

Alle Angaben ohne Gewähr. Keine Garantie auf Vollständigkeit oder Richtigkeit.

1	Einführung	2
1.1	Ausgangspunkte für Angriffe	2
1.2	Angriffsarten	2
1.3	Historische Verschlüsselungsverfahren	2
2	Blockchiffren	2
2.1	Definitionen	2
2.1.1	Definition: Blockchiffre	2
2.1.2	Anforderungen an Blockchiffren	2
2.1.3	Definition: Ideal Cipher	2
2.1.4	Anforderungen an Ideal Cipher	3
2.2	DES (Data Encryption Standard)	3
2.2.1	Beispiel für Encryption-Schritt	3
2.2.2	Beispiel für Decryption-Schritt	3
2.2.3	F-Funktion	3
2.2.4	2DES	4
2.2.5	Angriff auf 2DES	4

1 Einführung

1.1 Ausgangspunkte für Angriffe

Angriffe können nach den zur Verfügung stehenden Informationen unterteilt werden:

- **Ciphertext-Only-Attack**: Nur das *Chiffre*, also die verschlüsselte Nachricht, ist bekannt
- **Known-Plaintext-Attack**: Es gibt bekannte Klartext-Chiffre-Paare. Hilfreich sind bekannte Anfangs- und Endphrasen, die in mehreren Nachrichten vorkommen.
- **Chosen-Plaintext-Attack**: Es besteht die Möglichkeit, beliebige Texte zu verschlüsseln und somit Klartext-Chiffre-Paare zu erzeugen.

1.2 Angriffsarten

- Brute-Force (z.B. alle Schlüssel ausprobieren)
- Statistische Methoden (z.B. Häufigkeitsanalysen von Buchstaben)
- Strukturelle Angriffe (z.B. Lineare Kryptoanalyse)

1.3 Historische Verschlüsselungsverfahren

Historisch wurden zur Verschlüsselung zwei grundlegende Operationen verwendet:

- **Substitution**
- **Permutation**

Alleine sind beide Verfahren meistens nicht sicher, jedoch verwenden moderne Verschlüsselungsverfahren eine Kombination beider Operationen.

2 Blockchiffren

2.1 Definitionen

2.1.1 Definition: Blockchiffre

Gegeben seien zwei endliche Alphabete A, B und $n, m \in \mathbb{N}$ sowie ein Schlüsselraum \mathcal{K} . Eine **Blockchiffre** ist gegeben durch eine Familie von injektiven Abbildungen $f_k : A^n \rightarrow B^m$ mit $k \in \mathcal{K}$. In der Regel gilt $A = B = \{0, 1\}$ und $n = m$.

2.1.2 Anforderungen an Blockchiffren

- Gegeben den Schlüssel k müssen sowohl f_k als auch f_k^{-1} **effizient** berechenbar sein
- Ein Angreifer soll nicht zwischen einer *zufälligen Abbildung* und der Blockchiffre mit *zufälligem Schlüssel* unterscheiden können

2.1.3 Definition: Ideal Cipher

Eine **Ideal Cipher** (IC) ist eine (Über-)Idealisierung einer Blockchiffre. Jedem Schlüssel $k \in \{0, 1\}^\lambda$ ist eine vollkommen zufällige Permutation $P_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$ zugeordnet (hierbei sind λ und n Sicherheitsparameter) und per Orakelzugriff kann jede Maschine im Modell die Funktionen P_k und P_k^{-1} auswerten. Die Existenz einer solchen IC wird zur Vereinfachung von Beweisen angenommen, man spricht dann von dem **Ideal-Cipher-Modell**.

2.1.4 Anforderungen an Ideal Cipher

- Alle Parteien können über Orakelzugriff P_k und P_k^{-1} auswerten
- Ideal Cipher liefert zu jedem Paar (k, m) ein c "zufällig" gewählt
- Ideal Cipher liefert zu jedem Paar (k, c) ein m "zufällig" gewählt
- Orakel muss jede Ausgabe speichern, damit für gleiche Nachrichten immer das gleiche Chiffre zurückgegeben wird (nicht parallelisierbar)

2.2 DES (Data Encryption Standard)

Der **Data Encryption Standard** ist eine Blockchiffre mit Schlüssellänge $k = 56$ und Blocklänge $n = 64$, die Verschlüsselungsfunktion ist also $\{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$. Er besteht aus einer **Feistel-Struktur** mit 16 Runden und wurden aufgrund der kurzen Schlüssellänge **gebrochen**.

2.2.1 Beispiel für Encryption-Schritt

$$L_1 = R_0$$

$$R_1 = L_0 \oplus F_{k_1}(R_0)$$

$$L_{16} = R_{15}$$

$$R_{16} = L_{15} \oplus F_{k_{16}}(R_{15})$$

2.2.2 Beispiel für Decryption-Schritt

$$R_{15} = L_{16}$$

$$L_{15} = R_{16} \oplus F_{k_{16}}(R_{15})$$

$$= R_{16} \oplus F_{k_{16}}(L_{16})$$

2.2.3 F-Funktion

Die **F-Funktion** ist eine nicht-reversible Funktion, die in jeder Runde der Feistel-Struktur ausgeführt wird. Der Ablauf ist folgender:

1. **Expansion:** Die 32 Eingabebits werden auf 48 Bits erweitert
2. Das bitweise XOR zwischen Expansion und dem Schlüssel wird berechnet
3. Das Ergebnis wird in 6-Bit-Blöcken auf 8 **Substitutionsboxen** verteilt
4. Die Substitution wird permutiert und ausgegeben

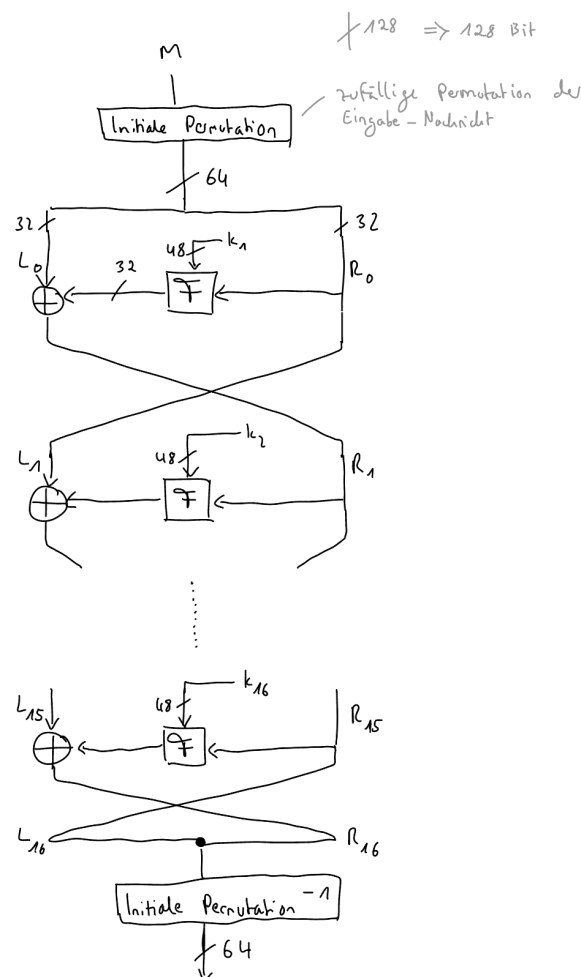


Abbildung 1: DES-Verschlüsselungsalgorithmus

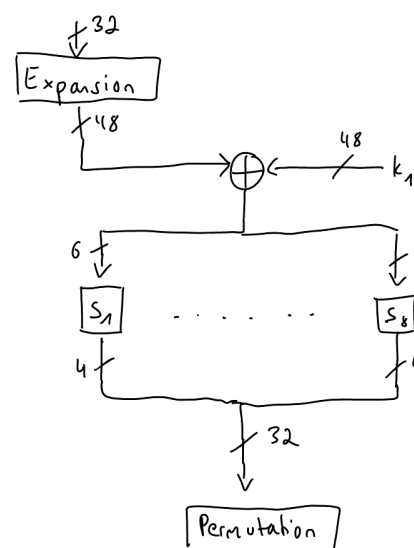


Abbildung 2: F-Funktion

2.2.4 2DES

DES wurde aufgrund der kurzen Schlüssellänge (56 Bit) gebrochen und sollte daher nicht mehr in seiner einfachen Form verwendet werden. Es gibt jedoch modifizierte Varianten, die die Sicherheit von DES verbessern, z.B. **2DES**:

Bei 2DES wird die Nachricht zuerst mit k_1 verschlüsselt und das Chiffre dann erneut mit k_2 verschlüsselt:
$$c = DES_{k_2}(DES_{k_1}(m))$$

2.2.5 Angriff auf 2DES

Gegen 2DES sind **Meet-in-the-Middle**-Angriffe möglich, dabei wird versucht, die Verschlüsselung von beiden Seiten zu brechen.

Gegeben sind zwei Paare (m_1, c_1) und (m_2, c_2) , dann funktioniert der Angriff wie folgt:

1. **Vorwärtsschritt**: Berechne $DES_k(m_1)$ und $DES_k(m_2)$ für alle $k = 0, \dots, 2^{56} - 1$ und speichere alle Werte in einer Tabelle T
2. **Sortierschritt**: Sortiere Tabelle T
3. **Rückwärtsschritt**: Berechne $DES_k^{-1}(c_1)$ und $DES_k^{-1}(c_2)$ für alle $k = 0, \dots, 2^{56} - 1$ und suche nach Treffern in T

Zeitaufwand des Angriffs:

1. **Vorwärtsschritt**: $2 * 2^{56}$ DES-Operationen
2. **Sortierschritt**: $56 * 2^{56}$ Vergleiche
3. **Rückwärtsschritt**: $2 * 2^{56}$ DES-Operationen (da man nach ungefähr der Hälfte fertig ist)

Speicheraufwand des Angriffs:

In der Tabelle müssen ungefähr 2^{60} Byte gespeichert werden, was ungefähr 1.150.000 TB entspricht. Damit ist der Angriff praktisch nicht praktikabel.