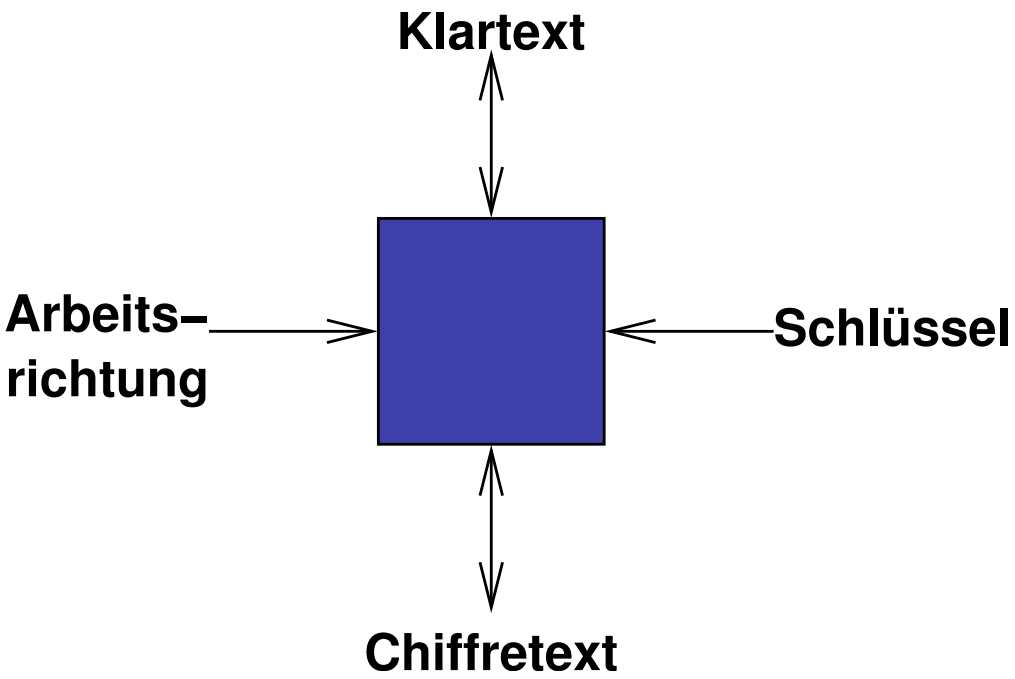
Data Encryption Standard

Prof. Rüdiger Weis

TFH Berlin

Sommersemester 2008

Blockchiffren



Wichtige Parameter: Blockgröße, Schlüssellänge

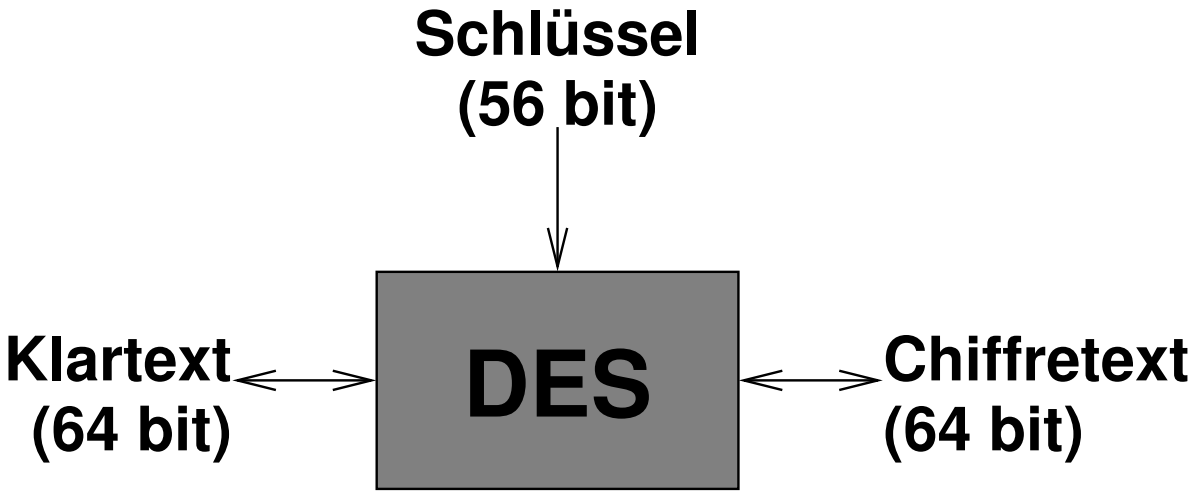
Blockchiffren

Für jeden Schlüssel K ist eine Blockchiffre E mit der Blockgröße b eine Permutation $E_k : \{0,1\}^b \rightarrow \{0,1\}^b$.

Sicherheitskriterium:
Sicherer Pseudozufallspermutationsgenerator!
Die Blockchiffre soll für jemanden der den Schlüssel nicht kennt (den „Angreifer“) möglichst „zufällig“ erscheinen:

- Der Angreifer kann Klartexte und Chiffretexte frei wählen und die zugehörige Verschlüsselung bzw. Entschlüsselung erfragen (zweiseitiger Angriff).
- Der Angreifer versucht, E_k von einer zufälligen Permutation $\{0,1\}^b \rightarrow \{0,1\}^b$ zu unterscheiden.

Der DES



Geschichte des DES

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

64-bit Blockchiffre mit 56-bit Schlüsseln.

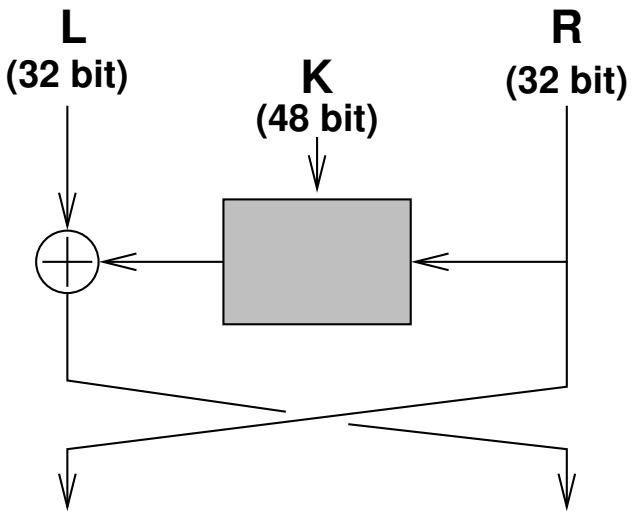
Ab 1977 Kritik an Schlüssellänge.
Trotzdem große Akzeptanz und riesige Verbreitung.

Ab 1990 Differentielle und lineare Kryptanalyse.

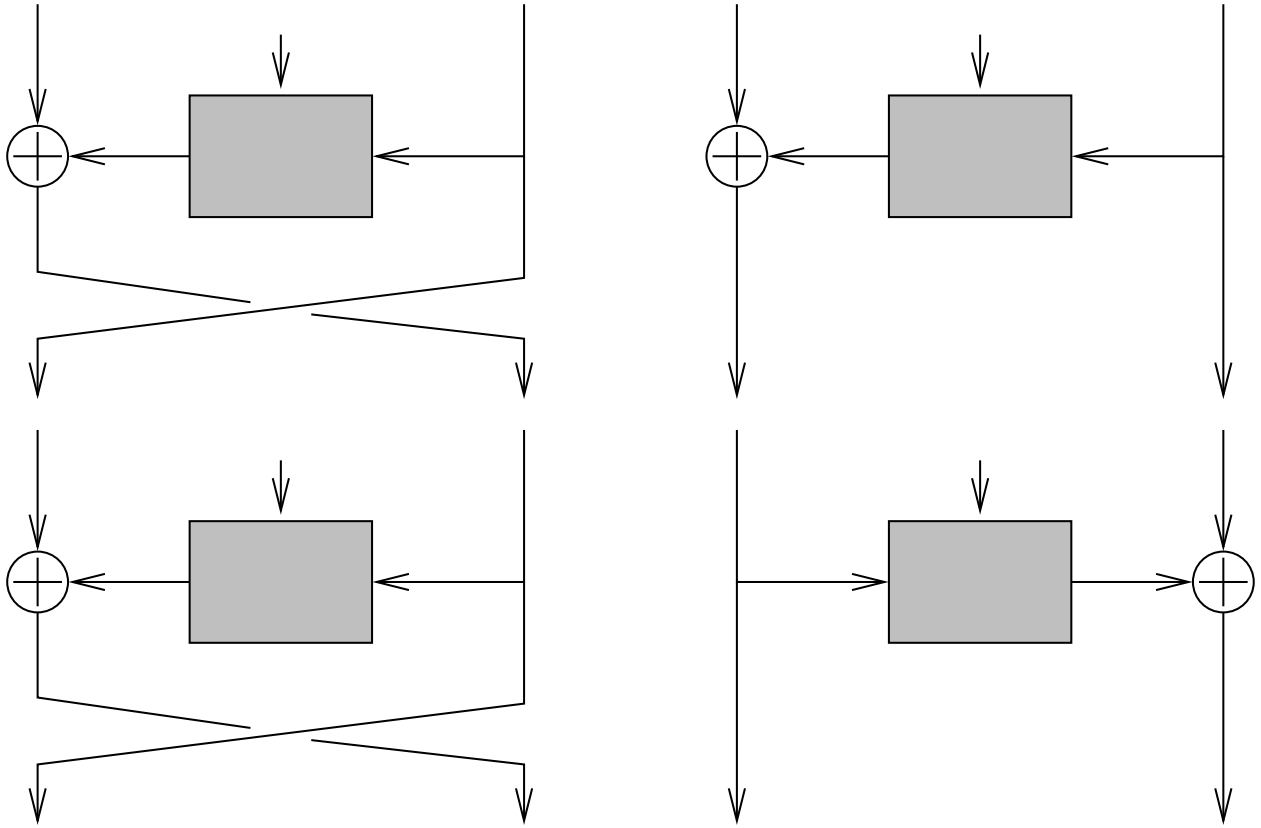
1997 DES-Challenge (1000e von Rechnern, 4 Mon.).

Struktur des DES („Feistel-Netzwerk“)

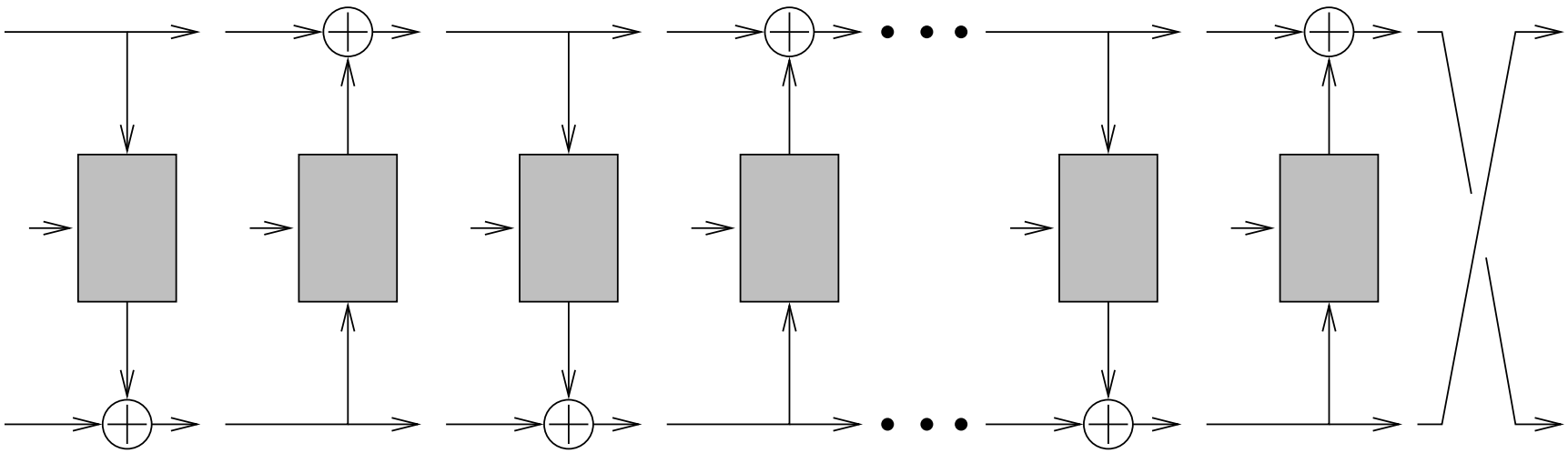
- Rundenfunktion
$$f_{K[i]} : \{0,1\}^{32} \to \{0,1\}^{32}$$
- 16 Runden
- 16 Rundenschlüssel
$$K[1] \ldots, K[16] \in \{0,1\}^{48},$$
abgeleitet aus einem 56-bit Chiffrierschlüssel.



Zwei verschiedene Darstellungweisen



DES: Insgesamt 16 Runden



Zusätzlich zur Rundenfunktion

Anwendung einer schlüsselunabhängigen "Initial Permutation" IP: $\{1,\ldots,64\} \to \{1,\ldots,64\}$ ("wire crossing") am Anfang.
Anwendung von IP^{-1} am Ende.

Beide stellen einfache Bit-Vertauschungen dar. 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?**)

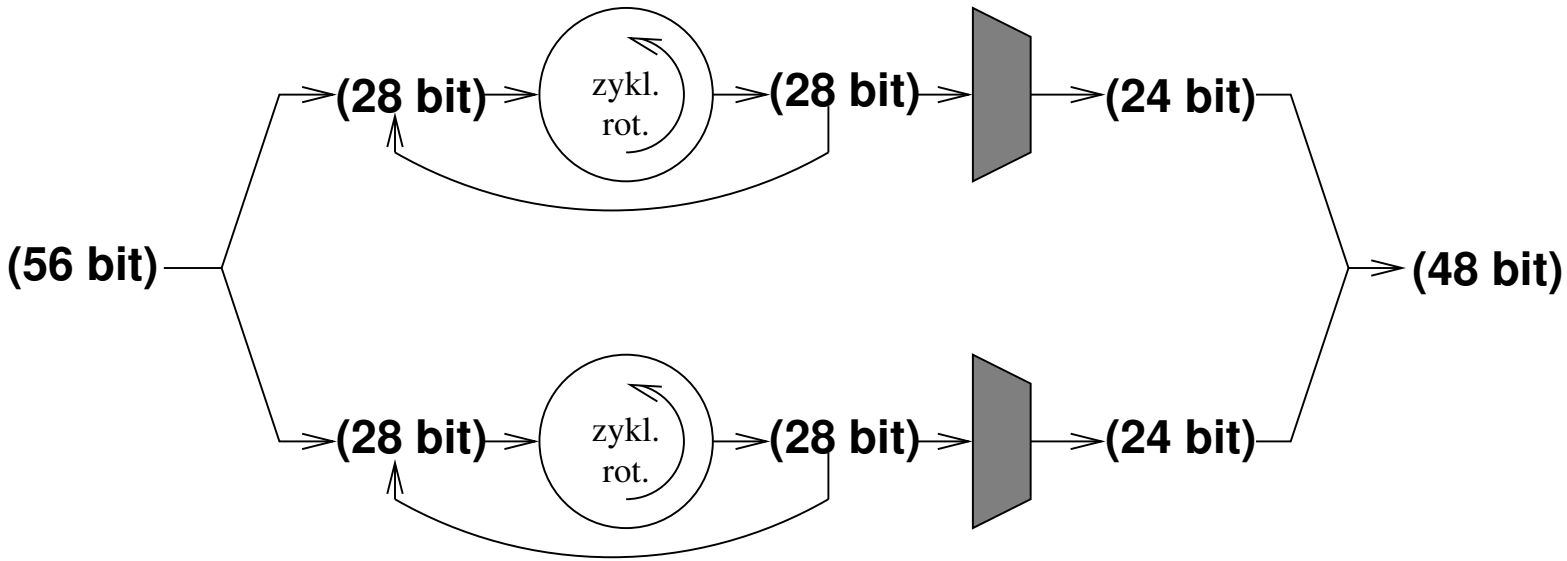
Im Folgenden werden diese Transformationen ignoriert.

Wie Entschlüsselt man?

Geg. einen Chiffretext Y, können die linke und die rechte Hälfte von Y tauschen und die Rundenfunktion unter den Teilschlüsseln $k[16], k[15], \ldots, k[1]$ anwenden anwenden.
Ver- und Entschlüsseln arbeiten also genau analog, nur dass sich die Reihenfolge der Teilschlüssel ändert.

Der DES Key-Schedule

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



Der DES Key-Schedule (2)

Es bezeichnen $\text{NULL}, \text{EINS} \in \{0,1\}^{28}$ die Konstanten 0...000 und 1...111.

Ist eine Hälfte von k entweder gleich Null oder gleich Eins, dann verändert sie sich im Verlauf des Key-Schedules nicht.

Ist der Schlüssel k in

$$\{(\text{NULL}, \text{NULL}), (\text{NULL}, \text{EINS}), (\text{EINS}, \text{NULL}), (\text{EINS}, \text{EINS})\},$$

dann ist

$$k = k^*[1] = k^*[2] = \cdots = k^*[16],$$

also

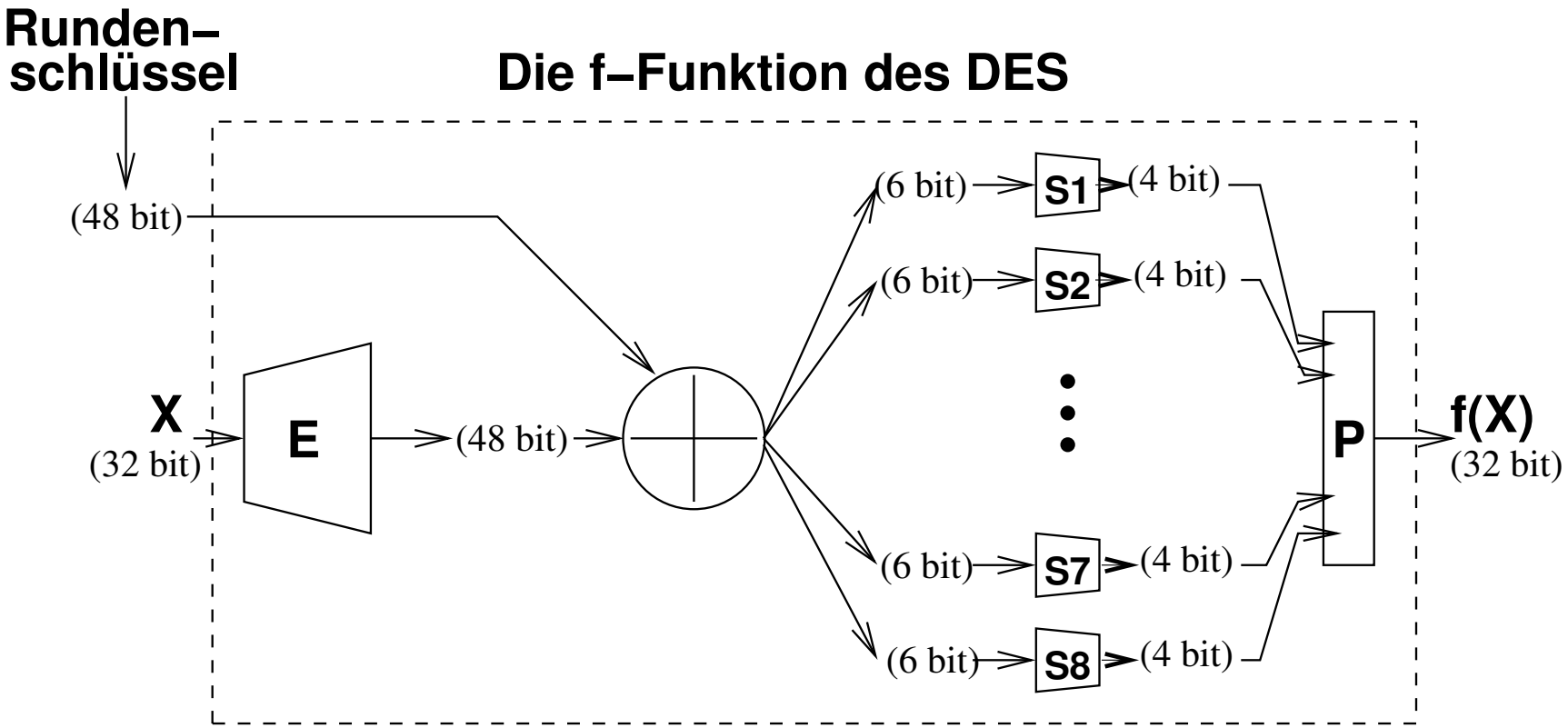
$$k[1] = k[2] = \cdots = k[16].$$

Der DES Key-Schedule (3)

- Für diese vier Schlüssel k gilt: $E_k = D_k$.
- Derartige Schlüssel bezeichnet man als schwach.
- Man kennt keine weiteren schwachen Schlüssel.
- Außerdem kennt man 6 Paare semi-schwacher Schlüssel. Dies sind Paare (k, l) mit $E_k = D_l$.

Die f-Funktion des DES

Die f-Funktion im Detail

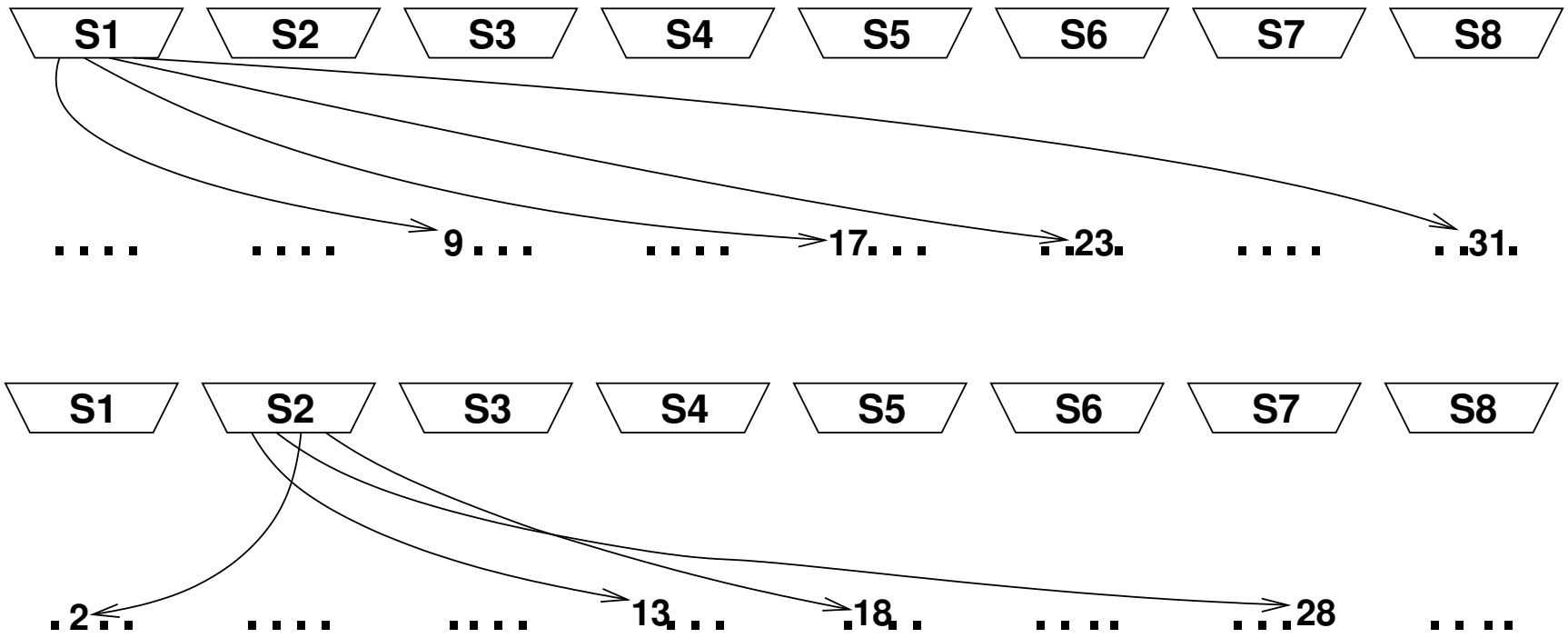


Die E-Expansion

32 Eingabebits werden auf 48 Ausgabebits abgebildet.

Die P-Permutation

Die $8 * 4 = 32$ Ausgabebits der DES S-Boxen werden auf 32 Ausgabebits abgebildet. Z.B. für **S1** und **S2**:



Complementation Property

Es bez. $\overline{s}$ das Inverse des Bit-Strings s.

Theorem (Complementation Property)

Für alle Schlüssel k und alle Klartexte x gilt

$$\overline{DES_k(x)} = DES_{\overline{k}}(\overline{x}).$$

Der „Lawineneffekt“

Für je zwei Klartexte X_1 und X_2 können wir die Differenz $\Delta X = X_1 \oplus X_2$ angeben.

Man kann leicht Klartextdifferenzen angeben, die in der ersten bzw. zweiten Runde nur den Input einer S-box verändern.

Das Design der S-Boxen und – vor allem – die Permutation P stellen sicher, daß Differenzen sich danach schnell auf die Inputs für die anderen S-Boxen ausbreiten ($\rightarrow$ „Lawineneffekt“).

Ein entsprechender Lawineneffekt tritt auch bei einer Änderung des Schlüssels ein.

Linearität

Es gibt die folgenden **linearen** Operationen:

- Unäre Operationen (E, P, IP):

$$E(X_1) \oplus E(X_2) = E(X_1 \oplus X_2)$$

  (entspr. für P und IP).
- Einmischen des Rundenschlüssels

$$(X_1 \oplus k[i]) \oplus (X_2 \oplus k[i]) = X_1 \oplus X_2 = \Delta X$$

Bis auf die S-Boxen sind alle Operationen linear. Sind die S-Boxen auch linear?

Nein – das wäre auch furchtbar!

Angriffe auf den DES

Die wichtigten Angriffe auf den DES:

- Differentielle Kryptanalyse
- Lineare Kryptanalyse
- Angriffe, die die kurze Schlüssellänge ausnutzen

Differentielle Kryptanalyse

Ansatz: Trotz der Nichtlinearität der S-Boxen, Aussagen über Eingabe-Ausgabedifferenzen

Beispiel: S1: $\{0,1\}^6 \rightarrow \{0,1\}^4$.

Für jeden Wert $\Delta x \in \{0,1\}^6$ gibt es 64 Paare $x, x' \in \{0,1\}^6$ mit $x \oplus x' = \Delta x$.

„Idealfall“: Für jedes $\Delta y \in \{0,1\}^4$ gibt es genau 4 derartige Paare x, x' mit $\mathrm{S1}(x) \oplus \mathrm{S1}(x') = \Delta y$.

Differentielle Kryptanalyse (2)

Sei $\Delta x = (110100)$. Wir erhalten die folgende Tabelle:

Δy	0000	0001	0010	0011	0100	0101	0110	0111
Anzahl	0	8	16	6	2	0	0	12
Δy	1000	1001	1010	1011	1100	1101	1110	1111
Anzahl	6	0	0	0	0	8	0	6

Differentielle Kryptanalyse (3)

Entsprechend kann man mit den 32-bit Input-Differenzen und Output-Differenzen ΔX und ΔY für die ganze f-Funktion umgehen.

Idee: Sei die Differenz zweier Klartexte bekannt.

$\longrightarrow$ Input-Differenz für die erste f-Funktion

$\longrightarrow$ wahrscheinlichste Output-Differenz der ersten f-Funktion

$\longrightarrow$ wahrscheinlichste Input-Differenz für die zweite f-Funktion

$\longrightarrow$ ...

Differentielle Kryptanalyse (4)

Eine r-Runden „Charakteristik“ beschreibt die auftretenden Differenzen von Runde zu Runde. Die Klartext-Differenzen sind bekannt (bzw. sogar von Angreifer gewählt).

Eine Charakteristik besitzt eine bestimmte Wahrscheinlichkeit. Bei deren Berechnung geht man vereinfachend davon aus, dass die Rundenschlüssel $k[i]$ zufällig und voneinander unabhängig sind.

Differentielle Kryptanalyse (5)

Bester bekannter differentieller Angriff auf DES (16 Runden):

- 2^{47} gewählte oder 2^{55} zufällige Klartexte
- vergleichsweise geringer Rechenaufwand
- 13-Runden Charakteristik

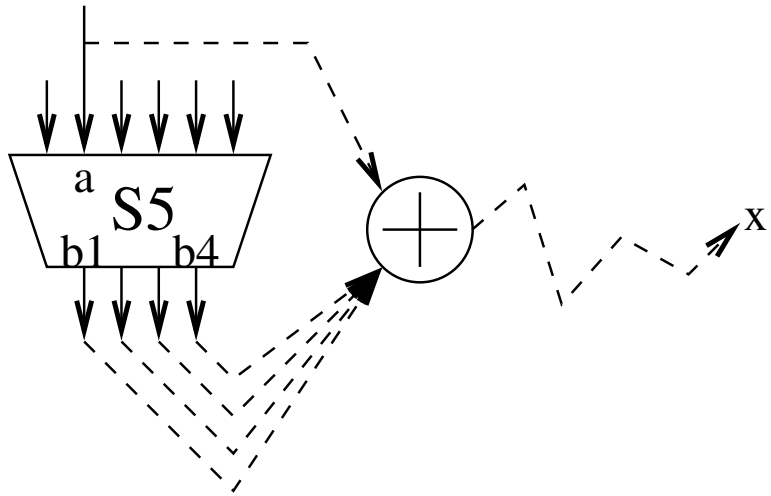
Varianten des DES mit weniger Runden oder veränderten S-Boxen sind erheblich verwundbarer!

DES wurde optimiert mit Blick auf differentielle Kryptanalyse.

Lineare Kryptanalyse

Statt der Differenz (dem XOR) von Klartextpaaren und Chiffretextpaaren kann man das XOR einzelner bits des Klar- und des Chiffretextes betrachten.

Beispiel S5:



Für die S-Box S5 gilt:

Für $x = a \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4$ ist $\mathrm{prob}[x = 0] = 20/64$.

Lineare Kryptanalyse (2)

Entsprechend betrachtet man die linearen Gleichungen bezüglich Ein- und Ausgabe der f-Funktion.

Der beste bekannte lineare Angriff auf 16 Runden des DES braucht etwa 2^{43} bekannte (nicht notw. gewählte) Klartexte.

DES wurde *nicht* optimiert gegen lineare Kryptanalyse.

Angriffe über die Schlüssellänge

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

Vollst. Suche known plaintext, known 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 Spionagesatelit $\approx$ 3 000 Mio. \$.

Effektive Schlüssellänge

Eine Chiffre hat die **effektive Schlüssellänge** L bit, wenn es keinen Angriff gibt, der im Durchschnitt schneller ist als 2^{L-1} Verschlüsselungsoperationen. (Maßstab: Brute Force.)

Andere Ressourcen, insbesondere Speicherplatz und Klar-/Chiffretextpaare, können ebenfalls im Umfang bis zu 2^{L-1} Einheiten beansprucht werden.

Für praktikable Chiffren kennt man die effektive Schlüssellänge nicht. (!)
Man kennt 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.

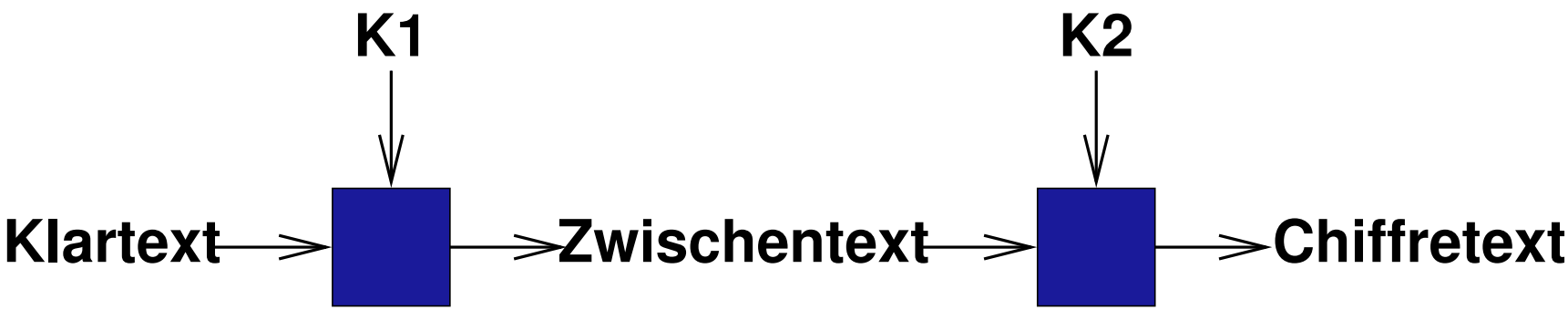
$\Rightarrow$ Effektive Schlüssellänge ≤ 44 bit.

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

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

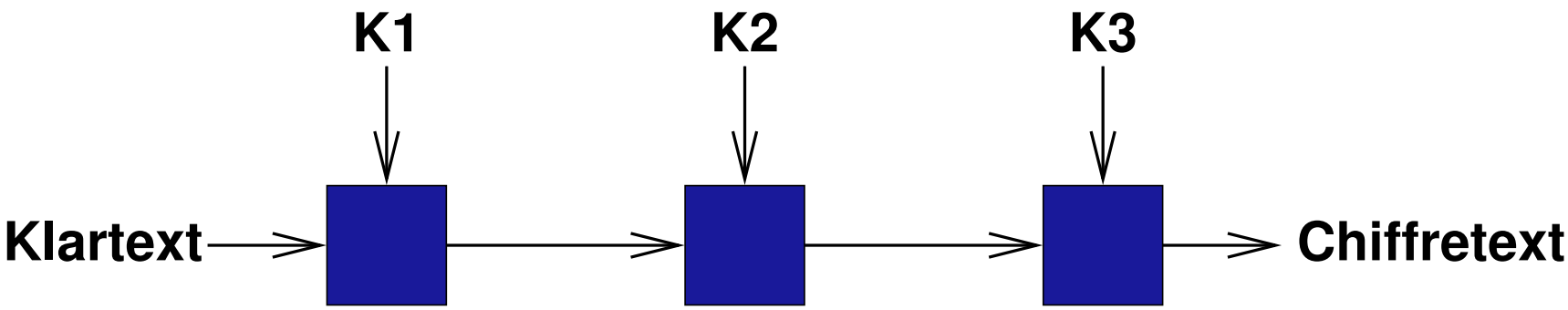
Double-DES

$$y = \mathrm{doubleDES}_{K1,K2}(x) = \mathrm{DES}_{K2}\left(\mathrm{DES}_{K1}\left(x\right)\right)$$



IDEE: Doppelte Anwendung von DES mit 2 unabhängigen Schlüsseln entspricht einem doppelt so großen Schlüssel, also 112 bit. **Stimmt das?** ($\rightarrow$ Tafel)

Triple DES



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 (K1=K3)	[1]	2^{56}	2^{56}
Three-Key	[2]	2^{45}	2^{108}

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

Der DES – Zusammenfassung

DES: 64-bit Blockchiffre

- Bekannteste und bestuntersuchte Blockchiffre
- massive Kritik an kurzen Schlüsseln
  Abhilfe: Triple DES
- Triple DES wird noch lange Zeit weiter genutzt werden (trotz des „DES-Nachfolgers“ AES → späteres Kapitel)

Danksagungen

Danksagungen

- Aus einer Vorlesung von Stefan Lucks
- Erstellt mit Freier Software