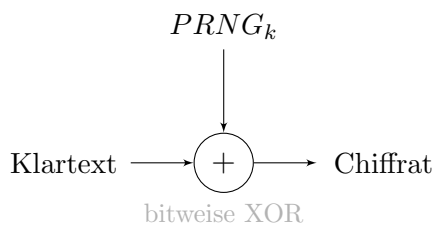


# III. Symmetrische Verschlüsselung

**FIXME:** Definition vernachlässigbare Funktion, S. 4

## III.1. Stromchiffren

pseudo-random number generator zum key  $k$



### III.1.1. Anforderungen

- PRNG muss effizient berechenbar sein
- Pseudozufall ununterscheidbar von echtem Zufall  
(formal: gegeben Orakel  $\mathcal{O}_{ideal}$ , welches echten Zufall ausgibt, und  $\mathcal{O}_{real}$ , welches  $PRNG_k$  mit geheimem Schlüssel  $k$  implementiert, gilt für alle (Polynomialzeit)Angreifer  $\mathcal{A}$ :

$$|Pr[\mathcal{A}^{\mathcal{O}_{ideal}} \rightarrow 0] - Pr[\mathcal{A}^{\mathcal{O}_{real}} \rightarrow 0]|$$

ist vernachlässigbar in  $|k|$ ).

## III.2. Blockchiffren

### III.2.1. Definition

Seien  $k$  ein Schlüssel aus dem Schlüsselraum  $\mathcal{K}$ ,  $A$  und  $B$  sind Ein- bzw. Ausgabealphabete und  $n$  und  $m$  die zugehörigen Blocklängen. Eine Blockchiffre ist eine Familie von injektiven Abbildungen  $\{f_k: A^n \rightarrow B^m\}_{k \in \mathcal{K}}$ .

**FIXME:** Bild Blockchiffre, S. 4

### III.2.2. Anforderungen

- gegeben ein Orakel  $\mathcal{O}_{ideal}$ , welches eine zufällige Injektion  $A^n \rightarrow B^m$  implementiert, und  $\mathcal{O}_{real}$ , welches  $f_k$  mit geheimem Schlüssel  $k$  implementiert, gilt für alle (Polynomialzeit)Angreifer  $\mathcal{A}$ :  $|Pr[\mathcal{A}^{\mathcal{O}_{ideal}} \rightarrow 0] - Pr[\mathcal{A}^{\mathcal{O}_{real}} \rightarrow 0]|$  ist vernachlässigbar in  $|k|$ .
- gegeben  $k$ , müssen  $f_k$  und  $f_k^{-1}$  effizient berechenbar sein

### III.2.3. Beispiel: DES (Data Encryption Standard)

**FIXME:** Bild DES, S. 5

#### Eigenschaften

- bis heute strukturell ungebrochen
- aber: Schlüssel zu kurz (Brute-Force-Attacken sind heute praktikabel)

→ Abhilfe: 3DES (Chiffre =  $DES_{k_3}(DES_{k_2}^{-1}(DES_{k_1}(Nachricht))))$ )  
→ Warum nicht 2DES? Antwort: Meet-in-the-Middle-Attacken

#### Meet-in-the-Middle (gegen 2DES):

**FIXME:** Bild Meet in the Middle, S. 6

Known-Plaintext-Angriff, gegeben ein Klartext-Chiffre-Paar  $(M, C)$ :

1. Vorwärts-Schritt: Tabelliere  $(DES_k(M), k)$  für alle Schlüssel  $k \in \{0, 1\}^{56}$ .
2. Sortiere die Tabelle.
3. Rückwärts-Schritt: Für jedes  $k \in \{0, 1\}^{56}$  berechne  $(DES_k(C))$  und suche Tabelleneintrag mittels binärer Suche.

**Aufwand:**  $\approx 56 \cdot 2^{56}$  für das binäre Sortieren,  $\approx 2^{56}$  für die binäre Suche, insgesamt also nur  $\approx 56$  mal mehr Aufwand als bei DES

### III.2.4. Beispiel: Rijndael/AES (Advanced Encryption Standard)

- 128 bit Blocklänge
- 3 Varianten:
  - 128, 192, 256 bit Schlüssel
  - 10, 12, 14 Runden
- Darstellung von State und Rundenschlüssel als  $4 \times 4$ -Byte-Matrix
- Ablauf einer Runde in 4 Schritten: **FIXME:** Bild AES, S. 7
- Bestimmen der Rundenschlüssel:
  - teile Schlüssel in 4-Byte-Worte
  - berechnen  $W[i] := W[i - 4] \oplus W[i - 1]$  und ab und zu Byteinvertierungen
- mögliche Schwäche: AES lässt sich als geschlossene algebraische Gleichung schreiben und ist damit theoretisch mathematisch brechbar

### III.2.5. Betriebsmodi

#### ECB (Electronic Codebook Mode)

**FIXME:** Bild ECB, S. 8

Nachteile:

- gleiche Klartextblöcke werden auf gleiche Chiffratblöcke abgebildet
- Angreifer kann Blöcke vertauschen, löschen, duplizieren

Vorteile:

- Übertragungsfehler (Bitflips) auf den betroffenen Block begrenzt<sup>1</sup>
- perfekt parallelisierbar (zum Ver- und Entschlüsseln der Blöcke ist jeweils nur der Schlüssel nötig)
- verschlüsselte Datenspeicher blockweise bearbeitbar

#### CBC (Cipher Block Chaining)

**FIXME:** Bild CBC, S. 8

Vorteile:

- Nachteile von ECB beseitigt
- Enschlüsselung selbstsynchronisierend (Klartextblock wird nur aus den letzten beiden Chiffratblöcken berechnet) → wahlfreier Lesezugriff

Nachteile:

- geringer Bandbreitenverlust, da Initialisierungsvektor übertragen werden muss
- Fehler breiten sich auf einen weiteren Block aus

#### OFB (Output Feedback Mode)

**FIXME:** Bild OFB, S. 9

Vorteile:

- Entschlüsselung muss nicht effizient sein
- keine Fehlerübertragung bei Bitflips
- Pseudozufallsstrom vorberechenbar

Nachteile:

- gleicher Initialisierungsvektor bewirkt:  $c_1 \oplus c_2 = m_1 \oplus m_2$
- nicht robust gegen Verlorengangen ganzer Blöcke
- Angreifer kann gezielt Klartextbits kippen

---

<sup>1</sup>aber: betrifft den ganzen Block, da Verschlüsselung jedes einzelnen Bits im Block von jedem Bit im Block abhängig

**CTR (Counter Mode)**

**FIXME:** Bild CTR, S. 9

Nachteile:

- wie OFB

Vorteile (wie OFB und ECB):

- gut parallelisierbar
- Pseudozufallsstrom vorberechenbar
- Fehlerfortpflanzung auf Blöcke begrenzt
- wahlfreier Zugriff auf verschlüsselten Speicher
- muss nicht invertierbar sein