Vorlesungsskript

Kryptologie 1

Wintersemester 07/08

Prof. Dr. Johannes Köbler
Humboldt-Universität zu Berlin
Lehrstuhl Komplexität und Kryptografie

5. November 2007

Inhaltsverzeichnis

1 Klassische Verfahren

1.1 Einführung

Kryptosysteme (Verschlüsselungsverfahren) dienen der Geheimhaltung von Nachrichten bzw. Daten. Hierzu gibt es auch andere Methoden wie z.B.

Physikalische Maßnahmen: Tresor etc.

Organisatorische Maßnahmen: einsamer Waldspaziergang etc.

Steganographische Maßnahmen: unsichtbare Tinte etc.

Andererseits können durch kryptographische Verfahren weitere **Schutzziele** realisiert werden.

- *Vertraulichkeit*
  - Geheimhaltung
  - Anonymität (z.B. Mobiltelefon)
  - Unbeobachtbarkeit (von Transaktionen)
- *Integrität*
  - von Nachrichten und Daten
- *Zurechenbarkeit*
  - Authentikation
  - Unabstreitbarkeit
  - Identifizierung
- *Verfügbarkeit*
  - von Daten
  - von Rechenressourcen

  - von Informationsdienstleistungen

In das Umfeld der Kryptographie fallen auch die folgenden Begriffe.

Kryptographie: Lehre von der Geheimhaltung von Informationen durch die Verschlüsselung von Daten. Im weiteren Sinne: Wissenschaft von der Übermittlung, Speicherung und Verarbeitung von Daten in einer von potentiellen Gegnern bedrohten Umgebung.

Kryptoanalysis: Erforschung der Methoden eines unbefugten Angriffs gegen ein Kryptoverfahren (Zweck: Vereitelung der mit seinem Einsatz verfolgten Ziele)

Kryptoanalyse: Analyse eines Kryptoverfahrens zum Zweck der Bewertung seiner kryptographischen Stärken bzw. Schwächen.

Kryptologie: Wissenschaft vom Entwurf, der Anwendung und der Analyse von kryptographischen Verfahren (umfasst Kryptographie und Kryptoanalyse).

1.2 Kryptosysteme

Es ist wichtig, Kryptosysteme von Codesystemen zu unterscheiden.

Codesysteme

- operieren auf semantischen Einheiten,
- starre Festlegung, welche Zeichenfolge wie zu ersetzen ist.

Beispiel 1.1 (Ausschnitt aus einem Codebuch der deutschen Luftwaffe)

`xve`	**Bis auf weiteres Wettermeldung gemäß Funkbefehl testen**
`yde`	**Frage**
`sLk`	**Befehl**
`fin`	**beendet**
`eom`	**eigene Maschinen**

Kryptosysteme

- operieren auf syntaktischen Einheiten,
- flexibler Mechanismus durch Schlüsselvereinbarung

Definition 1.2 (Alphabet) *Ein **Alphabet** $A = \{a_0, \ldots, a_{m-1}\}$ ist eine geordnete endliche Menge von **Zeichen** a_i. Eine Folge $x = x_1 \ldots x_n \in A^n$ heißt **Wort** (der **Länge** n). Die Menge aller Wörter über dem Alphabet A ist $A^* = \bigcup_{n \geq 0} A^n$.*

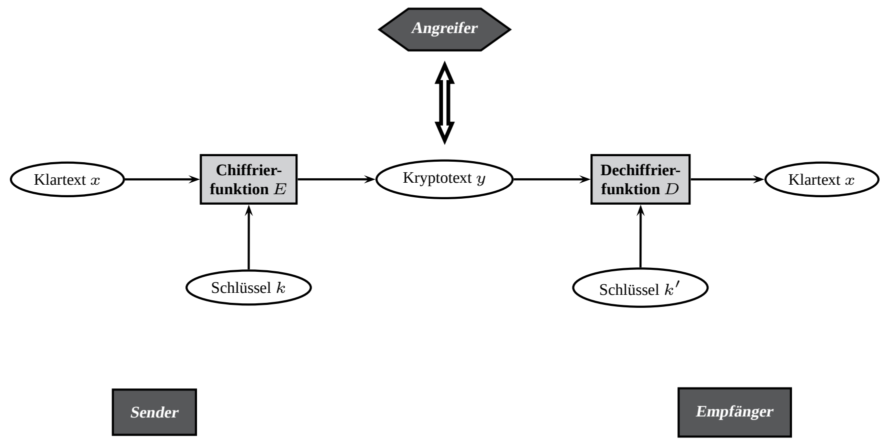
Beispiel 1.3 Das ***lateinische Alphabet*** A_{lat} enthält die 26 Buchstaben `A,...,Z`. Bei der Abfassung von Klartexten wurde meist auf den Gebrauch von Interpunktions- und Leerzeichen sowie auf Groß- und Kleinschreibung verzichtet ($\leadsto$ Verringerung der Redundanz im Klartext).

Definition 1.4 (Kryptosystem) *Ein **Kryptosystem** wird durch folgende Komponenten beschrieben:*

- *A, das **Klartextalphabet**,*
- *B, das **Kryptotextalphabet**,*
- *K, der **Schlüsselraum** (key space),*
- *$M \subseteq A^*$, der **Klartextraum** (message space),*
- *$C \subseteq B^*$, der **Kryptotextraum** (ciphertext space),*
- *$E : K \times M \to C$, die **Verschlüsselungsfunktion** (encryption function),*
- *$D : K \times C \to M$, die **Entschlüsselungsfunktion** (decryption function) und*
- *$S \subseteq K \times K$, eine Menge von Schlüsselpaaren (k, k') mit der Eigenschaft, dass für jeden Klartext $x \in M$ folgende Beziehung gilt:*

$$D(k', E(k, x)) = x \tag{1.1}$$

Bei symmetrischen Kryptosystemen ist $S = \{(k, k) \mid k \in K\}$, weshalb wir in diesem Fall auf die Angabe von S verzichten können.



Zu jedem Schlüssel $k \in K$ korrespondiert also eine **Chiffrierfunktion** $E_k : x \mapsto E(k, x)$ und eine **Dechiffrierfunktion** $D_k : y \mapsto D(k, y)$. Die Gesamtheit dieser Abbildungen wird auch **Chiffre** (englisch *cipher*) genannt. (Daneben wird der Begriff „Chiffre“ auch als Bezeichnung für einzelne Kryptotextzeichen oder kleinere Kryptotextsequenzen verwendet.)

Lemma 1.5 *Für jedes Paar $(k, k') \in S$ ist die Chiffrierfunktion E_k injektiv.*

Beweis Angenommen, für zwei unterschiedliche Klartexte $x_1 \neq x_2$ ist $E(k, x_1) = E(k, x_2)$. Dann folgt

$$D(k', E(k, x_1)) = D(k', E(k, x_2)) \overset{(1.1)}{=} x_2 \neq x_1,$$

im Widerspruch zu (1.1). ■

1.3 Die affine Chiffre

Die Moduloarithmetik erlaubt es uns, das Klartextalphabet mit einer Addition und Multiplikation auszustatten.

Definition 1.6 (teilt-Relation, modulare Kongruenz) *Seien a, b, m ganze Zahlen mit $m \geq 1$. Die Zahl a **teilt** b (kurz: $a|b$), falls ein $d \in \mathbb{Z}$ existiert mit $b = ad$. Teilt m die Differenz $a - b$, so schreiben wir hierfür*

$$a \equiv_m b$$

*(in Worten: a ist **kongruent** zu b modulo m). Weiterhin bezeichne*

$$a \bmod m = \min\{a - dm \geq 0 \mid d \in \mathbb{Z}\}$$

Tabelle 1.1: Werte der additiven Chiffrierfunktion ROT13 (Schlüssel $k = 13$).

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
$E(13, x)$	*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*

den bei der Ganzzahldivision von a durch m auftretenden ***Rest****, also diejenige ganze Zahl $r \in \{0, \ldots, m-1\}$, für die eine ganze Zahl $d \in \mathbb{Z}$ existiert mit $a = dm + r$.*

Die auf $\mathbb{Z}$ definierten Operationen

$$a \oplus_m b := (a + b) \bmod m$$

und

$$a \odot_m b := ab \bmod m.$$

sind abgeschlossen auf $\mathbb{Z}_m = \{0, \ldots, m-1\}$ und bilden auf dieser Menge einen kommutativen Ring mit Einselement, den sogenannten **Restklassenring** modulo m. Für $a \oplus_m -b$ schreiben wir auch $a \ominus_m b$.

Durch Identifikation der Buchstaben a_i mit ihren Indizes können wir die auf $\mathbb{Z}_m$ definierten Rechenoperationen auf Buchstaben übertragen.

Definition 1.7 (Buchstabenrechnung) *Sei $A = \{a_0, \ldots, a_{m-1}\}$ ein Alphabet. Für Indizes $i, j \in \{0, \ldots, m-1\}$ und eine ganze Zahl $z \in \mathbb{Z}$ ist*

$$\begin{array}{lll} a_i + a_j = a_{i \oplus_m j}, & a_i - a_j = a_{i \ominus_m j}, & a_i a_j = a_{i \odot_m j}, \\ a_i + z \;= a_{i \oplus_m z}, & a_i - z \;= a_{i \ominus_m z}, & z a_j \;= a_{z \odot_m j}. \end{array}$$

Mit Hilfe dieser Notation lässt sich die Verschiebechiffre, die auch als additive Chiffre bezeichnet wird, leicht beschreiben.

Definition 1.8 (additive Chiffre) *Bei der* ***additiven Chiffre*** *ist $A = B = M = C$ ein beliebiges Alphabet mit $m := \|A\| > 1$ und $K = \{1, \ldots, m-1\}$. Für $k \in K$, $x \in M$ und $y \in C$ gilt*

$$E(k, x) = x + k \;\; \textit{und} \;\; D(c, y) = y - k.$$

Im Fall des lateinischen Alphabets führt der Schlüssel $k = 13$ auf eine interessante Chiffrierfunktion, die in UNIX-Umgebungen auch unter der Bezeichnung ROT13 bekannt ist (siehe Tabelle 1.3). Natürlich kann mit dieser Substitution nicht ernsthaft die Vertraulichkeit von Nachrichten geschützt werden. Vielmehr soll durch sie ein unbeabsichtigtes Mitlesen – etwa von Rätsellösungen – verhindert werden.

ROT13 ist eine ***involutorische*** – also zu sich selbst inverse – Abbildung, d.h. für alle $x \in A$ gilt

$$\text{ROT13}(\text{ROT13}(x)) = x.$$

Da ROT13 zudem keinen Buchstaben auf sich selbst abbildet, ist sie sogar eine echt involutorische Abbildung.

Die Buchstabenrechnung legt folgende Modifikation der Caesar-Chiffre nahe: Anstatt auf jeden Klartextbuchstaben den Schlüsselwert k zu addieren, können wir die Klartextbuchstaben auch mit k multiplizieren. Allerdings erhalten wir hierbei nicht für jeden Wert von k eine injektive Chiffrierfunktion. So bildet etwa die Funktion $g : A_{lat} \to A_{lat}$ mit $g(x) = 2x$ sowohl A als auch N auf den Buchstaben $g(\mathsf{A}) = g(\mathsf{N}) = \mathsf{A}$ ab. Um die vom Schlüsselwert k zu erfüllende Bedingung angeben zu können, führen wir folgende Begriffe ein.

Definition 1.9 (ggT, kgV, teilerfremd) *Seien $a, b \in \mathbb{Z}$. Für $(a, b) \neq (0, 0)$ ist*

$$\text{ggT}(a, b) = \max\{d \in \mathbb{Z} \mid d \textit{ teilt die beiden Zahlen } a \textit{ und } b\}$$

*der **größte gemeinsame Teiler** von a und b. Für $a \neq 0, b \neq 0$ ist*

$$\text{kgV}(a, b) = \min\{d \in \mathbb{Z} \mid d \geq 1 \textit{ und die beiden Zahlen } a \textit{ und } b \textit{ teilen } d\}$$

*das **kleinste gemeinsame Vielfache** von a und b. Ist $\text{ggT}(a, b) = 1$, so nennt man a und b **teilerfremd**.*

Euklidscher Algorithmus: Der größte gemeinsame Teiler zweier Zahlen a und b lässt sich wie folgt bestimmen.

O. B. d. A. sei $a > b > 0$. Bestimme die natürlichen Zahlen (durch Divsision mit Rest):

$$r_0 = a > r_1 = b > r_2 > \cdots > r_n > r_{n+1} = 0 \quad \text{und} \quad d_2, d_3, \ldots d_{n+1}$$

mit

$$r_{i-1} = d_{i+1} r_i + r_{i+1} \quad \text{für} \quad i = 1, \ldots, n.^*$$

Hierzu sind n Divisionsschritte erforderlich. Wegen

$$\text{ggT}(r_{i-1}, r_i) = \text{ggT}(r_i, \underbrace{r_{i-1} - d_{i+1} r_i}_{r_{i+1}})$$

folgt $\text{ggT}(a, b) = \text{ggT}(r_n, r_{n+1}) = r_n$.

*Also: $d_i = r_{i-2} \text{ div } r_{i-1}$ und $r_i = r_{i-2} \bmod r_{i-1}$.

Beispiel 1.10

Für $a = 693$ und $b = 147$ erhalten wir

i	r_{i-1}	$=$	d_{i+1}	$\cdot$	r_i	$+$	r_{i+1}
1	693	$=$	4	$\cdot$	147	$+$	105
2	147	$=$	1	$\cdot$	105	$+$	42
3	105	$=$	2	$\cdot$	42	$+$	21
4	42	$=$	2	$\cdot$	**21**	$+$	0

und damit $\mathrm{ggT}(693, 147) = r_4 = 21$.

Der Euklidsche Algorithmus lässt sich sowohl iterativ als auch rekursiv implementieren.

Algorithmus 1.11 $\textsc{Euklid}_{it}(a, b)$

```
repeat
  r ← a mod b
  a ← b
  b ← r
until r = 0
return a
```

Algorithmus 1.12 $\textsc{Euklid}_{rek}(a, b)$

```
if b = 0 then
  return a
else
  return EUKLID(b, a mod b)
end
```

Zur Abschätzung von n verwenden wir die Folge der Fibonacci-Zahlen f_n:

$$f_n = \begin{cases} 0, & \text{falls } n = 0 \\ 1, & \text{falls } n = 1 \\ f_{n-1} + f_{n-2}, & \text{falls } n \geq 2 \end{cases}$$

Durch Induktion über $i = n, n-1, \ldots, 0$ folgt $r_i \geq f_{n+1-i}$; also $a \geq f_{n+1}$. Wegen $f_n \geq \Re^n$ (wobei $\Re = \frac{1+\sqrt{5}}{2}$; Beweis durch Induktion) ist dann $a \geq \Re^n$, d. h. $n \leq \log_{\Re} a$.

Theorem 1.13 *Der Euklidsche Algorithmus führt zur Berechnung von* $\mathrm{ggT}(a, b)$ *(unter der Annahme* $a > b > 0$*) höchstens* $\lfloor \log_R a \rfloor + 1$ *Divisionsschritte durch. Dies führt auf eine Zeitkomplexität von* $O(n^3)$*, wobei* n *die Länge der Eingabe in Binärdarstellung bezeichnet und wir* $O(n^2)$ *Rechenschritte für eine einzelne Ganzzahldivision ansetzen.*

Erweiterter Euklidscher bzw. **Berlekamp-Algorithmus:** Der Euklidsche Algorithmus kann so modifiziert werden, dass er eine lineare Darstellung

$$\mathrm{ggT}(a, b) = \lambda a + \mu b \quad \text{mit} \quad \lambda,\ \mu \in \mathbb{Z}$$

des ggT liefert (Zeitkomplexität ebenfalls $O(n^3)$). Hierzu werden neben r_i und d_i weitere Zahlen

$$p_i = p_{i-2} - d_i p_{i-1}, \quad \text{wobei} \quad p_0 = 1 \quad \text{und} \quad p_1 = 0,$$

und

$$q_i = q_{i-2} - d_i q_{i-1}, \quad \text{wobei} \quad q_0 = 0 \quad \text{und} \quad q_1 = 1,$$

für $i = 0, \ldots, n$ bestimmt. Dann gilt für $i = 0$ und $i = 1$,

$$ap_i + bq_i = r_i,$$

und durch Induktion über i,

$$\begin{aligned} ap_{i+1} + bq_{i+1} &= a(p_{i-1} - d_{i+1}p_i) + b(q_{i-1} - d_{i+1}q_i) \\ &= ap_{i-1} + bq_{i-1} - d_{i+1}(ap_i + bq_i) \\ &= (r_{i-1} - d_{i+1}r_i) \\ &= r_{i+1} \end{aligned}$$

zeigt man, dass dies auch für $i = 2, \ldots, n$ gilt. Insbesondere gilt also

$$ap_n + bq_n = r_n = \mathrm{ggT}(a, b).$$

Korollar 1.14 (Lemma von Bezout) *Der größte gemeinsame Teiler von a und b ist in der Form*

$$\mathrm{ggT}(a, b) = \lambda a + \mu b \quad \textit{mit} \quad \lambda,\ \mu \in \mathbb{Z}$$

darstellbar.

Beispiel 1.15 Für $a = 693$ und $b = 147$ erhalten wir wegen

i	$r_{i-1} = d_{i+1} \cdot r_i + r_{i+1}$	p_i	q_i	$p_i \cdot 693 + q_i \cdot 147 = r_i$
0		1	0	$1 \cdot 693 + 0 \cdot 147 = 693$
1	$693 = 4 \cdot 147 + 105$	0	1	$0 \cdot 693 + 1 \cdot 147 = 147$
2	$147 = 1 \cdot 105 + 42$	1	-4	$1 \cdot 693 - 4 \cdot 147 = 105$
3	$105 = 2 \cdot 42 + 21$	-1	5	$-1 \cdot 693 + 5 \cdot 147 = 42$
4	$42 = 2 \cdot \mathbf{21} + 0$	$\mathbf{3}$	$\mathbf{-14}$	$\mathbf{3} \cdot 693 - \mathbf{14} \cdot 147 = 21$

die lineare Darstellung $3 \cdot 693 - 14 \cdot 147 = 21$.

Aus der linearen Darstellbarkeit des größten gemeinsamen Teilers ergeben sich eine Reihe von nützlichen Schlussfolgerungen.

Korollar 1.16 $\operatorname{ggT}(a, b) = \min\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$.

Beweis Sei $m = \min\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$ und $g = \operatorname{ggT}(a, b)$. Dann folgt $g \geq m$, da g in der Menge $\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$ enthalten ist, und $g \leq m$, da g Teiler von jeder Zahl der Form $\lambda a + \mu b$ ist. ■

Korollar 1.17 *Der größte gemeinsame Teiler von a und b wird von allen gemeinsamen Teilern von a und b geteilt,*

$$x|a \ \wedge \ x|b \quad \Rightarrow \quad x| \operatorname{ggT}(a, b).$$

Beweis Sei $g = \operatorname{ggT}(a, b)$. Dann existieren Zahlen $\mu, \lambda \in \mathbb{Z}$ mit $\mu a + \lambda b = g$. Da x nach Voraussetzung sowohl a als auch b teilt, teilt x auch die Zahlen μa und λb und somit auch deren Summe $\mu a + \lambda b = g$. ■

Korollar 1.18 (Lemma von Euklid) *Teilt a das Produkt bc und sind a, b teilerfremd, so ist a auch Teiler von c,*

$$a|bc \ \wedge \ \operatorname{ggT}(a, b) = 1 \quad \Rightarrow \quad a|c.$$

Beweis Wegen $\operatorname{ggT}(a, b) = 1$ existieren Zahlen $\mu, \lambda \in \mathbb{Z}$ mit $\mu a + \lambda b = 1$. Da a nach Voraussetzung das Produkt bc teilt, muss a auch $c\mu a + c\lambda b = c$ teilen. ■

Korollar 1.19 *Wenn sowohl a als auch b zu einer Zahl $m \in \mathbb{Z}$ teilerfremd sind, so ist auch das Produkt ab teilerfremd zu m,*

$$\operatorname{ggT}(a, m) = \operatorname{ggT}(b, m) = 1 \quad \Rightarrow \quad \operatorname{ggT}(ab,\ m) = 1.$$

Beweis Da a und b teilerfremd zu m sind, existieren Zahlen $\mu, \lambda, \mu', \lambda' \in \mathbb{Z}$ mit $\mu a + \lambda m = \mu' b + \lambda' m = 1$. Somit ergibt sich aus der Darstellung

$$1 = (\mu a + \lambda m)(\mu' b + \lambda' m) = \underbrace{\mu\mu'}_{\mu''} ab + \underbrace{(\mu a \lambda' + \mu' b \lambda + \lambda\lambda' m)}_{\lambda''} m,$$

dass auch ab teilerfremd zu m ist. ■

Damit nun eine Abbildung $g : A \to A$ von der Bauart $g(x) = bx$ injektiv (oder gleichbedeutend, surjektiv) ist, muss es zu jedem Buchstaben $y \in A$ genau einen Buchstaben $x \in A$ mit $bx = y$ geben. Wie der folgende Satz zeigt, ist dies genau dann der Fall, wenn b und m teilerfremd sind.

Satz 1.20 *Sei $m \geq 1$. Die lineare Kongruenzgleichung $bx \equiv_m y$ besitzt genau dann eine eindeutige Lösung $x \in \{0, \ldots, m-1\}$, wenn $\operatorname{ggT}(b, m) = 1$ ist.*

Beweis Angenommen, $\mathrm{ggT}(b,m) = g > 1$. Dann ist mit x auch $x' = x + m/g$ eine Lösung von $bx \equiv_m y$ mit $x \not\equiv_m x'$. Gilt umgekehrt $\mathrm{ggT}(b,m) = 1$, so folgt aus den Kongruenzen

$$bx_1 \equiv_m y$$

und

$$bx_2 \equiv_m y$$

sofort $b(x_1 - x_2) \equiv_m 0$, also $m|b(x_1 - x_2)$. Wegen $\mathrm{ggT}(b,m) = 1$ folgt mit dem Lemma von Euklid $m|(x_1 - x_2)$, also $x_1 \equiv_m x_2$.

Dies zeigt, dass die Abbildung $f : \mathbb{Z}_m \to \mathbb{Z}_m$ mit $f(x) = bx \bmod m$ injektiv ist. Da jedoch Definitions- und Wertebereich von f identisch sind, muss f dann auch surjektiv sein. Dies impliziert, dass die Kongruenz $bx \equiv_m y$ für jedes $y \in \mathbb{Z}_m$ lösbar ist. ■

Korollar 1.21 *Im Fall $\mathrm{ggT}(b,m) = 1$ hat die Kongruenz $bx \equiv_m 1$ genau eine Lösung, die das **multiplikative Inverse** von b modulo m genannt und mit $b^{-1} \bmod m$ (oder einfach mit b^{-1}) bezeichnet wird. Die invertierbaren Elemente von $\mathbb{Z}_m$ werden in der Menge*

$$\mathbb{Z}_m^* = \{b \in \mathbb{Z}_m \mid \mathrm{ggT}(b,m) = 1\}$$

zusammengefasst.

Korollar 1.19 zeigt, dass $\mathbb{Z}_m^*$ unter der Operation $\odot_m$ abgeschlossen ist, und mit Korollar 1.21 folgt, dass $(\mathbb{Z}_m^*, \odot_m)$ eine multiplikative Gruppe bildet.

Das multiplikative Inverse von b modulo m ergibt sich aus der linearen Darstellung $\lambda b + \mu m = \mathrm{ggT}(b,m) = 1$ zu $b^{-1} = \lambda \bmod m$. Bei Kenntnis von b^{-1} kann die Kongruenz $bx \equiv_m y$ leicht zu $x = yb^{-1} \bmod m$ gelöst werden. Die folgende Tabelle zeigt die multiplikativen Inversen b^{-1} für alle $b \in \mathbb{Z}_{26}^*$.

b	1	3	5	7	9	11	15	17	19	21	23	25
b^{-1}	1	9	21	15	3	19	7	23	11	5	17	25

Nun lässt sich die additive Chiffre leicht zur affinen Chiffre erweitern.

Definition 1.22 (affine Chiffre) *Bei der **affinen Chiffre** ist $A = B = M = C$ ein beliebiges Alphabet mit $m := \|A\| > 1$ und $K = \mathbb{Z}_m^* \times \mathbb{Z}_m$. Für $k = (b,c) \in K$, $x \in M$ und $y \in C$ gilt*

$$E(k,x) = bx + c \;\; \textit{und} \;\; D(k,y) = b^{-1}(y - c).$$

In diesem Fall liefert die Schlüsselkomponente $b = -1$ für jeden Wert von c eine involutorische Chiffrierfunktion $x \mapsto E(b, c; x) = c - x$ (**verschobenes komplementäres Alphabet**). Wählen wir für c ebenfalls den Wert -1, so ergibt sich die Chiffrierfunktion $x \mapsto -x - 1$, die auch als **revertiertes Alphabet** bekannt ist. Offenbar ist diese Funktion genau dann echt involutorisch, wenn m gerade ist.

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
$-x-1$	*Z Y X W V U T S R Q P O N M L K J I H G F E D C B A*

Als nächstes illustrieren wir die Ver- und Entschlüsselung mit der affinen Chiffre an einem kleinen Beispiel.

Beispiel 1.23 (affine Chiffre) Sei $A = \{\mathsf{A}, \ldots, \mathsf{Z}\} = B$, also $m = 26$. Weiter sei $k = (9, 2)$, also $b = 9$ und $c = 2$. Um den Klartextbuchstaben $x = \mathsf{F}$ zu verschlüsseln, berechnen wir

$$E(k, x) = bx + c = 9\mathsf{F} + 2 = \mathit{V},$$

da der Index von F gleich 5, der von V gleich 21 und $9 \cdot 5 + 2 = 47 \equiv_{26} 21$ ist. Um einen Kryptotextbuchstaben wieder entschlüsseln zu können, benötigen wir das multiplikative Inverse von $b = 9$, das sich wegen

i	$r_{i-1} = d_{i+1} \cdot r_i + r_{i+1}$	$p_i \cdot 26 + q_i \cdot 9 = r_i$
0		$1 \cdot 26 + 0 \cdot 9 = 26$
1	$26 = 2 \cdot 9 + 8$	$0 \cdot 26 + 1 \cdot 9 = 9$
2	$9 = 1 \cdot 8 + 1$	$1 \cdot 26 + (-2) \cdot 9 = 8$
3	$8 = 8 \cdot \mathbf{1} + 0$	$(-1) \cdot 26 + \mathbf{3} \cdot 9 = 1$

zu $b^{-1} = q_3 = 3$ ergibt. Damit erhalten wir für den Kryptotextbuchstaben $y = \mathit{V}$ den ursprünglichen Klartextbuchstaben

$$D(k, y) = b^{-1}(y - c) = 3(\mathit{V} - 2) = \mathsf{F}$$

zurück, da $3 \cdot 19 = 57 \equiv_{26} 5$ ist.

Eine wichtige Rolle spielt die Funktion

$$\varphi : \mathbb{N} \to \mathbb{N} \quad \text{mit} \quad \varphi(n) = \|\mathbb{Z}_n^*\| = \|\{a \mid 0 \le a \le n-1,\ \mathrm{ggT}(a, n) = 1\}\|,$$

die sogenannte *Eulersche φ-Funktion*.

n	1	2	3	4	5	6	7	8	9
$\mathbb{Z}_n^*$	$\{0\}$	$\{1\}$	$\{1,2\}$	$\{1,3\}$	$\{1,2,3,4\}$	$\{1,5\}$	$\{1,\cdots,6\}$	$\{1,3,5,7\}$	$\{1,2,4,5,7,8\}$
$\varphi(n)$	1	1	2	2	4	2	6	4	6

Wegen

$$\mathbb{Z}_{p^e} - \mathbb{Z}_{p^e}^* = \{0, p, 2p, \ldots, (p^{e-1} - 1)p\}$$

folgt sofort

$$\varphi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1).$$

Um hieraus für beliebige Zahlen $m \in \mathbb{N}$ eine Formel für $\varphi(m)$ zu erhalten, genügt es, $\varphi(ab)$ im Fall $\mathrm{ggT}(a,b) = 1$ in Abhängigkeit von $\varphi(a)$ und $\varphi(b)$ zu bestimmen. Hierzu betrachten wir die Abbildung $f : \mathbb{Z}_{ml} \to \mathbb{Z}_m \times \mathbb{Z}_l$ mit

$$f(x) := (x \bmod m, x \bmod l).$$

Beispiel 1.24

Sei $m = 5$ und $l = 6$. Dann erhalten wir die Funktion $f : \mathbb{Z}_{30} \to \mathbb{Z}_5 \times \mathbb{Z}_6$ mit

x	0	**1**	2	3	4	5	6	**7**	8	9
$f(x)$	$(0,0)$	$(\mathbf{1},\mathbf{1})$	$(\mathbf{2},2)$	$(\mathbf{3},3)$	$(\mathbf{4},4)$	$(0,\mathbf{5})$	$(\mathbf{1},0)$	$(\mathbf{2},\mathbf{1})$	$(\mathbf{3},2)$	$(\mathbf{4},3)$

x	10	**11**	12	**13**	14	15	16	**17**	18	**19**
$f(x)$	$(0,4)$	$(\mathbf{1},\mathbf{5})$	$(\mathbf{2},0)$	$(\mathbf{3},\mathbf{1})$	$(\mathbf{4},2)$	$(0,3)$	$(\mathbf{1},4)$	$(\mathbf{2},\mathbf{5})$	$(\mathbf{3},0)$	$(\mathbf{4},\mathbf{1})$

x	20	21	22	**23**	24	25	26	27	28	**29**
$f(x)$	$(0,2)$	$(\mathbf{1},3)$	$(\mathbf{2},4)$	$(\mathbf{3},\mathbf{5})$	$(\mathbf{4},0)$	$(0,\mathbf{1})$	$(\mathbf{1},2)$	$(\mathbf{2},3)$	$(\mathbf{3},4)$	$(\mathbf{4},\mathbf{5})$

Man beachte, dass f eine Bijektion zwischen $\mathbb{Z}_{30}$ und $\mathbb{Z}_5 \times \mathbb{Z}_6$ ist. Zudem fällt auf, dass ein x-Wert genau dann in $\mathbb{Z}_{30}^*$ liegt, wenn der Funktionswert $f(x) = (y,z)$ zu $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$ gehört (die Werte $x \in \mathbb{Z}_{30}^*$, $y \in \mathbb{Z}_5^*$ und $z \in \mathbb{Z}_6^*$ sind **fett** gedruckt). Folglich bildet f die Argumente in $\mathbb{Z}_{30}^*$ bijektiv auf die Werte in $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$ ab. Für f^{-1} erhalten wir somit folgende Tabelle:

f^{-1}	0	**1**	2	3	4	**5**
0	0	25	20	15	10	5
1	6	**1**	26	21	16	**11**
2	12	**7**	2	27	22	**17**
3	18	**13**	8	3	28	**23**
4	24	**19**	14	9	4	**29**

◁

Der Chinesische Restsatz, den wir im nächsten Abschnitt beweisen, besagt, dass f im Fall $\mathrm{ggT}(m,l) = 1$ bijektiv und damit invertierbar ist. Wegen

$$\begin{aligned} \mathrm{ggT}(x, ml) = 1 \quad &\Leftrightarrow \quad \mathrm{ggT}(x,m) = \mathrm{ggT}(x,l) = 1 \\ &\Leftrightarrow \quad \mathrm{ggT}(x \bmod m, m) = \mathrm{ggT}(x \bmod l, l) = 1 \end{aligned}$$

ist daher die Einschränkung $\hat{f}$ von f auf den Bereich $\mathbb{Z}_{ml}^*$ eine Bijektion zwischen $\mathbb{Z}_{ml}^*$ und $\mathbb{Z}_m^* \times \mathbb{Z}_l^*$, d.h. es gilt

$$\varphi(ml) = \|\mathbb{Z}_{ml}^*\| = \|\mathbb{Z}_m^* \times \mathbb{Z}_l^*\| = \|\mathbb{Z}_m^*\| \cdot \|\mathbb{Z}_l^*\| = \varphi(m)\varphi(l).$$

Theorem 1.25 *Die Eulersche φ-Funktion ist multiplikativ, d. h. für teilerfremde Zahlen m und l gilt $\varphi(ml) = \varphi(m)\varphi(l)$.*

Korollar 1.26 *Sei $m = \prod_{i=1}^k p_i^{e_i}$ die Primfaktorzerlegung von m. Dann gilt*

$$\varphi(m) = \prod_{i=1}^k p_i^{e_i-1}(p_i - 1) = m \prod_{i=1}^k (p_i - 1)/p_i.$$

Beweis Es gilt

$$\varphi(\prod_{i=1}^k p_i^{e_i}) = \prod_{i=1}^k \varphi(p_i^{e_i}) = \prod_{i=1}^k (p_i^{e_i} - p_i^{e_i-1}) = \prod_{i=1}^k p_i^{e_i-1}(p_i - 1).$$

■

Der Chinesische Restsatz

Die beiden linearen Kongruenzen

$$\begin{aligned} x &\equiv_3 0 \\ x &\equiv_6 1 \end{aligned}$$

besitzen je eine Lösung, es gibt aber kein x, das beide Kongruenzen gleichzeitig erfüllt. Der nächste Satz zeigt, dass unter bestimmten Voraussetzungen gemeinsame Lösungen existieren, und wie sie berechnet werden können.

Theorem 1.27 (Chinesischer Restsatz) *Falls $m_1, \ldots, m_k$ paarweise teilerfremd sind, dann hat das System*

$$\begin{aligned} x &\equiv_{m_1} b_1 \\ &\vdots \\ x &\equiv_{m_k} b_k \end{aligned} \tag{1.2}$$

genau eine Lösung modulo $m = \prod_{i=1}^k m_i$.

Beweis Da die Zahlen $n_i = {}^m/_{m_i}$ teilerfremd zu m_i sind, existieren Zahlen μ_i und λ_i mit

$$\mu_i n_i + \lambda_i m_i = \mathrm{ggT}(n_i, m_i) = 1.$$

Dann gilt

$$\mu_i n_i \equiv_{m_i} 1$$

und

$$\mu_i n_i \equiv_{m_j} 0$$

für $j \neq i$. Folglich erfüllt $x = \sum_{j=1}^{k} \mu_j n_j b_j$ die Kongruenzen

$$x \equiv_{m_i} \mu_i n_i b_i \equiv_{m_i} b_i$$

für $i = 1, \ldots, k$. Dies zeigt, dass (1.2) lösbar, also die Funktion

$$f : \mathbb{Z}_m \to \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k}$$

mit $f(x) = (x \bmod m_1, \ldots, x \bmod m_k)$ surjektiv ist. Da der Definitions- und der Wertebereich von f die gleiche Mächtigkeit haben, muss f jedoch auch injektiv sein, d.h. (1.2) ist sogar eindeutig lösbar. ■

Man beachte, dass der Beweis des Chinesischen Restsatzes konstruktiv ist und die Lösung x unter Verwendung des erweiterten Euklidschen Algorithmus' effizient berechenbar ist.

1.4 Die Hill-Chiffre

Die von Hill im Jahr 1929 publizierte Chiffre ist eine Erweiterung der multiplikativen Chiffre auf Buchstabenblöcke, d.h. der Klartext wird nicht zeichenweise, sondern blockweise verarbeitet. Sowohl der Klartext- als auch der Kryptotextraum enthält alle Wörter x über A einer festen Länge l. Zur Chiffrierung wird eine $(l \times l)$-Matrix $k = (k_{ij})$ mit Koeffizienten in $\mathbb{Z}_m$ benutzt, die einen Klartextblock $x = x_1 \ldots x_l \in A^l$ in den Kryptotextblock $y_1 \ldots y_l \in A^l$ transformiert, wobei

$$y_i = x_1 k_{1i} + \cdots + x_l k_{li},\ i = 1, \ldots, l$$

ist (hierbei machen wir von der Buchstabenrechnung Gebrauch). y entsteht also durch Multiplikation von x mit der Schlüsselmatrix k:

$$(x_1, \cdots, x_l) \begin{pmatrix} k_{11} & \ldots & k_{1l} \\ \vdots & \ddots & \vdots \\ k_{l1} & \ldots & k_{ll} \end{pmatrix} = (y_1, \cdots, y_l)$$

Wir bezeichnen die Menge aller $(l \times l)$-Matrizen mit Koeffizienten in $\mathbb{Z}_m$ mit $\mathbb{Z}_m^{l\times l}$. Als Schlüssel können nur invertierbare Matrizen k benutzt werden, da sonst der Chiffriervorgang nicht injektiv ist. k ist genau dann invertierbar, wenn die Determinante von k teilerfremd zu m ist (siehe Übungen).

Definition 1.28 (Determinante) *Sei $A = (a_{ij})$ eine $l\times l$-Matrix. Für $1 \le i, j \le l$ sei A_{ij} die durch Streichen der i-ten Zeile und j-ten Spalte aus K hervorgehende Matrix. Die **Determinante** von A ist dann $\det(A) = a_{11}$, falls $l = 1$, und*

$$\det(A) = \sum_{j=1}^{l} (-1)^{i+j} a_{i,j} \det(A_{ij}),$$

wobei $i \in \{1, \cdots, l\}$ (beliebig wählbar) ist.

Für die Dechiffrierung wird die zu k inverse Matrix k^{-1} benötigt, wofür effiziente Algorithmen bekannt sind (siehe Übungen).

Satz 1.29 *Sei A ein Alphabet und sei $k \in \mathbb{Z}_m^{l\times l}$ ($l \ge 1$, $m = \|A\|$). Die Abbildung $f : A^l \to A^l$ mit*

$$f(x) = xk,$$

ist genau dann injektiv, wenn $\mathrm{ggT}(\det(k), m) = 1$ ist.

Beweis Siehe Übungen. ■

Definition 1.30 (Hill-Chiffre) *Sei $A = \{a_0, \ldots, a_{m-1}\}$ ein beliebiges Alphabet und für eine natürliche Zahl $l \ge 2$ sei $M = C = A^l$. Bei der **Hill-Chiffre** ist $K = \{k \in \mathbb{Z}_m^{l\times l} \mid \mathrm{ggT}(\det(k), m) = 1\}$ und es gilt*

$$E(k, x) = xk \ \textit{ und } \ D(k, y) = yk^{-1}.$$

Beispiel 1.31 (Hill-Chiffre) Benutzen wir zur Chiffrierung von Klartextblöcken der Länge $l = 4$ über dem lateinischen Alphabet A_{lat} die Schlüsselmatrix

$$k = \begin{pmatrix} 11 & 13 & 8 & 21 \\ 24 & 17 & 3 & 25 \\ 18 & 12 & 23 & 17 \\ 6 & 15 & 2 & 15 \end{pmatrix},$$

so erhalten wir beispielsweise für den Klartext $\mathtt{HILL}$ wegen

$$(\mathtt{H\,I\,L\,L}) \begin{pmatrix} 11 & 13 & 8 & 21 \\ 24 & 17 & 3 & 25 \\ 18 & 12 & 23 & 17 \\ 6 & 15 & 2 & 15 \end{pmatrix} = (\mathit{N\,E\,R\,X}) \ \text{ bzw. } \ \begin{array}{r} 11\,\mathtt{H} + 24\,\mathtt{I} + 18\,\mathtt{L} + 6\,\mathtt{L} = \mathit{N} \\ 13\,\mathtt{H} + 17\,\mathtt{I} + 12\,\mathtt{L} + 15\,\mathtt{L} = \mathit{E} \\ 8\,\mathtt{H} + 3\,\mathtt{I} + 23\,\mathtt{L} + 2\,\mathtt{L} = \mathit{R} \\ 21\,\mathtt{H} + 25\,\mathtt{I} + 17\,\mathtt{L} + 15\,\mathtt{L} = \mathit{X} \end{array}$$

den Kryptotext $E(k, \mathtt{HILL}) = \mathit{NERX}$. Für die Entschlüsselung wird die inverse Matrix k^{-1} benötigt. Diese wird in den Übungen berechnet.

1.5 Die Vigenère-Chiffre und andere Stromsysteme

Bei der nach dem Franzosen Blaise de Vigenère (1523–1596) benannten Chiffre werden zwar nur einzelne Buchstaben chiffriert, aber je nach Position im Klartext unterschiedlich.

Definition 1.32 (Vigenère-Chiffre) *Sei $A = B$ ein beliebiges Alphabet. Die **Vigenère-Chiffre** chiffriert unter einem Schlüssel $k = k_0 \dots k_{d-1} \in K = A^*$ einen Klartext $x = x_0 \dots x_{n-1}$ beliebiger Länge zu*

$$E(k, x) = y_0 \dots y_{n-1}, \text{ wobei } y_i = x_i + k_{(i \bmod d)} \text{ ist,}$$

und dechiffriert einen Kryptotext $y = y_0 \dots y_{n-1}$ zu

$$D(k, y) = x_0 \dots x_{n-1}, \text{ wobei } x_i = y_i - k_{(i \bmod d)} \text{ ist.}$$

Beispiel 1.33 (Vigenère-Chiffre) Verwenden wir das lateinische Alphabet A_{lat} als Klartextalpabet und wählen wir als Schlüssel das Wort $k = \mathit{WIE}$, so ergibt sich für den Klartext VIGENERE beispielsweise der Kryptotext

$$\begin{aligned} E(\mathit{WIE}, \mathtt{VIGENERE}) &= \underbrace{\mathtt{V}+\mathit{W}}_{\mathit{R}}\ \underbrace{\mathtt{I}+\mathit{I}}_{\mathit{Q}}\ \underbrace{\mathtt{G}+\mathit{E}}_{\mathit{K}}\ \underbrace{\mathtt{E}+\mathit{W}}_{\mathit{A}}\ \underbrace{\mathtt{N}+\mathit{I}}_{\mathit{V}}\ \underbrace{\mathtt{E}+\mathit{E}}_{\mathit{I}}\ \underbrace{\mathtt{R}+\mathit{W}}_{\mathit{N}}\ \underbrace{\mathtt{E}+\mathit{I}}_{\mathit{M}} \\ &= \mathit{RQKAVINM} \end{aligned}$$

Um einen Klartext x zu verschlüsseln, wird also das Schlüsselwort $k = k_0 \dots k_{d-1}$ so oft wiederholt, bis der dabei entstehende **Schlüsselstrom** $\hat{k} = k_0, k_1, \dots, k_{d-1}, k_0 \dots$ die Länge von x erreicht. Dann werden x und $\hat{k}$ zeichenweise addiert, um den zugehörigen Kryptotext y zu bilden. Aus diesem kann der ursprüngliche Klartext x zurückgewonnen werden, indem man den Schlüsselstrom $\hat{k}$ wieder subtrahiert.

Beispiel 1.33 ((Vigenère-Chiffre, Fortsetzung))

Chiffrierung:		Dechiffrierung:	
VIGENERE	(Klartext x)	*RQKAVINM*	(Kryptotext y)
+ *<u>WIEWIEWI</u>*	(Schlüsselstrom $\hat{k}$)	− *<u>WIEWIEWI</u>*	(Schlüsselstrom $\hat{k}$)
RQKAVINM	(Kryptotext y)	VIGENERE	(Klartext x)

Die Chiffrierarbeit lässt sich durch Benutzung einer Additionstabelle erleichtern (auch als **Vigenère-Tableau** bekannt).

+	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

Um eine involutorische Chiffre zu erhalten, schlug Sir Francis Beaufort, ein Admiral der britischen Marine, vor, den Schlüsselstrom nicht auf den Klartext zu addieren, sondern letzteren von ersterem zu subtrahieren.

Beispiel 1.34 (Beaufort-Chiffre) Verschlüsseln wir den Klartext `BEAUFORT` beispielsweise unter dem Schlüsselwort $k = \mathtt{\textit{WIE}}$, so erhalten wir den Kryptotext *`XMEQNSNB`*. Eine erneute Verschlüsselung liefert wieder den Klartext `BEAUFORT`:

Chiffrierung:

$$\begin{array}{rll} & \underline{\mathtt{\textit{WIE}}}\mathtt{\textit{WIEWI}} & (\text{Schlüsselstrom}) \\ - & \mathtt{BEAUFORT} & (\text{Klartext}) \\ \hline & \mathtt{\textit{XMEQNSNB}} & (\text{Kryptotext}) \end{array}$$

Dechiffrierung:

$$\begin{array}{rll} & \underline{\mathtt{\textit{WIE}}}\mathtt{\textit{WIEWI}} & (\text{Schlüsselstrom}) \\ - & \mathtt{\textit{XMEQNSNB}} & (\text{Kryptotext}) \\ \hline & \mathtt{BEAUFORT} & (\text{Klartext}) \end{array}$$

Bei den bisher betrachteten Chiffren wird aus einem Schlüsselwort $k = k_0 \dots k_{d-1}$ ein **periodischer Schlüsselstrom** $\hat{k} = \hat{k}_0 \dots \hat{k}_{n-1}$ erzeugt, das heißt, es gilt $\hat{k}_i = \hat{k}_{i+d}$ für alle $i = 0, \dots, n-d-1$. Da eine kleine Periode das Brechen der Chiffre erleichtert, sollte entweder ein Schlüsselstrom mit sehr großer Periode oder noch besser ein **fortlaufender Schlüsselstrom** zur Chiffrierung benutzt werden. Ein solcher nichtperiodischer Schlüsselstrom lässt sich beispielsweise ohne großen Aufwand erzeugen, indem man an das Schlüsselwort den Klartext oder den Kryptotext anhängt (sogenannte ***Autokey*-Chiffrierung**).[†]

Beispiel 1.35 (*Autokey*-Chiffre) Benutzen wir wieder das Schlüsselwort *WIE*, um den Schlüsselstrom durch Anhängen des Klar- bzw. Kryptotextes zu erzeugen, so erhalten wir für den Klartext VIGENERE folgende Kryptotexte:

Klartext-Schlüsselstrom:

$$\begin{array}{cll} & \mathtt{VIGENERE} & (\text{Klartext}) \\ + & \underline{\underline{\mathit{WIE}}\mathit{VIGEN}} & (\text{Schlüsselstrom}) \\ & \mathit{RQKZVKVR} & (\text{Kryptotext}) \end{array}$$

Kryptotext-Schlüsselstrom:

$$\begin{array}{cll} & \mathtt{VIGENERE} & (\text{Klartext}) \\ + & \underline{\underline{\mathit{WIE}}\mathit{RQKVD}} & (\text{Schlüsselstrom}) \\ & \mathit{RQKVDOMH} & (\text{Kryptotext}) \end{array}$$

Auch die Dechiffrierung ist in beiden Fällen einfach. Bei der ersten Alternative kann der Empfänger durch Subtraktion des Schlüsselworts den Anfang des Klartextes bilden und gleichzeitig den Schlüsselstrom verlängern, so dass sich auf diese Weise Stück für Stück der gesamte Kryptotext entschlüsseln lässt. Noch einfacher gestaltet sich die Dechiffrierung im zweiten Fall, da sich hier der Schlüsselstrom vom Kryptotext nur durch das vorangestelle Schlüsselwort unterscheidet.

1.6 Der One-Time-Pad

Es besteht auch die Möglichkeit, eine Textstelle in einem Buch als Schlüssel zu vereinbaren und den dort beginnenden Text als Schlüsselstrom zu benutzen (Lauftextverschlüsselung). Besser ist es jedoch, aus einem relativ kurzen Schlüssel einen möglichst zufällig erscheinenden Schlüsselstrom zu erzeugen. Hierzu können beispielsweise Pseudozufallsgeneratoren eingesetzt werden. Absolute Sicherheit wird dagegen erreicht, wenn der Schlüsselstrom rein zufällig erzeugt und nach einmaliger Benutzung wieder vernichtet wird.[‡] Ein solcher „Wegwerfschlüs-

[†] Die Idee, den Schlüsselstrom durch Anhängen des Klartextes an ein Schlüsselwort zu bilden, stammt von Vigenère, während er mit der Erfindung der nach ihm benannten Vigenère-Chiffre „nichts zu tun“ hatte. Diese wird vielmehr Giovan Batista Belaso (1553) zugeschrieben.

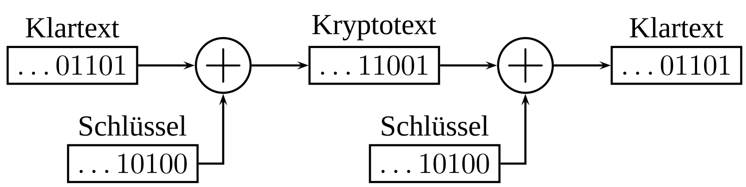
[‡] Diese Art der Schlüsselerzeugung schlug der amerikanische Major Joseph O. Mauborgne im Jahr 1918 vor, nachdem ihm ein von Gilbert S. Vernam für den Fernschreibverkehr entwickeltes Chiffriersystem vorgestellt wurde.

sel“ (*pad* oder *One-time-tape*, im Deutschen auch als **individueller Schlüssel** bezeichnet) lässt sich allerdings nur mit großem Aufwand generieren und verteilen, weshalb diese Chiffre nur wenig praktikabel ist. Dennoch wurde diese Methode beispielsweise beim „heißen Draht“, der 1963 eingerichteten, direkten Fernschreibverbindung zwischen dem Weißen Haus in Washington und dem Kreml in Moskau, angewandt.

Beispiel 1.36 (*pad*) Sei $A = \{a_0, \ldots, a_{m-1}\}$ ein beliebiges Klartextalphabet. Um einen Klartext $x = x_0 \ldots x_{n-1}$ zu verschlüsseln, wird auf jeden Klartextbuchstaben x_i ein neuer, zufällig generierter Schlüsselbuchstabe k_i addiert,

$$y = y_0 \ldots y_{n-1}, \text{ wobei } y_i = x_i + k_i.$$

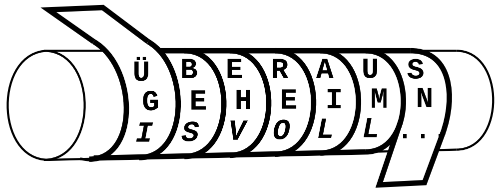
Der Klartext wird also wie bei einer additiven Chiffre verschlüsselt, nur dass der Schlüssel nach einmaligem Gebrauch gewechselt wird. Dies entspricht dem Gebrauch einer Vigenère-Chiffre, falls als Schlüssel ein zufällig gewähltes Wort von der Länge des Klartextes benutzt wird. Wie diese ist der *pad* im Binärfall also involutorisch.



1.7 Klassifikation von Kryptosystemen

Die bisher betrachteten Chiffrierfunktionen handelt es sich um **Substitutionen**, d.h. sie erzeugen den Kryptotext aus dem Klartext, indem sie Klartextzeichen – einzeln oder in Gruppen – durch Kryptotextzeichen *ersetzen*. Dagegen verändern **Transpositionen** lediglich die *Reihenfolge* der einzelnen Klartextzeichen.

Beispiel 1.37 (Skytale-Chiffre) Die älteste bekannte Verschlüsselungstechnik stammt aus der Antike und wurde im 5. Jahrhundert v. Chr. von den Spartanern entwickelt: Der Sender wickelt einen Papierstreifen spiralförmig um einen Holzstab (die sogenannte **Skytale**) und beschreibt ihn in Längsrichtung mit der Geheimbotschaft.



ÜBERAUS GEHEIMNISVOLL...
$\leadsto$ *ÜGI...BES...EHV...REO...AIL...UML...SN...*

Besitzt der Empfänger eines auf diese Weise beschrifteten Papierstreifens einen Stab mit dem gleichen Umfang, so kann er ihn auf dieselbe Art wieder entziffern.

Als Schlüssel fungiert hier also der Stabumfang bzw. die Anzahl k der Zeilen, mit denen der Stab beschrieben wird. Findet der gesamte Klartext x auf der Skytale Platz und beträgt seine Länge ein Vielfaches von k, so geht x bei der Chiffrierung in den Kryptotext

$$E(k, x_1 \cdots x_{km}) = x_1 x_{m+1} x_{2m+1} \cdots x_{(k-1)m+1} x_2 x_{m+2} x_{2m+2} \cdots x_{(k-1)m+2} \cdots x_m x_{2m} x_{3m} \cdots x_{km}$$

über. Dasselbe Resultat stellt sich ein, wenn wir x zeilenweise in eine $k \times m$-Matrix schreiben und spaltenweise wieder auslesen (sogenannte **Spaltentransposition**):

$$\begin{array}{|llll|}
\hline
x_1 & x_2 & \cdots & x_m \\
x_{m+1} & x_{m+2} & \cdots & x_{2m} \\
x_{2m+1} & x_{2m+2} & \cdots & x_{3m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{(k-1)m+1} & x_{(k-1)m+2} & \cdots & x_{km} \\
\hline
\end{array}$$

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

Von der Methode, die letzte *Zeile* nur zum Teil zu füllen, ist dagegen abzuraten. In diesem Fall würden nämlich auf dem abgewickelten Papierstreifen Lücken entstehen, aus deren Anordnung man Schlüsse auf den benutzten Schlüssel k ziehen könnte. Andererseits ist nichts dagegen einzuwenden, dass der Sender die letzte *Spalte* der Skytale nur zum Teil beschriftet.

Bevor wir weitere Beispiele für Transpositionen betrachten, wenden wir uns der Klassifikation von Substitutionschiffren zu. Ein wichtiges Unterscheidungsmerkmal ist z.B. die Länge der Klartexteinheiten, auf denen die Chiffre operiert.

Monographische Substitutionen ersetzen Einzelbuchstaben.

Polygraphische Substitutionen ersetzen dagegen aus mehreren Zeichen bestehende Klartextsegmente auf einmal.

Eine polygraphische Substitution, die auf Buchstabenpaaren operiert, wird **digraphisch** genannt. Das älteste bekannte polygraphische Chiffrierverfahren wurde von Giovanni Porta im Jahr 1563 veröffentlicht. Dabei werden je zwei aufeinanderfolgende Klartextbuchstaben durch ein einzelnes Kryptotextzeichen ersetzt.

Beispiel 1.38 Bei der **Porta-Chiffre** werden 400 (!) unterschiedliche von Porta für diesen Zweck entworfene Kryptotextzeichen verwendet. Diese sind in einer 20×20-Matrix $M = (y_{ij})$ angeordnet, deren Zeilen und Spalten mit den 20 Klartextbuchstaben $\mathtt{A}, \ldots, \mathtt{I}, \mathtt{L}, \ldots, \mathtt{T}, \mathtt{V}, \mathtt{Z}$ indiziert sind. Zur Ersetzung des Buchstabenpaars $a_i a_j$ wird das in Zeile i und Spalte j befindliche Kryptotextzeichen

$$E(M, a_i a_j) = y_{ij}$$

benutzt.

Eine Substitution heißt **monopartit**, falls sie die Klartextsegmente durch Einzelzeichen ersetzt, sonst **multipartit**. Wird der Kryptotext aus Buchstabenpaaren zusammengesetzt, so spricht man von einer **bipartiten** Substitution.

Ein frühes (monographisches) Beispiel einer bipartiten Chiffriermethode geht auf Polybios (circa 200 – 120 v. Chr.) zurück:

M	*0*	*1*	*2*	*3*	*4*
0	A	B	C	D	E
1	F	G	H	I	J
2	K	L	M	N	O
3	P	Q	R	S	T
4	U	V	W	X/Y	Z

$\boxed{\mathtt{POLYBIOS} \rightsquigarrow \mathit{30\,24\,21\,43\,01\,13\,24\,33}}$

Bei der **Polybios-Chiffre** dient eine 5×5-Matrix, die aus sämtlichen Klartextbuchstaben gebildet wird, als Schlüssel.[§] Die Verschlüsselung des Klartextes erfolgt buchstabenweise, indem man einen in Zeile i und Spalte j eingetragenen Klartextbuchstaben durch das Koordinatenpaar ij ersetzt. Der Kryptotextraum besteht also aus den Ziffernpaaren $\{\mathit{00}, \mathit{01}, \ldots, \mathit{44}\}$.

[§]Da nur 25 Plätze zur Verfügung stehen, muss bei Benutzung des lateinischen Alphabets entweder ein Buchstabe weggelassen oder ein Platz mit zwei Buchstaben besetzt werden.

Die Frage, ob bei der Ersetzung der einzelnen Segmente des Klartextes eine einheitliche Strategie verfolgt wird oder ob diese von Segment zu Segment verändert wird, führt uns auf ein weiteres wichtiges Unterscheidungsmerkmal bei Substitutionen.

Monoalphabetische Substitutionen ersetzen die einzelnen Klartextsegment unabhängig von ihrer Position im Klartext.

Polyalphabetische Substitutionen verwenden dagegen eine variable Ersetzungsregel, auf die sich auch die bereits verarbeiteten Klartextsegmente auswirken.

Die Bezeichnung „monoalphabetisch“ bringt zum Ausdruck, dass der Ersetzungsmechanismus auf einem einzelnen Alphabet beruht (sofern wir das Klartextalphabet als bekannt voraussetzen). Die von Caesar benutzte Chiffriermethode kann beispielsweise vollständig durch Angabe des Ersetzungsalphabets

$$\{\texttt{D,E,F,G,W,...,Y,Z,A,B,C}\}$$

beschrieben werden. Auch im Fall, dass nicht einzelne Zeichen, sondern ganze Buchstabengruppen auf einmal ersetzt werden, genügt im Prinzip ein einzelnes Alphabet zur Beschreibung. Hierzu sortiert man die Klartexteinheiten, auf denen der Ersetzungsmechanismus operiert, und bildet die Folge (sprich: das Alphabet) der zugeordneten Kryptotextsegmente.

Monoalphabetische Chiffrierverfahren ersetzen meist Texteinheiten einer festen Länge $l \geq 1$ durch Kryptotextsegmente derselben Länge.

Definition 1.39 (Blockchiffre) *Sei A ein beliebiges Alphabet und es gelte $M = C = A^l$, $l \geq 1$. Eine **Blockchiffre** realisiert für jeden Schlüssel $k \in K$ eine bijektive Abbildung g auf A^l und es gilt*

$$E(k,x) = g(x) \text{ und } D(k,y) = g^{-1}(y)$$

*für alle $x \in M$ und $y \in C$. Im Fall $l = 1$ spricht man auch von einer **einfachen Substitutionschiffre**.*

Polyalphabetische Substitutionen greifen im Wechsel auf verschiedene Ersetzungsalphabete zurück, so dass unterschiedliche Vorkommen eines Zeichens (oder einer Zeichenkette) auch auf unterschiedliche Art ersetzt werden können. Welches Ersetzungsalphabet wann an der Reihe ist, wird dabei in Abhängigkeit von der Länge oder der Gestalt des bereits verarbeiteten Klartextes bestimmt.

Fast alle polyalphabetischen Chiffrierverfahren operieren – genau wie monoalphabetische Substitutionen – auf Klartextblöcken einer festen Länge l, die sie in Kryptotextblöcke einer festen Länge l' überführen, wobei meist $l = l'$ ist. Da diese Blöcke jedoch vergleichsweise kurz sind, kann der Klartext der Chiffrierfunktion ungepuffert zugeführt werden. Man nennt die einzelnen Klartextblöcke in diesem Zusammenhang auch nicht ‚Blöcke' sondern ‚Zeichen' und spricht von **sequentiellen Chiffren** oder von **Stromchiffren**.

Definition 1.40 (Stromchiffre) *Sei A ein beliebiges Alphabet und sei $M = C = A^l$ für eine natürliche Zahl $l \geq 1$. Weiterhin seien K und $\hat{K}$ Schlüsselräume. Eine **Stromchiffre** wird durch eine Verschlüsselungsfunktion $E : \hat{K} \times M \to C$ und einen Schlüsselstromgenerator $g : K \times A^* \to \hat{K}$ beschrieben. Der Generator g erzeugt aus einem externen Schlüssel $k \in K$ für einen Klartext $x = x_0 \ldots x_{n-1}$, $x_i \in M$, eine Folge $\hat{k}_0, \ldots, \hat{k}_{n-1}$ von internen Schlüsseln $\hat{k}_i = g(k, x_0 \ldots x_{i-1}) \in \hat{K}$, unter denen x in den Kryptotext*

$$E_g(k, x) = E(\hat{k}_0, x_0) \ldots E(\hat{k}_{n-1}, x_{n-1})$$

überführt wird.

Der interne Schlüsselraum kann also wie bei der Blockchiffre eine maximale Größe von $(m^l)!$ annehmen (im häufigen Spezialfall $l = 1$ also $m!$). Die Aufgabe des Schlüsselstromgenerators g besteht darin, aus dem externen Schlüssel k und dem bereits verarbeiteten Klartext $x_0 \ldots x_{i-1}$ den aktuellen internen Schlüssel $\hat{k}_i$ zu berechnen. Die bisher betrachteten Stromchiffren benutzen z.B. die folgenden Schlüsselstromgeneratoren.

Stromchiffre	Chiffrierfunktionen	Schlüsselstromgenerator
Vigenère	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \ldots k_{d-1}, x_0 \ldots x_{i-1}) = k_{(i \bmod m)}$
Beaufort	$E(\hat{k}, x) = \hat{k} - x$	$g(k_0 \ldots k_{d-1}, x_0 \ldots x_{i-1}) = k_{(i \bmod m)}$
Autokey mit Klartext-Schlüsselstrom	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \ldots k_{d-1}, x_0 \ldots x_{i-1}) = \begin{cases} k_i, & i < d \\ x_{i-d}, & i \geq d \end{cases}$
Autokey mit Kryptotext-Schlüsselstrom	$E(\hat{k}, x) = x + \hat{k}$	$g(k_0 \ldots k_{d-1}, x_0 \ldots x_{i-1}) = \begin{cases} k_i, & i < d \\ y_{i-d}, & i \geq d \end{cases}$ $= k_{(i \bmod d)} + \sum_{j=1}^{\lfloor i/d \rfloor} x_{i-jd}$

Bei der Vigenère- und Beaufortchiffre hängt der Schlüsselstrom nicht vom Klartext, sondern nur vom externen Schlüssel k ab, d.h. sie sind **synchron**. Die *Autokey*-Chiffren sind dagegen **asynchron** (und aperiodisch).

Gespreizte Substitutionen

Bei den bisher betrachteten Substitutionen haben die einzelnen Blöcke, aus denen der Kryptotext zusammengesetzt wird, eine einheitliche Länge. Es liegt nahe, einem Gegner die unbefugte Rekonstruktion des Klartextes dadurch zu erschweren, dass man Blöcke unterschiedlicher Länge verwendet. Man spricht hierbei auch von einer **Spreizung** (*straddling*) des Kryptotextalphabets. Ein bekanntes Beispiel für diese Technik ist die sogenannte Spionage-Chiffre, die vorzugsweise von der ehemaligen sowjetischen Geheimpolizei NKWD (*Naródny Komissariàt Wnutrennich Del*; zu deutsch: Volkskommissariat des Innern) benutzt wurde.

Beispiel 1.41 Bei der **Spionage-Chiffre** wird in die erste Zeile einer 3×10-Matrix ein Schlüsselwort w geschrieben, welches keinen Buchstaben mehrfach enthält und eine Länge von 6 bis 8 Zeichen hat (also zum Beispiel *SPIONAGE*). Danach werden die anderen beiden Zeilen der Matrix mit den restlichen Klartextbuchstaben (etwa in alphabetischer Reihenfolge) gefüllt.

	4	1	9	6	0	3	2	7	5	8
	S	*P*	*I*	*O*	*N*	*A*	*G*	*E*		
8	B	C	D	F	H	J	K	L	M	Q
5	R	T	U	V	W	X	Y	Z		

GESPREIZT
$\rightsquigarrow$ *274154795751*

Man überzeugt sich leicht davon, dass sich die von der Spionage-Chiffre generierten Kryptotexte wieder eindeutig dechiffrieren lassen, da die Kryptotextsegmente 1, 2, ..., 8, 01, 02, ..., 08, 91, 92, ..., 98, die für die Klartextbuchstaben eingesetzt werden, die **Fano-Bedingung** erfüllen: Keines von ihnen bildet den Anfang eines anderen. Da die Nummern 5 und 8 der beiden letzten Spalten der Matrix auch als Zeilennummern verwendet werden, liefert dies auch eine Erklärung dafür, warum keine Schlüsselwortbuchstaben in die beiden letzten Spalten eingetragen werden dürfen.

Verwendung von Blendern und Homophonen

Die Verwendung von gespreizten Chiffren zielt offenbar darauf ab, die „**Fuge**“ zwischen den einzelnen Kryptotextsegmenten, die von unterschiedlichen Klartextbuchstaben herrühren, zu verdecken, um dem Gegner eine unbefugte Dechiffrierung zu erschweren. Dennoch bietet die Spionage-Chiffre noch genügend Angriffsfläche, da im Klartext häufig vorkommende Wortmuster auch im Kryptotext zu Textwiederholungen führen.

Eine Möglichkeit, diese Muster aufzubrechen, besteht darin, Blender in den Klartext einzustreuen. Abgesehen davon, dass das Entfernen der Blender auch für den rechtmäßigen Empfänger mit Mühe verbunden ist, muss für den Zugewinn an Sicherheit auch mit einer Expansion des Kryptotextes bezahlt werden.

Ist man bereit, dies in Kauf zu nehmen, so gibt es auch noch eine wirksamere Methode, die Übertragung struktureller und statistischer Klartextmerkmale auf den Kryptotext abzumildern. Die Idee dabei ist, zur Chiffrierung der einzelnen Klartextzeichen a nicht nur jeweils eines, sondern eine Menge $H(a)$ von Chiffrezeichen vorzusehen, und daraus für jedes Vorkommen von a im Klartext eines auszuwählen (am besten zufällig). Da alle Zeichen in $H(a)$ für dasselbe Klartextzeichen stehen, werden sie auch **Homophone** genannt.

Definition 1.42 (homophonen Substitutionschiffre) *Sei A ein Klartextalphabet und sei $M = A$. Weiter sei C ein Kryptotextraum der Größe $\|C\| > \|A\| = m$. In einer (einfachen)* ***homophonen Substitutionschiffre*** *beschreibt jeder Schlüssel $k \in K$ eine Zerlegung von C in m disjunkte Mengen $H(a)$, $a \in A$.*

Um ein Zeichen $a \in A$ unter k zu chiffrieren, wird nach einer bestimmten Methode ein Homophon y aus der Menge $H(a)$ gewählt und für a eingesetzt.

Durch den Einsatz einer homophonen Substitution wird also erreicht, dass verschiedene Vorkommen eines Klartextzeichens auch auf unterschiedliche Weise ersetzt werden können. Damit der Empfänger den Kryptotext auch wieder eindeutig dechiffrieren kann, dürfen sich die Homophonmengen zweier verschiedener Klartextzeichen aber nicht überlappen. Daher kann es nicht vorkommen, dass zwei verschiedene Klartextbuchstaben durch dasselbe Geheimtextzeichen ersetzt werden. Man beachte, dass der Chiffriervorgang $x \mapsto E(k, x)$ nicht durch eine Funktion beschreibbar ist, da derselbe Klartext x in mehrere verschiedene Kryptotexte y übergehen kann.

Durch eine geringfügige Modifikation der Polybios-Chiffre lässt sich die folgende bipartite homophone Chiffre erhalten.

Beispiel 1.43 (homophone Substitution) Sei $A = \{\mathtt{A}, \dots, \mathtt{Z}\}$, $B = \{\mathit{0}, \dots, \mathit{9}\}$ und $C = \{\mathit{00}, \dots, \mathit{99}\}$.

M	1,0	2,9	3,8	4,7	5,6
1,6	A	F	K	P	U
2,7	B	G	L	Q	V
3,8	C	H	M	R	W
4,9	D	I	N	S	X/Y
5,0	E	J	O	T	Z

HOMOPHON $\rightsquigarrow$ *82 03 88 53 17 32 08 98*

Genau wie bei Polybios wird eine 5×5-Matrix M als Schlüssel benutzt. Die Zeilen und Spalten von M sind jedoch nicht nur mit jeweils einer, sondern mit zwei Ziffern versehen, so dass jeder Klartextbuchstabe x über vier verschiedene Koordinatenpaare ansprechbar ist. Der Kryptotextraum wird durch M also in 25 Mengen $H(a)$, $a \in A$, mit je 4 Homophonen partitioniert.

Wie wir noch sehen werden, sind homophone Chiffrierungen auch deshalb schwerer zu brechen, weil durch sie die charakteristische Häufigkeitsverteilung der Klartextbuchstaben zerstört wird. Dieser Effekt kann dadurch noch verstärkt werden, dass man für häufig vorkommende Klartextzeichen a eine entsprechend größere Menge $H(a)$ an Homophonen vorsieht. Damit lässt sich erreichen, dass die Verteilung der im Geheimtext auftretenden Zeichen weitgehend nivelliert wird.

Beispiel 1.44 (homophone Substitution, verbesserte Version) Ist $p(a)$ die Wahrscheinlichkeit, mit der ein Zeichen $a \in A$ in der Klartextsprache auftritt, so sollte $\|H(a)\| \approx 100 \cdot p(a)$ sein.

a	$p(a)$	$H(a)$
A	0.0647	$\{15, 26, 44, 59, 70, 79\}$
B	0.0193	$\{01, 84\}$
C	0.0268	$\{13, 28, 75\}$
D	0.0483	$\{02, 17, 36, 60, 95\}$
E	0.1748	$\{04, 08, 12, 30, 43, 46, 47, 53, 61, 67, 69, 72, 80, 86, 90, 92, 97\}$
$\vdots$	$\vdots$	$\vdots$

Da der Buchstabe A im Deutschen beispielsweise mit einer Wahrscheinlichkeit von $p(\mathsf{A}) = 0{,}0647$ auftritt, sind für ihn sechs verschiedene Homophone vorgesehen.

Um den Suchaufwand bei der Dechiffrierung zu reduzieren, empfiehlt es sich, eine 10×10-Matrix anzulegen, in der jeder Klartextbuchstabe a an allen Stellen vorkommt, deren Koordinaten in $H(a)$ enthalten sind.

M'	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*
1	N	E	C	S	A	O	D	X	I	N
2	R	G	S	N	N	A	U	C	H	Y
3	T	L	I	O	U	D	Z	M	N	E
4	H	R	E	A	N	E	E	S	I	T
5	N	I	E	T	P	H	S	L	A	R
6	E	U	M	F	R	J	E	N	E	D
7	N	E	K	S	C	T	I	T	A	A
8	H	N	I	B	R	E	U	G	V	E
9	T	E	L	S	D	R	E	O	S	E
0	B	D	W	E	Q	I	F	E	I	R

HOMOPHON $\rightsquigarrow$ *56 98 63 34 55 29 16 68*

Offenbar kann man diese Matrix auch zur Chiffrierung benutzen, was sogar den positiven Nebeneffekt hat, dass dadurch eine zufällige Wahl der Homophone begünstigt wird.

Transpositionen

Eng verwandt mit der Skytale-Chiffre ist die Zick-Zack-Transposition.

Beispiel 1.45 Bei Ausführung einer ***Zick-Zack-Transposition*** wird der Klartext in eine Zick-Zack-Linie geschrieben und horizontal wieder ausgelesen. Die Höhe der Zick-Zack-Linie kann als Schlüssel vereinbart werden.

```
Z       Z       L       E
  I   K   A   K   I   I
    C       C       N
```

$\boxed{\texttt{ZICKZACKLINIE} \rightsquigarrow \textit{ZZLEIKAKIICCN}}$

Bei einer Zick-Zack-Transposition werden Zeichen im vorderen Klartextbereich bis fast ans Ende des Kryptotextes verlagert und umgekehrt. Dies hat den Nachteil, dass für die Generierung des Kryptotextes der gesamte Klartext gepuffert werden muss. Daher werden meist **Blocktranspositionen** verwendet, bei denen die Zeichen nur innerhalb fester Blockgrenzen transponiert werden.

Definition 1.46 (Blocktranspositionschiffre) *Sei $A = B$ ein beliebiges Alphabet und für eine natürliche Zahl $l \geq 2$ sei $M = C = A^l$. Bei einer* ***Blocktranspositionschiffre*** *wird durch jeden Schlüssel $k \in K$ eine Permutation π beschrieben, so dass für alle Zeichenfolgen $x_1 \cdots x_l \in M$ und $y_1 \cdots y_l \in C$*

$$E(k, x_1 \cdots x_l) = x_{\pi(1)} \cdots x_{\pi(l)}$$

und

$$D(k, y_1 \cdots y_l) = y_{\pi^{-1}(1)} \cdots y_{\pi^{-1}(l)}$$

gilt.

Eine Blocktransposition mit Blocklänge l lässt sich durch eine Permutation $\pi \in S_l$ (also auf der Menge $\{1, \ldots, l\}$) beschreiben.

Beispiel 1.47 Eine Skytale, die mit 4 Zeilen der Länge 6 beschrieben wird, realisiert beispielsweise folgende Blocktransposition:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$\pi(i)$	1	7	13	19	2	8	14	20	3	9	15	21	4	10	16	22	5	11	17	23	6	12	18	24

Für die Entschlüsselung muss die zu π **inverse Permutation** π^{-1} benutzt werden.

Beispiel 1.48

i	1	2	3	4	5	6
$\pi(i)$	4	6	1	3	5	2

i	1	2	3	4	5	6
$\pi^{-1}(i)$	3	6	4	1	5	2

Wird π durch Zyklen $(i_1\ i_2\ i_3\ \dots\ i_n)$ dargestellt, wobei i_1 auf i_2, i_2 auf i_3 usw. und schließlich i_3 auf i_1 abgebildet wird, so ist π^{-1} sehr leicht zu bestimmen.

Beispiel 1.49 Obiges π hat beispielsweise die Zyklendarstellung

$$\pi = (1\ 4\ 3)\ (2\ 6)\ (5) \text{ oder } \pi = (1\ 4\ 3)\ (2\ 6),$$

wenn, wie allgemein üblich, Einerzklen weggelassen werden. Daraus erhalten wir unmittelbar π^{-1} zu

$$\pi^{-1} = (3\ 4\ 1)\ (6\ 2) \text{ oder } (1\ 3\ 4)\ (2\ 6),$$

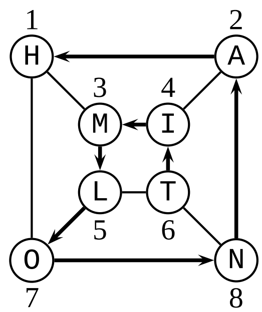
wenn wir jeden Zyklus mit seinem kleinsten Element beginnen lassen und die Zyklen nach der Größe dieser Elemente anordnen.

Beispiel 1.50 Bei der ***Matrix-Transposition*** wird der Klartext zeilenweise in eine $k \times m$-Matrix eingelesen und der Kryptotext spaltenweise gemäß einer Spaltenpermutation π, die als Schlüssel dient, wieder ausgelesen. Für $\pi = (1\ 4\ 3)\ (2\ 6)$ wird also zuerst Spalte $\pi(1) = 4$, dann Spalte $\pi(2) = 6$ und danach Spalte $\pi(3) = 1$ usw. und zuletzt Spalte $\pi(6) = 2$ ausgelesen.

3	6	4	1	5	2
D	I	E	S	E	R
K	L	A	R	T	E
X	T	I	S	T	N
I	C	H	T	S	E
H	R	L	A	N	G

DIESER KLARTEXT IST NICHT SEHR LANG
$\rightsquigarrow$ *SRSTA RENEG DKXIH EAIHL ETTSN ILTCR*

Beispiel 1.51 Bei der ***Weg-Transposition*** wird als Schlüssel eine Hamiltonlinie in einem Graphen mit einer vorgegebenen Knotennummerierung benutzt. (Eine Hamiltonlinie ist eine Anordnung aller Knoten des Graphen, in der je zwei aufeinanderfolgende Knoten durch eine Kante verbunden sein müssen.) Der Klartext wird gemäß der Knotennumerierung in den Graphen eingelesen und der Kryptotext entlang der Hamiltonlinie wieder ausgelesen.



HAMILTON $\rightsquigarrow$ *TIMLONAH*

Es ist leicht zu sehen, dass sich jede Blocktransposition durch eine Hamiltonlinie in einem geeigneten Graphen realisieren lässt. Der Vorteil, eine Hamiltonlinie als Schlüssel zu benutzen, besteht offenbar darin, dass man sich den Verlauf einer Hamiltonlinie bildhaft vorstellen und daher besser einprägen kann als eine Zahlenfolge.

Sehr beliebt ist auch die Methode, eine Permutationen in Form eines **Schlüsselworts** (oder einer aus mehreren Wörtern bestehenden **Schlüsselphrase**) im Gedächtnis zu behalten. Aus einem solchen Schlüsselwort lässt sich die zugehörige Permutation σ leicht rekonstruieren, indem man das Wort auf Papier schreibt und in der Zeile darunter für jeden einzelnen Buchstaben seine Position i innerhalb des Wortes vermerkt.

Schlüsselwort für σ	*C A E S A R*
i	1 2 3 4 5 6
$\sigma(i)$	3 1 4 6 2 5
Zyklendarstellung von σ	(1 3 4 6 5 2)

DIE BLOCKLAENGE IST SECHS $\rightsquigarrow$
EDBOIL LCANKE IGSSET EXCSYH

Die Werte $\sigma(i)$, die σ auf diesen Nummern annimmt, werden nun dadurch ermittelt, dass man die Schlüsselwort-Buchstaben in alphabetischer Reihenfolge durchzählt. Dabei werden mehrfach vorkommende Buchstaben gemäß ihrer Position im Schlüsselwort an die Reihe genommen. Alternativ kann man auch alle im Schlüsselwort wiederholt vorkommenden Buchstaben streichen, was im Fall des Schlüsselworts *CAESAR* auf eine Blocklänge von 5 führen würde.

1.8 Realisierung von Blocktranspositionen und einfachen Substitutionen

Abschließend möchten wir eine einfache elektronische Realisierungsmöglichkeit von Blocktranspositionen erwähnen, die auf binär kodierten Klartexten operieren

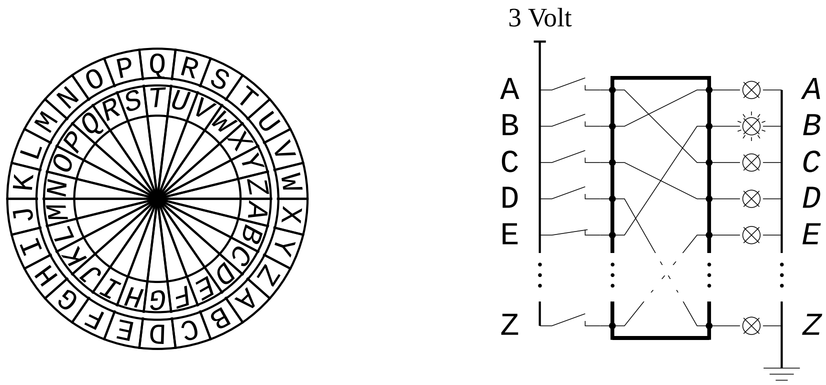
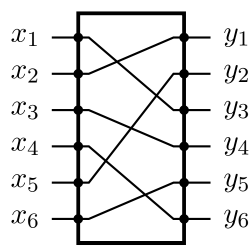


Abbildung 1.1: Realisierung von einfachen Substitutionen mit einer Drehscheibe und mit Hilfe von Steckverbindungen.

(d.h. $A = \{0, 1\}$). Um einen Binärblock $x_1 \cdots x_l$ der Länge l zu permutieren, müssen die einzelnen Bits lediglich auf l Leitungen gelegt und diese gemäß π in einer sogenannten **Permutationsbox** (kurz **P-Box**) vertauscht werden.



Die Implementierung einer solchen P-Box kann beispielsweise auf einem VLSI-Chip erfolgen. Allerdings kann hierbei für größere Werte von l aufgrund der hohen Zahl von Überkreuzungspunkten ein hoher Flächenbedarf anfallen.

Blocktranspositionen können auch leicht durch Software als eine Folge von Zuweisungen

$$\mathsf{Y1} := \mathsf{X2}; \quad \mathsf{Y2} := \mathsf{X5}; \quad \ldots \quad \mathsf{Y6} := \mathsf{X4};$$

implementiert werden. Bei großer Blocklänge und sequentieller Abarbeitung erfordert diese Art der Implementierung jedoch einen relativ hohen Zeitaufwand.

Von Alberti stammt die Idee, das Klartext- und Kryptotextalphabet auf zwei konzentrischen Scheiben unterschiedlichen Durchmessers anzuordnen. In Abbildung 1.1 ist gezeigt, wie sich mit einer solchen Drehscheibe beispielsweise die additive Chiffre realisieren lässt. Zur Einstellung des Schlüssels k müssen die Scheiben so gegeneinander verdreht werden, dass der Schlüsselbuchstabe a_k auf der inneren Scheibe mit dem Klartextzeichen $a_0 = \mathsf{A}$ auf der äußeren Scheibe zur Deckung kommt. Auf der Drehscheibe in Abbildung 1.1 ist beispielsweise der Schlüssel $k = 3$ eingestellt, das heißt, $a_k = \mathit{D}$. Die Verschlüsselung geschieht nun durch bloßes Ablesen der zugehörigen Kryptotextzeichen auf der inneren Scheibe, so dass von der Drehfunktion der Scheiben nur bei einem Schlüsselwechsel Gebrauch gemacht wird.

Aufgrund ihrer engen Verwandtschaft mit der Klasse der Blocktranspositionen lassen sich einfache Substitutionen auch mit Hilfe einer P-Box realisieren (vergleiche Abbildung). Hierfür können beispielsweise zwei Steckkontaktleisten verwendet werden. Der aktuelle Schlüssel wird in diesem Fall durch Verbinden der entsprechenden Kontakte mit elektrischen Kabeln eingestellt (siehe Abbildung 1.1). Um etwa den Klartextbuchstaben E zu verschlüsseln, drückt man auf die entsprechende Taste, und das zugehörige Kryptotextzeichen *B* wird im selben Moment durch ein aufleuchtendes Lämpchen signalisiert.

Schließlich lassen sich Substitutionen auch leicht durch Software realisieren. Hierzu wird ein Feld (*array*) deklariert, dessen Einträge über die Klartextzeichen $x \in A$ adressierbar sind. Das mit x indizierte Feldelemente enthält das Kryptotextzeichen, durch welches x beim Chiffriervorgang zu ersetzen ist.

Ein Nachteil hierbei ist, dass das Feld nach jedem Schlüsselwechsel neu beschrieben werden muss. Um dies zu umgehen, kann ein zweidimensionales Feld deklariert werden, dessen Einträge zusätzlich über den aktuellen Schlüsselwert k adressierbar sind. Ist genügend Speicherplatz vorhanden, um für alle $x \in A$ und alle $k \in K$ die zugehörigen Kryptotextzeichen $E(k, x)$ abspeichern zu können, so braucht das Feld nur einmal initialisiert und danach nicht mehr geändert werden.

Schlüsselwert	**Klartextbuchstabe**			
	A	B	...	Z
0	*U*	*H*	...	*C*
1	*E*	*H*	...	*A*
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
63	*Y*	*F*	...	*W*

Die Tabelle zeigt ein Feld, dessen Einträge mit (zufällig gewählten) Kryptotextzeichen $E(k, x) \in B = \{\mathit{A}, \dots, \mathit{Z}\}$ initialisiert wurden, wobei k einen beliebigen Wert in dem Schlüsselraum $K = \{0, 1, \dots, 63\}$ annehmen kann und x alle Klartextbuchstaben des Alphabets $A = \{\mathsf{A}, \dots, \mathsf{Z}\}$ durchläuft.

2 Kryptoanalyse der klassischen Verfahren

2.1 Klassifikation von Angriffen gegen Kryptosysteme

Die Erfolgsaussichten eines Angriffs gegen ein Kryptosystem hängen sehr stark davon ab, wie gut die Ausgangslage ist, in der sich der Gegner befindet. Prinzipiell sollte man die Fähigkeiten des Gegners genauso wenig unterschätzen wie die Unvorsichtigkeit der Anwender von Kryptosystemen. Bereits vor mehr als einem Jahrhundert postulierte Kerckhoffs, dass die Frage der Sicherheit keinesfalls von irgendwelchen obskuren Annahmen über den Wissensstand des Gegners abhängig gemacht werden darf.

Goldene Regel für Kryptosystem-Designer (Kerckhoffs' Prinzip)
: *Unterschätze niemals den Kryptoanalytiker. Gehe insbesondere immer von der Annahme aus, dass dem Gegner das angewandte System bekannt ist.*[*]

In der folgenden Liste sind eine Reihe von Angriffsszenarien mit zunehmender Gefährlichkeit aufgeführt. Auch wenn nicht alle Eventualitäten eines Angriffs vorhersehbar sind, so vermittelt diese Aufstellung doch eine gute Vorstellung davon, welchen unterschiedlichen Bedrohungen ein Kryptosystem im praktischen Einsatz ausgesetzt sein kann.

Angriff bei bekanntem Kryptotext (*ciphertext-only attack*)
: Der Gegner fängt Kryptotexte ab und versucht, allein aus ihrer Kenntnis Rückschlüsse auf die zugehörigen Klartexte oder auf die benutzten Schlüssel zu ziehen.

Angriff bei bekanntem Klartext (*known-plaintext attack*)
: Der Gegner ist im Besitz von einigen zusammengehörigen Klartext-Kryptotext-Paaren. Hierdurch wird erfahrungsgemäß die Entschlüsselung

[*]Diese Annahme ergibt sich meist schon aus der Tatsache, dass die Prinzipien fast aller heute im Einsatz befindlichen Kryptosysteme allgemein bekannt sind.

weiterer Kryptotexte oder die Bestimmung der benutzten Schlüssel wesentlich erleichtert.

Angriff bei frei wählbarem Klartext (*chosen-plaintext attack*)
Der Angriff des Gegners wird zusätzlich dadurch erleichtert, dass er in der Lage ist (oder zumindest eine Zeit lang war), sich zu Klartexten seiner Wahl die zugehörigen Kryptotexte zu besorgen. Kann hierbei die Wahl der Kryptotexte in Abhängigkeit von zuvor erhaltenen Verschlüsselungsergebnissen getroffen werden, so spricht man von einem **Angriff bei adaptiv wählbarem Klartext** (***adaptive chosen-plaintext attack***).

Angriff bei frei wählbarem Kryptotext (*chosen-ciphertext attack*)
Vor der Beobachtung des zu entschlüsselnden Kryptotextes konnte sich der Gegner zu Kryptotexten seiner Wahl die zugehörigen Klartexte besorgen, ohne dabei jedoch in den Besitz des Dechiffrierschlüssels zu kommen (**Mitternachtsattacke**). Das dabei erworbene Wissen steht ihm nun bei der Durchführung seines Angriffs zur Verfügung. Auch in diesem Fall können sich die Erfolgsaussichten des Gegners erhöhen, wenn ein **Angriff bei adaptiv wählbarem Kryptotext** (***adaptive chosen-ciphertext attack***) möglich ist, also der Kryptotext in Abhängigkeit von den zuvor erzielten Entschlüsselungsergebnissen wählbar ist.

Angriff bei frei (oder adaptiv) wählbarem Text (*chosen-text attack*)
Sowohl Klartexte als auch Kryptotexte sind frei (oder sogar adaptiv) wählbar.

Ohne Frage ist ein Kryptosystem, das bereits bei einem Angriff mit bekanntem Kryptotext Schwächen erkennen lässt, für den praktischen Einsatz vollkommen ungeeignet. Tatsächlich müssen aber an ein praxistaugliches Kryptosystem noch weit höhere Anforderungen gestellt werden. Denn häufig unterlaufen den Anwendern sogenannte **Chiffrierfehler**, die einen Gegner leicht in eine sehr viel günstigere Ausgangsposition versetzen als dies sonst der Fall wäre. So ermöglicht beispielsweise das Auftreten stereotyper Klartext-Formulierungen einen Angriff bei bekanntem Klartext, sofern der Gegner diese Formulierungen kennt oder auch nur errät. Begünstigt durch derartige Unvorsichtigkeiten, die im praktischen Einsatz nicht vollständig vermeidbar sind, können sich selbst winzige Konstruktionsschwächen eines Kryptosystems sehr schnell zu einer ernsthaften Bedrohung der damit verfolgten Sicherheitsinteressen auswachsen. Die Geschichte der Kryptographie belegt sehr eindrucksvoll, dass es häufig die Anwender eines Kryptosystems selbst sind, die – im unerschütterlichen Glauben an seine kryptographische Stärke – dem Gegner zum Erfolg verhelfen.

Zusammenfassend lässt sich also festhalten, dass die Gefährlichkeit von Angriffen, denen ein Kryptosystem im praktischen Einsatz ausgesetzt ist, kaum zu überschätzen ist. Andererseits kann selbst das beste Kryptosystem keinen Schutz vor einer unbefugten Dechiffrierung mehr bieten, wenn es dem Gegner etwa gelingt, in den Besitz des geheimen Schlüssels zu kommen – sei es aus Unachtsamkeit der Anwender oder infolge einer Gewaltandrohung des Gegners (**kompromittierte Schlüssel**).

2.2 Kryptoanalyse von einfachen Substitutionschiffren

Durch eine Häufigkeitsanalyse können insbesondere einfache Substitutionen g leicht gebrochen werden, sofern die einzelnen Buchstaben a in der benutzten Klartextsprache mit voneinander differierenden Häufigkeiten $p(a)$ auftreten (vergleiche Tabelle 2.1). Selbst wenn, was insbesondere bei kurzen Texten zu erwarten ist, die tatsächliche Häufigkeitsverteilung nur in etwa der vom Gegner angenommenen Verteilung entspricht, reduziert sich dadurch die Zahl der in Frage kommenden einfachen Substitutionen ganz erheblich. Berechnet man die relativen Häufigkeiten h der Kryptotextbuchstaben im Kryptotext, so gilt $p(a) \approx h(g(a))$ (vorausgesetzt der Kryptotext ist genügend lang). Für die Schilderung einer nach dieser Methode durchgeführten Kryptoanalse sei auf die Erzählung „Der Goldkäfer“ von Edgar Allan Poe verwiesen.

Tabelle 2.1: Einteilung von Buchstaben in Cliquen mit vergleichbaren Häufigkeitswerten.

	Deutsch	Englisch	Französisch
sehr häufig	E	E	E
häufig	N | I R S | A T	T | A O I N | S R H	N | A R S I T U
durchschnittlich	D H U | L G O | C M	L D | C U M F	L D | C M P
selten	B F W K Z | P V	P G W Y B | V K	V | F B G Q H X
sehr selten	J Y X Q	X J Q Z	J Y Z K W

Manche der bisher betrachteten Chiffrierverfahren verwenden einen so kleinen Schlüsselraum, dass ohne großen Aufwand eine vollständige Schlüsselsuche ausgeführt werden kann.

Beispiel 2.1 (vollständige Schlüsselsuche) Es sei bekannt, dass das Kryptotextstück $y = \textsf{\textit{SAXP}}$ mit einer additiven Chiffre erzeugt wurde ($K = A = B = A_{lat}$).

Entschlüsseln wir y probeweise mit allen möglichen Schlüsselwerten, so erhalten wir folgende Zeichenketten.

k	B	C	D	E	F	G	H	I	J	K	L	M
$D(k, y)$	RZWO	QYVN	PXUM	OWTL	NVSK	MURJ	LTQI	KSPH	JROG	IQNF	HPME	GOLD
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
FNKC	EMJB	DLIA	CKHZ	BJGY	AIFX	ZHEW	YGDV	XFCU	WEBT	VDAS	UCZR	TBYQ

Unter diesen springen vor allem die beiden Klartextkandidaten x = GOLD (Schlüsselwert $k =$ M) und $x =$ WEBT ($k =$ W) ins Auge.

Ist $s = \|K\|$ die Größe des Schlüsselraums, so kann der Gegner bei bekanntem Kryptotext y die Suche nach dem zugehörigen Klartext x auf eine Menge von maximal s Texten $x_1, \ldots, x_s$ beschränken. Daneben hat der Gegner ein gewisses *a priori* Wissen über den Klartext, wie zum Beispiel dass er in deutscher Sprache verfasst ist, das es ihm gestattet, einen Großteil der Texte x_i auszuschließen. Ferner erscheinen aufgrund dieses Hintergrundwissens manche der übrig gebliebenen Klartextkandidaten plausibler als andere (sofern nicht nur ein einziger übrig bleibt). Mit jedem Text x_i, der nicht als Klartext in Frage kommt, kann auch mindestens ein Schlüssel ausgeschlossen werden. Sind noch mehrere Schlüsselwerte möglich, so kann weiteres Kryptotextmaterial Klarheit bringen. Manchmal hilft aber auch eine Inspektion der verbliebenen Schlüsselwerte weiter, etwa wenn der Schlüssel nicht rein zufällig erzeugt wurde, sondern aus einem einprägsamen Schlüsselwort ableitbar ist.

Meist kennt der Gegner zumindest die Sprache, in der der gesuchte Klartext abgefasst ist. Mit zunehmender Länge gleichen sich die Häufigkeitsverteilungen der Buchstaben in natürlichsprachigen Texten einer „Grenzverteilung“ an, die in erster Linie von der benutzten Sprache und nur in geringem Umfang von der Art des Textes abhängt. Diese Verteilungen weisen typischerweise eine sehr starke Ungleichmäßigkeit auf, was darauf zurückzuführen ist, dass in natürlichen Sprachen relativ viel Redundanz enthalten ist.

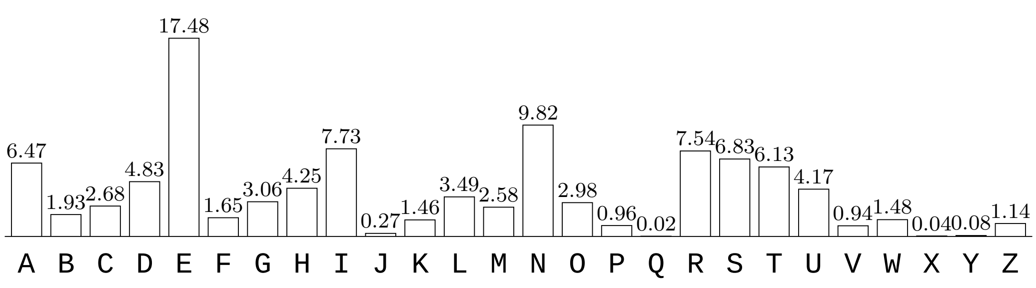


Abbildung 2.1: Häufigkeitsverteilung der Einzelbuchstaben im Deutschen (in %).

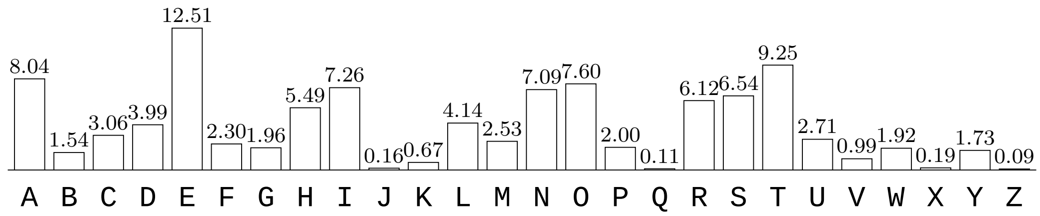


Abbildung 2.2: Häufigkeitsverteilung der Buchstaben im Englischen (in %).

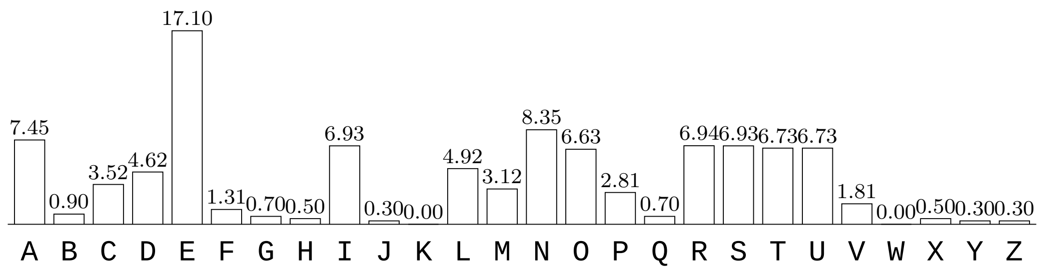


Abbildung 2.3: Häufigkeitsverteilung der Buchstaben im Französischen (in %).

Die Abbildungen 2.1, 2.2 und 2.3, zeigen typische Verteilungen von Einzelbuchstaben in der deutschen, englischen und französischen Sprache (ohne Berücksichtigung von Interpunktions- und Leerzeichen). Ein typischer deutscher Text besteht demnach zu 62% aus den sieben häufigsten Zeichen E, N, I, R, S, A, T (das sind nicht einmal 27% der Klartextzeichen).