Vorlesungsskript

# Einführung in die Kryptologie

Sommersemester 2020

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

*28. Mai 2020*

# 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.

**Steganografische Maßnahmen:** unsichtbare Tinte etc.

Andererseits können durch kryptografische 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 Kryptografie fallen auch die folgenden Begriffe.

**Kryptografie:** 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 kryptografischen Stärken bzw. Schwächen.

**Kryptologie:** Wissenschaft vom Entwurf, der Anwendung und der Analyse von kryptografischen Verfahren (umfasst Kryptografie 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** (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 2.** *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* $A^* = \bigcup_{n \geq 0} A^n$*.*

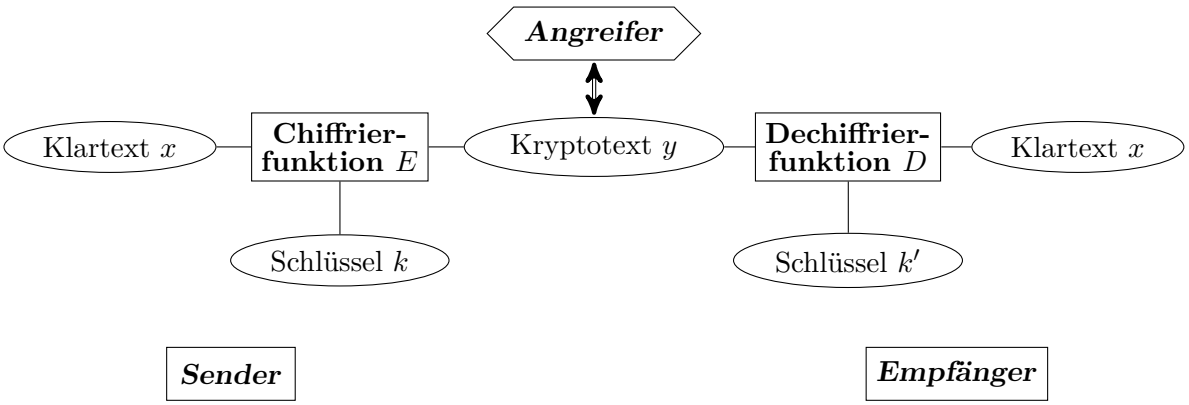
**Beispiel 3.** *Das* ***lateinische Alphabet*** $A_{lat}$ *enthält die 26 Zeichen* `A,...,Z`*. Bei der Abfassung von Klartexten wurde meist auf den Gebrauch von Interpunktions- und Leerzeichen sowie auf Groß- und Kleinschreibung verzichtet (*$\leadsto$ *Verringerung der Redundanz im Klartext).* ◁

**Definition 4.** *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 5.** *Für jedes Paar $(k, k') \in S$ ist die Chiffrierfunktion $E_k$ injektiv.*

*Beweis.* Angenommen, für zwei Klartexte $x_1$ und $x_2$ gilt $E(k, x_1) = E(k, x_2)$. Dann folgt

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

□

## 1.3 Die affine Chiffre

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

**Definition 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*

$$a \equiv_m b \text{ oder } a \equiv b \pmod{m}$$

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

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

*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 $a = dm + r$. Sowohl $r$ als auch $d$ sind hierbei eindeutig bestimmt (siehe Übungen) und die Zahl $d$ wird auch mit $a$ div $m$ bezeichnet.*

Die auf $\mathbb{Z}$ definierten Operationen

$$a \oplus_m b := (a + b) \bmod m \text{ und } a \odot_m b := ab \bmod m$$

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

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* |

$a \oplus_m -b$ schreiben wir auch $a \ominus_m b$. Wenn aus dem Kontext klar ist, dass $a, b \in \mathbb{Z}_m$ sind, schreiben wir anstelle von $a \oplus_m b$, $a \ominus_m b$ und $a \odot_m b$ auch einfach $a + b$, $a - b$ bzw. $ab$.

Durch Identifikation der Zeichen $a_i$ eines Alphabets $A = \{a_0, \ldots, a_{m-1}\}$ mit ihren Indizes können wir die auf $\mathbb{Z}_m$ definierten Rechenoperationen auf Buchstaben übertragen.

**Definition 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*

$$\begin{aligned} a_i + a_j &= a_{i+j}, \quad a_i - a_j = a_{i-j}, \quad a_i a_j = a_{ij}, \\ a_i + z &= a_{i+z}, \quad a_i - z = a_{i-z}, \quad z a_j = a_{zj \bmod m}. \end{aligned}$$

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

**Definition 8.** *Bei der **additiven Chiffre** ist $A = B = M = C$ ein beliebiges Alphabet mit $m := \|A\|$ und $K = \{0, \ldots, m-1\}$. Für $k \in K$, $x \in M$ und $y \in C$ gilt*

$$E(k, x) = x + k \quad \text{und} \quad D(k, y) = y - k.$$

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 1.1). Natürlich kann mit dieser Substitution nicht ernsthaft die Vertraulichkeit von Nachrichten gewahrt 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

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

Da ROT13 zudem keinen Buchstaben auf sich selbst abbildet, ist sie sogar ***echt involutorisch***.

Die Buchstabenrechnung legt folgende Modifikation der Caesar-Chiffre nahe. Anstatt auf jedes Klartextzeichen den Schlüsselwert $k$ zu addieren, können wir die Klartextzeichen 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 das Zeichen $g(\mathtt{A}) = g(\mathtt{N}) = \mathtt{a}$ ab. Um eine hinreichende und notwendige Bedingung für die Zulässigkeit eines Schlüsselwerts $k$ formulieren zu können, führen wir folgende Begriffe ein.

**Definition 9 (ggT, kgV, teilerfremd).** *Seien $a, b \in \mathbb{Z}$. Für $(a, b) \neq (0, 0)$ ist*

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

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

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

*das **kleinste gemeinsame Vielfache** von $a$ und $b$. Ist $\mathrm{ggT}(a, b) = 1$, so nennt man $a$ und $b$ **teilerfremd** oder man sagt, $a$ ist **relativ prim** zu $b$.*

**Lemma 10.** *Seien* $a, b, c \in \mathbb{Z}$ *mit* $(a, b) \neq (0, 0)$*. Dann gilt* $\mathrm{ggT}(a, b) = \mathrm{ggT}(b, a + bc)$ *und somit* $\mathrm{ggT}(a, b) = \mathrm{ggT}(b, a \bmod b)$*, falls* $b \geq 1$ *ist.*

*Beweis.* Jeder Teiler $d$ von $a$ und $b$ ist auch ein Teiler von $b$ und $a + bc$ und umgekehrt. □

**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*):

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

mit

$$r_{i-1} = d_{i+1} r_i + r_{i+1} \text{für } i = 1, \ldots, s.$$

Hierzu sind $s$ Divisionsschritte erforderlich. Wegen

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

folgt $\mathrm{ggT}(a, b) = \mathrm{ggT}(r_s, r_{s+1}) = r_s$.

**Beispiel 11.** *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.

**Prozedur** $\texttt{Euklid}_{\texttt{it}}(a, b)$

```
1 repeat
2     r := a mod b
3     a := b
4     b := r
5 until r = 0
6 return(a)
```

**Prozedur** $\texttt{Euklid}_{\texttt{rek}}(a, b)$

```
1 if b = 0 then
2     return(a)
3 else
4     return(Euklid_rek(b, a mod b))
```

Zur Abschätzung von $s$ verwenden wir die Folge der Fibonacci-Zahlen $F_n$.

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

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

Durch Induktion über $i = s+1, s, \ldots, 0$ folgt $r_i \geq F_{s+1-i}$ und somit $a = r_0 \geq F_{s+1}$. Weiterhin lässt sich durch Induktion über $n \geq 0$ zeigen, dass $F_{n+1} \geq \phi^{n-1}$ ist, wobei $\phi = (1+\sqrt{5})/2$ der *goldene Schnitt* ist. Der Induktionsanfang ($n = 0$ oder 1) ist klar, da $F_2 = F_1 = 1 = \phi^0 \geq \phi^{-1}$ ist. Unter der Induktionsannahme $F_{i+1} \geq \phi^{i-1}$ für $i \leq n-1$ folgt wegen $\phi^2 = \phi + 1$

$$F_{n+1} = F_n + F_{n-1} \geq \phi^{n-2} + \phi^{n-3} = \phi^{n-3}(\phi+1) = \phi^{n-1}.$$

Somit ist $a \geq \phi^{s-1}$, d. h. $s \leq 1 + \lfloor \log_\phi a \rfloor$.

**Satz 12.** *Seien $a > b > 0$ ganze Zahlen und sei $n$ die Länge von $a$ in Binärdarstellung. Dann führt der Euklidsche Algorithmus $O(n)$ Divisionsschritte zur Berechnung von $\mathrm{ggT}(a,b)$ durch. Dies führt auf eine Zeitkomplexität von $O(n^3)$, da jede Ganzzahldivision in Zeit $O(n^2)$ durchführbar ist.*

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

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

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

$$p_i = p_{i-2} - d_i p_{i-1} \text{ (mit } p_0 = 1 \text{ und } p_1 = 0)$$

und

$$q_i = q_{i-2} - d_i q_{i-1} \text{ (mit } q_0 = 0 \text{ und } q_1 = 1$$

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

$$ap_i + bq_i = r_i,$$

und wegen

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

folgt induktiv über $i = 2, \ldots, s$, dass diese Gleichung auch für $i = s$ gilt:

$$ap_s + bq_s = r_s = \mathrm{ggT}(a,b).$$

**Korollar 13** (Lemma von Bezout)**.** *Der größte gemeinsame Teiler von $a$ und $b$ ist in der Form*

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

*darstellbar.*

**Beispiel 14.** *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 | **3** | **$-14$** | $3\cdot 693 - 14\cdot$ | $147 =$ | $21$ |

*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 15.** *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.* Seien $\mu, \lambda \in \mathbb{Z}$ mit $\mu a + \lambda b = \operatorname{ggT}(a,b)$. Falls $x$ sowohl $a$ als auch $b$ teilt, dann teilt $x$ auch die Produkte $\mu a$ und $\lambda b$ und somit auch deren Summe. □

**Korollar 16.** $\operatorname{ggT}(a,b) = \min\{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$.

*Beweis.* Sei $M = \{\lambda a + \mu b \geq 1 \mid \lambda, \mu \in \mathbb{Z}\}$, $m = \min M$ und $g = \operatorname{ggT}(a,b)$. Dann folgt $g \geq m$, da $g$ in der Menge $M$ enthalten ist, und $g \leq m$, da $g$ jede Zahl in $M$ teilt. □

**Korollar 17.** *Zwei Zahlen $a$ und $b$ sind genau dann zu einer Zahl $m \in \mathbb{Z}$ teilerfremd, wenn ihr Produkt $ab$ teilerfremd zu $m$ ist,*

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

*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

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

und Korollar 16, dass auch $ab$ teilerfremd zu $m$ ist.

Gilt umgekehrt $\operatorname{ggT}(ab,m) = 1$, so existieren Zahlen $\mu, \lambda \in \mathbb{Z}$ mit $\mu ab + \lambda m = 1$. Mit Korollar 16 folgt sofort $\operatorname{ggT}(a,m) = \operatorname{ggT}(b,m) = 1$. □

**Korollar 18** (Lemma von Euklid)**.** *Sind $a$ und $b$ teilerfremd und teilt $a$ das Produkt $bc$, so teilt $a$ auch $c$,*

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

*Beweis.* Wegen $\operatorname{ggT}(a,b) = 1$ existieren Zahlen $\mu, \lambda \in \mathbb{Z}$ mit $\mu a + \lambda b = 1$. Falls $a$ das Produkt $bc$ teilt, muss $a$ auch die Zahl $\mu ac + \lambda bc = c$ teilen. □

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

**Satz 19.** *Seien $b, y, m$ ganze Zahlen mit $m \geq 1$. 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, $\operatorname{ggT}(b,m) = g > 1$. Dann ist mit $x$ auch $x' = x + m/g$ eine Lösung von $bx \equiv_m y$ mit $x \not\equiv_m x'$. Folglich ist die Kongruenz $bx \equiv_m y$ nicht eindeutig lösbar.

Gilt umgekehrt $\operatorname{ggT}(b,m) = 1$, 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 $x_1 \equiv_m x_2$. Folglich hat die Kongruenz $bx \equiv_m y$ für jedes $y \in \mathbb{Z}_m$ höchstens eine Lösung $x \in \{0, \ldots, m-1\}$. Zudem folgt, dass die Abbildung $f : \mathbb{Z}_m \to \mathbb{Z}_m$ mit $f(x) = bx \bmod m$ injektiv ist. Da aber der Definitions- und der Wertebereich von $f$ die gleiche Mächtigkeit haben, muss $f$ dann auch surjektiv sein. Somit hat die Kongruenz $bx \equiv_m y$ für jedes $y \in \mathbb{Z}_m$ sogar genau eine Lösung $x \in \{0, \ldots, m-1\}$. □

**Korollar 20.** *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.*

Korollar 17 zeigt, dass die Menge

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

aller invertierbaren Elemente von $\mathbb{Z}_m$ unter der Operation $\odot_m$ abgeschlossen ist. Mit Korollar 20 folgt daher, dass $(\mathbb{Z}_m^*, \odot_m, 1)$ eine multiplikative Gruppe bildet. Allgemeiner zeigt man, dass die Multiplikation eines beliebigen Rings $(R, +, \cdot, 0, 1)$ mit Eins auf der Menge $R^* = \{a \in R \mid \exists b \in R : ab = 1 = ba\}$ aller **Einheiten** von $R$ eine Gruppe bildet (siehe Übungen). Diese Gruppe $(R^*, \cdot, 1)$ wird als **Einheitengruppe** von $R$ bezeichnet.

Das multiplikative Inverse von $b$ modulo $m$ ergibt sich aus der linearen Darstellung $\lambda b + \mu m = \mathrm{ggT}(b, m) = 1$ zu $b^{-1} = \lambda \bmod m$. Die folgende Tabelle gibt für jedes $b \in \mathbb{Z}_{26}^*$ das multiplikative Inverse $b^{-1}$ an.

| $b$ | 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 |

Bei Kenntnis von $b^{-1}$ kann die Kongruenz $bx \equiv_m y$ leicht zu $x = yb^{-1} \bmod m$ gelöst werden.

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

**Definition 21.** *Bei der* ***affinen Chiffre*** *ist* $A = B = M = C$ *ein beliebiges Alphabet mit* $m := \|A\|$ *und* $K = \mathbb{Z}_m^* \times \mathbb{Z}_m$*. Für* $k = (b, c) \in K$*,* $x \in M$ *und* $y \in C$ *gilt*

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

In diesem Fall liefert die Schlüsselkomponente $b = -1$ für jeden Wert von $c \in \mathbb{Z}_m$ eine involutorische Chiffrierfunktion $x \mapsto E_{(-1,c)}(x) = c - x$ (**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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $-x$ | a | 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 |
| $-x-1$ | 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 22** (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 das Klartextzeichen* $x = \mathsf{F}$ *zu verschlüsseln, berechnen wir*

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

*da der Index von* $\mathsf{F}$ *gleich* 5*, der von* $\mathsf{v}$ *gleich* 21 *und* $9 \cdot 5 + 2 = 47 \equiv_{26} 21$ *ist. Um ein Kryptotextzeichen 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 das Kryptotextzeichen* $y = \mathsf{v}$ *das ursprüngliche Klartextzeichen*

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

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

Zur Berechnung der Schlüsselzahl bei der multiplikativen und affinen Chiffre benötigen wir die Funktion

$$\varphi : \mathbb{N} \to \mathbb{N} \quad \text{mit } \varphi(m) = \|\mathbb{Z}_m^*\| = \|\{a \in \mathbb{Z}_m \mid \mathrm{ggT}(a, m) = 1\}\|,$$

die sogenannte *Eulersche* $\varphi$*-Funktion*. Die folgende Tabelle zeigt die Werte $\varphi(m)$ für $m = 1, \ldots, 10$ (für die Menge $\{1, \ldots, n\}$, $n \in \mathbb{N}$, schreiben wir auch kurz $[n]$).

| $m$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbb{Z}_m^*$ | $\{0\}$ | $\{1\}$ | $[2]$ | $\{1,3\}$ | $[4]$ | $\{1,5\}$ | $[6]$ | $\{1,3,5,7\}$ | $\{1,2,4,5,7,8\}$ | $\{1,3,7,9\}$ |
| $\varphi(m)$ | 1 | 1 | 2 | 2 | 4 | 2 | 6 | 4 | 6 | 4 |

Für primes $p$ gilt offensichtlich $\varphi(p) = p - 1$, da $\mathbb{Z}_p^* = [p - 1]$ ist. Wegen

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

folgt zudem

$$\varphi(p^k) = p^k - p^{k-1} = p^{k-1}(p - 1) \text{ für } k \geq 1.$$

Um hieraus für beliebige Zahlen $n \in \mathbb{N}$ eine Formel für $\varphi(n)$ zu erhalten, genügt es, $\varphi(ml)$ im Fall $\mathrm{ggT}(m, l) = 1$ in Abhängigkeit von $\varphi(m)$ und $\varphi(l)$ 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 23.** *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** |

*Die fett gedruckten Einträge bilden dann die Tabelle der Einschränkung $\hat{f}^{-1}$ von $f^{-1}$ auf die Menge $\mathbb{Z}_5^* \times \mathbb{Z}_6^*$. Das Bild dieser Einschränkung ist genau die Menge $\mathbb{Z}_{30}^*$.* ◁

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

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

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

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

**Satz 24.** *Die Eulersche $\varphi$-Funktion ist multiplikativ, d. h. für teilerfremde Zahlen $m$ und $\ell$ gilt $\varphi(m\ell) = \varphi(m)\varphi(\ell)$.*

**Korollar 25.** *Sei $m = \prod_{i=1}^{\ell} p_i^{k_i}$ die Primfaktorzerlegung von $m$. Dann gilt*

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

*Beweis.* Es gilt

$$\varphi(\textstyle\prod_{i=1}^{\ell} p_i^{k_i}) = \prod_{i=1}^{\ell} \varphi(p_i^{k_i}) = \prod_{i=1}^{\ell} (p_i^{k_i} - p_i^{k_i - 1}) = \prod_{i=1}^{\ell} p_i^{k_i - 1}(p_i - 1).$$ □

### 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.

**Satz 26** (Chinesischer Restsatz (CRS))**.** *Falls $m_1, \ldots, m_k$ paarweise teilerfremd sind, dann hat das System*

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

*für beliebige Zahlen $b_1, \ldots, b_k \in \mathbb{Z}$ genau eine Lösung modulo $m = \prod_{i=1}^{k} m_i$.*

*Beweis.* Zu jeder Zahl $n_i = {}^{m}/_{m_i}$ existieren wegen $\mathrm{ggT}(n_i, m_i) = 1$ Zahlen $\mu_i$ und $\lambda_i$ mit

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

Für $i = 1, \ldots, k$ löst daher die Zahl $s_i = \mu_i n_i$ das System

$$x \equiv_{m_j} \begin{cases} 0, & j \neq i \quad (a) \\ 1, & j = i \quad (b) \end{cases} \tag{1.3}$$

Folglich gelten für $s = \sum_{i=1}^{k} b_i s_i$ die Kongruenzen $s \overset{(1.3a)}{\equiv}_{m_j} b_j s_j \overset{(1.3b)}{\equiv}_{m_j} b_j$, d.h. $s$ löst das System (1.2). Dies zeigt, dass die Funktion

$$f : \mathbb{Z}_m \to \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k} \text{ mit } f(x) = (x \bmod m_1, \ldots, x \bmod m_k)$$

surjektiv ist. Da der Definitions- und der Wertebereich von $f$ gleich groß sind, muss $f$ auch injektiv sein und (1.2) ist eindeutig lösbar. □

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

Man verifiziert auch leicht, dass $f$ ein Isomorphismus zwischen dem Ring $(\mathbb{Z}_m, \oplus_m, \odot_m)$ und dem direkten Produkt der Ringe $(\mathbb{Z}_{m_i}, \oplus_{m_i}, \odot_{m_i})$, $1 \leq i \leq k$, ist. Dies ist nicht nur für theoretische Überlegungen nützlich, sondern hat auch praktische Konsequenzen. Beispielsweise lässt sich dadurch die Laufzeit von bestimmten Berechnungen im Ring $\mathbb{Z}_m$ deutlich reduzieren, sofern die Primzahlzerlegung von $m$ bekannt ist.

## 1.4 Die Hill-Chiffre

Die von Hill im Jahr 1929 publizierte Chiffre ist eine Erweiterung der multiplikativen Chiffre auf Buchstabenblöcke. Der Klartext wird also nicht zeichen- sondern blockweise verarbeitet. Die Blöcke haben eine feste Länge $l$ und sowohl Klar- als auch Kryptotextraum bestehen aus allen Wörtern $x \in A^l$. Als Schlüssel dient eine $(l \times l)$-Matrix $k = (k_{ij})$ mit Koeffizienten in $\mathbb{Z}_m$. Diese transformiert einen Klartext $x = x_1 \dots x_l \in A^l$ in den Kryptotext $y = y_1 \dots y_l$ mit $y_i = x_1 k_{1i} + \dots + x_l k_{li}$ für $i = 1, \dots, l$:

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

Wir bezeichnen die Menge aller $(l \times l)$-Matrizen $(k_{ij})$ mit Koeffizienten $k_{ij} \in \mathbb{Z}_m$ mit $\mathbb{Z}_m^{l \times l}$. Als Schlüssel können nur invertierbare Matrizen $k$ benutzt werden, da sonst der Chiffriervorgang nicht injektiv ist. Ob eine Matrix $k \in \mathbb{Z}_m^{l \times l}$ invertierbar ist, lässt sich an ihrer Determinante erkennen.

**Definition 27** (**Determinante**). *Sei $R$ ein kommutativer Ring mit Eins und sei $A = (a_{ij}) \in R^{n \times n}$. Eine Funktion $f : R^{n \times n} \to R$ heißt **Determinantenfunktion**, falls sie folgende drei Eigenschaften erfüllt*

- *$f$ ist **multilinear**, d.h. für jede Matrix $A = (a_1, \dots, a_n) \in R^{n \times n}$ mit Spalten $a_1, \dots, a_n \in (R^n)^T$, jeden Spaltenvektor $b \in (R^n)^T$ und jedes $r \in R$ gilt*

$$f(a_1, \dots, ra_i + b, \dots, a_n) = rf(a_1, \dots, a_i, \dots, a_n) + f(a_1, \dots, b, \dots, a_n).$$

- *$f$ ist **alternierend**, d.h. im Fall $a_i = a_j$ für $i \neq j$ gilt $f(a_1, \dots, a_n) = 0$.*
- *$f$ ist **normiert**, d.h. $f(E) = 1$, wobei $E$ die Einheitsmatrix ist.*

*Tatsächlich ist $f$ durch diese drei Eigenschaften eindeutig festgelegt und wir bezeichnen $f(A)$ wie üblich mit $det(A)$.*

Eine explizite Darstellung für die Determinantenfunktion liefert der laplacesche Entwicklungssatz. Für $1 \leq i, j \leq n$ sei $A_{ij}$ die durch Streichen der $i$-ten Zeile und $j$-ten Spalte aus $A$ hervorgehende Matrix. Dann ist $det(A) = a_{11}$, falls $n = 1$, und für $n > 1$ ist

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

wobei $i \in \{1, \cdots, n\}$ beliebig wählbar ist (Entwicklung nach der $i$-ten Zeile). Das Produkt $(-1)^{i+j} det(A_{ij})$ wird **Kofaktor** genannt und mit $\tilde{a}_{ij}$ bezeichnet. Aus dieser Formel lässt sich zwar ein Algorithmus zur Berechnung der Determinante ableiten, allerdings hat dieser eine exponentielle Laufzeit. Das Gauß-Verfahren führt dagegen auf eine effiziente Berechnungsmethode für die Determinante (siehe Übungen).

Für die Dechiffrierung eines mit dem Schlüssel $k$ berechneten Kryptotextes wird die inverse Matrix $k^{-1}$ benötigt. Invertierbare Matrizen werden auch als **regulär** bezeichnet. Eine Matrix $k \in \mathbb{Z}_m^{l \times l}$ ist genau dann regulär, wenn $\mathrm{ggT}(det(k), m) = 1$ ist. In diesem Fall lässt sich $k^{-1}$ mit dem Gauß-Jordan-Algorithmus effizient berechnen (siehe Übungen).

**Definition 28.** *Sei $A = \{a_0, \dots, a_{m-1}\}$ ein beliebiges Alphabet und für eine natürliche Zahl $\ell \geq 2$ sei $M = C = A^\ell$. Bei der **Hill-Chiffre** ist $K = \{k \in \mathbb{Z}_m^{\ell \times \ell} \mid \mathrm{ggT}(det(k), m) = 1\}$ und es gilt*

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

**Beispiel 29** (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* HILL *wegen*

$$(\mathtt{HILL}) \begin{pmatrix} 11 & 13 & 8 & 21 \\ 24 & 17 & 3 & 25 \\ 18 & 12 & 23 & 17 \\ 6 & 15 & 2 & 15 \end{pmatrix} = (\mathtt{nerx}) \quad \textit{bzw.} \quad \begin{aligned} 11\,\mathtt{H} + 24\,\mathtt{I} + 18\,\mathtt{L} + 6\,\mathtt{L} &= \mathtt{n} \\ 13\,\mathtt{H} + 17\,\mathtt{I} + 12\,\mathtt{L} + 15\,\mathtt{L} &= \mathtt{e} \\ 8\,\mathtt{H} + 3\,\mathtt{I} + 23\,\mathtt{L} + 2\,\mathtt{L} &= \mathtt{r} \\ 21\,\mathtt{H} + 25\,\mathtt{I} + 17\,\mathtt{L} + 15\,\mathtt{L} &= \mathtt{x} \end{aligned}$$

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

## 1.5 Die Vigenère-Chiffre und andere Stromsysteme

Die nach dem Franzosen Blaise de Vigenère (1523–1596) benannte Chiffre ersetzt den Klartext zeichenweise, allerdings je nach Position im Klartext unterschiedlich.

**Definition 30.** *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*

$$E(k, x) = y_0 \dots y_{n-1} \quad \textit{mit} \quad y_i = x_i + k_{(i \bmod d)} \ \textit{für} \ i = 1, \dots, n-1$$

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

$$D(k, y) = x_0 \dots x_{n-1} \quad \textit{mit} \quad x_i = y_i - k_{(i \bmod d)} \ \textit{für} \ i = 1, \dots, n-1.$$

**Beispiel 31** (Vigenère-Chiffre)**.** *Verwenden wir das lateinische Alphabet $A_{lat}$ als Klartextalpabet und wählen wir als Schlüssel das Wort $k =$* WIE*, so ergibt sich für den Klartext* VIGENERE *beispielsweise der Kryptotext*

$$E(\mathtt{WIE}, \mathtt{VIGENERE}) = \underbrace{\mathtt{V}+\mathtt{W}}_{\mathtt{r}} \ \underbrace{\mathtt{I}+\mathtt{I}}_{\mathtt{q}} \ \underbrace{\mathtt{G}+\mathtt{E}}_{\mathtt{k}} \ \underbrace{\mathtt{E}+\mathtt{W}}_{\mathtt{a}} \ \underbrace{\mathtt{N}+\mathtt{I}}_{\mathtt{v}} \ \underbrace{\mathtt{E}+\mathtt{E}}_{\mathtt{i}} \ \underbrace{\mathtt{R}+\mathtt{W}}_{\mathtt{n}} \ \underbrace{\mathtt{E}+\mathtt{I}}_{\mathtt{m}} \ = \ \mathtt{rqkavinm}$$

◁

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 32.** *Vigenère-Chiffre*

| *Chiffrierung:* | | *Dechiffrierung:* | |
|---|---|---|---|
| VIGENERE | (*Klartext $x$*) | rqkavinm | (*Kryptotext $y$*) |
| $+$ WIEWIEWI | (*Schlüsselstrom $\hat{k}$*) | $-$ WIEWIEWI | (*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 33** (**Beaufort-Chiffre**). *Verschlüsseln wir den Klartext* `BEAUFORT` *beispielsweise unter dem Schlüsselwort* $k =$ `WIE`*, so erhalten wir den Kryptotext* `xmeqnsnb`*. Eine erneute Verschlüsselung liefert wieder den Klartext* `BEAUFORT`*:*

*Chiffrierung:*

| | | |
|---|---|---|
| | `WIEWIEWI` | (*Schlüsselstrom*) |
| − | `BEAUFORT` | (*Klartext*) |
| | `veecdqfp` | (*Kryptotext*) |

*Dechiffrierung:*

| | | |
|---|---|---|
| | `WIEWIEWI` | (*Schlüsselstrom*) |
| − | `veecdqfp` | (*Kryptotext*) |
| | `BEAUFORT` | (*Klartext*) |

◁

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***).[†]

[†]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.

**Beispiel 34** (*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:* | | *Kryptotext-Schlüsselstrom:* | |
|---|---|---|---|
| VIGENERE | (*Klartext*) | VIGENERE | (*Klartext*) |
| + WIEVIGEN | (*Schlüsselstrom*) | + WIERQKVD | (*Schlüsselstrom*) |
| rqkzvkvr | (*Kryptotext*) | rqkvdomh | (*Kryptotext*) |

◁

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

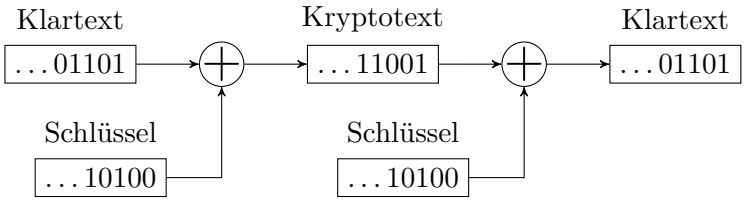
Eine weitere Möglichkeit ist, eine Textstelle in einem Buch als Schlüssel zu vereinbaren und den dort beginnenden Text als aperiodischen Schlüsselstrom zu benutzen (Lauftextverschlüsselung). Besser ist es jedoch, mithilfe von Pseudozufallsgeneratoren aus einem relativ kurzen Schlüssel einen deutlich längeren Schlüsselstrom zu erzeugen. Noch besser ist es, den Schlüsselstrom wirklich zufällig zu erzeugen. Dies führt auf eine absolut sichere Verschlüsselung, sofern der Schlüsselstrom nicht mehrmals benutzt wird.[‡] Ein solcher „Wegwerfschlüssel“ (engl. *One-Time-Pad* oder kurz *OTP*; im Deutschen auch als **individueller Schlüssel** bezeichnet) lässt sich für längere Klartexte allerdings nur mit großem Aufwand generieren und auf einem sicheren Kanal zwischen Sender und Empfänger verteilen, weshalb diese Chiffre nur wenig praktikabel ist.[§]

**Beispiel 35** (One-Time-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 jedes Klartextzeichen* $x_i$ *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. Wie diese ist der One-Time-Pad im Binärfall involutorisch.



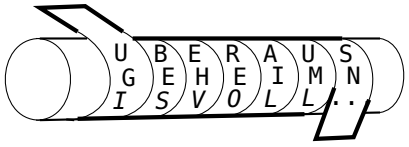

[‡] Diese Methode schlug der amerikanische Major Joseph O. Mauborgne im Jahr 1918 vor, nachdem ihm ein von Gilbert S. Vernam für den Fernschreibverkehr entwickeltes Chiffriersystem vorgestellt wurde.

[§] Diese Methode wurde beispielsweise beim „heißen Draht“, der 1963 eingerichteten, direkten Fernschreibverbindung zwischen dem Weißen Haus in Washington und dem Kreml in Moskau, angewandt.

## 1.7 Klassifikation von Kryptosystemen

Bei den bisher betrachteten Chiffrierfunktionen handelt es sich um **Substitutionen**, d.h. sie bilden 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 36** (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 ...
↝ *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

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

über. Dasselbe Resultat erhält man, wenn $x$ zeilenweise in eine $k \times m$-Matrix geschrieben und spaltenweise wieder ausgelesen wird (sogenannte **Spaltentransposition**):

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

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.

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

**Beispiel 37.** *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
```

ZICKZACKLINIE $\rightsquigarrow$ *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 38.** *Sei $A = B$ ein beliebiges Alphabet und für eine natürliche Zahl $l \geq 2$ sei $M = C = A^\ell$. Bei einer **Blocktranspositionschiffre** wird durch jeden Schlüssel $k \in K$ eine Permutation $\pi$ beschrieben, so dass für alle Zeichenfolgen $x_1 \cdots x_\ell \in M$ und $y_1 \cdots y_\ell \in C$*

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

*und*

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

*gilt.*

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

**Beispiel 39.** *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 $\pi$ **inverse Permutation** $\pi^{-1}$ benutzt werden. Wird $\pi$ durch eine Folge von Zyklen $(i_1\ i_2\ i_3\ \ldots\ i_n)$ dargestellt, wobei $i_1$ auf $i_2$, $i_2$ auf $i_3$ usw. und schließlich $i_n$ auf $i_1$ abgebildet wird, so ist $\pi^{-1}$ sehr leicht zu bestimmen.

**Beispiel 40.**

| $i$ | 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 |

*Obiges $\pi$ hat beispielsweise die Zyklendarstellung*

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

*wenn, wie allgemein üblich, Einerzyklen weggelassen werden. Daraus erhalten wir unmittelbar $\pi^{-1}$ 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.* ◁

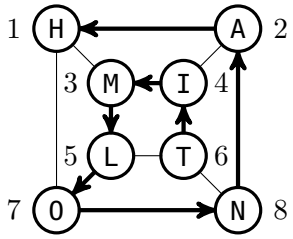
**Beispiel 41.** *Bei der **Matrix-Transposition** wird der Klartext zeilenweise in eine $k \times l$-Matrix eingelesen und der Kryptotext spaltenweise gemäß einer Spaltenpermutation $\pi \in S_l$, die als Schlüssel dient, wieder ausgelesen. Für $\pi = (1\ 4\ 3)\ (2\ 6)$ wird also zuerst Spalte $\pi(1) = 4$, 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 42.** *Bei der* ***Weg-Transposition*** *wird als Schlüssel eine Hamiltonlinie in einem Graphen mit den Knoten* $1, \ldots, l$ *benutzt. (Eine Hamiltonlinie ist eine Anordnung aller Knoten, in der je zwei aufeinanderfolgende Knoten durch eine Kante verbunden sind.) Der Klartextblock* $x_1 \cdots x_l$ *wird gemäß der Knotennumerierung in den Graphen eingelesen und der zugehörige 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, sich eine Permutationen in Form eines **Schlüsselworts** (oder einer aus mehreren Wörtern bestehenden **Schlüsselphrase**) ins Gedächtnis einzuprägen. Aus einem solchen Schlüsselwort lässt sich die zugehörige Permutation $\sigma$ leicht rekonstruieren, indem man das Wort auf Papier schreibt und in der Zeile darunter für jedes einzelne Zeichen seine Position $i$ innerhalb des Wortes vermerkt.

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

DIE BLOCKLAENGE IST SECHS $\rightsquigarrow$
*edboil lcanke igsset excsyh*

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

Wir wenden uns nun der Klassifikation von Substitutionen zu. Ein wichtiges Unterscheidungsmerkmal ist z.B. die Länge der Klartexteinheiten, auf denen die Chiffre operiert.

**Monografische Substitutionen** ersetzen Einzelbuchstaben.

**Polygrafische Substitutionen** ersetzen dagegen aus mehreren Zeichen bestehende Klartextsegmente auf einmal.

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

**Beispiel 43.** *Bei der* ***Porta-Chiffre*** *werden 400 (!) unterschiedliche von Porta für diesen Zweck entworfene Kryptotextzeichen verwendet. Diese sind in einer* $20 \times 20$*-Matrix* $M = (y_{ij})$ *angeordnet, deren Zeilen und Spalten mit den* 20 *Klartextzeichen* $\mathtt{A}, \ldots, \mathtt{I}, \mathtt{L}, \ldots, \mathtt{T}, \mathtt{V}, \mathtt{Z}$ *indiziert sind. Zur Ersetzung des Zeichenpaars* $a_i a_j$ *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 Zeichenpaaren zusammengesetzt, so spricht man von einer **bipartiten** Substitution. Ein frühes (monografisches) Beispiel einer bipartiten Chiffriermethode geht auf Polybios (circa 200 – 120 v. Chr.) zurück:

| | *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 |

POLYBIOS ⇝ *30 24 21 43 01 13 24 33*

Die **Polybios-Chiffre** benutzt als Schlüssel eine $5 \times 5$-Matrix, die aus sämtlichen Klartextzeichen gebildet wird.[¶] Die Verschlüsselung des Klartextes erfolgt zeichenweise, indem man einen in Zeile $i$ und Spalte $j$ eingetragenen Klartextzeichen durch das Koordinatenpaar $ij$ ersetzt. Der Kryptotextraum besteht also aus den Ziffernpaaren $\{\mathtt{00}, \mathtt{01}, \ldots, \mathtt{44}\}$.

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 jedes einzelne Klartextsegment unabhängig von seiner Position im Klartext auf dieselbe Weise.

**Polyalphabetische Substitutionen** verwenden eine Ersetzungsregel, die in Abhängigkeit von den bereits verarbeiteten Klartextsegmenten variieren kann.

Die Bezeichnung „monoalphabetisch“ bringt zum Ausdruck, dass der Ersetzungsmechanismus im monografischen Fall für jeden Schlüssel auf einem festen Alphabet beruht. Die von Caesar benutzte Chiffriermethode mit dem Schlüssel $k = 3$ kann beispielsweise vollständig durch Angabe des Ersetzungsalphabets $\{\mathtt{d,e,f,g,w,...,y,z,a,b,c}\}$ beschrieben werden. Monoalphabetische Chiffrierverfahren ersetzen meist Texteinheiten einer festen Länge $l \geq 1$ durch Kryptotextsegmente derselben Länge.

**Definition 44.** *Sei* $A$ *ein beliebiges Alphabet und es gelte* $M = C = A^\ell$, $\ell \geq 1$. *Eine* ***Blockchiffre*** *realisiert für jeden Schlüssel* $k \in K$ *eine bijektive Abbildung* $g$ *auf* $A^\ell$ *und es gilt für alle* $x \in M$ *und* $y \in C$,

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

*Im Fall* $\ell = 1$ *spricht man auch von einer* ***einfachen Substitutionschiffre****.*

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

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 45.** *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 $x = x_0 \dots x_{n-1}$, $x_i \in M$, eine Folge $\hat{k}_0, \dots, \hat{k}_{n-1}$ von internen Schlüsseln $\hat{k}_i = g(k, x_0 \dots x_{i-1}) \in \hat{K}$, unter denen $x$ in den Kryptotext*

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

*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 $x_0 \dots x_{i-1}$ den aktuellen internen Schlüssel $\hat{k}_i$ zu berechnen. Die bisher betrachteten Stromchiffren benutzen z.B. die folgenden Schlüsselstromgeneratoren.

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

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 46.** *Bei der* ***Spionage-Chiffre*** *wird in die erste Zeile einer* $3 \times 10$*-Matrix ein Schlüsselwort* $w$ *geschrieben, welches kein Zeichen 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 Klartextzeichen (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 Klartextzeichen 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 Schlüsselwortzeichen 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 Klartextzeichen 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 47.** *Sei* $A$ *ein Klartextalphabet und sei* $M = A$*. Weiter sei* $C$ *ein Kryptotextraum der Größe* $\|C\| > \|A\| = m$*. In einer* ***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 Klartextzeichen 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 48** (homophone Substitution)**.** *Sei* $A = \{\mathsf{A}, \ldots, \mathsf{Z}\}$, $B = \{\mathtt{0}, \ldots, \mathtt{9}\}$ *und* $C = \{\mathtt{00}, \ldots, \mathtt{99}\}$.

| | *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* $5 \times 5$*-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 Klartextzeichen 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 49** (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*} |
| ⋮ | ⋮ | ⋮ |

*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 $10 \times 10$-Matrix anzulegen, in der jeder Klartextbuchstabe $a$ an allen Stellen vorkommt, deren Koordinaten in $H(a)$ enthalten sind.

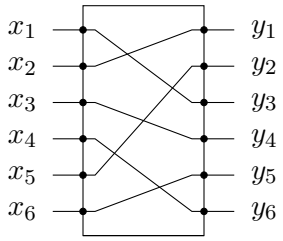
| | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *0* |
|---|---|---|---|---|---|---|---|---|---|---|
| *1* | N | E | C | S | A | O | D | X | I | N |
| *2* | R | G | S | N | N | A | U | C | H | Y |
| *3* | T | L | I | O | U | D | Z | M | N | E |
| *4* | H | R | E | A | N | E | E | S | I | T |
| *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.

## 1.8 Realisierung von Blocktranspositionen und einfachen Substitutionen

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

$$y1 := x2;\quad y2 := x5;\quad \ldots\quad y6 := x4;$$

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

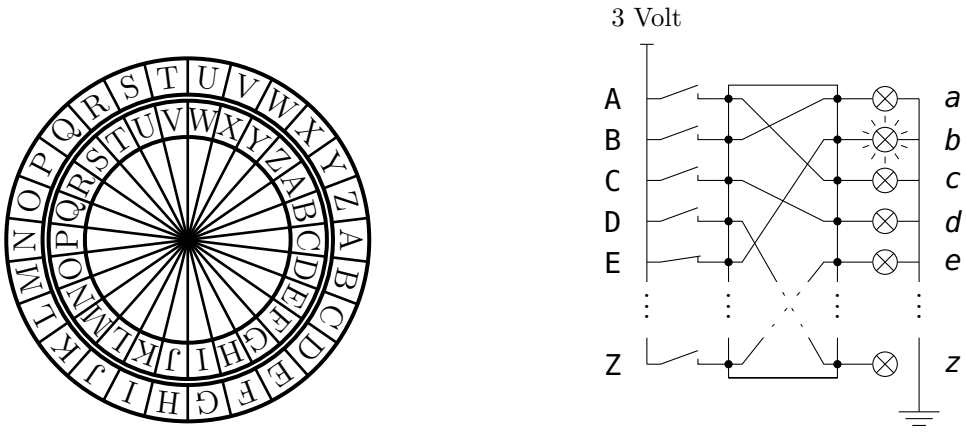



Abbildung 1.1: Realisierung von einfachen Substitutionen mit einer Drehscheibe und mit Hilfe von Steckverbindungen.

beispielsweise der Schlüssel $k = 2$ eingestellt, das heißt, $a_k = \mathsf{c}$. 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. 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 das Klartextzeichen 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 Feldelement 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 muss das Feld nur einmal initialisiert und danach nicht mehr geändert werden.

| **Schlüssel-wert** | **Klartextbuchstabe** | | | |
|---|---|---|---|---|
| | A | B | ... | Z |
| 0 | *u* | *h* | ... | *c* |
| 1 | *e* | *h* | ... | *a* |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| 63 | *y* | *f* | ... | *w* |

# 2 Kryptoanalyse der klassischen Verfahren

## 2.1 Klassifikation von Angriffen gegen Kryptosysteme

Die Erfolgsaussichten eines Angriffs gegen ein Kryptosystem hängen sehr stark von der Ausgangslage des Angreifers ab. 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 nicht 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 von den unterschiedlichen Bedrohungen, denen 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 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 (zumindest vorübergehend), sich zu Klartexten seiner Wahl die zugehörigen Kryptotexte zu besorgen. Kann hierbei die Wahl der Klartexte 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.

[*]Tatsächlich sind die Prinzipien fast aller heute im Einsatz befindlichen Kryptosysteme bekannt. Nur so kann einer Vielzahl von Kryptoanalytikern die Suche nach Schwachstellen ermöglicht werden.

**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 die meisten Anwendungen 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 können. So ermöglicht beispielsweise das Auftreten stereotyper Klartext-Formulierungen einen Angriff bei bekanntem Klartext, sofern der Gegner diese Formulierungen kennt bzw. errät. Begünstigt durch derartige Unvorsichtigkeiten, die im praktischen Einsatz meist nicht vermeidbar sind, können sich selbst winzige Konstruktionsschwächen eines Kryptosystems sehr schnell zu einer ernsthaften Bedrohung auswachsen. Die Geschichte der Kryptografie belegt sehr eindrucksvoll, dass es häufig die Anwender eines Kryptosystems selbst sind, die – im unerschütterlichen Glauben an seine kryptografische Stärke – einen erfolgreichen Angriff ermöglichen.

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 bieten, wenn es dem Gegner etwa gelingt, in den Besitz des geheimen Schlüssels zu kommen – sei es aus Unachtsamkeit der Anwender oder infolge von Manipulationsversuchen von Seiten des Gegners (**Social Engineering bzw. Social Hacking**). Auch **Implementierungsangriffe** nutzen nicht Schwachstellen des Kryptoverfahrens aus. Vielmehr zielen sie darauf ab, durch physikalische Messungen wie bspw. des Stromverbrauchs oder der Laufzeit von Berechnungen (sog. **Seitenkanalangriffe**) Informationen über den unbekannten Schlüssel zu gewinnen.

## 2.2 Kryptoanalyse von einfachen Substitutionschiffren

Durch eine Häufigkeitsanalyse können insbesondere einfache Substitutionen $g$ leicht gebrochen werden. Der Grund dafür ist, dass die einzelnen Zeichen $a$ in der Klartextsprache meist mit unterschiedlichen Wahrscheinlichkeiten $p(a)$ auftreten (vergleiche Tabelle 2.1). Berechnet man die relativen Häufigkeiten $h$ der Zeichen im Kryptotext, so gilt $p(a) \approx h(g(a))$ (vorausgesetzt der Klartext ist genügend lang). Für die Schilderung einer nach dieser Methode durchgeführten Kryptoanalyse 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** (auch **Brute-Force Angriff** genannt) ausgeführt werden kann.

**Beispiel 50** (vollständige Schlüsselsuche)**.** *Es sei bekannt, dass das Kryptotextstück* $y =$ `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.*

| $k$ | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $D(k,y)$ | SAXP | 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. Weiß er zum Beispiel, dass er in deutscher Sprache verfasst ist, kann er einen Großteil der Texte $x_i$ auszuschließen. Mit jedem Text $x_i$, der nicht als Klartext infrage 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.

Auch wenn der Gegner die Klartextsprache nicht kennt, kann eine Häufigkeitsanalyse erfolgreich sein. 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. Selbst zwischen unterschiedlichen Sprachen gibt es oft Gemeinsamkeiten. So kommt in fast allen europäischen Sprachen der Buchstabe `E` sehr häufig vor, während `X, Y` und `Z` nur selten auftreten. Diese für natürliche Sprachen typische Ungleichmäßigkeit der Buchstabenhäufigkeiten ist darauf zurückzuführen, dass sie relativ viel Redundanz enthalten.

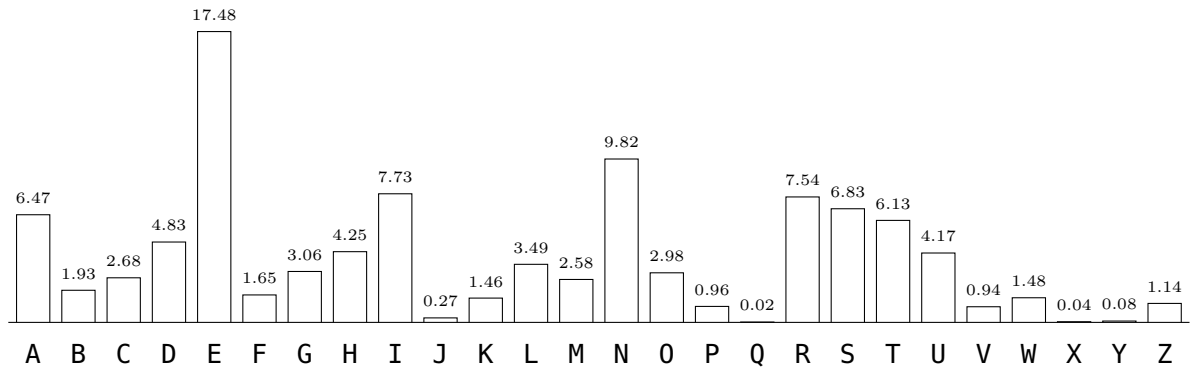



Abbildung 2.1: Häufigkeitsverteilung der Einzelbuchstaben im Deutschen (in %).

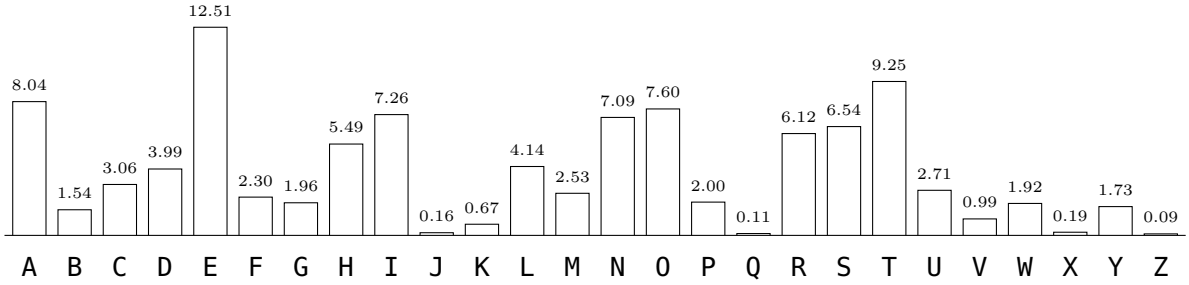



Abbildung 2.2: Häufigkeitsverteilung der Buchstaben im Englischen (in %).

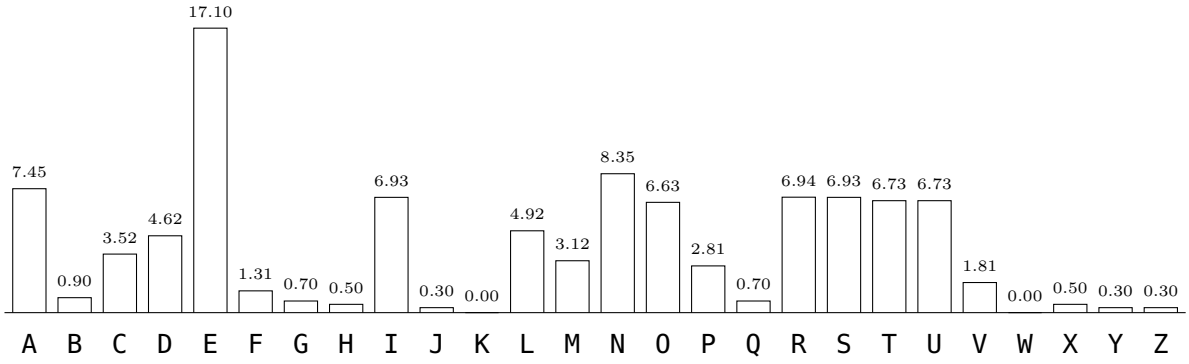



Abbildung 2.3: Häufigkeitsverteilung der Buchstaben im Französischen (in %).

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).

Bei additiven Chiffren reicht es oftmals, den häufigsten Buchstaben im Kryptotext zu bestimmen, und davon den häufigsten Buchstaben der Klartextsprache zu subtrahieren, um den Schlüssel $k$ zu erhalten. Bei affinen Chiffren müssen gewöhnlich nur die beiden häufigsten Buchstaben bestimmt werden. Diese führen auf zwei Gleichungen mit zwei Unbekannten für den gesuchten Schlüssel $k = (b, c)$.

**Beispiel 51** (Analyse einer affinen Chiffre mittels Buchstabenhäufigkeiten)**.** *Es sei bekannt, dass sich hinter dem Kryptotext*

```
laoea ehoap hwvae ixobg jcbho thlob lokhe ixope vbcix ockix qoppo boapo
mohqc euogk opeho jhkpl eappj seobe ixoap opmcu
```

*ein deutscher Klartext verbirgt, der mit einer affinen Chiffre verschlüsselt wurde. Berechnen wir für jedes Chiffrezeichen $y_i$ die (absolute) Häufigkeit $H_y(y_i)$ seines Auftretens in obigem Kryptotext $y$,*

| $y_i$ | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $H_y(y_i)$ | 7 | 6 | 5 | 0 | 10 | 0 | 2 | 8 | 5 | 3 | 4 | 4 | 2 | 0 | 19 | 11 | 2 | 0 | 1 | 1 | 2 | 2 | 1 | 5 | 0 | 0 |

*so liegt die Vermutung nahe, dass das am häufigsten vorkommende Chiffrezeichen* `o` *für das Klartextzeichen* `E` *und das am zweithäufigsten vorkommende* `p` *für* `N` *steht. Unter*

*dieser Annahme kann der gesuchte Schlüssel $k = (b, c)$ als Lösung der beiden Gleichungen*

$$\begin{aligned} b \cdot \mathsf{E} + c &= \mathit{o} \\ b \cdot \mathsf{N} + c &= \mathit{p} \end{aligned}$$

*bestimmt werden. Subtrahieren wir nämlich die erste von der zweiten Gleichung, so erhalten wir die Kongruenz $9 \cdot b \equiv_{26} 1$, woraus sich $b = 3$ und damit $c = 2$ ergibt. Tatsächlich weist der Schlüssel $k = (3, 2)$ nicht nur für die beiden Paare $(\mathsf{E}, \mathit{o})$ und $(\mathsf{N}, \mathit{p})$, sondern auch für alle übrigen Paare $(D(k, y_i), y_i)$ eine gute Übereinstimmung zwischen der Häufigkeit $H_y(y_i)$ im Kryptotext $y$ und der erwarteten Häufigkeit $H_{100}(D(k, y_i))$ auf, mit der das Zeichen $D(k, y_i)$ in einem typischen deutschen Text der Länge 100 vorkommt (die Tabelle zeigt die Werte von $H_{100}$ gerundet):*

| $y_i$ | *o* | *p* | *e* | *h* | *a* | *b* | *c* | *x* | *i* | *l* | *k* | *j* | *u* | *m* | *g* | *v* | *q* | *s* | *t* | *w* | *r* | *f* | *n* | *z* | *y* | *d* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $H_y(y_i)$ | 19 | 11 | 10 | 8 | 7 | 6 | 5 | 5 | 5 | 4 | 4 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $H_{100}(D(k, y_i))$ | 17 | 10 | 7 | 6 | 8 | 8 | 6 | 4 | 3 | 5 | 4 | 3 | 3 | 3 | 1 | 1 | 1 | 3 | 0 | 0 | 2 | 2 | 1 | 1 | 0 | 0 |
| $D(k, y_i)$ | E | N | S | T | I | R | A | H | C | D | U | L | G | M | K | P | W | O | X | Y | F | B | V | Z | Q | J |

◁

## 2.3 Kryptoanalyse von Blocktranspositionen

Mit Hilfe von Bigrammhäufigkeiten, die manchmal auch als Kontakthäufigkeiten bezeichnet werden, lassen sich Blocktranspositionen sehr leicht brechen, sofern genügend Kryptotext vorliegt. Ist die Blocklänge $\ell$ bekannt, so trägt man hierzu den Kryptotext zeilenweise in eine Matrix $S = (s_{ij}) = (S_1 \dots S_\ell)$ mit $\ell$ Spalten $S_1, \dots, S_\ell$ ein. Da jede Zeile dieser Matrix aus dem zugehörigen Klartextblock mit derselben Permutation $\pi$ erzeugt wurde, müssen die Spalten $S_j$ jetzt nur noch in die „richtige“ Reihenfolge gebracht werden, um den gesuchten Klartext zu erhalten. Die Nachfolgespalte $S_k$ von $S_j$ (bzw. die Vorgängerspalte $S_j$ von $S_k$) kann sehr gut anhand der Werte von $\hat{p}(S_j, S_k) = \sum_i p(s_{ij}, s_{ik})$ bestimmt werden.

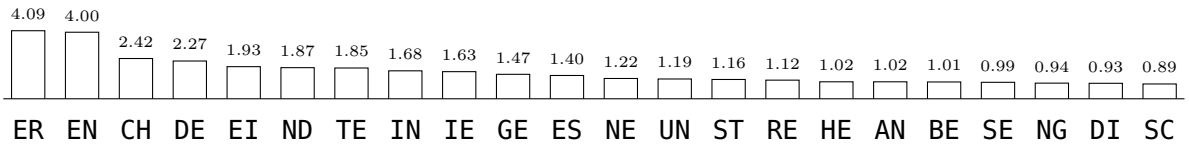



Abbildung 2.4: Die häufigsten Bigramme im Deutschen (Angaben in %).

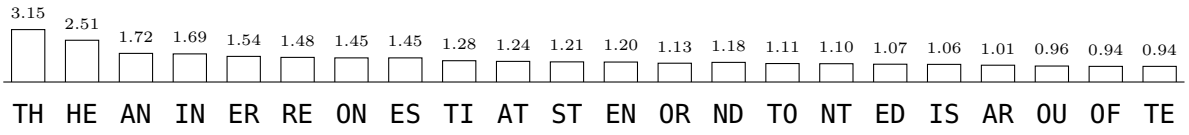



Abbildung 2.5: Die häufigsten Bigramme im Englischen (in %; nach O.P. Meaker, 1939).

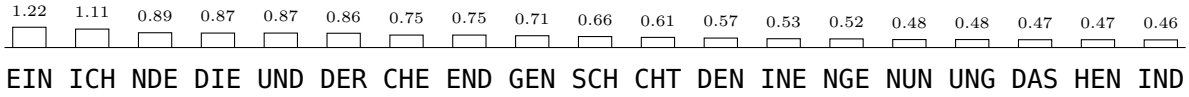



Abbildung 2.6: Die häufigsten Trigramme im Deutschen (in %).

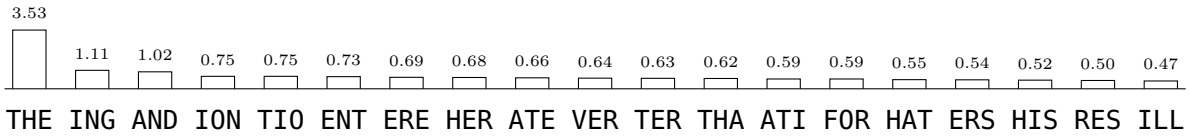



Abbildung 2.7: Die häufigsten Trigramme im Englischen (in %).

**Beispiel 52** (Häufigkeitsanalyse von Bigrammen)**.** *Für den mit einer Blocktransposition (mit vermuteter Blocklänge 5) erzeugten Kryptotext*

```
ihehr bwean rneii nrkeu elnzk rxtae vlotr engie
```

*erhalten wir eine Matrix $S$ mit den folgenden fünf Spalten.*

| $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ |
|---|---|---|---|---|
| I | H | E | H | R |
| B | W | E | A | N |
| R | N | E | I | I |
| N | R | K | E | U |
| E | L | N | Z | K |
| R | X | T | A | E |
| V | L | O | T | R |
| E | N | G | I | E |

*Um die richtige Vorgänger- oder Nachfolgerspalte von $S_1$ zu finden, bestimmen wir für jede potentielle Spalte $S_j$, $j = 2, \ldots, 5$, wieviele der Bigramme $s_{ij}s_{i1}$ (bzw. $s_{i1}s_{ij}$) zu den 20 häufigsten (aus Abbildung 2.4) gehören.*

| | | | | ↓ | | ↓ | | |
|---|---|---|---|---|---|---|---|---|
| $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ |
| H | **E** | H | R | **I** | H | **E** | H | R |
| W | E | A | N | **B** | W | **E** | A | N |
| N | **E** | I | I | **R** | N | **E** | I | I |
| R | K | **E** | **U** | **N** | R | K | **E** | U |
| L | **N** | Z | K | **E** | L | **N** | Z | K |
| X | T | A | **E** | **R** | X | T | A | **E** |
| L | O | T | R | V | L | O | T | R |
| **N** | **G** | **I** | E | **E** | **N** | G | **I** | E |
| 1 | 4 | 2 | 2 | | 1 | 4 | 2 | 1 |

*Da die beiden Spaltenpaare $(S_3, S_1)$ und $(S_1, S_3)$ jeweils vier häufige Bigramme bilden, können wir annehmen, dass im Klartext $S_1$ auf $S_3$ oder $S_3$ auf $S_1$ folgen muss. Entscheiden wir uns für die zweite Möglichkeit, so sollten wir als nächstes die Spaltenpaare $(S_j, S_1)$ und $(S_3, S_j)$, $j = 2, 4, 5$ betrachten.*

| | | | | ↓ | | | ↓ |
|---|---|---|---|---|---|---|---|
| $S_2$ | $S_4$ | $S_5$ | $S_1$ | $S_3$ | $S_2$ | $S_4$ | $S_5$ |
| H | H | R | I | **E** | H | H | **R** |
| W | A | N | B | **E** | W | A | **N** |
| N | I | I | R | **E** | **N** | **I** | **I** |
| R | **E** | **U** | **N** | K | R | E | U |
| L | Z | K | E | N | L | Z | K |
| X | A | **E** | **R** | **T** | X | A | **E** |
| L | T | R | V | O | L | T | R |
| **N** | **I** | E | **E** | **G** | N | I | **E** |
| 1 | 2 | 2 | | | 1 | 1 | 5 |

*Aufgrund des hohen Wertes von $\hat{p}(S_3, S_5)$ können wir annehmen, dass auf $S_3$ die Spalte $S_5$ folgt. Im nächsten Schritt erhalten wir daher die folgende Tabelle.*

| | ↓ | ↓ | | ↓ | ↓ | |
|---|---|---|---|---|---|---|
| $S_2$ | $S_4$ | $S_1$ | $S_3$ | $S_5$ | $S_2$ | $S_4$ |
| H | H | I | E | R | H | H |
| W | A | B | E | N | W | A |
| N | I | R | E | **I** | **N** | I |
| R | **E** | **N** | K | U | R | E |
| L | Z | E | N | K | L | Z |
| X | A | R | T | E | X | A |
| L | T | V | O | R | L | T |
| **N** | **I** | **E** | G | **E** | **N** | **I** |
| 1 | 2 | | | | 2 | 1 |

*Diese lässt die Spaltenanordnung $S_4, S_1, S_3, S_5, S_2$ vermuten, welche tatsächlich auf den gesuchten Klartext führt:*

| $S_4$ | $S_1$ | $S_3$ | $S_5$ | $S_2$ |
|---|---|---|---|---|
| H | I | E | R | H |
| A | B | E | N | W |
| I | R | E | I | N |
| E | N | K | U | R |
| Z | E | N | K | L |
| A | R | T | E | X |
| T | V | O | R | L |
| I | E | G | E | N |

◁

## 2.4 Kryptoanalyse von polygrafischen Chiffren

Blocksysteme mit kleiner Blocklänge $\ell$ (beispielsweise bigrafische Systeme) lassen sich ähnlich wie einfache Substitutionen durch Häufigkeitsanalysen brechen. Wird bei Hill-Chiffren $l$ sehr groß gewählt, so ist eine solche statistische Analyse nicht mehr möglich. Das Hill-System kann dann zwar einem Kryptotextangriff widerstehen, jedoch kaum einem Angriff mit bekanntem Klartext und schon gar nicht einem Angriff mit gewähltem Klartext.

*Angriff mit gewähltem Klartext* O. B. d. A. sei $A = \{0, 1, \ldots, m-1\}$. Bei einem GK-Angriff verschafft sich der Gegner den Kryptotext zu $100\ldots0,\ 010\ldots0,\ \ldots,\ 0\ldots001 \in A^l$:

$$\begin{aligned} g(100\ldots0) &= k_{1\,1}\, k_{1\,2} \ldots k_{1\,l} \\ g(010\ldots0) &= k_{2\,1}\, k_{2\,2} \ldots k_{2\,l} \\ &\vdots \\ g(0\ldots001) &= k_{l\,1}\, k_{l\,2} \ldots k_{l\,l} \end{aligned}$$

und erhält damit die Schlüsselmatrix $k$.

*BK-Angriff* (bekannter Klartext). Ist bei einem BK-Angriff eine ausreichende Menge von Klartext-Kryptotextpaaren bekannt, so kann das Hill-System folgendermaßen gebrochen werden: Sind $x_i$, $y_i$ $(i = 1, \ldots, \mu)$ Paare mit $x_i k = y_i$ und gilt $\mathrm{ggT}(det(X), m) = 1$ für eine aus $l$ Blöcken $x_i$, $i \in I$, als Zeilen gebildete Matrix $X$, so lässt sich die Schlüsselmatrix $k$ zu $k = YX^{-1}$ bestimmen ($Y$ ist die aus den Blöcken $y_i$, $i \in I$, gebildete Matrix).

## 2.5 Kryptoanalyse von polyalphabetischen Chiffren

Die Vigenère-Chiffre galt bis ins 19. Jahrhundert als sicher. Da der Schlüsselstrom bei der Vigenère-Chiffre periodisch ist, lässt sie sich mit statistischen Methoden ebenfalls leicht brechen, insbesondere wenn der Kryptotext im Verhältnis zur Periode $d$ (Länge des Schlüsselwortes) genügend lang ist.

### Bestimmung der Schlüsselwortlänge

Ist die Periode $d$ bekannt, gibt es mehrere Methoden, eine Vigenère-Chiffre zu brechen. So kann man beispielsweise den Kryptotext zeilenweise in eine $d$-spaltige Matrix schreiben. Verfahrensbedingt wurden dann die einzelnen Spalten $y_1, \ldots, y_d$ durch eine monoalphabetische Substitution (genauer: durch eine additive Chiffre) verschlüsselt. Sie können daher einzeln durch eine Häufigkeitsanalyse gebrochen werden. Hierbei liefert jede Spalte $y_i$ den Buchstaben $k_i$ des Schlüsselwortes.

Zur Bestimmung der Schlüsselwortlänge betrachten wir zwei Vorgehensweisen: den Kasiski-Test und die Koinzidenzindex-Untersuchung.

*Der Kasiski-Test.* Die früheste generelle Methode zur Bestimmung der Periode bei der Vigenère-Chiffre stammt von Friedrich W. Kasiski (1860). Kommt ein Wort an zwei verschiedenen Stellen im Kryptotext vor, so kann es sein, dass die gleiche Klartextsequenz zweimal auf die gleiche Weise, d. h. mit der gleichen Schlüsselsequenz, verschlüsselt wurde. In diesem Fall ist die Entfernung $\delta$ der beiden Vorkommen ein Vielfaches der Periode $d$. Werden mehrere Paare mit verschiedenen Entfernungen $\delta_i$ gefunden, so liegt die Vermutung nahe, dass $d$ gemeinsamer Teiler aller (oder zumindest vieler) $\delta_i$ ist, was die Anzahl der noch in Frage kommenden Werte für $d$ stark einschränkt.

**Beispiel 53** (Kasiski-Test)**.**

```
  DERERSTEUNDLETZTEVERS...   (Klartext x)
+ KASKASKASKASKASKASKAS...   (Schlüsselstrom k̂)
  nejorkdemxddotrdenork...   (Kryptotext y)
```

*Da die Textstücke* `ork`*, bzw.* `de` *im Kryptotext in den Entfernungen* $\delta_1 = 15$ *und* $\delta_2 = 9$ *vorkommen, liegt die Vermutung nahe, dass die Periode* $d = \mathrm{ggT}(9, 15) = 3$ *ist.* ◁

*Koinzidenzindex-Untersuchungen.* Zur Bestimmung der Periode $d$ gibt es neben heuristischen Methoden auch folgenden statistischen Ansatz, der erstmals von William Frederick Friedman im Jahr 1920 beschrieben wurde. Er basiert auf der Beobachtung, dass eine längere Periode eine zunehmende *Glättung* der Buchstabenhäufigkeiten im Kryptotext bewirkt.

**Definition 54.** *Der* ***Koinzidenzindex*** *(engl.* ***i****ndex of* ***c****oincidence) eines Textes* $y$ *der Länge* $n$ *über dem Alphabet* $\mathcal{B}$ *ist definiert als*

$$IC(y) = \frac{1}{n \cdot (n-1)} \cdot \sum_{a \in \mathcal{B}} H_y(a) \cdot (H_y(a) - 1).$$

*Hierbei ist* $H_y(a)$ *die absolute Häufigkeit des Buchstabens* $a$ *im Text* $y$*.*

$IC(y)$ gibt also die Wahrscheinlichkeit an, mit der man im Text $y$ an zwei zufällig gewählten Positionen den gleichen Buchstaben vorfindet. Er ist umso größer, je ungleichmäßiger die Häufigkeiten $H_y(a)$ sind (siehe unten).

Um die Periode $d$ einer Vigenère-Chiffre zu bestimmen, schreibt man den Kryptotext $y$ für $d = 1, 2, 3, \ldots$ in eine Matrix mit $d$ Spalten und berechnet für jede Spalte $y_i$ den Koinzidenzindex $IC(y_i)$. Für genügend lange Kryptotexte ist dasjenige $d$, welches das maximale arithmetische Mittel der Spaltenindizes $IC(y_i)$ liefert mit hoher Wahrscheinlichkeit die gesuchte Periode. Enthält eine Spalte nämlich nur Kryptozeichen, die alle mit demselben Schlüsselbuchstaben $k$ erzeugt wurden, so stimmt der Koinzidenzindex dieser Spalte mit dem Koinzidenzindex des zugehörigen Klartextes überein, nimmt also einen relativ großen Wert an. Wurden dagegen die Kryptozeichen einer Spalte mit unterschiedlichen Schlüsselbuchstaben generiert, so wird hierdurch eine Glättung der Häufigkeitsverteilung bewirkt, weshalb der Spaltenindex kleiner ausfällt.

Ist die Einzelbuchstabenverteilung $p : A \to [0, 1]$ der Klartextsprache bekannt, so kann der Suchraum für den Wert der Periode $d$ erheblich eingeschränkt werden. Hierzu berechnet man den erwarteten Koinzidenzindex

$$\mathrm{E}_{d,n}(IC) = E(IC(Y)),$$

wobei $Y$ ein mittels einer Vigenère-Chiffre mit einem zufälligen Schlüsselwort der Länge $d$ aus einem zufälligen Klartext der Länge $n$ generierter Kryptotext ist. Im Fall $d = 1$ gilt $IC(y) = IC(x)$. Zudem können wir bei längeren Texten von den gegenseitigen Abhängigkeiten der Zeichen im Text absehen und erhalten

$$\mathrm{E}_{1,\infty}(IC) = \sum_{a \in A} p(a)^2.$$

Dieser Wert wird auch als Koinzidenzindex der zugrunde liegenden Sprache bezeichnet.

**Definition 55.** *Der* ***Koinzidenzindex*** $IC_L$ ***einer Sprache*** *mit Buchstabenverteilung* $p : A \to [0, 1]$ *ist definiert als*

$$IC_L = \sum_{a \in A} p(a)^2.$$

$IC_L$ ist zudem ein Maß für die Rauheit der Verteilung $p$.

**Definition 56** (**Rauheitsgrad; Measure of Roughness**). *Der* ***Rauheitsgrad*** $MR_L$ *einer Sprache* $L$ *mit Einzelbuchstabenverteilung* $p$ *ist*

$$MR_L = \sum_{a \in A} (p(a) - 1/m)^2 = \sum_{a \in A} p(a)^2 - 1/m = IC_L - 1/m,$$

*wobei* $m = \|A\|$ *ist.*

**Beispiel 57.** *Für die englische Sprache* ($m = 26$) *gilt beispielsweise* $IC_{\text{Englisch}} \approx 0.0687$ *und* $MR_{\text{Englisch}} \approx 0.0302$. ◁

Übersteigt dagegen die Periode $d$ die Klartextlänge $n$, so ist der Kryptotext bei zufälliger Wahl des Schlüsselswortes ebenfalls rein zufällig, was auf einen erwarteten Koinzidenzindex von

$$\mathrm{E}_{d,n}(IC) = \sum_{a \in A} \|A\|^{-2} = \|A\|^{-1} = 1/m, \quad d \geq n \geq 2$$

führt. Allgemein gilt für hinreichend großes $n$,

$$\mathrm{E}_{d,n}(IC) = \frac{n-d}{d \cdot (n-1)} \cdot IC_L + \frac{n \cdot (d-1)}{d \cdot (n-1)} \cdot m^{-1}, \quad 1 \leq d \leq n,$$

da von den $\binom{n}{2}$ möglichen Positionspaaren ungefähr $d \cdot \binom{n/d}{2} = n(n-d)/2d$ Paare nur eine Spalte (was einem Anteil von $(n-d)/d(n-1)$ entspricht) und $\binom{d}{2}(n/d)^2 = n^2(d-1)/2d$ Paare zwei unterschiedliche Spalten betreffen (was einem Anteil von $n(d-1)/d(n-1)$ entspricht).
Untenstehende Tabelle gibt den Erwartungswert $\mathrm{E}_{d,n}(IC)$ des Koinzidenzindexes für Kryptotexte der Länge $n = 100$ in Abhängigkeit von der Periodenlänge $d$ einer Vigenère-Chiffre wieder (in Promille; Klartext ist ein zufällig gewählter Text der englischen Sprache mit 100 Buchstaben).

| $d$ | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 10 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| $\mathrm{E}_{d,100}(IC)$ | 69 | 54 | 48 | 46 | 44 | 43 | 42 | 41 | 39 |

**Beispiel 58.** *Berechnet sich der Koinzidenzindex eines Vigenère-Kryptotextes der Länge* 100 *zu* 0.045*, so liegt die Vermutung nahe, dass das verwendete Schlüsselwort die Länge vier oder fünf hat, falls* $y$ *aus einem Klartext der englischen Sprache erzeugt wurde.* ◁

Der Koinzidenzindex kann auch Hinweise dafür liefern, mit welchem Kryptoverfahren ein vorliegender Kryptotext erzeugt wurde. Bei Transpositionschiffren sowie bei einfachen Substitutionen bleibt nämlich der Koinzidenzindex im Gegensatz zu polyalphabetischen und polygrafischen Verfahren erhalten. Erstere lassen sich von letzteren zudem dadurch unterscheiden, dass bei ihnen sogar die Buchstabenhäufigkeiten unverändert bleiben.

Zur Bestimmung des Schlüsselwortes bei bekannter Periode $d$ kann auch wie folgt vorgegangen werden. Man schreibt den Kryptotext $y$ in Spalten $y_i$ auf und berechnet für $a \in A$ und $i = 1, \ldots, d$ die relativen Häufigkeiten $h_i(a)$ von $a$ in $y_i$. Da $y_i$ aus dem Klartext durch Addition von $k_i$ entstanden ist, kommt die Verteilung

$$h_i(a + k), a \in A$$

für $k = k_i$ der Klartextverteilung $p(a)$, $a \in A$, näher als für $k \neq k_i$. Da

$$\alpha_i(k) := \sum_{a \in A} p(a) h_i(a + k)$$

ein Maß für die Ähnlichkeit der beiden Verteilungen $p(a)$ und $h_i(a+k)$ ist (siehe Übungen), wird der Wert von $\alpha_i(k)$ wahrscheinlich für $k = k_i$ maximal werden.

**Beispiel 59.** *Der folgende Kryptotext y*

```
huds kuae zgxr avtf pgws wgws zhtp pbil lrtz pzhw loij vfic
vbth lugi lgpr khwm yhti uaxr bhtw ucgx ospw aoch imcs yhwq
hwcf yocg ogtz lbil swbf lohx zwsi zvds atgs thwi ssux lmts
mhwi kspx ogwi hrpf lsam usuv vail lhgi lhwv vivl avtw ocij
ptic mstx vii
```

*der Länge* 203 *wurde von einer Vigenère-Chiffre mit Schlüssellänge* $d = 4$ *aus englischem Klartext erzeugt. Schreiben wir den Kryptotext in vier Spalten* $y_1, \ldots, y_4$ *der Länge* $|y_1| = |y_2| = |y_3| = 51$ *und* $|y_4| = 50$*, so ergeben sich folgende Werte für* $\alpha_i(k)$ *(in Promille):*

| $k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha_1(k)$ | 36 | 31 | 31 | 45 | 38 | 26 | 42 | **73** | 44 | 26 | 36 | 47 | 30 | 32 | 36 | 29 | 28 | 39 | 48 | 42 | 42 | 39 | 42 | 42 | 35 | 31 |
| $\alpha_2(k)$ | 44 | 41 | 40 | 51 | 41 | 31 | 37 | 43 | 34 | 28 | 36 | 26 | 28 | 43 | **68** | 45 | 35 | 27 | 42 | 43 | 40 | 35 | 30 | 24 | 31 | 45 |
| $\alpha_3(k)$ | 47 | 41 | 48 | 37 | 49 | 40 | 35 | 30 | 48 | 32 | 25 | 42 | 31 | 26 | 43 | **76** | 37 | 31 | 39 | 45 | 35 | 34 | 37 | 26 | 30 | 25 |
| $\alpha_4(k)$ | 38 | 40 | 27 | 41 | **65** | 47 | 28 | 34 | 39 | 33 | 35 | 36 | 30 | 30 | 48 | 44 | 35 | 42 | 47 | 38 | 39 | 34 | 27 | 38 | 36 | 37 |

*Da* $\alpha_1(k)$ *für* $k = 7 =$ `H`*,* $\alpha_2(k)$ *für* $k = 14 =$ `O`*,* $\alpha_3(k)$ *für* $k = 15 =$ `P` *und* $\alpha_4(k)$ *für* $k = 4 =$ `E` *einen Maximalwert annimmt, lautet das Schlüsselwort* `HOPE`*. Damit ergibt sich folgender Klartext (aus der Erzählung „Der Goldkäfer“ von Edgar Allan Poe).*

```
A GOOD GLASS IN THE BISHOPS HOSTEL IN THE DEVILS SEAT
FORTYONE DEGREES AND THIRTEEN MINUTES NORTH EAST AND
BY NORTH MAIN BRANCH SEVENTH LIMB EAST SIDE SHOOT FROM
THE LEFT EYE OF THE DEATHS HEAD A BEE LINE FROM THE TREE
THROUGH THE SHOT FIFTY FEET OUT
```

◁

Zur Bestimmung des Schlüsselwortes kann man auch die Methode des *gegenseitigen Koinzidenzindexes* verwenden. Dabei ist die verwendete Klartextsprache (und somit deren Häufigkeitsverteilung) irrelevant, da die Spalten – wie der Name schon sagt – gegenseitig in Relation gesetzt werden. Aber zuerst die Definition.

**Definition 60.** *Der* ***gegenseitge Koinzidenzindex*** *von zwei Texten y und* $y'$ *mit den Längen n und* $n'$ *über dem Alphabet* $\mathcal{B}$ *ist definiert als*

$$IC(y, y') = \frac{1}{n \cdot n'} \cdot \sum_{a \in \mathcal{B}} H_y(a) \cdot H_{y'}(a).$$

$IC(y, y')$ ist also die Wahrscheinlichkeit, dass bei zufälliger Wahl einer Position in $y$ und einer Position in $y'$ der gleiche Buchstabe vorgefunden wird. $IC(y, y')$ ist umso größer, je besser die Häufigkeitsverteilungen von $y$ und $y'$ (d. h. $H_y$ und $H_{y'}$) übereinstimmen.

Ist nun $y$ ein Kryptotext, der mit einem Schlüsselwort bekannter Länge $d$ erzeugt wurde, und sind $y_i$ $(i = 1, \ldots, d)$ die zugehörigen Spalten, so gibt der gegenseitige Koinzidenzindex der Spalten $y_i + \delta$ und $y_j$ (für $1 \leq i < j \leq d$ und $0 \leq \delta \leq 25$) die Wahrscheinlichkeit an, dass man bei zufälliger Wahl einer Position in $y_i + \delta$ und in $y_j$ denselben Buchstaben vorfindet. Da die Einzelzeichenverteilungen von $y_i - k_i$ und von $y_j - k_j$ der der Klartextsprache entsprechen, haben $y_i + \delta$ und $y_j$ für $\delta = k_j - k_i$ eine ähnliche Verteilung. Mit großer Wahrscheinlichkeit nimmt also $IC(y_i + \delta, y_j)$ für $\delta = \delta_{ij} = k_j - k_i$ einen relativ großen Wert an, während für $\delta \neq \delta_{ij}$ mit kleinen Werten zu rechnen ist.

**Beispiel 61.** *Betrachten wir den Kryptotext aus vorigem Beispiel, so ergeben sich für $IC(y_i + \delta, y_j)$ die folgenden Werte (in Promille):*

| $\delta$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $IC(y_1 + \delta, y_2)$ | 40 | 31 | 25 | 38 | 25 | 21 | 46 | **74** | 50 | 33 | 31 | 44 | 43 | 34 | 31 | 28 | 24 | 31 | 44 | 45 | 37 | 48 | 64 | 44 | 25 | 31 |
| $IC(y_1 + \delta, y_3)$ | 26 | 47 | 25 | 21 | 47 | 32 | 18 | 49 | **91** | 42 | 27 | 51 | 45 | 31 | 29 | 32 | 23 | 29 | 27 | 39 | 45 | 46 | 39 | 58 | 44 | 24 |
| $IC(y_1 + \delta, y_4)$ | 38 | 40 | 29 | 31 | 35 | 24 | 32 | 58 | 42 | 32 | 44 | 50 | 43 | 39 | 31 | 20 | 34 | 36 | 30 | 40 | 45 | 24 | 42 | **78** | 47 | 22 |
| $IC(y_2 + \delta, y_3)$ | 50 | **85** | 49 | 21 | 28 | 35 | 24 | 34 | 46 | 25 | 24 | 27 | 59 | 50 | 50 | 53 | 51 | 24 | 22 | 26 | 43 | 36 | 35 | 32 | 24 | 34 |
| $IC(y_2 + \delta, y_4)$ | 46 | 53 | 40 | 37 | 51 | 42 | 29 | 23 | 24 | 32 | 40 | 55 | 38 | 31 | 32 | 45 | **67** | 49 | 25 | 27 | 29 | 29 | 34 | 37 | 38 | 35 |
| $IC(y_3 + \delta, y_4)$ | 49 | 36 | 38 | 60 | 36 | 25 | 34 | 19 | 29 | 42 | 41 | 33 | 54 | 27 | 36 | **78** | 47 | 25 | 29 | 33 | 27 | 28 | 47 | 32 | 27 | 54 |

*Also ist (mit großer Wahrscheinlichkeit)*

$$\delta_{12} = 7,\ \delta_{13} = 8,\ \delta_{14} = 23,\ \delta_{23} = 1,\ \delta_{24} = 16,\ \delta_{34} = 15.$$

*Wir können nun alle Spalten relativ zur ersten Spalte so verschieben, dass der ganze Text eine einheitliche Verschiebung $\delta$ hat, also die zweite Spalte um $-7$, die dritte um $-8$ und die vierte um $-23$. Für die Bestimmung von $\delta$, muss man nur den häufigsten Buchstaben in dem auf diese Weise erzeugten Text bestimmen (oder eine vollständige Suche durchführen). Dieser ist* L *$(16,3\%)$. Also ist $\delta =$* L$-$E $=$ *H $= 7$ und das Schlüsselwort lautet* HOPE *(*H $+ 7 =$ O*,* H $+ 8 =$ P*,* H $+ 23 =$ E*).* ◁

### Analyse der Lauftextverschlüsselung

Zum Brechen einer Stromchiffre mit Klartextschlüsselstrom kann man wie folgt vorgehen. Man geht zunächst davon aus, dass jedes Kryptotextzeichen durch Summation eines Klartext- und Schlüsselstromzeichens mit jeweils mittlerer bis hoher Wahrscheinlichkeit entstanden ist. Dies sind etwa im Englischen die Zeichen E, T, A, O, I, N, S, R, H. Zu einem Teilwort $w$ des Kryptotextes bestimmt man dann alle Paare von Wörtern $(w_1, w_2)$ mit $w_1 + w_2 = w$ und $w_1, w_2 \in \{\mathsf{E}, \mathsf{T}, \mathsf{A}, \mathsf{O}, \mathsf{I}, \mathsf{N}, \mathsf{S}, \mathsf{R}, \mathsf{H}\}^*$. In der Regel ergeben sich nur sehr wenige sinnvolle Paare, aus denen durch Kontextbetrachtungen und Erweitern von $w$ nach links und rechts der Kryptotext entschlüsselt werden kann. Wird die Analyse durch ein Computerprogramm durchgeführt, kann an die Stelle der Kontextbetrachtungen auch die Häufigkeitsverteilung von $n$-Grammen der Sprache treten. Das Programm wählt dann solche Wortpaare $(w_1, w_2)$, die eine hohe Wahrscheinlichkeit haben.

**Beispiel 62.** *Gegeben ist der Kryptotext* moqkthcblmwxf*... Wir beginnen die Untersuchung mit einer Wortlänge von vier Buchstaben, also $w =$* moqk*. Der erste Buchstabe* m *kann nur auf eine der folgenden Arten zustande gekommen sein:*

| | | |
|---|---|---|
| | *abcde...i...t...z* | *(Klartextzeichen)* |
| + | *MLKJI...E...T...N* | *(Schlüsselzeichen)* |
| = | MMMMM...M...M...M | *(Kryptotextzeichen)* |

*Es ergeben sich folgende wahrscheinliche Paare für die Zeichen von $w$:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *m:* | (E,*I*) | *o:* | (A,*O*) | *q:* | (I,*I*) | *k:* | (R,*T*) |
| | (I,*E*) | | (H,*H*) | | | | (S,*S*) |
| | (T,*T*) | | (O,*A*) | | | | (T,*R*) |

*Diese führen auf folgende $3 \cdot 3 \cdot 1 \cdot 3 = 27$ Wortpaare $(w_1, w_2)$:*

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $w_1$ | EAIR | EAIS | EAIT | EHIR | ... | THIS | ... | TOIT |
| $w_2$ | *IOIT* | *IOIS* | *IOIR* | *IHIT* | ... | *THIS* | ... | *TAIR* |

*Als sinnvoll stellt sich aber nur die Wahl $w_1 = w_2 =$ THIS heraus.* ◁

### *Autokey* Chiffren

*Kryptotextschlüsselstrom.* Diese Systeme bieten so gut wie keinen Schutz, da sie ohne Kenntnis des Schlüsselwortes sehr leicht entschlüsselt werden können (falls die Länge des Schlüsselwortes im Verhältnis zur Länge des Kryptotextes relativ kurz ist). Man subtrahiert dazu den Kryptotext $y$ für $\delta = 1, 2, \ldots$ von dem um $\delta$ Positionen verschobenen Kryptotext – also $y_{0+\delta}\, y_{1+\delta}\, y_{2+\delta}\, y_{3+\delta}\, \ldots$ minus $y_0\, y_1\, y_2\, y_3 \ldots$ –, bis sinnvoller (Klar-) Text erscheint:

| | | |
|---|---|---|
| | *dumsqmozkfn...* | (Kryptotext $y$) |
| − | *DUMSQMO...* | („Kryptotextschlüsselstrom“) |
| = | ....NSCHUTZ... | (Klartext $x$) |

*Klartextschlüsselstrom.* Neben der oben beschriebenen Analyse der Lauftextverschlüsselung kann das Brechen der *Autokey*-Systeme mit Klartextschlüsselstrom auch analog zur Kasiski-Methode erfolgen: Sei $d$ die Länge des Schlüsselwortes $k_0 \ldots k_{d-1}$. Falls im Klartext die gleiche Buchstabenfolge $x_i \ldots x_{i+l-1}$ im Abstand $2d$ auftritt (beispielsweise $d = 3$ und $l = 2$),

| | | |
|---|---|---|
| | $\downarrow\ \downarrow \quad \downarrow\ \downarrow$ | |
| | $x_0\, x_1\, x_2\, x_3\, \underset{\sim}{x_4\, x_5}\, x_6\, x_7\, x_8\ \ x_9\, \underset{\sim}{x_{10}\, x_{11}}\, x_{12}\, x_{13}\, x_{14}\ \ldots$ | Klartext $x$ |
| + | $k_0\, k_1\, k_2\, x_0\, x_1\, x_2\ \ x_3\, \underset{\sim}{x_4\, x_5}\, x_6\, x_7\, x_8\ \ x_9\ \ \underset{\sim}{x_{10}\, x_{11}}\ldots$ | Klartextschlüsselstrom $kx$ |
| = | $y_0\, y_1\, y_2\, y_3\, y_4\, y_5\ \ y_6\ \underline{y_7\, y_8}\ \ y_9\ \underline{y_{10}\, y_{11}}\ \ y_{12}\, y_{13}\, y_{14}\ \ldots$ | Kryptotext $y$ |

so tritt im Kryptotext die gleiche Buchstabenfolge im Abstand $d$ auf, d. h. $d$ kann auf diese Art unter Umständen leicht bestimmt werden. Ist $d$ bekannt, so können die Buchstaben $k_1 \ldots k_d$ des Schlüsselwortes der Reihe nach bestimmt werden: Da durch $k_i$ die Klartextzeichen an den Positionen $i,\ d+i,\ 2d+i,\ \ldots$ eindeutig festgelegt sind, kann jedes einzelne $k_i$ unabhängig von den anderen Schlüsselwortbuchstaben durch eine statistische Analyse bestimmt werden.

# 3 Sicherheit von Kryptosystemen

## 3.1 Informationstheoretische Sicherheit

Claude E. Shannon untersuchte die Sicherheit kryptografischer Systeme auf informationstheoretischer Basis (1945, freigegeben 1949). Seinen Untersuchungen liegt das Modell einer Nachrichtenquelle $X$ zugrunde, die einzelne Klartextnachrichten $x$ aus dem Klartextraum $M$ unter einer bestimmten Wahrscheinlichkeitsverteilung $p(x) = \Pr[X = x]$ generiert.

Zudem nehmen wir an, dass der zur Verschlüsselung benutzte Schlüssel $k \in K$ von einem Schlüsselgenerator $S$ unter einer bekannten Wahrscheinlichkeitsverteilung $p(k) = \Pr[S = k]$ erzeugt wird. Da der Schlüssel unabhängig vom Klartext gewählt wird, ist $p(k, x) = p(k)p(x)$ die Wahrscheinlichkeit dafür, dass $X$ den Klartext $x$ generiert und dieser mit dem Schlüssel $k$ verschlüsselt wird. Dabei gehen wir davon aus, dass für jede Nachricht $x \in M$ ein neuer Schlüssel gewählt wird. Dies bedeutet, dass wir beispielsweise bei der additiven Chiffre den Klartextraum auf $M = A^n$ vergrößern müssen, falls der Schlüssel nach $n$ Zeichen gewechselt wird.

Die Zufallsvariablen $X$ und $S$ induzieren eine Verteilung auf dem Kryptotextraum, die wir durch die Zufallsvariable $Y$ beschreiben. Die Wahrscheinlichkeit eines Kryptotextes $y$ berechnet sich zu

$$p(y) = \Pr[Y = y] = \sum_{k,x:E(k,x)=y} p(k, x)$$

und für einen beobachteten Kryptotext $y$ (mit $p(y) > 0$) ist

$$p(x|y) = \frac{p(x, y)}{p(y)} = \sum_{k:E(k,x)=y} \frac{p(k, x)}{p(y)}$$

die (bedingte) Wahrscheinlichkeit dafür, dass sich hinter dem Kryptotext $y$ der Klartext $x$ verbirgt. Da der Schlüsselgenerator für die Sicherheit eines Kryptosystems eine wichtige Rolle spielt, nehmen wir bei Sicherheitsbetrachtungen die Schlüsselverteilung $S$ als sechste Komponente eines Kryptosystems hinzu.

**Definition 63.** *Ein Kryptosystem $KS = (M, C, E, D, K, S)$ mit Schlüsselverteilung $S$ heißt **informationstheoretisch** (oder **absolut**) **sicher**, falls jede Klartextverteilung $X$ auf $M$ unabhängig von der zugehörigen Kryptotextverteilung $Y$ auf $C$ ist.*

Bei einem absolut sicheren Kryptosystem ist demnach die *A-posteriori-Wahrscheinlichkeit* $p(x|y)$ einer Klartextnachricht $x$ gleich der *A-priori-Wahrscheinlichkeit* $p(x)$, d.h. die Wahrscheinlichkeit von $x$ ändert sich nicht, ob nun der Kryptotext $y$ bekannt ist oder nicht. Die Kenntnis von $y$ erlaubt somit keinerlei Rückschlüsse auf die gesendete Nachricht. Dies bedeutet, dass es dem Gegner nicht möglich ist, das System zu brechen; auch nicht mit unbegrenzten Rechenressourcen. Wie wir sehen werden, lässt sich dieses Maß an Sicherheit nur mit einem sehr hohen Aufwand erreichen.

Sind $p(x), p(y) > 0$, so gilt wegen $p(x|y)p(y) = p(x, y) = p(y|x)p(x)$ die Gleichheit

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad \text{(Satz von Bayes)}$$

und daher ist die Bedingung $p(x) = p(x|y)$ gleichbedeutend mit $p(y) = p(y|x)$, was wiederum mit der Unabhängigkeit der Ereignisse $X = x$ und $Y = y$ gleichbedeutend ist.

**Beispiel 64.** *Sei $KS = (M, C, E, D, K)$ ein Kryptosystem mit $M = \{x_1, \ldots, x_4\}$, $K = \{k_1, \ldots, k_4\}$, $C = \{y_1, \ldots, y_4\}$ und Verschlüsselungsfunktion*

| $E$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $k_1$ | $y_1$ | $y_4$ | $y_3$ | $y_2$ |
| $k_2$ | $y_2$ | $y_1$ | $y_4$ | $y_3$ |
| $k_3$ | $y_3$ | $y_2$ | $y_1$ | $y_4$ |
| $k_4$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ |

*Unter der Schlüssel- und Klartextverteilung*

| $k_i$ | $k_1$ | $k_2$ | $k_3$ | $k_4$ |
|---|---|---|---|---|
| $p(k_i)$ | 1/2 | 1/4 | 1/8 | 1/8 |

*bzw.*

| $x_i$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $p(x_i)$ | 1/2 | 1/6 | 1/6 | 1/6 |

*ergibt sich wegen $p(y) = \sum_{k,x:E(k,x)=y} p(k, x)$ folgende Kryptotextverteilung:*

$$\begin{aligned} p(y_1) &= 1/2 \cdot 1/2 + (1/4 + 1/8 + 1/8) \cdot 1/6 = 1/3 \\ p(y_2) &= 1/4 \cdot 1/2 + (1/8 + 1/8 + 1/2) \cdot 1/6 = 1/4 \\ p(y_3) &= 1/8 \cdot 1/2 + (1/8 + 1/2 + 1/4) \cdot 1/6 = 5/24 \\ p(y_4) &= 1/8 \cdot 1/2 + (1/2 + 1/4 + 1/8) \cdot 1/6 = 5/24 \end{aligned}$$

*Die bedingten Wahrscheinlichkeiten $p(x|y_1)$ berechnen sich wie folgt:*

$$\begin{aligned} p(x_1|y_1) &= p(k_1, x_1)/p(y_1) = (1/2)(1/2)/(1/3) = 3/4 \\ p(x_2|y_1) &= p(k_2, x_2)/p(y_1) = (1/4)(1/6)/(1/3) = 1/8 \\ p(x_3|y_1) &= p(k_3, x_3)/p(y_1) = (1/8)(1/6)/(1/3) = 1/16 \\ p(x_4|y_1) &= p(k_4, x_4)/p(y_1) = (1/8)(1/6)/(1/3) = 1/16 \end{aligned}$$

*Wegen $p(x_1) = 1/2 \neq 3/4 = p(x_1|y_1)$ ist das Kryptosystem nicht absolut sicher.* ◁

Lässt sich das Kryptosystem *KS* aus obigem Beispiel durch Verwendung eines anderen Schlüsselgenerators absolut sicher machen?

- Die Bedingung $p(x) = p(x|y)$ ist nach dem Satz von Bayes genau dann erfüllt, wenn $p(y) = p(y|x)$ ist.
- Da es jedoch in *KS* für jedes Paar $(x_i, y_j)$ genau einen Schlüssel $k = k_{i,j} \in K$ mit $E(k, x_i) = y_j$ gibt, also $p(y_j|x_i) = p(k_{i,j})$ ist, ist dies äquivalent zu $p(y_j) = p(k_{i,j})$.
- Für $j = 1$ bedeutet die Gleichheit $p(y_j) = p(k_{i,j})$ für alle $i$ zum Beispiel, dass alle vier Schlüssel $k_{i,1} = k_i$ $(i = 1, \ldots, 4)$ die gleiche Wahrscheinlichkeit haben müssen.
- Wegen $p(y_j) = \sum_{i=1}^{4} p(x_i)p(y_j|x_i) = 1/4 \sum_{i=1}^{4} p(x_i) = 1/4 = p(k_{i,j}) = p(y_j|x_i)$ ist das System in diesem Fall tatsächlich absolut sicher.

Demnach ist das Kryptosystem *KS* aus Beispiel 64 genau dann absolut sicher, wenn der Schlüssel gleichverteilt ist. In Verallgemeinerung dieses Beispiels lässt sich für eine wichtige Klasse von Kryptosystemen die absolute Sicherheit wie folgt charakterisieren.

**Satz 65.** *Sei $KS = (M, C, E, D, K, S)$ ein Kryptosystem mit $\|M\| = \|C\| = \|K\|$, dessen Schlüsselraum $K$ für jedes Klartext-Kryptotext-Paar $(x, y) \in M \times C$ genau einen Schlüssel $k$ mit $E(k, x) = y$ enthält. Dann ist KS genau dann absolut sicher, wenn $S$ auf $K$ gleichverteilt ist.*

*Beweis.* Bezeichne $k_{x,y}$ den eindeutigen Schlüssel, der den Klartext $x$ auf den Kryptotext $y$ abbildet. Falls $S$ auf $K$ gleichverteilt ist, folgt wegen $p(k_{x,y}) = \|K\|^{-1}$ für alle $x, y$ mit $p(x) > 0$ zunächst

$$p(y|x) = \sum_{k:E(k,x)=y} p(k) = p(k_{x,y}) = \|K\|^{-1}$$

und

$$p(y) = \sum_{x,p(x)>0} p(x)p(y|x) = \|K\|^{-1} \sum_{x} p(x) = \|K\|^{-1},$$

also $p(x, y) = p(x)p(y|x) = p(x)p(y)$, d.h. $KS$ ist absolut sicher. Die Umkehrung wird in den Übungen gezeigt. □

Verwendet man beim One-Time-Pad nur Klartexte einer festen Länge $n$, so ist dieser nach obigem Satz genau dann absolut sicher, wenn der Schlüssel unter Gleichverteilung gewählt wird. Variiert die Klartextlänge, so kann ein Gegner aus $y$ nur die Länge des zugehörigen Klartextes $x$ ableiten. Wird jedoch derselbe Schlüssel $k$ zweimal verwendet, so kann aus den Kryptotexten die Differenz der zugehörigen Klartexte ermittelt werden:

$$\left.\begin{array}{rcl} y_1 & = & E(x_1, k) = x_1 + k \\ y_2 & = & E(x_2, k) = x_2 + k \end{array}\right\} \rightsquigarrow y_1 - y_2 = x_1 - x_2$$

Sind die Klartexte natürlichsprachig, so können aus $y_1 - y_2$ die beiden Nachrichten $x_1$ und $x_2$ ähnlich wie bei der Analyse einer Lauftextverschlüsselung (siehe Abschnitt 2.5) rekonstruiert werden.

Da in einem absolut sicheren Kryptosystem der Schlüsselraum $K$ mindestens die Größe des Klartextraumes $X$ haben muss (siehe Übungen), erfordert die absolute Sicherheit einen extrem hohen Aufwand. Vor der Kommunikation muss ein Schlüssel, der mindestens so lang wie der zu übertragende Klartext ist, zufällig generiert und zwischen den Partnern auf einem sicheren Kanal ausgetauscht werden.

Für die meisten Anwendungen ist jedoch keine absolute Sicherheit erforderlich. Wie wir bei der Betrachtung von Stromsystemen gesehen haben, kann der Schlüsselstrom auch von einem Pseudo-Zufallsgenerator erzeugt werden. Dieser erhält als Eingabe eine Zufallszahl $s_0$ (den sogenannten *Keim*) und erzeugt daraus eine lange Folge $v_0\, v_1 \ldots$ von Pseudo-Zufallszahlen. Als Schlüssel muss jetzt nur noch der Keim ausgetauscht werden.

## 3.2 Der Entropiebegriff

In der Informationstheorie wird die Unsicherheit, mit der eine durch eine Zufallsvariable $X$ beschriebene Quelle ihre Nachrichten aussendet, nach ihrer Entropie bemessen. Dabei entspricht die Unsicherheit über $X$ genau dem Informationsgewinn, der sich aus der Beobachtung der Quelle $X$ ziehen lässt. Intuitiv ist die in einer einzelnen Nachricht $x$ steckende Information umso größer, desto unwahrscheinlicher sie ist. Tritt eine Nachricht $x$ mit einer positiven Wahrscheinlichkeit $p(x) = \Pr[X = x] > 0$ auf, dann ist

$$Inf_X(x) = \log_2(1/p(x))$$

der **Informationsgehalt** von $x$. Ist dagegen $p(x) = 0$, so sei $Inf_X(x) = 0$. Diese Definition des Informationsgehalts ergibt sich zwangsläufig aus den beiden folgenden Axiomen:

- Der (gemeinsame) Informationsgehalt $Inf_{X,Y}(x, y)$ von zwei Nachrichten $x$ und $y$, die aus unabhängigen Quellen $X$ und $Y$ stammen, ist $Inf_X(x) + Inf_Y(y)$.
- Eine Nachricht $x$, die mit Wahrscheinlichkeit $\Pr[X = x] = 1/2$ auftritt, hat den Informationsgehalt $Inf_X(x) = 1$.

Der Informationsgehalt wird in der Einheit bit (basic indissoluble information unit) gemessen. Die Entropie von $X$ ist nun der erwartete Informationsgehalt einer von $X$ generierten Nachricht.

**Definition 66.** *Sei $X$ eine Zufallsvariable mit Wertebereich $W(X) = \{x_1, \ldots, x_n\}$ und sei $p_i = \Pr[X = x_i]$. Dann ist die **Entropie** von $X$ definiert als*

$$\mathcal{H}(X) = \sum_{i=1}^{n} p_i \, Inf_X(x_i) = \sum_{i=1}^{n} p_i \log_2(1/p_i) = -\sum_{i=1}^{n} p_i \log_2(p_i).$$

**Beispiel 67.** *Sei $X$ eine Zufallsvariable mit der Verteilung*

| $x_i$ | *sonnig* | *leicht bewölkt* | *bewölkt* | *stark bewölkt* | *Regen* | *Schnee* | *Nebel* |
|---|---|---|---|---|---|---|---|
| $p_i$ | $1/4$ | $1/4$ | $1/8$ | $1/8$ | $1/8$ | $1/16$ | $1/16$ |

*Dann ergibt sich die Entropie von $X$ zu*

$$\mathcal{H}(X) = 1/4 \cdot (2 + 2) + 1/8 \cdot (3 + 3 + 3) + 1/16 \cdot (4 + 4) = 2{,}625. \qquad \triangleleft$$

Die Entropie nimmt für $p_1 = \cdots = p_n = 1/n$ den Wert $\log_2(n)$ an. Für jede andere Verteilung $p_1, \ldots, p_n$ gilt dagegen $\mathcal{H}(X) < \log_2(n)$ (Beweis siehe unten). Bei vorgegebener Größe des Wertebereichs von $X$ ist die Unsicherheit über $X$ um so größer, je gleichmäßiger $X$ verteilt ist. Bringt $X$ dagegen nur einen einzigen Wert mit positiver Wahrscheinlichkeit hervor, dann (und nur dann) nimmt $\mathcal{H}(X)$ den Wert 0 an. Für den Nachweis von oberen Schranken für die Entropie benutzen wir folgende Hilfsmittel aus der Analysis.

**Definition 68.** *Sei $I \subseteq \mathbb{R}$ ein Intervall. Eine Funktion $f : I \to \mathbb{R}$ heißt **konkav** auf $I$, falls für alle $x \neq y \in I$ und $0 \leq t \leq 1$ gilt:*

$$f(tx + (1 - t)y) \quad \geq \quad tf(x) + (1 - t)f(y).$$

*Gilt sogar „$>$“ anstelle von „$\geq$“, so heißt $f$ **streng konkav** auf $I$.*

**Beispiel 69.** *Die Funktion $f(x) = \log_2(x)$ ist streng konkav auf $(0, \infty)$.* $\triangleleft$

Für den Beweis des nächsten Satzes benötigen wir die Jensensche Ungleichung, die wir ohne Beweis angeben.

**Satz 70** (Jensensche Ungleichung)**.** *Sei $f$ eine streng konkave Funktion auf $I$ und seien $0 < a_1, \ldots, a_n < 1$ reelle Zahlen mit $\sum_{i=1}^{n} a_i = 1$. Dann gilt für alle $x_1, \ldots, x_n \in I$,*

$$f\left(\sum_{i=1}^{n} a_i x_i\right) \quad \geq \quad \sum_{i=1}^{n} a_i f(x_i).$$

*Hierbei tritt Gleichheit genau dann ein, wenn alle $x_i$ den gleichen Wert haben.*

**Satz 71.** *Sei $X$ eine Zufallsvariable auf einer $n$-elementigen Menge $\{x_1, \ldots, x_n\}$ mit der Verteilung $p_i = \Pr[X = x_i]$ für $i = 1, \ldots, n$. Dann ist $H(X) \leq \log_2(n)$, wobei Gleichheit genau im Fall $p_i = 1/n$ für $i = 1, \ldots, n$ eintritt.*

*Beweis.* Aufgrund der Jensenschen Ungleichung gilt

$$\mathcal{H}(X) = \sum_{i=1}^{n} p_i \log_2(1/p_i) \leq \log_2 \sum_{i=1}^{n} p_i/p_i = \log_2 n,$$

wobei Gleichheit genau im Fall $1/p_1 = \cdots = 1/p_n$ eintritt. Letzteres ist mit der Bedingung $p_i = 1/n$ für $i = 1, \ldots, n$ gleichbedeutend. □

Die Entropie liefert eine sehr gute untere Schranke für die mittlere Codewortlänge von Binärcodes. Ein **Binärcode** für $X$ ist eine (geordnete) Menge $C = \{y_1, \ldots, y_n\}$ von binären Codewörtern $y_i$ für die Nachrichten $x_i$ mit der Eigenschaft, dass die Abbildung $c : X^* \to \{0,1\}^*$ mit $c(x_{i_1} \cdots x_{i_k}) = y_{i_1} \cdots y_{i_k}$ injektiv ist. Die Injektivität von $c$ stellt sicher, dass jede Folge $y_{i_1} \cdots y_{i_k}$ von Codewörtern eindeutig decodierbar ist.

Die **mittlere Codewortlänge** von $C$ unter $X$ ist

$$L(C) = \sum_{i=1}^{n} p_i \cdot |y_i|.$$

$C$ heißt **optimal**, wenn kein anderer Binärcode für $X$ eine kürzere mittlere Codewortlänge besitzt. Für einen optimalen Binärcode $C$ für $X$ gilt (ohne Beweis)

$$\mathcal{H}(X) \leq L(C) < \mathcal{H}(X) + 1.$$

**Beispiel 72.** *Sei $X$ die Zufallsvariable aus dem letzten Beispiel mit der Verteilung $p_1 = p_2 = 1/4$, $p_3 = p_4 = p_5 = 1/8$ und $p_6 = p_7 = 1/16$. Betrachten wir die beiden Codes $C_1 = \{\mathtt{001}, \mathtt{010}, \mathtt{011}, \mathtt{100}, \mathtt{101}, \mathtt{110}, \mathtt{111}\}$ und $C_2 = \{\mathtt{00}, \mathtt{01}, \mathtt{100}, \mathtt{101}, \mathtt{110}, \mathtt{1110}, \mathtt{1111}\}$, so hat $C_1$ die mittlere Codewortlänge $L(C_1) = 3$, während $C_2 = \{y_1, \ldots, y_7\}$ wegen $|y_i| = \log_2(1/p_i)$ den optimalen Wert $L(C_2) = \mathcal{H}(X) = 2{,}625$ erreicht.* ◁

Die Redundanz eines Codes für eine Zufallsvariable $X$ ist um so höher, je größer seine mittlere Codewortlänge im Vergleich zur Entropie von $X$ ist. Um auch Codes über unterschiedlichen Alphabeten miteinander vergleichen zu können, ist es notwendig, die Codewortlänge in einer festen Einheit anzugeben. Hierzu definiert man die **Bitlänge** eines Wortes $x$ über einem Alphabet $A$ mit $m > 2$ Buchstaben zu $|x|_2 = |x| \log_2(m)$. Beispielsweise ist die Bitlänge von `GOLD` (über dem lateinischen Alphabet) $|\mathtt{GOLD}|_2 = 4 \log_2(26) = 18{,}8$. Entsprechend berechnet sich für einen Code $C = \{y_1, \ldots, y_n\}$ unter einer Verteilung $p_1, \ldots, p_n$ die mittlere Codewortlänge (in bit) zu

$$L_2(C) = \sum_{i=1}^{n} p_i \cdot |y_i|_2.$$

Damit können wir die Redundanz eines Codes als den mittleren Anteil der Codewortbuchstaben definieren, die keine Information tragen.

**Definition 73.** *Die **(relative) Redundanz** eines Codes $C$ für $X$ ist definiert als*

$$\mathcal{R}(C) = \frac{L_2(C) - \mathcal{H}(X)}{L_2(C)}.$$

**Beispiel 74.** *Während eine von $X$ generierte Nachricht im Durchschnitt $\mathcal{H}(X) = 2.625$ bit an Information enthält, haben die Codewörter von $C_1$ eine Bitlänge von 3. Der Anteil an „überflüssigen“ Zeichen pro Codewort beträgt also*

$$\mathcal{R}(C_1) = \frac{3 - 2.625}{3} = 12{,}5\%,$$

*wogegen $C_2$ keine Redundanz besitzt.* ◁

## 3.3 Redundanz von Sprachen

Auch Schriftsprachen wie Deutsch oder Englisch und Programmiersprachen wie C oder PASCAL können als eine Art Code aufgefasst werden. Es ist zu erwarten, dass eine Sprache umso mehr Redundanz aufweist, je restriktiver die Gesetzmäßigkeiten sind, unter denen in ihr Worte und Sätze gebildet werden. Um die statistischen Eigenschaften einer Sprache $L$ zu erforschen, erweist es sich als zweckmäßig, die Textstücke der Länge $n$ ($n$-Gramme) von $L$ für unterschiedliche $n$ getrennt voneinander zu betrachten. Sei also $L_n$ die Zufallsvariable, die die Verteilung aller $n$-Gramme in $L$ beschreibt. Interpretieren wir diese $n$-Gramme als Codewörter einer festen Codewortlänge $n$, so ist

$$\mathcal{R}(L_n) = \frac{n \log_2 m - \mathcal{H}(L_n)}{n \log_2 m}$$

die Redundanz dieses Codes.

**Definition 75** (**Entropie einer Sprache**). *Für eine Sprache $L$ über einem Alphabet $A$ mit $\|A\| = m$ ist $\mathcal{H}(L_n)/n$ die **$n$-Gramm-Entropie von** $L$ (pro Buchstabe). Falls dieser Wert für $n$ gegen $\infty$ von oben gegen einen Grenzwert*

$$\mathcal{H}(L) = \lim_{n\to\infty} \mathcal{H}(L_n)/n$$

*konvergiert, so wird dieser Grenzwert als die **Entropie von** $L$ bezeichnet. In diesem Fall konvergiert $\mathcal{R}(L_n)$ von unten gegen den Grenzwert*

$$\mathcal{R}(L) = \lim_{n\to\infty} \mathcal{R}(L_n) = \frac{\log_2 m - \mathcal{H}(L)}{\log_2 m},$$

*der als die **(relative) Redundanz** von $L$ bezeichnet wird. Der Zähler*

$$\mathcal{R}_{abs}(L) = \log_2 m - \mathcal{H}(L) = \mathcal{R}(L) \log_2 m$$

*wird auch als **absolute Redundanz** von $L$ bezeichnet (gemessen in bit/Zeichen).*

Die Redundanz von natürlichen Sprachen lässt sich näherungsweise bestimmen, indem man die Entropien $\mathcal{H}(L_n)$ ihrer $n$-Gramme empirisch ermittelt.

**Beispiel 76.** *Im Deutschen hat die Einzelzeichenverteilung eine Entropie von $\mathcal{H}(L_1) = 4,1$ bit, während eine auf $A_{lat}$ gleichverteilte Zufallsvariable $U$ einen Entropiewert von $\mathcal{H}(U) = \log(26) = 4,7$ bit hat. Für die Bigramme ergibt sich ein Entropiewert von $\mathcal{H}(L_2)/2 = 3,5$ bit pro Buchstabe. Mit wachsender Länge sinkt die Entropie von deutschsprachigen Texten weiter ab und strebt gegen einen Grenzwert $\mathcal{H}(L)$ von $1,5$ bit pro Buchstabe.*

| $n$ | $\mathcal{H}(L_n)$ | $\mathcal{H}(L_n)/n$ | $\mathcal{R}_{abs}(L_n)/n$ | $\mathcal{R}(L_n)$ |
|---|---|---|---|---|
| 1 | $4,1$ | $4,1$ | $0,6$ | 13% |
| 2 | $7,0$ | $3,5$ | $1,2$ | 26% |
| 3 | $9,6$ | $3,2$ | $1,5$ | 32% |
| 6 | $12,2$ | $2,0$ | $2,7$ | 57% |
| 15 | $27,6$ | $1,8$ | $2,9$ | 62% |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $\infty$ | $\infty$ | $\mathcal{H}(L) = 1,5$ | $\mathcal{R}_{abs}(L) = 3,2$ | $\mathcal{R}(L) = 67\%$ |

*Deutsche Texte hinreichender Länge besitzen also eine durchschnittliche Redundanz von ca. 67%, so dass ihre Länge bei optimaler Kodierung auf ca. 1/3 komprimierbar ist.* ◁

## 3.4 Die Eindeutigkeitsdistanz

Wir betrachten nun den Fall, dass mit einem Kryptosystem Klartexte einer variablen Länge $n$ verschlüsselt werden, ohne dabei den Schlüssel zu wechseln. Die Chiffrierfunktion hat also die Form

$$E_n : K \times A^n \to C_n,$$

wobei die Klartextlänge $n$ variabel ist und wir der Einfachheit halber annehmen, dass die Menge $C_n$ der entsprechenden Kryptotexte die gleiche Kardinalität $\|C_n\| = \|A^n\| = m^n$ wie der Klartextraum hat. Ist $y$ ein abgefangener Kryptotext, so ist

$$K(y) = \{k \in K \mid \exists x \in A^n : E_n(k, x) = y \wedge p(x) > 0\}$$

die Menge aller infrage kommenden Schlüssel. $K(y)$ besteht aus einem „echten“ (d. h. dem zur Generierung von $y$ tatsächlich benutzten) und $\|K(y)\| - 1$ so genannten „unechten“ Schlüsseln. Aus informationstheoretischer Sicht ist das Kryptosystem unter der Klartextverteilung $X$ umso sicherer, desto größer die erwartete Anzahl

$$\bar{s}_n = \sum_{y \in C_n} p(y) \cdot (\|K(y)\| - 1) = \sum_{y \in C_n} p(y) \cdot \|K(y)\| - 1$$

der unechten Schlüssel ist. Im besten Fall kommen für jeden Kryptotext alle Schlüssel infrage, d.h. $\bar{s}_n = \|K\| - 1$. Ist dagegen $\bar{s}_n$ gleich 0, so liefert der abgefangene Kryptotext dem Gegner genügend Information, um den benutzten Schlüssel und somit den zugehörigen Klartext eindeutig bestimmen zu können (sofern er über genügend Ressourcen verfügt).

**Definition 77.** *Die **Eindeutigkeitsdistanz** $n_0$ eines Kryptosystems unter Klartextverteilung $X$ ist der kleinste Wert von $n$, für den $\bar{s}_n = 0$ wird.*

Als nächstes wollen wir eine untere Schranke für $\bar{s}_n$ (und damit für $n_0$) herleiten. Hierzu benötigen wir den Begriff der bedingten Entropie $\mathcal{H}(X|Y)$ von $X$, wenn der Wert von $Y$ bereits bekannt ist.

**Definition 78.** *Seien $X, Y$ Zufallsvariablen. Die **bedingte Entropie** von $X$ unter $Y$ ist definiert als*

$$\mathcal{H}(X|Y) = \sum_{y \in W(Y)} p(y) \cdot \mathcal{H}(X|y),$$

*wobei $X|y$ die Zufallsvariable mit der Verteilung $p_y(x) = p(x|y) = \Pr[X = x \mid Y = y]$ ist (d.h. $X|y$ hat die Entropie $\mathcal{H}(X|y) = \sum_{x \in W(X)} p(x|y) \cdot \log_2(1/p(x|y))$.*

**Satz 79.** *Es gilt*

1. *$\mathcal{H}(X, Y) = \mathcal{H}(Y) + \mathcal{H}(X|Y)$ und*
2. *$\mathcal{H}(X, Y) \leq \mathcal{H}(X) + \mathcal{H}(Y)$, wobei Gleichheit genau dann eintritt, wenn $X$ und $Y$ unabhängig sind.*

*Beweis.* s. Übungen. □

**Korollar 80.** *Es gilt $\mathcal{H}(X|Y) \leq \mathcal{H}(X)$, wobei Gleichheit genau dann eintritt, wenn $X$ und $Y$ unabhängig sind.*

**Satz 81.** *In jedem Kryptosystem gilt für die Klartextentropie $\mathcal{H}(X)$, die Schlüsselentropie $\mathcal{H}(S)$ und die Kryptotextentropie $\mathcal{H}(Y)$ die Gleichung*

$$\mathcal{H}(S|Y) = \mathcal{H}(S) + \mathcal{H}(X) - \mathcal{H}(Y).$$

*Beweis.* Zunächst ist $\mathcal{H}(S|Y) = \mathcal{H}(S,Y) - \mathcal{H}(Y)$. Es reicht also zu zeigen, dass

$$\mathcal{H}(S,Y) = \mathcal{H}(S) + \mathcal{H}(X)$$

ist. Da bei Kenntnis des Schlüssels der Wert von $X$ bereits eindeutig durch $Y$ und der Wert von $Y$ eindeutig durch $X$ festgelegt ist, folgt unter Berücksichtigung der gemachten Annahme, dass $X$ und $S$ unabhängig sind,

$$\mathcal{H}(S,Y) = \mathcal{H}(S,X,Y) - \underbrace{\mathcal{H}(X|S,Y)}_{=0} = \mathcal{H}(S,X) + \underbrace{\mathcal{H}(Y|S,X)}_{=0} = \mathcal{H}(S) + \mathcal{H}(X).$$

□

Jetzt verfügen wir über alle Hilfsmittel, um die erwartete Anzahl

$$\bar{s}_n = \sum_{y \in C_n} p(y) \cdot \|K(y)\| - 1$$

der unechten Schlüssel nach unten abschätzen zu können.

**Lemma 82.** *Seien $X_n$ und $Y_n$ die Zufallsvariablen, die die Verteilungen der $n$-Gramme der Klartextsprache und der zugehörigen Kryptotexte beschreiben. Dann gilt*

1. $\mathcal{H}(S|Y_n) \leq \log_2(\bar{s}_n + 1)$,
2. $\mathcal{H}(S|Y_n) \geq \mathcal{H}(S) - n\mathcal{R}(L_n)\log_2 m$.

*Beweis.*

1. Unter Verwendung der Jensenschen Ungleichung folgt

$$\begin{aligned}\mathcal{H}(S|Y_n) &= \sum_{y \in C_n} p(y) \cdot \mathcal{H}(S|y) \leq \sum_{y \in C_n} p(y) \cdot \log_2 \|K(y)\| \leq \log_2 \sum_{y \in C_n} p(y) \cdot \|K(y)\| \\ &= \log_2(\bar{s}_n + 1).\end{aligned}$$

2. Mit Satz 81 folgt

$$\mathcal{H}(S|Y_n) = \mathcal{H}(S) + \mathcal{H}(X_n) - \mathcal{H}(Y_n).$$

Für die Klartextentropie $\mathcal{H}(X_n)$ gilt

$$\mathcal{H}(X_n) = \mathcal{H}(L_n) = (1 - \mathcal{R}(L_n))n\log_2 m,$$

wobei $m = \|A\|$ ist. Zudem lässt sich die Kryptotextentropie $\mathcal{H}(Y_n)$ wegen $W(Y_n) = C_n$ und $\|C_n\| = m^n$ durch

$$\mathcal{H}(Y_n) \leq n\log_2 m$$

abschätzen. Somit ist

$$\mathcal{H}(S|Y_n) = \mathcal{H}(S) + \mathcal{H}(X_n) - \mathcal{H}(Y_n) \geq \mathcal{H}(S) - n\mathcal{R}(L_n)\log_2 m$$ □

Zusammen ergibt sich also

$$\log_2(\bar{s}_n + 1) \geq \mathcal{H}(S) - n\mathcal{R}(L_n)\log_2 m \geq \mathcal{H}(S) - n\mathcal{R}(L)\log_2 m.$$

Im Fall eines gleichverteilten Schlüssels erreicht $\mathcal{H}(S)$ den maximalen Wert $\log_2 \|K\|$, was auf die gesuchte Abschätzung für $\bar{s}_n$ führt.

**Satz 83.** *Werden mit einem Kryptosystem $(M, C, E, D, K)$ mit $M = A^n$ und $\|C\| = m^n$ Klartexte einer Sprache $L$ der festen Länge $n$ mit gleichverteiltem Schlüssel $k \in K$ verschlüsselt, so gilt für die erwartete Anzahl $\bar{s}_n$ der unechten Schlüssel,*

$$\bar{s}_n \geq \frac{\|K\|}{m^{n\mathcal{R}(L_n)}} - 1.$$

Setzen wir in obiger Abschätzung $\bar{s}_n = 0$, so erhalten wir folgende untere Schranke für die Eindeutigkeitsdistanz $n_0$ eines Kryptosystems.

**Korollar 84.** *Unter den Bedingungen des obigen Satzes gilt*

$$n_0 \geq \frac{\log_2 \|K\|}{\mathcal{R}(L_n) \log_2 m} \geq \frac{\log_2 \|K\|}{\mathcal{R}(L) \log_2 m} = \frac{\log_2 \|K\|}{\mathcal{R}_{abs}(L)}.$$

Man beachte, dass wir die Mindestmenge an Kryptotext, der zur eindeutigen Bestimmung des Schlüssels benötigt wird, nur nach unten abgeschätzt haben und die tatsächlich benötigte Menge deutlich größer sein kann. Natürlich erlaubt die eindeutige Bestimmung des Schlüssels auch die eindeutige Bestimmung des Klartexts. Unter Umständen kann jedoch der Klartext auch schon mit einer wesentlich geringeren Menge an Kryptotext eindeutig rekonstruierbar sein.

**Beispiel 85.** *Für Substitutionen bei deutschsprachigem Klartext ergeben sich folgende Werte $\log_2 \|K\| / \mathcal{R}_{abs}(L)$ als untere Schranke für die Eindeutigkeitsdistanz $n_0$ (wobei wir von einer absoluten Redundanz von $\mathcal{R}_{abs}(L) = 3.2$ bit/Zeichen ausgehen, was einer relativen Redundanz von $\mathcal{R}(L) = 3,2/4,7 \approx 67\%$ entspricht):*

| *Kryptosystem* | *Schlüsselanzahl* $\|K\|$ | $\log_2 \|K\|$ | $\log_2 \|K\| / \mathcal{R}_{abs}(L)$ |
|---|---|---|---|
| *additive Chiffre* | 26 | 4.7 | $\frac{4.7}{3.2} \approx 1.5$ |
| *affine Chiffre* | $12 \cdot 26 = 312$ | 8.3 | 2.6 |
| *einfache Substitution* | 26! | 88.4 | 27.6 |
| *Vigenère-Chiffre* | $26^d$ | $4.7 \cdot d$ | $1.5 \cdot d$ |

*Dagegen erhalten wir für Blocktranspositionen folgende unteren Schranken für die Mindestmenge an Kryptotext, die zur eindeutigen Schlüsselbestimmung benötigt wird. Hierbei unterscheiden wir zusätzlich nach der Länge der bei der Häufigkeitsanalyse benutzten $n$-Gramme. Dies entspricht der Situation, dass die Wahrscheinlichkeit jedes Zeichens im Klartext höchstens von den $n-1$ vorausgehenden bzw. nachfolgenden Zeichen abhängt.*

| *Untere Schranken für $n_0$ bei einer Analyse von Blocktranspositionen auf der Basis von* | | *Blocklänge $\ell$* | | | | |
|---|---|---|---|---|---|---|
| | | *10* | *20* | *50* | *100* | *1 000* |
| *Einzelzeichen-Häufigkeiten* | $(\mathcal{R}(L_1) = 0,6)$ | *59* | *165* | *578* | *1415* | *22 986* |
| *Bigramm-Häufigkeiten* | $(\mathcal{R}(L_2) = 1,2)$ | *40* | *111* | *390* | *954* | *15 502* |
| *Trigramm-Häufigkeiten* | $(\mathcal{R}(L_3) = 1,5)$ | *24* | *65* | *226* | *553* | *9 473* |
| *$n$-Gramm-Häufigkeiten, $n \to \infty$* | $(\mathcal{R}(L) = 3,2)$ | *7* | *19* | *67* | *164* | *2 665* |

◁

## 3.5 Weitere Sicherheitsbegriffe

Da die Benutzung eines informationstheoretisch sicheren Kryptosystems einen immensen Aufwand erfordert, begnügt man sich in der Praxis meist mit schwächeren Sicherheitsanforderungen.

- Ein Kryptosystem gilt als **komplexitätstheoretisch sicher** oder als **berechnungssicher** (***computationally secure***), falls es dem Gegner nicht möglich ist, das System mit einem für ihn lohnenswerten Aufwand zu brechen. Das heißt, der Zeitaufwand und die Kosten für einen erfolgreichen Angriff (sofern er überhaupt möglich ist) übersteigen den potentiellen Nutzen bei weitem.
- Ein Kryptosystem gilt als **nachweisbar sicher** (***provably secure***), wenn seine Sicherheit mit bekannten komplexitätstheoretischen Hypothesen verknüpft werden kann, deren Gültigkeit gemeinhin akzeptiert wird.
- Als **praktisch sicher** (***practically secure***) werden dagegen Kryptosysteme eingestuft, die über mehrere Jahre hinweg jedem Versuch einer erfolgreichen Kryptoanalyse widerstehen konnten, obwohl sie bereits eine weite Verbreitung gefunden haben und allein schon deshalb ein attraktives Ziel für einen Angriff darstellen.

Die komplexitätstheoretische Analyse eines Kryptosystems ist äußerst schwierig, da der Aufwand für einen erfolgreichen Angriff unabhängig von der dabei benutzten Technik abgeschätzt werden muss. Es reicht also nicht, alle bekannten kryptoanalytischen Ansätze in Betracht zu ziehen, sondern alle *möglichen*. Dabei darf sich die Aufwandsanalyse nicht ausschließlich an einer vollständigen Rekonstruktion des Klartextes orientieren, da bereits ein kleiner Unterschied zwischen der A-posteriori- und A-priori-Wahrscheinlichkeit für den Gegner einen Vorteil bedeuten könnte.

Aus den genannten Gründen ist noch für kein praktikables Kryptosystem der Nachweis gelungen, dass es komplexitätstheoretisch sicher ist. Damit ist auch nicht so schnell zu rechnen, zumindest nicht solange der Status fundamentaler komplexitätstheoretischer Fragen wie etwa des berühmten $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$-Problems offen ist. Dagegen gibt es eine ganze Reihe praktikabler Kryptosysteme, die als nachweisbar sicher oder praktisch sicher gelten.

Wir schließen diesen Abschnitt mit einer Präzisierung des komplexitätstheoretischen Sicherheitsbegriffs, die unter dem Namen IND-CPA (indistinguishability under a chosen-plaintext attack) bekannt ist. Hierzu ist es erforderlich, die Verletzung der Vertraulichkeit als ein algorithmisches Problem für den Gegner zu formulieren. Konkret läuft ein IND-CPA-Angriff wie folgt ab.

1. Zuerst wählt der Gegner zwei Klartexte $x_0 \neq x_1 \in M$.
2. Dann wird $x_0$ oder $x_1$ zufällig ausgewählt und der zugehörige Kryptotext $y$ gebildet.
3. Dem Gegner wird der Kryptotext $y$ vorgelegt und er muss raten, welcher der beiden Klartexte sich hinter $y$ verbirgt.
4. Der Angriff ist erfolgreich, falls der Gegner richtig rät.

Die Erfolgsaussichten des Gegners bei diesem Angriff lassen sich wie folgt formalisieren. Dabei gehen wir davon aus, dass das gewünschte Maß an Sicherheit durch einen Parameter $s \in \mathbb{N}$ reguliert wird. Typischerweise werden Kryptosysteme nach ihrer Schlüssellänge $s = |k|$ parameterisiert. Aus Praktikabilitätsgründen sollten dann alle legalen Operationen (wie die Chiffrierung oder die Schlüsselgenerierung) effizient (d.h. in Zeit $s^{O(1)}$) durchführbar sein. Natürlich darf dann auch der Aufwand des Gegners in Abhängigkeit von $s$ steigen, weshalb er zusätzlich den Parameterwert $s$ erhält.

**Definition 86 (IND-CPA Angriff).** *Sei $(M, C, E, D, K, S)$ ein Kryptosystem mit Sicherheitsparameter $s \in \mathbb{N}$. Ein (**IND-CPA-**)**Gegner** ist ein Tripel $G = (X_0, X_1, V)$ von probabilistischen Algorithmen, wobei $X_0, X_1$ bei Eingabe $s$ zwei Klartexte aus $M$ generieren und $V$ bei Eingabe von $s$, $x_0, x_1 \in M$ und $y \in C$ ein Bit $V(s, x_0, x_1, y) \in \{0, 1\}$ ausgibt. Der **Vorteil** von $G$ bei Parameterwert $s$ ist*

$$\alpha_G(s) = 2(\Pr[V(X_0(s), X_1(s), E(S, X_B(s))) = B] - 1/2),$$

*wobei $B$ auf $\{0, 1\}$ gleichverteilt und von $S, X_0, X_1, V$ unabhängig ist.*

Ist der Wert des Sicherheitsparameters $s$ irrelevant, fest vorgegeben oder aus dem Kontext ersichtlich, so verzichten wir meist auf seine explizite Angabe.

Wird beispielsweise eine Folge von Klartextblöcken $a_1, a_2, \ldots$ mit einer Blockchiffre verschlüsselt, indem die einzelnen Blöcke unabhängig voneinander mit demselben Schlüssel $k$ zu einer Folge $b_1, b_2, \ldots$ von Kryptotextblöcken $b_i = E(k, a_i)$ verschlüsselt werden (so genannter *ECB-Modus*; electronic code book mode), so kann ein Gegner ohne großen Aufwand einen Vorteil von 1 erzielen (d.h. mit Wahrscheinlichkeit 1 den richtigen Klartext raten). Hierzu wählt er (deterministisch) zwei beliebige Klartexte $x_0 = a_1 a_2 \ldots$ und $x_1 = a_1' a_2' \ldots$ mit der Eigenschaft $a_1 = a_2$ und $a_1' \neq a_2'$. Dann kann er bei Vorlage eines Kryptotextes $y = b_1 b_2 \ldots$ leicht erkennen, aus welchem Klartext $y$ generiert wurde:

$$V(x_0, x_1, y) = \begin{cases} 0, & b_1 = b_2 \\ 1, & \text{sonst.} \end{cases}$$

Erwartungsgemäß sind absolut sichere Kryptosysteme gegen IND-CPA-Angriffe resistent.

**Satz 87.** *Der maximale Vorteil gegenüber einem absolut sicheren Kryptosystem ist gleich Null (d.h. ein IND-DPA-Gegner kann höchstens mit Wahrscheinlichkeit $1/2$ den richtigen Klartext raten, auch wenn er über unbeschränkte Rechenressourcen verfügt).*

*Beweis.* Bei einem absolut sicheren Kryptosystem sind der Kryptotext $Y = E(S, X)$ und der Klartext $X$ unabhängig. Daher sind auch die Zufallsvariablen $V(X_0, X_1, E(S, X_B))$ und $B$ unabhängig und es folgt

$$\begin{aligned}
&\Pr[V(X_0, X_1, E(S, X_B)) = B] \\
&= \Pr[V(X_0, X_1, E(S, X_B)) = B = 0] + \Pr[V(X_0, X_1, E(S, X_B)) = B = 1] \\
&= \Pr[V(X_0, X_1, E(S, X_B)) = 0] \cdot \underbrace{\Pr[B = 0 \mid V(X_0, X_1, E(S, X_B)) = 0]}_{= \Pr[B=0] = 1/2} \\
&\quad + \Pr[V(X_0, X_1, E(S, X_B)) = 1] \cdot \underbrace{\Pr[B = 1 \mid V(X_0, X_1, E(S, X_B)) = 1]}_{= \Pr[B=1] = 1/2} \\
&= 1/2.
\end{aligned}$$

□

In den Übungen wird auch die umgekehrte Implikation bewiesen. Ein Kryptosystem ist somit genau dann absolut sicher, wenn kein Gegner einen Vorteil größer als 0 erzielen kann. Für die Präzisierung der komplexitätstheoretischen Sicherheit sind nun die folgenden beiden Fragen von entscheidender Bedeutung:

1. Über welche Rechenressourcen verfügt ein Gegner realistischerweise?
2. Wie groß darf der vom Gegner erzielbare Vorteil höchstens sein, ohne die Vertraulichkeit der verschlüsselten Nachricht zu verletzen?

Bezüglich Frage 1 geht man typischerweise davon aus, dass der Gegner über probabilistische Schaltkreise polynomieller Größe verfügt.

**Definition 88.**

*a) Ein* ***boolescher Schaltkreis*** *der Größe $m$ mit Eingängen $x_1, \ldots, x_n$ und Ausgängen $i_1, \ldots, i_l \in [m]$ ist eine Folge $c = (g_1, \ldots, g_m)$ von* ***Gattern***

$$g_l \in \{0, 1, x_1, \ldots, x_n, (\neg, j), (\wedge, j, k), (\vee, j, k)\} \text{ mit } 1 \leq j, k < l.$$

*b) Der von $c$ bei Eingabe $a \in \{0,1\}^n$ am Gatter $g_l$ berechnete Wert $g_l(a) \in \{0,1\}$ ist induktiv wie folgt definiert:*

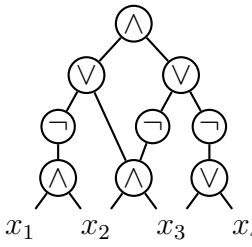
| $g_l$ | $0$ | $1$ | $x_i$ | $(\neg, j)$ | $(\wedge, j, k)$ | $(\vee, j, k)$ |
|---|---|---|---|---|---|---|
| $g_l(a)$ | $0$ | $1$ | $a_i$ | $1 - g_j(a)$ | $g_j(a)g_k(a)$ | $g_j(a) + g_k(a) - g_j(a)g_k(a)$ |

*c) Die* ***Ausgabe*** *von $c$ bei Eingabe $a \in \{0,1\}^n$ ist die Bitfolge $c(a) = g_{i_1}(a) \ldots g_{i_l}(a)$.*

**Beispiel 89.** *Der Schaltkreis*

$$\begin{aligned} c = \quad & (x_1, x_2, x_3, x_4, (\wedge, 1, 2), (\wedge, 2, 3), \\ & (\vee, 3, 4), (\neg, 5), (\neg, 6), (\neg, 7), \\ & (\vee, 6, 8), (\vee, 9, 10), (\wedge, 11, 12)) \end{aligned}$$

*mit den Eingängen $x_1, x_2, x_3, x_4$ und Ausgängen $(11, 12, 13)$ gibt bei Eingabe $a = 0110$ die Bitfolge $c(0110) = 100$ aus.*




◁

Ein **probabilistischer Schaltkreis** $c$ hat neben den regulären Eingabegattern $x_1, \ldots, x_n$ noch eine beliebige Anzahl von Zufallsgattern $z_1, \ldots, z_m$. Hierbei werden die Eingabegatter $x_i$ wie bisher mit den Bits $a_i$ eines Eingabevektors $a = a_1 \ldots a_n \in \{0,1\}^n$ belegt, während die $m$ Zufallsgatter unabhängig gleichverteilte Bits $Z_1, \ldots, Z_m$ erzeugen (d.h. es gilt $\Pr[Z_1 \ldots Z_m = b] = 2^{-m}$ für alle $b \in \{0,1\}^m$). Dadurch wird die Ausgabe $c(a, Z_1, \ldots, Z_m)$ zu einer Zufallsvariablen, die wir auch kurz mit $C(a)$ bezeichnen.

Bezüglich der zweiten Frage verlangt man, dass der Gegner für jedes Polynom $p$ höchstens für endlich viele Parameterwerte $s$ einen Vorteil größer gleich $1/p(s)$ erzielen darf. Andernfalls wäre die Sicherheit gefährdet, da er für jedes solche $s$ nach polynomiell vielen Wiederholungen der probabilistischen Berechnung von $V(s, x_0, x_1, y)$ fast sicher den richtigen Klartext ausfindig machen könnte, indem er das mehrheitlich berechnete Bit ausgibt.

**Definition 90.** *Sei KS ein Kryptosystem mit variablem Sicherheitsparameter $s \in \mathbb{N}$.*

- *Eine Funktion $\varepsilon : \mathbb{N} \to \mathbb{R}$ heißt* ***vernachlässigbar****, wenn für jedes Polynom $p$ eine Zahl $n_0 \in \mathbb{N}$ existiert, so dass $\varepsilon(n) < 1/p(n)$ für alle $n \geq n_0$ gilt.*
- *Ein Gegner $G = (X_0, X_1, V)$ heißt* ***effizient****, wenn probabilistische Schaltkreise $c$ und $c'$ der Größe $s^{O(1)}$ mit $C(s) = (X_0(s), X_1(s))$ und $C'(s, x_0, x_1, y) = V(s, x_0, x_1, y)$ existieren, wobei die Ein- und Ausgaben von $c$ und $c'$ binär kodiert sind.*
- *KS heißt* ***komplexitätstheoretisch sicher****, wenn jeder effiziente Gegner $G$ nur einen vernachlässigbaren Vorteil erzielen kann (d.h. die Funktion $\alpha_G(s)$ ist vernachlässigbar).*

# 4 Moderne symmetrische Kryptosysteme & ihre Analyse

## 4.1 Produktchiffren

Produktchiffren erhält man durch die sequentielle Anwendung mehrerer Verschlüsselungsverfahren. Sie können extrem schwer zu brechen sein, auch wenn die einzelnen Komponenten leicht zu brechen sind.

**Definition 91.** *Seien $KS_1 = (M_1, C_1, E_1, D_1, K_1, S_1)$ und $KS_2 = (M_2, C_2, E_2, D_2, K_2, S_2)$ Kryptosysteme mit $C_1 = M_2$. Dann ist das **Produktkryptosystem** $KS_1 \times KS_2$ von $KS_1$ und $KS_2$ definiert als $(M_1, C_2, E, D, K_1 \times K_2, S)$ mit $S = (S_1, S_2)$ und*

$$E(k_1, k_2; x) = E_2(k_2, E_1(k_1, x)) \text{ sowie } D(k_1, k_2; y) = D_1(k_1, D_2(k_2, y))$$

*für alle $x \in M_1$, $y \in C_2$ und $(k_1, k_2) \in K_1 \times K_2$.*

Der Schlüsselraum von $KS_1 \times KS_2$ umfasst also alle Paare $(k_1, k_2)$ von Schlüsseln $k_1 \in K_1$ und $k_2 \in K_2$, wobei wir voraussetzen, dass die Schlüssel unabhängig gewählt werden (d.h. es gilt $p(k_1, k_2) = p(k_1)p(k_2)$).

**Beispiel 92.** *Sei $A = \{a_0, \ldots, a_{m-1}\}$. Man sieht leicht, dass die affine Chiffre $KS = (M, C, K, E, D)$ mit $M = C = \mathcal{A}$ und $K = \mathbb{Z}_m^* \times \mathbb{Z}_m$ das Produkt $KS = KS_1 \times KS_2$ der multiplikativen Chiffre $KS_1 = (M, C, K_1, E_1, D_1)$ und der additiven Chiffre $KS_2 = (M, C, K_2, E_2, D_2)$ ist, da für jeden Schlüssel $k = (k_1, k_2) \in K = K_1 \times K_2 = \mathbb{Z}_m^* \times \mathbb{Z}_m$ gilt:*

$$E(k, x) = k_1 x + k_2 = E_2(k_2, E_1(k_1, x)).$$

*Das ist exakt die affine Chiffre. Welche Chiffre erhalten wir, wenn wir die Reihenfolge von $KS_1$ und $KS_2$ vertauschen? Für $KS' = KS_2 \times KS_1$ ergibt sich das Kryptosystem $KS' = (M, C, K', E', D')$ mit $K' = K_2 \times K_1 = \mathbb{Z}_m \times \mathbb{Z}_m^*$ und*

$$E'(k_2, k_1; x) = k_1(x + k_2) = k_1 x + k_1 k_2 = E(k_1, k_1 k_2; x)$$

*für jeden Schlüssel $(k_2, k_1) \in K'$. Wir sehen also, dass die Abbildung*

$$(k_2, k_1) \mapsto (k_1, k_1 k_2)$$

*eine Bijektion zwischen den Schlüsselräumen $K'$ und $K$ ist und der Schlüssel $(k_2, k_1)$ im System $KS'$ die gleiche Chiffrierfunktion realisiert wie der Schlüssel $(k_1, k_1 k_2)$ in $KS$. Zudem können wir jeden Schlüsselgenerator $S'$ für $KS'$ in einen Schlüsselgenerator $S$ für $KS$ transformieren (und auch $S$ wieder zurück in $S'$), so dass $S$ in $KS$ jede Chiffrierfunktion mit der gleichen Wahrscheinlichkeit erzeugt wie $S'$ in $KS'$. Daher können wir die Kryptosysteme $KS = KS_1 \times KS_2$ und $KS' = KS_2 \times KS_1$ als gleich (genauer: äquivalent, siehe Übungen) ansehen, d.h. $KS_1$ und $KS_2$ kommutieren.* ◁

**Definition 93.** *Ein Kryptosystem $KS = (M, C, K, D, E)$ mit $M = C$ heißt **endomorph**. Ein endomorphes Kryptosystem $KS$ heißt **idempotent**, falls $KS \times KS$ äquivalent zu $KS$ ist (in Zeichen: $KS \times KS = KS$).*

**Beispiel 94.** *Eine leichte Rechnung zeigt, dass die additive Chiffre, die multiplikative Chiffre und die affine Chiffre idempotent sind. Ebenso die Blocktransposition sowie die Vigenère- und Hill-Chiffre.* ◁

Will man durch mehrmalige Anwendung (Iteration) derselben Chiffriermethode eine höhere Sicherheit erreichen, so darf diese nicht idempotent sein. Man kann beispielsweise versuchen, ein nicht idempotentes System $KS$ durch die Kombination $KS = KS_1 \times KS_2$ zweier idempotenter Verfahren $KS_1$ und $KS_2$ zu erhalten. Da $KS$ im Fall $KS_1 \times KS_2 = KS_2 \times KS_1$ wegen

$$\begin{aligned}(KS_1 \times KS_2) \times (KS_1 \times KS_2) &= KS_1 \times (KS_2 \times KS_1) \times KS_2 \\ &= KS_1 \times (KS_1 \times KS_2) \times KS_2 \\ &= (KS_1 \times KS_1) \times (KS_2 \times KS_2) \\ &= KS_1 \times KS_2\end{aligned}$$

idempotent ist, dürfen hierbei $KS_1$ und $KS_2$ jedoch nicht kommutieren.

Im Rest dieses Kapitels werden wir nur noch das Binäralphabet $A = \{0, 1\}$ als Klar- und Kryptotextalphabet benutzen und auch der Schlüsselraum wird von der Form $\{0, 1\}^k$ sein, wobei $k$ die Schlüssellänge bezeichnet. Einzelne Schlüssel eines Kryptosystems werden wir in diesem Kapitel mit $K$ bezeichnen.

Eine iterierte Blockchiffre wird typischerweise durch eine **Rundenfunktion** (*round function*) $g$ und einen **Key Schedule Algorithmus** $f$ beschrieben. Ist $N$ die Rundenzahl, so erzeugt $f$ bei Eingabe eines Schlüssels $K$ eine Folge $f(K) = (K^1, \ldots, K^N)$ von $N$ Rundenschlüsseln $K^i$ für $g$. Mit diesen wird ein Klartext $x = w^0$ durch $N$-malige Anwendung der Rundenfunktion $g$ zu einem Kryptotext $y = w^N$ verschlüsselt:

$$\begin{aligned}w^1 &:= g(K^1, w^0) \\ &\vdots \\ w^N &:= g(K^N, w^{N-1})\end{aligned}$$

Um $y$ wieder zu entschlüsseln, muss die inverse Rundenfunktion $g^{-1}$ mit umgekehrter Rundenschlüsselfolge $K^N, \ldots, K^1$ benutzt werden:

$$\begin{aligned}w^{N-1} &:= g^{-1}(K^N, w^N) \\ &\vdots \\ w^0 &:= g^{-1}(K^1, w^1)\end{aligned}$$

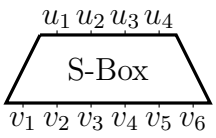
Beispiele für iterierte Chiffren sind der aus 16 Runden bestehende DES-Algorithmus und der AES mit einer variablen Rundenzahl $N \in \{10, 12, 14\}$, die wir in späteren Abschnitten behandeln werden.

## 4.2 Substitutions-Permutations-Netzwerke

In diesem Abschnitt betrachten wir den prinzipiellen Aufbau von iterierten Blockchiffren. Als Basisbausteine für die Rundenfunktion eignen sich Substitutionen und Transpositionen besonders gut. Aus Effizienzgründen sollten die Substitutionen nur eine relativ kleine Blocklänge $\ell$ haben.

**Definition 95.** *Für ein Wort $u = u_1 \cdots u_n \in \{0, 1\}^n$ und Indizes $1 \leq i \leq j \leq n$ bezeichne $u[i, j]$ das **Teilwort** $u_i \cdots u_j$ von $u$. Im Fall $n = lm$ bezeichnen wir das Teilwort $u[(i-1)l + 1, il]$ auch einfach mit $u_{(i)}$, d.h. es gilt $u = u_{(1)} \cdots u_{(m)}$, wobei $|u_{(i)}| = l$ ist.*

Sei $\sigma_S : \{0,1\}^l \to \{0,1\}^{l'}$ eine Substitution, die Binärblöcke $u$ der Länge $l$ in Binärblöcke $v = \sigma_S(u)$ der Länge $l'$ überführt (engl. auch als **S-Box** bezeichnet).




Durch parallele Anwendung von $m$ dieser S-Boxen erhalten wir folgende Substitution $S : \{0,1\}^{lm} \to \{0,1\}^{l'm}$,

$$S(u_1 \cdots u_{lm}) = \sigma_S(u_{(1)}) \cdots \sigma_S(u_{(m)}).$$

Für die Speicherung einer S-Box $\sigma_S : \{0,1\}^l \to \{0,1\}^{l'}$ auf einem Speicherchip werden $l'2^l$ Bit Speicherplatz benötigt (im Fall $l = l'$ also $l2^l$ Bit). Für $l = l' = 16$ wären dies beispielsweise $2^{20}$ Bit, was Smartcard-Anwendungen bereits ausschließen würde.

Für eine Transposition $P$ auf $\{0,1\}^{lm}$ bezeichnen wir die zugehörige Permutation auf $\{1,\ldots,lm\}$ mit $\pi_P$ oder einfach mit $\pi$, falls $P$ aus dem Kontext bekannt ist, d.h.

$$P(u_1 \cdots u_{lm}) = u_{\pi(1)} \cdots u_{\pi(lm)}.$$

**Definition 96.** *Sei $M = C = \{0,1\}^{lm}$ für natürliche Zahlen $l, m \geq 1$. Ein **Substitutions-Permutations-Netzwerk** (SPN) wird durch Permutationen $\sigma_S$ auf $\{0,1\}^l$ (S-Box) und $\pi_P$ auf $\{1,\ldots,lm\}$ (Blocktransposition) sowie durch eine Funktion $f : \{0,1\}^k \to \{0,1\}^{lm(N+1)}$ (Key Schedule Algorithmus) beschrieben. Die Funktion $f$ transformiert einen (externen) Schlüssel $K \in \{0,1\}^k$ in eine Folge $f(K) = (K^1, \ldots, K^{N+1})$ von $N+1$ Rundenschlüsseln $K^r$, $r = 1, \ldots, N+1$, unter denen ein Klartext $x \in \{0,1\}^{lm}$ durch folgenden Algorithmus in einen Kryptotext $y = E_{f,\sigma_S,\pi_P}(K,x) \in \{0,1\}^{lm}$ überführt wird.*

**Chiffrierfunktion** $E_{f,\sigma_S,\pi_P}(K,x)$

```
1  w^0 := x
2  for r := 1 to N − 1 do
3     u^r := w^{r−1} ⊕ K^r
4     v^r := S(u^r)
5     w^r := P(v^r)
6  u^N := w^{N−1} ⊕ K^N
7  v^N := S(u^N)
8  y := v^N ⊕ K^{N+1}
```

Zu Beginn jeder Runde $r \in \{1,\ldots,N\}$ wird $w^{r-1}$ zunächst einer XOR-Operation mit dem Rundenschlüssel $K^r$ unterworfen (dies wird *round key mixing* genannt), deren Resultat $u^r$ den S-Boxen zugeführt wird. Auf die Ausgabe $v^r$ der S-Boxen wird in jeder Runde $r \leq N-1$ die Transposition $P$ angewendet, was die Eingabe $w^r$ für die nächste Runde $r+1$ liefert.

Am Ende der letzten Runde $r = N$ wird nicht die Transposition $P$ angewandt, sondern der Rundenschlüssel $K^{N+1}$ auf $v^N$ addiert. Durch diese (*whitening* genannte) Vorgehensweise wird einerseits erreicht, dass auch für den letzten Chiffrierschritt der Schlüssel benötigt und somit der Gegner von einer partiellen Entschlüsselung des Kryptotexts abgehalten wird. Zum Zweiten ermöglicht dies eine (legale) Entschlüsselung nach fast demselben Verfahren (siehe Übungen).

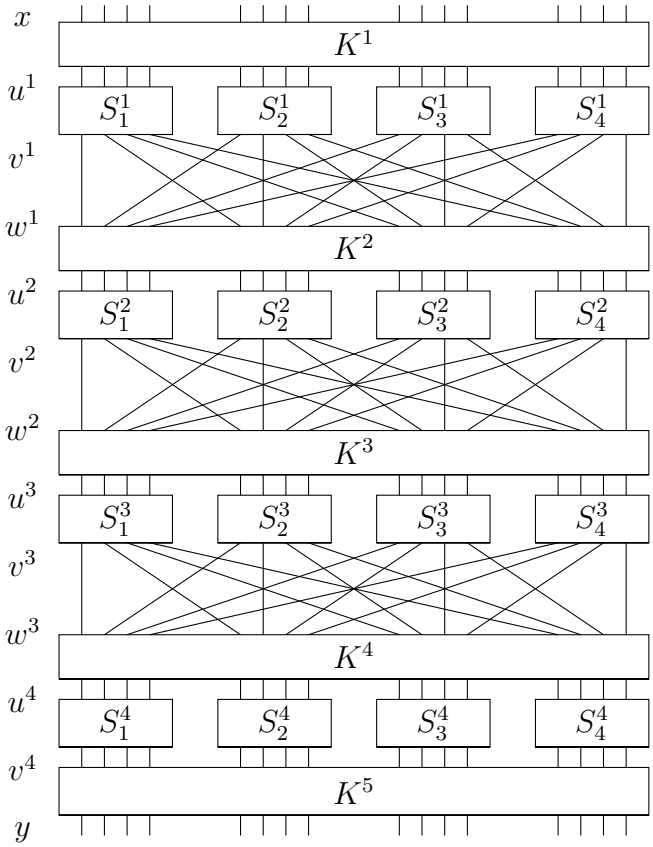



Abbildung 4.1: Ein Substitutions-Permutations-Netzwerk.

**Beispiel 97.** *Sei $l = m = N = 4$ und sei $k = 32$. Für $f$ wählen wir die Funktion $f(K) = (K^1, \ldots, K^5)$ mit $K^r = K[4(r-1)+1, 4(r-1)+16]$. Weiter seien $\sigma_S : \{0,1\}^4 \to \{0,1\}^4$ und $\pi_P : \{1,\ldots,16\} \to \{1,\ldots,16\}$ die folgenden Permutationen (wobei die Argumente und Werte von $\sigma_S$ hexadezimal dargestellt sind; siehe auch Abbildung 4.1):*

| $z$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\sigma_S(z)$ | E | 4 | D | 1 | 2 | F | B | 8 | 3 | A | 6 | C | 5 | 9 | 0 | 7 |

*und*

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\pi_P(i)$ | 1 | 5 | 9 | 13 | 2 | 6 | 10 | 14 | 3 | 7 | 11 | 15 | 4 | 8 | 12 | 16 |

*Für den Schlüssel $K = 0011\,1010\,1001\,0100\,1101\,0110\,0011\,1111$ liefert $f$ beispielsweise die Rundenschlüssel $f(K) = (K^1, \ldots, K^5)$ mit*

$$\begin{aligned} K^1 &= 0011\,1010\,1001\,0100, \\ K^2 &= 1010\,1001\,0100\,1101, \\ K^3 &= 1001\,0100\,1101\,0110, \\ K^4 &= 0100\,1101\,0110\,0011, \\ K^5 &= 1101\,0110\,0011\,1111, \end{aligned}$$

*unter denen der Klartext $x = 0010\,0110\,1011\,0111$ die folgenden Chiffrierschritte durch-*

*läuft:*

$$\begin{aligned} x &= 0010\,0110\,1011\,0111 = w^0 \\ w^0 \oplus K^1 &= 0001\,1100\,0010\,0011 = u^1 \\ S(u^1) &= 0100\,0101\,1101\,0001 = v^1 \\ P(v^1) &= 0010\,1110\,0000\,0111 = w^1 \\ &\vdots \\ P(v^3) &= 1110\,0100\,0110\,1110 = w^3 \\ w^3 \oplus K^4 &= 1010\,1001\,0000\,1101 = u^4 \\ S(u^4) &= 0110\,1010\,1110\,1001 = v^4 \\ u^4 \oplus K^5 &= 1011\,1100\,1101\,0110 = y. \end{aligned}$$

◁

## 4.3 Lineare Approximationen

Sei $\sigma_S : \{0,1\}^l \to \{0,1\}^{l'}$ die funktionale Beschreibung einer S-Box $S$. Wählen wir die Eingabe $U = U_1 \cdots U_l$ zufällig unter Gleichverteilung, so gilt für die zugehörige Ausgabe $V = \sigma_S(U) = V_1 \cdots V_{l'}$,

$$\Pr[V = v \mid U = u] = \begin{cases} 1 & \sigma_S(u) = v, \\ 0 & \text{sonst} \end{cases}$$

für alle $u \in \{0,1\}^l$ und $v \in \{0,1\}^{l'}$. Wegen $\Pr[U = u] = 2^{-l}$ folgt

$$\Pr[V = v, U = u] = \begin{cases} 2^{-l} & \sigma_S(u) = v, \\ 0 & \text{sonst.} \end{cases}$$

Ist $\sigma_S$ eine lineare Funktion, so lässt sich jedes Ausgabebit $v_j$ von $S$ über eine Funktion der Form $v_j = u_{i_1} \oplus \cdots \oplus u_{i_k}$ für geeignete Indizes $1 \le i_1 < \cdots < i_k \le l$ berechnen. In diesem Fall würde also

$$\Pr[V_j = U_{i_1} \oplus \cdots \oplus U_{i_k}] = 1$$

gelten. Die Idee hinter der linearen Kryptoanalyse ist nun, Gleichungen der Form

$$V_{j_1} \oplus \cdots \oplus V_{j_{k'}} = U_{i_1} \oplus \cdots \oplus U_{i_k} \oplus c$$

mit $1 \le i_1 < \cdots < i_k \le l$, $1 \le j_1 < \cdots < j_{k'} \le l'$ und $c \in \{0,1\}$ zu finden, die mit möglichst großer Wahrscheinlichkeit gelten. Definieren wir für $a \in \{0,1\}^l$ und $b \in \{0,1\}^{l'}$ die Zufallsvariablen

$$U_a = \bigoplus_{i=1}^{l} a_i U_i \quad \text{und} \quad V_b = \bigoplus_{i=1}^{l'} b_i V_i,$$

so sind wir also an solchen Werten für $a, b$ und $c$ interessiert, für die das Ereignis $V_b = U_a \oplus c$ (oder gleichbedeutend: $U_a \oplus V_b = c$) mit großer Wahrscheinlichkeit eintritt. In diesem Fall lässt sich nämlich der Wert von $V_b$ bei Kenntnis von $U_a$ entsprechend gut vorhersagen. Wegen $\Pr[U_a \oplus V_b = c] = 1 - \Pr[U_a \oplus V_b = c \oplus 1]$ kommt es nur darauf an, wie stark die Wahrscheinlichkeit $\Pr[U_a \oplus V_b = 0]$ von $1/2$ abweicht. Die durch das Paar $(a, b)$ beschriebene **lineare Approximation** $U_a \oplus V_b$ an die S-Box $S$ ist also um so besser, je größer der Absolutbetrag $|\Pr[U_a \oplus V_b = 0] - 1/2|$ ist.

**Definition 98.** *Für eine Zufallsvariable $X$ mit Wertebereich $W(X) = \{0, 1\}$ bezeichne $\varepsilon(X)$ den Wert $\varepsilon(X) = \Pr[X = 0] - 1/2$ (auch **Bias** von $X$ genannt).*

Unter Benutzung dieser Notation lässt sich also die Güte einer linearen Approximation $U_a \oplus V_b$ an eine S-Box $S$ durch den Absolutbetrag $|\varepsilon(U_a \oplus V_b)|$ ihres Bias-Wertes bemessen.

**Beispiel 99.** *Wir betrachten wieder die S-Box $S$ aus Beispiel 97. Dann nimmt die Zufallsvariable $(U_1, \ldots, U_4, V_1, \ldots, V_4)$ die 16 Werte in folgender Tabelle jeweils mit Wahrscheinlichkeit $2^{-4} = 1/16$ an.*

| $U_1$ | $U_2$ | $U_3$ | $U_4$ | $V_1$ | $V_2$ | $V_3$ | $V_4$ | $U_3 \oplus U_4 \oplus V_1 \oplus V_4$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

*Um nun $\varepsilon(U_a \oplus V_b)$ zu berechnen, genügt es, die Anzahl $L(a, b)$ der Zeilen zu bestimmen, für die $U_a = V_b$ ist. Dann gilt $\Pr[U_a \oplus V_b = 0] = \Pr[U_a = V_b] = L(a, b)/16$ und somit*

$$\varepsilon(U_a \oplus V_b) = L(a, b)/16 - 1/2 = (L(a, b) - 8)/16.$$

*Für $a = 0011$ und $b = 1001$ gibt es z.B. $L(a, b) = 2$ Zeilen (Zeile 5 und Zeile 10) mit $U_a = U_3 \oplus U_4 = V_b = V_1 \oplus V_4$, d.h. $\varepsilon(U_3 \oplus U_4 \oplus V_1 \oplus V_4) = (L(a, b) - 8)/16 = -3/8$. Die folgende Tabelle zeigt für alle Werte von $a$ und $b$ (hexadezimal dargestellt) die Anzahlen $L(a, b)$.*

| | $b$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
| 0 | 16 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 1 | 8 | 8 | 6 | 6 | 8 | 8 | 6 | 14 | 10 | 10 | 8 | 8 | 10 | 10 | 8 | 8 |
| 2 | 8 | 8 | 6 | 6 | 8 | 8 | 6 | 6 | 8 | 8 | 10 | 10 | 8 | 8 | 2 | 10 |
| 3 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 10 | **2** | 6 | 6 | 10 | 10 | 6 | 6 |
| 4 | 8 | 10 | 8 | 6 | 6 | 4 | 6 | 8 | 8 | 6 | 8 | 10 | 10 | 4 | 10 | 8 |
| | | | | | | | | ⋮ | | | | | | | | |
| $B$ | 8 | 12 | 8 | 4 | 12 | 8 | 12 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| | | | | | | | | ⋮ | | | | | | | | |
| $F$ | 8 | 6 | 4 | 6 | 6 | 8 | 10 | 8 | 8 | 6 | 12 | 6 | 6 | 8 | 10 | 8 |

◁