_1\} = \{0,1\}$. (Warum?)
- Verwenden $\mathcal{A}$ um einen Angreifer $\mathcal{A}_{hc}$ für *hc* zu konstruieren.

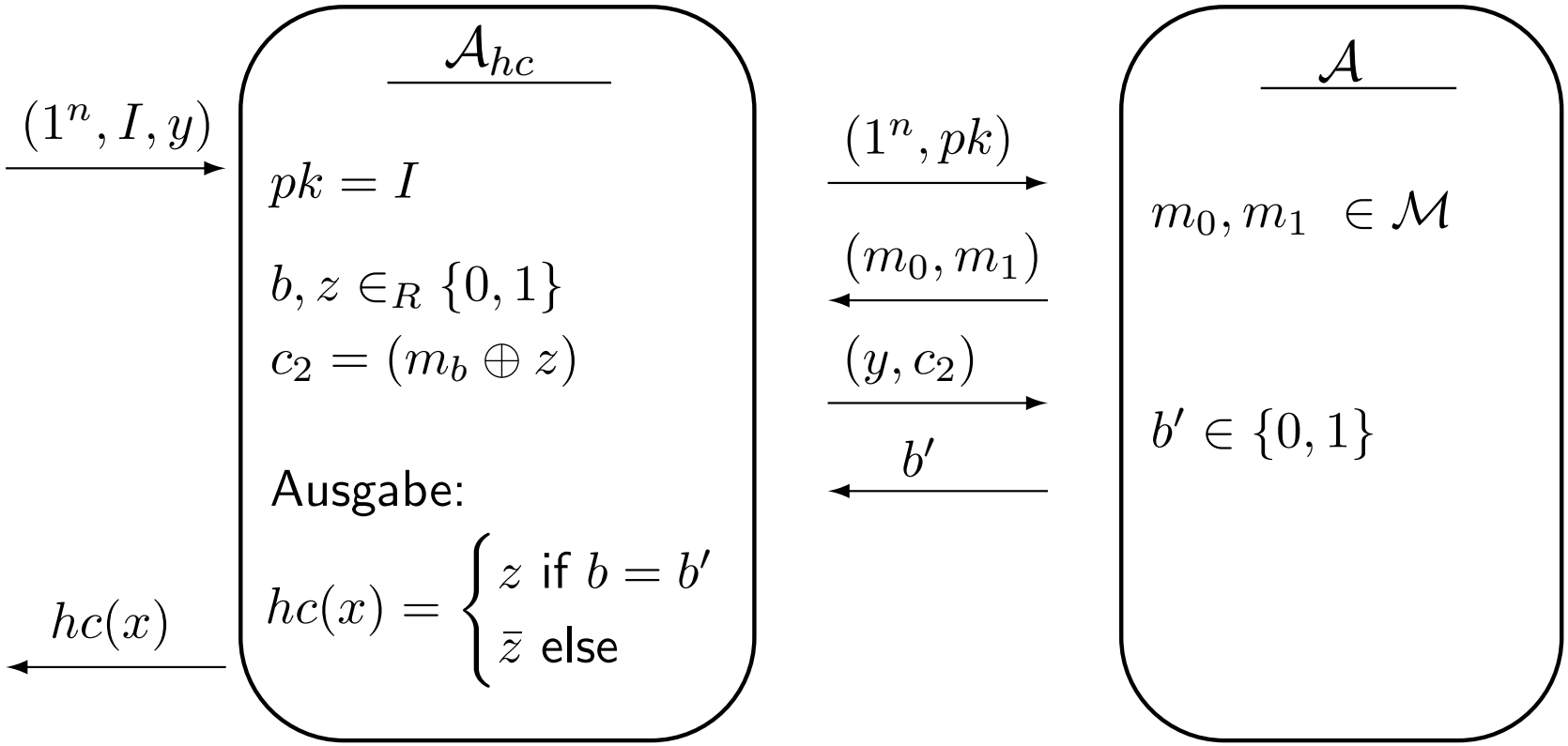
Algorithmus Angreifer $\mathcal{A}_{hc}$

Eingabe: $I, y = f(x) \in D$

1. Setze $pk \leftarrow I$ und berechne $(m_0, m_1) \leftarrow \mathcal{A}(pk)$.
2. Wähle $b, z \in_R \{0,1\}$. Setze $c_2 \leftarrow m_b \oplus z$.
3. $b' \leftarrow \mathcal{A}(y, c_2)$

Ausgabe: $hc(x) = \begin{cases} z & \text{falls } b = b' \\ \bar{z} & \text{sonst} \end{cases}$.

Angreifer $\mathcal{A}_{hc}$



CPA-Sicherheit von VERSCHLÜSSELUNG$_\Pi$

Beweis: Fortsetzung

- Sei $x = f^{-1}(y)$. $\mathcal{A}_{hc}$ rät $z = hc(x)$.
- Es gilt $\mathrm{Ws}[\mathcal{A}_{hc}(f(x)) = hc(x)] =$
$$\frac{1}{2} \cdot \mathrm{Ws}[b = b' \mid z = hc(x)] + \frac{1}{2} \cdot \mathrm{Ws}[b \neq b' \mid z \neq hc(x)].$$
- **1. Fall** $z = hc(x)$: (y, c_2) ist korrekte Verschlüsselung von m_b, d.h.
$$\mathrm{Ws}[b = b' \mid z = hc(x)] = \epsilon(n).$$
- **2. Fall** $z \neq hc(x)$: (y, c_2) ist Verschlüsselung von $\bar{m}_b = m_{\bar{b}}$, d.h.
$$\mathrm{Ws}[b \neq b' \mid z \neq hc(x)] = \epsilon(n).$$
- Da hc ein Hardcore-Prädikat ist, folgt
$$\frac{1}{2} + \mathrm{negl}(n) \geq \mathrm{Ws}[A_{hc}(f(x)) = hc(x)] = \epsilon(n).$$