

## Definition Ununterscheidbare Chiffretexte

Ein Verschlüsselungsschema  $\Pi = (Gen, Enc, Dec)$  besitzt *ununterscheidbare Chiffretexte gegenüber KPA* falls für alle ppt  $\mathcal{A}$  gilt

$$Ws[PrivK_{\mathcal{A}, \Pi}^{eav}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Der Wsraum ist definiert über die Münzwürfe von  $PrivK_{\mathcal{A}, \Pi}^{eav}$ . (Genauer: die Münzwürfe von  $Gen$ ,  $b$ ,  $Enc$  und  $\mathcal{A}$ .)

Die Differenz  $Ws[PrivK_{\mathcal{A}, \Pi}^{eav}(n) = 1] - \frac{1}{2}$  bezeichnen wir als Vorteil von  $\mathcal{A}$ .  $\Pi$  heißt *KPA-sicher*, falls der Vorteil vernachlässigbar ist.

# Chiffretext liefert kein einzelnes Bit des Klartextes

**Notation:** Sei  $m^i$  das  $i$ -te Bit einer Nachricht  $m \in \{0, 1\}^n$ .

## Satz

Sei  $\Pi$  KPA-sicher. Dann gilt für alle ppt  $\mathcal{A}$  und alle  $i \in [n]$ :  
 $\text{Ws}[\mathcal{A}(\text{Enc}_k(m)) = m^i] \leq \frac{1}{2} + \text{negl}(n)$ .

## Beweis:

- Sei  $I_0^n = \{m \in \{0, 1\}^n \mid m^i = 0\}$  und  $I_1^n = \{m \in \{0, 1\}^n \mid m^i = 1\}$ .
- Sei  $\mathcal{A}$  ein Unterscheider für das  $i$ -te Bit mit Vorteil  $\epsilon(n)$ .
- Konstruieren Angreifer  $\mathcal{A}'$ , der  $m_0 \in I_0^n$  und  $m_1 \in I_1^n$  unterscheidet.

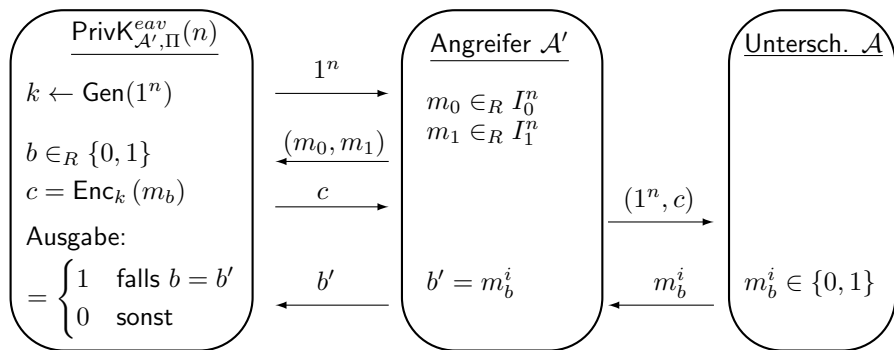
## Algorithmus KPA-Angreifer $\mathcal{A}'$

EINGABE:  $1^n$

- 1 Wähle  $m_0 \in_R I_0^n$ ,  $m_1 \in_R I_1^n$ .
- 2 Empfange  $\text{Enc}_k(m_b)$  aus  $\text{PrivK}_{\mathcal{A}', \Pi}(n)$ -Spiel mit  $b \in_R \{0, 1\}$ .
- 3  $b' \leftarrow \mathcal{A}(\text{Enc}_k(m_b))$

AUSGABE:  $b' \in \{0, 1\}$

# Algorithmus KPA-Angreifer $A'$



# Chiffretext liefert kein einzelnes Bit des Klartextes

**Beweis:** Es gilt  $\text{Ws}[\text{PrivK}_{\mathcal{A}', \Pi}^{\text{eav}}(n) = 1] = \text{Ws}[\mathcal{A}'(\text{Enc}_k(m_b) = b)]$

$$\begin{aligned} &= \sum_{j=0}^1 \text{Ws}[b = j] \cdot \text{Ws}[\mathcal{A}'(\text{Enc}_k(m_b)) = b \mid b = j] \\ &= \sum_{j=0}^1 \text{Ws}[b = j] \cdot \text{Ws}[\mathcal{A}(\text{Enc}_k(m_b)) = m_b^j \mid b = j] \\ &= \frac{1}{2} \cdot \text{Ws}[\mathcal{A}(\text{Enc}_k(m_0)) = 0] + \frac{1}{2} \cdot \text{Ws}[\mathcal{A}(\text{Enc}_k(m_1)) = 1] \\ &= \text{Ws}[\mathcal{A}(\text{Enc}_k(m)) = m^j] = \frac{1}{2} + \epsilon(n) \end{aligned}$$

- Da  $\Pi$  KPA-sicher ist, gilt  $\text{Ws}[\text{PrivK}_{\mathcal{A}', \Pi}^{\text{eav}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n)$ .
- Daraus folgt, dass  $\mathcal{A}$  Vorteil  $\epsilon \leq \text{negl}(n)$  besitzt.

# Chiffretext liefert ppt $\mathcal{A}$ keine Information

**Ziel:**  $\mathcal{A}$  kann aus  $Enc_k(m)$  keine nicht-triviale Funktion  $f(m)$  berechnen.

- Sei  $\mathcal{M} \subseteq \{0, 1\}^*$  und  $S_n = \mathcal{M} \cap \{0, 1\}^n$ .
- Wir wählen ein  $m \in_R S_n$ .

## Satz Nicht-Berechenbarkeit von Funktionen

Sei  $\Pi$  KPA-sicher. Für jeden ppt Angreifer  $\mathcal{A}$  existiert ein ppt Algorithmus  $\mathcal{A}'$ , so dass für alle ppt-berechenbaren Funktionen  $f$

$$|\text{Ws}[\mathcal{A}(1^n, Enc_k(m)) = f(m)] - \text{Ws}[\mathcal{A}'(1^n) = f(m)]| \leq \text{negl}(n).$$

Wsraum: Zufällige Wahl von  $m, k$ , Münzwürfe von  $\mathcal{A}, \mathcal{A}', Enc$ .

### Beweis:

- Wir zeigen zunächst, dass für alle ppt  $\mathcal{A}$  gilt
$$|\text{Ws}[\mathcal{A}(1^n, Enc_k(m)) = f(m)] - \text{Ws}[\mathcal{A}(1^n, Enc_k(1^n)) = f(m)]| \leq \text{negl}(n).$$
- Wir konstruieren dazu KPA-Angreifer  $D$  auf  $\Pi$  mittels  $\mathcal{A}$ .

$\mathcal{A}$  kann  $Enc_k(m)$  und  $Enc_k(1^n)$  nicht unterscheiden.

### Algorithmus Angreifer $D$ im Spiel $PrivK_{D,\Pi}^{eav}(n)$

EINGABE:  $1^n$ .

- 1 Wähle  $m_0 = m \in_R \mathcal{S}_n$ ,  $m_1 = 1^n$ . Erhalte  $Enc_k(m_b)$  für  $b \in_R \{0, 1\}$ .
- 2 Sende  $(1^n, Enc_k(m_b))$  an  $\mathcal{A}$ . Erhalte Ausgabe  $f(m_b)$ .

AUSGABE:  $b' = \begin{cases} 0 & \text{falls } f(m) = f(m_b) \\ 1 & \text{sonst} \end{cases}$ .

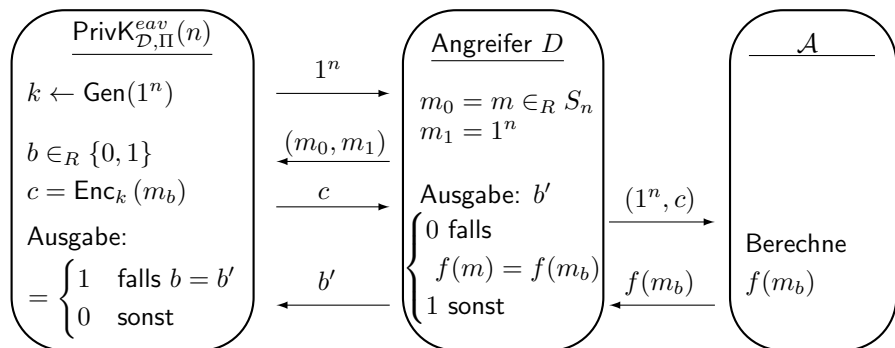
**Fall 1:**  $b = 0$ , d.h.  $\mathcal{A}$  erhält  $Enc_k(m)$ .

- Es gilt  $\text{Ws}[PrivK_{D,\Pi} = 1 \mid b = 0] = \text{Ws}[\mathcal{A}(1^n, Enc_k(m)) = f(m)]$ .

**Fall 2:**  $b = 1$ , d.h.  $\mathcal{A}$  erhält  $Enc_k(1^n)$ .

- Es gilt  $\text{Ws}[PrivK_{D,\Pi} = 1 \mid b = 1] = \text{Ws}[\mathcal{A}(1^n, Enc_k(1^n)) \neq f(m)]$   
 $= 1 - \text{Ws}[\mathcal{A}(1^n, Enc_k(1^n)) = f(m)]$ .

# Angreifer $\mathcal{D}$



$\mathcal{A}$  kann  $Enc_k(m)$  und  $Enc_k(1^n)$  nicht unterscheiden.

- Insgesamt folgt damit aus der KPA-Sicherheit von  $\Pi$

$$\begin{aligned} \text{negl}(n) &\geq \left| \frac{1}{2} - \text{Ws}[\text{PrivK}_{D,\Pi} = 1] \right| \\ &= \left| \frac{1}{2} - \sum_{i \in \{0,1\}} \text{Ws}[\text{PrivK}_{D,\Pi} = 1 \mid b = i] \cdot \text{Ws}[b = i] \right| \\ &= \left| \frac{1}{2} - \frac{1}{2} (\text{Ws}[\mathcal{A}(1^n, Enc_k(m)) = f(m)] \right. \\ &\quad \left. + 1 - \text{Ws}[\mathcal{A}(1^n, Enc_k(1^n)) = f(m)]) \right|. \end{aligned}$$

- Daraus folgt wie gewünscht

$$|\text{Ws}[\mathcal{A}(1^n, Enc_k(m)) = f(m)] - \text{Ws}[\mathcal{A}(1^n, Enc_k(1^n)) = f(m)]| \leq \underbrace{2\text{negl}(n)}_{\text{negl}(n)}.$$



# Konstruktion von $\mathcal{A}'$

## Algorithmus $\mathcal{A}'$

EINGABE:  $1^n$

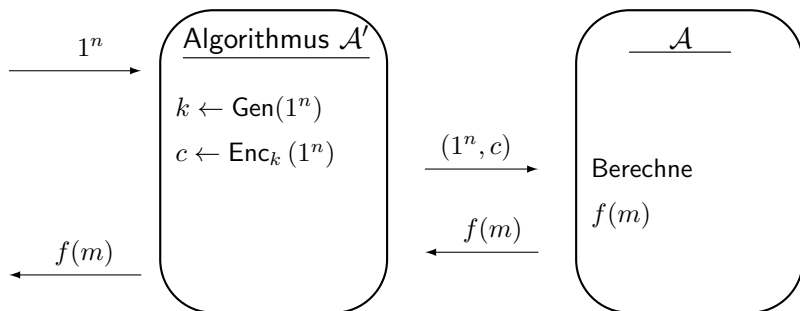
- 1 Berechne  $k \leftarrow \text{Gen}(1^n)$  und  $c \leftarrow \text{Enc}_k(1^n)$ .
- 2  $f(m) \leftarrow \mathcal{A}(1^n, \text{Enc}_k(1^n))$ .

AUSGABE:  $f(m)$

Unser Satz zur Nicht-Berechenbarkeit von Funktionen folgt aus

$$\text{Ws}[\mathcal{A}'(1^n) = f(m)] = \text{Ws}[\mathcal{A}(1^n, \text{Enc}_k(1^n)) = f(m)].$$

# Semantische Sicherheit



**Semantische Sicherheit (informal):** Erweiterung auf:

- Beliebige Verteilung anstatt Gleichverteilung  $m \in_R S_n$ .
- $\mathcal{A}$  und  $\mathcal{A}'$  erhalten zusätzliche Information über den Klartext.

Man kann zeigen:

Semantische Sicherheit ist äquivalent zu KPA-Sicherheit.

# Pseudozufälligkeit

## Motivation: Pseudozufallsgenerator

- One-Time Pad: Sicherheit von  $m \oplus k$  für  $m \in \{0, 1\}^n$ ,  $k \in_R \{0, 1\}^n$ .
- D.h. wir benötigen einen echten Zufallsstring  $k \in \{0, 1\}^n$ .
- Sei  $G$  ein Algorithmus, der eine Verteilung  $\mathcal{D}$  auf  $\{0, 1\}^n$  liefert.
- Falls es für ppt  $D$  unmöglich ist,  $\mathcal{D}$  von der Gleichverteilung auf  $\{0, 1\}^n$  zu unterscheiden, so können wir  $k$  mittels  $G$  wählen.

## Definition Pseudozufallsgenerator

Sei  $G$  ein ppt Algorithmus, der eine Funktion  $\{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  mit  $\ell(n) > n$  berechne.  $G$  heißt *Pseudozufallsgenerator* falls für alle ppt  $D$

$$|\text{Ws}[D(r) = 1] - \text{Ws}[D(G(s)) = 1]| \leq \text{negl}(n),$$

wobei  $r \in_R \{0, 1\}^{\ell(n)}$  und  $s \in_R \{0, 1\}^n$ , die sogenannte *Saat*.

Wsraum: Zufällige Wahl von  $r, s$ , Münzwürfe von  $D$ .

**Anmerkung:**  $G$  expandiert die echt zufällige Saat  $s \in \{0, 1\}^n$  in ein pseudozufälliges  $G(s) \in \{0, 1\}^{\ell(n)}$  mit Expansionsfaktor  $\ell(n)$ .

# Unterscheider $D$ mit beliebiger Laufzeit

## Satz Unterscheider $D$ mit beliebiger Laufzeit

Sei  $G : \{0, 1\}^n \rightarrow \{0, 1\}^{\ell(n)}$  ein Pseudozufallsgenerator. Dann existiert ein Unterscheider  $D$  mit Laufzeit  $\mathcal{O}(2^n \cdot \text{Laufzeit}(G))$  und Erfolgsws

$$\text{Ws}[D(r) = 1] - \text{Ws}[D(G(s)) = 1] \geq \frac{1}{2}.$$

### Beweis:

- $D$  prüft, ob  $w$  mittels  $G$  generiert werden kann.

## Algorithmus Unterscheider $D$

EINGABE:  $w \in \{0, 1\}^{\ell(n)}$

- 1 Berechne  $G(s)$  für alle  $s \in \{0, 1\}^n$ .

AUSGABE:  $= \begin{cases} 0 & \text{falls } G(s) = w \text{ für ein } s \in \{0, 1\}^n \\ 1 & \text{sonst} \end{cases}.$

# Unterscheider $D$ mit beliebiger Laufzeit

**1. Fall:**  $w \in G(s)$ , d.h.  $w$  wurde mittels  $G$  generiert

- Dann gilt  $Ws[D(G(s)) = 1] = 0$ .

**2. Fall:**  $w = r \in_R \{0, 1\}^{\ell(n)}$ , d.h.  $w$  ist echt zufällig.

- Es gilt  $|\{y \in \{0, 1\}^{\ell(n)} \mid y = G(s) \text{ für ein } s \in \{0, 1\}^n\}| \leq 2^n$ .
- Damit ist  $r \in_R \{0, 1\}^{\ell(n)}$  mit  $Ws \leq 2^{n-\ell(n)}$  im Bildraum von  $G$ .
- D.h.  $Ws[D(r) = 1] \geq 1 - 2^{n-\ell(n)} \geq \frac{1}{2}$  wegen  $\ell(n) > n$ .

Daraus folgt insgesamt  $Ws[D(r) = 1] - Ws[D(G(s)) = 1] \geq \frac{1}{2}$ .