KAPITEL 1

Einführung

1. Ein Grundproblem der Kryptographie

Die folgenden Beispiele sollen an Hand einiger Situationen ein Grundproblem der Kryptographie aufzeigen:

(1) Sie wollen eine Postkarte, einen Brief an einen Freund schicken, möchten nicht, dass der Inhalt von jedem gelesen werden kann. Was kann man tun?
(2) Sie benutzen ein tragbares Telefon oder Handy. Sie wollen nicht, dass jemand den Inhalt eines Gesprächs mitbekommt, wenn er auf dem Übertragungsweg die elektromagnetischen Wellen auffängt. Was macht man hier?
(3) Sie wollen jemandem per E-Mail etwas mitteilen, was andere nichts angeht. Sie wissen nicht, was während der Übertragung der E-Mail alles passiert. Was kann man hier tun?
(4) Sie benutzen Onlinebanking via Internet. Natürlich wollen Sie nicht, dass jemand wichtige Informationen (Geheimzahl, Passwort) mitbekommt, wenn er auf dem Übertragungsweg lauscht.

Die obigen Beispiele finden sich in folgender allgemeinen Situation wieder: Ein Sender A will eine Nachricht T an einen Empfänger B übermitteln. Dies soll aber so geschehen, dass ein Unbefugter C nichts damit anfangen kann, auch wenn er bei der Übermittlung an die Nachricht kommt. (Bei der Nachricht kann es sich auch um einen Text, eine Information oder eine Datei handeln.)

$$\begin{array}{ccc} A & \text{Übermittlung} & B \\ T & \stackrel{?}{\longrightarrow} & T \end{array}$$

Die Idee der Kryptographie ist es, die Ausgangsnachricht T nicht direkt zu übertragen, sondern sie mit Hilfe einer Transformation f zu einer Nachricht $\widetilde{T} = f(T)$ zu transformieren, zu verändern. Anschließend übermittelt A die Nachricht $\widetilde{T}$ an B. Der Empfänger B sollte die Umkehrtransformation f^{-1} kennen um sich damit die ursprüngliche Nachricht $T = f^{-1}(\widetilde{T})$ zu bestimmen.

$$\begin{array}{ccccc} & A & & \text{Übermittlung} & & B \\ T \stackrel{f}{\longrightarrow} f(T) = \widetilde{T} & & & \longrightarrow & & \widetilde{T} \stackrel{f^{-1}}{\longrightarrow} f^{-1}(\widetilde{T}) = T \end{array}$$

Der entscheidende Punkt ist jetzt, dass ein Unbefugter C nichts mit $\widetilde{T}$ anfangen kann, auch wenn er auf dem Übertragungsweg an die Nachricht $\widetilde{T}$ kommt. D.h. die Transformation f muss so beschaffen sein, dass ein Unbefugter von $\widetilde{T}$ nicht auf T schließen kann. Insbesondere muss die Umkehrtransformation f^{-1} geheimgehalten werden.

Sprechweisen: Die **Ausgangsnachricht** T wird auch **Klartext** oder **plaintext** genannt. Die Transformation f bezeichnet man als **Verschlüsselungsfunktion** oder **Chiffrierfunktion**. Das Anwenden der Funktion f bezeichnet man als **verschlüsseln** oder **chiffrieren**. Die transformierte Nachricht $\widetilde{T} = f(T)$ wird auch **verschlüsselte Nachricht**, **verschlüsselter Text**, **Geheimtext**, **Chiffretext** oder **ciphertext** genannt. Das Anwenden von f^{-1} nennt man **entschlüsseln**. Eine Zusammenstellung von Regeln, wie man einen Text verschlüsselt und wieder entschlüsselt, bezeichnet man auch als **Kryptosystem** (zur Verschlüsselung).

An Hand einiger einfacher Beispiele wollen wir die obigen Begriffe erläutern.

Datei: ki_einf.tex. Version vom 15.10.2024

2. Einfache Substitutionschiffren

CAESAR-3-Verschlüsselung: Auf Gaius Julius Caesar (100–44 v.Chr.) soll folgendes Verschlüsselungsverfahren zurückgehen:

(1) Wir definieren eine Funktion f auf den Großbuchstaben A,B,C,...,Z:

x	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$f(x)$	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

(2) **Verschlüsselung:** Ein (aus Großbuchstaben bestehender) Text $a_1a_2a_3\ldots$ wird verschlüsselt, indem man jeden Buchstaben x des Textes durch $f(x)$ ersetzt, man erhält also den Chiffretext $f(a_1)f(a_2)f(a_3)\ldots$.

(3) **Entschlüsselung:** Die Entschlüsselung funktioniert wie die Verschlüsselung, nur dass man statt f die Funktion f^{-1} verwendet, die man erhält, wenn man obige Tabelle von unten nach oben liest.

Beispiel: Mit der CAESAR-3-Verschlüsselung wird der Text „HEUTE IST MITTWOCH“ zu „KHXWH LVW PLWWZRFK“ verschlüsselt.

Bemerkung: Caesars Verschlüsselungsverfahren wird bei Sueton (De vita Caesarum, liber I, capitulum LVI) erwähnt: ‘Epistulae quoque eius ad senatum extant, quas primum videtur ad paginas et formam memorialis libellis convertisse, cum antea consules et duces non nisi transversa charta scriptas mitterent. Exstant et ad Ciceronem, item ad familiares domesticis de rebus, in quibus, si qua occultius perferenda erant, per notas scripsit, id est sic structo litterarum ordine, ut nullum verbum effici posset; quae si qui investigare et persequi velit, quartam elementorum litteram, id est D pro A et perinde reliquas commutet.’

Ein Text besteht aus einer Folge von Zeichen/Buchstaben/Symbolen. Die Menge aller (für die Situation) in Frage kommenden Zeichen wird als **Alphabet** Σ bezeichnet. Wichtig ist, dass man sich für ein Kryptosystem auf ein bestimmtes Alphabet einigt. Hat das Alphabet N Zeichen, so kann man eine Bijektion mit der Menge der ganzen Zahlen $\{0,1,2,\ldots,N-2,N-1\}$ herstellen.

Beispiele:

(1) Wir haben in unserem ersten Kryptosystem das Alphabet $\Sigma = \{\text{A,B,C,...,Z}\}$ mit 26 Buchstaben verwendet.

(2) Ein etwas größeres Alphabet erhält man, wenn man noch die Kleinbuchstaben a, b, c, ..., z, Leerzeichen, Interpunktionszeichen . , ! ? hinzunimmt.

(3) Wir werden auch das Alphabet mit den 2 Zeichen 0 und 1 betrachten.

(4) Eine Computerdatei ist eine Bitfolge, wobei ein Bit den Wert 0 oder 1 haben kann. Im Allgemeinen sind jeweils 8 Bits zu einem Byte zusammengefasst. (Statt ‘Byte’ findet man auch ‘octet’ (englisch) bzw. ‘Oktett’ (deutsch).) Ein Byte wird also durch eine Zahl zwischen 0 und $2^8-1=255$ dargestellt. Bytes schreibt man oft auch als 2-stellige Hexadezimalzahlen (mit den Ziffern 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f oder 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F).
   (a) Ist der Dateiname „Datei“, führt man in Python die Befehle
   ```
   f=open("Datei","rb")
   a=f.read()
   f.close()
   b=[z for z in a]
   ```
   aus, so enthält b die zugehörige Bytefolge (in Dezimaldarstellung).
   (b) Will man eine Liste b mit Bytes in eine Datei mit Namen „Datei“ schreiben, so lautet der Befehl in Python
   ```
   f=open("Datei","wb")
   f.write(bytes(b))
   f.close()
   ```

(5) Ein ascii-Zeichensatz (ascii=american standard code for imformation interchange) besteht aus 256 Zeichen, die durch die Zahlen $0,\ldots,255=2^8-1$ repräsentiert werden, z.B. $A\leftrightarrow 65$, $B\leftrightarrow 66$, ..., $Z\leftrightarrow 90$, $a\leftrightarrow 97$, ..., $z\leftrightarrow 122$, Return/Wagenrücklauf $\leftrightarrow 10$, Blank/Leerzeichen $\leftrightarrow 32$.

(a) In Python liefert `ord(z)` den ascii-Wert eines Zeichens z, während `chr(u)` das dem Zahlenwert u entsprechende Zeichen liefert.

Bemerkung: Identifiziert man die Großbuchstaben A,B,C,...,Z des ersten Kryptosystems mit den Zahlen von 0 bis 25, so lässt sich die Verschlüsselungsfunktion f schreiben als

$$f(x) = \begin{cases} x+3 & \text{für } 0 \le x \le 22, \\ x+3-26 & \text{für } 23 \le x \le 25, \end{cases}$$

oder zusammengefasst als

$$f(x) = x + 3 \bmod 26.$$

Dabei ist für $a \in \mathbb{Z}$ und $m \in \mathbb{N}$ die Zahl „$a \bmod m$“ der Rest der Division von a durch m, also $a - \lfloor \frac{a}{m} \rfloor m$. (Die modulo-Rechnung wird später ausführlich behandelt.) Die Umkehrabbildung lässt sich dann in der Form

$$f^{-1}(x) = x - 3 \bmod 26 \quad \text{oder} \quad f^{-1}(x) = x + 23 \bmod 26$$

schreiben.

Wollen zwei Personen sicher Nachrichten übermitteln, können sie sich eine Verschlüsselungsfunktion f ausdenken, die dann geheim gehalten werden muss. Hat man viele Personen, die paarweise geheim Informationen austauschen wollen, liegt gewöhnlich ein festgewähltes Verschlüsselungsverfahren zugrunde, das allgemein bekannt sein kann, das aber von einem Parameter K, dem sogenannten Schlüssel K abhängt, d.h. man hat f_K und f_K^{-1}. Für die Ver- und Entschlüsselungsfunktionen findet man oft auch die Schreibweisen $E_K = f_K$ (encrypt - verschlüsseln) und $D_K = f_K^{-1}$ (decrypt - entschlüsseln). Alles hängt dann von der Geheimhaltung des Schlüssels K ab.

Wir können jetzt unser erstes Kryptosystem CAESAR-3 leicht verallgemeinern:

CAESAR-Verschlüsselung/CAESAR-K-Verschlüsselung:

(1) Es werden nur Großbuchstaben A,...,Z berücksichtigt, die mit den Zahlen $0, \dots, 25$ identifiziert werden.
(2) **Schlüssel:** Ein Schlüssel besteht aus einer ganzen Zahl K. Dadurch wird eine Verschlüsselungsfunktion f_K definiert:
$$f_K(x) = x + K \bmod 26,$$
wobei $x + K \bmod 26$ den Rest der Division von $x + K$ durch 26 bezeichnet, der zwischen 0 und 25 liegt. (f_K ist also eine zyklische Verschiebung des Alphabets um K Stellen.)
(3) **Verschlüsselung:** Der Klartext $a_1a_2a_3\dots$ wird dann zu $f_K(a_1)f_K(a_2)f_K(a_3)\dots$ verschlüsselt, d.h. man ersetzt jeden Buchstaben x des Klartexts durch $f_K(x)$.
(4) **Entschlüsselung:** Es ist $f_K^{-1} = f_{-K}$. Aus dem Chiffretext $b_1b_2b_3\dots$ erhält man den Klartext, indem man auf jedes Zeichen f_K^{-1} anwendet: $f_K^{-1}(b_1)f_K^{-1}(b_2)f_K^{-1}(b_3)\dots$.

Bemerkungen:

(1) Haben K_1 und K_2 den gleichen Divisionsrest bei der Division durch 26, d.h. gilt $K_1 \bmod 26 = K_2 \bmod 26$, so gilt $f_{K_1} = f_{K_2}$. Daher gibt es eigentlich nur 26 verschiedene Schlüssel dieser CAESAR-Verschlüsselung.
(2) Vermutet man, dass die aus Großbuchstaben bestehende, verschlüsselte Nachricht T' CAESAR-verschlüsselt wurde, so kann man für alle Schlüssel $K \in \{0, 1, \dots, 25\}$ den Text $f_{-K}(T')$ bestimmen und sehen, ob etwas Sinnvolles dabei herauskommt.
(3) Natürlich überträgt sich die CAESAR-Verschlüsselung auch auf andere Alphabete Σ. Beispielsweise kann man Dateien CAESAR-verschlüsseln, wenn man die Datei als Bytefolge $a_1, a_2, a_3, \dots$ auffasst mit $0 \le a_i \le 255$ und die Verschlüsselungsfunktionen $f_K(x) = x + K \bmod 256$ verwendet. Hier hat man dann 256 Schlüssel.

Beispiel: Wir vermuten, dass `IEPPSKYD` CAESAR-verschlüsselt ist. Wir verschieben die Buchstaben jeweils um eine Stelle weiter:

```
IEPPSKYD -> JFQQTLZE -> KGRRUMAF -> LHSSVNBG -> MITTWOCH
```

Vermutlich war der Klartext also `MITTWOCH`, verschlüsselt mit CAESAR-22.

Bemerkung: Hinweise für Übungsaufgaben, die in Großbuchstaben geschrieben sind, werden wir meist CAESAR-13-verschlüsseln. (Man kann dann eine Aufgabe zunächst ohne Hinweis zu lösen versuchen.)

Bei der CAESAR-Chiffrierung werden die einzelnen Zeichen einer Zeichenkette mit einer Funktion $f_K : \Sigma \to \Sigma$ (wie oben) verschlüsselt. Die Funktionen f_K sind dabei bijektiv, also Permutationen des Alphabets Σ. Wir können die CAESAR-Chiffrierung nun verallgemeinern, indem wir für die Funktionen $f_K : \Sigma \to \Sigma$ beliebige Permutationen von Σ zulassen. Wir beschreiben eine praktikable Version für das aus den 26 Großbuchstaben bestehende Alphabet.

MASC-Verschlüsselung: (MASC - **m**ono**a**lphabetische **S**ubstitutions**c**hiffrierung)

(1) Es werden nur Großbuchstaben A,B,C,...,Z berücksichtigt.
(2) **Schlüssel:**
   (a) Der Schlüssel wird durch ein Wort K gegeben, also eine Zeichenfolge $K = k_1k_2k_3\dots k_n$.
   (b) Wir testen für $i = 1,\dots,n$, ob der Buchstabe k_i an einer Stelle $j > i$ nochmals vorkommt, d.h. ob $k_i = k_j$ gilt. Wenn ja, wird er an der Stelle j herausgestrichen. Das (eventuell abgeänderte) Wort K besteht nun aus lauter verschiedenen Buchstaben.
   (c) An K wird jetzt das (zyklisch permutierte) Alphabet angehängt, wobei mit dem nach k_n folgenden Buchstaben begonnen wird und alle Buchstaben weggelassen werden, die schon da waren. Es bleibt schließlich ein Wort mit 26 (verschiedenen) Buchstaben übrig: $K = k_1k_2\dots k_{26}$.
   (d) Die Verschlüsselungsabbildung $f_K : \{\text{A,B,C,}\dots\text{,Z}\} \to \{\text{A,B,C,}\dots\text{,Z}\}$ wird durch $f_K(\text{A}) = k_1$, $f_K(\text{B}) = k_2$, $f_K(\text{C}) = k_3$, ..., $f_K(\text{Z}) = k_{26}$ definiert:

x	A	B	C	...	Z
$f_K(x)$	k_1	k_2	k_3	...	k_{26}

   (Man schreibt unter das Alphabet ABC...Z das erweiterte Schlüsselwort $k_1k_2k_3\dots k_{26}$.) Liest man die Tabelle von unten nach oben, so erhält man die inverse Funktion f_K^{-1}.
(3) **Verschlüsselung:** Auf den zu verschlüsselnden Text $a_1a_2a_3\dots$ wendet man zeichenweise die Funktion f_K an und erhält so den Chiffretext $f_K(a_1)f_K(a_2)f_K(a_3)\dots$.
(4) **Entschlüsselung:** Auf den Chiffretext $b_1b_2b_3\dots$ wendet man zeichenweise die Entschlüsselungsfunktion f_K^{-1} an und erhält somit den Ausgangstext: $f_K(b_1)f_K(b_2)f_K(b_3)\cdots = a_1a_2a_3\dots$.

Beispiel: Das Schlüsselwort „ERLANGEN“ wird zuerst zu „ERLANG“ verkürzt, dann zu „ERLANGHIJKMOPQSTUVWXYZBCDF“ ergänzt und liefert die folgende Abbildung f_K:

x	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$f_K(x)$	E	R	L	A	N	G	H	I	J	K	M	O	P	Q	S	T	U	V	W	X	Y	Z	B	C	D	F

Damit wird „HEUTE IST FREITAG“ zu „INYXN JWX GVNJXEH“ verschlüsselt.

Bemerkungen:

(1) Mit dem Verfahren kann man alle

$$26! = 403291461126605635584000000 \approx 0.4 \cdot 10^{27}$$

Permutationen der 26 Großbuchstaben erhalten. (Obwohl es sehr viele Schlüssel gibt, ist dieses Chiffrierverfahren nicht sicher.)
(2) MASC ist ein Substitutionsverfahren, d.h. jeder Buchstabe des Klartexts wird durch ein anderes Zeichen ersetzt, die Position des Zeichens wird nicht verändert. Die Substitution ist monoalphabetisch, d.h. unabhängig von der Position im Text wird ein Buchstabe immer durch das gleiche Zeichen ersetzt.
(3) Die CAESAR-Chiffre ist ein Spezialfall der MASC-Chiffrierung, indem man als Schlüsselwort nur einen Buchstaben wählt. Der ergänzte Schlüssel ist dann ein zyklisch permutiertes Alphabet.

Bisher haben wir für Klartext und Geheimtext das gleiche Alphabet benutzt. Das muss natürlich nicht der Fall sein. Wir sprechen dann vom Klartextalphabet für den Klartext und vom Geheimtextalphabet für den Chiffretext.

Beispiel: $\Sigma_1 = \{A,\ldots,Z\}$ bestehe aus den Zeichen mit ascii-Werten zwischen 65 und 90, $\Sigma_2 = \{',\ldots,@\}$ bestehe aus den Zeichen mit den ascii-Werten zwischen 39 und 64. Die Verschlüsselungsabbildung werde dann für die ascii-Werte durch $f(x) = x - 26$ definiert, also

x	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$f(x)$	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	@

Aus „DEPARTMENT MATHEMATIK“ wird dann „*+6'8:3+4: 3':.+3':/1“.

Beispiel: In der Erzählung „The Gold-Bug“ von Edgar Allan Poe kommt folgender Chiffretext vor:

53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;1‡(;:‡*8†83(88
5*†;46(;88*96*?;8)*‡(;485);5*†2:*‡(;4956*2(5*-4)8¶8*;40692
85);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†528806*81(‡9;48;(8
8;4(‡?34;48)4‡;161;:188;‡?;

Es handelt sich um eine monoalphabetische Substitutionschiffrierung (MASC), wobei aber das Klartextalphabet nicht mit dem Chiffretextalphabet übereinstimmt. Die zugehörige Verschlüsselungsabbildung ergibt sich aus folgender Tabelle:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
5	2	-	†	8	1	3	4	6			0	9	*	‡	.		(	)	;	?	¶			:	

Als Klartext erhält man:

```
AGOODGLASSINTHEBISHOPSHOSTELINTHEDEVILSSEATFORTYONEDEGREESANDTHIRTEENMINUTESNORT
HEASTANDBYNORTHMAINBRANCHSEVENTHLIMBEASTSIDESHOOTFROMTHELEFTEYEOFTHEDEATHSHEADAB
EELINEFROMTHETREETHROUGHTHESHOTFIFTYFEETOUT
```

In der Erzählung von Poe wird auch der Chiffretext systematisch entschlüsselt und das Ergebnis angegeben:

```
A GOOD GLASS IN THE BISHOP'S HOSTEL IN THE DEVIL'S SEAT - FORTY-ONE
DEGREES AND THIRTEEN MINUTES - NORTHEAST AND BY NORTH - MAIN BRANCH
SEVENTH LIMB EAST SIDE - SHOOT FROM THE LEFT EYE OF THE DEATH'S HEAD - A
BEELINE FROM THE TREE THROUGH THE SHOT FIFTY FEET OUT.
```

3. Häufigkeitsanalyse

Häufigkeitsanalyse bedeutet, dass man in einem vorgegebenen Text zählt, wie häufig Zeichen oder Zeichenfolgen auftreten. (Eine Folge von n Zeichen bezeichnet man auch als n-Gramm.) Wir beginnen mit einem Beispiel.

Beispiel: Wir haben den Abschnitt „Ankunft“ des ersten Kapitels von „Der Zauberberg“ von Thomas Mann hergenommen, Umlaute umgewandelt (Ä in Ae, Ö in Oe, Ü in Ue, ä in ae, ö in oe, ü in ue, ß in ss), dann alles in Großbuchstaben umgewandelt und die Sonderzeichen weggelassen. Es blieben 15442 Großbuchstaben übrig. Wir haben dies in 5 Teile $T_1, \ldots, T_5$ geteilt zu 3000, 3000, 3000, 3000 und 3442 Zeichen und für die einzelnen sowie für den gesamten Abschnitt die Häufigkeiten der einzelnen Buchstaben

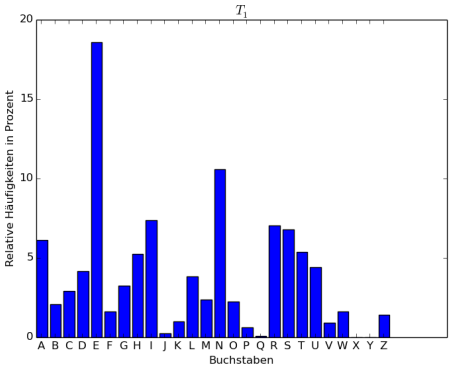
bestimmt. Das Ergebnis findet sich in folgender Tabelle, wobei die Zahlen in Prozent angegeben sind:

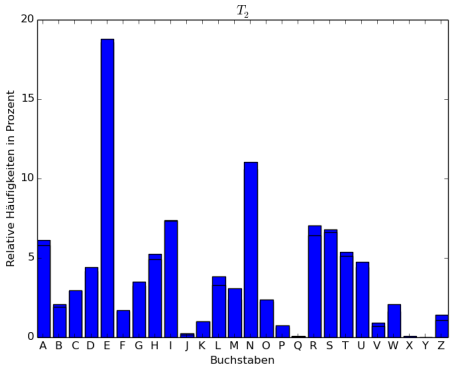
	T_1	T_2	T_3	T_4	T_5	gesamt
A	6.13	5.80	6.53	7.23	6.07	6.35
B	2.07	1.93	1.83	2.03	1.92	1.96
C	2.93	2.97	3.60	3.60	3.63	3.35
D	4.17	4.43	5.47	4.67	4.79	4.71
E	18.60	18.80	15.67	16.80	16.56	17.26
F	1.63	1.70	1.23	1.20	1.54	1.46
G	3.27	3.50	2.47	3.20	2.73	3.02
H	5.27	4.93	5.70	5.90	5.98	5.57
I	7.37	7.33	7.30	7.00	8.11	7.44
J	0.23	0.17	0.43	0.57	0.49	0.38
K	1.00	1.00	1.27	0.93	1.02	1.04
L	3.83	3.30	2.13	3.50	3.11	3.17
M	2.37	3.07	2.77	2.20	2.59	2.60
N	10.57	11.03	11.03	9.00	9.33	10.17
O	2.23	2.37	3.57	2.63	3.14	2.80
P	0.63	0.73	0.70	0.73	0.73	0.71
Q	0.07	0.03	0.00	0.03	0.00	0.03
R	7.03	6.40	6.73	6.23	6.80	6.64
S	6.80	6.63	7.50	7.43	7.93	7.28
T	5.37	5.13	5.67	7.13	6.30	5.93
U	4.43	4.77	4.53	4.47	3.95	4.42
V	0.93	0.70	0.83	0.80	0.46	0.74
W	1.63	2.10	1.83	1.63	1.51	1.74
X	0.00	0.10	0.00	0.00	0.00	0.02
Y	0.00	0.00	0.03	0.00	0.12	0.03
Z	1.43	1.07	1.17	1.07	1.19	1.19

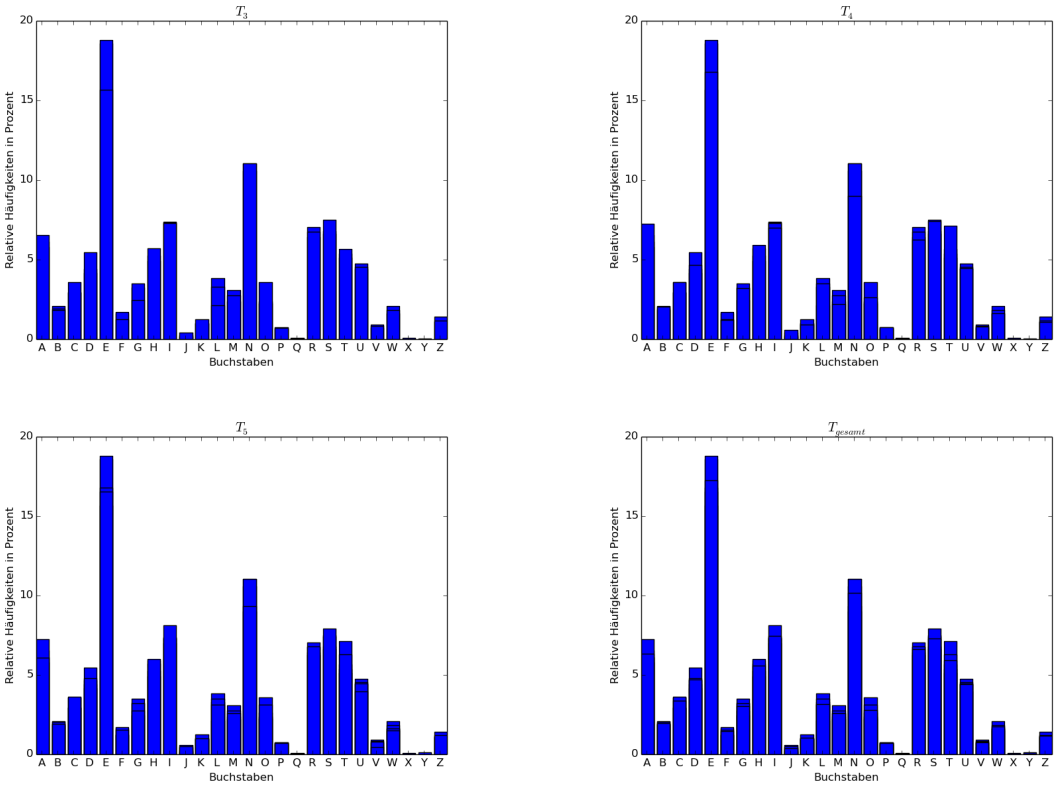
Wenn wir die Buchstaben der Häufigkeit nach ordnen, erhalten wir für die Teile $T_1, \ldots, T_5$ folgende Tabelle:

T_1	E	N	I	R	S	A	T	H	U	D	L	G	C	M	O	B	F	W	Z	K	V	P	J	Q	X	Y
T_2	E	N	I	S	R	A	T	H	U	D	G	L	M	C	O	W	B	F	Z	K	P	V	J	X	Q	Y
T_3	E	N	S	I	R	A	H	T	D	U	C	O	M	G	L	B	W	K	F	Z	V	P	J	Y	Q	X
T_4	E	N	S	A	T	I	R	H	D	U	C	L	G	O	M	B	W	F	Z	K	V	P	J	Q	X	Y
T_5	E	N	I	S	R	T	A	H	D	U	C	O	L	G	M	B	F	W	Z	K	P	J	V	Y	Q	X

Man sieht also, dass E der häufigste Buchstabe ist. Mit einigem Abstand folgt dann N.
Im Folgenden stellen wir die **Häufigkeitsverteilungen** auch graphisch in Form eines sogenannten **Häufigkeitsgebirges** dar.







Auch bei Auswertung anderer deutscher Texte wurde gefunden, dass E, dann N die häufigsten Buchstaben sind [**Bauer**, S.297]. Bei [**Bauer**, S.304] findet sich auch folgende Tabelle mit „hypothetischen Zeichenwahrscheinlichkeiten“ der englischen und deutschen Sprache, die durch Auswertung längerer Texte erstellt wurde:

Zeichen	englisch	deutsch	Zeichen	englisch	deutsch
A	8.04	6.47	N	7.09	9.84
B	1.54	1.93	O	7.60	2.98
C	3.06	2.68	P	2.00	0.96
D	3.99	4.83	Q	0.11	0.02
E	12.51	17.48	R	6.12	7.54
F	2.30	1.65	S	6.54	6.83
G	1.96	3.06	T	9.25	6.13
H	5.49	4.23	U	2.71	4.17
I	7.26	7.73	V	0.99	0.94
J	0.16	0.27	W	1.92	1.48
K	0.67	1.46	X	0.19	0.04
L	4.14	3.49	Y	1.73	0.08
M	2.53	2.58	Z	0.09	1.14

(Natürlich kann man nicht erwarten, dass sich jeder Text an eine solche Verteilung hält.)

Anwendung: Vermutet man, dass ein Text CAESAR-chiffriert ist, so bestimmt man das häufigste Zeichen c des Chiffretexts und dann die Translation K mit $f_K(\mathrm{E}) = c$. Anschließend testet man, ob sich durch Anwendung von f_{-K} auf den Chiffretext ein sinnvoller Text ergibt.

Beispiel: Wir vermuten, dass folgender Text Caesar-chiffriert ist:

```
RLQFNRBBWRLQCFJBBXUUNBKNMNDCNWMJBBRLQBXCAJDARPKRWNRWVJNALQNWJDBJUCNWIN
RCNWMJBTXVVCVRAWRLQCJDBMNVBRWW
```

Die häufigsten Buchstaben sind B und N, die jeweils 12-mal vorkommen. Es ist $f_{23}(\mathrm{E}) = \mathrm{B}$ und $f_9(\mathrm{E}) = \mathrm{N}$. Wendet man f_{-23} auf den Chiffretext an, erhält man die (nicht recht sinnvolle) Zeichenfolge

```
UOTIQUEEZUOTFIMEEAXXQENQPQGFQZPMEEUOTEAFDMGDUSNUZQUZYMQDOTQZMGEMXFQZLQ
UFQZPMEWAYYFYUDZUOTFMGEPQYEUZZ
```

Wendet man f_{-9} an, erhält man die sinnvolle Buchstabenfolge

```
ICHWEISSNICHTWASSOLLESBEDEUTENDASSICHSOTRAURIGBINEINMAERCHENAUSALTENZE
ITENDASKOMMTMIRNICHTAUSDEMSINN
```

Wir haben uns bisher nur für die Verteilung von Großbuchstaben interessiert. Werden auch andere Zeichen verschlüsselt, sollte man auch hier Häufigkeitsverteilungen studieren. Wir betrachten nur ein Beispiel:

Beispiel: Wir haben 5 TeX-Dateien betrachtet und die Zeichen gezählt. Die Häufigkeit ist jeweils in % angegeben. Wir haben nur die 10 häufigsten Zeichen aufgelistet, wobei Nr. 32 für Blank/Leerzeichen, Nr. 10 für Return/Wagenrücklauf steht.

1. Datei		2. Datei		3. Datei		4. Datei		5. Datei	
198407 Bytes		117364 Bytes		96471 Bytes		74909 Bytes		55790 Bytes	
(Nr. 32)	12.52	(Nr. 32)	8.19	(Nr. 32)	13.59	(Nr. 32)	10.53	(Nr. 32)	9.61
e	4.81	\	6.31	e	3.24	e	7.12	&	5.09
n	3.02	e	5.60	1	3.13	i	5.41	a	4.97
1	2.83	a	5.32	0	3.06	n	5.12	e	4.58
i	2.77	i	4.93	A	2.66	\	4.23	\	4.42
0	2.49	n	3.71	4	2.64	a	3.83	n	3.15
(Nr. 10)	2.39	$	3.50	9	2.59	$	3.65	i	3.00
2	2.35	t	3.42	3	2.53	t	3.50	l	2.63
3	2.21	r	3.29	8	2.44	r	3.10	s	2.62
9	2.18	{	2.86	D	2.40	(Nr. 10)	2.77	$	2.60

Mit Abstand ist also Nr. 32, Blank/Leerzeichen, das häufigste Zeichen.

Neben der Häufigkeitsverteilung einzelner Zeichen ist auch die Häufigkeitsverteilung von Buchstabenkombinationen, also von n-Grammen interessant. Ein Buchstabenpaar wird auch als Bigramm, ein Buchstabentripel als Trigramm bezeichnet.

Beispiel: Wir betrachten wieder den zuvor verwendeteten Text von Thomas Mann. Die folgenden Bigramme kommen mit einer Häufigkeit von mehr als einem Prozent vor:

EN	ER	CH	TE	ND	EI	IN	DE	GE	IE	ES	UN	NE	IC	SE	ST	BE	AN
3.82	3.62	2.99	2.17	2.14	1.94	1.85	1.78	1.60	1.51	1.45	1.42	1.26	1.24	1.24	1.22	1.15	1.11

Dies sind die häufigsten Trigramme:

ICH	EIN	UND	SCH	DER	NDE	INE	END	CHT	GEN	CHE	ENS	TEN	TER	DIE	ERS	DEN
1.20	1.04	0.82	0.81	0.78	0.67	0.60	0.59	0.56	0.55	0.54	0.49	0.48	0.47	0.45	0.40	0.40

Die 10 häufigsten 4-Gramme (Tetragramme) sind (In den Klammern stehen die absoluten Häufigkeiten):

```
EINE (83), ICHT (61), SICH (52), SEIN (48), NICH (45), LICH (44), NDER (42), CHEN (40),
SCHE (36), ENDE (36)
```

Bei [**Bauer**, S.308] findet sich eine Liste von häufigen Bigrammen im Deutschen. Als die zehn häufigsten deutschen Wörter nennt [**Bauer**, S.309]

die, der, und, den, am, in, zu, ist, dass, es.

Bei [**Bauer**] finden sich noch viele weitere sprachliche Eigenschaften zusammengestellt. Dies kann helfen, eine monoalphabetische Substitutionschiffrierung zu entschlüsseln.

4. Einfache Transpositionschiffren

Bei einfachen Transpositionschiffren werden die Zeichen nur an eine andere Stelle gesetzt. Es gibt vielfältige Möglichkeiten dafür. Wir begnügen uns mit zwei Beispielen.

TRANSMAT-Verschlüsselung:

(1) Es werden nur die Großbuchstaben A,...,Z verschlüsselt.
(2) **Schlüssel:** Der Schlüssel besteht aus zwei natürlichen Zahlen m und n.
(3) **Verschlüsselung:**
   (a) Der Text wird in Blöcke der Länge mn aufgeteilt. Ist die Zeichenanzahl des Texts nicht durch mn teilbar, werden am Schluss willkürlich Buchstaben angehängt.
   (b) Je mn Zeichen eines Blocks des Ausgangstexts werden zeilenweise in eine $m \times n$-Matrix geschrieben. Der Chiffretext entsteht, indem die Matrix spaltenweise ausgegeben wird.
(4) **Entschlüsselung:** Bei der Entschlüsselung muss man jeden Block (aus mn Zeichen) des Chiffretexts spaltenweise in eine $m \times n$-Matrix schreiben, dann zeilenweise auslesen. (Alternativ: Man schreibt den Chiffretext zeilenweise in eine $n \times m$-Matrix. Den Klartext erhält man durch spaltenweises Auslesen.)

Beispiel: Mit $m = 3$ und $n = 4$ wird der Text 'DEPARTMENT MATHEMATIK' zeilenweise in 3×4-Matrizen geschrieben, am Schluss 4 Zeichen (WXYZ) ergänzt:

```
DEPA  THEM
RTME  ATIK
NTMA  WXYZ
```

Der Chiffretext ist dann 'DRNETTPMMAEATAWHTXEIYMKZ'.

Bemerkung: Es ist klar, dass jeder einzelne Buchstabe im Klartext genauso oft wie im Chiffretext vorkommt. Klartext und Chiffretext haben also die gleiche Häufigkeitsverteilung (einzelner Buchstaben).

TRANSSPA-Verschlüsselung:

(1) **Schlüssel:**
   (a) Der Schlüssel besteht aus einem Wort. Es bestehe aus n Buchstaben.
   (b) Man legt eine Tabelle/Matrix mit n Spalten an und schreibt das Schlüsselwort über die Tabelle/Matrix, sodass über jeder Spalte ein Buchstabe steht.
   (a') Alternativ kann man als Schlüssel auch eine natürliche Zahl n und eine Permutation π der Zahlen von 1 bis n wählen: $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$. Man schreibt dann die Zahl i über die Spalte mit der Nummer $\pi(i)$.
(2) **Verschlüsselung:**
   (a) Der zu verschlüsselnde Text wird zeilenweise in die Tabelle/Matrix (mit n Spalten) geschrieben.
   (b) Die Buchstaben werden nun spaltenweise ausgegeben, wobei die Reihenfolge durch die Buchstaben des Schlüsselworts gegeben wird; bei gleichen Buchstaben geht es von links nach rechts. Dies liefert den Chiffretext.
   (c') Wurde der Schlüssel durch eine Permutation angegeben, wird zuerst die Spalte, über der 1 steht ausgegeben, dann die Spalte, über der 2 steht, etc., also zuerst die $\pi(1)$-te Spalte, dann die $\pi(2)$-te Spalte, etc.
(3) **Entschlüsselung:**
   (a) Hat das Schlüsselwort n Zeichen und der Chiffretext N Zeichen, so zerlegt man zunächst $N = mn + r$ mit $m = \lfloor \frac{N}{n} \rfloor$ und $r = N \bmod n$. Nun legt man eine Tabelle/Matrix mit n Spalten an; die ersten r Spalten werden $m + 1$ Zeichen, die letzten Spalten nur m Zeichen enthalten. Man schreibt das Schlüsselwort über die Tabelle/Matrix.
   (b) In der durch das Schlüsselwort bestimmten Reihenfolge wird nun der Chiffretext spaltenweise in die Matrix geschrieben, wobei zu beachten ist, wieviele Zeichen die Spalte enthält.
   (c) Schreibt man die Matrix nun zeilenweise aus, erhält man den Ausgangstext.

Beispiel: (TRANSSPA-Verschlüsselung) Als Schlüsselwort wählen wir „ERLANGEN“. Dazu legen wir eine Tabelle an:

E	R	L	A	N	G	E	N

Zuerst wird dann die Spalte mit der Bezeichnung A ausgegeben, ... Wir schreiben die Reihenfolge ebenfalls über die Spalten:

E	R	L	A	N	G	E	N
2	8	5	1	6	4	3	7

Wir wollen den Text „AMFREITAGENTFAELLTDIEVORLESUNG“ verschlüsseln. Wir schreiben ihn zeilenweise in die angelegte Tabelle:

E	R	L	A	N	G	E	N
2	8	5	1	6	4	3	7
A	M	F	R	E	I	T	A
G	E	N	T	F	A	E	L
L	T	D	I	E	V	O	R
L	E	S	U	N	G		

Den Chiffretext erhalten wir durch spaltenweises Ausschreiben des Text, wobei die Reihenfolge der Spalten durch die darüberstehenden Zahlen gegeben wird: „RTIUAGLLTEOIAVGFNDSEFENALRMETE“.

Beispiel: (TRANSSPA-Entschlüsselung) „ESASENIRISAOTOLPNTTPNN“ ist ein mit dem Schlüsselwort „NUERNBERG“ TRANSSPA-verschlüsselter Text. Zur Entschlüsselung legen wir wieder eine Tabelle an:

N	U	E	R	N	B	E	R	G
5	9	2	7	6	1	3	8	4

Der Chiffretext hat 22 Zeichen, das Schlüsselwort hat 9 Zeichen. Wir zerlegen $22 = 2 \cdot 9 + 4$. Daher enthalten die ersten 4 Spalten 3 Zeichen, die restlichen nur 2 Zeichen. Wir bereiten dies bereits vor:

N	U	E	R	N	B	E	R	G
5	9	2	7	6	1	3	8	4
				x	x	x	x	x

Nun schreiben wir den Chiffretext spaltenweise in die Matrix, wobei mit der Spalte mit der Nummer 1 begonnen wird:

N	U	E	R	N	B	E	R	G
5	9	2	7	6	1	3	8	4
S	P	A	L	T	E	N	T	R
A	N	S	P	O	S	I	T	I
O	N	E	N	x	x	x	x	x

Zeilenweises Auslesen ergibt den Klartext „SPALTENTRANSPOSITIONEN“.

Beispiel: (TRANSSPA-Entschlüsselung) Mit dem Schlüsselwort „MATHEMATIK“ wurde ein Text zu „EIIGSNASEGTATAHNFEHRSRHESEIUNUUTEETDC“ TRANSSPA-verschlüsselt. Das Schlüsselwort hat 10 Buchstaben, der Chiffretext 37 Buchstaben. Wegen $37 = 3 \cdot 10 + 7$ legen wir eine Matrix mit 10 Spalten an, wobei die ersten 7 Spalten 4 Zeichen, die letzten 3 Spalten nur 3 Zeichen enthalten sollen. Wir schreiben das Schlüsselwort über die Matrix und nummerieren die Spalten entsprechend der

Reihenfolge der Buchstaben des Schlüsselworts:

M	A	T	H	E	M	A	T	I	K
7	1	9	4	3	8	2	10	5	6
							x	x	x

Nun schreiben wir den Chiffretext spaltenweise in die Matrix, beginnende mit der Spalte mit Nummer 1:

M	A	T	H	E	M	A	T	I	K
7	1	9	4	3	8	2	10	5	6
H	E	U	T	E	I	S	T	F	R
E	I	T	A	G	U	N	D	E	S
S	I	E	H	T	N	A	C	H	R
E	G	E	N	A	U	S	x	x	x

Als Klartext ergibt sich „HEUTE IST FREITAG UND ES SIEHT NACH REGEN AUS“.

5. Blockchiffren

Eine Verallgemeinerung von Substitutions- und Transpositionschiffren sind Blockchiffren:

Blockchiffrierung: Wir fassen jeweils k Zeichen eines Textes zu einem Block bzw. einer Nachrichteneinheit zusammen, d.h. wir schreiben den Text T als Folge $T_1T_2T_3T_4\ldots$, wobei T_i aus k Zeichen/Buchstaben des Alphabets Σ besteht. Die Verschlüsselungsfunktionen f_K sollen jetzt zunächst auf der Menge der Blöcke bzw. Nachrichteneinheiten wirken, d.h. wir haben bijektive Funktionen $f_K : \Sigma^k \to \Sigma^k$. Man spricht dann von einer Blockchiffrierung oder einer Blockchiffre.

Wir geben ein mathematisch einfaches Beispiel einer Blockchiffrierung.

ALBC-2-Verschlüsselung: (Affin-lineare Blockchiffrierung mit Blocklänge 2)

(1) Nur Großbuchstaben werden verschlüsselt. Diese werden mit den Zahlen $0,\ldots,25$ identifiziert.

(2) **Schlüssel:** Ein Schlüssel k ist ein 6-Tupel

$$k = (k_1, k_2, k_3, k_4, k_5, k_5) \text{ mit } 0 \le k_i \le 25 \text{ und } \mathrm{ggT}(k_1k_5 - k_2k_4, 26) = 1.$$

Die zugehörige Verschlüsselungsabbildung f_k operiert auf Blöcken der Länge 2:

$$f_k(x, y) = (k_1x + k_2y + k_3 \bmod 26, k_4x + k_5y + k_6 \bmod 26).$$

Identifizieren wir den den Block (x, y) mit dem Vektor $\begin{pmatrix} x \\ y \end{pmatrix}$, so können wir die Verschlüsselungsabbildung f_k auch schreiben als

$$f_k(\begin{pmatrix} x \\ y \end{pmatrix}) = \begin{pmatrix} k_1 & k_2 \\ k_4 & k_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} k_3 \\ k_6 \end{pmatrix} \bmod 26.$$

(Die ggT-Bedingung ist äquivalent mit der Bijektivität von f_K.)

(3) **Inverser Schlüssel:** Die ggT-Bedingung $\mathrm{ggT}(k_1k_5 - k_2k_4, 26) = 1$ impliziert, dass ein $d \in \mathbb{Z}$ existiert mit

$$d(k_1k_5 - k_2k_4) \bmod 26 = 1,$$

das beispielsweise durch Ausprobieren gefunden werden kann. Definiert man dann

$$\begin{aligned}
\ell_1 &= dk_5 \bmod 26,\\
\ell_2 &= (-dk_2) \bmod 26,\\
\ell_3 &= d(k_2k_6 - k_3k_5) \bmod 26,\\
\ell_4 &= (-dk_4) \bmod 26,\\
\ell_5 &= dk_1 \bmod 26,\\
\ell_6 &= d(k_3k_4 - k_1k_6) \bmod 26,
\end{aligned}$$

so gilt für

$$\ell = (\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, \ell_6) \quad \text{die Beziehung} \quad f_k^{-1} = f_\ell.$$

ℓ ist der zu k **inverse Schlüssel**.

(4) **Verschlüsselung:** Der Ausgangstext wird in Blöcke der Länge $k = 2$ unterteilt, wobei am Schluss eventuell ein Zeichen zu ergänzen ist. Der Text ist also $x_1y_1x_2y_2x_3y_3\ldots$. Mit

$$\widetilde{x}_i = k_1x_i + k_2y_i + k_3 \bmod N, \quad \widetilde{y}_i = k_4x_i + k_5y_i + k_6 \bmod N$$

wird der Chiffretext zu $\widetilde{x}_1\widetilde{y}_1\widetilde{x}_2\widetilde{y}_2\widetilde{x}_3\widetilde{y}_3\ldots$

(5) **Entschlüsselung:** Die Entschlüsselung funktioniert wie die Verschlüsselung, nur dass man statt des Schlüssels $k = (k_1, \ldots, k_6)$ den inversen Schlüssel $\ell = (\ell_1, \ldots, \ell_6)$ benutzt:

$$x_i = \ell_1\widetilde{x}_i + \ell_2\widetilde{y}_i + \ell_3 \bmod 26, \quad y_i = \ell_4\widetilde{x}_i + \ell_5\widetilde{y}_i + \ell_6 \bmod 26.$$

Beispiel: Wir betrachten $k = (5, 8, 17, 23, 15, 14)$. Es ist

$$k_1k_5 - k_2k_4 = 5 \cdot 15 - 8 \cdot 23 = -109 = -5 \cdot 26 + 21.$$

Durch Probieren findet man, dass $d = 5$ die Bedingung $d(k_1k_5 - k_2k_4) \bmod 26 = 1$ erfüllt. Damit kann man gleich den inversen Schlüssel berechnen:

$$\begin{aligned}
\ell_1 &= dk_5 \bmod 26 = 75 \bmod 26 = 23,\\
\ell_2 &= (-dk_2) \bmod 26 = (-40) \bmod 26 = 12,\\
\ell_3 &= d(k_2k_6 - k_3k_5) \bmod 26 = (-715) \bmod 26 = 13,\\
\ell_4 &= (-dk_4) \bmod 26 = (-115) \bmod 26 = 15,\\
\ell_5 &= dk_1 \bmod 26 = 25 \bmod 26 = 25,\\
\ell_6 &= d(k_3k_4 - k_1k_6) \bmod 26 = 1605 \bmod 26 = 19.
\end{aligned}$$

Der inverse Schlüssel ist also $\ell = (23, 12, 13, 15, 25, 19)$.
Wir wollen „ZAHLENTHEORIE“ verschlüsseln. Wir hängen ein X an, damit die Zeichenzahl gerade ist, wandeln den Text in eine Zahlenfolge $x_1y_1\ldots x_7y_7$ um, berechnen

$$\widetilde{x}_i = k_1x_i + k_2y_i + k_3 \bmod 26, \quad \widetilde{y}_i = k_4x_i + k_5y_i + k_6 \bmod 26$$

und wandeln $\widetilde{x}_1\widetilde{y}_1\ldots\widetilde{x}_7\widetilde{y}_7$ wieder in eine Buchstabenfolge um:

ZA	(12, 17)	(25, 0)	MR
HL	(10, 2)	(7, 11)	KC
EN	(11, 15)	(4, 13)	LP
TH	(12, 10)	(19, 7)	MK
EO	(19, 4)	(4, 14)	TE
RI	(10, 5)	(17, 8)	KF
EX	(13, 9)	(4, 23)	NJ

Der Chiffretext ist also „MRKCLPMKTEKFNJ“.

Bemerkungen:

(1) Wenn wir später die Kongruenzrechnung ausführlich behandeln, werden wir sehen, woher die ggT-Bedingung kommt und wie sich der inverse Schlüssel aus dem Schlüssel berechnet.

(2) Natürlich verallgemeinert sich ALBC-2 sofort auf Alphabete mit N Zeichen.

(3) Verwendet man ALBC-2 mit einem N-elementigen Alphabet, das mit den Zahlen $0, \ldots, N-1$ identifiziert wird, so ist die Menge der Schlüssel

$$\{(k_1, k_2, k_3, k_4, k_5, k_6) : 0 \le k_i \le N-1, \mathrm{ggT}(N, k_1k_5 - k_2k_4) = 1\}.$$

Man kann zeigen, dass die Anzahl der Schlüssel

$$N^6 \prod_{p|N} (1 - \frac{1}{p})(1 - \frac{1}{p^2})$$

ist. Für $N = 26$ ergibt sich die Zahl 106299648.

Bemerkung: Wir nehmen an, wir kennen den Chiffretext einer ALBC-2-Verschlüsselung, also $\widetilde{x}_1\widetilde{y}_1\widetilde{x}_2\widetilde{y}_2\ldots$. Außerdem kennen wir oder raten wir Teile des Klartexts, beispielsweise $x_1y_1x_2y_2x_3y_3$. Dann erhalten wir (mit obigen Bezeichnungen) Gleichungen

$$\begin{aligned} x_1 &\equiv \ell_1\widetilde{x}_1 + \ell_2\widetilde{y}_1 + \ell_3 \bmod 26, & y_1 &\equiv \ell_4\widetilde{x}_1 + \ell_5\widetilde{y}_1 + \ell_6 \bmod 26,\\ x_2 &\equiv \ell_1\widetilde{x}_2 + \ell_2\widetilde{y}_2 + \ell_3 \bmod 26, & y_2 &\equiv \ell_4\widetilde{x}_2 + \ell_5\widetilde{y}_2 + \ell_6 \bmod 26,\\ x_3 &\equiv \ell_1\widetilde{x}_3 + \ell_2\widetilde{y}_3 + \ell_3 \bmod 26, & y_3 &\equiv \ell_4\widetilde{x}_3 + \ell_5\widetilde{y}_3 + \ell_6 \bmod 26. \end{aligned}$$

Dies sind 6 Gleichungen mit 6 unbekannten Zahlen $\ell_1,\ldots,\ell_6$. Man kann hoffen, dass man diese Gleichungen lösen kann. (Bei SAGE gibt es eine Funktion `solve_mod`.) Daher ist ALBC-2 ziemlich unsicher.

Eine mathematisch naheliegende Verallgemeinerung des letzten Chiffrierverfahrens ist das folgende Verfahren, das wir nur kurz skizzieren.

ALBC-k-Verschlüsselung: (Affin-lineare Blockchiffrierung mit Blocklänge k) Man lege ein Alphabet mit N Zeichen zugrunde, das mit den Zahlen zwischen 0 und $N-1$ identifiziert wird, fasse jeweils k Zeichen $x_1x_2\ldots x_k$ zu einer Nachrichteneinheit zusammen, bilde daraus einen Vektor und definiere die Verschlüsselungsabbildung durch

$$\begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} \mapsto A \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix} \bmod N$$

mit einer ganzzahligen Matrix A und ganzen Zahlen $b_1,\ldots,b_k$, so dass $\mathrm{ggT}(\det A, N) = 1$ ist. (Die ggT-Bedingung sichert die Bijektivität der Verschlüsselungsabbildungen.)

Bemerkung: Für die Praxis sind heutzutage andere Blockchriffrierungen interessant, wovon hier einige genannt seien.

(1) Auf Ubuntu-Rechnern steht die freie Software GnuPG 2.4.4 (von 2024) zur Verfügung. Damit kann man folgende Verfahren zur Verschlüsselung von Dateien verwenden. Als Alphabet liegt $\Sigma = \{0,1\}$ zugrunde (Bits).

Chiffre	Blocklänge	Schlüssellänge
IDEA	64	128
3DES	64	56
CAST5	64	40-128
BLOWFISH	64	128
AES	128	128
AES192	128	192
AES256	128	256
TWOFISH	128	128, 192, 256
CAMILLIA128	128	128
CAMILLIA192	128	192
CAMILLIA256	128	256

(`gpg -c --cipher-algo CAMELLIA256 file` verschlüsselt die Datei `file` zu `file.gpg`.)

(2) Ein US-amerikanisches Standard-Verschlüsselungsverfahren ist AES (Advanced Encryption Standard) mit Blocklänge 128, Alphabet $\{0,1\}$, Schlüssellängen 128, 192 oder 256 FIPS 197 (Published November 26, 2001; Updated May 9, 2023).

Betriebsarten: Blockchiffrierungen operieren auf Blöcken der Länge k, mathematisch formuliert: Ist Σ das zugrundeliegende Alphabet, so hat man Funktionen

$$E_K, D_K : \Sigma^k \to \Sigma^k \quad \text{mit } D_K \circ E_K = id$$

in Abhängigkeit von einem Parameter, dem Schlüssel K. Man kann nun Blockchiffrierungen in verschiedener Weise zum Verschlüsseln ganzer Texte einsetzen, wovon wir zwei Verfahren skizzieren wollen:

(1) **ECB-Modus** (Electronic-Codebook-Modus): Dies ist die einfachste Art, eine Blockchiffrierung auf einen Text T anzuwenden: Man teilt den Text in Blöcke der Länge k ein:

$$T = a_1a_2a_3a_4\ldots,$$

dann verschlüsselt man den i-ten Block zu $b_i = E_K(a_i)$ und erhält den verschlüsselten Text $b_1b_2b_3b_4\ldots$. Die Entschlüsselung erfolgt durch $a_i = D_K(b_i)$. (Ein Nachteil des ECB-Modus ist, dass Regelmäßigkeiten im Klartext zu Regelmäßigkeiten im verschlüsselten Text führen.)

(2) **CBC-Modus** (Cipherblock-Chaining-Modus): Wir setzen als Alphabet $\Sigma \simeq \{0, 1, \ldots, N-1\}$ voraus, teilen den Text wieder in Blöcke der Länge k, also $T = a_1a_2a_3a_4\ldots$, wobei wir uns $a_i \in \Sigma^k$ als Vektor der Länge k vorstellen. Wir wählen einen sogenannten Initialisierungsvektor $b_0 \in \Sigma^k$ der Länge k und berechnen

$$b_i = E_K(b_{i-1} + a_i) \quad \text{für} \quad i \geq 1,$$

wobei $b_{i-1} + a_i$ für die komponentenweise Addition modulo N steht. Der verschlüsselte Text wird $b_0b_1b_2b_3b_4\ldots$. Wegen

$$D_K(b_i) = D_K(E_K(b_{i-1} + a_i) = b_{i-1} + a_i$$

erhält man mit der Formel

$$a_i = D_K(b_i) - b_{i-1} \pmod N$$

den Ausgangstext. (Verschiedene Initialisierungsvektoren führen bei gleichem Schlüssel zu verschiedenen Verschlüsselungen.)

Padding: Was tut man, wenn die Zeichenzahl des Ausgangstextes nicht durch die Blocklänge k teilbar ist, d.h. wenn der letzte Block weniger als k Zeichen hat? Dann kann die Verschlüsselungsfunktion E_K auf den letzten Block nicht angewandt werden. Eine Möglichkeit besteht im sogenannten **Auffüllen** (**padding**), was wir hier für Blocklängen $k \leq 255$ beschreiben wollen. Wir nehmen an, wir haben eine Datei, die aus n Bytes besteht. Wir zerlegen $n = \ell k + r$ mit $0 \leq r < k$, sodass also r die Anzahl der Bytes im unvollständigen letzten Block bzw. 0 ist. Wir ergänzen jetzt den letzten Block durch $k - r - 1$ beliebige Bytes und hängen dann noch ein Byte an, das die Zahl $k - r$ enthält. Der vervollständigte Text kann jetzt verschlüsselt werden. Nach dem Entschlüsseln gibt das letzte Byte dann an, wieviele Bytes zu streichen sind, damit man den Ausgangstext wieder erhält.

6. Stromchiffren

Hier soll kurz die Idee von Stromchiffren skizziert werden.

STROM-Chiffrierung:

(1) Es werden nur Großbuchstaben berücksichtigt.

(2) **Schlüssel:** Der Schlüssel besteht aus einer Folge von Großbuchstaben $k_1, k_2, k_3, \ldots$. (Die Folge sollte mindestens so lang sein wie der zu verschlüsselnde Text.)

(3) Es werden Funktionen $f, g : \{\mathrm{A}, \ldots, \mathrm{Z}\} \times \{\mathrm{A}, \ldots, \mathrm{Z}\} \to \{\mathrm{A}, \ldots, \mathrm{Z}\}$ vereinbart mit der Eigenschaft, dass $g(f(x, y), y) = x$ für alle Buchstaben x, y gilt. Identifiziert man die Großbuchstaben A,B,...,Z mit den Zahlen 0,1,...,25, so kann man beispielsweise

$$f(x, y) = x + y \bmod 26, \quad g(x, y) = x - y \bmod 26$$

wählen. Eine zugehörige Tabelle ist unten angegeben.

(4) **Verschlüsselung:** Der zu verschlüsselnde Text $a_1, a_2, a_3, \ldots$ wird mit der Schlüsselfolge $k_1, k_2, k_3, \ldots$ mittels $b_i = f(a_i, k_i)$ in den Chiffretext $b_1, b_2, b_3, \ldots$ umgewandelt:

Text:	a_1	a_2	a_3	a_4	a_5	$\ldots$
Schlüssel:	k_1	k_2	k_3	k_4	k_5	$\ldots$
Chiffretext:	$f(a_1, k_1)$	$f(a_2, k_2)$	$f(a_3, k_3)$	$f(a_4, k_4)$	$f(a_5, k_5)$	$\ldots$

(5) **Entschlüsselung:** Aus dem Chiffretext $b_1, b_2, b_3, \ldots$ und der Schlüsselfolge $k_1, k_2, k_3, \ldots$ erhält man den Ausgangstext $a_1, a_2, a_3, \ldots$ durch Bestimmung von $a_i = g(b_i, k_i)$:

Chiffretext:	b_1	b_2	b_3	b_4	b_5	$\ldots$
Schlüssel:	k_1	k_2	k_3	k_4	k_5	$\ldots$
Text:	$g(b_1, k_1)$	$g(b_2, k_2)$	$g(b_3, k_3)$	$g(b_4, k_4)$	$g(b_5, k_5)$	$\ldots$

(Identifiziert man die Großbuchstaben mit den Zahlen $0, 1, \ldots, 25$, so ist die Entschlüsselung die Verschlüsselung des Chiffretexts mit der Schlüsselfolge $-k_1, -k_2, -k_3, \ldots$.)

$f(x,y)$	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Beispiel: Verschlüsselt werden soll der Text „HEUTE IST EIN SCHOENER TAG“. Als Schlüsselfolge nehmen wir das Gedicht „Die Bürgschaft“ von Schiller:

Text	H	E	U	T	E	I	S	T	E	I	N	S	C	H	O	E	N	E	R	T	A	G
Schlüssel	Z	U	D	I	O	N	Y	S	D	E	M	T	Y	R	A	N	N	E	N	S	C	H
Chiffretext	G	Y	X	B	S	V	Q	L	H	M	Z	L	A	Y	O	R	A	I	E	L	C	N

Legt man als Alphabet $\Sigma = \{0, 1\}$ zugrunde, so erhält man folgende Variante:

Vernam-Cipher oder **One-Time-Pad:**

(1) Als Alphabet liegt $\{0, 1\}$ zugrunde, d.h. man muss sich auf Methoden einigen, wie man Texte in 0-1-Folgen umwandelt.
(2) **Schlüssel:** Der Schlüssel besteht aus einer 0-1-Folge $k_1, k_2, k_3, \ldots$.
(3) **Verschlüsselung:** Der zu verschlüsselnde Text $a_1, a_2, a_3, \ldots$ wird mit der Schlüsselfolge $k_1, k_2, k_3, \ldots$ zu $b_1, b_2, b_3, \ldots$ mit
$$b_i = a_i + k_i \bmod 2$$
verschlüsselt.
(4) **Entschlüsselung:** Man erhält die Klartextfolge $a_1, a_2, a_3, \ldots$ aus $a_i = b_i + k_i \bmod 2$.

Bemerkung: Diese Verschlüsselung gilt als sicher, wenn die Schlüsselfolge eine zufällige 0-1-Folge ist. Die Schlüsselfolge sollte nur einmal benutzt werden.

Bemerkung: Es gibt verschiedene Varianten, aus einem Schlüsselwort eine längere Schlüsselfolge zu machen. Beispielhaft werden wir die **Autokey-Chiffrierung** und später die **Vigenère-Chiffrierung** betrachten.

AUTOKEY-Verschlüsselung: Als Schlüssel ist zunächst ein Schlüsselwort $k_1k_2\dots k_n$ gegeben. Soll der Text $a_1a_2a_3\dots$ verschlüsselt werden, hängt man den Text an das Schlüsselwort um eine Schlüsselfolge zu erhalten:

$$k_1, k_2, k_3, \dots, k_n \text{ und } k_{n+i} = a_i \text{ für } i \geq 1.$$

Dann strom-chiffriert man den Text mit dem so konstruierten Schlüsselstrom.

Beispiel: Verschlüsselt werden soll „WINTERSEMESTER“, als Schlüsselwort wird „FREITAG“ gewählt:

Text	W	I	N	T	E	R	S	E	M	E	S	T	E	R
Schlüssel	F	R	E	I	T	A	G	W	I	N	T	E	R	S
Chiffretext	B	Z	R	B	X	R	Y	A	U	R	L	X	V	J

Bemerkung: Wählt man zum Verschlüsseln die Vorschrift

$$b_i = -a_i - k_i \bmod N,$$

(statt $b_i = a_i + k_i \bmod N$), so funktioniert wegen $a_i = -b_i - k_i \bmod N$ das Entschlüsseln wie das Verschlüsseln.

7. Angriffe - Kryptanalyse

Man verwendet Kryptographie um Informationen vor Unbefugten zu schützen. Jemand, der unbefugt an verschlüsselte Informationen kommen will, wird auch als Angreifer, Gegner oder Feind bezeichnet, die zugehörige Tätigkeit als Angriff oder Attacke. Ein paar mögliche Situationen sind folgende:

- Ciphertext-only-Angriff: Der Angreifer hat nur einen verschlüsselten Text zur Verfügung und will diesen entschlüsseln.
- Known-plaintext-Angriff: Der Gegner kennt einen Ausgangstext (oder Teile davon) und den zugehörigen verschlüsselten Text. Er will Verschlüsselungsverfahren und Schlüssel finden um andere verschlüsselte Texte zu entschlüsseln.
- Chosen-plaintext-Angriff: Der Angreifer kann Ausgangstexte wählen und sehen, wie sie verschlüsselt werden. Er will das zugehörige Verfahren mit Schlüssel finden.

Wird ein Kryptosystem von einer größeren Anzahl von Personen genutzt, kann man nicht darauf vertrauen, dass das Verfahren geheim bleibt. Die Sicherheit hängt dann wesentlich von der Geheimhaltung des Schlüssels ab (Maxime von Kerckhoff). Erfahrungsgemäß ist ein Kryptosystem dann am sichersten, wenn der Algorithmus allgemein bekannt ist und von vielen Leuten getestet wurde.

Die systematische Untersuchung der Sicherheit von Kryptosystemen wird als Kryptanalyse oder auch als Kryptoanalyse bezeichnet.

Bemerkung: Verwendet man eines der vorgestellten Kryptosysteme, so muss man natürlich vorher auf anderem Wege, der aber trotzdem sicher sein muss, den Schlüssel ausgetauscht haben. Dies ist vor allem bei einer großen Zahl von Teilnehmern problematisch.

Literatur:

- [**Bauer**] F. L. Bauer. Entzifferte Geheimnisse. Dritte, überarbeitete und erweiterte Auflage. Springer, 2000.