DES – der vergangene Standard für Bitblock-Chiffren

Klaus Pommerening
Fachbereich Mathematik
der Johannes-Gutenberg-Universität
Saarstraße 21
D-55099 Mainz

Vorlesung Kryptologie
12. März 1991, letzte Änderung: 13. Oktober 2002

Der 'Data Encryption Standard' (DES) wurde im wesentlichen bei der IBM von einer Forschungsgruppe um Feistel und Copeersmith entwickelt; die NSA wirkte mit: Sie sorgte für eine Modifikation der S-Boxen und die Reduzierung der Schlüssellänge auf 56 Bit - entgegen allen Vermutungen ist beides nach heutigen Erkenntnissen keine Schwächung. Der DES wurde 1977 vom NBS ('National Bureau of Standards' – heute NIST) in den USA genormt. Das Ziel der Entwicklung war, für 10 bis 15 Jahre ein zuverlässiges Verschlüsselungssystem für sensible (aber nicht hochgeheime) Daten der Regierung zur Verfügung zu haben. Die Norm verlangt eine Hardware-Implementation des Algorithmus; von 1989 bis 1998 unterlagen DES-Chips der US-Ausfuhrbeschränkung. Der Algorithmus selbst heißt eigentlich DEA, schon lange werden aber auch Software-Implementationen als DES bezeichnet.

Verschlüsselt werden 64-Bit-Blöcke, wobei ein 56-Bit-Schlüssel verwendet wird. Die Verschlüsselung eines Blocks beginnt mit einer festen (bekannten) Permutation und endet mit der Umkehrpermutation. Obwohl diese Permutationen bekannt sind, wird dadurch schon eine gewisse Diffusion erreicht. Dazwischen werden 16 Runden durchgeführt, in denen sowohl Diffusion als auch Konfusion erhöht werden. Die einzelnen Runden unterscheiden sich nur dadurch, dass jeweils eine andere 48-Bit-Gruppe aus dem Schlüssel gewählt wird. Die Entschlüsselung unterscheidet sich von der Verschlüsselung nur dadurch, daß die Runden in umgekehrter Reihenfolge durchlaufen werden.

Im folgenden wird der Algorithmus stufenweise „von innen nach außen" beschrieben. $\oplus$ ist immer die bitweise Addition modulo 2 (XOR).

1 Die Kern-Abbildung

Im Innern des DES steckt die „Kern-Abbildung“

$$f\colon \mathbb{F}_2^{32} \times \mathbb{F}_2^{48} \longrightarrow \mathbb{F}_2^{32},$$

die als Input 32 Textbits und einen 48-Bit-Teilschlüssel hat. Zuerst werden die 32 Textbits durch teilweise Wiederholung zu 48 Bits aufgebläht; die „Expansionsabbildung“

$$E\colon \mathbb{F}_2^{32} \longrightarrow \mathbb{F}_2^{48}$$

wird durch die folgende Tabelle beschrieben:

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

Die Tabelle ist so zu interpretieren dass

$$E(b_1 b_2 \dots b_{32}) = b_{32} b_1 b_2 b_3 \dots b_{31} b_{32} b_1.$$

Die expandierten 48 Bits werden mit dem 48-Bit-Teilschlüssel per $\oplus$ überlagert. Die resultierenden 48 Bits werden in 8 Gruppen zu je 6 Bits zerteilt und auf diese die 1. bis 8. S(ubstitutions)-Box

$$S_j\colon \mathbb{F}_2^6 \longrightarrow \mathbb{F}_2^4 \quad (j = 1, \dots, 8)$$

angewendet. Die S-Boxen werden im nächsten Abschnitt beschrieben.

Insgesamt erhält man die (polyalphabetisch zusammengesetzte) Substitution

$$S\colon \mathbb{F}_2^{48} \longrightarrow \mathbb{F}_2^{32}.$$

Schließlich wird noch die P(ermutations)-Box

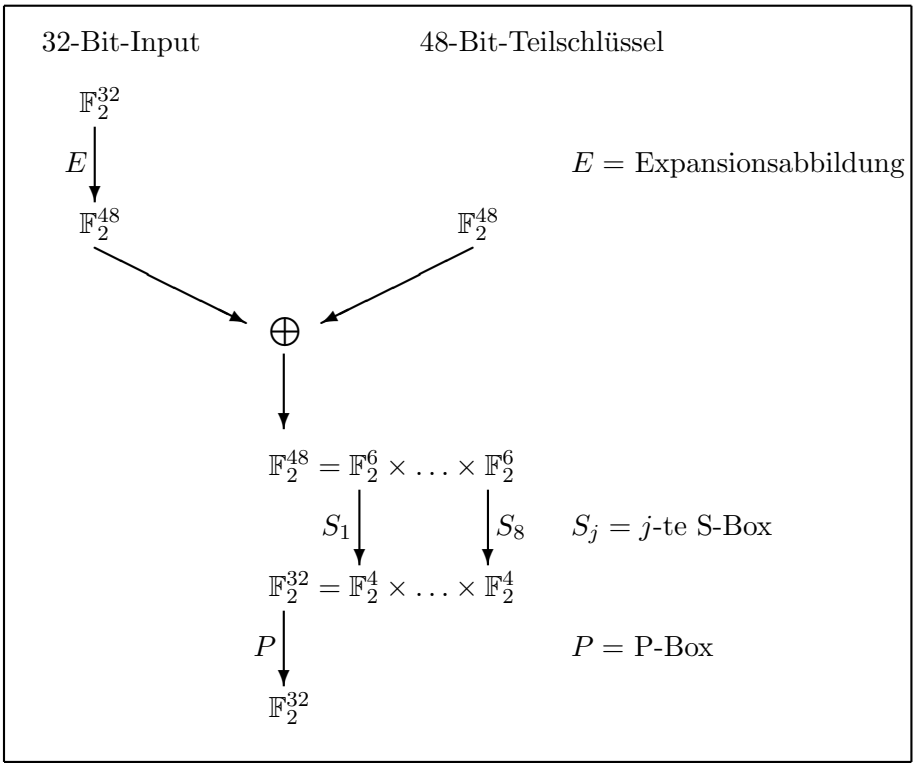
$$P\colon \mathbb{F}_2^{32} \longrightarrow \mathbb{F}_2^{32}$$

ausgeführt, die durch die folgende Tabelle beschrieben wird; das heißt,

$$P(b_1 b_2 \dots b_{32}) = b_{16} b_7 \dots b_4 b_{25}.$$

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

Zusammengefasst wird die Kernabbildung in der folgenden Abbildung:



2 Die S-Boxen

Jede der acht S-Boxen S_j wird durch eine 4×16-Matrix beschrieben, siehe die Tabelle:

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

Jede Zeile ist eine Permutation der Zahlen $0, \ldots, 15$. Um $S_j(b_1 \ldots b_6)$ zu bestimmen, deutet man $b_1 b_6$ als Binärdarstellung einer Zahl in $\{0, 3\}$ und $b_2 b_3 b_4 b_5$ als Binärdarstellung einer Zahl in $\{0, 15\}$, liest in der Matrix zu S_j die Zahl in Zeile $b_1 b_6$ und Spalte $b_2 b_3 b_4 b_5$ ab und stellt sie binär dar. Beispiel:

$$S_3(101100) = 0011 \quad \rightarrow \quad \text{Zeile 2, Spalte 6.}$$

3 Die Runden

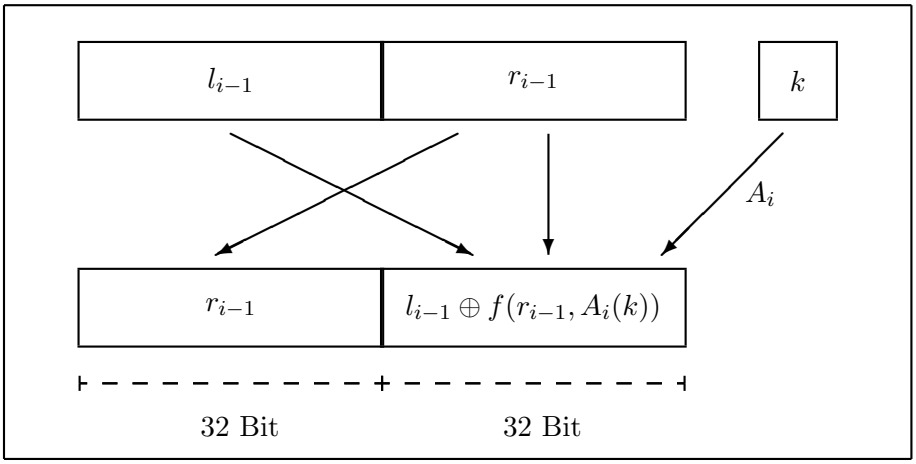
Die 16 Runden im DES bestehen aus je einer Abbildung

$$R_i \colon \mathbb{F}_2^{64} \times \mathbb{F}_2^{56} \longrightarrow \mathbb{F}_2^{64} \quad (i = 1, \ldots, 16),$$

die mit Hilfe der i-ten Schlüsselauswahl

$$A_i \colon \mathbb{F}_2^{56} \longrightarrow \mathbb{F}_2^{48} \quad (i = 1, \ldots, 16),$$

wie in der folgenden Abbildung beschrieben wird.



Die Runden unterscheiden sich also nur durch den verwendeten Teilschlüssel $A_i(k)$. Man erkennt hier das Feistel-Schema.

4 Die Schlüsselauswahl

Zur Beschreibung der Runden gehört noch die Beschreibung der Schlüsselauswahl. Zunächst wird der 56-Bit-Schlüssel auf 64 Bit aufgebläht, indem nach je 7 Bits ein Paritätsbit eingefügt wird; welches, ist egal, man kann sogar beliebige Bits einfügen, da die zusätzlichen Bits nicht weiter verwendet werden. Jedenfalls ist der erste Schritt eine Abbildung

$$Par\colon \mathbb{F}_2^{56} \longrightarrow \mathbb{F}_2^{64}.$$

Im zweiten Schritt werden die ursprünglichen 56 Bits wieder extrahiert, allerdings in der Reihenfolge der folgenden Tabelle.

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

Das ist eine Abbildung

$$PC_1\colon \mathbb{F}_2^{64} \longrightarrow \mathbb{F}_2^{56}$$

('Permuted Choice 1'). Nun werden die 56 Bits in zwei 28-Bit-Hälften geteilt und diese jeweils zyklisch nach links geschoben, insgesamt 16 mal. Das sind also 16 Abbildungen

$$LS_i\colon \mathbb{F}_2^{28} \longrightarrow \mathbb{F}_2^{28} \quad (i = 1, \ldots 16);$$

wie weit geschoben wird, zeigt die Tabelle:

1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Die ersten beiden Male wird also um ein Bit geschoben, dann 6 mal um zwei Bits usw. Für die i-te Schlüsselauswahl A_i wird nach der i-ten Verschiebung noch die 'Permuted Choice 2',

$$PC_2\colon \mathbb{F}_2^{56} \longrightarrow \mathbb{F}_2^{48}$$

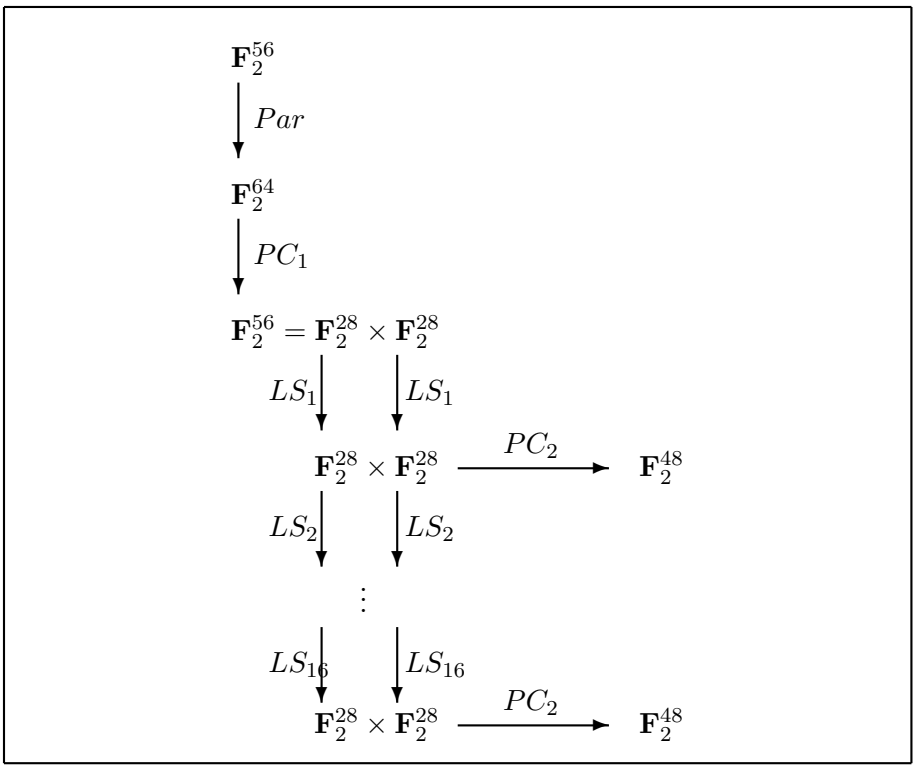
ausgeführt, wobei die Auswahl in der Reihenfolge der folgenden Tabelle geschieht (die Bits 9, 18, 22, 25, 35, 38, 43, 54 entfallen dabei).

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

Insgesamt ist

$$A_i = PC_2 \circ LS_i \circ \cdots \circ LS_1 \circ PC_1 \circ Par.$$

Diese Konstruktion wird noch einmal in dieser Abbildung zusammengefaßt:



5 Der gesamte Algorithmus

Nun ist noch die Initial-Permutation

$$IP\colon \mathbf{F}_2^{64} \longrightarrow \mathbf{F}_2^{64}$$

zu beschreiben; das geschieht durch diese Tabelle:

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

Invers zu IP ist die Ausgabe-Permutation IP^{-1}; der Bequemlichkeit halber wird die zugehörige Tabelle ebenfalls angegeben:

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

Der gesamte DES-Algorithmus DES_k zum Schlüssel $k \in \mathbf{F}_2^{56}$ ist nun die Zusammensetzung

$$\mathbf{F}_2^{64} \xrightarrow{IP} \mathbf{F}_2^{64} \xrightarrow{R_1(\bullet,k)} \dots \xrightarrow{R_{16}(\bullet,k)} \mathbf{F}_2^{64} \xrightarrow{T} \mathbf{F}_2^{64} \xrightarrow{IP^{-1}} \mathbf{F}_2^{64}.$$

Dabei ist T die Vertauschung der linken und der rechten 32-Bit-Hälften, die man einschiebt, damit DES_k^{-1} bis auf die umgekehrte Reihenfolge der Runden wie DES_k aussieht.

Anmerkung. Der Sinn von Initial- und Ausgabe-Permutation liegt wohl in einer bequemen Verdrahtung von Input und Output auf kleinen Prozessoren. Kryptologisch ergeben die Permutationen keinen Effekt, da beide ja vom Kryptoanalytiker ohne weiteres abgestreift werden können. Für eine Software-Implementierung wirken sie nur als Bremsen; dennoch dürfen sie nicht weggelassen werden, wenn man keine zum Standard inkompatible Implementation will.