

Modes of Operation – Electronic Code Book (ECB)

Ziel: Verschlüsseln von Nachrichten $m = m_1 \dots m_\ell \in (\{0, 1\}^n)^\ell$ mittels Blockchiffre unter Verwendung kleiner Nachrichtenexpansion.

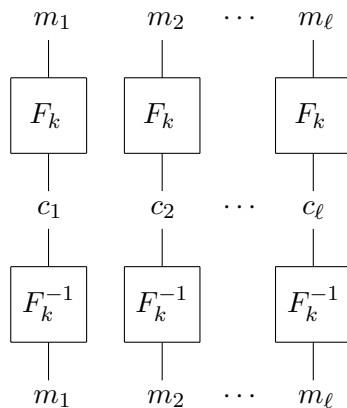
Algorithmus Electronic Code Book (ECB) Modus

- 1 **Enc:** $c := (F_k(m_1), \dots, F_k(m_\ell))$
- 2 **Dec:** $m := F_k^{-1}(c_1), \dots, F_k^{-1}(m_\ell)$

Nachteil:

- Enc ist deterministisch, d.h. ECB ist nicht mult-KPA sicher.
- Daher sollte der ECB Modus nie verwendet werden.

ECB



Modes of Operation – Cipher Block Chaining (CBC)

Algorithmus Cipher Block Chaining (CBC) Modus

❶ **Enc:** Wähle Initialisierungsvektor $c_0 := IV \in_R \{0, 1\}^n$. Berechne

$$c_i := F_k(c_{i-1} \oplus m_i) \quad \text{für } i = 1, \dots, \ell.$$

❷ **Dec:** Für $c = (c_0, c_1, \dots, c_\ell)$ berechne

$$m_i := F_k^{-1}(c_i) \oplus c_{i-1} \quad \text{für } i = 1, \dots, \ell.$$

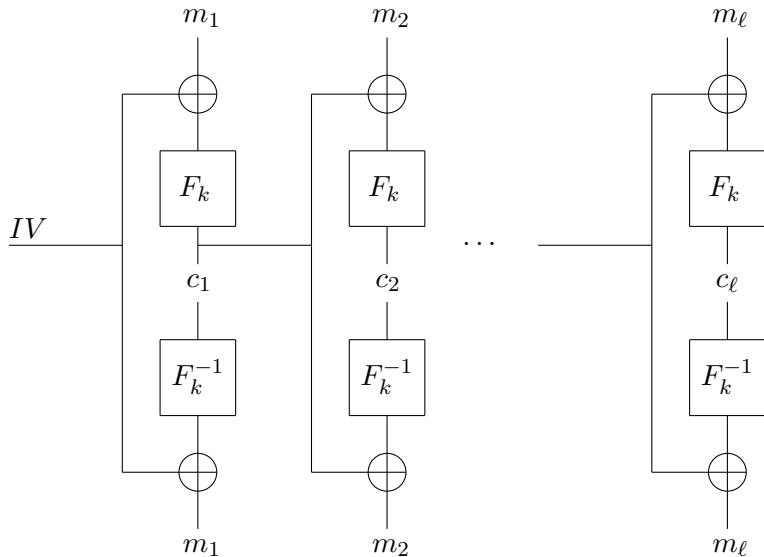
Vorteile:

- CPA-Sicherheit von CBC kann gezeigt werden.
- Nachrichtenexpansion ist $\frac{\ell+1}{\ell}$.

Nachteil:

- Verschlüsselung muss sequentiell durchgeführt werden.

CBC



Modes of Operation – Output Feedback (OFB)

Algorithmus Output Feedback (OFB) Modus

- 1 Enc:** Wähle $r_0 := IV \in_R \{0, 1\}^n$. Berechne $r_i := F_k(r_{i-1})$ für $i \in [\ell]$,
 $c_i := r_i \oplus m_i$ für $i = 1, \dots, \ell$.
- 2 Dec:** Für $c = (r_0, c_1, \dots, c_\ell)$ berechne $r_i := F_k(r_{i-1})$ für $i \in [\ell]$,
 $m_i := c_i \oplus r_i$ für $i = 1, \dots, \ell$.

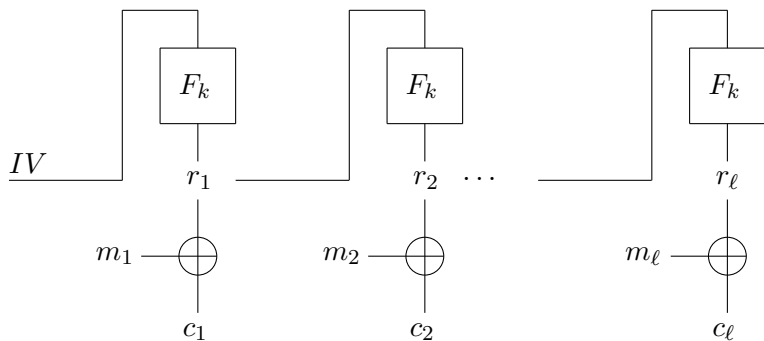
Vorteile:

- CPA-Sicherheit von OFB kann gezeigt werden.
- Nachrichtenexpansion ist $\frac{\ell+1}{\ell}$.
- Pseudozufallsfunktion F_k genügt, Invertierbarkeit nicht nötig.
- Die Berechnung der Zufallspads r_i kann unabhängig von der Nachricht nur mittels IV durchgeführt werden.

Nachteil:

- Berechnung der Zufallspads r_i ist sequentiell.

OFB



Modes of Operation – Counter (CTR)

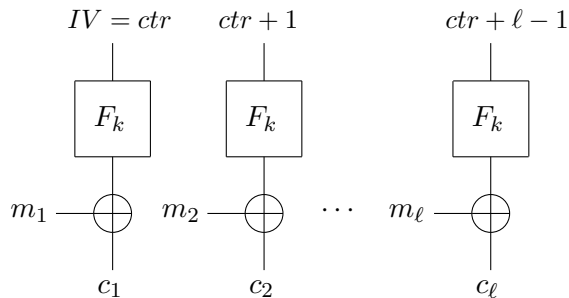
Algorithmus Counter (CTR) Modus

- Enc:** Wähle $ctr := IV \in_R \{0, 1\}^n$.
Berechne $r_i := F_k(ctr + i - 1 \bmod 2^n)$ für $i \in [\ell]$ und
$$c_i := r_i \oplus m_i \quad \text{für } i = 1, \dots, \ell.$$
- Dec:** Für Chiffretexte $c = (ctr, c_1, \dots, c_\ell)$:
Berechne $r_i := F_k(ctr + i - 1 \bmod 2^n)$ für $i \in [\ell]$ und
$$m_i := c_i \oplus r_i \quad \text{für } i = 1, \dots, \ell.$$

Vorteile:

- CPA-Sicherheit von CTR kann gezeigt werden.
- Nachrichtenexpansion ist $\frac{\ell+1}{\ell}$.
- Pseudozufallsfunktion F_k genügt, Invertierbarkeit nicht nötig.
- Berechnung der Zufallspads r_i unabhängig von der Nachricht.
- Ver-/Entschlüsselung sind vollständig parallelisierbar.

CTR



Sicherheit des Counter Modes

Satz Sicherheit des CTR Modes

Sei F eine PRF. Dann ist der CTR Modus CPA-sicher.

Beweisskizze:

- Können Unterscheider D mittels CPA-Angreifers \mathcal{A} konstruieren.
- Beweis verläuft analog zum Beweis der CPA-Sicherheit von Π_B .
- Pseudozufälliges Pad $F_k(r)$ darf nicht wiederverwendet werden.
- Hier verbraucht aber jede $Enc(\cdot)$ -Anfrage einen Block von Pads.

Sicherheit des Counter Modes

Beweisskizze: Fortsetzung

- Sei $q(n)$ eine polynomielle obere Schranke sowohl für
 - ▶ die Anzahl der Anfragen von \mathcal{A} an das Orakel $Enc(\cdot)$ als auch für
 - ▶ die Anzahl der zu verschlüsselnden Blöcke.
- D.h. $Enc(m)$ -Anfragen erfolgen für $m \in (\{0, 1\}^n)^\ell$ mit $\ell \leq q(n)$.
- Jede solche Anfrage verbraucht ein Intervall $ctr \dots ctr + \ell - 1$ von Pads $F_k(ctr), \dots, F_k(ctr + \ell - 1)$ der Länge höchstens $q(n)$.
- Sei I das Intervall zum Verschlüsseln der Challenge m_b .
- Sei I_j das Intervall aus der j -ten $Enc(\cdot)$ -Anfrage für $j \leq q(n)$.
- $Ws[PrivK_{\mathcal{A}, \Pi_{ctr}}^{cpa}(n) = 1] = 1$, falls sich I mit einem I_j überschneidet.
- $Ws[I \text{ überschneidet sich mit } I_j] \leq \frac{2q(n)}{2^n}$ für jedes feste j .
- Damit $Ws[I \text{ überschneidet sich mit einem der } I_j] \leq \frac{2q^2(n)}{2^n} = \text{negl}(n)$.

Anmerkung: Wahl der Blocklänge

- Vorige Beweisskizze zeigt Angriff mittels Intervallüberschneidung.
- Erreichen Erfolgsws der Größenordnung $\frac{q(n)^2}{2^n}$ für Blocklänge n .
- D.h. wir erhalten einen generischen Angriff für Blockchiffren mit Hilfe von $q(n) = 2^{\frac{n}{2}}$ Anfragen von Nachrichten $m \in (\{0, 1\}^n)^{q(n)}$.
- Damit muss nicht nur die Schlüssellänge einer Blockchiffre hinreichend groß gewählt werden, sondern auch die Blocklänge n .

CCA-Spiel

Szenario: CCA-Sicherheit

- \mathcal{A} erhält im *PrivK*-Spiel Zugriff auf Orakel $Enc_k(\cdot)$ und $Dec_k(\cdot)$.

Spiel CCA Ununterscheidbarkeit von Chiffretexten $PrivK_{\mathcal{A}, \Pi}^{cca}(n)$

Sei Π ein Verschlüsselungsverfahren und \mathcal{A} ein Angreifer.

- 1 $k \leftarrow Gen(1^n)$.
- 2 $(m_0, m_1) \leftarrow \mathcal{A}^{Enc_k(\cdot), Dec_k(\cdot)}(1^n)$, d.h. \mathcal{A} darf $Enc_k(m)$ und $Dec_k(c')$ für beliebige m und c' anfragen.
- 3 Wähle $b \in_R \{0, 1\}$ und verschlüssele $c \leftarrow Enc_k(m_b)$.
- 4 $b' \leftarrow \mathcal{A}^{Enc_k(\cdot), Dec_k(\cdot)}(c)$, d.h. \mathcal{A} darf $Enc_k(m)$ und $Dec_k(c')$ für beliebige m und $c' \neq c$ anfragen.
- 5 $PrivK_{\mathcal{A}, \Pi}^{cca}(n) = \begin{cases} 1 & \text{für } b = b' \\ 0 & \text{sonst} \end{cases}$.

Anmerkung:

- Ohne die Einschränkung $c' \neq c$ kann \mathcal{A} das Spiel stets gewinnen.

CCA Spiel

$\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cca}}(n)$

$k \leftarrow \text{Gen}(1^n)$

$\hat{c}_i \leftarrow \text{Enc}_k(\hat{m}_i)$

$m'_j := \text{Dec}_k(c'_j)$

$b \in_R \{0, 1\}$

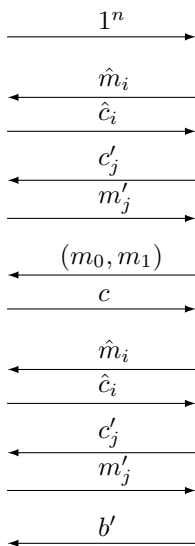
$c \leftarrow \text{Enc}_k(m_b)$

$\hat{c}_i \leftarrow \text{Enc}_k(\hat{m}_i)$

$m'_j := \text{Dec}_k(c'_j)$

Ausgabe:

$= \begin{cases} 1 & \text{falls } b = b' \\ 0 & \text{sonst} \end{cases}$



\mathcal{A}

Für alle $i, j \leq q$ wähle

$\hat{m}_i \in \mathcal{M}, c'_j \in \mathcal{C}$

$m_0, m_1 \in \mathcal{M}$

Von nun an wähle

$c'_j \in \mathcal{C} \setminus \{c\}$

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

Definition CCA Sicherheit

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

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

Der Wsraum ist definiert über die Münzwürfe von \mathcal{A} und $PrivK_{\mathcal{A}, \Pi}^{cca}$.

Notation: Wir bezeichnen Π als *CCA-sicher*.

CCA-Unsicherheit von Π_B

Beobachtung: Π_B ist nicht CCA-sicher.

Algorithmus CCA-Angreifer \mathcal{A} für Π_B

EINGABE: 1^n , Zugriff auf Orakel $Enc(\cdot)$ und $Dec(\cdot)$

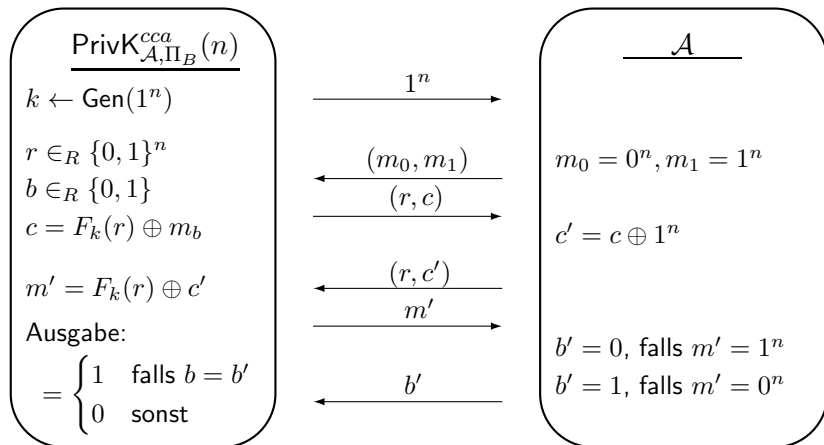
- 1 Wähle $(m_0, m_1) = (0^n, 1^n)$
- 2 Erhalte Chiffretext $c = (c_1, c_2) = (r, F_k(r) \oplus m_b)$.
- 3 Berechne $c' = (c_1, c_2 \oplus 1^n)$. Sei m' die Antwort auf die Entschlüsselungsanfrage $Dec_k(c')$.

AUSGABE: $b' = \begin{cases} 0 & \text{für } m' = 1^n \\ 1 & \text{sonst} \end{cases}$.

Es gilt $\mathbb{W}_s[\text{PrivK}_{\mathcal{A}, \Pi_B}^{\text{cca}}(n) = 1] = 1$.

Ziel: Werden CCA-sichere Verschlüsselung mittels sogenannter MACs (Message Authentication Codes) konstruieren.

CCA-Unsicherheit von Π_B



Problem: Wir können einen Chiffretexte so manipulieren, dass er die gültige Verschlüsselung einer anderen Nachricht ist.

Integrität und Authentizität von Nachrichten

- 1 **Integrität:** Überprüfen, dass eine Nachricht nicht verändert wurde.
- 2 **Authentizität:** Überprüfen, dass eine Nachricht wirklich vom Absender kommt.

Ziel: Können Angriffe nicht verhindern, müssen sie aber erkennen.

Anm.: Verschlüsselung liefert weder Integrität noch Authentizität.

- Bsp: Verwenden unsere CPA-sichere Verschlüsselung Π_B .
- Chiffretexte besitzen Form $c = (c_1, c_2) = (r, F_k(r) \oplus m) \in \{0, 1\}^{2n}$.

Keine Integrität:

- Sei e_i ein Einheitsvektor der Länge n . Dann ist $c' = (c_1, c_2 \oplus e_i)$ eine gültige Verschlüsselung von $m' = m + e_i$.
- D.h. \mathcal{A} kann beliebige Bits von m verändern, ohne m zu kennen.

Keine Authentizität:

- Jedes $c = (c_1, c_2)$ ist eine Verschlüsselung von $m = F_k(c_1) \oplus c_2$.
- D.h. \mathcal{A} kann ein gültiges c erzeugen, ohne k zu kennen.