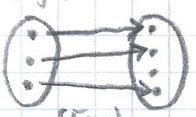
CRYPTO2-Übersicht: VL02: (Betriebsmodi) allg. Blockchiffren; OTP; CPA

Definition Blockchiffre: Sei $\mathbb{B} = \{0,1\} = \mathbb{F}_2$ der Körper mit 2 Elementen ($+ = \text{XOR}$, $\cdot = \text{AND}$) und $\mathbb{B}^\lambda = \{(0,\dots,0),\dots,(1,\dots,1)\}$ Menge aller Tupel d. Länge λ. Dann besteht eine Blockchiffre mit Blocklänge λ_b und Schlüssellänge λ_k aus den zwei Funktionen

- $\text{Enc}: \mathbb{B}^{\lambda_k} \times \mathbb{B}^{\lambda_b} \to \mathbb{B}^{\lambda_b}$
- $\text{Dec}: \mathbb{B}^{\lambda_k} \times \mathbb{B}^{\lambda_b} \to \mathbb{B}^{\lambda_b}$

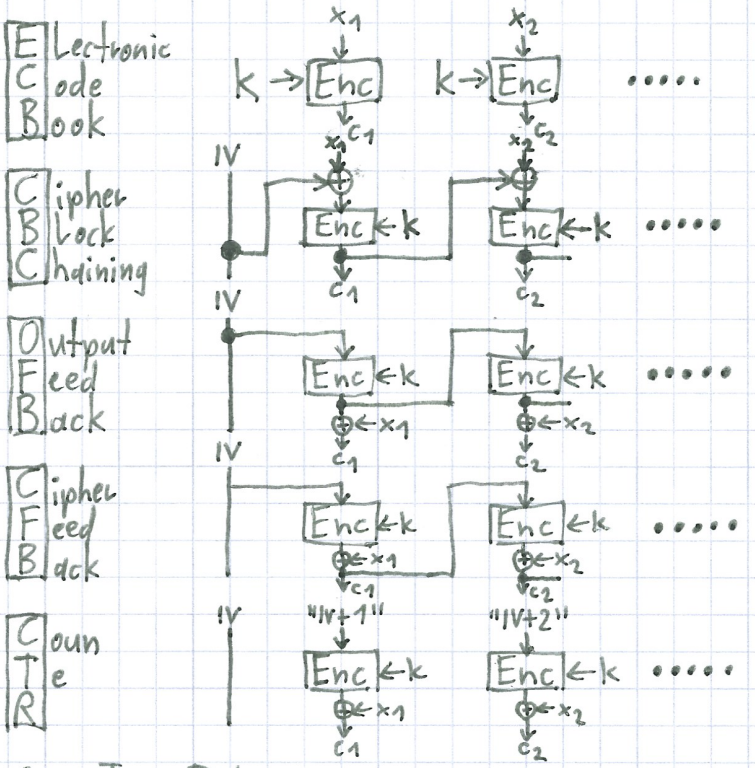
mit der Eigenschaft $\text{Dec}(k, \text{Enc}(k,x)) = x$.

surjektiv: 

injektiv: 

Weitere Eig.:
- $\text{Enc}(k,\cdot)$ und $\text{Dec}(k,\cdot)$ sind bijektiv $\Leftarrow$ Enc injektiv, Dec surj., $|\mathbb{B}^{\lambda_b}| = |\mathbb{B}^{\lambda_b}| < \infty$
- $\text{Enc}(k, \text{Dec}(k,c)) = c$ $\Leftarrow \text{Enc}(k,x) = \text{Dec}^{-1}(k,x) \Leftarrow \text{Dec}(k,\text{Enc}(k,x)) = x$
  ($\Leftarrow$ da Dec surjektiv, exist. c; $\Rightarrow$ Dec^{-1} exist., da Dec bijektiv)

Betriebsmodi: (hier alles Betriebsmodi ohne Authentifizierung; mit: CCM, GCM, EAX)



- ECB = deterministisch ↯ (da kein IV) ↯⇒ nicht IND-CPA-sicher (s.u.) ⇒ schicke nur $(x_1,x_2),(x_1,x_3)$
- CBC ✓ Klartextmuster werden zerstört; ↯ n. parallelisierbar; ↯ Bitfehler multiplizieren sich
- OFB = Stromchiffre = „OTP für Arme" = „Streckung des IV ins ∞e"; ✓ statt AES ginge auch Hashfkt.
- CFB = Stromchiffre = Selbstsynchronisation (Zustand/IV müssen n. bekannt sein!)
- CTR = Stromchiffre (wie OFB); ✓ Bitfehler verurs. geringsten Schaden; ✓ wahlfreier Zugriff; parallelisierbar

One Time Pad:
Beim OTP gilt: Länge der Nachricht $= \lambda_b = \lambda_k =$ Länge des Schlüssels $= \lambda$
Enc: $c = x + k$ Dec: $x = c + k$
Wenn k jedes Mal gleichförmig zufällig gewählt wird ($Pr = \frac{1}{2^\lambda}$), ist OTP absolut sicher!
Problem: Schlüssel jedes Mal neu!

Claude Shannon:
- **Konfusion:** Zusammenhang x, c, k sehr komplex!
- **Diffusion:** Ändern eines Bits von x ändert jedes Bit von c mit Wk. $\frac{1}{2}$. (sonst könnte man Angriff auf ein paar Bit beschränken & Brute-Force machen!)

Kerckhoffs' Forderung/Maxime: die Sicherheit einer (Block-)Chiffre liegt nur in der Geheimhaltung des Schlüssels

IND-CPA (Indistinguishability Chosen Plaintext Attack) = Bsp. für Sicherheitsdefinition: C = Challenger/Herausforderer; A = Angreifer

Kriege ich wenigstens 1 bit an Info?!

A darf C polynomiell in der Schlüssellänge λ_k oft zwei Plaintexte (x_0,x_1) schicken.
C verschlüsselt stets entweder den 1. oder 2. Plaintext & schickt den Ciphertext an A.
A darf dennoch nicht in der Lage sein mit Wk. >50% zu sagen, ob C immer den ersten ($b=0$) oder den zweiten ($b=1$) Plaintext verschlüsselt hat. (d.h. nicht mal 1 bit an Info)

$b \in \{0,1\}$ kennt nur C

1. C: $k \in_R \mathbb{B}^{\lambda_k}$
2. A: $x_0, x_1 \to C$ } wdh.
3. C: $\text{Enc}(k, x_b) \to A$ } poly. oft
4. A: $b' \to C$ mit Behauptg. $b = b'$

Def.: Ein Chiffre heißt IND-CPA-sicher, wenn:

$$\left| Pr[\text{IND-CPA}_0(A, \lambda_k) = 1] - Pr[\text{IND-CPA}_1(A, \lambda_k) = 1] \right| < \epsilon(\lambda) < \frac{1}{p(\lambda)}$$

(IND-CPA_b: b ∈R {0,1}) für jedes Polynom p