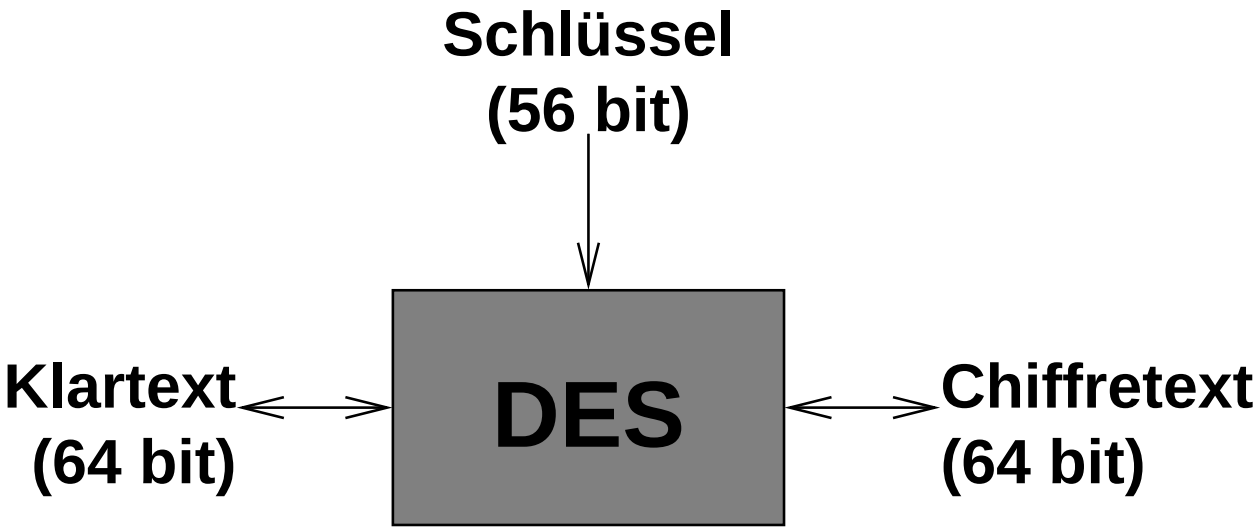
11: Der Data Encryption Standard (DES)

11: Der Data Encryption Standard (DES)



Geschichte des DES

1973-77 Zwei Ausschreibungen, ein geeigneter Kandidat (“Lucipher”) nach Überarbeitung als DES (“Data Encryption Standard”) standardisiert:

<u>64-Bit-Blockchiffre mit 56-Bit-Schlüsseln</u>

Ab 1977 Kritik an Schlüssellänge
Trotzdem große Akzeptanz und riesige Verbreitung

Ab 1990 Differentielle und lineare Kryptanalyse

1997 DES-Challenge (Tausende Rechner, 4 Monate)

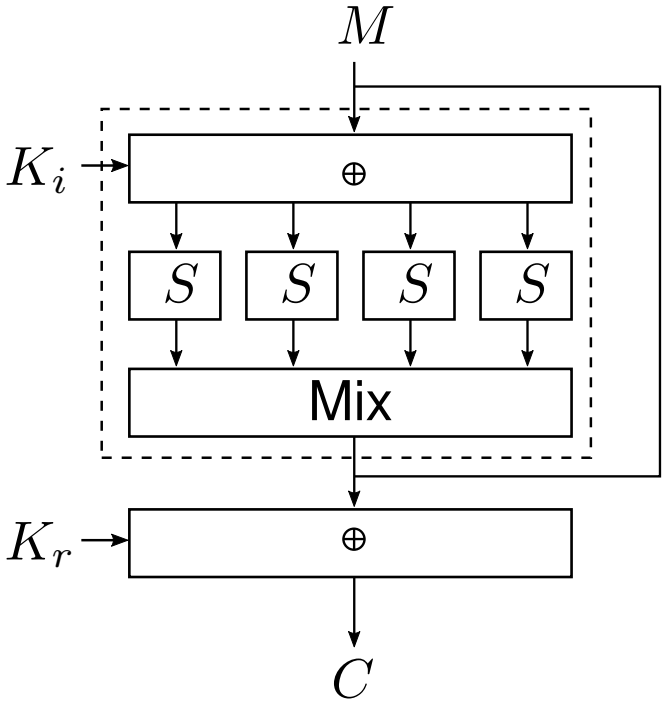
Blockchiffren: Konfusion und Diffusion

Shannon 1945:

- **Konfusion:** Jedes Bit des Chiffretexts hängt von möglichst vielen Bits des Schlüssels ab.
- **Diffusion:** Eine Änderung in einem Bit des Klartextes bewirkt (statistisch) eine Änderung von 50% der Bits des Chiffretexts.

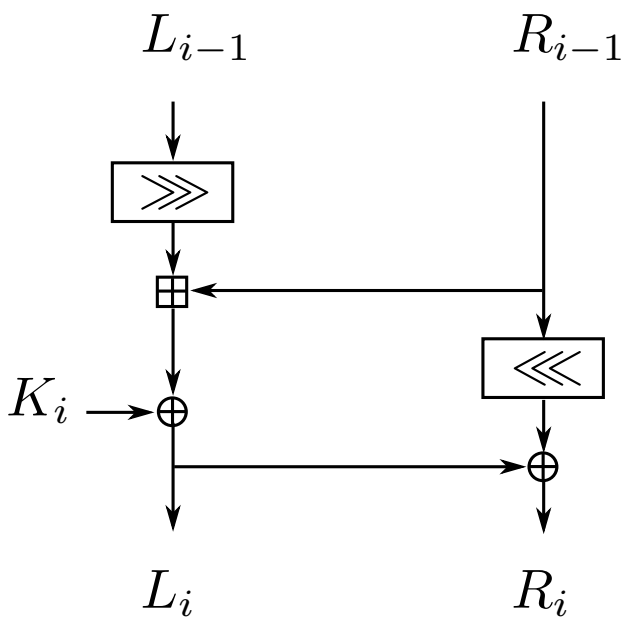
Blockchiffren: Komponenten

- **Produktchiffre** (Shannon):
  - Rundenfunktion
  - r Runden
  - Kompromiss zw. Sicherheit und Effizienz
- Komponenten:
  - Addition mit (Runden-)**Schlüssel** K_i (Konfusion)
  - **Lineare** Operation zur Durchmischung (Diffusion)
  - **Nichtlineare** Operation (z. B. Substitutionstabelle)



Häufigste Arten von Blockchiffren

- Feistel-Netzwerke
  - Entschlüsselung benötigt keine inverse Rundenfunktion
  - Z. B. P_2, P_3, P_4, DES, ...
- Substitution-Permutation-Netzwerke
  - Z. B. AES (später)
- ARX (Addition, Rotation, XOR)



Struktur des DES (Feistel-Netzwerk)

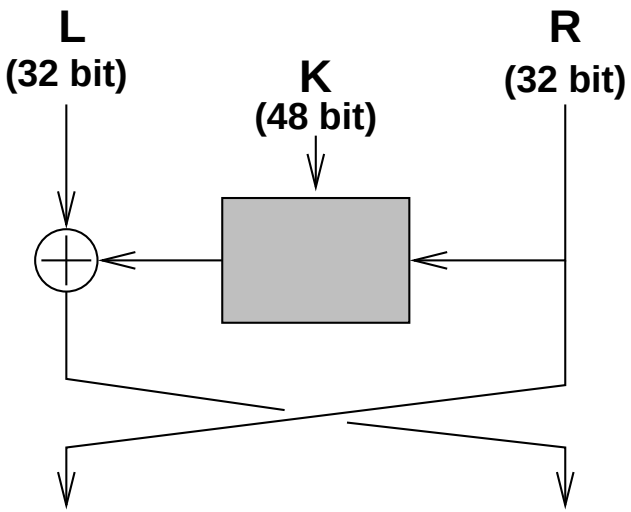
- Rundenfunktion:

$$f : \{0,1\}^{48} \times \{0,1\}^{32} \to \{0,1\}^{32}$$

- 16 Runden
- 16 Rundenschlüssel

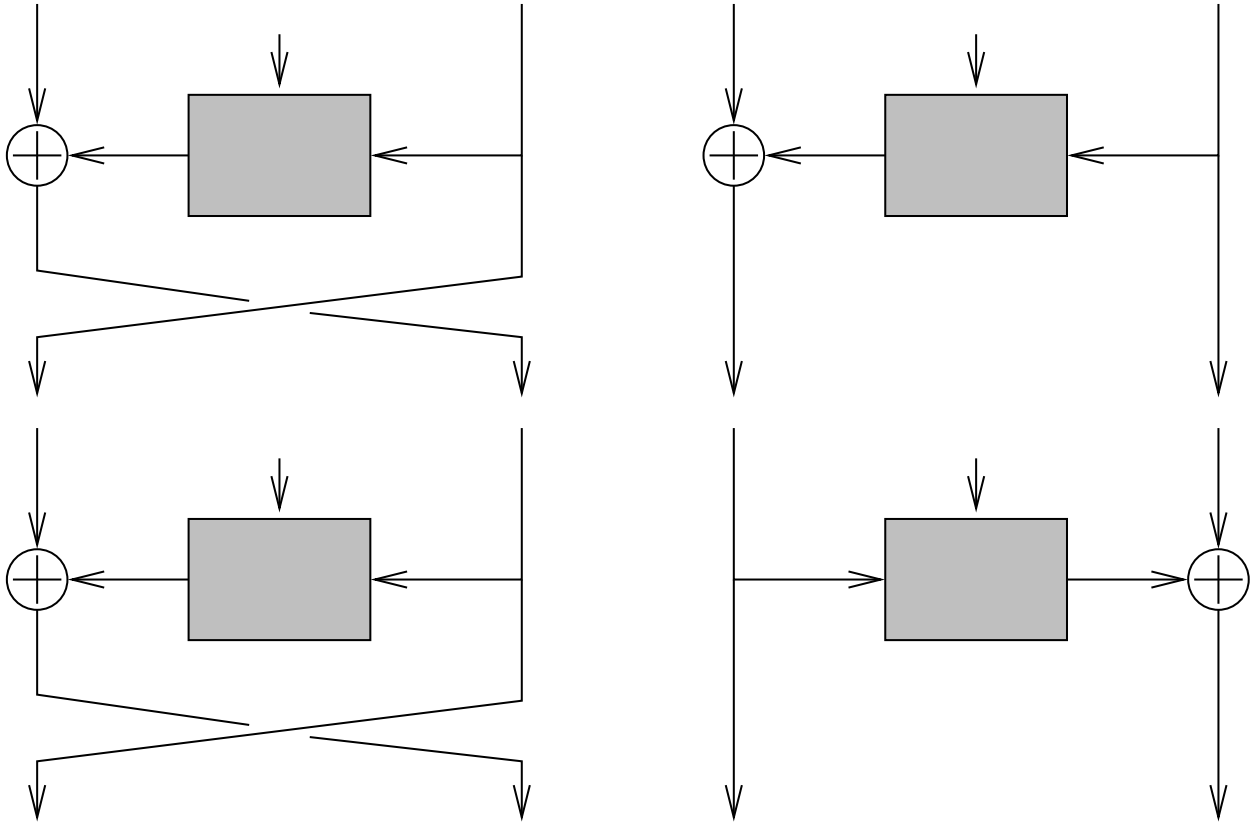
$$\mathbf{K}[1], \ldots, \mathbf{K}[16] \in \{\mathbf{0},\mathbf{1}\}^{\mathbf{48}},$$

abgeleitet aus einem 56-bit Chiffrierschlüssel.

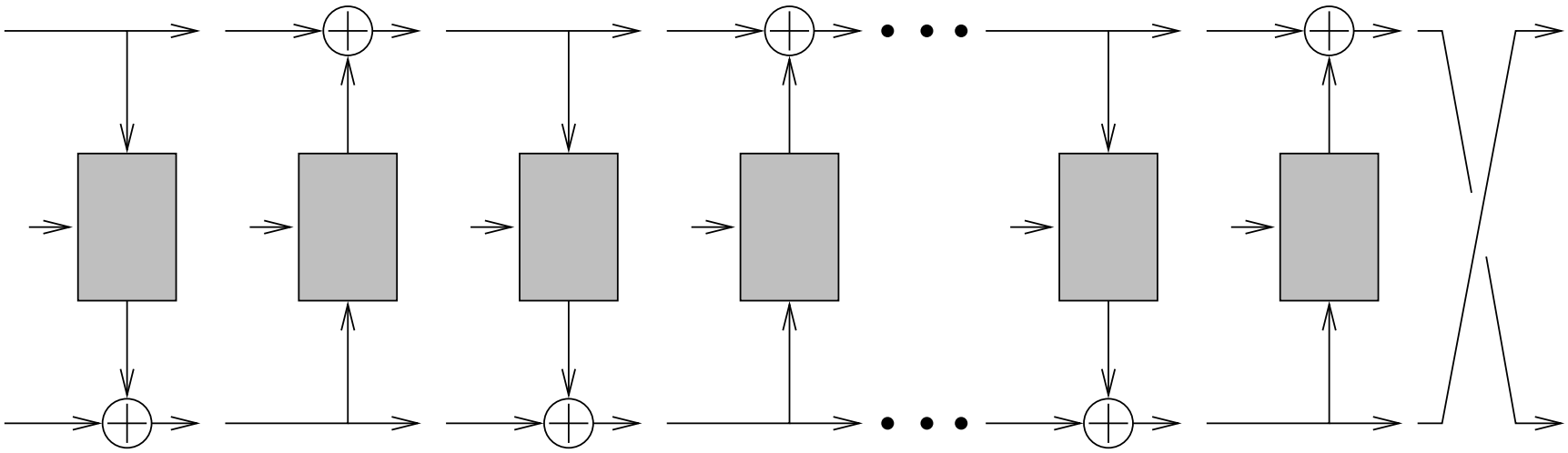


Diese “Feistel-Chiffre” ist die Verallgemeinerung der abstrakten Blockchiffren P_2, P_3 und P_4.

Feistel-Netzwerk: Verschiedene Darstellungsweisen



DES: Insgesamt 16 Runden



Zusätzlich zur Rundenfunktion

- Anwendung einer schlüssel<u>un</u>abhängigen "Initial Permutation"

$$\mathsf{IP} : \{1, \ldots, 64\} \to \{1, \ldots, 64\}$$

  am Anfang. Anwendung von IP^{-1} am Ende.

$$DES_{\mathbf{K}}(M) := \mathsf{IP}^{-1}(f_{\mathbf{K}[16]}(\cdots(f_{\mathbf{K}[1]}(\mathsf{IP}(M))))).$$

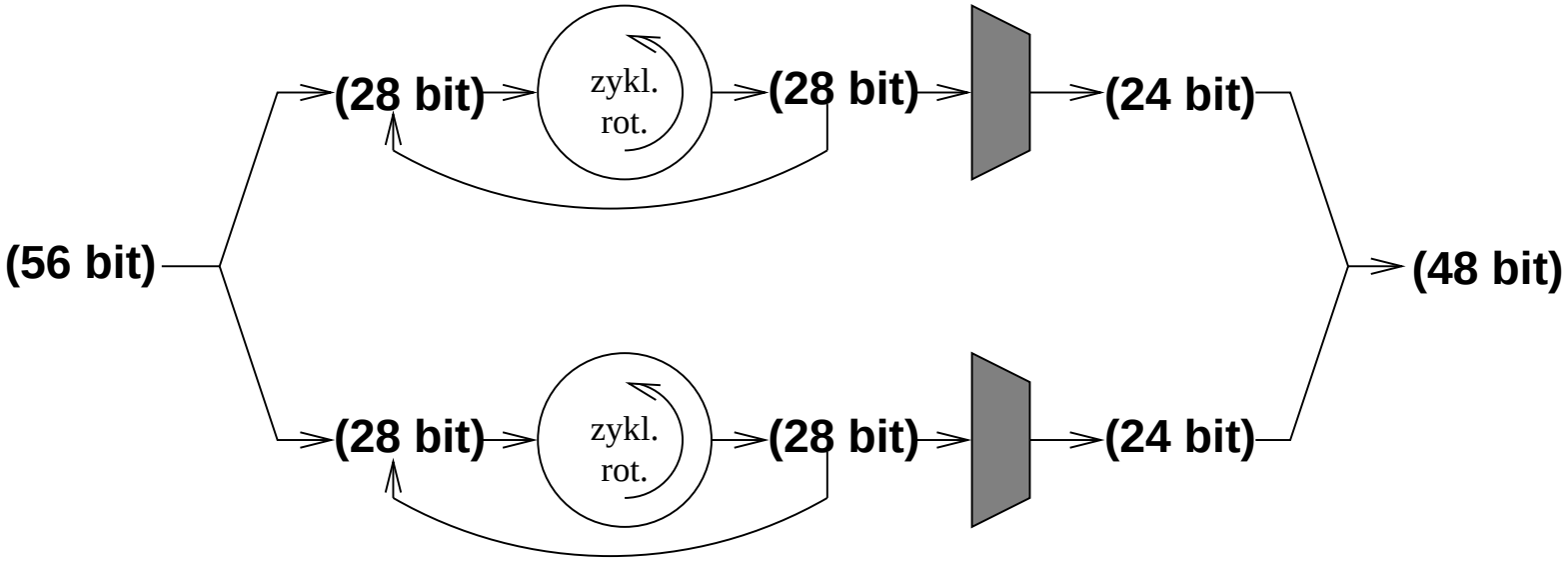
- In Hardware ist das praktisch "kostenlos", in Software typischerweise etliche Rechenschritte bzw. Takte.
- Der Sinn von IP und IP^{-1} ist unklar. Für die Sicherheit des DES sind beide irrelevant. (**Warum?**)
- Wir können IP/IP^{-1} ignorieren.

Wie entschlüsselt man?

($\rightarrow$ Tafel)

Der DES Key-Schedule

Der Key-Schedule nimmt 56 Schlüsselbits als Eingabe und produziert 16 Rundenschlüssel zu jeweils 48 Bits.



Der DES Key-Schedule (2)

- **NULL**, **EINS** $\in \{0,1\}^{28}$ bezeichnen die Konstanten $\mathbf{0...000}$ und $\mathbf{1...111}$.
- Ist eine Hälfte von $\mathbf{K}$ entweder gleich **NULL** oder gleich **EINS**, dann verändert sie sich im Verlauf des Key-Schedules nicht.
- Sinde beide Hälften gleich **NULL** oder gleich **EINS**, d.h. $\mathbf{K} \in \{(\textbf{NULL}, \textbf{NULL}), (\textbf{NULL}, \textbf{EINS}), (\textbf{EINS}, \textbf{NULL}), (\textbf{EINS}, \textbf{EINS})\}$, dann gilt:

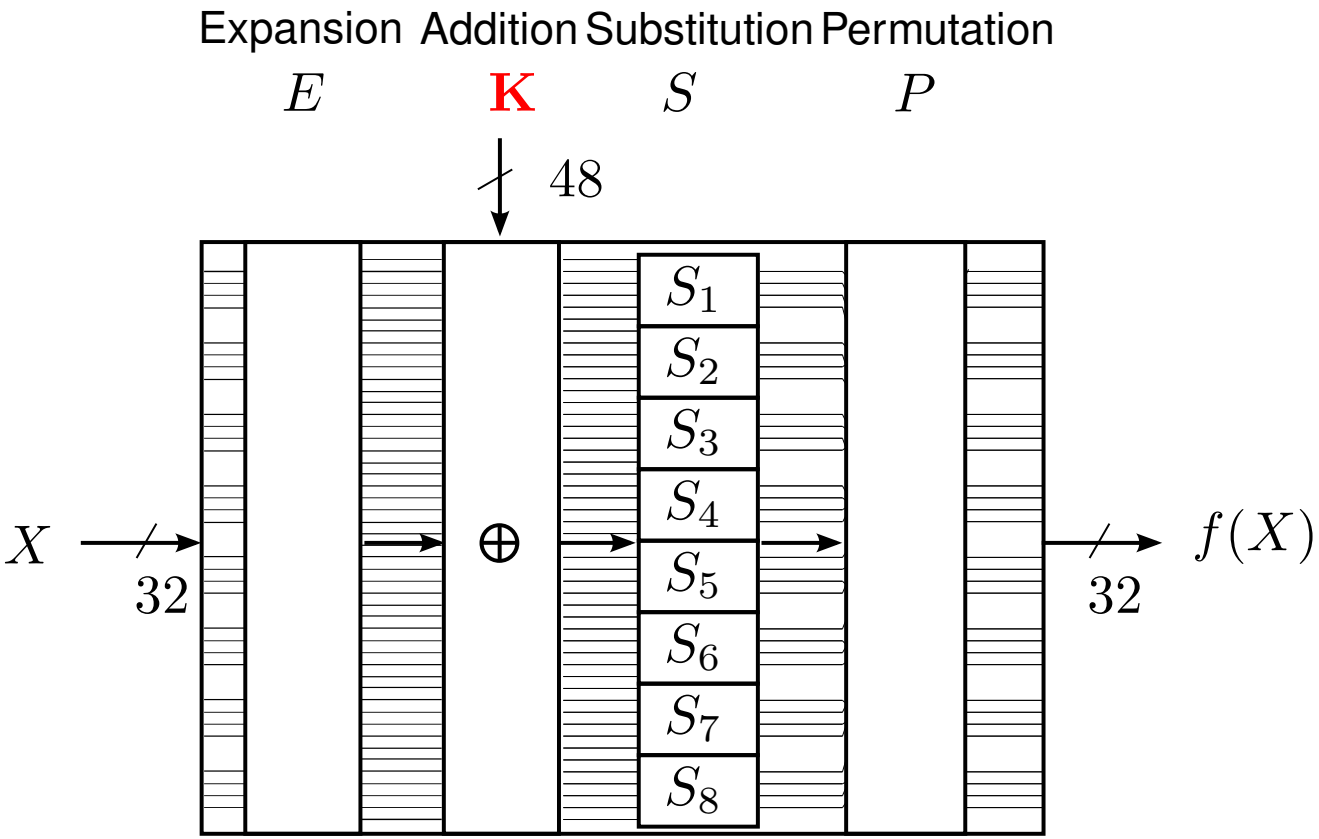
$$\mathbf{K}[1] = \mathbf{K}[2] = \cdots = \mathbf{K}[16].$$

Der DES Key-Schedule (3)

- Für diese vier Schlüssel $\mathbf{K}$ gilt: $E_{\mathbf{K}} = D_{\mathbf{K}}$.
- Derartige Schlüssel bezeichnet man als <u>schwach</u>.
- Man kennt keine weiteren schwachen Schlüssel.
- Außerdem kennt man 6 Paare <u>semi-schwacher</u> Schlüssel. Dies sind Paare $(\mathbf{K}, \mathbf{L})$ mit $E_{\mathbf{K}} = D_{\mathbf{L}}$.

Die f-Funktion des DES

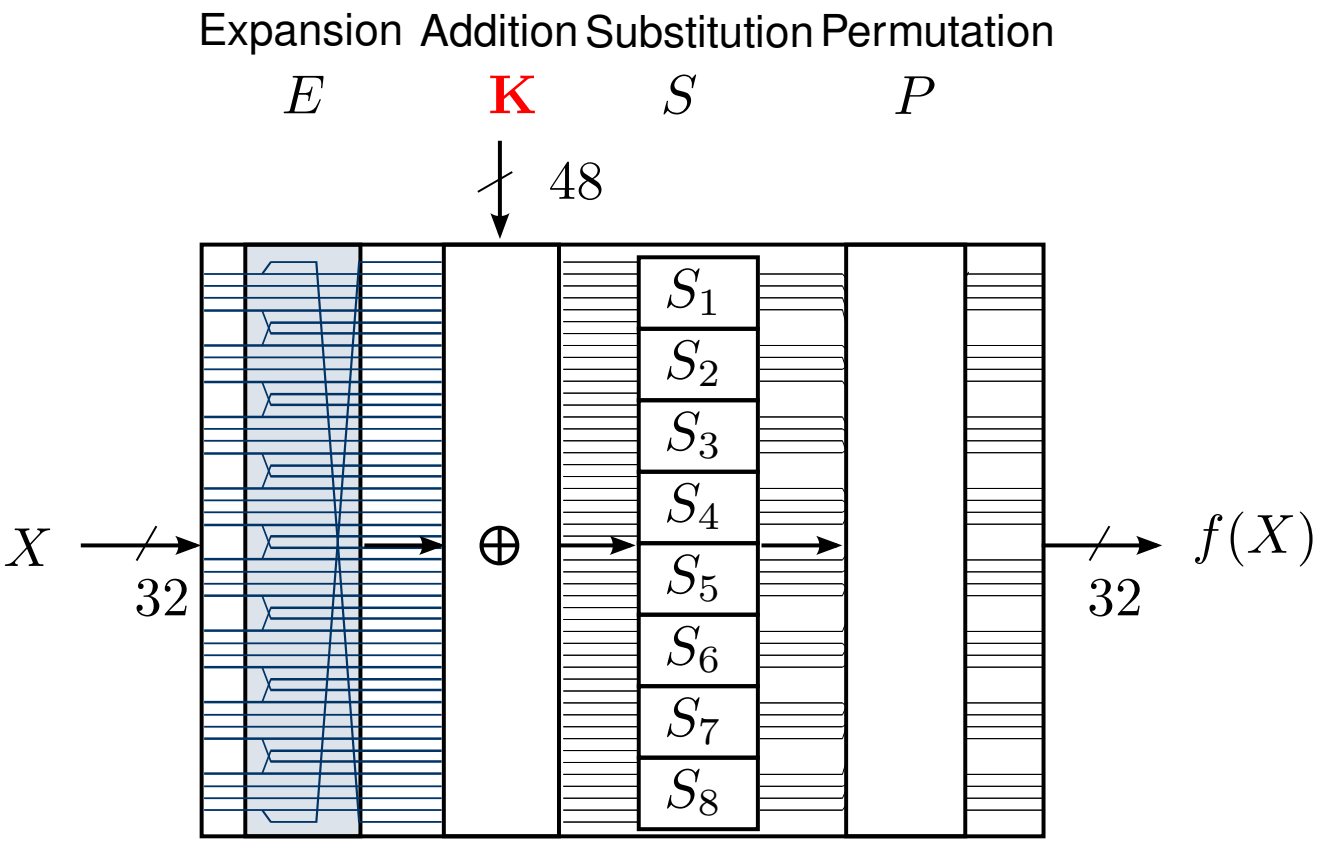
Die f-Funktion im Detail



$$f_{\mathbf{K}[i]}(X) := P(S(E(X) \oplus \mathbf{K}[i])).$$

Die Expansionsfunktion (E)

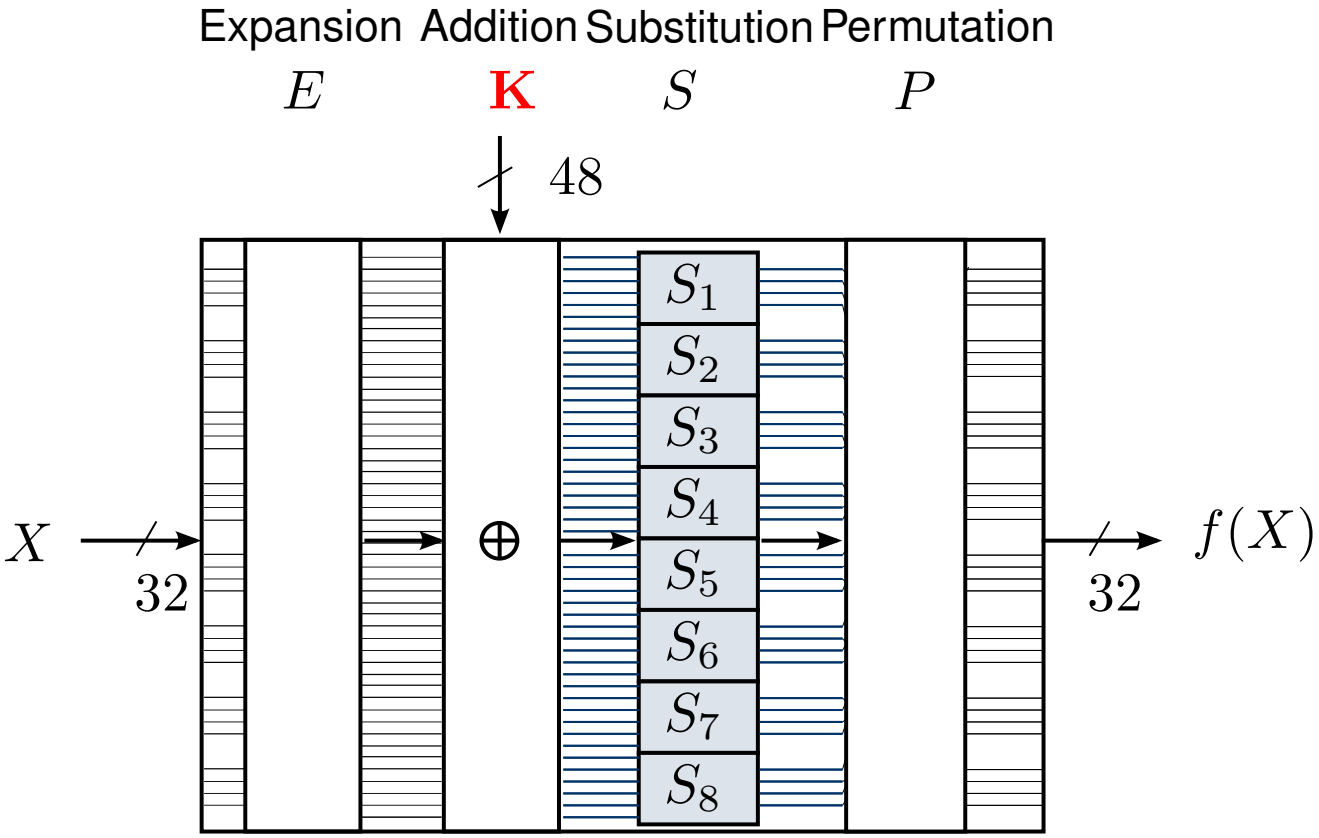
Die Expansionsfunktion $E : \{0,1\}^{32} \to \{0,1\}^{48}$ expandiert 32 zu 48 Bits.



Die Substitution (S)

Die acht Substitutionsboxen (S-Box $S_1, \ldots, S_8$) ersetzen jeweils sechs Eingabe- durch vier Ausgabebits:

$$S(X) : (\{0,1\}^6)^8 \rightarrow (\{0,1\}^4)^8$$



Die Substitution: S-Box 1 (S_1)

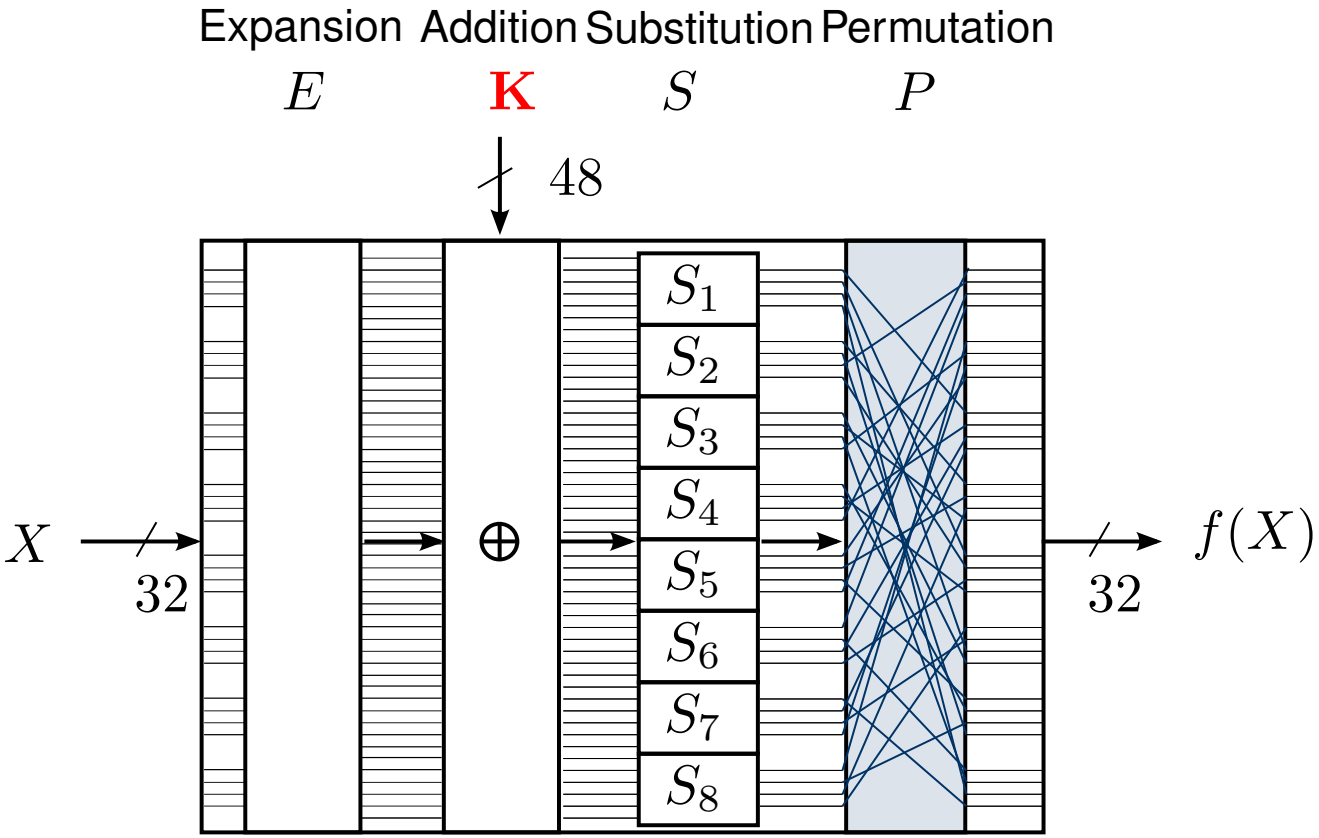
	Mittlere vier Bits der Eingabe															
S_1	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0 0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0 1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1 0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1 1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	3

Beispiele

$$S_1(1) = S_1(\mathbf{000001}) = (0)_{10} = (0000)_2$$
$$S_1(20) = S_1(\mathbf{010100}) = (6)_{10} = (0110)_2$$
$$S_1(56) = S_1(\mathbf{111000}) = (3)_{10} = (0011)_2$$
$$S_1(57) = S_1(\mathbf{111001}) = (10)_{10} = (1010)_2$$

Die Permutation (P)

P-Permutation: 32 bit $\rightarrow$ 32 bit



Linearität

Linearität

Wir nennen eine Funktion $F : \mathcal{X} \to \mathcal{Y}$ **linear** (bzgl. einer Operation $\circ$) wenn für alle Eingabepaare $X, X' \in \mathcal{X}$ gilt:

$$F(X) \circ F(X') = F(X \circ X') \circ F(0).$$

- Wir beziehen uns hier auf Linearität bzgl. XOR:

$$F(X) \oplus F(X') = F(X \oplus X') \oplus F(0).$$

Linearität der DES-Operationen

Bis auf die S-Boxen sind alle Operationen der DES-Rundenfunktion **linear** (bzgl. XOR):

- Unäre Operationen (E, P):

$$E(X_1) \oplus E(X_2) = E(X_1 \oplus X_2) \oplus E(0^{32})$$
$$P(X_1) \oplus P(X_2) = P(X_1 \oplus X_2) \oplus P(0^{32})$$

- Addition des Rundenschlüssels $KA_{\mathbf{K}[i]}(X) = X \oplus \mathbf{K}[i]$:

$$(X_1 \oplus \cancel{\mathbf{K}[i]}) \oplus (X_2 \oplus \cancel{\mathbf{K}[i]}) = X_1 \oplus X_2$$

Das heißt, sei $f \in \{E, P, KA\}$:

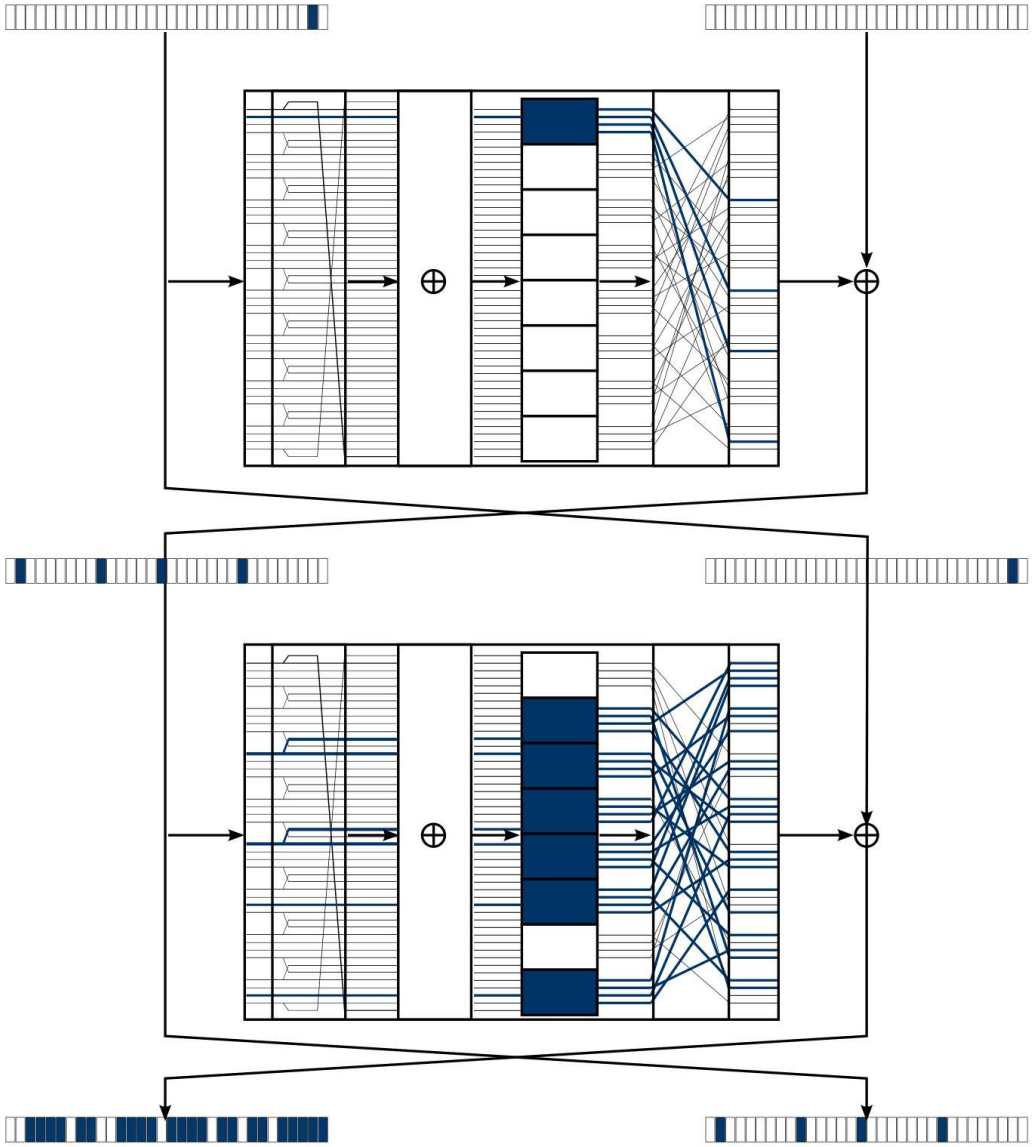
- Sind X_1, $f(X_1)$ und die Änderung $X_2 \oplus X_1$ bekannt, so kann man $f(X_2)$ **ohne Kenntnis des Schlüssels** einfach berechnen.
- $\rightarrow$ eine Chiffre benötigt nicht-lineare Operation(en)!

Lawineneffekt (Diffusion)

- Jedes Bit der Ausgabe hängt von jedem Bit der Eingabe ab.
- *Kleine* Änderungen in der Eingabe verursachen *große* Änderung in der Ausgabe.
- Webster und Tavers haben 1985 das strenge Lawinenkriterium formuliert.

  Flippen eines Eingabebits $\rightarrow$ *jedes Ausgabebit ändert sich mit einer Wahrscheinlichkeit von* 50%

Beispiel Lawineneffekt: 2 Runden DES



- Beim DES verursacht die Kombination aus S-Boxen und der Permutation P den Lawineneffekt.
- Blau = Von der Differenz beeinflusste Bits

Komplementäreigenschaft

Sei $\overline{X}$ das Inverse des Bit-Strings X.

Theorem 18 (Komplementäreigenschaft)

Für alle Schlüssel $\mathbf{K}$ *und alle Klartexte* M *gilt*

$$\overline{DES_{\mathbf{K}}(M)} = DES_{\overline{\mathbf{K}}}(\overline{M}).$$

Angriff auf den vollen DES (16 Runden)

- **1991 Biham und Shamir:** 2^{49} Chosen-Plaintext-Anfragen
- **1994 Matsui:** 2^{43} Chosen-Plaintext-Anfragen
- **1997 Biham und Biryukov:** 2^{50} Known-Plaintext-Anfragen.

Varianten des DES mit weniger Runden oder veränderten S-Boxen sind erheblich verwundbarer!

Angriffe über die Schlüssellänge

Da DES-Schlüssel aus nur 56 Bits bestehen, sind Brute-Force-Angriffe mit Rechenzeit $N = O(2^{56})$ durchaus praktikabel:

Vollständige Suche Known Plain- and Ciphertext

Zeit $O(N)$, Platz $O(1)$

Tabellensuche Chosen Plaintext, Known Plaintext

Vorbereitungszeit $O(N)$, Platz $O(N)$,
Ausführungszeit $O(1)$

Time-Memory-Tradeoff (Hellman, 1980)

Chosen Plaintext, prinzipiell Known Plaintext

Vorbereitungszeit $O(N)$, Platz: $O(N^{2/3})$,
Ausführungszeit $O(N^{2/3})$

Geschichte:

1980 Hellman time-memory-tradeoff
(Spezialrechner + Massenspeicher):
4 Mio. $, 2 Jahre Vorbereitungszeit, 100 Schlüssel/Tag.

1993 Wiener (Spezialrechner):
1 Mio. $, 7 Schlüssel/Tag.

1997 Erste DES-Challenge
(Internet und *idle time* tausender Rechner):
keine Kosten, 4 Monate/Schlüssel.

1998 DES-Cracker der EFF (Spezialrechner):
0.25 Mio. $, einige Tage/Schlüssel.

Vergleich: 1 Spionagesatellit 3 000 Mio. $ bis 6 000 Mio. $ (geschätzt).

Effektive Schlüssellänge

- Eine Chiffre hat die **effektive Schlüssellänge** ℓ bit, wenn es keinen Angriff gibt, der im Durchschnitt schneller ist als $2^{\ell-1}$ Verschlüsselungsoperationen. (Maßstab: Brute Force.)
- Andere Ressourcen, insbesondere Speicherplatz und Klar-/Chiffretextpaare, können ebenfalls im Umfang bis zu $2^{\ell-1}$ Einheiten beansprucht werden.
- Für praktikable Chiffren kennt man die effektive Schlüssellänge nicht, nur obere Schranken ($\rightarrow$ Angriffe).

Folgerungen für den DES

- Der beste bekannte analytische Angriff (mittels linearer Kryptanalyse) braucht etwa 2^{43} bekannte Klar-Chiffretext-Paare.

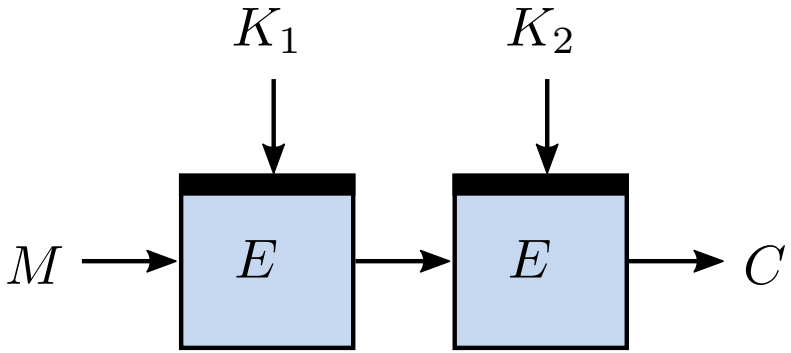
$\Rightarrow$ Effektive Schlüssellänge $\leq$ 42 bit.

- Alle bekannten analytischen Angriffe sind kaum praktikabel. Brute-Force-Angriffe sind praktikabel.

$\Rightarrow$ Der DES ist bemerkenswert stark gegen analytische Methoden, aber die Schlüssel sind zu klein.

Double-DES

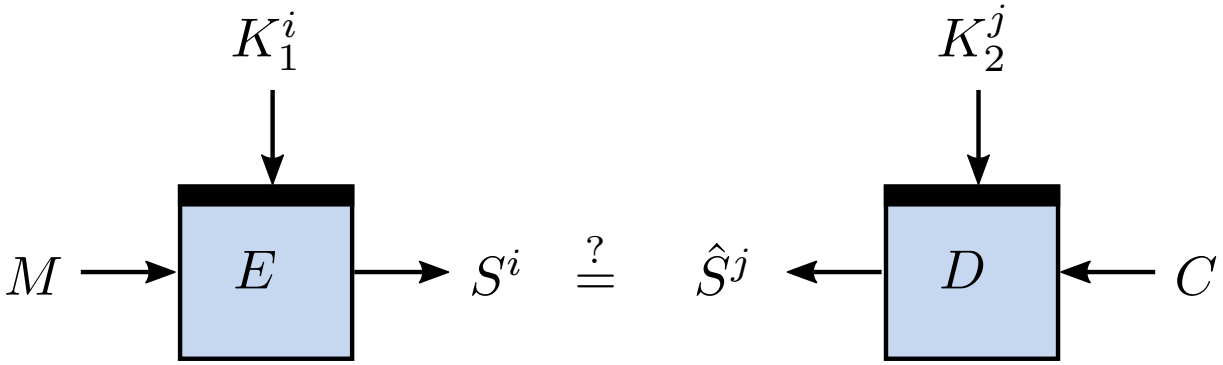
$$C = 2\mathrm{DES}_{\mathbf{K}_1,\mathbf{K}_2}(M) = \mathrm{DES}_{\mathbf{K}_2}\left(\mathrm{DES}_{\mathbf{K}_1}(M)\right)$$



Idee: Doppelte Anwendung von DES mit zwei unabhängigen Schlüsseln entspricht einem doppelt so großen Schlüssel, also 112 bit.
Stimmt das? ($\rightarrow$ Tafel)

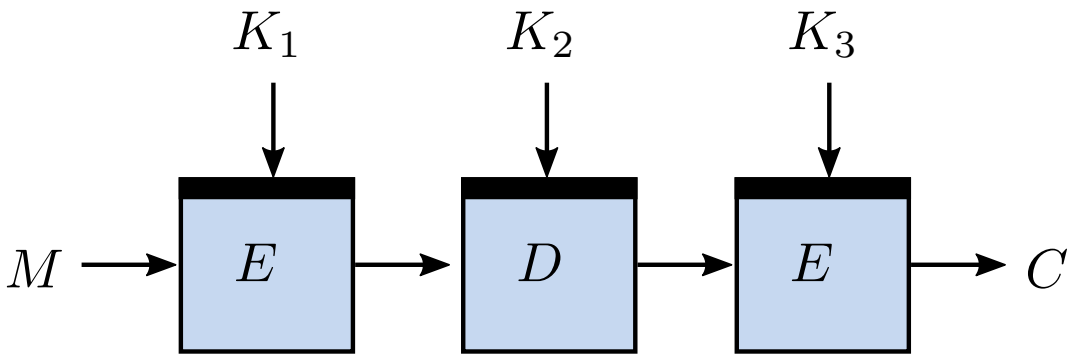
Meet-in-the-Middle-Angriff

$$C = 2\mathrm{DES}_{\mathbf{K}_1,\mathbf{K}_2}(M) = \mathrm{DES}_{\mathbf{K}_2}\left(\mathrm{DES}_{\mathbf{K}_1}\left(M\right)\right)$$



Triple-DES

$$3\mathrm{DES}_{\mathbf{K}_1,\mathbf{K}_2,\mathbf{K}_3}(M) := \mathrm{DES}_{\mathbf{K}_3}\left(\mathrm{DES}^{-1}_{\mathbf{K}_2}\left(\mathrm{DES}_{\mathbf{K}_1}(M)\right)\right)$$



Üblich: Statt der zweiten DES-Verschlüsselungsoperation eine DES-Entschlüsselungsoperation (“EDE”-Modus).

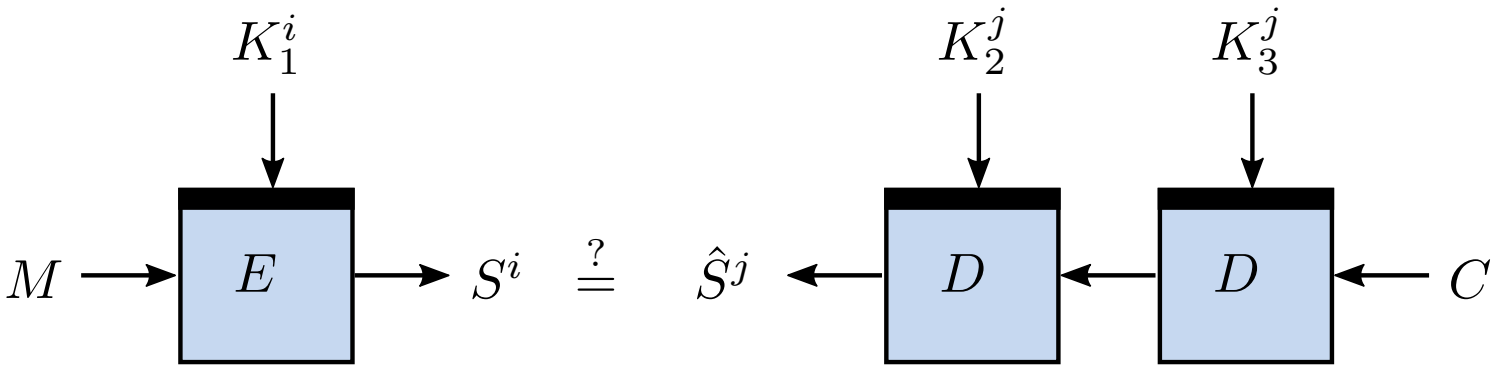
Angriffe auf Triple-DES

Variante	Angriff	# Paare	Rechenaufwand
Three-Key	MITM	3	2^{112}
Two-Key ($\mathbf{K}_1 = \mathbf{K}_3$)	[1]	2^{56}	2^{56}
Three-Key	[2]	2^{45}	2^{108}

[1] Merkle, Hellman (C. ACM, 1981).
[2] Lucks (FSE 1998).

Meet-in-the-Middle-Angriff auf 3DES

$$3\mathrm{DES}_{\mathbf{K}_1,\mathbf{K}_2,\mathbf{K}_3}(M) := \mathrm{DES}_{\mathbf{K}_3}\left(\mathrm{DES}^{-1}_{\mathbf{K}_2}\left(\mathrm{DES}_{\mathbf{K}_1}(M)\right)\right)$$



DES: Zusammenfassung, Bemerkungen

- 64-Bit-Blockchiffre mit 56-Bit-Schlüssel
- bekannte und intensiv analysierte Blockchiffre
- massive Kritik an kurzen Schlüsseln (Abhilfe: Triple DES)
- Triple-DES wird noch lange Zeit weiter genutzt werden (trotz des “DES-Nachfolgers” AES)

Sie sollten insbesondere

- wissen, wie der DES funktioniert,
- und die Sicherheit von mehrfacher (doppelter, dreifacher, . . .) Verschlüsselung abschätzen können (mit Begründung).