Übungen - Angewandte Kryptologie

Alexander Weigl

April 13, 2011

Contents

1. Übung 1

1.1. Aufgabe 1

Chiffre	**Alphabet**	**Geheimtext**	**Schlüsselraum**	**Länge**
Caesar	$\{A, \cdots, Z\}$	$\{A, \cdots, Z\}$	$\{3\}, \{1, \cdots, 25\}$	4,64
OTP	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}^*$	∞
DES	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}^{56}$	56

1.2. Aufgabe 2

a. Beweis: Äquivalenzrelation: $\equiv_n$

Reflexivität $\forall x : x \equiv_n x$

$$x \mod n \equiv r = x \mod n \Rightarrow x \equiv_n x \tag{1}$$

#

Symmetrie $\forall x, y : x \equiv_n y \Rightarrow y \equiv_n x$

$$\begin{aligned} x \equiv_n y \mod n \Rightarrow x \mod n = r = y \mod n \\ \Rightarrow y \mod n = x \mod n \Rightarrow y \equiv_n x \end{aligned}$$

#

Transitiviät $\forall x, y, z : x \equiv_n y, y \equiv_n z => x \equiv_n z$

$$\begin{aligned} n.V.x \mod n = r_x, \\ y \mod n = r_y \\ z \mod n = r_z \wedge r_x = r_y, r_y = r_z \Rightarrow \\ r_x = r_z \Rightarrow x \equiv_n z \mod n \end{aligned}$$

#

b.

z. Z. $[i]_n + [j]_n = [i + j]_n$

$$\text{Sei } a, b \in \mathbb{Z} \Rightarrow a = q_a n + r_a, b = q_b n + r_b$$

$$\begin{aligned}
&\Rightarrow a \in [r_a]_n, b \in [r_b]_n \\
&\Rightarrow [r_a]_n + [r_b]_n = \{\forall i : in(r_a + r_b)\} \\
&\Rightarrow a + b = q_a n + r_a + q_b n + r_b \\
&\equiv_n n(q_a + q_b) + r_a + r_b \\
&\equiv_n r_a + r_b \Rightarrow [r_a + r_b]_n
\end{aligned}$$

z. Z. $[i]_n \cdot [j]_n = [i \cdot j]_n$

$$\text{Sei } a, b \in \mathbb{Z} \Rightarrow a = q_a n + r_a, b = q_b n + r_b$$

$$\begin{aligned}
&\Rightarrow a \in [r_a]_n, b \in [r_b]_n \\
&\Rightarrow [r_a]_n * [r_b]_n = \{\forall i : in(r_a * r_b)\} \\
&\Rightarrow a * b \equiv_n (q_a n + r_a) * (q_b n + r_b) \\
&\equiv_n q_a q_b n + q_a n r_b + q_b n r_a + r_a * r_b \\
&\equiv_n n(q_a q_b + q_a r_b + q_b r_a) + r_a * r_b \\
&\equiv_n r_a * r_b = [r_a + r_b]_n
\end{aligned}$$

1.3.

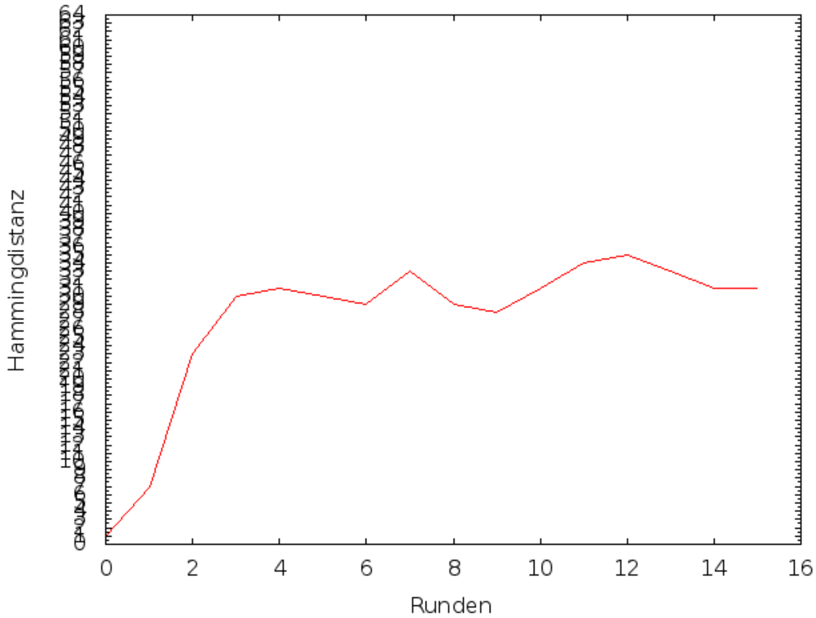
a.

$$\begin{aligned}
(23.145 \cdot = 12.479 + 14.543) \cdot \quad \mod 9 &\equiv_9 \\
(6 \cdot 5 + 8) \cdot 5 &\equiv_9 3 \\
8 \cdot 5 &\equiv_9 \\
2 \cdot 5 &\equiv_9 10 = 1 \mod 9
\end{aligned}$$

b.

$$\begin{aligned}
123 \cdot (123.983 \cdot 789.345 + 676.345) mod 11 &\equiv_{11} \\
2 \cdot (2 \cdot 7 + 10) &\equiv_{11} \\
2 \cdot (14 + 10) &\equiv_{11} 4 \mod 11
\end{aligned}$$

1.4.



Round k	$HD(M_k, M_k)$	$HD(M_{1_k}, M_{1_k})$	$HD(M_{1_k}, M_{1_k})$
0	1	39	35
1	7	35	31
2	23	32	33
3	30	36	35
4	31	37	34
5	30	31	34
6	29	26	30
7	33	26	26
8	29	29	26
9	28	29	34
10	31	25	36
11	34	28	35
12	35	30	38
13	33	29	37
14	31	32	38
15	31		

2. Übung 2

2.1. Aufgabe 1

a. Verschiebechiffren

$$E_1 : z \mapsto (z + k_1) \mod n \tag{2}$$
$$E_2 : z \mapsto (z + k_2) \mod n \tag{3}$$

Dann wäre die Verkettung $E_2 \circ E_1$:

$$\begin{aligned} E_2 \circ E_1 = E_2(E_1(z)) &= (((z + k_1) \mod n) + k_2) \mod n && (4)\\ &= z + \underbrace{k_1 + k_2}_{k_3} \mod n && (5)\\ &= z + k_3 \mod n = E_3(z) && (6) \end{aligned}$$

Wir folgern daraus, dass eine Verkettung von zwei Verschiebechiffren keine zusätzlichen Gewinn bringt.

b. Multiplikative Chiffren

$$E_1 : z \mapsto (z \cdot t_1) \mod n \tag{7}$$
$$E_2 : z \mapsto (z \cdot t_2) \mod n \tag{8}$$

Dann wäre die Verkettung $E_2 \circ E_1$:

$$\begin{aligned} E_2 \circ E_1 = E_2(E_1(z)) &= (((z \cdot t_1) \mod n) \cdot t_2) \mod n && (9)\\ &= z \cdot \underbrace{t_1 \cdot t_2}_{t_3} \mod n && (10)\\ &= z \cdot t_3 \mod n = E_3(z) && (11) \end{aligned}$$

Wir folgern daraus, dass eine Verkettung von zwei Multiplikativen Chiffren keine zusätzlichen Gewinn bringt.

c. Tauschchiffren

$$E_1 : z \mapsto (z \cdot t_1 + k_1) \mod n \tag{12}$$
$$E_2 : z \mapsto (z \cdot t_2 + k_2) \mod n \tag{13}$$

Dann wäre die Verkettung $E_2 \circ E_1$:

$$\begin{aligned} E_2 \circ E_1 = E_2(E_1(z)) &= (((z \cdot t_1 + k_1) \mod n) \cdot t_2 + k_2) \mod n && (14)\\ &= z \cdot \underbrace{t_1 \cdot t_2}_{t_3} + \underbrace{k_1 \cdot t_2 + k_2}_{k_3} \mod n && (15)\\ &= z \cdot t_3 + k_3 \mod n = E_3(z) && (16) \end{aligned}$$

Wir folgern daraus, dass eine Verkettung von zwei Tauschchiffren keine zusätzlichen Gewinn bringt.

2.2. Aufgabe 2

Berechnen Sie die multiplikativen Inverse zu $3, 5$ und 22 in Z_23.

$$\begin{aligned} 23 &= 7 \cdot 3 + 2\\ 3 &= 1 \cdot 2 + 1\\ 2 &= 2 \cdot 1 + 0 \end{aligned}$$

$$\begin{aligned} 1 &= 3 - 2\\ 1 &= 3 - (23 - 7 \cdot 3) =\\ 1 &= \underline{8} \cdot -1 \cdot 23 \end{aligned}$$

$$\begin{aligned} 23 &= 1 \cdot 15 + 8\\ 15 &= 1 \cdot 8 + 7\\ 8 &= 1 \cdot 7 + 1\\ 7 &= 7 \cdot 1 + 0 \end{aligned}$$

$$\begin{aligned} 1 &= 8 - 7\\ 1 &= (23 - 15) - (15 - 8)\\ 1 &= (23 - 15) - (15 - (23 - 15))\\ 1 &= 23 - 15 - 15 + 23 - 15\\ 1 &= \underbrace{\underline{-3}}_{20} \cdot 15 + 2 \cdot 23 \end{aligned}$$

$$\begin{aligned} 23 &= 1 \cdot 22 + 1\\ 22 &= 22 \cdot 1 + 0 \end{aligned}$$

$$1 = 1 \cdot 23 \underbrace{-1}_{22} \cdot 22$$

Berechnen Sie die multiplikativen Inversen zu $3,\ 15$ **und** 22 **in** Z_{24}.

$$\begin{aligned} 24 &= 3 \cdot 8 + 0 \\ &\Rightarrow \neg\exists \text{ multiplikatives Inverses} \end{aligned}$$

$$\begin{aligned} 24 &= 1 \cdot 15 + 9 \\ 15 &= 1 \cdot 9 + 6 \\ 9 &= 1 \cdot 6 + 3 \\ 6 &= 2 \cdot 3 + 0 \\ &\Rightarrow \neg\exists \text{ multiplikatives Inverses} \end{aligned}$$

$$\begin{aligned} 24 &= 1 \cdot 22 + 2 \\ 22 &= 11 \cdot 2 + 0 \\ &\Rightarrow \neg\exists \text{ multiplikatives Inverses} \end{aligned}$$

Zeigen Sie, dass $(n-1)$ **in** Z_n **bzgl. der Multiplikation zu sich selbst invers ist.**

$$\begin{aligned} (n-1) \cdot (n-1) &\equiv_n n^2 - 2n + 1 && (17) \\ &\equiv_n 1 \mod n && (18) \end{aligned}$$

2.3. Aufgabe 3

Alphabet: `ABCDEFGHIJKLMNOPQRSTUVWXYZ` Alphabet: `UEBRDNWOLKMSIFHTACGJPQVXYZ`

> Wind Nord-Ost, Startbahn null-drei,
> Bis hier hör ich die Motoren.
> Wie ein Pfeil zieht sie vorbei,
> Und es dröhnt in meinen Ohren.
> Und der nasse Asphalt bebt,
> Wie ein Schleier staubt der Regen,
> Bis sie abhebt und sie schwebt
> Der Sonne entgegen.
> Über den Wolken

Muß die Freiheit wohl grenzenlos sein.
Alle Ängste, alle Sorgen, sagt man,
Blieben darunter verborgen und dann
Würde, was hier gross und wichtig erscheint,
Plötzlich nichtig und klein.

2.4. Aufgabe 4

a.

Die Playfair-Verschlüsselung stellt eine Substitution für Buchstaben-Paare dar. Es handelt sich um eine bigraphische monoalphabetische Methode. Ähnlich wie bei der einfachen (monographischen) Buchstabensubstitution, beruhen Methoden zur Entzifferung von Playfair im Wesentlichen auf einer Analyse der Häufigkeitsverteilung hier der Buchstabenpaare (Bigramme). In der deutschen Sprache beispielsweise sind die Bigramme "er", "en" und "ch" sehr häufig. Im Beispieltext fallen die "Doppler" (also Bigramm-Wiederholungen) ME. . . ME, IK. . . IK, QC. . . QC und TE. . . TE sowie die "Reversen" (Wiederholung eines umgedrehten Bigramms) CQ. . . QC auf, die sich in gleicher Weise im englischen Klartext wiederfinden. Da kein Buchstabe mit sich selbst gepaart wird, gibt es nur 600 (25×24) mögliche Buchstabenkombinationen, die substituiert werden. Überdies gibt es eine Reihe von Symmetrien, die teilweise schon am obigen Beispieltext erkannt werden können. So hilft der erwähnte Klartext-Geheimtext-Zusammenhang EL ↔ CQ und LE ↔ QC beim Bruch des Textes. Ist nämlich ein Bigramm geknackt, dann ist auch sofort das reverse (umgedrehte) Bigramm bekannt. In den Fällen des Überkreuz-Schrittes gibt es darüber hinaus noch weitere Beziehungen zwischen den vier auftretenden Buchstaben in der Art (vgl. beispielsweise obere linke Ecke des Quadrats) DC ↔ EB, CD ↔ BE, EB ↔ DC sowie BE ↔ CD, die der Angreifer zur Entzifferung ausnutzen kann. Ferner hat auch die geschilderte Methode zur Erzeugung des Playfair-Quadrats Schwächen, denn es endet häufig – wie auch im Beispiel – auf "XYZ". Die Playfair-Verschlüsselung ist somit weit entfernt von einer allgemeinen bigraphischen Methode mit völlig willkürlicher Zuordnung der Buchstabenpaare und stellt in der heutigen Zeit kein sicheres Verschlüsselungsverfahren mehr dar. So lassen sich mit modernen Mitteln auch relativ kurze Playfair-Texte in sehr kurzer Zeit brechen.

2.5. Aufgabe 5

Herleitung des Gleichungssystems:

$$Hx_1 + Ix_2 = \ddot{A} \qquad Lx_1 + Lx_2 = U \tag{19}$$

$$Hx_3 + Ix_4 = U \qquad Lx_3 + Lx_4 = K \tag{20}$$

$$\tag{21}$$

$$\begin{pmatrix} 7 & 8 & 0 & 0 \\ 0 & 0 & 7 & 8 \\ 11 & 11 & 0 & 0 \\ 0 & 0 & 11 & 11 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 26 \\ 20 \\ 20 \\ 10 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 8 & 0 & 0 & 26 \\ 0 & 0 & 7 & 8 & 20 \\ 11 & 11 & 0 & 0 & 20 \\ 0 & 0 & 11 & 11 & 10 \end{pmatrix} \quad \begin{pmatrix} 7 & 8 & 0 & 0 & 26 \\ 11 & 11 & 0 & 0 & 20 \\ 0 & 0 & 7 & 8 & 20 \\ 0 & 0 & 11 & 11 & 10 \end{pmatrix} \begin{array}{l} \mid \cdot 7^{-1} = 25 \\ \mid \cdot 11^{-1} = 8 \\ \mid \cdot 7^{-1} = 25 \\ \mid \cdot 7^{-1} = 8 \end{array} \tag{22}$$

$$\begin{pmatrix} 1 & 26 & 0 & 0 & 12 \\ 1 & 1 & 0 & 0 & 15 \\ 0 & 0 & 1 & 26 & 7 \\ 0 & 0 & 1 & 1 & 22 \end{pmatrix} \begin{array}{l} \cdot -1 \\ + \\ \cdot -1 \\ + \end{array} \quad \begin{pmatrix} 1 & 26 & 0 & 0 & 12 \\ 0 & 4 & 0 & 0 & 3 \\ 0 & 0 & 1 & 26 & 7 \\ 0 & 0 & 0 & 4 & 22 \end{pmatrix} \begin{array}{l} \\ \mid \cdot 4^{-1} = 22 \\ \\ \mid \cdot 4^{-1} = 22 \end{array} \tag{23}$$

$$\begin{pmatrix} 1 & 26 & 0 & 0 & 12 \\ 0 & 1 & 0 & 0 & 8 \\ 0 & 0 & 1 & 26 & 7 \\ 0 & 0 & 0 & 1 & 11 \end{pmatrix} \begin{array}{l} + \\ \cdot -26 \\ + \\ \cdot -26 \end{array} \quad \begin{pmatrix} 7 \\ 8 \\ 11 \\ 11 \end{pmatrix} = \begin{pmatrix} H \\ I \\ L \\ L \end{pmatrix} \tag{24}$$

Bildung der Inversen K^{-1}

$$\begin{pmatrix} 7 & 8 & 1 & 0 \\ 11 & 11 & 0 & 1 \end{pmatrix} \mid \cdot 7^{-1} = 25 \quad \begin{pmatrix} 1 & 26 & 25 & 0 \\ 11 & 11 & 0 & 1 \end{pmatrix} \begin{array}{l} \cdot 11^{1} \\ + \end{array}$$

$$\begin{pmatrix} 1 & 26 & 25 & 0 \\ 0 & 15 & 15 & 1 \end{pmatrix} \begin{array}{l} \qquad\qquad\qquad + \\ \mid \cdot 15^{-1} = 2 \quad \cdot 26 \end{array} \quad \begin{pmatrix} 1 & 0 & 28 & 6 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

$$K^{-1} = \begin{pmatrix} 28 & 6 \\ 1 & 2 \end{pmatrix} \tag{25}$$

Lösung: HILLISTEINFACHZUKNACKEN

2.6. Aufgabe 6

a. Warum gilt für zufällige Texte $I_r = \frac{1}{26} = 0.0385$?

Wirklicher Zufall würde bedeuten das jeder Buchstaben $a \in A$ gleich oft im Text vorkommt. Folglich handelt es sich um einen Laplace-Raum (wie beim Würfel) und die Warscheinlichkeit für $P(X = a) = \frac{1}{|A|}$. In unseren Fall ist $|A| = 26$.

b.

c.

2.7. Übungsaufgabe: One-Time-Pad

$$P_1 = hike = 001\ 010\ 011\ 000 \tag{26}$$
$$P_2 = rike = 101\ 010\ 011\ 000 \tag{27}$$
$$C = eier = 000\ 010\ 000\ 101 \tag{28}$$
$$K = klet = 011\ 100\ 000\ 111 \tag{29}$$
$$\tag{30}$$

a.

$$001\ 010\ 011\ 000\,\mathrm{xor} \tag{31}$$
$$000\ 010\ 000\ 101 = \tag{32}$$
$$001\ 000\ 011\ 101 = hekr \tag{33}$$

$$101\ 010\ 011\ 000\,\mathrm{xor} \tag{34}$$
$$000\ 010\ 000\ 101 = \tag{35}$$
$$101\ 000\ 011\ 101 = rekr \tag{36}$$

b.

$$001\ 010\ 011\ 000\,\mathrm{xor} \tag{37}$$
$$011\ 100\ 000\ 111 = \tag{38}$$
$$010\ 110\ 011\ 111 = iskt \tag{39}$$

$$101\ 010\ 011\ 000\ \ \mathrm{xor} \tag{40}$$
$$011\ 100\ 000\ 111\ \ = \tag{41}$$
$$110\ 110\ 011\ 010\ \ = sski \tag{42}$$

c.

wenn $C_{1_i} = C_{2_i} \Rightarrow P_{1_i} = P_{2_i}$

$$C_1\,\mathrm{xor}\,C_2 = (P_1\,\mathrm{xor}\,K)\,\mathrm{xor}(P_2\,\mathrm{xor}\,K) = P_1\,\mathrm{xor}\,P_2 \tag{43}$$

d.

$$K = C_1 \operatorname{xor} P_1 \tag{44}$$
$$C_2 \operatorname{xor} K = P_2 \tag{45}$$

2.8. Skytale

a.

$$E(k, x_1, \dots, x_{km}) = \tag{46}$$
$$x_1 x_{m+1} x_{2m+1} \dots x_{(k-1)m+1} x_2 x_{m+2} x_{2m+2} \dots x_{(k-1)m+2} \dots x_m x_{2m} x_{3m} \dots x_k m \tag{47}$$
$$\begin{pmatrix} x_1 & x_2 & \cdots & x_m \\ x_{m+1} & x_{m+2} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(k-1)m+1} & x_{(k-1)m+2} & \cdots & x_{km} \end{pmatrix} \tag{48}$$

Ist die Klartextlänge kein Vielfaches von k, so kann der Klartext durch das Ein- bzw. Anfügen von sogenannten Blendern (Füllzeichen) verlängert werden. Damit der Emp fänger diese Füllzeichen nach der Entschlüsselung wieder entfernen kann, ist lediglich darauf zu achten, dass sie im Klartext leicht als solche erkennbar sind.

3. Moderne Symmetrische Chiffren

3.1. Lineare Abbildungen

3.2. Feistel-Cipher

a.

$F : \{0,1\}^4 \times \{0,1\}^4 \to \{0,1\}^4$ mit $F(X,Y) = X \operatorname{xor} Y$
die Rundenzahl $n = 2$,
der Plaintext $P = 10011100$ und
die Rundenschlüssel $K_1 = 0101$und $K_2 = 1100$.

Berechnen Sie den Ciphertext C.

$$L_i = R_{i-1} \tag{49}$$
$$R_i = L_{i-1} \operatorname{xor} F(R_{i-1}, K_i) \tag{50}$$

Rnd	K_i	L_i	R_i
0	—	1001	1100
1	0101	1100	0000
2	1100	0000	0000

Berechnen Sie aus C wieder den Plaintext P .

$$R_i = L_{i+1} \tag{51}$$
$$L_i = R_{i+1} \operatorname{xor} F(R_i, K_{i+1}) \tag{52}$$

Rnd	K_i	L_i	R_i
2	1100	0000	0000
1	0101	1100	0000
0	—	1001	1100

b.

Eine Feistel-Funktion ist definiert durch $F(X,Y) = X$. Berechnen Sie den Ciphertext C in Abhängigkeit von einer beliebigen Rundenzahl n und dem Plaintext $P = (L_0, R_0)$ Wie gut ist die dadurch erreichte Verschlüsselung?

Rnd	K_i	L_i	R_i
0	—	a	b
1	—	b	$b \operatorname{xor} a$
2	—	$b \operatorname{xor} a$	a
3	—	a	b

$$f(n,(a,b)) = \begin{cases} (a,b), & n \mod 3 = 0 \\ (b, b \operatorname{xor} a), & n \mod 3 = 1 \\ (b \operatorname{xor} a, a), & n \mod 3 = 2 \end{cases} \tag{53}$$

3.3. DES-Details

a. Zeigen Sie, dass die DES-Expansionspermutation eine lineare Abbildung ist.

$$A = \begin{pmatrix} 31 & 0 & 1 & 2 & 3 & 4 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 8 & 9 & 10 & 11 & 12 & 11 & 12 & 13 & 14 & 15 & 16 \\ 15 & 16 & 17 & 18 & 19 & 20 & 19 & 20 & 21 & 22 & 23 & 24 \\ 23 & 24 & 25 & 26 & 27 & 28 & 27 & 28 & 29 & 30 & 31 & 0 \end{pmatrix} \tag{54}$$

Sei $P \in \mathbb{N}^{32\times 48}$ die entsprechende Permutationsmatrix für A und $f: A^{32} \to A^{48}, \quad x \mapsto P \cdot x$ die Permutationsfunktion.

zZ: f ist linear

Sei $x, y \in A^{32}$ mit $x = (x_0, x_1, \ldots, x_{31})$ und $y = (y_0, y_1, \ldots, y_{31})$ dann ist

$$f(\alpha x) = \begin{pmatrix} \alpha x_{31} & \alpha x_0 & \cdots & \alpha x_8 \\ \alpha x_7 & \alpha x_8 & \cdots & \alpha x_{16} \\ \alpha x_{15} & \alpha x_{16} & \ddots & \vdots \\ \alpha x_{23} & \alpha x_{24} & \cdots & \alpha x_0 \end{pmatrix} \tag{55}$$

$$= \alpha \begin{pmatrix} x_{31} & x_0 & \cdots & x_8 \\ x_7 & x_8 & \cdots & x_{16} \\ x_{15} & x_{16} & \ddots & \vdots \\ x_{23} & x_{24} & \cdots & x_0 \end{pmatrix} \tag{56}$$

$$= \alpha f(x) \tag{57}$$

$$f(x+y) = \begin{pmatrix} x_{31}+y_{31} & x_0+y_0 & \cdots & x_8+y_8 \\ x_7+x_7 & x_8+y_8 & \cdots & x_{16}+y_{16} \\ x_{15}+y_{15} & x_{16}+y_{16} & \ddots & \vdots \\ x_{23}+y_{23} & x_{24}+y_{24} & \cdots & x_0+y_0 \end{pmatrix} \tag{58}$$

$$= \begin{pmatrix} x_{31} & x_0 & \cdots & x_8 \\ x_7 & x_8 & \cdots & x_{16} \\ x_{15} & x_{16} & \ddots & \vdots \\ x_{23} & x_{24} & \cdots & x_0 \end{pmatrix} + \begin{pmatrix} y_{31} & y_0 & \cdots & y_8 \\ y_7 & y_8 & \cdots & y_{16} \\ y_{15} & y_{16} & \ddots & \vdots \\ y_{23} & y_{24} & \cdots & y_0 \end{pmatrix} \tag{59}$$

$$= f(x) + f(y) \tag{60}$$

#

b. Zeigen Sie, dass die DES S-Boxen keine lineare Abbildung sind.

zZ. S_i ist nicht linear. Wir nehmen die S1-Box und sei $x = 0 \wedge \alpha = 2$

$$\begin{aligned} \mathrm{S1}(\alpha x) &= \mathrm{S1}(000000_2) && (61) \\ &= 1110_2 = 14_{10} && (62) \end{aligned}$$

$$\begin{aligned} \alpha\,\mathrm{S1}(x) &= 2\,\mathrm{S1}(000000_2) && (63) \\ &= 2_{10} * 1110_2 = 28_{10} && (64) \end{aligned}$$

$$\Rightarrow \mathrm{S1}(\alpha x) \neq \alpha\,\mathrm{S1}(x) \qquad (65)$$

#

c. Geben Sie für die DES-Permutation die Zykelschreibweise an.

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \\ 15 & 6 & 19 & 20 & 28 & 11 & 27 & 16 & 0 & 14 & 22 & 25 & 4 & 17 & 30 & 9 & 1 & 7 & 23 & 13 & 31 & 26 & 2 \end{pmatrix} \qquad (66)$$

$$\text{Zyklen:}(0\,15\,9\,14\,30\,3\,20\,31\,24\,18\,23\,8)(1\,6\,27\,5\,11\,25\,12\,4\,28\,21\,26\,29\,10\,22\,2\,19\,13\,17\,7\,16) \qquad (67)$$

d. Zeigen Sie, dass für den DES Schlüssel $K = 0xE0E0E0E0F1F1F1F1$ (inklusive Parity-Bits) alle Rundenschlüssel identisch sind. Warum gilt in diesem Fall $DES(K, DES(K, M)) = M$ (d.h. Ver- und Entschlüsselung sind identisch)?

$$K = \begin{matrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{matrix} \xrightarrow{PC1} \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad (68)$$

Da alle in den folgenden Runden lediglich zu Spaltentransposition kommt und jede Spalte gleich sind, sind alle C_i, D_i für alle $1 \leq i \leq 16$.

3.4. A5/1

$$\begin{aligned} X &= (x_0, x_1 \cdots, x_{18}) = (1010101010101010101) && (69) \\ Y &= (y_0, y_1, \cdots, y_{21}) = (1100110011001100110011) && (70) \\ Z &= (z_0, z_1, \cdots, z_{22}) = (11100001111000011110000) && (71) \end{aligned}$$

while *True* **do**
  $m = maj(x_8, y_{10}, z_{10})$;
  if $x_8 = m$ **then**
    $t = x13 \operatorname{xor} x16 \operatorname{xor} x17 \operatorname{xor} x18$;
    shift t into X;
  if $y_{10} = m$ **then**
    $t = y_{20} \operatorname{xor} y_{21}$;
    shift t into Y;
  if $z_{10} = m$ **then**
    $t = z_7 \operatorname{xor} z_{20} \operatorname{xor} z_{21} \operatorname{xor} z_{22}$;
    shift t into Y;
  $k_i = x_{18} \operatorname{xor} y_{21} \operatorname{xor} z_{22}$;

a.

Rnd	x_8	y_{10}	z_{10}	m	X'	Y'	Z'	Bit
1	1	0	1	1	0101010101010101010*1*	1100110011001100110011	01110000111100001111000*0*	1
2	0	0	1	0	0010101010101010101*0*	0110011001100110011001*1*	01110000111100001111000	0
3	1	1	1	1	1001010101010101010*1*	0011001100110011001100*1*	10111000011110000111100*0*	0

b.

```java
import java.util.BitSet;

public class A51 {
    ShiftRegister x, y, z;

    public A51() {
        x = new ShiftRegister(19,349525);
        y = new ShiftRegister(22,3355443);
        z = new ShiftRegister(23,7401712);
    }

    public int maj(int i, int j, int k) {
        int a = i << 2 | j << 1 | k;
        switch (a) {
        case 0:// 000
        case 1:// 001
        case 4:// 100
        case 2:// 010
            return 0;
        }
        return 1;
```

```
}

public int next() {
    int m = maj(x.get(8), y.get(10), z.get(10));

    if (m == x.get(8)) {
        int t = xor(x.get(13), x.get(16), x.get(17), x.get(18));
        x.push(t);
    }
    if (m == y.get(10)) {
        int t = xor(y.get(20), y.get(21));
        y.push(t);
    }

    if (m == z.get(10)) {
        int t = xor(z.get(7), z.get(20), x.get(21), x.get(22));
        z.push(t);
    }
    return xor(x.get(18), y.get(21), z.get(22));
}

private int xor(int i, int j, int k, int l) {
    return i ^ j ^ k ^ l;
}

private int xor(int i, int j, int k) {
    return i ^ j ^ k;
}

private int xor(int i, int j) {
    return i ^ j;
}

@Override
public String toString() {
    return "x: " + x + "\ny: " + y + "\nz; " + z;
}

public static void main(String[] args) {
    A51 c = new A51();

    System.out.println(c);

    for (int i = 0; i < 1000; i++)
```

```
            System.out.print(c.next());
        System.out.println();
    }

    private static void checkBin(int i) {
        if (i != 0 && i != 1)
            throw new IllegalArgumentException();
    }

    class ShiftRegister {
        int reg;
        int size;

        public ShiftRegister(int size, int content) {
            if (size >= 64)
                throw new IllegalArgumentException();
            this.size = size;
            reg=content;
        }

        public void push(int t) {
            checkBin(t);
            reg = (reg << 1) | t;
        }

        public int get(int i) {
            return (int) ((reg >> i) & 1);
        }

        @Override
        public String toString() {
            StringBuilder b = new StringBuilder(Integer.toString(reg, 2));
            return String.format("%"+size+"s", b.reverse().toString());
        }
    }
}
```

3.5. RC4

```
import java.util.Arrays;

public class RC4 {
    int[] k, s = new int[256];
    byte L;
```

```
public RC4(int[] k2) {
    this.k = k2;
    L = (byte) k2.length;
    init();
}

private void init() {
    for (int i = 0; i < s.length; i++) {
        s[i] = (i % 256);
        assert s[i] >= 0;
    }
    int j = 0;
    for (int i = 0; i < s.length; i++) {
        j = (j + s[i] + k[i % L]) % 256;
        swap(s, j, i);
    }
}

private void swap(int[] b, int j, int i) {
    int t = b[i] % 256;
    b[i] = b[j];
    b[j] = t;
}

public void next(int[] plain, int[] ciph) {
    int i = 0, j = 0;
    for (int n = 0; n < plain.length; n++) {
        i = (i + 1) % 256;
        j = (j + s[i]) % 256;
        swap(s, i, j);
        int rand = s[(s[i] + s[j]) % 256];
        ciph[n] = (rand ^ plain[n]) % 256;
    }
}

@Override
public String toString() {
    return Arrays.toString(s);
}

public static void main(String[] args) {
    int k[] = { 0x1A, 0x2B, 0x3C, 0x4D, 0x5E, 0x6F, 0x77 };
    RC4 rc4 = new RC4(k);
```

```
        System.out.println(rc4);
        int b[] = new int[100];
        rc4.next(b, new int[100]);
        System.out.println(rc4);
    }
}
```