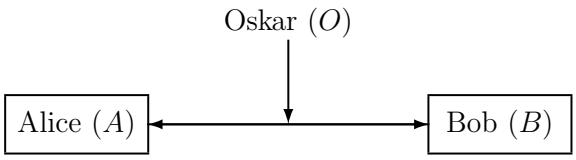
Vorlesung am 21.04.2015

3 Symmetrische Verschlüsselung

- Alice (A) und Bob (B) wollen sicher kommunizieren (vgl. Schutzziele)
- Oskar (O) versucht, die Schutzziele zu durchbrechen
  - Passiver Angriff: Abhören der Daten
  - Aktiver Angriff: Manipulation (z.B. Fälschung) der Daten



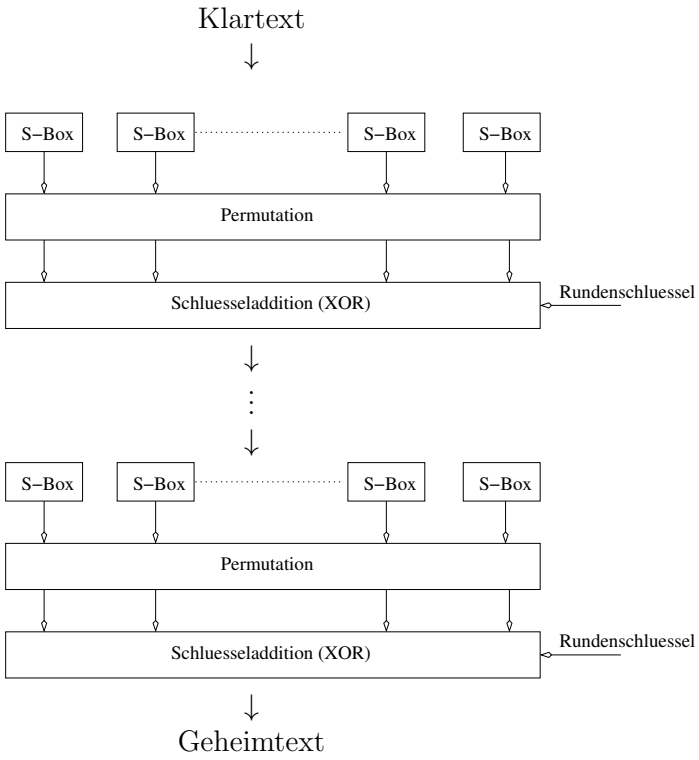
Verschlüsselung: Schutzziels Vertraulichkeit (passiver Angriff)
Symmetrische Verschl.: Schlüssel zum ver- und entschlüsseln sind gleich

Wir lernen kennen: Blockchiffren (inkl. Betriebsmodi), Stromchiffren

- Einsatz für sehr große Datenmengen: Effizient in Soft-/ Hardware
- Nutzung einfacher Grundfunktionen: $\oplus$, Listenauswertung

Blockchiffren: Konstruktionsprinzipien nach Shannon 1948:
Eine Blockchiffre ist eine Abb. $F : \underbrace{\{0,1\}^n}_{Klartexte} \times \underbrace{\{0,1\}^m}_{Schluessel} \longrightarrow \underbrace{\{0,1\}^n}_{Geheimtexte}$

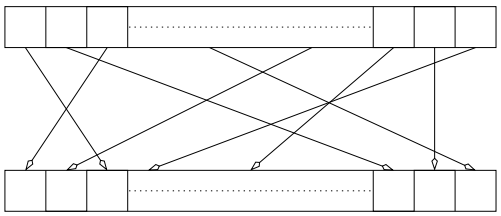
- Zunächst nur für kleine Nachrichten (z.B. $x \in \{0,1\}^{128}$)
- Bitlänge n heißt auch Blockgröße
- Für längere Nachrichten: Betriebsmodi (Modes of Operation)



Grundbausteine: Substitution, Permutation (jeweils Umsetzung über Listen)
Schlüsseladdition: $\oplus$

Permutationen Blockgröße $n = 128$.

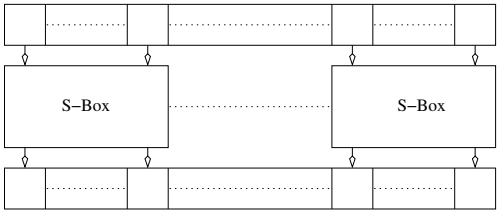
- Spezielle lineare Abbildung $P : \{0,1\}^n \longrightarrow \{0,1\}^n$.



- Eff. Implementierung:: $n-$Tupel $(x_1, \ldots, x_n)$ wird mittels bij. Abb. $\pi : \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}$ zu $(x_{\pi(1)}, \ldots, x_{\pi(n)})$ permutiert.

Substitutionen

- Nichtlineare Abbildung $S : \{0,1\}^n \longrightarrow \{0,1\}^n$
- Implementierung als array über den gesamten Block nicht möglich. 2^{128} Bitstrings müssen abgebildet werden: Länge array: 2^{128}
- Beschränkung auf Teilblöcke Länge 8 oder 16 (parallele Ausführung)



Effiziente Implementierung:

Interpretiere Bitstrings der Länge 8 als natürliche Zahl $0, 1, \ldots 2^8 - 1 = 255$
der Länge 16 als natürliche Zahl $0, 1, \ldots, 2^{16} - 1 = 65.536$

Schlüsseladdition XOR ($\oplus$) eines Rundenschlüssel
Schlüsselexpansion: Erzeuge aus einem Schlüssel mehrere Rundenschlüssel

Erreichen zweier Ziele:

1. Diffusion (Durchmischung): Permutationen
2. Konfusion (Komplexität/Nichtlinearität): Substitutionen

Wiederholtes Anwenden von Diffusion und Konfusion erhöht die Sicherheit.

Bsp.: Substitutions-Permutations-Netzwerk: Advanced Encryption Alg. (AES)

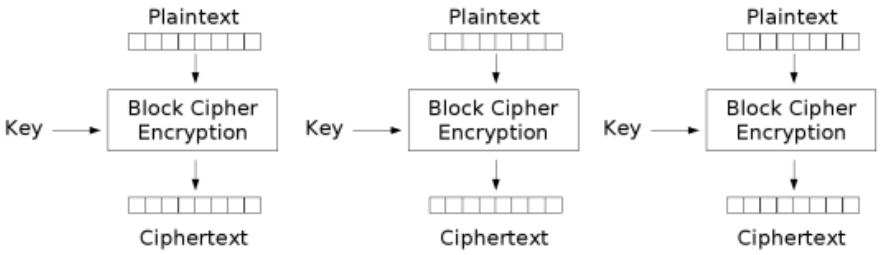
Übung: Sicherheitsschwächen bei Weglassen 1) Permutation, 2) Substitution
Übung:

Betriebsarten: Bisher nur Verschlüsselung von Bitblöcken (z.B. 128 Bit).

Für längere Nachrichten:

- Teilung der Nachricht in Blöcke (Länge = Blockgröße der Blockchiffre).
- Wenn letzter Block zu klein: auffüllen (Padding)

Electronic Code Book (ECB): (Einfachste Lösung)



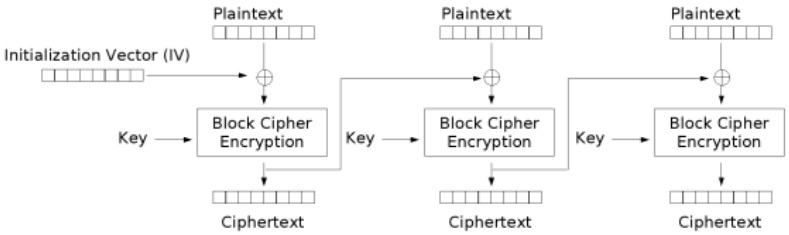
Electronic Codebook (ECB) mode encryption

Nachteil: Gleiche Klartextblöcke führen zu gleichen Geheimtextblöcken.

Angreifer erkennt, ob gleiche Texte verschlüsselt wurden.

Also: $n-$ter Geheimtextblock sollte nicht nur von $n-$ten Klartextblock und Schlüssel abhängen, sondern von einem weiteren Wert.

Cipher Block Chaining Weiterer Wert: $(n-1)-$ter Geheimtextblock.

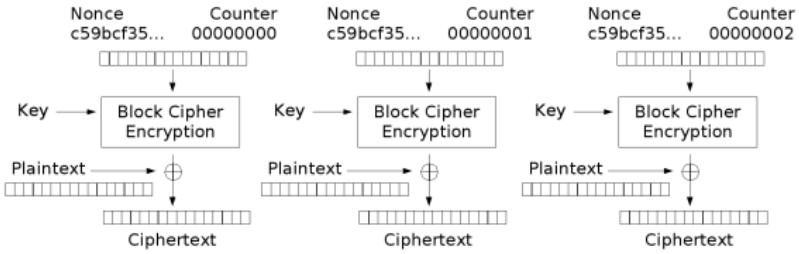


Cipher Block Chaining (CBC) mode encryption

Für ersten Block wird ein Initialisierungsvektor benötigt.

Übung: Watermark-Angriff

Counter Mode Weiterer Wert: Zähler.

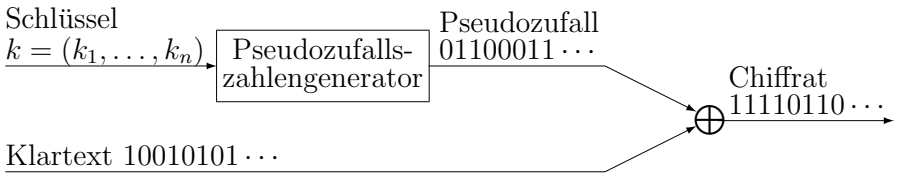


Counter (CTR) mode encryption

Übung: Wie wird entschlüsselt, wie wird bei Bitfehlern synchronisiert?

Stromchiffren

- Erzeuge aus Schlüssel $k \in \{0,1\}^n$ pseudozufälligen Schlüsselstrom
- Schlüsselstrom wird komponentenweise mit Klartext addiert (XOR)



Bsp.: Blockchiffre als Zufallszahlengenerator (Counter Mode)

Welche Bedingungen muss der Pseudozufallszahlengenerator erfüllen?

- Erste Idee: $k \to k, k, k, \ldots$ (unsicher, siehe Modifikation OTP)
- Aus 100 Bit Zufall lässt sich nicht 200 Bit Zufall (determ.) berechnen
  - 200 Bit Zufall: Angreifer rät Schlüsselstrom mit Wkeit $1/2^{200}$
  - Er muss aber nur Schlüssel raten (Wahrscheinlichkeit $1/2^{100}$) und dann den Schlüsselstrom berechnen
- Wir benötigen nur praktische Sicherheit
- Aus Angreifersicht (beschränkte Ressourcen) kein Unterschied zwischen
  - echtem Zufall
  - Pseudozufall, der von einem Pseudozufallszahlengenerator stammt
- Minimalforderung:
  - Aus Teilen des Schlüsselstroms keine Nachfolger bestimmbar
  - Aus Teilen des Schlüsselstroms keine Vorgänger bestimmbar
- Ansonsten Angriff wie beim modifizierten One-time Pad möglich