Expansion: 1 Bit $\Rightarrow$ viele Bits

Satz

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

$$G' : \{0,1\}^n \to \{0,1\}^{n+poly(n)}.$$

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 PRNG. Dann existiert eine PRF

$$F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n.$$

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
$$F_k(x) = G_{x_n}(G_{x_{n-1}} \dots (G_{x_1}(k)) \dots).$$

Existenz und Verschlüsselung mit Pseudozufallsfkt

Fakt Existenz von Pseudozufallsfunktionen

PRFs existieren gdw PRNGs existieren.

Beweis: $\Leftarrow$: siehe beide Sätze zuvor, $\Rightarrow$: siehe Übung.

Algorithmus Verschlüsselung Π_B

Sei F eine PRF auf n Bits. Wir definieren $\Pi_B = (Gen, Enc, Dec)$ für Nachrichten der Länge n.

1. **Gen:** Wähle $k \in_R \{0,1\}^n$.
2. **Enc:** Für $m \in \{0,1\}^n$ wähle $r \in_R \{0,1\}^n$ und berechne
$$c := (r, F_k(r) \oplus m).$$
3. **Dec:** Für $c = (c_1, c_2) \in \{0,1\}^n \times \{0,1\}^n$ berechne
$$m := F_k(c_1) \oplus c_2.$$

Sicherheit von Π_B

Satz Sicherheit von Π_B

Sei F eine PRF. Dann ist Π_B CPA-sicher.

Intuition:

- $F_k(r)$ ist nicht unterscheidbar von n-Bit Zufallsstring.
- D.h. in der zweiten Komponente ist die Verteilung ununterscheidbar von einem One-Time Pad.
- Vorsicht: Benötigen, dass r nicht wiederverwendet wird.

Beweis:

- Sei $\mathcal{A}$ ein CPA-Angreifer mit Vorteil $\epsilon(n)$.
- Konstruieren mittels $\mathcal{A}$ einen Unterscheider D für $F_k(\cdot)$ und $f(\cdot)$.

Unterscheider D

Algorithmus Unterscheider *D*

EINGABE: 1^n, $\mathcal{O} : \{0,1\}^n \leftarrow \{0,1\}^n$ (mit $\mathcal{O} = F_k(\cdot)$ oder $\mathcal{O} = f(\cdot)$)

1. Beantworte Verschlüsselungsanfragen $Enc_k(m_i')$ von $\mathcal{A}$ wie folgt: Wähle $r_i \in_R \{0,1\}^n$ und sende $(r_i, \mathcal{O}(r_i) \oplus m)$ an $\mathcal{A}$.
2. Beantworte Challenge (m_0, m_1) von $\mathcal{A}$ wie folgt: Wähle $r \in_R \{0,1\}^n$, $b \in_R \{0,1\}$ und sende $(r, \mathcal{O}(r) \oplus m_b)$ an $\mathcal{A}$.
3. Erhalte nach weiteren Verschlüsselsanfragen von $\mathcal{A}$ Bit b'.

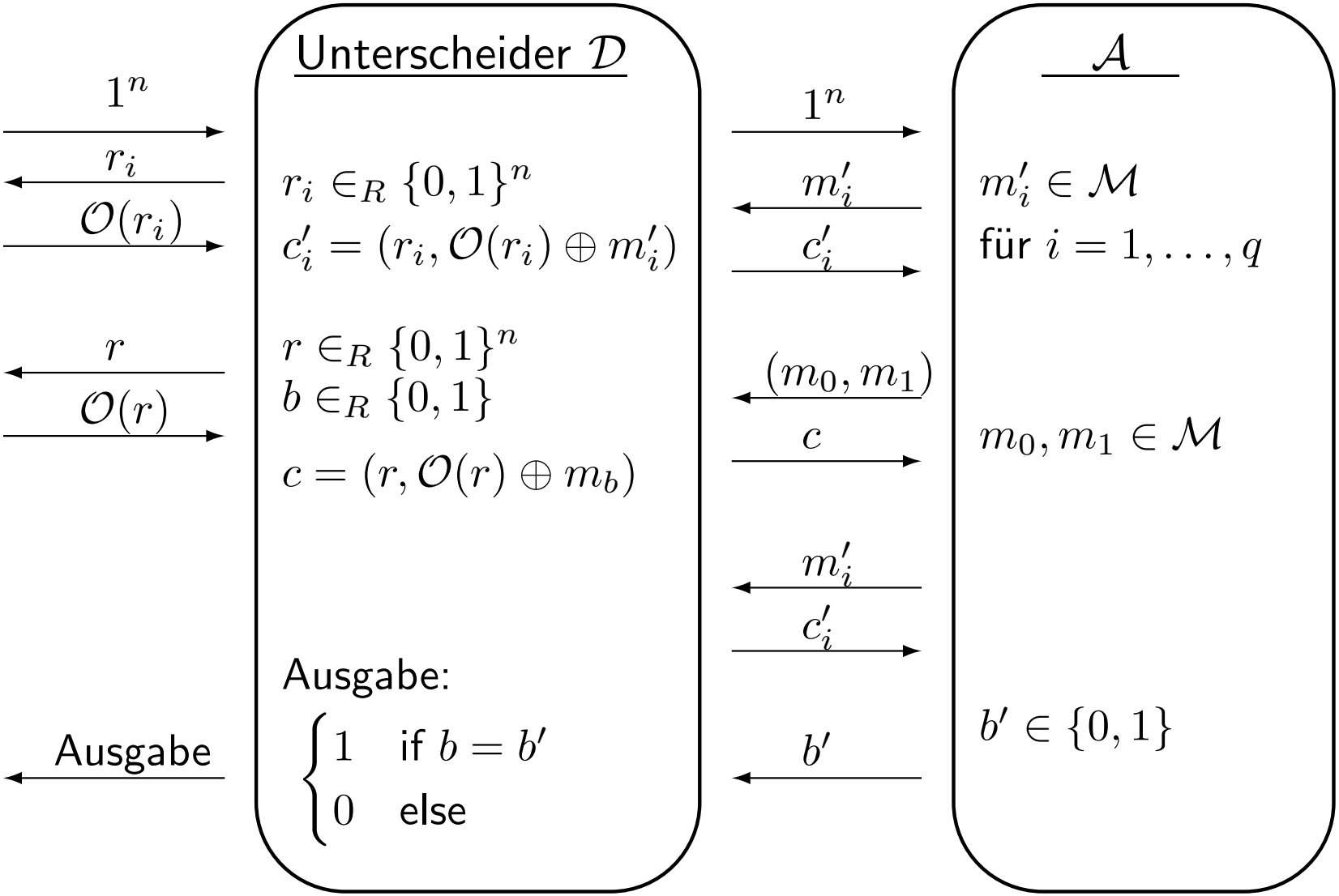
$$\text{AUSGABE:} = \begin{cases} 1 & \text{falls } b' = b, \text{ Interpretation: } \mathcal{O} = F_k(\cdot) \\ 0 & \text{sonst,} \qquad\quad \text{Interpretation: } \mathcal{O} = f(\cdot) \end{cases}.$$

Fall 1: $\mathcal{O} = F_k(\cdot)$, d.h. wir verwenden eine Pseudozufallsfunktion.

- Dann ist die Verteilung von $\mathcal{A}$ identisch zu Π_B. Damit gilt

$$\text{Ws}[D^{F_k(\cdot)}(1^n) = 1] = \text{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi_B}(n) = 1] = \tfrac{1}{2} + \epsilon(n).$$

Unterscheider D



Verwenden einer echten Zufallsfunktion

Fall 2: $\mathcal{O} = f(\cdot)$, d.h. wir verwenden eine echte Zufallsfunktion.

- Sei Π' das Protokoll Π_B unter Verwendung von $f(\cdot)$ statt $F_k(\cdot)$.
- Sei *Repeat* das Ereignis, dass r in einer der Verschlüsselungsanfragen verwendet wurde.
- Für alle Angreifer $\mathcal{A}$ gilt $\mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1]$

$$\begin{aligned} &= \mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1 \cap Repeat] + \mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1 \cap \overline{Repeat}] \\ &\leq \mathrm{Ws}[Repeat] + \mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1 \mid \overline{Repeat}] \end{aligned}$$

- Ein ppt Angreifer $\mathcal{A}$ stelle insgesamt polynomiell viele Anfragen.
- Sei $q(n)$ die Anzahl der Anfragen. Dann gilt

$$\begin{aligned} \mathrm{Ws}[Repeat] &= \mathrm{Ws}[r = r_1 \cup \ldots \cup r = r_q] \\ &\leq \mathrm{Ws}[r = r_1] + \ldots + \mathrm{Ws}[r = r_q] = \frac{q}{2^n} = \mathrm{negl}(n). \end{aligned}$$

Fall 2: (Fortsetzung)

- Aufgrund der perfekten Sicherheit des One-Time Pads gilt
$$\mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1 \mid \overline{Repeat}] = \tfrac{1}{2}.$$
- Es folgt $\mathrm{Ws}[D^{f(\cdot)}(1^n) = 1] = \mathrm{Ws}[PrivK^{cpa}_{\mathcal{A},\Pi'}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$.

- Aus der Pseudozufälligkeit von F folgt insgesamt
$$\mathrm{negl}(n) \geq \Big| \underbrace{\mathrm{Ws}[D^{F_k(\cdot)}(1^n) = 1]}_{\frac{1}{2}+\epsilon(n)} - \underbrace{\mathrm{Ws}[D^{f(\cdot)}(1^n) = 1]}_{\leq \frac{1}{2}+\mathrm{negl}(n)} \Big|.$$
- Es folgt $\epsilon \leq \mathrm{negl}(n)$ für alle polynomiellen Angreifer $\mathcal{A}$.

Nachrichten beliebiger Länge

Algorithmus Verschlüsselung Π'_B

Sei F eine PRF auf n Bits. Wir definieren $\Pi'_B = (Gen, Enc, Dec)$ für Nachrichten $m \in \{0,1\}^*$.

1. **Gen:** Wähle $k \in_R \{0,1\}^n$.
2. **Enc:** Für $m = m_1 \dots m_\ell$ mit $m_i \in \{0,1\}^n$ wähle $r_1, \dots r_\ell$ mit $r_i \in_R \{0,1\}^n$ und berechne
$$c := (r_1, \dots, r_\ell, F_k(r_1) \oplus m_1, \dots, F_k(r_\ell) \oplus m_\ell).$$
3. **Dec:** Für $c = (c_1, ..., c_{2\ell}) \in (\{0,1\}^n)^{2\ell}$ berechne
$$m := F_k(c_1) \oplus c_{\ell+1} \dots F_k(c_\ell) \oplus F_k(c_{2\ell}).$$

CPA-Sicherheit von Π'_B

Satz CPA-Sicherheit von Π'_B

Sei F eine PRF. Dann ist Π'_B CPA-sicher.

Beweis:

- Aus der CPA-Sicherheit von Π_B folgt die mult-CPA Sicherheit von Π_B und damit die CPA-Sicherheit von Π'_B.

Nachteil: Chiffetexte sind doppelt so lang wie Klartexte (Nachrichtenexpansion 2).

Pseudozufallspermutationen

Definition schlüsselabhängige Permutation

Seien F, F^{-1} pt Algorithmen. F heißt *schlüsselabhängige Permutation* auf n Bits falls

1. F berechnet eine Funktion $\{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$, so dass für alle $k \in \{0,1\}^m$ die Funktion $F_k(\cdot)$ eine Bijektion ist.
2. $F_k^{-1}(\cdot)$ berechnet die Umkehrfunktion von $F_k(\cdot)$.

Definition Pseudozufallspermutation

Sei F eine schlüsselabhängige Permutation auf n Bits. Wir bezeichnen F als *Pseudozufallspermutation* (PRP), falls für alle ppt D gilt

$$\left|\mathrm{Ws}[D^{F_k(\cdot)}(1^n) = 1] - \mathrm{Ws}[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathrm{negl}(n),$$

mit $k \in_R \{0,1\}^m$ und $f \in_R Perm_n$, wobei $Perm_n$ die Menge aller Permutationen auf n Bits ist.

Starke Pseudozufallspermutationen

Satz Pseudozufallspermutationen und Pseudozufallsfunktionen

Jede PRP ist eine PRF.

Beweis: Übung.

Definition Starke Pseudozufallspermutation (Blockchiffre)

Sei F eine schlüsselabhängige Permutation auf n Bits. Wir bezeichnen F als *starke Pseudozufallspermutation (Blockchiffre)*, falls für alle ppt D gilt

$$\left|\mathrm{Ws}[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n)=1]-\mathrm{Ws}[D^{f(\cdot),f^{-1}(\cdot)}(1^n)=1]\right|\leq \mathrm{negl}(n),$$

mit $k \in_R \{0,1\}^n$ und $f \in_R Perm_n$.

Blockchiffren als kryptographische Primitive

Anmerkungen: Blockchiffren

- Praktische Realisierungen von starken Pseudozufallspermutationen bezeichnet man als *Blockchiffren*.
- Wir haben gesehen, dass Blockchiffren $F_k(\cdot)$ zur Konstruktion CPA-sicherer Verschlüsselung verwendet werden können.
- Vorsicht: Blockchiffren selbst sind keine sicheren Verschlüsselungsverfahren.
- $c := F_k(m)$ ist eine deterministische, unsichere Verschlüsselung.
- D.h. wir benötigen einen Randomisierungsprozess bei Enc.

Bsp: DES (Data Encryption Standard, 1976)

- $F : \{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$
- Problem des zu kleinen Schlüsselraums
- bester bekannter KPA Angriff mit 2^{43} Klartexten

AES (Advanced Encryption Standard, 2002)

- $F : \{0,1\}^{k} \times \{0,1\}^{128} \to \{0,1\}^{128}$ mit $k \in \{128, 192, 256\}$
- Derzeit kein erfolgreicher Angriff bekannt.