Sommersemester 2005

Hasso- Plattner- Institut, Universität Potsdam

Übung zur Vorlesung

Fachgebiet Internet- Technologien und - Systeme

Meinel / Kutzner

Aufgabenblatt 7

(NUR FÜR ULI-STUDENTEN)

Abgabe: Mo, 06.06.2005, bis 12 Uhr MEZ (per E-Mail an mathias.kutzner@hpi.uni- potsdam.de)

Thema: Secret- Key Krypto- Systeme

Erreichbare Punkte: 25

Aufgabe 1: 2 Punkte

Der Schlüsseltext JIVSOMPMQUVQA wurde mit der Verschiebungschiffre erzeugt ($Alphabnet = \{A, B, C, \ldots, Z\}$). Ermitteln Sie den Schlüssel und den Klartext.

Aufgabe 2: 5 Punkte

Entschlüsseln Sie den Chiffretext OFJDFOHFXOL, der mit einer affinen Chiffre verschlüsselt wurde. Benutzt wurden ein Alphabet mit 27 Zeichen, wobei die Buchstaben A bis Z durch die Zahlen 0 bis 25 präsentiert werden und das Leerzeichen durch die Zahl 26.
Sie konnten in Erfahrung bringen, dass das erste Wort „I“ ist (also ist das zweite Zeichen ein Leerzeichen).

Geben Sie den Entschlüsselungsschlüssel und den englischen Originaltext an. Stellen Sie Ihren Lösungsweg dar.

Hinweis: Es ist erlaubt, für die Entschlüsselung ein selbst entwickeltes Programm zu benutzen.

Aufgabe 3: 18 Punkte

Das folgende binäre Klartextwort *p* soll mit dem DES-Algorithmus mit Hilfe des Schlüssels *k* verschlüsselt werden. Berechnen Sie die erste Runde und geben Sie Ihren kompletten Rechenweg an.

p

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

k

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

Bitte wenden!

Hinweise:
Die initiale Permutation *IP* und ihre Inverse lauten bei DES:

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

Durch die Permutation *IP* wird Bit 58 des Klartextes an die erste Stelle verschoben, Bit 50 an die zweite Stelle und so weiter bis Bit 7 an der letzten Stelle steht.
Es gilt also: $p \in \{0,1\}^{64}$, $p = p_1 p_2 p_3 \dots p_{64} \Rightarrow IP(p) = p_{58} p_{50} p_{42} \dots p_7$.

Für die Generierung der Rundenschlüssel werden die Funktionen *PC1* und *PC2* benutzt:

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

Die Funktion *PC1* bildet einen Bitstring der Länge 64 auf zwei Bitstrings *C* und *D* der Länge 28 ab. Die obere Hälfte der Tabelle beschreibt, welche Bits aus *K* in *C* verwendet werden. Ist $k = k_1 k_2 \dots k_{64}$, dann ist $C = k_{57} k_{49} \dots k_{36}$. Die untere Hälfte dient der Konstruktion von *D*, also $D = k_{63} k_{55} \dots k_4$. Die Funktion *PC2* bildet umgekehrt ein Paar *(C, D)* von Bitstrings der Länge 28 (also zusammen ein Bitstring der Länge 56) auf einen Bitstring der Länge 48 ab. Der Wert . $PC2(b_1 \dots b_{56}) = b_{14} b_{17} \dots b_{32}$
Die Expansionsfunktion *E* und die Permutation *P* lauten:

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 der Expansionsfunktion *E* auf 48 Bit verlängert. Ist $R = R_1 R_2 \dots R_{32}$ dann ist $E(R) = R_{32} R_1 R_2 \dots R_{32} R_1$. Einige Schritte später wird auf den String *C* die Permutation *P* angewendet.

Weiter auf Seite 3

Die S-Boxen werden verwendet, um aus der Bitfolge $E(R) \oplus K = B_1B_2B_3B_4B_5B_6B_7B_8$ die Bittfolge $C = C_1C_2C_3C_4C_5C_6C_7C_8$ zu berechnen, wobei $C_i = S_i(B_i), 1 \le i \le 8$ ist. Jede S-Box wird durch eine Tabelle mit vier Zeilen und 16 Spalten beschrieben.

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

Bitte wenden!

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 $b_2 b_3 b_4 b_5$ als Spaltenindex. Den Eintrag in dieser Zeile und Spalte stellt man binär dar und füllt diese Binärnetwicklung vorne so mit Nullen auf, dass ihre Länge 4 wird. Das Ergebnis ist $S_i(B)$.