Der DES-Algorithmus

5

In Kap. 3 wurden Kryptosysteme eingeführt und einige historische symmetrische Verschlüsselungsverfahren beschrieben. Die Verfahren aus Kap. 3 konnten aber alle gebrochen werden, weil sie affin linear sind. Perfekt sichere Systeme wurden in Kap. 4 beschrieben. Sie stellten sich aber als ineffizient heraus. In diesem Kapitel wird das DES-Verfahren beschrieben. Das DES-Verfahren war viele Jahre lang der Verschlüsselungsstandard in den USA und wird weltweit eingesetzt. Das einfache DES-Verfahren gilt aber nicht mehr als sicher genug. Inzwischen wurde vom US-amerikanischen National Institute of Standards als Nachfolger von DES das Rijndael-Verschlüsselungsverfahren ausgewählt [1], das im nächsten Kapitel beschrieben wird. Als sicher gilt aber nach wie vor die Dreifachvariante Triple-DES (siehe Abschn. 3.9). Außerdem ist DES ein wichtiges Vorbild für neue symmetrische Verfahren.

5.1 Feistel-Chiffren

Der DES-Algorithmus ist eine sogenannte *Feistel-Chiffre*. Wir erläutern in diesem Abschnitt, wie Feistel-Chiffren funktionieren.

Benötigt wird eine Blockchiffre mit Alphabet $\{0, 1\}$. Die Verschlüsselungsfunktion zum Schlüssel K sei f_K. Daraus kann man folgendermaßen eine Feistel-Chiffre konstruieren. Die Feistel-Chiffre ist eine Blockchiffre mit Blocklänge $2t$, $t \in \mathbb{N}$ und Alphabet $\{0, 1\}$. Man legt einen Schlüsselraum $\mathcal{K}$ fest. Außerdem legt man eine *Rundenzahl* $r \geq 1$ fest und wählt eine Methode, die aus einem Schlüssel $k \in \mathcal{K}$ eine Folge $K_1, \ldots, K_r$ von Rundenschlüsseln konstruiert. Die Rundenschlüssel gehören zum Schlüsselram der zugrundeliegenden Blockchiffre.

Die Verschlüsselungsfunktion E_k der Feistel-Chiffre zum Schlüssel $k \in \mathcal{K}$ funktioniert so: Sei p ein Klartext der Länge $2t$. Den teilt man in zwei Hälften der Länge t auf. Man schreibt also $p = (L_0, R_0)$. Dabei ist L_0 die linke Hälfte des Klartextes, und R_0 ist seine rechte Hälfte. Danach konstruiert man eine Folge $((L_i, R_i))_{1 \leq i \leq r}$ nach folgender

J. Buchmann, *Einführung in die Kryptographie*, Springer-Lehrbuch,
DOI 10.1007/978-3-642-39775-2_5

Vorschrift:

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus f_{K_i}(R_{i-1})), \quad 1 \leq i \leq r. \tag{5.1}$$

Dann setzt man

$$E_k(L_0, R_0) = (R_r, L_r).$$

Die Sicherheit der Feistel-Chiffre hängt natürlich zentral von der Sicherheit der internen Blockchiffre ab. Deren Sicherheit wird aber durch iterierte Verwendung noch gesteigert.

Wir erläutern die Entschlüsselung der Feistel-Chiffre. Aus (5.1) folgt unmittelbar

$$(R_{i-1}, L_{i-1}) = (L_i, R_i \oplus f_{K_i}(L_i)), \quad 1 \leq i \leq r. \tag{5.2}$$

Daher kann man unter Verwendung der Schlüsselfolge $(K_r, K_{r-1}, \ldots, K_1)$ in r Runden das Paar (R_0, L_0) aus dem Schlüsseltext (R_r, L_r) zurückgewinnen. Die Feistel-Chiffre wird also entschlüsselt, indem man sie mit umgekehrter Schlüsselfolge auf den Schlüsseltext anwendet.

5.2 Der DES-Algorithmus

Das DES-Verschlüsselungsverfahren ist eine leicht modifizierte Feistel-Chiffre mit Alphabet $\{0, 1\}$ und Blocklänge 64. Wir erläutern seine Funktionsweise im Detail.

5.2.1 Klartext- und Schlüsselraum

Klartext- und Schlüsseltextraum des DES ist $\mathcal{P} = \mathcal{C} = \{0, 1\}^{64}$. Die DES-Schlüssel sind Bitstrings der Länge 64, die folgende Eigenschaft haben: Teilt man einen String der Länge 64 in acht Bytes auf, so ist jeweils das letzte Bit eines jeden Bytes so gesetzt, dass die Quersumme aller Bits im betreffenden Byte ungerade ist. Es ist also

$$\mathcal{K} = \left\{ (b_1, \ldots, b_{64}) \in \{0, 1\}^{64} : \sum_{i=1}^{8} b_{8k+i} \equiv 1 \mod 2, 0 \leq k \leq 7 \right\}.$$

Die ersten sieben Bits eines Bytes in einem DES-Schlüssel legen das achte Bit fest. Dies ermöglicht Korrektur von Speicher- und Übertragungsfehlern. In einem DES-Schlüssel sind also nur 56 Bits frei wählbar. Insgesamt gibt es $2^{56} \sim 7.2 * 10^{16}$ viele DES-Schlüssel. Der DES-Schlüssel für Ver- und Entschlüsselung ist derselbe.

Beispiel 5.1 Ein gültiger DES Schlüssel ist hexadezimal geschrieben

$$133457799BBCDFF1.$$

Seine Binärentwicklung findet sich in Tab. 5.1.

Tab. 5.1 Gültiger DES-Schlüssel

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

5.2.2 Die initiale Permutation

Wir werden jetzt den DES-Algorithmus im einzelnen beschreiben. Bei Eingabe eines Klartextwortes p arbeitet er in drei Schritten.

Zusätzlich zur Feistel-Verschlüsselung wird im ersten Schritt auf p eine *initiale Permutation* IP angewandt. Dies ist eine für das Verfahren fest gewählte, vom Schlüssel unabhängige, Bitpermutation auf Bitvektoren der Länge 64. Die Permutation IP und die entsprechende inverse Permutation findet man in Tab. 5.2

Tab. 5.2 ist folgendermaßen zu verstehen. Ist $p \in \{0,1\}^{64}$, $p = p_1 p_2 p_3 \dots p_{64}$, dann ist $\mathrm{IP}(p) = p_{58} p_{50} p_{42} \dots p_7$.

Auf das Ergebnis dieser Permutation wird eine 16-Runden Feistel-Chiffre angewendet. Zuletzt wird die Ausgabe als

$$c = \mathrm{IP}^{-1}(R_{16} L_{16})$$

erzeugt.

5.2.3 Die interne Blockchiffre

Als nächstes wird die interne Blockchiffre beschrieben. Ihr Alphabet ist $\{0,1\}$, ihre Blocklänge ist 32 und ihr Schlüsselraum ist $\{0,1\}^{48}$. Wir erläutern, wie die Verschlüsselungsfunktion f_K zum Schlüssel $K \in \{0,1\}^{48}$ funktioniert (siehe Abb. 5.1).

Tab. 5.2 Die initiale Permutation IP

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

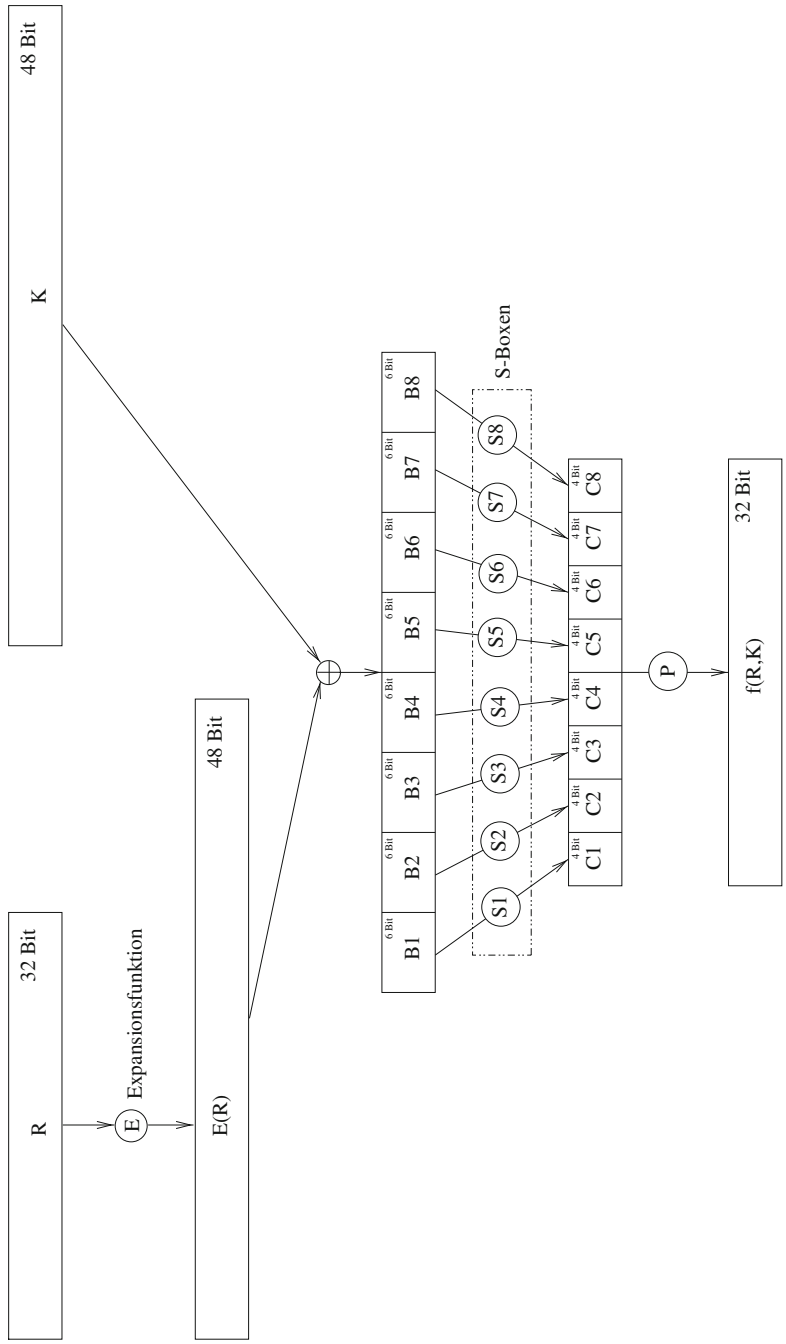

Abb. 5.1 Schema der f-Funktion im DES

Tab. 5.3 Die Funktionen E und P

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Das Argument $R \in \{0,1\}^{32}$ wird mittels einer Expansionsfunktion $\mathrm{E} : \{0,1\}^{32} \to \{0,1\}^{48}$ verlängert. Diese Funktion ist in Tab. 5.3 dargestellt. Ist $R = R_1 R_2 \ldots R_{32}$, dann ist $E(R) = R_{32} R_1 R_2 \ldots R_{32} R_1$.

Anschließend wird der String $E(R) \oplus K$ gebildet und in 8 Blöcke B_i, $1 \leq i \leq 8$, der Länge 6 aufgeteilt. Es wird also

$$E(R) \oplus K = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 \tag{5.3}$$

gebildet mit $B_i \in \{0,1\}^6$, $1 \leq i \leq 8$. Im nächsten Schritt werden Funktionen

$$S_i : \{0,1\}^6 \to \{0,1\}^4, \quad 1 \leq i \leq 8$$

verwendet (die sogenannten S-Boxen), die unten noch genauer beschrieben sind. Mit diesen Funktionen wird der String

$$C = C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8$$

berechnet, wobei $C_i = S_i(B_i)$, $1 \leq i \leq 8$, ist. Er hat die Länge 32. Dieser Bitstring wird gemäß der Permutation P aus Tab. 5.3 permutiert. Das Ergebnis ist $f_K(R)$.

5.2.4 Die S-Boxen

Wir beschreiben nun die Funktionen S_i, $1 \leq i \leq 8$. Diese Funktionen heißen S-Boxen. Sie werden in Tab. 5.4 dargestellt. Jede S-Box wird durch eine Tabelle mit vier Zeilen und 16 Spalten beschrieben. Für einen String $B = b_1 b_2 b_3 b_4 b_5 b_6$ wird der Funktionswert $S_i(B)$ folgendermaßen berechnet. Man interpretiert die natürliche Zahl mit Binärentwicklung $b_1 b_6$ als Zeilenindex und die natürliche Zahl mit Binärentwicklung $b_2 b_3 b_4 b_5$ als Spaltenindex. Den Eintrag in dieser Zeile und Spalte der S-Box stellt man binär dar und füllt diese Binärentwicklung vorne so mit Nullen auf, dass ihre Länge 4 wird. Das Ergebnis ist $S_i(B)$.

Zeile	Spalte															
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
S_1																
[0]	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
[1]	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
[2]	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
[3]	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2																
[0]	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
[1]	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
[2]	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
[3]	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3																
[0]	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
[1]	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
[2]	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
[3]	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4																
[0]	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
[1]	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
[2]	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
[3]	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5																
[0]	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
[1]	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
[2]	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
[3]	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6																
[0]	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
[1]	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
[2]	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
[3]	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7																
[0]	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
[1]	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
[2]	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
[3]	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8																
[0]	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
[1]	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
[2]	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
[3]	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Tab. 5.4 S-Boxen des DES

Beispiel 5.2 Wir berechnen $S_1(001011)$. Das erste Bit des Argumentes ist 0 und und das letzte Bit ist 1. Also ist der Zeilenindex die ganze Zahl mit Binärentwicklung 01, also 1. Die vier mittleren Bits des Argumentes sind 0101. Dies ist die Binärentwicklung von 5. Also ist der Spaltenindex 5. In der ersten S-Box steht in Zeile 1 und Spalte 5 die Zahl 2. Die Binärentwicklung von 2 ist 10. Also ist $S_1(001011) = 0010$.

5.2.5 Die Rundenschlüssel

Zuletzt muss noch erklärt werden, wie die Rundenschlüssel berechnet werden. Sei ein DES-Schlüssel $k \in \{0,1\}^{64}$ gegeben. Daraus werden Rundenschlüssel K_i, $1 \leq i \leq 16$, der Länge 48 generiert. Dazu definiert man v_i, $1 \leq i \leq 16$, folgendermaßen:

$$v_i = \begin{cases} 1 & \text{für } i \in \{1,2,9,16\} \\ 2 & \text{andernfalls.} \end{cases}$$

Nun werden zwei Funktionen

$$\text{PC1} : \{0,1\}^{64} \to \{0,1\}^{28} \times \{0,1\}^{28}, \quad \text{PC2} : \{0,1\}^{28} \times \{0,1\}^{28} \to \{0,1\}^{48},$$

benutzt. Diese Funktionen werden unten beschrieben. Mit diesen Bausteinen erhält man die Schlüssel so:

1. Setze $(C_0, D_0) = \text{PC1}(k)$.
2. Für $1 \leq i \leq 16$ berechne K_i folgendermaßen. Setze C_i auf den String, den man durch einen zirkulären Linksshift um v_i Stellen aus C_{i-1} gewinnt und D_i auf den String, den man durch einen zirkulären Linksshift um v_i Stellen aus D_{i-1} gewinnt. Berechne dann $K_i = \text{PC2}(C_i, D_i)$.

Die Funktion PC1 bildet einen Bitstring k der Länge 64 auf zwei Bitstrings C und D der Länge 28 ab. Dies geschieht gemäß der Tab. 5.5. Die obere Hälfte der Tabelle beschreibt, welche Bits aus K in C verwendet werden. Ist $k = k_1 k_2 \ldots k_{64}$, dann ist $C = k_{57} k_{49} \ldots k_{36}$. Die untere Hälfte dient der Konstruktion von D, also $D = k_{63} k_{55} \ldots k_4$. Die Funktion PC2 bildet umgekehrt ein Paar (C, D) von Bitstrings der Länge 28 (also einen Bitstring der Länge 56) auf einen Bitstring der Länge 48 ab. Die Funktion wird in Tab. 5.5 dargestellt. Der Wert $\text{PC2}(b_1 \ldots b_{56})$ ist $b_{14} b_{17} \ldots b_{32}$.

Dies beendet die Beschreibung des DES-Algorithmus.

PC1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Tab. 5.5 Die Funktionen PC1 und PC2

5.2.6 Entschlüsselung

Um einen Chiffretext zu entschlüsseln, wendet man DES mit der umgekehrten Schlüsselfolge auf ihn an.

5.3 Ein Beispiel für DES

Im folgenden illustrieren wir die Arbeit von DES an einem Beispiel.

Verschlüsselt wird das Wort $p = 0123456789ABCDEF$. Dessen Binärentwicklung ist

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Die Anwendung von IP ergibt

1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0

In der ersten Zeile von IP(p) steht die umgekehrte zweite Spalte von p, in der zweiten Zeile von IP(p) steht die umgekehrte vierte Spalte von p usw. Damit ist

$$L_0 = 11001100000000001100110011111111,$$
$$R_0 = 11110000101010101111000010101010.$$

Wir verwenden den DES-Schlüssel aus Beispiel 5.1. Der ist

$$133457799BBCDFF1.$$

Seine Binärentwicklung ist

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Daraus berechnen wir den ersten Rundenschlüssel. Es ist

$$C_0 = 1111000011001100101010101111,\ D_0 = 0101010101100110011110001111$$
$$C_1 = 1110000110011001010101011111,\ D_1 = 1010101011001100111100011110$$

und daher

$$K_1 = 0001\ 1011\ 0000\ 0010\ 1110\ 1111\ 1111\ 1100\ 0111\ 0000\ 0111\ 0010.$$

Daraus gewinnt man

$$E(R_0) \oplus K_1 = 0110\ 0001\ 0001\ 0111\ 1011\ 1010\ 1000\ 0110\ 0110\ 0101\ 0010\ 0111,$$
$$f_{K_1}(R_0) = 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011$$

und schließlich

$$R_1 = 1110\ 1111\ 0100\ 1010\ 0110\ 0101\ 0100\ 0100.$$

Die anderen Runden werden analog berechnet.

5.4 Sicherheit des DES

Seit seiner Einführung ist der DES sorgfältig erforscht worden. Dabei wurden spezielle Verfahren entwickelt, mit denen man den DES angreifen kann. Die wichtigsten sind die differentielle und die lineare Kryptoanalyse. Beschreibungen dieser Angriffe und Referenzen finden sich in [49] und [71]. Bis jetzt ist aber der erfolgreichste Angriff die vollständige Durchsuchung des Schlüsselraums. Mit speziellen Computern und weltweiten Computernetzen ist es heute möglich, DES-verschlüsselte Dokumente in wenigen Tagen zu entschlüsseln. Es wird erwartet, dass in Kürze DES auf einem PC gebrochen werden kann, weil PCs immer schneller werden.

DES kann heute nur noch als sicher gelten, wenn die in Abschn. 3.9 vorgestellte Dreifachverschlüsselung verwendet wird. Es ist dazu wichtig, festzustellen, dass die DES-Verschlüsselungsfunktionen nicht abgeschlossen unter Hintereinanderausführung sind. Sie bilden also keine Untergruppe der Permutationsgruppe $S_{64!}$. Würden die DES-Verschlüsselungsfunktionen eine Gruppe bilden, dann könnte man für zwei DES-Schlüssel k_1, k_2 einen dritten DES-Schlüssel k_3 finden, für den $\mathrm{DES}_{k_1} \circ \mathrm{DES}_{k_2} = \mathrm{DES}_{k_3}$ gelten würde. Mehrfachverschlüsselung würde also keinen Sicherheitsvorteil bieten. Es ist bekannt, dass die 2^{56} DES-Verschlüsselungsfunktionen eine Gruppe erzeugen, die wenigstens die Ordnung 10^{2499} hat (siehe [49]).

5.5 Übungen

Übung 5.1 Verifizieren Sie das Beispiel aus Abschn. 5.3 und berechnen Sie die zweite Runde.

Übung 5.2 Berechnen Sie die dritte Verschlüsselungsrunde in Abschn. 5.3.

Übung 5.3 Zeigen Sie, dass für $m, k \in \{0,1\}^{64}$ immer $\overline{\mathrm{DES}(m,k)} = \mathrm{DES}(\bar{m}, \bar{k})$ gilt.

Übung 5.4 Zeigen Sie, dass man C_{16} und D_{16} aus C_1 und D_1 durch einen zirkulären Rechtsshift um eine Position erhält.

Übung 5.5

1. Zeigen Sie, dass C_{16} und D_{16} aus C_1 und D_1 durch einen zirkulären Rechtsshift der Länge 1 entstehen.
2. Gelte $K_1 = K_2 = \ldots = K_{16}$. Zeigen Sie, dass alle Bits in C_1 gleich sind und ebenso alle Bits in D_1 gleich sind.
3. Folgern Sie, dass es genau vier DES-Schlüssel gibt, für die alle Teilschlüssel gleich sind. Dies sind die *schwachen DES-Schlüssel*.
4. Geben Sie die vier schwachen DES-Schlüssel an.

Übung 5.6 Welche der im DES verwendeten Funktionen IP $E(R) \oplus K$, S_i, $1 \leq i \leq 8$, P, PC1, PC2 sind für festen Schlüssel K linear und welche nicht? Beweisen sie die Linearität oder geben Sie Gegenbeispiele an.