



DR. DR. DAVID MEINTRUP



Institut für Mathematik und Datenverarbeitung  
Fakultät für Elektrotechnik und Informationstechnik  
85577 Neubiberg  
Tel.: 089/6004 3785  
Fax: 089/6004 2615  
david.meintrup@unibw-muenchen.de

---

# **Ringvorlesung**

## **Teil 1: Information, Codierung, Verschlüsselung**

Herbst 2003

---



# Vorwort

Das vorliegende Skript begleitet den ersten Teil der dreiteiligen Ringvorlesung Informationsverarbeitung. Ziel ist es, einen ersten Einblick in die Themen Informationstheorie, Codierungstheorie und Kryptographie zu geben. Alle drei Gebiete spielen senderseitig in der digitalen Nachrichtenübertragung eine wichtige Rolle.

Nach einem kurzen Überblick über Codes in der Nachrichtentechnik behandeln wir aus der Informationstheorie die mathematische Modellierung von Information, Quellen und Kanäle, Entropie und die Schannonschen Sätze. Im Anschluss daran legen wir einen Schwerpunkt bei der Behandlung linearer Blockcodes. Anschließend wenden wir uns der Kryptographie zu und erläutern die Public-Key-Kryptographie und das Zero-Knowledge-Verfahren.

Wir möchten uns bei allen bedanken, die einen Beitrag zu diesem Skriptum geleistet haben. Insbesondere sei Herrn C. Hillermeier für die Überlassung eines Skripts zur Informationstheorie [Hi2001] und Herrn S. Schäffler sowie Frau A. Sutor für das Skript zur Kryptographie [ScSu02] gedankt.

# Inhaltsverzeichnis

|          |                                                                     |           |
|----------|---------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Codes in der Nachrichtentechnik</b>                              | <b>1</b>  |
| 1.1      | Nachrichtenübertragungssysteme . . . . .                            | 1         |
| 1.2      | Das Grundprinzip der Blockcodierung . . . . .                       | 4         |
| 1.3      | Hammingdistanz und Minimaldistanz . . . . .                         | 6         |
| 1.4      | Der Grundgedanke der Kanalcodierung . . . . .                       | 8         |
| <b>2</b> | <b>Informationstheorie</b>                                          | <b>10</b> |
| 2.1      | Motivation der probabilistischen Beschreibung von Nachrichten . . . | 10        |
| 2.2      | Einschub zur Wahrscheinlichkeitstheorie . . . . .                   | 11        |
| 2.3      | Mathematische Modellierung von Information . . . . .                | 18        |
| 2.3.1    | Diskrete Quellen und Kanäle . . . . .                               | 18        |
| 2.3.2    | Der Informationsgehalt eines Ereignisses . . . . .                  | 19        |
| 2.3.3    | Die Entropie . . . . .                                              | 21        |
| 2.3.4    | Eigenschaften der Entropie . . . . .                                | 22        |
| 2.4      | Quellcodierung . . . . .                                            | 26        |
| 2.4.1    | Problemstellung . . . . .                                           | 26        |
| 2.4.2    | Codes mit Präfixeigenschaft . . . . .                               | 27        |
| 2.4.3    | Das Shannon'sche Codierungstheorem . . . . .                        | 32        |
| 2.5      | Das Shannon'sche Kanalcodierungstheorem . . . . .                   | 34        |
| <b>3</b> | <b>Einschub: "Codierung im Alltag"</b>                              | <b>37</b> |
| <b>4</b> | <b>Lineare Blockcodes</b>                                           | <b>40</b> |
| 4.1      | Elementare Eigenschaften . . . . .                                  | 40        |
| 4.2      | Schranken für Codelängen . . . . .                                  | 43        |
| 4.3      | Zyklische Codes . . . . .                                           | 44        |
| 4.4      | BCH-Codes . . . . .                                                 | 47        |
| <b>5</b> | <b>Ziele und historische Entwicklung</b>                            | <b>50</b> |
| 5.1      | Nachricht und Information . . . . .                                 | 50        |
| 5.2      | Transposition und Substitution . . . . .                            | 51        |
| 5.3      | Kryptoanalyse monoalphabetischer Chiffren . . . . .                 | 55        |

|          |                                                                |           |
|----------|----------------------------------------------------------------|-----------|
| 5.4      | Perfekte Sicherheit . . . . .                                  | 57        |
| 5.5      | Kryptosysteme . . . . .                                        | 57        |
| 5.5.1    | Definition Kryptosystem . . . . .                              | 57        |
| 5.6      | Perfekte Sicherheit . . . . .                                  | 57        |
| 5.6.1    | Definition perfekt sicherer Kryptosysteme . . . . .            | 59        |
| 5.6.2    | Kriterium für perfekte Sicherheit von Claude Shannon . . . . . | 59        |
| <b>6</b> | <b>Mathematische Grundlagen</b>                                | <b>60</b> |
| 6.1      | Gruppen, erzeugende Elemente . . . . .                         | 60        |
| 6.2      | Rechnen in Restklassengruppen . . . . .                        | 62        |
| 6.3      | Der größte gemeinsame Teiler . . . . .                         | 63        |
| 6.4      | Primzahlen . . . . .                                           | 64        |
| 6.5      | Körper . . . . .                                               | 65        |
| <b>7</b> | <b>Public-Key-Kryptographie</b>                                | <b>65</b> |
| 7.1      | Trapdoor-Einwegfunktionen . . . . .                            | 65        |
| 7.2      | Das RSA-Verfahren . . . . .                                    | 66        |
| 7.3      | Der diskrete Logarithmus . . . . .                             | 67        |
| 7.4      | Das Verschlüsselungsverfahren von El Gamal . . . . .           | 68        |
| 7.5      | Elliptische Kurven . . . . .                                   | 68        |
| <b>8</b> | <b>Zero-Knowledge-Verfahren</b>                                | <b>71</b> |
| 8.1      | Authentikationssysteme . . . . .                               | 71        |
| 8.2      | Der Fiat-Shamir-Algorithmus . . . . .                          | 72        |
|          | <b>Literaturverzeichnis</b>                                    | <b>74</b> |

# 1 Codes in der Nachrichtentechnik

## 1.1 Nachrichtenübertragungssysteme

Die Grundaufgabe der digitalen Nachrichtentechnik besteht darin, Information von einem Sender, Quelle genannt, zu einem Empfänger, Senke genannt, zu übertragen. Quelle und Senke können dabei viele verschiedene konkrete Ausprägungen haben. So kann es sich beispielsweise um Kommunikation zwischen zwei Mobiltelefonen handeln, um das Verschicken von Datenpaketen über Kabelverbindungen oder um eine Satellitenfunkstrecke. Je nach Anwendung sind die Anforderungen an die Übertragung sehr unterschiedlich. So kommt es beim Telefonieren in erster Linie auf eine gute Sprachqualität an, die von kleinen Fehlern bei der Übertragung unter Umständen nicht wesentlich beeinträchtigt wird. Bei der Datenübertragung ist hingegen die Fehlertoleranz in der Regel geringer. Für eine Satellitenfunkstrecke ist typischerweise die im All zur Verfügung stehende Energie eine knappe Ressource, so dass die Kommunikation mit einer möglichst geringen Energie sichergestellt werden muss. Für diese unterschiedlichen Anforderungen stehen zwischen Quelle und Senke mehrere Komponenten zur Verfügung, die je nach Anwendung variieren können. Wir haben einen typischen Verlauf in Abbildung 1 skizziert. Im Folgenden werden wir die einzelnen Komponenten kurz erläutern. Auf einige Bausteine, die für den weiteren Verlauf entscheidend sind, gehen wir in den späteren Abschnitten ausführlicher ein.

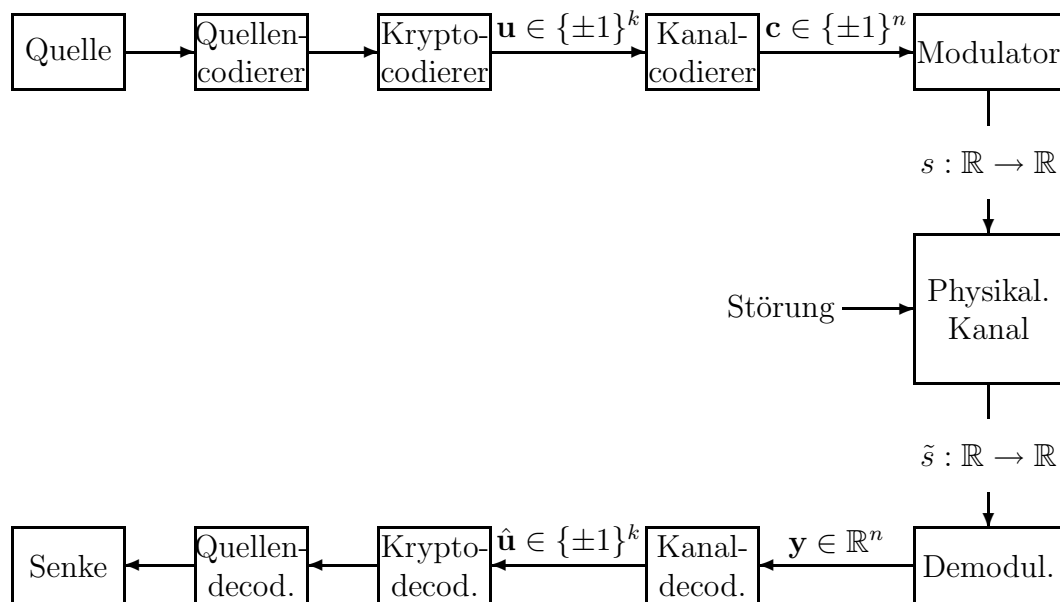


Abbildung 1: Digitale Nachrichtenübertragung

**Quelle:** Wie bereits erwähnt, bezeichnet man den Ursprung der Nachrichten als Quelle. Dabei kann es sich im Mobilfunk um einen Sprecher bzw. Sprache handeln, genauso aber um einen Computer o.Ä. Die genaue Form der Quelle ist für die weiteren Betrachtungen nicht relevant.

**Quellencodierer:** Die Quellencodierung ist die erste von drei Codierungsarten, die in der Shannon'schen Theorie unterschieden werden. Ihre Aufgabe ist es, die Nachrichten der Quelle so in digitale Wertefolgen zu transformieren, dass dabei einerseits keine Information verloren geht, andererseits Redundanz beseitigt wird. Der Vorteil der Quellencodierung besteht darin, dass ohne Informationsverlust durch die Eliminierung von Redundanz Zeit und Energie bei der Übertragung gespart werden. Zur Realisierung dieser Datenreduktion können Datenkomprimierungsalgorithmen verwendet werden.

**Kryptocodierer:** Die Aufgabe des Kryptocodierers besteht darin, die Nachricht zu verschlüsseln und damit für Unbefugte nicht lesbar und nicht verfälschbar zu machen. Die Notwendigkeit einer Verschlüsselung ist nicht immer gegeben, daher ist diese Komponente optional. Ziel des Kryptocodierers ist es, auch bei ungestörter Übertragung die Nachricht so zu verändern, dass ein Empfänger ohne Kenntnis des Verschlüsselungsprinzips keine Chance erhält, an die Information zu gelangen. Die dazu benötigten theoretischen Grundlagen werden in der Kryptographie untersucht.

**Kanalcodierer:** Im Kanalcodierer wird der Ausgabe des Kryptocodierers gezielt und kontrolliert Redundanz hinzugefügt. Dies ist notwendig, da im physikalischen Kanal, also bei der eigentlichen Übertragung, Störungen auftreten können, welche die Nachricht verfälschen. Wir bezeichnen die Ausgabe des Kryptocodierers mit  $\mathbf{u} \in \{\pm 1\}^k$ ,  $k \in \mathbb{N}$ . Dabei setzen wir für jedes  $\mathbf{u}$  voraus, dass mit gleicher Wahrscheinlichkeit jede einzelne Komponente  $u_i, i = 1, \dots, k$ , den Wert  $+1$  oder  $-1$  annimmt. Diese Voraussetzung ist gerechtfertigt, da sie sich aus informationstheoretischer Sicht als Ziel der ersten beiden Codierungen ergibt. Durch Hinzufügen kontrollierter Redundanz senderseitig kann auf der Empfängerseite im günstigsten Fall ein Fehler erkannt und korrigiert werden. Der Kanalcodierer bildet demnach ein Wort  $\mathbf{u} \in \{\pm 1\}^k$  auf ein Codewort  $\mathbf{c} \in \{\pm 1\}^n$ ,  $n \geq k$ , ab. Durch Kanalcodierung kann man daher hohe Übertragungszuverlässigkeit erreichen. Auf der anderen Seite kostet jedes hinzugefügte Redundanzbit Energie, so dass immer eine Abwägung zwischen Übertragungsqualität und Energiebedarf stattfinden muss.

**Modulator:** Über den physikalischen Kanal können keine diskreten Werte, etwa Bits, übertragen werden, sondern nur zeitkontinuierliche Signale. Die Zuordnung der diskreten Codeworte  $\mathbf{c} \in \{\pm 1\}^n$  auf kontinuierliche Signale  $s : \mathbb{R} \rightarrow \mathbb{R}$  ist Aufgabe des Modulators. Dabei sind die Nebenbedingungen des physikalischen Kanals, z.B. sein Spektrum, zu beachten. Es ist naheliegend, dass der Modulator je nach gewählter Modulationsart einen entscheidenden Einfluss auf die Übertragungsqualität haben kann. Daher bilden Kanalcodierer und Modulator in manchen Szenarien eine Einheit, deren Komponenten nicht mehr scharf voneinander zu trennen sind.

**Physikalischer Kanal:** Das physikalische Medium, das der eigentlichen Übertragung dient, wird als physikalischer Kanal bezeichnet. Es kann sich dabei um drahtgebundene Medien (z.B. Koaxialkabel, Glasfaserkabel), oder Funkkanäle (z.B. Mobilfunk, Rundfunk) oder auch Speichermedien (z.B. Magnetmedien, elektronische oder optische Speicher) handeln oder um eine beliebige Kombination dieser Kanäle. Charakteristisch für einen physikalischen Kanal ist, dass in ihm Störungen auftreten, d.h. er ist nicht ideal. Bei der Übertragung des Signals  $s : \mathbb{R} \rightarrow \mathbb{R}$  wird man daher am Ausgang des physikalischen Kanals ein verfälschtes Signal  $\tilde{s} : \mathbb{R} \rightarrow \mathbb{R}$  vorfinden. Die zufällige Störung in geeigneter Weise zu beschreiben, ist im Allgemeinen, z.B. im Mobilfunk, eine schwierige Aufgabe der stochastischen Modellierung.

**Demodulator:** Der Demodulator ist das Gegenstück zum Modulator. Er wandelt das verfälschte Signal  $\tilde{s} : \mathbb{R} \rightarrow \mathbb{R}$  wieder in einen diskreten Vektor  $\mathbf{y} \in \mathbb{R}^n$  um. Dabei gilt im Allgemeinen nicht  $\mathbf{y} \in \{\pm 1\}^n$ , d.h. die Ausgabe des Demodulators ist nicht notwendigerweise ein Codewort  $\tilde{\mathbf{c}} \in \{\pm 1\}^n$ . Dieser Unterschied erweist sich als sehr wertvoll, da die Absolutbeträge der Komponenten von  $\mathbf{y}$  als Zuverlässigkeitsinformation für das entsprechende Vorzeichen Verwendung finden. Ist im Demodulator noch ein Entscheider eingebaut, also eine Abbildung von  $\mathbf{y} \in \mathbb{R}^n$  auf  $\tilde{\mathbf{c}} \in \{\pm 1\}^n$ , so bedeutet dies zwar eine wesentliche Vereinfachung der nachfolgenden Schritte, die aber mit einem erheblichen Informationsverlust einhergeht, der sich wiederum negativ auf die Übertragungsqualität auswirkt.

**Kanaldecodierer:** Der Kanaldecodierer hat die Aufgabe, aus der Ausgabe des Demodulators die Information  $\mathbf{u} \in \{\pm 1\}^k$  zu rekonstruieren. Dabei unterscheidet man grundsätzlich zwei Arten der Decodierung. Liegt als Grundlage ein Vektor  $\tilde{\mathbf{c}} \in \{\pm 1\}^n$ , so spricht man von Hard-Decision-Decodierung. Hat man hingegen zusätzlich Zuverlässigkeitsinformation in Form eines Vektors  $\mathbf{y} \in \mathbb{R}^n$  zur Verfügung, so kann man auch das Ergebnis der Decodierung  $\tilde{\mathbf{u}} \in \{\pm 1\}^k$  mit einem Zuverlässigkeitsmaß versehen. In diesem Fall spricht man von Soft-Decision-Decodierung. Der Vorteil der Hard-Decision-Decodierung ist ihr geringerer Aufwand, den man in der Regel mit einem Verlust an Übertragungsqualität bezahlen muss.

**Kryptodecodierer:** Die Entschlüsselung der Ausgabe der Decodierung erfolgt im Kryptodecodierer. Er rekonstruiert also die quellencodierte Nachricht.

**Quellendecodierer:** Der Quellendecodierer verarbeitet die ankommende Information so, dass die Senke sie verstehen kann. Im Mobilfunk z.B. erzeugt der Quellencodierer im Allgemeinen Sprache.

**Senke:** Die Senke ist der gewünschte Empfänger der Nachricht.

In der digitalen Nachrichtenübertragung ist die chronologisch erste Aufgabe die Quellencodierung. Diese ist Teil der Informationstheorie, mit der wir uns im zweiten Abschnitt beschäftigen. Um präzise Aussagen treffen zu können, benötigen wir dazu insbesondere eine mathematische Beschreibung von Quellen und Kanälen, die wiederum einiger Vorbereitung aus dem Bereich der Wahrscheinlichkeitstheorie bedarf. Da der Hauptgegenstand unserer Beschäftigung jedoch die Kanalcodierung ist, be-

ginnen wir mit einigen Grundlagen der Blockcodierung, die man problemlos ohne genauere Kenntniss der Quellcodierung einführen kann.

## 1.2 Das Grundprinzip der Blockcodierung

Physikalische Nachrichtenkanäle sind gestört. Diese Störung wird üblicherweise Rauschen genannt. Diese Bezeichnung erklärt sich von selbst, wenn man einmal versucht, alte Schallplatten zu hören oder Radiosender zu empfangen. Das Rauschen stört die Übertragung der Information und verfälscht diese. Um dennoch die vollständige Information empfängerseitig zu erhalten, wird der Information gezielt Redundanz hinzugefügt. Diesen Prozess nennt man Kanalcodierung. Die verwendeten Codes kann man grundsätzlich in zwei Klassen unterteilen, Faltungscodes und Block-Codes. Wir werden uns in diesem Abschnitt mit Block-Codes beschäftigen und verweisen für die Faltungscodes auf die Literatur ([Fr96], [Pr95]). Bei Block-Codes wird aus einem Informationswort  $\mathbf{u} \in \{\pm 1\}^k$  durch Codierung ein Codewort  $\mathbf{c} \in \{\pm 1\}^n$ . Als Minimalforderung an eine solche Zuordnung wird man verlangen, dass Codewörter, die vor der Kanalcodierung verschieden waren, es auch nach der Codierung bleiben, es ist ja sogar das Ziel, sie so unterschiedlich wie möglich zu machen. Mit anderen Worten: Jeder Kanalcodierer muss injektiv sein, und daher bezeichnet man ganz allgemein für zwei Mengen  $X$  und  $Y$  sowie zwei natürliche Zahlen  $n, k$  mit  $n \geq k$  jede

$$\text{injektive Abbildung } E : X^k \hookrightarrow Y^n \text{ als } (n, k)\text{-Block-Code.} \quad (1)$$

Für unsere Zwecke reichen viel speziellere Code-Klassen, die wir nun einführen wollen. Insbesondere beschränken wir uns auf binäre Codes, deren Zeichenvorrat also aus einem  $n$ -dimensionalen Vektorraum  $\mathbb{F}_2^n$ ,  $n \in \mathbb{N}$ , über dem Körper  $\mathbb{F}_2$  stammt. Der Körper  $\mathbb{F}_2$  ist der bis auf Isomorphie eindeutige Körper mit genau zwei Elementen. In der Nachrichtentechnik wird  $\mathbb{F}_2$  in der Regel mit  $\{\pm 1\}$  identifiziert, so wie wir es bisher auch gemacht haben. Dies hat den Vorteil, dass den unterschiedlichen Bits einfacher eine Energie zugeteilt werden kann. Dabei ist die Körperstruktur des zwei-elementigen Körpers  $\{\pm 1\}$  mit Addition  $\oplus$  und Multiplikation  $\odot$  wie folgt definiert:

$$\begin{array}{rcl} -1 \oplus -1 & = & 1 \\ 1 \oplus 1 & = & 1 \\ 1 \oplus -1 & = & -1 \\ -1 \odot -1 & = & -1 \\ 1 \odot -1 & = & 1 \\ 1 \odot 1 & = & 1 \end{array} .$$

Da diese Verknüpfungen etwas gewöhnungsbedürftig sind, werden wir im Folgenden möglichst einfach das Symbol  $\mathbb{F}_2$  verwenden. Für explizite Beschreibungen werden wir uns von nun an auf die (in der mathematischen Codierungstheorie) übliche Konvention  $\mathbb{F}_2 = \mathbb{Z}/(2) = \{0, 1\}$  beziehen.

Für zwei natürliche Zahlen  $n, k$  mit  $n \geq k$  heißt jede

injektive Abbildung  $E : \mathbb{F}_2^k \hookrightarrow \mathbb{F}_2^n$  binärer  $(n, k)$ -Block-Code.

$$\mathbf{u} \mapsto \mathbf{c} := E(\mathbf{u})$$

Den Definitionsbereich eines Codes nennen wir Informationsraum, die Worte  $\mathbf{u} \in \mathbb{F}_2^k$  Infobits. Das Bild  $\mathcal{C} := E(\mathbb{F}_2^k)$  wird als Coderaum oder einfach als Code  $\mathcal{C}$  bezeichnet. Denn oft kommt es nicht auf die Entstehung des Codes an, sondern lediglich auf die Menge der Codeworte  $\mathcal{C}$ . Dabei heißt ein binärer  $(n, k)$ -Block-Code

- *systematisch*, falls die ersten  $k$  Komponenten von  $\mathbf{c}$  den Vektor  $\mathbf{u}$  bilden,
- *linear*, falls  $\mathcal{C}$  ein  $(k)$ -dimensionaler linearer Unterraum von  $\mathbb{F}_2^n$  ist.

Wir werden im Folgenden immer systematische Codes betrachten und lassen daher in der Regel das adjektiv "systematisch" weg.

Wir wollen uns nun einige Beispiele systematischer, linearer  $(n, k)$ -Block-Codes anschauen. Die ersten beiden Beispiele stellen zwei Extremfälle da, die aber als Testfälle oft sehr nützlich sind.

**Beispiel 1.1 (uncodierte Übertragung).** Rein formal können wir  $n = k$  und als Code die Identität wählen:

$$E_{\text{uncod}} = \text{id} : \mathbb{F}_2^k \hookrightarrow \mathbb{F}_2^k \quad (2)$$

$$\mathbf{u} \mapsto \text{id}(\mathbf{u}) = \mathbf{u}. \quad (3)$$

In diesem Fall spricht man von uncodierter Übertragung. Es handelt sich formal um einen binären, linearen  $(k, k)$ -Block-Code.

**Beispiel 1.2 (Wiederholungscode).** In diesem Fall sei  $k = 1$  und  $n > 1$ . Die Abbildungsvorschrift ist gegeben durch

$$E_R : \mathbb{F}_2 \hookrightarrow \mathbb{F}_2^n \quad (4)$$

$$\mathbf{u} \mapsto (\mathbf{u}, \dots, \mathbf{u}). \quad (5)$$

Das Bit  $\mathbf{u}$  wird also einfach  $n$ -mal wiederholt. In diesem Fall spricht man von einem  $n$ -fachen Wiederholungscode. Es handelt sich um einen binären linearen  $(n, 1)$ -Block-Code.

**Beispiel 1.3 (Der  $(7, 4)$ -Hamming Code).** Der Hamming-Code läßt sich am einfachsten durch folgende Matrix angeben:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (6)$$

Der Code ist in diesem Fall gegeben durch die lineare Abbildung

$$E_{Ham} : \mathbb{F}_2^4 \hookrightarrow \mathbb{F}_2^7 \quad (7)$$

$$\mathbf{u} \mapsto \mathbf{c} = G^T \mathbf{u}. \quad (8)$$

Der Hamming-Code ist ein systematischer linearer  $(7, 4)$ -Code.

Folgende Eigenschaften haben alle Block-Codes gemeinsam:

- Block-Codes sind "eindeutig" und "umkehrbar", indem zwei verschiedenen Infowörtern zwei verschiedene Codewörter zugeordnet werden. Dies ist nichts anderes als eine Beschreibung der Injektivität der Abbildung  $E$ .
- Block-Codes sind "zeitinvariant", die Zuordnungsvorschrift  $E$  hängt nicht von der Zeit ab. Die Infowörter brauchen daher nicht nummeriert zu werden, da es auf die Reihenfolge nicht ankommt. Das gleiche Infowort wird immer gleich codiert.
- Block-Codes sind "gedächtnislos", jedes Codewort wird genau durch ein Infowort bestimmt und umgekehrt. Diese auf den ersten Blick vielleicht selbstverständlich scheinende Eigenschaft haben die so genannten Faltungscodes nicht.

Das Verhältnis von Blocklänge  $n$  zur Zahl der Infobits  $k$  wird als

$$\text{Coderate } R := \frac{k}{n} \quad (9)$$

bezeichnet. Die Coderate  $R \leq 1$  gibt an, wie viele Infobits pro Kanalbenutzung im Durchschnitt über den Kanal geschickt werden. Grundsätzlich ist man immer an möglichst hohen Datenraten und damit möglichst großen Coderaten interessiert. Dieses Interesse steht allerdings im Wettstreit mit einer möglichst guten Übertragungsqualität, die durch viele Redundanzbits und damit eine kleine Übertragungsrate erzielt wird.

### 1.3 Hammingdistanz und Minimaldistanz

Je stärker sich Codewörter unterscheiden, desto unwahrscheinlicher ist eine Fehlentscheidung bei der Decodierung. Daher ist die Unterschiedlichkeit von Codewörtern ein Gütekriterium für einen Code. Um diese Größe messen zu können, führen wir die folgenden Begriffe ein:

**Definition 1.4 (Hamming-Distanz, Hamming-Gewicht).** Sei  $X$  eine endliche Menge. Der Hamming-Abstand auf  $X$  ist definiert durch:

$$d : X^n \times X^n \rightarrow \mathbb{R}, \quad (10)$$

$$(x, y) \mapsto d(x, y) := |\{i : x_i \neq y_i\}|. \quad (11)$$

Der Begriff Abstand ist aufgrund des folgenden Satzes gerechtfertigt:

**Proposition 1.5 (Hamming-Abstand ist Metrik).** *Der Hamming-Abstand ist eine Metrik auf  $X^n$ , d.h. es gelten die folgenden Eigenschaften:*

- (i) *Symmetrie:*  $d(x, y) = d(y, x)$  für alle  $x, y \in X^n$ .
- (ii)  $d(x, y) = 0 \Rightarrow x = y$ .
- (iii) *Dreiecksungleichung:*  $d(x, z) \leq d(x, y) + d(y, z)$  für alle  $x, y, z \in X^n$ .

Entscheidend ist der kleinste auftretende Abstand zwischen zwei Codeworten, daher bekommt dieser einen eigenen Namen:

**Definition 1.6 (Minimaldistanz).** Die Minimaldistanz  $d_{min}$  eines Codes  $\mathcal{C}$  ist definiert als die minimale Hamming-Distanz zwischen allen paarweise verschiedenen Codeworten:

$$d_{min} := \min\{d(x, y) : x, y \in \mathcal{C}, x \neq y\} \quad (12)$$

Die Minimaldistanz ist der wichtigste Parameter, der die Güte eines Codes bestimmt. Um dies zu betonen, wird es gelegentlich mit in die Notation aufgenommen, man spricht von einem

$$(n, k, d_{min}) - \text{Block-Code.} \quad (13)$$

Zur Bestimmung der Minimaldistanz müssen, zumindest vorerst, paarweise alle Abstände betrachtet werden. Je größer die Minimaldistanz ist, je stärker sich also Codewörter voneinander unterscheiden, desto besser ist der Code. Bei gegebener Coderate  $R = \frac{k}{n}$  und Blocklänge  $n$  wird man sich für den Code entscheiden, der zu großem  $d_{min}$  führt. Im Allgemeinen wird  $d_{min}$  größer und damit der Code besser, wenn die Coderate kleiner wird, wofür man aber wiederum Einbußen in der Übertragungsgeschwindigkeit oder eine größere Bandbreite in Kauf nehmen muss.

**Beispiel 1.7 (uncodierte Übertragung).** Bei der uncodierten Übertragung

$$E_{uncod} = \text{id} : \mathbb{F}_2^k \hookrightarrow \mathbb{F}_2^k \quad (14)$$

$$\mathbf{u} \mapsto \text{id}(\mathbf{u}) = \mathbf{u}. \quad (15)$$

erhalten wir den schlechtest möglichen Minimalabstand  $d_{min} = 1$  und die best mögliche Coderate  $R = 1$ . Die Übertragung ist somit sehr effizient, aber sehr unsicher.

**Beispiel 1.8 (Wiederholungscode).** Beim Wiederholungscode ist die Situation genau umgekehrt:

$$E_R : \mathbb{F}_2 \hookrightarrow \mathbb{F}_2^n \quad (16)$$

$$\mathbf{u} \mapsto (\mathbf{u}, \dots, \mathbf{u}). \quad (17)$$

Es handelt sich um einen linearen  $(n, 1)$ -Block-Code mit dem best möglichen Minimalabstand  $d_{\min} = n$  und der schlechtesten Coderate  $R = \frac{1}{n}$ . Die Übertragung ist also ineffizient, aber sehr sicher.

**Beispiel 1.9 (Der  $(7, 4)$ -Hamming Code).** Der  $(7, 4)$ -Hamming-Code läßt sich explizit durch die Angabe aller 16 Codeworte beschreiben:

$$\{ 0000000, 0001111, 0010110, 0011001, 0100101, 0101010, 0110011, 0111100, \\ 1000011, 1001100, 1010101, 1011010, 1100110, 1101001, 1110000, 1111111 \}$$

Daher kann man explizit die Hammingabstände bestimmen und stellt fest, dass der  $(7, 4)$ -Hamming-Code eine Minimaldistanz von  $d_{\min} = 3$  hat, bei einer Coderate von  $R = \frac{4}{7}$ .

Das letzte Beispiel verdeutlicht, dass es unbedingt erforderlich ist, bessere Methoden zur Bestimmung der Minimaldistanz zu entwickeln. Dazu werden wir später eine algebraische Struktur auf der Codemenge einführen.

## 1.4 Der Grundgedanke der Kanalcodierung

Wir wollen in diesem Abschnitt zur Abrundung des ersten Teils nochmals ein paar grundlegende Überlegungen zusammenfassen. Der Grundgedanke der Kanalcodierung wird insbesondere bei Block-Codes deutlich. Im Prinzip besteht eine Analogie zur Digitalisierung, mit ganz ähnlichen Vor- und Nachteilen:

**Digitalisierung** bedeutet Quantisierung im Wertebereich. Ist  $\Delta$  die Quantisierungsbreite, so ist der maximale Fehler  $\Delta/2$ . Daraus ergeben sich als Vor- und Nachteile:

**Vorteil:** Kleine Fehler (kleiner als  $\Delta/2$ ) werden eliminiert, bei analoger Übertragung wird hingegen in jeder Verstärkerstufe das Signal/Rausch-Verhältnis immer schlechter.

**Nachteil:** Auch bei einem theoretisch ungestörten Kanal tritt immer ein mittlerer Übertragungsfehler von  $\sqrt{(\Delta^2/12)}$  auf. Große Übertragungsfehler werden durch Zuordnung zur falschen Quantisierungsstufe noch verstärkt.

**Fazit:** Quantisierung ist nur dort sinnvoll, wo der Hauptanteil der Störung kleiner als  $\Delta/2$  ist.

**Kanalcodierung** bedeutet Quantisierung ganzer Symbolfolgen im Zeitbereich. Die  $n - k$  Redundanzbits werden so gewählt, dass je zwei Codeworte sich an mindestens  $d_{\min}$  Stellen unterscheiden. Das ist deswegen möglich, weil  $\mathcal{C} \subset Y^n$  ein echter Teilraum, im binären Fall ein  $k$ -dimensionaler Unterraum des  $\mathbb{F}_2^n$  ist. Es ergibt sich, wie wir im Einzelnen noch sehen werden:

**Vorteil:** Bei weniger als  $d_{min}/2$  liegt das Empfangswort noch am nächsten am gesendeten Wort und kann daher vollständig richtig decodiert werden.

**Nachteil:** Bei mehr als  $d_{min}/2$  Fehler wird u.U. auf ein falsches Codewort entschieden, und die Zahl der Fehler wird möglicherweise erhöht.

**Fazit:** Kanalcodierung lohnt sich nur dann, wenn in der Mehrzahl weniger als  $d_{min}/2$  Fehler pro Codewort auftreten. Dies ist insbesondere bei relativ guten Kanälen der Fall, an die eine hohe Anforderung bzgl. der Zuverlässigkeit besteht.

Die zentrale Idee der Kanalcodierung ist es, lange Informationsblöcke in noch längere Codeblöcke abzubilden. Das Ausmaß der hinzugefügten Redundanz ist im einzelnen von der Kanalqualität und der gewünschten Übertragungsqualität abhängig. Ziel dabei ist immer die Erreichung eines *Codiergewinns*. Damit ist im Einzelnen Folgendes gemeint: Ziel der Codierung muss es sein, die Fehlerwahrscheinlichkeit bei der Decodierung zu reduzieren. Diese kann bspw. als Bitfehlerrate BER (bit error rate) gemessen werden, also als relative Häufigkeit falsch decodierter Infobits im Verhältnis zur Gesamtzahl der Infobits. Es werden also nur die Infobits in die BER einbezogen. Die Fehlerrate hängt im Allgemeinen von der Qualität des Kanals ab, die sich wiederum durch das Signal-Rausch-Verhältnis beschreiben läßt. Um Codes untereinander vergleichen zu können, betrachtet man daher einerseits die

$$\text{Energie pro Infobit } E_b \quad (18)$$

und andererseits

$$\text{Rauschleistungsdichte } N_0 \quad (19)$$

Die BER  $P_b$  wird logarithmisch über dem Verhältnis  $E_b/N_0$  in Dezibel aufgetragen, siehe Abbildung 2. Durch die Codierung mit einer Coderate  $R$  verringert sich die zur Verfügung stehende Energie  $E_c$  für jedes einzelne Codebit um den Faktor  $R$ :

$$E_c = R \cdot E_b \quad (20)$$

Dadurch erhöht sich zunächst einmal die Wahrscheinlichkeit für einen Fehler, da weniger Energie pro Bit zur Verfügung steht. Dieser Effekt wird aber ab einer gewissen Energieschwelle durch die Korrektoreigenschaften des Codes ausgeglichen, wie in Abbildung 2 zu sehen ist. Bei hohen Energien tritt dann der Codiergewinn ein, also eine Energiedifferenz für eine fest gewählte Fehlerrate zwischen codierter und uncodierter Übertragung. Der Codiergewinn wird mit  $E_b/N_0 \rightarrow \infty$  nicht beliebig groß, sondern konvergiert gegen einen Grenzwert.

Bevor wir uns weitere konkrete Beispiele von Codes anschauen, wollen wir den Fragen nachgehen, was ein guter Code ist und ob es gute Codes überhaupt gibt. Zur Beantwortung dieser Fragen entwickeln wir im nächsten Abschnitt die Informationstheorie.

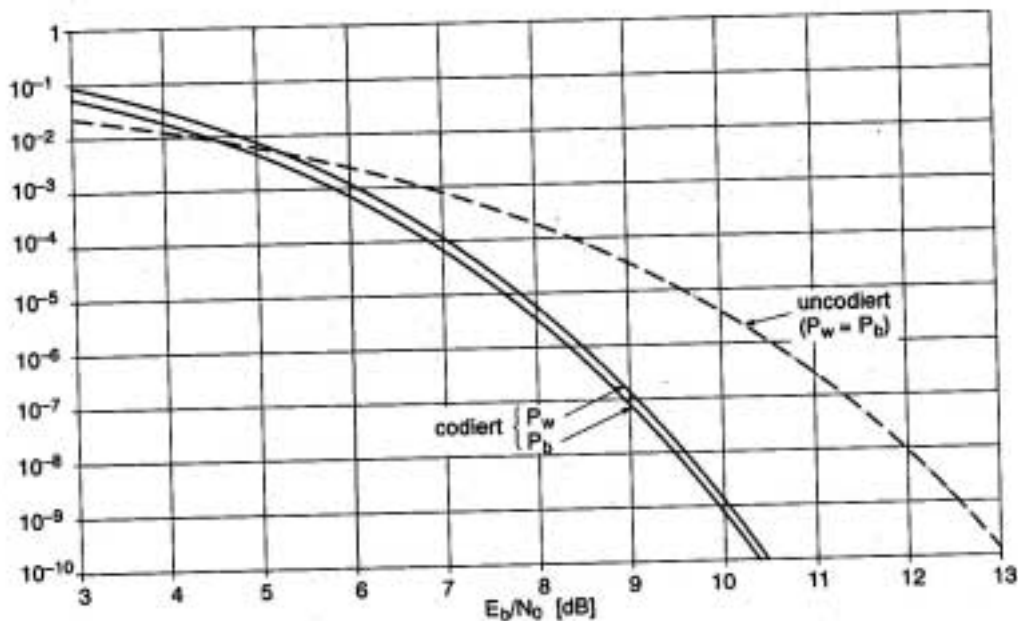


Abbildung 2: Fehlerkurven zum Codiergewinn

## 2 Informationstheorie

Die Informationstheorie, manchmal auch als statistische Theorie der Kommunikation bezeichnet, stellt sich die Aufgabe, Nachrichtenquellen und Übertragungssysteme mathematisch zu modellieren und zu analysieren. Ziel ist die Beantwortung folgender Fragen:

- Wie läßt sich der Informationsgehalt einer Nachricht messen?
- Wie kann ein technisches Nachrichtenübertragungssystem optimiert werden, so dass möglichst viel Informationsgehalt pro Zeiteinheit übertragen wird?
- Wo liegen die theoretisch erreichbaren oberen Grenzen des übertragbaren Informationsgehaltes?

### 2.1 Motivation der probabilistischen Beschreibung von Nachrichten

Die Informationstheorie modelliert sowohl die (Nachrichten-)Quelle als auch den (Übertragungs-)Kanal mit Methoden der Wahrscheinlichkeitstheorie (siehe auch die Alternativbezeichnung 'statistische Theorie der Kommunikation'). Während aber die

unvorhersagbaren Störungen beim *Kanal* eine probabilistische Modellierung geradezu herausfordern, ist ein solches Modell für die *Quelle* auf den ersten Blick weniger überzeugend und soll daher kurz motiviert werden.

Die Nachrichtenquelle, von der wir ausgehen, reiht Zeichen bzw. Symbole, die einem endlichen Zeichenvorrat (= Symbolalphabet) entstammen, zu einer Zeichenfolge, der Nachricht, aneinander. Im Allgemeinen erzeugt die Nachrichtenquelle eine solche Zeichenfolge (z.B. 'Ich habe Hunger') gezielt, um eine Bedeutung zu transportieren. Die Informationstheorie, 1948 durch Claude Shannon begründet, kümmert sich jedoch nicht um den Bedeutungsgehalt von Nachrichten. Der Bedeutungsgehalt einer Nachricht ist nämlich **subjektiv** und vom Kontext der Kommunikationspartner abhängig und daher technisch nicht fassbar. Eine Nachricht 'Feuer!' trägt z.B. für die Kommunikationspartner 'Brandmelder und Feuerwehr' eine andere Bedeutung als für die Kommunikationspartner 'Offizier und Soldat'.

In der Informationstheorie wird der Prozess der Entstehung einer Nachricht als wiederholte Durchführung eines **Zufallsexperiments** modelliert. Ähnlich wie bei der Lottoziehung eine Loskugel zufällig aus der Lostrommel ausgewählt wird, wird in dieser Sichtweise das einzelne von der Nachrichtenquelle generierte Zeichen zufallssteuert aus dem Zeichenvorrat ausgewählt. Ein generiertes Einzelzeichen ist somit das beobachtete Resultat des besagten Zufallsexperiments, die Nachricht (= Zeichenfolge) das Resultat der fortgesetzten Wiederholung dieses Zufallsexperiments.

Offensichtlich gibt dieses probabilistische Modell nicht die volle Bandbreite des Phänomens 'Nachricht' wieder. Dennoch wird es bei der Konstruktion und Analyse von Systemen, die digitale Nachrichten bzw. Daten speichern und übertragen, erfolgreich eingesetzt. Weshalb?

Der Konstrukteur eines Systems zur digitalen Nachrichtenübertragung kann a priori nicht wissen, welche Nachricht aus dem Ensemble möglicher Nachrichten einmal wirklich übertragen werden wird. Aufgrund dieses fehlenden Wissens muss er die (später einmal) zu übertragende Nachricht als eine Zufallsauswahl aus der Gesamtheit aller möglichen Nachrichten ansehen und das System so konstruieren, dass es alle potenziellen Nachrichten übertragen kann. Um universelle Verwendbarkeit des Übertragungssystems für alle Arten potenzieller (digitaler) Nachrichten zu erzielen, muss der Systemkonstrukteur also zwangsläufig von einem probabilistischen Modell der Nachrichtenerzeugung (und -quelle) ausgehen.

## 2.2 Einschub zur Wahrscheinlichkeitstheorie

Wahrscheinlichkeitstheorie ist die mathematische Modellierung und Analyse von Zufallsexperimenten, d.h. von Experimenten mit nicht vorhersagbarem Ausgang. Für die Informationstheorie sind die folgenden Begriffe aus der Wahrscheinlichkeitstheorie besonders wichtig:

- (1) Der **Basisraum**  $\Omega$  ist die Menge aller möglichen Ergebnisse  $\omega$  des zu modellierenden Zufallsexperiments.

Beachte:  $\omega$  steht i.A. nicht nur für eine einzige Beobachtungsgröße, wie z.B. die Augenzahl beim Würfeln. In der abstrakten Größe  $\omega$  ist vielmehr häufig die gesamte Information über das Ergebnis des Zufallsexperiments, im Würfelbeispiel also Muskelzuckungen und Handhaltung während des Wurfs, Luftbewegungen, Rollbewegung etc., zusammengefasst.

- (2) Meist sind wir gar nicht am einzelnen Ergebnis  $\omega$  des Zufallsexperiments interessiert, da jeweils viele Ergebnisse dieselben Folgen nach sich ziehen (z.B.: zu derselben Augenzahl beim Würfeln führen). Eine Teilmenge von  $\Omega$ , die in diesem Sinne (d.h. durch eine spür- oder beobachtbare Konsequenz aus dem Ergebnis des Zufallsexperiments) ausgezeichnet ist, heißt **Ereignis**. Die Menge aller Ereignisse nennen wir  $\sigma$ -**Algebra**  $S$ . Dabei handelt es sich um ein System von (Teil-)Mengen aus  $\Omega$  – also um eine Teilmenge der Potenzmenge über  $\Omega$  – mit speziellen Eigenschaften.
- (3) Das **Wahrscheinlichkeitsmaß**  $P$  ordnet jedem Ereignis (also jeder Teilmenge aus  $\Omega$ , die  $S$  angehört) eine nichtnegative reelle Zahl zu. Die Abbildung  $P : S \rightarrow \mathbb{R}$  hat zum einen die Eigenschaften eines **Maßes**, die motiviert sind durch den Wunsch nach einer Inhalts- bzw. Volumenmessung:

a)  $P(A) \geq 0 \quad \forall A \in S$ .

b)  $P(\emptyset) = 0$ .

c)  $\sigma$ -Additivität:

Für eine Folge  $\{A_i\}_{i \in \mathbb{N}}$  paarweise disjunkter Mengen  $A_i$  gilt:

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

(vgl. Volumenmessung: Die Volumina nicht überlappender Mengen sollen sich aufsummieren.)

Über diese Maßeigenschaften hinaus gilt für ein *Wahrscheinlichkeitsmaß*  $P$ :

d)  $P(\Omega) = 1$ ,

d.h. die Wahrscheinlichkeit des Gesamtraums ist auf 1 normiert.

Bemerkung: Dass wir  $P(\Omega)$  gerade auf die Zahl 1 normieren, ist reine Konvention. Die wichtige Anforderung an  $P$  ist die **Normierbarkeit** von  $P(\Omega)$  auf eine endliche Zahl, eine Anforderung, die beispielsweise das Lebesguemaß  $\lambda$  im  $\mathbb{R}^n$  nicht erfüllt.

- (4) Das Tripel  $(\Omega, S, P)$  wird als **Wahrscheinlichkeitsraum** bezeichnet.

- (5) Während bei vielen Zufallsexperimenten das Ergebnis  $\omega$  selbst der Beobachtung gar nicht zugänglich ist, äußert sich die (Zufalls-)Wahl von  $\omega$  doch typischerweise in Form einer beobachtbaren Größe. Das mathematische Modell einer solchen von  $\omega$  abhängigen Beobachtungsgröße ist eine **Zufallsvariable**  $X$ . Eine reellwertige Beobachtungsgröße (z.B. beim Scheibenschießen der Abstand zwischen einem Treffer und dem Mittelpunkt der Schützenscheibe) wird durch eine reellwertige Zufallsvariable modelliert. Darunter versteht man eine Funktion  $X : \Omega \rightarrow \mathbb{R}$  mit der Eigenschaft,  $S$ - $\mathcal{B}$ -messbar zu sein. Letzteres heißt: Die Urbildmenge einer jeden Menge aus  $\mathcal{B}$  (der Borelschen  $\sigma$ -Algebra auf  $\mathbb{R}$ , die alle offenen und alle abgeschlossenen Teilmengen von  $\mathbb{R}$  enthält) muss eine Teilmenge von  $\Omega$  sein, die in  $S$  liegt. Kann die Beobachtungsgröße nur diskrete Werte aus einer endlichen Menge  $\mathcal{A} := \{a_1, \dots, a_n\}$  annehmen (z.B. beim Würfelspiel die Augenzahl, d.h.  $\mathcal{A} = \{1, \dots, 6\}$ ), so modellieren wir dies durch eine diskrete Zufallsvariable  $X : \Omega \rightarrow \mathcal{A}$ . Da die 'natürliche'  $\sigma$ -Algebra einer endlichen Menge identisch ist mit der Potenzmenge, bedeutet die Forderung der  $S$ - $\sigma(\mathcal{A})$ -Messbarkeit, dass gelten muss:

$$X^{-1}(a_i) \in S \quad \forall i = 1, \dots, n.$$

Dass Zufallsvariable messbar (bezüglich der jeweiligen  $\sigma$ -Algebren) sein müssen, scheint auf den ersten Blick eine eher technische Forderung zu sein, hat aber wichtige Konsequenzen:

- (6) Als direkte Konsequenz ihrer Messbarkeit überträgt eine Zufallsvariable auf natürliche Weise das Wahrscheinlichkeitsmaß  $P$  des Wahrscheinlichkeitsraums  $(\Omega, S, P)$  auf den Bildraum der Zufallsvariable  $X$ , also auf  $\mathbb{R}$  (falls  $X$  reellwertig) oder  $\mathcal{A}$  (falls  $X$  diskret).

Das durch eine reellwertige Zufallsvariable  $X : \Omega \rightarrow \mathbb{R}$  erzeugte **Bildmaß**  $P_X$  ist definiert als

$$P_X(\underbrace{A}_{A \in \mathcal{B}}) := P(\underbrace{X^{-1}(A)}_{\text{wegen Messbarkeit aus } S}) .$$

Analog stattet die diskrete Zufallsvariable  $X : \Omega \rightarrow \mathcal{A}$  die Menge  $\mathcal{A}$  mit dem Bildmaß  $P_X$  aus, das gegeben ist durch

$$P_X(a_i) := P(X^{-1}(a_i)) .$$

In beiden Fällen wird das Bildmaß  $P_X$  auch **Verteilung** von  $X$  genannt.

Bemerkung:

In der ingenieurwissenschaftlichen Literatur ist es gängig, zur Modellierung eines Zufallsexperiments nur den Bildraum der Zufallsvariablen, also beispielsweise  $\mathbb{R}$  oder  $\mathcal{A}$  und die darauf gegebene Verteilung  $P_X$  zu betrachten. In einer solchen 'verkürzten' Beschreibung taucht der 'zugrundeliegende' Wahrscheinlichkeitsraum  $(\Omega, S, P)$  gar nicht (mehr) auf.

Im Falle einer diskreten Beobachtungsgröße mit Werten aus  $\mathcal{A} := \{a_1, \dots, a_n\}$  wird dann also direkt der endliche Wahrscheinlichkeitsraum  $(\mathcal{A}, P_X)$  betrachtet.

Als zugehörige  $\sigma$ -Algebra wird dabei stets die Potenzmenge über  $\mathcal{A}$  verstanden. Aufgrund der Maßeigenschaft von  $P_X$  ist das Wahrscheinlichkeitsmaß auf  $\mathcal{A}$  durch die Angabe von  $P_X(a_i)$ ,  $i = 1, \dots, n$ , bereits eindeutig bestimmt. Kurzschreibweise:  $P_X(a_i) =: p(a_i) =: p_i \quad \forall i = 1, \dots, n$ .

- (7) Um den Begriff des **Erwartungswerts** einer Zufallsvariablen zu motivieren, denken wir uns ein Glücksspiel mit einer diskreten Beobachtungsgröße, die  $n$  jeweils reellwertige feste Werte annehmen kann:  $\mathcal{A} := \{x_1, \dots, x_n\}$  mit  $x_i \in \mathbb{R}$ . Beim Auftreten des Ereignisses  $x_i$  erhält der Spieler eine Zahlung von  $x_i$  DM. Bezeichnet  $p_i$  die Wahrscheinlichkeit für das Auftreten von  $x_i$  (s.o.), so wird man die wahrscheinlichkeitsgewichtete Summe der möglichen Gewinne  $x_i$ , also die Größe  $\sum_{i=1}^n x_i p_i$ , als erwarteten Gewinn oder Erwartungswert dieses Glücksspiels ansehen.

In Verallgemeinerung dieser Situation ist für eine diskrete, reellwertige Zufallsvariable  $X : \Omega \rightarrow \mathcal{A} = \{x_1, \dots, x_n\}$  mit  $x_i \in \mathbb{R}$  der Erwartungswert  $E(X)$  definiert als

$$E(X) := \sum_{i=1}^n x_i P(X = x_i) = \sum_{i=1}^n x_i p_i .$$

Um für kontinuierliche Zufallsvariable  $X : \Omega \rightarrow \mathbb{R}$  eine entsprechende Größe (als Antwort auf die Frage, welcher Wert von  $X$  zu erwarten ist) zu bilden, muss die Summation über die möglichen Ereignisse durch eine Integration ersetzt werden. Diese Integration kann wahlweise im ursprünglichen Wahrscheinlichkeitsraum  $(\Omega, \mathcal{S}, P)$  erfolgen – integriert wird dann bezüglich des Wahrscheinlichkeitsmaßes  $P$  –, d.h.

$$E(X) := \int_{\Omega} X dP ,$$

oder im Bildraum  $\mathbb{R}$  mit Hilfe des Bildmaßes  $P_X$ ,

$$E(X) := \int_{\mathbb{R}} Id_{\mathbb{R}} dP_X ,$$

wobei  $Id_{\mathbb{R}}$  die Identitätsfunktion auf  $\mathbb{R}$  bezeichnet. Beide Definitionen, die zu demselben Resultat führen, setzen voraus, dass die Zufallsvariable  $X$  ( $P$ )-integrierbar ist.

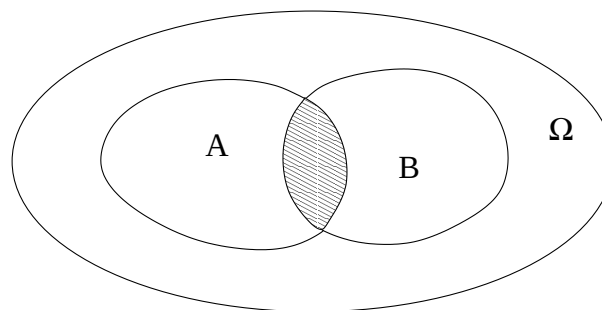
Es sei angemerkt, dass die – in den obigen Definitionen vorausgesetzte – Einführung eines Integrationsbegriffs für kontinuierliche reellwertige Zufallsvariable wesentlich auf den Maßeigenschaften von  $P$  und auf der  $\mathcal{S}$ - $\mathcal{B}$ -Messbarkeit von  $X$  fußt.

- (8) Die bislang vorgestellten Begriffsbildungen entstammen alle der Welt der Maßtheorie, ist doch ein Wahrscheinlichkeitsraum  $(\Omega, \mathcal{S}, P)$  nichts anderes als ein normierter Maßraum. Käme keine weitere spezifische Begriffsbildung hinzu,

wäre Stochastik also ein Teilgebiet der Maßtheorie. Was die Stochastik zu einem eigenständigen Zweig der Mathematik macht, ist der Begriff der stochastischen Unabhängigkeit (bzw. stochastischen Abhängigkeit) von Ereignissen und Zufallsvariablen.

Hierzu brauchen wir zunächst den eng damit zusammenhängenden Begriff der **bedingten Wahrscheinlichkeit**. Dieser soll die – in Anwendungen häufige – Situation modellieren, dass die Wahrscheinlichkeit für ein Ereignis  $A$  anzugeben ist unter der Nebenbedingung, dass ein bestimmtes Ereignis  $B$  (sicher) eintritt.

Beispiel: Wie groß ist die Wahrscheinlichkeit, dass eine Glühbirne noch 100 Stunden brennen wird, wenn sie bereits 24 Stunden gebrannt hat?



Um eine derartige Situation zu modellieren, setzen wir das Wahrscheinlichkeitsmaß der Ergebnisse  $\omega$ , die in  $B$  und *gleichzeitig* in  $A$  liegen, ins Verhältnis zum Wahrscheinlichkeitsmaß *aller* Ergebnisse aus  $B$ , und gelangen zu folgender Definition:

Gegeben sei ein Wahrscheinlichkeitsraum  $(\Omega, S, P)$ . Für  $A, B \in S$  mit  $P(B) > 0$  wird die bedingte Wahrscheinlichkeit von  $A$  unter der Bedingung  $B$ , geschrieben  $P(A|B)$ , definiert durch

$$P(A|B) := \frac{P(A \cap B)}{P(B)} .$$

Für festgehaltenes  $B \in S$  (mit  $P(B) > 0$ ) ist  $P^B(A) := P(A|B)$  ein Wahrscheinlichkeitsmaß auf  $(\Omega, S)$  mit der Eigenschaft  $P^B(\Omega \setminus B) = 0$  (in Übereinstimmung damit, dass wir ja (das Eintreffen des Ereignisses)  $B$  voraussetzen, uns also für Ergebnisse aus  $\Omega \setminus B$  nicht interessieren).

Formel von der totalen Wahrscheinlichkeit:

Betrachte eine Partition  $\{B_i \subseteq \Omega \mid i \in \mathbb{N}\}$  von  $\Omega$  mit  $B_i \in S$  und  $P(B_i) > 0$ .

$$\implies \boxed{P(A) = \sum_{i=1}^{\infty} P(A|B_i) P(B_i)} \quad \forall A \in S.$$

Beweis:

$$\begin{aligned}
 & \sum_{i=1}^{\infty} P(A|B_i) P(B_i) \stackrel{(\text{Def.})}{=} \sum_{i=1}^{\infty} P(A \cap B_i) \stackrel{(\sigma\text{-Additivitat von } P)}{=} \\
 & = P\left(\bigcup_{i=1}^{\infty} (A \cap B_i)\right) \stackrel{(\text{Mengenlehre})}{=} P\left(A \cap \underbrace{\bigcup_{i=1}^{\infty} B_i}_{\Omega}\right) = P(A).
 \end{aligned}$$

q.e.d.

Formel von Bayes:

Gilt zusätzlich  $P(A) > 0$ , so gilt:

$$P(B_i | A) = \frac{P(A | B_i) P(B_i)}{\sum_{i=1}^{\infty} P(A | B_i) P(B_i)}$$

Beweis:

$$\begin{aligned} P(B_i | A) &\stackrel{(\text{Def.})}{=} \frac{P(B_i \cap A)}{P(A)} \quad [B_i \cap A \stackrel{=}{=} A \cap B_i] \\ &= \frac{P(A | B_i) \cdot P(B_i)}{P(A)} = \frac{P(A | B_i) P(B_i)}{\sum_{i=1}^{\infty} P(A | B_i) P(B_i)} \\ &\quad \uparrow \\ &\quad \text{Formel von der totalen Wahrscheinlichkeit} \end{aligned}$$

q.e.d.

- (9) Wir sehen zwei Ereignisse  $A$  und  $B$  als **stochastisch unabhängig** an, wenn die Wahrscheinlichkeit für  $A$  unberührt davon bleibt, ob wir Kenntnis über das Eintreffen von  $B$  erlangt haben oder nicht, wenn also gilt:  $P^B(A) = P(A)$ , was wiederum äquivalent ist zu

$$P(A \cap B) = P(A) \cdot P(B) \quad (*)$$

Diese Definition läßt sich sofort von zwei auf  $n$  Ereignisse  $A_1, \dots, A_n$  verallgemeinern: Die Ereignisse  $A_1, \dots, A_n \in S$  heißen stochastisch unabhängig, wenn für jede Auswahl von  $k$  Ereignissen (aus  $A_1, \dots, A_n$ ) gilt:

$$P\left(\bigcap_{j=1}^k A_{i_j}\right) = \prod_{j=1}^k P(A_{i_j})$$

[zur Indizierung:  $j \in \{1, \dots, k\}$  zählt die ausgewählten Ereignisse durch,  $i_j$  ist der ursprüngliche Index des an  $j$ -ter Stelle ausgewählten Ereignisses.]

Um stochastische Unabhängigkeit nicht nur für Ereignisse, also Mengen, sondern auch für Zufallsvariablen definieren zu können, betrachten wir ein Mengensystem, das für eine Zufallsvariable  $X$  charakteristisch ist, nämlich die durch  $X$  erzeugte  $\sigma$ -Algebra, geschrieben  $\sigma(X)$ . Darunter versteht man die kleinste  $\sigma$ -Algebra auf  $\Omega$ , für die  $X$  messbar ist.  $\sigma(X)$  enthält alle Urbilder potenzieller Beobachtungsergebnisse und ist diejenige Struktur – in Form eines Mengensystems –, die dem Basisraum  $\Omega$  durch die Zufallsvariable  $X$  aufgeprägt wird.

Nachdem wir uns diese Brücke zwischen Mengensystemen und Zufallsvariablen

gebaut haben, können wir definieren:

Zwei Zufallsvariablen  $X$  und  $Y$  heißen stochastisch unabhängig, wenn die von ihnen erzeugten  $\sigma$ -Algebren  $\sigma(X)$  und  $\sigma(Y)$  stochastisch unabhängig sind.

## 2.3 Mathematische Modellierung von Information

### 2.3.1 Diskrete Quellen und Kanäle

Nachdem wir die Grundlagen der Wahrscheinlichkeitstheorie bereit gestellt haben, können wir nun die Klasse von Quellen und Kanälen definieren, die uns im Folgenden begleiten werden. Beginnen wir mit den Quellen.

**Definition 2.1 (diskrete Quelle).** Gegeben sei eine diskrete Zufallsvariable  $X : \Omega \longrightarrow \mathcal{A} = \{x_1, \dots, x_n\}$  auf einem Wahrscheinlichkeitsraum  $(\Omega, \mathcal{S}, P)$ . Der Vektor  $p$  enthalte die Elementarwahrscheinlichkeiten des Bildmaßes:

$$p_i := P_X(x_i), \quad 1 \leq i \leq n. \quad (21)$$

Dann ist eine diskrete endliche Quelle  $Q$  ohne Gedächtnis mit dem Alphabet  $\mathcal{A}$  gegeben durch das Tupel  $Q = (\mathcal{A}, p)$ .

Natürlich ist  $Q$  nichts Anderes als ein diskreter Wahrscheinlichkeitsraum, wenn man die Potenzmenge  $\mathcal{P}(\mathcal{A})$  als  $\sigma$ -Algebra hinzunimmt und bedenkt, dass  $p$  ein Wahrscheinlichkeitsmaß auf  $\mathcal{P}(\mathcal{A})$  induziert. Durch die Schreibweise  $Q = (\mathcal{A}, p)$  beschränken wir uns auf die wesentlichen Elemente unserer Vorstellung:  $Q$  sendet zu diskreten Zeitpunkten  $t = 0, 1, \dots$  Signale aus, die dem Alphabet  $\mathcal{A}$  entnommen sind. Die Wahrscheinlichkeit für das Auftreten von  $x_i$  zum Zeitpunkt  $t$  ist gleich  $p_i$ , unabhängig von  $t$  und unabhängig davon, was zu den Zeitpunkten  $0, \dots, t-1$  gesendet wurde. Die Quelle hat kein Gedächtnis. Nach Einführung der Quellen kommen wir nun zu den Kanälen:

**Definition 2.2 (DMC).** Ein diskreter, gedächtnisloser Kanal (DMC) mit Eingangsalphabet  $\mathcal{A}$  und Ausgangsalphabet  $\mathcal{B}$  ist ein Tripel  $(\mathcal{A}, \mathcal{B}, R)$ , bei dem  $\mathcal{A}$  und  $\mathcal{B}$  endliche Mengen sind und  $R$  eine Familie  $(r_a)_{a \in \mathcal{A}}$  von Wahrscheinlichkeitsverteilungen auf  $\mathcal{B}$  ist.

Die Zahl  $r_a(b)$  gibt an, mit welcher Wahrscheinlichkeit  $b$  empfangen wird, wenn  $a$  geschickt wurde. Diese Wahrscheinlichkeit hängt nicht von der "Vorgeschichte" ab, daher nennt man auch den Kanal gedächtnislos.

**Beispiel 2.3 (BSC).** Ein wichtiges Beispiel ist der binäre, symmetrische Kanal BSC( $\alpha$ ) mit Irrtumswahrscheinlichkeit  $\alpha$ . Für ihn ist  $\mathcal{A} = \mathcal{B} = \{0, 1\}$  und

$$r_a(b) = \begin{cases} 1 - \alpha, & \text{falls } a = b, \\ \alpha, & \text{falls } a \neq b. \end{cases} \quad (22)$$

Quellen  $Q = (\mathcal{A}, p)$  und Kanäle  $(\mathcal{A}, \mathcal{B}, r)$  kann man folgendermaßen kombinieren: Wir setzen

$$r(a, b) := p(a) \cdot r_a(b), \quad a \in A, b \in B. \quad (23)$$

Dann ist  $(\mathcal{A} \times \mathcal{B}, r)$  ein zusammengesetzter Wahrscheinlichkeitsraum, wobei wir der gleichen Konvention wie bei der Definition einer Quelle folgen. Insgesamt können wir also vier diskrete Verteilungen betrachten, zwei Randverteilungen und zwei bedingte Verteilungen. Wir wollen diese nochmals aufführen und interpretieren:

- (i) Randverteilung zu  $\mathcal{A}$ :  $(\mathcal{A}, r|\mathcal{A}) = (\mathcal{A}, p)$ : Die zu  $\mathcal{A}$  gehörende Randverteilung ist die Verteilung der Quelle, die ja die Sendewahrscheinlichkeit der einzelnen Symbole angibt.
- (ii) Randverteilung zu  $\mathcal{B}$ :  $(\mathcal{B}, r|\mathcal{B}) = (\mathcal{B}, q)$ : Die zu  $\mathcal{B}$  gehörende Randverteilung gibt an, mit welcher Wahrscheinlichkeit ein Element von  $\mathcal{B}$  empfangen wird.
- (iii) bedingte Verteilung bezüglich  $\mathcal{A}$ :  $(\mathcal{B}, r(\cdot|a))$ . Es gilt  $r(b|a) = r_a(b)$ . Daher gibt  $r(b|a)$  die Wahrscheinlichkeit an, dass  $b$  empfangen wird, wenn  $a$  gesendet wurde.
- (iv) bedingte Verteilung bezüglich  $\mathcal{B}$ :  $(\mathcal{A}, r(\cdot|b))$ . Die Verteilung  $r(a|b)$  kann als Wahrscheinlichkeit gedeutet werden, dass  $a$  gesendet wurde, wenn  $b$  empfangen wird.

Im nächsten Abschnitt nehmen wir zunächst wieder einen allgemeinen Standpunkt ein und sprechen von diskreten Zufallsvariablen. Man kann sich aber immer eine diskrete Quelle beziehungsweise einen diskreten Kanal vorstellen.

### 2.3.2 Der Informationsgehalt eines Ereignisses

Gegeben sei eine diskrete Zufallsvariable  $X : \Omega \rightarrow \mathcal{A}$ , wobei  $\mathcal{A} := \{x_1, \dots, x_n\}$  eine endliche Menge sei (z.B. ein endliches Alphabet von Symbolen), mit einer Verteilung  $P_X$ .

Schreibweise:  $P_X(x_i) =: p(x_i) =: p_i \quad \forall i = 1, \dots, n$ .

Angenommen, wir haben soeben das Ereignis  $x_i$  beobachtet. Welche Eigenschaften sollte eine Größe besitzen, die – im Einklang mit unserem gewohnten Verständnis des Begriffs ‘Information’ – den **Informationsgehalt** dieser Beobachtung beschreibt?

Die durch die Beobachtung gewonnene ‘Information’ ist um so größer, je größer die Überraschung während der Beobachtung war. Da das Eintreten eines unwahrscheinlichen Ereignisses für mehr Überraschung sorgt als das Eintreten eines wahrscheinlichen Ereignisses, sollte der ‘Informationsgehalt’ eines Ereignisses eine monotone Funktion von  $\frac{1}{p_i}$  sein.

In der Tat werden wir zu einer Größe  $I(p)$  mit dieser Eigenschaft geführt, wenn wir folgende unmittelbar einleuchtende Forderungen stellen:

- (a)  $I(p) \geq 0$ , d.h. unser Maß für den Informationsgehalt soll eine nichtnegative reelle Größe sein.
- (b) Die Information, die wir durch die gleichzeitige Beobachtung zweier stochastisch unabhängiger Ereignisse gewinnen, soll gleich der Summe der Informationsgewinne aufgrund der beiden Einzelereignisse sein. Da andererseits die Wahrscheinlichkeit für das gleichzeitige Auftreten zweier stochastisch unabhängiger Ereignisse (mit den Einzelwahrscheinlichkeiten  $p_1$  und  $p_2$ )  $p_1 \cdot p_2$  beträgt, impliziert diese Forderung:

$$I(p_1 \cdot p_2) = I(p_1) + I(p_2)$$

- (c)  $I(p)$  soll eine stetige Funktion der Ereigniswahrscheinlichkeit  $p$  sein.

Wie im Folgenden skizziert, legen diese Forderungen die Funktion  $I(p)$  bis auf eine Proportionalitätskonstante eindeutig fest.

Aus Forderung (b) folgt:

$$I(p^2) = I(p \cdot p) = I(p) + I(p) = 2I(p)$$

Per vollständiger Induktion läßt sich dies verallgemeinern zu

$$\text{Setze nun} \quad \underbrace{I(p^n)}_{p^n = y}, \quad \underbrace{I(p)}_{p = y^{\frac{1}{n}}} \quad \forall p \in (0, 1) \quad (*)$$

$$\implies \quad I(y) = n \cdot I(y^{\frac{1}{n}}) \quad \forall y \in (0, 1) \quad (**)$$

$$\text{Laut } (*) \text{ gilt} \quad I\left(y^{\frac{1}{n}}\right) = \frac{I\left(\left(y^{\frac{1}{n}}\right)^m\right)}{m} = \frac{I\left(y^{\frac{m}{n}}\right)}{m}.$$

$$\text{Einsetzen in } (**) \text{ ergibt} \quad I(y) = \frac{n}{m} I\left(y^{\frac{m}{n}}\right)$$

$$\iff I\left(y^{\frac{m}{n}}\right) = \frac{m}{n} I(y) \quad \forall y \in (0, 1), \quad m, n \in \mathbb{N}$$

Hält man nun  $y \in (0, 1)$  fest, so ist damit  $I(p)$  für alle  $p$  der Form  $p = y^{\frac{m}{n}}$  ( $m, n \in \mathbb{N}$ ) auf den Wert  $I(y)$  zurückgeführt. Da die Werte des (natürlichen) Logarithmus auf genau die gleiche Weise aus dem Wert  $\log_e(y)$  [bei festgehaltenem  $y$ ] hervorgehen, ist für alle Argumente  $p$  der Form  $p = y^{\frac{m}{n}}$  gezeigt:

$$I(p) = \text{const} \cdot \log_e(p)$$

Aufgrund der Stetigkeitsforderung (c) überträgt sich diese Aussage auf alle Argumente  $p \in [0, 1]$ .

Forderung (a) legt noch das Vorzeichen der Konstanten fest:  $\text{const} < 0$ .

Den Betrag des Proportionalitätsfaktors *const* und die Basis des Logarithmus können wir frei wählen. Die gängige Konvention lautet: *const* = −1, Basis des Logarithmus = 2, d.h.

$$I(p) = -\log_2(p) = \log_2\left(\frac{1}{p}\right).$$

Die (aus der obigen Konvention) resultierende Einheit, in der wir Information(sgewinn) messen, wird **bit** genannt.

### 2.3.3 Die Entropie

Kehren wir zurück zur diskreten Zufallsvariablen  $X : \Omega \rightarrow \mathcal{A}$  und fragen nach der im statistischen Mittel zu erwartenden Information, die wir gewinnen, wenn wir das Zufallsexperiment einmal durchführen und das beobachtete Ereignis (also: welcher der Werte  $x_1, \dots, x_n$  "aus der Lostrommel gezogen" wurde) notieren. Diese Größe wird **Entropie**  $H(X)$  der Zufallsvariable  $X$  genannt und berechnet sich als Erwartungswert des mit den Ereignissen  $\{X = x_i\}$  verbundenen Informationsgewinns:

$$H(X) := \sum_{i=1}^n p(x_i) I(x_i) = - \sum_{i=1}^n p(x_i) \log_2(p(x_i))$$

Bemerkungen:

- (1) Die Begriffe "Unsicherheit", "Überraschung" und "Informationsgewinn" hängen eng miteinander zusammen:  
*Vor* der Durchführung bzw. Beobachtung des Zufallsexperiments herrscht Unsicherheit/Unbestimmtheit (engl. uncertainty).  
*während* der Beobachtung: Überraschung.  
*nach* der Beobachtung: Information(sgewinn).  
 Daher wird die Entropie manchmal als 'Mittelwert der zu erwartenden Information', manchmal als 'mittlere Unsicherheit' oder als 'mittlere Unbestimmtheit' bezeichnet.
- (2) Wir haben die Entropie für allgemeine diskrete Zufallsvariable  $X$  eingeführt. Da sie nicht von den Werten  $x_i$  selbst, sondern nur von den Wahrscheinlichkeiten der Ereignisse  $\{X = x_i\}$  abhängt, ist die Entropie eine Eigenschaft einer (Wahrscheinlichkeits-)Verteilung.  
 Der Begriff Entropie hat wichtige Anwendungen außerhalb der Nachrichtentechnik, beispielsweise in der statistischen Theorie der Wärme.
- (3) Wir erinnern daran, dass im Kontext der Nachrichtentechnik die diskrete Zufallsvariable  $X : \Omega \rightarrow \mathcal{A}$  als Modell einer Nachrichtenquelle  $Q = (\mathcal{A}, p)$  dient.  $\mathcal{A} := \{x_1, \dots, x_n\}$  ist der Zeichenvorrat (= das Symbolalphabet) der Quelle, die  $x_1, \dots, x_n$  sind die Elementarzeichen, die für die Bildung einer Nachricht

(=Symbolsequenz) zur Verfügung stehen.

$p(x_1), \dots, p(x_n)$  sind die Auftretswahrscheinlichkeiten der Elementarzeichen, die z.B. durch Messung der Buchstabenhäufigkeiten in langen Texten gewonnen wurden. (Morse, der Erfinder des Morse-Codes, besorgte sich Näherungswerte der Auftretswahrscheinlichkeiten der einzelnen Buchstaben, indem er die Setzkästen der Schriftsetzer auszählte.)

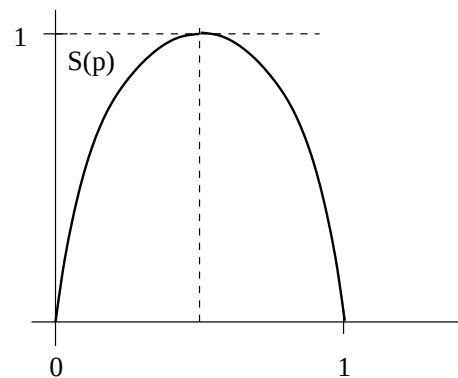
Die Entropie  $H(Q) := H(X)$  ist in diesem Zusammenhang der mittlere Informationsgehalt einer (Nachrichten-)Quelle.

**Beispiel 2.4.** Betrachte eine binäre Quelle  $Q$ , d.h.  $\mathcal{A} = \{x_1, x_2\}$ , mit den Auftretswahrscheinlichkeiten

$$\begin{aligned} p(x_1) &= p \\ p(x_2) &= 1 - p \end{aligned}$$

$$\Rightarrow \underbrace{H(Q)}_{\text{hängt nur von } p \text{ ab.}} = \underbrace{-p \log_2(p) - (1-p) \log_2(1-p)}_{=: S(p) \leftarrow \text{„Shannon-Funktion“}}$$

Maximum von  $S(p)$  bei  $p = \frac{1}{2}$ ,  
Wert beim Maximum: 1 bit/Zeichen

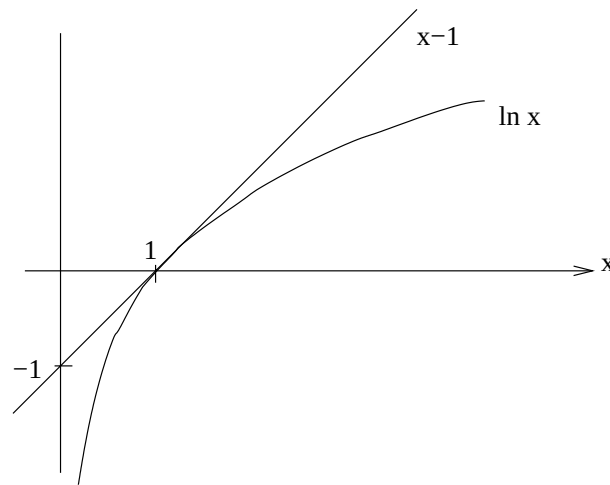


Entropie der binären Quelle

### 2.3.4 Eigenschaften der Entropie

Zunächst stellen wir uns zwei Abschätzungen bereit, die für das Rechnen mit Entropien sehr nützlich sind.

- ( $\alpha$ ) Aus Gründen der Bequemlichkeit betrachten wir im Folgenden die Funktion  $\log_e x \equiv \ln x$ . Wegen  $\log_2(x) = \log_2 e \cdot \ln x$  sind alle Aussagen sofort auf  $\log_2(x)$  übertragbar.



Der Logarithmus ist eine konkave Funktion, d.h. der Graph von  $\ln x$  liegt stets unterhalb seiner Tangente (in einem beliebigen Punkt). Anwendung dieser Beobachtung auf die Tangente im Punkt  $x = 1$ , d.h. auf die Gerade  $y = x - 1$  (denn  $\left. \frac{d \ln x}{dx} \right|_{x=1} = 1$ ), ergibt

$$\ln x \leq x - 1,$$

wobei das Gleichheitszeichen genau für  $x = 1$  gilt.

Für  $ld(x)$  lautet die Ungleichung entsprechend

$$ld(x) \leq (\log_2 e) \cdot (x - 1) = \frac{1}{\log_e 2} \cdot (x - 1).$$

( $\beta$ ) Sind zwei  $n$ -Tupel  $(p_1, \dots, p_n)$  und  $(q_1, \dots, q_n)$  positiver reeller Zahlen mit  $\sum_{i=1}^n q_i \leq 1$  und  $\sum_{i=1}^n p_i = 1$  gegeben – also beispielsweise zwei Wahrscheinlichkeitsverteilungen auf jeweils  $n$  Ereignisse –, so gilt stets folgende Beziehung, auch **Gibbs'sche Ungleichung** genannt:

$$\sum_{i=1}^n p_i \, ld\left(\frac{q_i}{p_i}\right) \leq 0,$$

mit Gleichheit genau dann, wenn  $p_i = q_i \quad \forall i = 1, \dots, n$ .

Beweis:

Aus ( $\alpha$ ) folgt (für alle  $i$ ):

$\ln\left(\frac{q_i}{p_i}\right) \leq \frac{q_i}{p_i} - 1$ , wobei Gleichheit genau im Falle  $p_i = q_i$  gilt. Wir multiplizieren diese Ungleichung mit  $p_i$ , summieren über  $i$  und erhalten

$$\sum_{i=1}^n p_i \ln\left(\frac{q_i}{p_i}\right) \leq \sum_{i=1}^n (q_i - p_i) \leq 1 - 1 = 0,$$

mit Gleichheit genau dann, wenn  $p_i = q_i \quad \forall i = 1, \dots, n$ .

Multiplizieren mit  $\log_2 e$  (d.h. Übergang von  $\ln$  zu  $ld$ ) ergibt die Gibbs'sche Ungleichung. q.e.d.

Beim Entwurf eines Zufallsexperiments möchte man oft den (bei Durchführung des Experiments) zu erwartenden Informationsgewinn maximieren. Wenn möglich, wird man also die Wahrscheinlichkeiten der beobachtbaren Ereignisse so einstellen, dass die Entropie maximal wird.

Die Gibbs'sche Ungleichung verschafft uns eine elegante Antwort auf die Frage, was denn diejenige **Wahrscheinlichkeitsverteilung** (bei  $n$  Ereignissen) mit der **maximalen Entropie** ist.

Zu untersuchen ist  $H(X) = \sum_{i=1}^n p_i ld\left(\frac{1}{p_i}\right)$ , mit  $\sum_{i=1}^n p_i = 1$ .

Wir betrachten zunächst die Größe

$$\begin{aligned} H(X) - ld(n) &= \sum_{i=1}^n p_i ld\left(\frac{1}{p_i}\right) - ld(n) \sum_{i=1}^n p_i \\ &= \sum_{i=1}^n p_i \underbrace{\left[ ld\left(\frac{1}{p_i}\right) + ld\left(\frac{1}{n}\right) \right]}_{ld\left(\frac{1}{p_i}\right)} \end{aligned}$$

Dieser Term erlaubt nun die Anwendung der Gibbs-Ungleichung (mit  $q_i = \frac{1}{n}$ , d.h.  $\sum_{i=1}^n q_i = 1$  ist erfüllt), und wir erhalten

$$H(X) - ld(n) \leq 0 \quad \Longleftrightarrow \quad H(X) \leq ld(n).$$

Um Gleichheit, also den Maximalwert für  $H(X)$ , zu erzielen, muss (laut Gibbs'scher Ungleichung) gelten:

$$p_i = \frac{1}{n} \quad \forall i = 1, \dots, n,$$

also: **gleiche Wahrscheinlichkeit** für alle  $n$  Ereignisse.

Bemerkung:

Alternativ läßt sich dieses Resultat als Lösung des folgenden Optimierungsproblems herleiten:

$$\max_{p \in \mathbb{R}^n} \left[ H(X) \equiv \sum_{i=1}^n p_i ld\left(\frac{1}{p_i}\right) \right]$$

unter der Nebenbedingung  $\sum_{i=1}^n p_i = 1$ .

Jede Lösung dieses Problems erfüllt notwendigerweise die Optimalitäts-Bedingung 1. Ordnung, die da lautet:

Es gibt einen Lagrange-Multiplikator  $\lambda \in \mathbb{R}$ , so dass gilt:

$$(a) \quad \nabla_p \left[ \frac{1}{\log_e(2)} \underbrace{\sum_{i=1}^n p_i \log_e \left( \frac{1}{p_i} \right)}_{=: f(p)} + \lambda \left( \sum_{i=1}^n p_i - 1 \right) \right] = 0$$

$$(b) \quad \sum_{i=1}^n p_i = 1$$

Wegen

$$\begin{aligned} \frac{\partial}{\partial p_i} f(p) &= 1 \cdot \log_e \left( \frac{1}{p_j} \right) + p_j \frac{1}{p_j} \cdot (-1) \cdot \frac{1}{p_j^2} \\ &= \log_e \left( \frac{1}{p_j} \right) - 1 \end{aligned}$$

folgt aus (a):

$$(c) \quad \frac{1}{\log_e(2)} \left[ \log_e \left( \frac{1}{p_j} \right) - 1 \right] + \lambda = 0 \quad \forall j = 1, \dots, n.$$

Da in dieser Gleichung außer  $p_j$  nur Konstanten vorkommen, müssen alle  $p_j$  denselben Wert annehmen. Aufgrund der Normierung (b) gilt daher:  $p_j = \frac{1}{n} \quad \forall j = 1, \dots, n$ .

Wenn man eine Größe näher kennenlernen will, ist es oft nützlich, ihre Extremwerte zu betrachten.

Neben dem Maximalwert der Entropie interessieren wir uns also für diejenige Verteilung (über  $n$  Ereignisse), bei der die Entropie **minimal** wird.

Als Summe nichtnegativer reeller Zahlen kann  $H(X)$  nicht kleiner als 0 werden. Der Wert  $H(X) = 0$  ergibt sich für alle Verteilungen, in denen *ein* Ereignis mit Sicherheit auftritt (keine Überraschung!), d.h. für die gilt:  $p_j = 1$  für ein  $j \in \{1, \dots, n\}$  und  $p_i = 0$  für alle  $i \neq j$ :

$$H(X) = \sum_{i=1}^n p_i \cdot \underbrace{\log_e \left( \frac{1}{p_j} \right)}_{=0 \text{ für } i=j} = 0$$

$\uparrow$   
 $=0 \text{ für } i \neq j$

## 2.4 Quellcodierung

### 2.4.1 Problemstellung

Gegeben sei eine Nachrichtenquelle  $Q = (\mathcal{A}, p)$  in Form einer Zufallsvariablen  $X : \Omega \rightarrow \mathcal{A} = \{x_1, \dots, x_n\}$ . Die (Auftritts-)Wahrscheinlichkeiten der Elementarzeichen  $x_1, \dots, x_n$  seien  $p_1, \dots, p_n$ .

Das durch  $X$  beschriebene Zufallsexperiment (d.h. die Auswahl eines Elementarzeichens) soll immer wieder, (stochastisch) unabhängig voneinander, durchgeführt und beobachtet werden, so dass eine Sequenz aus den Elementarzeichen  $\{x_1, \dots, x_n\}$ , eine sogenannte Nachricht, entsteht.

Den Übertragungskanal wollen wir vorläufig als störungsfrei annehmen. Allerdings ist das Symbolalphabet, das zu übertragen er in der Lage ist, nicht identisch mit  $\mathcal{A}$ . Der Einfachheit halber nehmen wir an, dass der Kanal auf der Basis des Binäralphabets  $\mathcal{B} = \{0, 1\}$  arbeitet. Für die momentan betrachtete Aufgabe denken wir uns den Kanal also als Apparat, der immer wieder von Seiten der Quelle ein Symbol aus der Menge  $\{0, 1\}$  der „Codebuchstaben“ einliest und dem Empfänger dasselbe Symbol (d.h. ohne Fehler) wieder ausgibt.

Eine konkrete Übertragung der Nachricht durch einen solchen fehlerfreien Kanal (engl.: noiseless channel) macht es also erforderlich, vor der Eingabe in den Kanal jedes Symbol  $x_i$  der Nachricht durch eine Sequenz von Binärzeichen zu codieren. Eine Lösung dieser Aufgabe, als **Quellcodierung** bezeichnet, ist also eine Abbildung, die jedem Symbol  $x_i \in \mathcal{A}$  eine Sequenz von Binärzeichen, ein sogenanntes **Codewort**, zuordnet. Für die formale Definition benötigen wir folgende Notation für eine Menge  $M$ :

$$M^* := \bigcup_{n \in \mathbb{N}} M^n. \quad (24)$$

Die Menge  $M^*$  ist also nichts Anderes als die Menge aller Wörter aus  $M$ .

**Definition 2.5 (Codierung allgemein).** Seien  $\mathcal{A}, \mathcal{B}$  endliche Mengen. Eine Codierung von  $\mathcal{A}$  durch  $\mathcal{B}$  ist eine injektive Abbildung

$$\phi : \mathcal{A} \rightarrow \mathcal{B}^* \quad (25)$$

Die Menge aller Codewörter  $\phi(\mathcal{A})$  wird auch als **Code** bezeichnet. Ist  $\mathcal{B} = \mathbb{F}_2$ , so heißt der Code binär.

Wegen der Fehlerfreiheit des zugrundegelegten Kanalmodells heißt die Aufgabe der Quellcodierung in der englischsprachigen Literatur auch „noiseless coding“.

Wir nehmen an, dass die Zahl von Binärzeichen, die der Kanal pro Zeiteinheit übertragen kann, eine vorgegebene Konstante ist, die sog. Übertragungsrate. Um eine

möglichst schnelle Übertragung einer gegebenen Nachricht zu ermöglichen, sollte die Quellcodierung also möglichst kurze Codewörter verwenden. Im Sinne einer Langzeit-Effizienz der Übertragung heißt dies: Die mittlere Länge eines Codewortes ist zu minimieren. Wenn das dem Symbol  $x_i$  zugeordnete Codewort die Länge  $l_i$  hat, lautet also das Ziel der Quellcodierung:

Finde Codes, die die **mittlere Codewortlänge**  $\sum_{i=1}^n p_i l_i$  **minimieren**.

### 2.4.2 Codes mit Präfixeigenschaft

Klarerweise soll die ursprüngliche Nachricht aus der übertragenen Binärsequenz eindeutig rekonstruierbar sein.

Folgende Anforderung an den Code stellt dies sicher: Ein Code heißt **eindeutig entzifferbar**, wenn jede endliche Sequenz von Binärsymbolen höchstens einer Nachricht entspricht.

Dass es hierfür nicht ausreicht, von der Codierungsabbildung lediglich Injektivität zu fordern, zeigt folgendes Beispiel eines Binärcodes:

$$\begin{array}{ll} x_1 & 0 \\ x_2 & 0\ 1\ 0 \\ x_3 & 0\ 1 \\ x_4 & 1\ 0 \end{array}$$

Die Binärsequenz 0 1 0 könnte durch jede der drei Nachrichten  $x_2$ ,  $x_1x_4$  oder  $x_3x_1$  erzeugt worden sein und kann daher nicht eindeutig decodiert werden.

Anhand dieses Beispiels wird das fundamentale Problem von **Codes variabler Länge** (d.h.  $l_i$  darf verschieden sein für die verschiedenen Elementarzeichen  $x_i$ ) deutlich: Wie erkennt man an der Kanalausgabe, wo das eine Codewort aufhört und das nächste beginnt?

Ein Weg, eindeutige Entzifferbarkeit zu gewährleisten, ist, den Code so zu konstruieren, dass **kein** Codewort gleichzeitig den Beginn (d.h. ein „Präfix“) eines anderen Codewortes darstellt.

[Im obigen Codebeispiel war dagegen das  $x_1$ -Codewort (0) Präfix des  $x_2$ -Codewortes (0 1), und das wiederum Präfix zum  $x_3$ -Codewort (0 1 0).]

Codes, bei denen kein Codewort Präfix eines anderen Codewortes ist, heißen **Codes mit Präfixeigenschaft**.

Die englische Bezeichnung „instantaneous codes“ weist darauf hin, dass ein solcher Code es erlaubt, eine beliebige Sequenz von Binärsymbolen Zeichen für Zeichen zu decodieren, und zwar eindeutig.

Wir machen uns das Vorgehen anhand des folgenden Beispiels klar:

Gegeben sei die Binärsequenz 1 0 1 1 1 1 0 0 0, erzeugt durch folgenden Code mit

Präfixeigenschaft:

$$\begin{array}{ll} x_1 & 0 \\ x_2 & 1\ 0\ 0 \\ x_3 & 1\ 0\ 1 \\ x_4 & 1\ 1 \end{array}$$

Wir beginnen links und schreiten solange nach rechts fort, bis die abgeschrittene Teilsequenz  $T$  einem Codewort entspricht:  $T = 1\ 0\ 1$ . Da  $T$  nicht der Beginn eines anderen Codewortes sein darf, muss  $T$  dem ersten Zeichen (hier:  $x_3$ ) der Nachricht entsprechen. Wir setzen dieses Vorgehen fort, bis ein weiteres Codewort gebildet wird, usw., bis zum Ende der Nachricht. Auf diese Weise decodieren wir obige Binärsequenz als  $x_3x_4x_2x_1$ .

Bemerkungen:

- (1) Die Präfixeigenschaft ist zwar, wie eben gezeigt, hinreichend für die eindeutige Entzifferbarkeit eines Codes, aber nicht notwendig.

Eindeutig entzifferbare Codes ohne Präfixeigenschaft haben aber den Nachteil, dass die Identität des ersten Codewortes eventuell erst durch Betrachtung eines sehr langen Abschnittes der Binärsequenz geklärt werden kann ( $\rightarrow$  Speicherbedarf, Zeitverzögerung!).

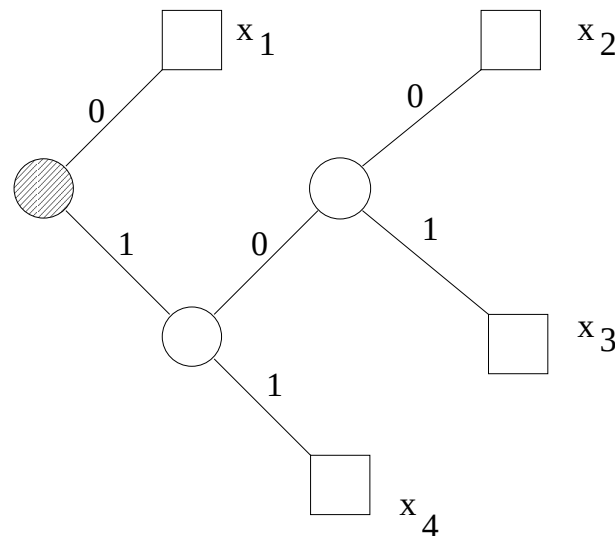
Beispiel eines nicht instantanen, aber eindeutig entzifferbaren Codes:

$$\begin{array}{ll} x_1 & 0 \\ x_2 & \underbrace{0 \dots 0}_n 1 \end{array}$$

Beim Empfang der Sequenz  $\underbrace{0 \dots 0}_{n+1} 1$  enthüllt erst die 1 am Ende der Sequenz, dass die Nachricht mit  $x_1$  beginnt.

- (2) Die Decodierung eines instantanen Codes lässt sich in Form eines binären **Entscheidungsbaums**, also eines Systems aus Punkten („Knoten“) und Verbindungslinien („Kanten“) darstellen.

Betrachten wir erneut den instantanen Beispielcode und die empfangene Sequenz  $1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0$ .



Ausgehend vom (schraffierten) Anfangsknoten verursacht die erste empfangene Binärziffer eine Verzweigung, entweder zum Endknoten  $x_1$ , falls die Ziffer eine 0 ist, oder zu einem weiteren Entscheidungsknoten, falls es eine 1 ist (wie im Beispiel). Die nächste empfangene Ziffer verursacht eine weitere Verzweigung: Zum Endknoten  $x_4$ , falls der Wert 1 empfangen wurde, oder zu einem dritten Entscheidungsknoten, falls eine 0 eintrifft (wie im Beispiel), usw.

Bei der Ankunft in einem Endknoten (erster erreichter Endknoten im Beispiel:  $x_3$ ) wird das zugehörige Zeichen ausgegeben und auf den Anfangsknoten (= Anfangszustand) zurückgesetzt.

Die Darstellung als Entscheidungsbaum verdeutlicht nochmals den instantanen Charakter der Decodierung eines Codes mit Präfixeigenschaft: Der Decodierer merkt sofort, wenn ein komplettes Zeichen  $x_i$  empfangen wurde, und seine (Decodierungs-)Entscheidung für dieses Zeichen hängt nicht von späteren Abschnitten der Binärsequenz ab.

Aufgrund ihrer angenehmen Eigenschaften wollen wir uns hinfert auf instantane Codes konzentrieren.

Angenommen, jemand gibt einen Satz von Codewortlängen  $l_1, \dots, l_n$  vor. Existiert ein instantaner Code mit diesen Codewortlängen?

Diese Frage beantwortet die folgende

**Proposition 2.6.** *Eine notwendige und hinreichende Bedingung dafür, dass für die  $n$  Symbole  $x_i$  ein instantaner Binärcode existiert, dessen Codewörter die Längen  $l_1 := l(x_1), \dots, l_n := l(x_n)$  haben, ist die **Kraft'sche***

**Ungleichung:** 
$$\sum_{i=1}^n \frac{1}{2^{l_i}} \leq 1$$

**Beweis:** Der Beweis ist anhand der Entscheidungsbaum-Darstellung leicht zu

führen:

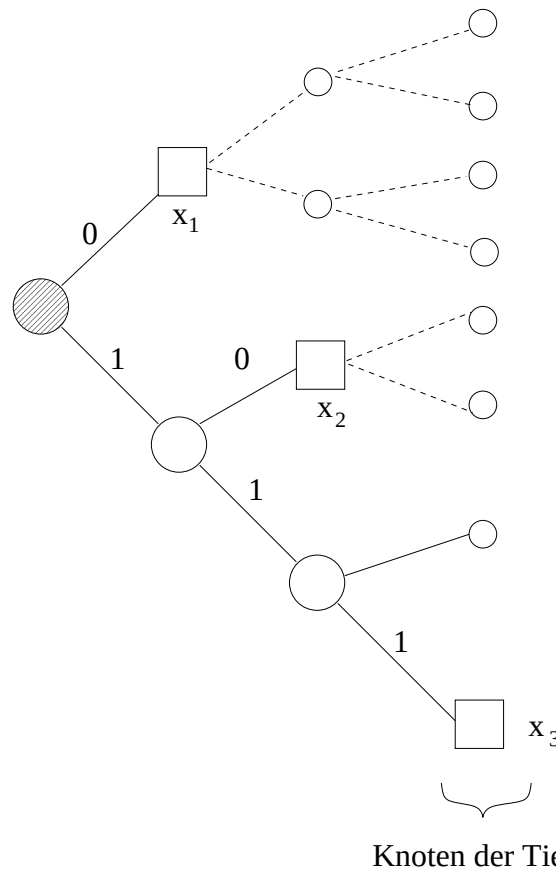
(A) „Kraft'sche Ungleichung notwendig“:

Sei ein instantaner Binärcode mit Codewortlängen  $l_1, \dots, l_n$  gegeben (o.B.d.A. können wir annehmen:  $l_1 \leq l_2 \leq \dots \leq l_n$ ).

Wir betrachten nun einen Entscheidungsbaum der Tiefe  $l_n$ , d.h. einen binären Baum, dessen Endknoten alle durch einen Pfad der Länge  $l_n$  mit dem Anfangsknoten verbunden sind (Pfadlänge = Zahl der zu durchlaufenden Knoten).

Aufgrund der Präfixeigenschaft des Codes kann jedes Codewort mit einem Knoten dieses Entscheidungsbaums - bzw. mit dem Pfad vom Anfangsknoten bis zu diesem Knoten - identifiziert werden.

Beispielsweise kann der Code  $\{0, 10, 111\}$  folgendermaßen dargestellt werden:



Da kein Codewort Präfix eines anderen Codewortes sein darf, gilt: Ist ein Knoten  $P$  für ein bestimmtes Codewort reserviert, so stehen alle Knoten des Teilbaumes, der in  $P$  entspringt, nicht mehr als Codeworte zur Verfügung. In der obigen Skizze sind die verbotenen Teilbäume gestrichelt gezeichnet.

Ein Codewort der Länge  $l_i$  schließt daher  $2^{l_n - l_i}$  Knoten der Tiefe  $l_n$  (plus

die Knoten auf den Pfaden dorthin) von der Codewortbildung aus.

Insgesamt „verbraucht“ der Code also  $\sum_{i=1}^n 2^{l_n-l_i}$  Knoten der Tiefe  $l_n$ . Da aber ein Binärbaum der Tiefe  $l_n$  insgesamt nur  $2^{l_n}$  Knoten der Tiefe  $l_n$  besitzt, gilt:

$$\sum_{i=1}^n 2^{l_n-l_i} \leq 2^{l_n} \iff \sum_{i=1}^n \frac{1}{2^{l_i}} \leq 1.$$

(B) „Kraft'sche Ungleichung hinreichend“:

Wir nehmen an, uns seien  $n$  positive Zahlen  $l_1 \leq \dots \leq l_n$  mit  $\sum_{i=1}^n 2^{-l_i} \leq 1$  vorgegeben. Um einen zugehörigen instantanen Code zu konstruieren, wählen wir im Binärbaum der Tiefe  $l_n$  einen beliebigen Knoten der Tiefe (= Pfadlänge)  $l_1$  und erhalten so das Codewort der Länge  $l_1$ .

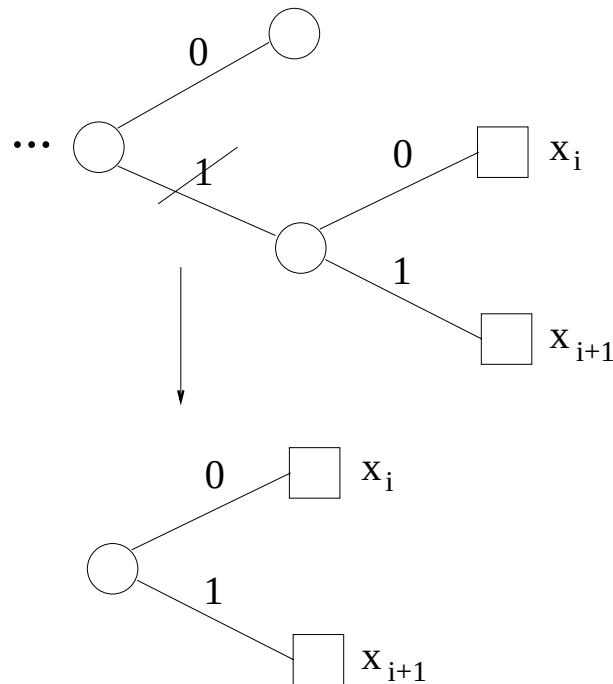
Dadurch „verbrauchen“ wir  $2^{l_n-l_1}$  Knoten der Tiefe  $l_n$ .

Wegen  $\sum_{i=1}^n 2^{-l_i} \leq 1 \iff \sum_{i=1}^n 2^{l_n-l_i} \leq 2^{l_n}$  gilt:  $2^{l_n-l_1} < 2^{l_n}$ , und mindestens ein Knoten der Tiefe  $l_n$  bleibt übrig. Auf dem Pfad zu diesem Knoten liegt ein Knoten der Tiefe  $l_2$ . Er verkörpere das Codewort der Länge  $l_2$ . Die Gesamtzahl verbrauchter Knoten (der Tiefe  $l_n$ ) ist nun  $2^{l_n-l_1} + 2^{l_n-l_2}$ .

Laut Voraussetzung ist diese Zahl ebenfalls kleiner als die Gesamtzahl  $2^{l_n}$  von Knoten der Tiefe  $l_n$ , und wir können einen Knoten zur Codewortlänge  $l_3$  wählen, etc. . Dieses Verfahren kann fortgesetzt werden, bis alle Codeworte gebildet sind. q.e.d.

Bemerkungen:

- (a) Die Kraft'sche Ungleichung besagt, dass ein instantanter Code nicht zu viele kurze Codewörter - die ja den Löwenanteil zur Summe beitragen - haben kann. Daher ist es nicht verwunderlich, dass die Kraft'sche Ungleichung eine Schlüsselrolle spielen wird, wenn es darum geht, eine untere Schranke für die mittlere Codewortlänge zu bestimmen.
- (b) Gleichheit erhält man genau für solche Codes, in denen **jedem** Endknoten des Binärbaums ein Symbol  $x_i$  zugeordnet ist.  
Falls ein Endknoten nicht genutzt wird für das binäre Codealphabet, so ist die vorausgegangene Entscheidung „verschwendet“, und die zugehörige Ziffer kann aus jedem Codewort, dessen Pfad diesen Knoten passiert, gestrichen werden:



Beweis der Gleichheit (im Falle vollständig belegter Endknoten):

Wir ordnen jeder 0/1-Entscheidung die Wahrscheinlichkeit  $\frac{1}{2}$  für den Wert 0 bzw. 1 zu und nehmen die Einzelentscheidungen als stochastisch unabhängig an. Laut Voraussetzung entspricht jeder (vom Anfangsknoten her) durchlaufbare Pfad des Binärbaums einem Codewort, also dem Binärstring eines Zeichens  $x_i$ . Die Wahrscheinlichkeit dafür, diesen konkreten Pfad (aus der Menge der  $n$  verfügbaren Pfade) zu durchlaufen, ist  $\frac{1}{2^{l(x_i)}}$ . Da mit Wahrscheinlichkeit 1 irgendein Pfad durchlaufen wird, ergibt sich  $\sum_{i=1}^n \frac{1}{2^{l(x_i)}} = 1$ .

### 2.4.3 Das Shannon'sche Codierungstheorem

Nehmen wir das eingangs formulierte Ziel der Quellcodierung wieder ins Visier: Die Suche nach instantanen Binärcodes (für ein gegebenes Alphabet  $\mathcal{A} = \{x_1, \dots, x_n\}$  und gegebene „Buchstaben“-Wahrscheinlichkeiten  $p_1, \dots, p_n$ ), die die mittlere Codewortlänge  $\bar{l} := \sum_{i=1}^n p_i l_i$  minimieren. Wir gehen dieses Problem in zwei Schritten an.

Zunächst bestimmen wir eine untere Schranke für  $\bar{l}$ . Anschließend untersuchen wir, wie nahe wir dieser unteren Schranke kommen können. Die untere Schranke liefert die folgende Proposition:

**Proposition 2.7 (Shannon'sche Codierungstheorem).** *Für jede Quelle  $Q$  und jede zugehörige Binärcodierung mit Präfixeigenschaft ist die mittlere Code-*

wortlänge  $\bar{l}$  nicht kleiner als die Entropie der Quelle  $H(Q)$ :

$$H(Q) \leq \bar{l} \quad (26)$$

**Beweis:** Aufgrund der Kraft'schen Ungleichung erhalten wir:

$$\sum_{i=1}^n \frac{1}{2^{l_i}} < \sum_{i=1}^n (p_i) = 1. \quad (27)$$

Wenden wir nun die Gibbs'sche Ungleichung auf  $p_i$  und  $q_i := \frac{1}{2^{l_i}}$  an, so folgt:

$$H(Q) = \sum_{i=1}^n p_i \log(1/p_i) \leq \sum_{i=1}^n p_i \log(1/q_i) = \bar{l}. \quad (28)$$

□

Wir wollen nun die Frage untersuchen, wie nahe wir an die durch die Entropie  $H(Q)$  gegebene Grenze herankommen. Dazu benötigen wir das  $n$ -fache Produkt einer Quelle:

**Definition 2.8 ( $n$ -fache Erweiterung).** Sei  $Q = (\mathcal{A}, p)$  eine Quelle. Dann wird das  $n$ -fache Produkt  $Q^n = (\mathcal{A}^n, p^n)$  als  $n$ -fache Erweiterung der Quelle  $Q$  bezeichnet.  $\mathcal{A}^n$  besteht aus allen Worten der Länge  $n$  über  $\mathcal{A}$ ,  $p^n$  ist in offensichtlicher Weise auf  $\mathcal{A}^n$  definiert.

Für die Entropie der  $n$ -fachen Erweiterung gilt:

$$H(Q^n) = nH(Q). \quad (29)$$

Durch Anwenden der Proposition 2.7 auf  $Q^n$  erhalten wir:

**Proposition 2.9.** Sei  $Q$  eine Quelle. So existiert zu jedem  $n \geq 1$  ein instantaner Code  $\phi_n$  auf der  $n$ -ten Erweiterung  $X^n$  derart, dass

$$\lim_{n \rightarrow \infty} \frac{\overline{l_{\phi_n}}}{n} = H(X). \quad (30)$$

Man beachte hierzu, dass jedes Symbol aus  $Q^n = (\mathcal{A}^n, p^n)$  genau  $n$  Symbole aus  $\mathcal{A}$  enthält. Das bedeutet, dass der Quotient

$$\frac{\overline{l_{\phi_n}}}{n} \quad (31)$$

gerade die mittlere Codewortlänge pro Symbol aus  $\mathcal{A}$  darstellt. Das Theorem besagt also, dass wir durch Übergang zu einer hinreichend großen Erweiterung in der Lage sind, die durchschnittliche Codewortlänge pro Symbol aus  $\mathcal{A}$  beliebig nah an die Grenze  $H(Q)$  zu bringen. Einzig unbefriedigend an dem Satz 2.9 ist der wenig konstruktive Beweis. Es gibt jedoch einen Algorithmus, der für jede Quelle eine optimale Codierung findet.

**Definition 2.10 (optimale Codierung).** Sei  $Q$  eine Quelle und  $\Phi$  die Menge aller instantanen Codes auf  $Q$ . Dann heißt  $\phi \in \Phi$  optimal, falls

$$\bar{l}_\phi \leq \bar{l}_\psi \text{ für alle } \psi \in \Phi. \quad (32)$$

Die so genannte *Huffman-Codierung*, für die es eine algorithmische Beschreibung gibt, ist eine optimale Codierung, vgl. [Ro91].

## 2.5 Das Shannon'sche Kanalcodierungstheorem

Wir wollen in diesem Abschnitt die Frage nach der Existenz von guten Codes beantworten. Dazu bedarf es allerdings noch einiger begrifflicher Vorbereitung.

**Definition 2.11.** Seien  $X$  und  $Y$  reellwertige, diskrete Zufallsvariablen auf einem diskreten Wahrscheinlichkeitsraum  $(\mathcal{A}, p)$ . Für ein  $y \in \text{Bild}(Y)$  ist die bedingte Entropie von  $X$  unter der Bedingung  $Y = y$  gegeben durch

$$H(X|Y = y) = - \sum_{x \in \text{Bild}(X)} p(X = x|Y = y) \log[p(X = x|Y = y)]. \quad (33)$$

Wir können  $H(X|Y = y)$  als Maß für die Ungewissheit über die Größe  $X$  verstehen, wenn bekannt ist, dass  $Y$  den Wert  $y$  annimmt. Wir mitteln diese Zahlen noch über  $y \in \text{Bild}(Y)$ :

**Definition 2.12 (bedingte Entropie).** Die bedingte Entropie von  $X$  unter  $Y$  ist

$$H(X|Y) = - \sum_{y \in \text{Bild}(Y)} p(Y = y) \cdot H(X|Y = y). \quad (34)$$

Die Interpretation ergibt sich aus der vorangegangenen:  $H(X|Y)$  ist die im Mittel verbleibende Ungewissheit, die über den Wert von  $X$  noch besteht, wenn der Wert von  $Y$  bekannt ist. Die Entropien von  $X \times Y$  und  $X$  unterscheiden sich gerade um die bedingte Entropie von  $Y$  unter  $X$ , wie der folgende Satz zeigt. Wir können ihn so interpretieren: Die Ungewissheit über den Wert von  $X \times Y$  ist die Summe aus der Ungewissheit über den Wert von  $X$  und der Ungewissheit über den von  $Y$ , sobald  $X$  bekannt ist.

**Proposition 2.13.**  $H(X \times Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$ .

Die Ungewissheit über den Wert von  $X$  sollte durch die Kenntnis von  $Y$  nicht größer werden, und das ist auch nicht der Fall:

**Proposition 2.14.** (i)  $H(X|Y) \leq H(X)$ ,

$$(ii) \ H(X \times Y) \leq H(X) + H(Y),$$

Wenn in den Ungleichungen des Satzes 2.14 Gleichheit gilt, so wissen wir über  $X$  mit und ohne Kenntniss von  $Y$  gleich viel. Dies bedeutet, dass  $Y$  über  $X$  keine Information enthält. Dieser Gedanke wird durch die folgende Definition formalisiert:

**Definition 2.15.** Der Informationsgehalt von  $X$  über  $Y$  ist  $I(X, Y) := H(Y) - H(Y|X)$ .

Die Information von  $X$  über  $Y$  ist also die Größe, um die die Ungewissheit über  $Y$  durch die Kenntniss von  $X$  vermindert wird. Sie kann nicht größer sein als die Ungewissheit über  $Y$  selbst (und nicht größer als die Ungewissheit über  $X$ ). Ferner gibt  $X$  die gleiche Information über  $Y$  wie  $Y$  über  $X$ :

**Proposition 2.16.**

$$(i) \ I(X, Y) \leq \min(H(X), H(Y)),$$

$$(ii) \ I(X, Y) = I(Y, X).$$

Wir wollen die Information jetzt in Bezug auf eine Quelle und Kanäle deuten. Betrachten wir dazu also eine Quelle  $Q = (\mathcal{A}, p)$  und einen Kanal  $(\mathcal{A}, \text{calb}, R)$ . Wir erinnern an den Wahrscheinlichkeitsraum  $(\mathcal{A} \times \mathcal{B}, r)$  mit

$$r(a, b) := p(a) \cdot r_a(b), \quad a \in \mathcal{A}, b \in \mathcal{B}. \quad (35)$$

Wir bezeichnen mit  $B$  die zur Randverteilung auf  $\mathcal{B}$  gehörende Zufallsvariable, also diejenige Zufallsvariable, die die Ausgabe des Kanals beschreibt. Als Information des Kanals bei gegebener Quelle  $Q$  wollen wir die Ausgabe in Abhängigkeit der Quelle betrachten, d.h.

$$I_r(\mathcal{A}, \mathcal{B}) := H(Q) - H(Q|B) \quad (36)$$

$$= H(Q) - \sum_{a \in \mathcal{A}, b \in \mathcal{B}} r(a, b) \log(r(a|b)) \quad (37)$$

Die Größe

$$I_r(\mathcal{A}, \mathcal{B})$$

beschreibt also die Menge an Information, die der Kanal für eine fest gewählte Quelle überträgt. Daher ist die folgende Definition einleuchtend:

**Definition 2.17 (Kapazität).** Die Zahl

$$C(K) := \max\{I_r(\mathcal{A}, \mathcal{B}) | (\mathcal{A}, p) \text{ Quelle mit Alphabet } \mathcal{A}\} \quad (38)$$

heißt Kapazität des Kanals  $K = (\mathcal{A}, \mathcal{B}, R)$ .

Der Kanal kann also maximal so viel Information übertragen, wie seine Kapazität angibt. Dies entspricht dem intuitiven Kapazitätsgebrauch. Die von uns eingeführte Kapazität wird endgültig durch den Kanal-Codierungssatz gerechtfertigt.

**Proposition 2.18 (Kanalcodierungstheorem).** *Sei  $(\mathcal{A}, \mathcal{B}, R)$  ein diskreter, gedächtnisloser Kanal mit der Kapazität  $C$ . Für jede Rate  $R < C$  und für jedes  $\varepsilon > 0$  gibt es eine natürliche Zahl  $n$  und einen Code  $\mathcal{C} = \{c_1, \dots, c_M\}$  bestehend aus Worten der Länge  $n$  mit den folgenden Eigenschaften:*

- (i)  $M \leq 2^{Rn}$ ,
- (ii)  $p_e(i) < \varepsilon$  für  $i = 1, \dots, M$  (dabei ist  $p_e(i)$  die Wahrscheinlichkeit, dass bei gesendetem  $c_i$  falsch decodiert wird.)

Das Kanalcodierungstheorem von Shannon liefert also tatsächlich die Existenz beliebig guter binärer Codes. Leider ist kein einziger konstruktiver Beweis bekannt. Die Hauptidee des Beweises ist, dass ein zufällig gewählter Code aus  $M$  Wörtern mit hoher Wahrscheinlichkeit eine kleine Irrtumswahrscheinlichkeit liefert. Die Informationstheorie garantiert also die Existenz beliebig guter Codes. Aber es bleibt ein Problem:

**Frage 1** Wie findet man gute Codes ?

Doch selbst wenn man dieses Problem gelöst hätte, wäre das für die Praxis noch längst nicht genug. So könnte es zum Beispiel sein, dass ein guter Code so komplizierte Codierungs- und Decodierungsvorschriften besitzt, dass er für praktische Anwendungen unbrauchbar wäre. Daher stellen sich zwei weitere Fragen:

**Frage 2** Wie implementiert man gute Codes?

**Frage 3** Wie implementiert man die Decodierung?

Die Codierungstheorie beschäftigt sich mit allen drei Fragen. Leider sind alle drei Fragen im Allgemeinen sehr schwierig zu beantworten.

### 3 Einschub: "Codierung im Alltag"

Wir wollen an dieser Stelle einige kurze Beispiele aus der Praxis vorstellen, bevor wir uns in den nächsten Kapiteln mit Klassen von Codes beschäftigen, die eine bestimmte algebraische Struktur aufweisen. Noch lange bevor es die Codierungstheorie als mathematische Disziplin gab, hat man zur Fehlererkennung Prüfziffern eingeführt. Diese dienen zur Erkennung von Eingabefehlern (nicht zur Korrektur!) und werden verwendet, wenn die Eingabe wiederholt werden kann. Laut [Ver69] ist die mit weitem Abstand häufigste Fehlerquelle bei der Eingabe eines Codewortes ein Einzelfehler ( $a \rightarrow b$ ), bei dem also genau ein Symbol falsch eingegeben wird. Dies liefert 79 % aller überhaupt auftretenden Fehler. Daher ist es von größtem Interesse, solche Fehler

| Fehlertyp                                                              | Symbol                                      | relative Häufigkeit |
|------------------------------------------------------------------------|---------------------------------------------|---------------------|
| Einzelfehler (Verwechslung einer Ziffer)                               | $a \rightarrow b$                           | 79,0%               |
| Nachbar-Transpositionen (Vertauschung benachbarter Ziffern)            | $ab \rightarrow ba$                         | 10,2%               |
| Sprungtranspositionen (Vertauschung einer Ziffer mit der übernächsten) | $abc \rightarrow cba$                       | 0,8%                |
| Zwillingsfehler                                                        | $aa \rightarrow bb$                         | 0,6%                |
| phonetische Fehler (z.B.: dreißig – dreizehn)                          | $a0 \leftrightarrow 1a \ (a = 3, \dots, 9)$ | 0,5%                |
| Sprung-Zwilling-Fehler                                                 | $aca \rightarrow bcb$                       | 0,3%                |
| übrige Fehler (zufällige Fehler)                                       |                                             | 8,6%                |

Abbildung 3: Tabelle der häufigsten Fehlerarten

zu erkennen. Dazu wird in der Praxis auf folgendes Schema zurück gegriffen:

**Definition 3.1.** Sei  $A$  eine endliche Gruppe. Eine Prüfzeichencodierung über  $A$  besteht aus  $n$  Permutationen  $\pi_1, \dots, \pi_n$  in der symmetrischen Gruppe  $S_A$  sowie einem Element  $c \in A$ . Dabei wird ein Wort  $a_1 \dots a_{n-1} \in A^{n-1}$  so um ein Prüfzeichen  $a_n$  erweitert, dass die folgende Kontrollgleichung gilt:

$$\pi_1(a_1) \cdots \pi_n(a_n) = c. \quad (39)$$

Dabei kann man sich leicht überlegen, dass für jede Prüfzeichen-Codierung Folgendes gilt:

- (i) Zu gegebenem  $a_1, \dots, a_{n-1}$  ist  $a_n$  eindeutig bestimmt.

(ii) Jede Prüfzeichen-Codierung über einer Gruppe erkennt alle Einzelfehler.

Ein besonders nahe liegender Spezialfall ist die Codierung modulo  $m$ :

**Definition 3.2.** Sei  $A = \mathbb{Z}_m$  (additiv geschrieben) sowie  $n$  Zahlen  $w_1, \dots, w_n$  mit  $\text{ggT}(w_i, m) = 1$  gegeben. Dann ist

$$\pi_i := \{x \mapsto w_i x\} \quad (40)$$

für  $i = 1, \dots, n$  eine Permutation von  $A$ . Mit  $c = 0$  und  $w_n = \pm 1$  wird aus der Kontrollgleichung 39

$$a_n = \pm \sum_{i=1}^{n-1} w_i a_i \pmod{m}. \quad (41)$$

**Beispiel 3.3 (ISBN-Nummer).** Jedem Buch wird eine ISBN-Nummer zugeordnet. Zum Beispiel

$$3 - 411 - 14263 - 4 \quad (42)$$

Dabei gibt die erste 3 das Land an (Deutschland), 411 den Verlag (BI) und 14263 das Buch. Die letzte Ziffer ist eine Prüfziffer, die mit  $m = 11$  folgendermaßen gebildet wird:

$$a_{10} = \sum_{i=1}^9 i a_i \pmod{11} \quad (43)$$

Falls sich  $a_{10} = 10$  ergibt, so verwendet man dafür die Prüfziffer  $X$ . (Z.B. hat Jacobsons Buch "Basic Algebra II" die ISBN-Nummer  $0 - 7167 - 1079 - X$ .)

Natürlich stellt sich die Frage, ob Prüfziffercodierungen auch andere Fehlerarten erkennen, wie z.B. die zweit-häufigste Fehlerursache, Nachbartranspositionen ( $ab \rightarrow ba$ ), die 10,2% der Fehler ausmachen. Diese Frage führt in die elementare Gruppentheorie. Ohne Details wollen wir zumindest das Resultat für abelsche Gruppen zitieren:

**Proposition 3.4.** Sei  $G$  eine abelsche Gruppe. Genau dann gibt es eine Prüfziffercodierung über  $G$ , die jede Nachbartransposition erkennt, wenn es einen Orthomorphismus für  $G$  gibt.

Dabei ist ein Orthomorphismus eine Permutation  $\delta$ , für die

$$\delta : G \rightarrow G, \quad (44)$$

$$x_i \mapsto x^{-1} \delta_i(x) \quad (45)$$

wieder eine Permutation ist.

Insbesondere folgt daraus:

**Proposition 3.5.** *Ist  $G$  abelsch und  $|G| \equiv 2 \pmod{4}$ , so kann es keine Prüfzeichencodierung über  $G$  geben, die alle Nachbartranspositionen erkennt.*

Dies gilt z.B. für die zyklische Gruppe der Ordnung 10, weshalb es kein Zufall ist, dass für die ISBN-Nummer die Ordnung 11 gewählt wurde. Dabei muss man als kleinen Nachteil das Symbol  $X$  in Kauf nehmen.

Nun gibt es aber noch eine weitere, nicht-abelsche Gruppe der Ordnung 10, die Dieder-Gruppe  $D_5$ , die gegeben ist als

$$D_5 = \langle a, b : a^2 = b^5 = 1, bab = a^{-1} \rangle \quad (46)$$

Geometrisch handelt es sich um die von Drehungen und Spiegelungen eines regelmäßigen Fünfecks erzeugte Gruppe. Diese Gruppe hat 10 Elemente. Wir betrachten die folgende Permutation:

$$\delta = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 5 & 7 & 6 & 2 & 8 & 3 & 0 & 9 & 4 \end{pmatrix} \quad (47)$$

Wie das folgende Beispiel zeigt, wurde dieses etwas esoterisch anmutende Prüfziffersystem tatsächlich (bis zur Einführung des Euro) verwendet.

**Beispiel 3.6.** Die Kennzeichnung der deutschen Banknoten ab 1990 erfolgte durch 11 Zeichen, die dem Alphabet  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, D, G, K, L, N, S, U, Y, Z\}$  entnommen sind, z.B.

$$AD6913433K8 \quad (48)$$

Ersetzt man die Buchstaben nach folgender Tabelle

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c} A & D & G & K & L & N & S & U & Y & Z \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \quad (49)$$

durch Ziffern, so erhält man Folgen der Länge 11 über  $D_5$ , im Beispiel also:

$$01691343338 \quad (50)$$

Jede solche Folge genügt der Prüfgleichung

$$\delta(a_1) \cdots \delta(a_{10})a_{11} = 0. \quad (51)$$

Die Permutation  $\delta$  aus (47) hat die Eigenschaft  $x\delta(y) \neq y\delta(x)$  für alle  $x, y \in D_5$ , und deswegen können nicht nur Einzelfehler, sondern auch (fast) alle Nachbartranspositionen erkannt werden.

## 4 Lineare Blockcodes

### 4.1 Elementare Eigenschaften

Lineare Codes haben den Vorteil, dass die Codierung sehr einfach ist. Nahezu alle in der Praxis verwendeten Codes gehören dieser Klasse an. Wir beschränken uns wie immer auf binäre Codes und erinnern also zuerst an die Definition:

**Definition 4.1 (Linearer Block-Code).** Ein  $(n, k, d)$ -Block-Code  $\mathcal{C}$  heißt linear, falls  $\mathcal{C} \subset \mathbb{F}_2^n$  ein  $(k\text{-dim.})$  linearer Unterraum des Vektorraums  $\mathbb{F}_2^n$  ist.

Für einen linearen Code  $\mathcal{C}$  ist die Bestimmung von  $d$  besonders einfach. Dazu benötigen wir einen weiteren Begriff:

**Definition 4.2.** Es sei  $x \in \{\pm 1\}^n$ . Das Gewicht  $w(x)$  ist die Anzahl der Koordinaten  $\neq 0$  von  $x$ . Das Minimalgewicht von  $\mathcal{C}$  ist

$$\min\{w(c) : c \in \mathcal{C} \setminus \{0\}\} \quad (52)$$

Folgender Zusammenhang besteht zwischen Hamming-Abstand, Gewicht, Minimalabstand und Minimalgewicht:

**Proposition 4.3.** (i) Für  $x, y \in \mathbb{F}_2^n$  gilt:  $d(x, y) = w(x - y)$ .

(ii) Für einen linearen Code ist der Minimalabstand  $d$  gleich dem Minimalgewicht.

Um den Minimalabstand eines linearen Codes zu bestimmen, genügt es also, sein Minimalgewicht zu kennen. Aufgrund der linearen Struktur gibt es dafür wieder Methoden aus der linearen Algebra. Überhaupt kann man das Studium linearer Codes als Teil der Linearen Algebra über endlichen Körpern verstehen. Jedoch ist diese Lineare Algebra basisabhängig: so ist etwa das Minimalgewicht keine Invariante unter linearen Abbildungen. Während man in der Linearen Algebra die Basis in der Regel möglichst einfach wählt, so wird z.B. die Basis eines Unterraums gerne zu einer Basis des gesamten Raums fortgesetzt, geht man hier eher umgekehrt vor: Damit das Minimalgewicht möglichst groß wird, muss die Basis des Codes möglichst "schräg" zur natürlichen Basis des  $\mathbb{F}_2^n$  liegen.

Wir wollen nun einige Überlegungen zur Erzeugung von linearen Codes anstellen. Als Erstes stellen wir fest, dass jeder lineare Code einen zweiten festlegt, den dualen Code, der durch den Orthogonalraum gegeben ist:

**Definition 4.4 (dualer Code).** Sei  $\mathcal{C}$  ein linearer  $(n, k, d)$ -Code. Dann heißt der orthogonale Unterraum  $\mathcal{C}^\perp$  (Orthogonalität bzgl. des Standardskalarprodukts)

der zu  $C$  duale Code. Offenbar ist  $C^\perp$  ein Code der Dimension  $n - k$ . Explizit gilt also:

$$C^\perp = \{y : \langle x, y \rangle = 0 \text{ für alle } x \in C\}, \quad (53)$$

wobei  $\langle x, y \rangle$  das übliche Skalarprodukt bezeichnet, also

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i. \quad (54)$$

Wie kann man lineare Codes beschreiben? Allgemein werden lineare Unterräume von Erzeugendensystemen erzeugt, daher kann man diese Beschreibung auch für lineare Codes verwenden.

**Definition 4.5 (Generatormatrix, Kontrollmatrix).** Sei  $C$  ein linearer  $(n, k)$ -Code. Ist  $a_1, \dots, a_k$  ein Erzeugendensystem für  $C$ , so heißt die Matrix mit den Zeilen  $a_1, \dots, a_k$  eine Generatormatrix  $G$  für  $C$ . Eine Generatormatrix  $H$  für  $C^\perp$  heißt Kontrollmatrix für  $C$ .

Der Name Kontrollmatrix wird durch den folgenden Satz gerechtfertigt:

**Proposition 4.6.** Ist  $H$  eine Kontrollmatrix des linearen  $(n, k)$ -Codes, so gilt für jeden Vektor  $c \in \mathbb{F}_2^n$ :

$$c \in C \Leftrightarrow Hc^T = 0 \quad (55)$$

Mit Hilfe der Matrix  $H$  kann man also "kontrollieren", ob ein Vektor ein Codewort darstellt oder nicht. Dies ist insbesondere für die Decodierung von Bedeutung.

Wir erinnern daran, dass wir einen Block-Code immer als systematisch vorausgesetzt haben. Dies ermöglicht immer eine Darstellung mittels einer systematischen Generatormatrix, also einer Matrix  $G$  der Gestalt

$$G = (I_k | A) \quad (56)$$

mit einer  $k \times (n - k)$ -Matrix  $A$ . Für eine systematische Generatormatrix läßt sich die Kontrollmatrix unmittelbar angeben.

**Proposition 4.7.** Sei  $G = (I_k | A)$  eine systematische Generatormatrix für einen Code  $C$ . Dann ist

$$H = (-A^T | I_{n-k}) \quad (57)$$

eine Kontrollmatrix für  $C$ .

**Beispiel 4.8.** Für den  $(7, 4)$ -Hamming-Code erhält man als Generatormatrix

$$\left( \begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right) \quad (58)$$

und als Kontrollmatrix

$$H = \left( \begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right) \quad (59)$$

Der nächste Satz gibt die Möglichkeit, das Minimalgewicht und damit den Minimalabstand mittels linearer Algebra zu ermitteln:

**Proposition 4.9.** *Sei  $H$  die Kontrollmatrix eines linearen  $(n, k)$ -Codes. Dann gilt:*

- (i)  $k = \dim \mathcal{C} = n - \text{rank } H$ .
- (ii) *Das Minimalgewicht von  $\mathcal{C}$  ist die kleinste Zahl  $d$ , für die es  $d$  linear abhängige Spalten in  $H$  gibt.*

**Beispiel 4.10.** Wir wollen anhand der Kontrollmatrix  $H$  des  $(7, 4)$ -Hamming-Codes den Minimalabstand bestimmen. Dazu müssen wir also die minimale Anzahl linear abhängiger Spalten in

$$H = \left( \begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right) \quad (60)$$

ermitteln. Man erkennt sofort, dass keine 2 Spalten linear abhängig sind. Dazu müsste eine Spalte ein Vielfaches einer anderen sein. Andererseits ist beispielsweise die erste Spalte eine Linearkombination der letzten beiden, es gilt:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (61)$$

Daher sind 3 Spalten linear abhängig. Somit ist der Minimalabstand des  $(7, 4)$ -Hamming-Codes 3.

Noch einfacher wird es, wenn man umgekehrt vorgeht, also Codes durch Angabe der Kontrollmatrix definiert. Dann kann man die Parameter direkt ablesen.

**Beispiel 4.11 (Hamming-Codes).** Zu gegebenem  $k \in \mathbb{N}$  setzen wir  $n = 2^k - 1$ . Da  $n$  die Anzahl der vom Nullvektor verschiedenen Elemente in  $\mathbb{F}_2^k$  ist, gibt es  $n$  ein-dimensionale Unterräume in  $\mathbb{F}_2^k$ . Jeder wird aufgespannt von einem Vektor der Länge  $k$ , so dass wir insgesamt  $n$  paarweise linear unabhängige Vektoren der Länge  $k$  erhalten, die wir zu einer  $k \times n$ -Matrix  $H$  zusammenfassen. Sei  $\mathcal{C}$  der durch die Kontrollmatrix  $H$  definierte Code. Da offensichtlich  $H$  den  $\text{rank } H = k$  hat, gilt  $\dim \mathcal{C} = n - k$ . Nach dem vorangegangenen Satz hat  $\mathcal{C}$  Minimalgewicht 3.  $\mathcal{C}$  ist also ein linearer  $(n, n - k, 3)$ -Code, der  $(n, n - k, 3)$ -**Hamming-Code**.

## 4.2 Schranken für Codelängen

Bei gleichem Minimalabstand werden wir denjenigen Code als besser ansehen, der die größere Rate hat. Fixieren wir die Codelänge  $n$ , so stellt sich die Frage, für welche  $k$  es noch einen  $(n, k, d)$ -Code gibt. Wir bezeichnen das maximale  $k$ , für das es noch einen linearen  $(n, k, d)$ -Code gibt, mit  $K(n, d)$ . Die Bestimmung der  $K(n, d)$  und die Herleitung guter Schranken, also Abschätzungen für  $K(n, k)$  sind ein zentrales Problem der kombinatorischen Codierungstheorie.

Als erstes Ergebnis wollen wir die Singleton-Schranke angeben:

**Proposition 4.12 (Singleton-Schranke).** *Sei  $\mathcal{C}$  ein linearer  $(n, k, d)$ -Code. Dann gilt:*

$$k \leq n - d + 1 \quad (62)$$

Ein Code, der die Singleton-Schranke mit Gleichheit erfüllt, heißt **MDS-Code**. Solche Codes finden beispielsweise im CD-Player Anwendung.

Es gibt auch untere Schranken, welche die Existenz bestimmter Codes garantieren:

**Proposition 4.13 (Gilbert-Varshamov-Schranke).** *Wenn*

$$\sum_{i=0}^{d-2} \binom{n-1}{i} < 2^{n-k} \quad (63)$$

*ist, dann gibt es einen linearen  $(n, k, d)$ -Code.*

**Beispiel 4.14.** So gilt zum Beispiel für  $n = 13, k = 4, d = 5$ :

$$\sum_{i=0}^{5-2} \binom{13-1}{i} = 299 < 512 = 2^{13-4}, \quad (64)$$

es gibt also einen linearen  $(13, 4, 5)$ -Code.

Natürlich kann man auch die umgekehrte Frage stellen. Will man  $k$  Informationsbits mit einer bestimmten Sicherheit  $d$  übertragen, wie groß muss dann mindestens die Blocklänge  $n$  sein? Wir bezeichnen analog den kleinsten Wert von  $n$ , für den es einen linearen  $(n, k, d)$ -Code gibt, mit  $N(k, d)$ . Im folgenden Satz bezeichnet

$$[x] : \text{ die größte ganze Zahl } \leq x. \quad (65)$$

**Proposition 4.15 (Griesmer-Schranke).** *Es gilt zu gegebenem  $k$  und  $d$ :*

$$N(k, d) \geq d + \left\lceil \frac{d}{2} \right\rceil + \dots + \left\lceil \frac{d}{2^{k-1}} \right\rceil \quad (66)$$

**Beispiel 4.16.** Nehmen wir an, wir wollten  $N := N(5, 8)$  bestimmen. Aus der Griesmer-Schranke folgt:

$$N \geq 8 + 4 + 2 + 1 + 1 = 16. \quad (67)$$

Es ist also mindestens eine Blocklänge von  $n = 16$  nötig, um einen linearen  $(n, 5, 8)$ -Code zu konstruieren. Auf anderem Wege kann man tatsächlich einen linearen  $(16, 5, 8)$ -Code konstruieren. Beide Ergebnisse zusammen bedeuten:  $N(5, 8) = 16$ .

### 4.3 Zyklische Codes

Zyklische Codes sind spezielle lineare Codes, bei denen die Codierung besonders einfach ist und auch gute Decodierverfahren existieren. Sie finden sich deshalb in zahlreichen Anwendungen.

**Definition 4.17.** Ein linearer  $(n, k, d)$ -Code heißt zyklisch, wenn aus  $(c_0, \dots, c_{n-1}) \in \mathcal{C}$  stets  $(c_{n-1}, c_0, \dots, c_{n-2}) \in \mathcal{C}$  folgt.

Das Studium zyklischer Codes erfordert etwas mehr Kenntnisse in Algebra, insbesondere über Ideale in Restklassenringen. Wir lassen daher die Details aus. Sie führen dazu, dass man zu jedem zyklischen Code ein Generatorpolynom definiert. Dieses ist eindeutig bestimmt und legt, wie der nächste Satz zeigt, eine Wahl der Generatormatrix fest.

**Proposition 4.18 (Generatorpolynom).** *Sei  $\mathcal{C}$  ein zyklischer Code. Dann gibt es ein Polynom*

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_rx^r, \quad (68)$$

*Generatorpolynom von  $\mathcal{C}$  genannt, so dass die folgende Matrix*

$$G = \begin{pmatrix} g_0 & g_1 & \dots & g_r & 0 & \dots & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_r & 0 & \dots & 0 \\ \ddots & \dots & \dots & & & & \dots & \ddots \\ \ddots & & \dots & \dots & & & \dots & \ddots \\ 0 & \dots & \dots & 0 & g_0 & g_1 & \dots & g_r \end{pmatrix} \quad (69)$$

*eine Generatormatrix für  $\mathcal{C}$  ist.*

Aus der Theorie weiß man, dass das Generatorpolynom  $g$  ein Teiler von  $x^n - 1$  ist. Auch das Polynom  $h = (x^n - 1)/g$  hat eine interessante Bedeutung.

**Proposition 4.19 (Kontrollpolynom).** *Sei  $\mathcal{C}$  ein zyklischer Code mit Generatorpolynom  $g$ . Dann ist  $h = (x^n - 1)/g$  ein Polynom*

$$h(x) = h_0 + h_1x + h_2x^2 + \dots + h_kx^k, \quad (70)$$

und die Matrix

$$H = \begin{pmatrix} 0 & \dots & \dots & 0 & h_k & \dots & h_1 & h_0 \\ 0 & \dots & 0 & h_k & \dots & h_1 & h_0 & 0 \\ \ddots & & & & & & & \ddots \\ 0 & \ddots & & \dots & & \dots & \dots & \ddots \\ h_k & \dots & h_1 & h_0 & 0 & & \dots & 0 \end{pmatrix} \quad (71)$$

ist eine Kontrollmatrix für  $\mathcal{C}$ . Daher heißt  $h$  Kontrollpolynom.

**Beispiel 4.20 ((7, 4)-Hamming-Code).** Der (7, 4)-Hamming-Code ist zyklisch und hat das Generatorpolynom

$$g(x) = 1 + x + x^3. \quad (72)$$

Daher erhalten wir als Kontrollpolynom

$$h(x) = (x^7 - 1)/g = 1 + x + x^2 + x^4. \quad (73)$$

Wir erhalten als Generatormatrix

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad (74)$$

und als Kontrollmatrix

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (75)$$

Auch der duale Code ist zyklisch und läßt sich mit Hilfe des Kontrollpolynoms bestimmen.

**Proposition 4.21.** *Sei  $\mathcal{C}$  ein zyklischer Code mit Generatorpolynom  $g$  und Kontrollpolynom  $h = h_0 + h_1x + h_2x^2 + \dots + h_kx^k$ . Dann ist auch  $\mathcal{C}^\perp$  ein zyklischer Code mit Generatorpolynom*

$$g^\perp = \frac{x^k h(x^{-1})}{h_0}. \quad (76)$$

**Beispiel 4.22.** Für den (7,4)-Hamming-Code wird der zyklische duale Code erzeugt von dem Polynom

$$g^\perp = \frac{x^4 h(x^{-1})}{1} = 1 + x^2 + x^3 + x^4. \quad (77)$$

Wir wollen weitere Codes als zyklische Codes ausmachen. Dazu müssen wir die Codes auf eine algebraische Art beschreiben. Dazu erinnern wir an ein paar Grundkenntnisse über Ringe. Betrachten wir den Körper  $\mathbb{F}_2$ , so können wir den Polynomring über  $\mathbb{F}_2$  bilden:

$$\mathbb{F}_2[x] = \{p(x) = \sum_{i=0}^r p_i x^i \mid p_i \in \mathbb{F}_2\} \quad (78)$$

Ein Element von  $\mathbb{F}_2[x]$  ist das Polynom  $x^n - 1$ . Teilen wir das von  $x^n - 1$  erzeugte Ideal aus dem Polynomring heraus, so entsteht der Quotientenring

$$R := \mathbb{F}_2[x]/(x^n - 1). \quad (79)$$

Offenbar ist  $R$  zugleich ein  $n$ -dimensionaler Vektorraum über  $\mathbb{F}_2$ , mit kanonischer Basis

$$1, x, x^2, \dots, x^{n-1} \quad \text{mod } (x^n - 1) \quad (80)$$

Jedem Vektor in  $\mathbb{F}_2^n$  ist damit in eindeutiger Weise ein Polynom in  $R$  zugeordnet:

$$(c_0, c_1, \dots, c_{n-1}) \leftrightarrow c_0 + c_1 x + \dots + c_{n-1} x^{n-1} \quad \text{mod } (x^n - 1) \quad (81)$$

Auf diese Art und Weise können wir jeden binären Blockcode  $\mathcal{C} \subset \mathbb{F}_2^n$  der Länge  $n$  als Teilmenge  $\mathcal{C} \subset R$  auffassen und das werden wir im Folgenden auch tun.

**Definition 4.23 (Nullstellen von  $\mathcal{C}$ ).** Sei  $\mathcal{C}$  ein zyklischer Code der Länge  $n$  über  $\mathbb{F}_2$  mit Generatorpolynom

$$g(x) = \prod_{i=1}^r (x - \alpha_i) \quad (82)$$

Dann heißen die  $\alpha_i$  die Nullstellen von  $\mathcal{C}$ .

Zyklische Codes lassen sich in wundersamer Weise über ihre Nullstellen charakterisieren:

**Proposition 4.24.** Sei  $\mathcal{C}$  ein zyklischer Code der Länge  $n$  über  $\mathbb{F}_2$  mit Nullstellen  $\alpha_1, \dots, \alpha_r$ . Dann gilt:

$$\mathcal{C} = \mathcal{C}(\alpha_1, \dots, \alpha_r) := \{c \in \mathcal{C} \mid c(\alpha_1) = \dots = c(\alpha_r) = 0\} \quad (83)$$

Ein zyklischer Code besteht also gerade aus den Codeworten, die an den Nullstellen des Codes verschwinden. Genauso kann man umgekehrt vorgehen. Beschränkt man sich auf ein Nullstellengebilde, so erhält man zyklische Codes. Im einfachsten Fall einer Nullstelle erhalten wir alte Bekannte wieder, von denen wir damit implizit zeigen, dass sie zyklisch sind.

**Proposition 4.25.** *Sei  $n = 2^k - 1$  und  $\alpha$  eine primitive  $n$ -te Einheitswurzel in  $\mathbb{F}_2^k$ . Dann ist der zyklische Code*

$$C = C(\alpha) = \{c \in R \mid c(\alpha) = 0\} \quad (84)$$

*äquivalent zum  $(n, n - k)$ -Hamming-Code, der damit insbesondere zyklisch ist. Das Generatorpolynom ist gegeben als*

$$g(x) = (x - \alpha)(x - \alpha^q) \cdot \dots \cdot (x - \alpha^{q^{k-1}}). \quad (85)$$

#### 4.4 BCH-Codes

BCH-Codes sind spezielle zyklische Codes. Sie sind benannt nach R.C. Bose, D.K. Ray-Chaudhuri und A. Hocquenghem. Sie haben eine bestimmte Struktur, für die gute Decodierverfahren existieren. Die so genannten Reed-Solomon-Codes sind spezielle BCH-Codes und werden beispielsweise zur Fehlerkorrektur im CD-Spieler verwendet.

**Definition 4.26 (BCH-Code).** Sei  $\alpha$  eine primitive  $n$ -te Einheitswurzel über  $\mathbb{F}_2$ . Der Code

$$\mathcal{C} = C(\alpha, \alpha^2, \alpha^3, \dots, \alpha^{\delta-1}) \quad (86)$$

heißt BCH-Code zum Abstand  $\delta$ . Im Falle  $n = 2^m - 1$  heißt  $\mathcal{C}$  primitiver BCH-Code.

Die Variable  $\delta$  heißt "of designed distance"  $\delta$ . Für den Minimalabstand  $d$  eines BCH-Codes gilt, dass er mindestens so groß wie  $\delta$  ist:

**Proposition 4.27.** *Sei  $\mathbb{F}_2^m$  der Zerfällungskörper von  $x^n - 1$  über  $\mathbb{F}_2$ . Für jeden BCH-Code der Länge  $n$  über  $\mathbb{F}_2$  zum Abstand  $\delta$  gilt:*

$$(i) \quad d \geq \delta.$$

$$(ii) \quad k \geq n - m(\delta - 1).$$

$$(iii) \quad \text{Es gibt einen BCH-Code zum Abstand } \delta = 2t + 1 \text{ mit } k \geq n - mt.$$

BCH-Codes bieten also den Vorteil, dass man ihre Eigenschaften bis zu einem gewissen Grad vorschreiben kann. Es ist allerdings im Allgemeinen schwierig, den Minimalabstand genau zu bestimmen,  $\delta$  gibt lediglich eine untere Grenze an. BCH-Codes lassen

**Tabelle 7.2.** Generatorpolynome für einige  $(n, k)_2$ -BCH-Codes (nach [135])

| $n$ | $k$ | $t$ | $g(x)$ oktal                               |
|-----|-----|-----|--------------------------------------------|
| 7   | 4   | 1   | 13                                         |
| 15  | 11  | 1   | 23                                         |
|     | 7   | 2   | 721                                        |
|     | 5   | 3   | 2467                                       |
| 31  | 26  | 1   | 45                                         |
|     | 21  | 2   | 3551                                       |
|     | 16  | 3   | 107657                                     |
|     | 11  | 5   | 5423325                                    |
|     | 6   | 7   | 313365047                                  |
| 63  | 57  | 1   | 103                                        |
|     | 51  | 2   | 12471                                      |
|     | 45  | 3   | 1701317                                    |
|     | 39  | 4   | 166623567                                  |
|     | 36  | 5   | 1033500423                                 |
|     | 30  | 6   | 157464165547                               |
|     | 24  | 7   | 17323260404441                             |
|     | 18  | 10  | 1363026512351725                           |
|     | 127 | 120 | 1                                          |
|     | 113 | 2   | 41567                                      |
| 127 | 106 | 3   | 11554743                                   |
|     | 99  | 4   | 3447023271                                 |
|     | 92  | 5   | 624730022327                               |
|     | 85  | 6   | 130704476322273                            |
|     | 78  | 7   | 26230002166130115                          |
|     | 71  | 9   | 6255010713253127753                        |
|     | 64  | 10  | 1206534025570773100045                     |
|     | 57  | 11  | 335265252505705053517721                   |
|     | 50  | 13  | 54446512523314012421501421                 |
|     | 255 | 247 | 1                                          |
|     | 239 | 2   | 267543                                     |
|     | 231 | 3   | 156720665                                  |
|     | 223 | 4   | 75626641375                                |
|     | 215 | 5   | 23157564726421                             |
|     | 207 | 6   | 16176560567636227                          |
| 199 | 199 | 7   | 7633031270420722341                        |
|     | 191 | 8   | 2663470176115333714567                     |
|     | 187 | 9   | 52755313540001322236351                    |
|     | 179 | 10  | 22624710717340432416300455                 |
|     | 171 | 11  | 15416214212342356077061630637              |
|     | 163 | 12  | 7500415510075602551574724514601            |
|     | 155 | 13  | 3757513005407665015722506464677633         |
|     | 147 | 14  | 1642130173537165525304165305441011711      |
|     | 139 | 15  | 461401732060175561570722730247453567445    |
|     | 131 | 18  | 215713331471510151261250277442142024165471 |

Abbildung 4: Tabelle aus [Fr96]: Generatorpolynome für BCH-Codes

sich aufgrund ihrer klaren Struktur für gewisse Werte von  $n$  schlichtweg auflisten, wie dies in Tabelle 4 geschehen ist. Wir wollen zur Klarstellung der Notation in Tabelle 4 ein Beispiel betrachten. Nehmen wir an, wir wollten den  $(15, 5)$ -BCH-Code

bestimmen. Aus der Tabelle lesen wir das Generatorpolynom ab:

$$2467_{\text{octal}} = 010100110111_{\text{dual}} \leftrightarrow x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1. \quad (87)$$

Dementsprechend erhalten wir die Generatormatrix:

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (88)$$

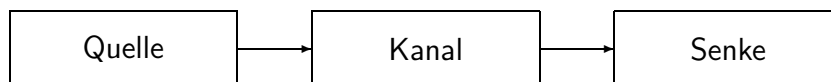
Durch elementare Zeilenumformungen läßt sich eine systematische Form erreichen:

$$G = \left( \begin{array}{cccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{array} \right) \quad (89)$$

## 5 Ziele und historische Entwicklung

### 5.1 Nachricht und Information

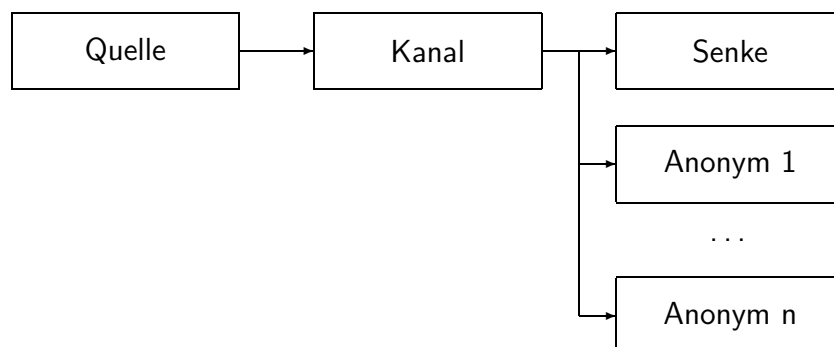
Um die Ziele der Kryptographie darstellen zu können, betrachten wir zunächst ein einfaches Nachrichtenübertragungssystem:



Dabei sollen die Quelle und die Senke über den gleichen Zeichenvorrat  $\mathcal{A}$  verfügen (etwa das Alphabet  $\{A, B, C, D, \dots\}$ ). Ziel ist es nun, eine Folge von Einzelzeichen aus  $\mathcal{A}$  über den Kanal von der Quelle zur Senke zu übertragen. Eine im allgemeinen endliche Folge von Zeichen aus  $\mathcal{A}$  wird als Nachricht bezeichnet. Eine Nachricht entsteht somit immer in der Quelle und hat das Ziel, der Senke etwas mitzuteilen. Daher besteht eine Nachricht aus einer Darstellungs- und einer Bedeutungskomponente. Die Bedeutungskomponente repräsentiert den Inhalt der Nachricht, während die Darstellungskomponente die Darstellung der Nachricht durch eine endliche Folge von Zeichen repräsentiert.

Von der Nachricht genau zu unterscheiden ist der Begriff der Information. Information entsteht immer in der Senke und zwar genau dann, wenn der Inhalt der Nachricht der Senke nicht vollständig bekannt war. In der Informationstheorie lassen sich die entsprechenden Begriffe mit wahrscheinlichkeitstheoretischen Hilfsmitteln exakt definieren.

Die Kryptographie beschäftigt sich in diesem Zusammenhang mit der Darstellung einer Nachricht, also nicht mit dem Inhalt und auch nicht mit der Information, die durch die Nachricht bei der Senke ankommt. Der Nachrichteninhalt ist vorgegeben und die Frage, ob diese Nachricht für die Senke einen Informationsgewinn bedeutet, ist in der Kryptographie irrelevant. Zur Verdeutlichung der Aufgabenstellung der Kryptographie betrachten wir ein erweitertes Nachrichtenübertragungssystem:



Nun wird davon ausgegangen, daß nicht nur die Senke die Nachricht erhält, sondern auch andere, im allgemeinen unbefugte Personen oder Institutionen die Nachricht (genauer: die endliche Folge von Zeichen) erhalten. Die Aufgabe der Kryptographie besteht darin, die Darstellung der Nachricht so zu wählen, daß Unbefugten die Bedeutung der Nachricht verschlossen bleibt. Die Kryptographie hat also *nicht* die Aufgabe zu verheimlichen, daß überhaupt eine Nachricht übertragen wird. Das ist das Ziel der Steganographie.

Eine Nachricht wird im folgenden als „Klartext“ bezeichnet. Die durch kryptographische Methoden verschlüsselte Nachricht (also die Nachricht, die tatsächlich übermittelt wird) wird „Geheimtext“ genannt.

## 5.2 Transposition und Substitution

Aus historischer Sicht sind zwei prinzipielle Vorgehensweisen bei der Verschlüsselung (Chiffrierung) von Nachrichten wichtig: die Transposition und die Substitution. Bei der Transposition wird einfach die Reihenfolge der einzelnen Zeichen nach einem festen Schema, das natürlich der Senke bekannt sein muß, geändert.

Kennt ein anonymer Empfänger dieses Schema nicht, so erhält er im allgemeinen eine Nachricht ohne Bedeutung. Die Transposition ist die älteste, durch literarische Quellen überlieferte Verschlüsselungstechnik. Nach dem Zeugnis des griechischen Historikers Plutarch benutzte die Regierung in Sparta die folgende Methode zur Übermittlung geheimer Nachrichten an ihre Generale:

Sender (Quelle) und Empfänger (Senke) mußten beide eine Skytale haben; das waren Zylinder mit genau dem gleichen Radius. Der Sender wickelte ein schmales Band aus Pergament spiralförmig um seinen Zylinder und schrieb dann die Nachricht der Länge nach auf das Band. War das Band abgewickelt, konnte die Nachricht nur von

einer Person gelesen werden, die einen Zylinder genau desselben Umfangs hatte - hoffentlich nur der Empfänger. Die Wirkung der Skytale läßt sich an folgendem Beispiel zeigen. Nehmen wir an, wir hätten einen Papierstreifen mit folgender Buchstabenfolge abgefangen:

SIEIYHITIEPICINLTEHSZDO!ETIEG RDGRR HAEKA ESZRP

Die Skytale des Senders hat einen Umfang, den wir durch die Anzahl  $u$  von Buchstaben ausdrücken können.

$u = 5$  :

SHP...  
III...  
ETC...  
III...  
YEN...

Man kann hier erkennen, daß aus den gegebenen Buchstabenkombinationen kein deutscher Text entstehen wird.

$u = 6$  :

SICHERHE  
ITISTDAS  
EINZIGES  
IELDERKR  
YPTOGRAP  
HIE!

Hier wurde  $u$  richtig gewählt; wie man oben sieht, lautet die Nachricht im Klartext: „*Sicherheit ist das einzige Ziel der Kryptographie!*“

Die Skytale ist der Prototyp eines Transpositionsalgorithmus. Die Buchstaben bleiben, was sie sind, aber nicht, wo sie sind. Im Gegensatz dazu wird bei der Substitution jedes Zeichen durch ein anderes Zeichen ersetzt.

Ein Spezialfall der Substitutionsmethode ist die von C. I. Caesar verwendete Verschiebetechnik.

ABCDEFGH...WXYZ  $\rightarrow$  Klartext  
DEFGHIJ...ZABC  $\rightarrow$  Geheimtext

Für die Ver- und Entschlüsselung muß man sich nur die Anzahl der Verschiebestellen (nach links oder rechts) merken.

Wenn man zur Verschlüsselung von Buchstaben Computer verwenden will, so identifiziert man möglicherweise „A“ mit 1, „B“ mit 2, ..., „Y“ mit 25 und „Z“ mit 0. Mit dieser Darstellung kann man die Verschiebechiffrierung besonders gut darstellen:

- Zunächst wird der Klartext in die entsprechende Zahl übersetzt;
- dann wird zu dieser Zahl die Zahl  $s$  addiert (entspricht einer Verschiebung um  $s$  Stellen);
- vom Ergebnis betrachtet man den Rest bei Division durch 26 (man rechnet *modulo* 26).
- Die so erhaltene Zahl repräsentiert den Geheimtextbuchstaben.

Beispiel: Caesar-Chiffre,  $s = 3$

| Klartext | entspr. Zahl | $+s$ | Ergebnis | <i>modulo</i> 26 | Geheimtext |
|----------|--------------|------|----------|------------------|------------|
| A        | 1            | 3    | 4        | 4                | D          |
| B        | 2            | 3    | 5        | 5                | E          |
| C        | 3            | 3    | 6        | 6                | F          |
| ⋮        | ⋮            | ⋮    | ⋮        | ⋮                | ⋮          |
| W        | 23           | 3    | 26       | 0                | Z          |
| X        | 24           | 3    | 27       | 1                | A          |
| Y        | 25           | 3    | 28       | 2                | B          |
| Z        | 0            | 3    | 3        | 3                | C          |

Bei der Dechiffrierung ist einfach 3 zu subtrahieren, wobei gilt:

$$2 - 3 = 25 \pmod{26}$$

$$1 - 3 = 24 \pmod{26}$$

$$0 - 3 = 23 \pmod{26}$$

Mit dieser Methode kann man quasi Buchstaben addieren. Es stellt sich nun die Frage, ob man auch Multiplizieren könnte. Dazu ein Beispiel. Multipliziert man zum Beispiel mit der Zahl 2, so erhält man:

| Klartext | entspr. Zahl | $\cdot t$ | Ergebnis | <i>modulo</i> 26 | Geheimtext |
|----------|--------------|-----------|----------|------------------|------------|
| A        | 1            | 2         | 2        | 2                | B          |
| $\vdots$ | $\vdots$     | $\vdots$  | $\vdots$ | $\vdots$         | $\vdots$   |
| H        | 8            | 2         | 16       | 16               | P          |
| $\vdots$ | $\vdots$     | $\vdots$  | $\vdots$ | $\vdots$         | $\vdots$   |
| U        | 21           | 2         | 42       | 16               | P          |

Beim Dechiffrieren könnte man dem Buchstaben „P“ im Geheimtext keinen eindeutigen Klartextbuchstaben zuordnen. Solche Verschlüsselungen sind unbrauchbar. Allerdings gibt es Faktoren  $t$ , die eine sinnvolle Chiffrierung liefern, nämlich  $t \in \{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$  (alle zu 26 teilerfremden natürlichen Zahlen). Dabei wird eine Chiffrierung als sinnvoll bezeichnet, falls jedem Buchstaben des Geheimtextes eindeutig ein Klartextbuchstabe zugeordnet werden kann.

Selbstverständlich kann man Multiplikation und Addition verknüpfen und erhält somit eine von  $12 \cdot 26 = 312$  möglichen Tauschchiffren, gegeben durch

$$t \in \{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\} \text{ und } s \in \{0, \dots, 25\}.$$

Die Verschlüsselung ist durch die Chiffrierabbildung

$$f : \{0, \dots, 25\} \rightarrow \{0, \dots, 25\}, x \mapsto (x \cdot t) + s \pmod{26}$$

gegeben.

Beispiel:  $t = 3$ ,  $s = 5$

| Klartext | entspr. Zahl | Ergebnis            | Geheimtext |
|----------|--------------|---------------------|------------|
| A        | 1            | 8                   | H          |
| B        | 2            | 11                  | K          |
| $\vdots$ | $\vdots$     | $\vdots$            | $\vdots$   |
| L        | 12           | $41 \rightarrow 15$ | O          |
| $\vdots$ | $\vdots$     | $\vdots$            | $\vdots$   |

Die Senke hat nun mit Wissen der beiden Zahlen  $(s, t)$  die Verschlüsselung rückgängig zu machen, also die Chiffrierabbildung  $f$  umzukehren (z. B. aus 15 (O) wird 12 (L)).

Es gilt somit  $(15 - 5) \div 3 = 12 \pmod{26}$ , denn  $((12 \cdot 3) + 5) \div 26 = 1 \text{ Rest } 15$ .

Das Zahlenpaar  $(s, t)$  wird als Schlüssel bezeichnet und repräsentiert die Information, die zum Entschlüsseln von Tauschchiffren nötig ist. Es gibt für das deutsche Alphabet 312 verschiedene Schlüssel.

Die entscheidende Frage: Sind diese 312 Tauschchiffren gute Chiffren? Die Antwort liefert das in der Kryptographie grundlegende **Prinzip von Kerckhoff**:

Die Sicherheit (und damit die Güte) einer Verschlüsselungstechnik darf nicht von der Geheimhaltung der Art der Verschlüsselung abhängen (z. B. Transposition, Tauschchiffren, ...), sondern nur von der Geheimhaltung des Schlüssels.

Da es bei den Tauschchiffren nur 312 verschiedene Schlüssel für das deutsche Alphabet gibt, ist diese Art der Verschlüsselung sehr unsicher.

### 5.3 Kryptoanalyse monoalphabetischer Chiffren

Eine Chiffrierung heißt monoalphabetisch, falls jeder Buchstabe des Alphabets stets zu demselben (Geheimtext-) Buchstaben verschlüsselt wird. Eine monoalphabetische Chiffrierung kann man immer so darstellen, da man unter das Klartextalphabet das Geheimtextalphabet schreibt. Diese Alphabete müssen nicht übereinstimmen:

|            |   |   |   |   |   |     |
|------------|---|---|---|---|---|-----|
| Klartext   | A | B | C | D | E | ... |
| Geheimtext | ℵ | ∫ | ∂ | ℜ | © | ... |

Die Substitutionsalgorithmen beschreiben monoalphabetische Chiffren. Während sich die Kryptographie mit der Verschlüsselung von Nachrichten beschäftigt, besteht die Aufgabe der Kryptoanalyse im (vom Sender unerwünschten) Brechen der Chiffrierung. Beide - die Kryptographie und die Kryptoanalyse - werden unter dem Oberbegriff Kryptologie zusammengefaßt. Da eine Nachricht im allgemeinen in einer Sprache formuliert wird und da in keiner Sprache jeder Buchstabe gleich häufig vorkommt, sind monoalphabetische Chiffren leicht zu brechen.

Für das Deutsche gilt folgende Statistik:

| Buchstabe | Häufigkeit in % | Buchstabe | Häufigkeit in % |
|-----------|-----------------|-----------|-----------------|
| A         | 6,51            | N         | 9,78            |
| B         | 1,89            | O         | 2,51            |
| C         | 3,06            | P         | 0,79            |
| D         | 5,08            | Q         | 0,02            |
| E         | 17,4            | R         | 7,00            |
| F         | 1,66            | S         | 7,27            |
| G         | 3,01            | T         | 6,15            |
| H         | 4,76            | U         | 4,35            |
| I         | 7,55            | V         | 0,67            |
| J         | 0,27            | W         | 1,89            |
| K         | 1,21            | X         | 0,03            |
| L         | 3,44            | Y         | 0,04            |
| M         | 2,53            | Z         | 1,13            |

Für Buchstabenpaare gilt:

| Buchstabenpaar | Häufigkeit in % | Buchstabenpaar | Häufigkeit in % |
|----------------|-----------------|----------------|-----------------|
| en             | 3,88            | er             | 3,75            |
| ch             | 2,75            | te             | 2,26            |
| de             | 2,00            | nd             | 1,99            |
| ei             | 1,88            | ie             | 1,79            |
| in             | 1,67            | es             | 1,52            |
| ⋮              | ⋮               | ⋮              | ⋮               |

Mit Statistiken dieser Art hat der Kryptoanalytiker keine Probleme, mit dem Rechner in Deutsch verfasste Klartexte, die monoalphabetisch chiffriert werden, aus dem Geheimtext zu rekonstruieren, obwohl es für das deutsche Alphabet  $26! \approx 4 \cdot 10^{26}$  verschiedene Chiffrierungen gibt.

Beispiel:

MRNBNA CNGC RBC WRLQC VNQA PNQNRV

(i) Setze  $E \Rightarrow N$ .

MRNBNA CNGC RBC WRLQC VNQA PNQNRV  
 ..E.E. .E.. ... ..E.. .E.E..

(ii) Teste alle Möglichkeiten mit R, C, Q ersetzt durch A, D, H, I, N, R, S, T, U ( $\geq 4\%$ ).

MRNBNA CNGC RBC WRLQC VNQA PNQNRV  
 .IE.E. TE.T I.T .I.HT .EH. .EHEI.

(iii) Vervollständige zu sinnvollen deutschen Wörtern, d. h. setze  $B \Rightarrow S$ ,  $M \Rightarrow D$ ,  $G \Rightarrow X$ .

MRNBNA CNGC RBC WRLQC VNQA PNQNRV  
 DIESE. TEXT IST .I.HT .EH. .EHEI.

(iv) ...

(v) Als Klartext ergibt sich schließlich: DIESER TEXT IST NICHT MEHR GEHEIM.

## 5.4 Perfekte Sicherheit

## 5.5 Kryptosysteme

Bei der Frage der Sicherheit von Chiffren benötigt man den Begriff des Kryptosystems.

### 5.5.1 Definition Kryptosystem

Ein Kryptosystem ist ein Tupel  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$  bestehend aus fünf Mengen.

- $\mathcal{P}$  heißt **Klartextrraum**. In dieser Menge werden alle in Betracht kommenden Klartexte (Nachrichten) gesammelt (z. B. alle möglichen endlichen Folgen von Buchstaben aus dem deutschen Alphabet, die aus Wörtern der deutschen Sprache gebildet sind und aus weniger als 1000 Zeichen bestehen).
- $\mathcal{C}$  heißt **Geheimtextraum**: in dieser Menge werden alle in Frage kommenden Geheimtexte gesammelt:  $\mathcal{C} := \{e_k(p), k \in \mathcal{K}, p \in \mathcal{P}\}$ .
- $\mathcal{K}$  heißt **Schlüsselraum**. Darin sind alle zu betrachtenden Schlüssel gesammelt.
- $\mathcal{E}$  ist eine Menge  $\{e_k : \mathcal{P} \rightarrow \mathcal{C}; k \in \mathcal{K}\}$  von **Verschlüsselungsfunktionen**.
- $\mathcal{D}$  ist eine Menge  $\{d_k : \mathcal{C} \rightarrow \mathcal{P}; k \in \mathcal{K}\}$  von **Entschlüsselungsfunktionen** derart, daß es zu jedem Schlüssel  $k_1 \in \mathcal{K}$  ein  $k_2 \in \mathcal{K}$  gibt mit  $d_{k_2}(e_{k_1}(p)) = p \forall p \in \mathcal{P}$ . Mit anderen Worten: zu jedem Schlüssel  $k_1 \in \mathcal{K}$  muß es einen anderen Schlüssel  $k_2 \in \mathcal{K}$  geben, so daß die mit  $k_1$  verschlüsselte Nachricht mit  $k_2$  wieder entschlüsselt werden kann. Ist  $k_1 = k_2$ , so nennt man das Kryptosystem „symmetrisch“, ansonsten „asymmetrisch“.

Betrachten wir ein Beispiel:

Sei  $\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0, \dots, 25\}$  und  $\mathcal{E} = \{e_k : \mathcal{P} \rightarrow \mathcal{C}; x + k \pmod{26}\}$ ,  $\mathcal{D} = \{d_k : \mathcal{C} \rightarrow \mathcal{P}; x - k \pmod{26}\}$ , so ist dies gerade die Verschiebechiffre. Es gilt  $d_k(e_k(p)) = ((p + k) \pmod{26} - k) \pmod{26} = p$ , also ist  $k_1 = k_2$ , d. h. bei diesem Kryptosystem muß der Schlüssel geheimgehalten werden.

## 5.6 Perfekte Sicherheit

Wir untersuchen die Frage, wann es sinnvoll ist, ein gegebenes Kryptosystem als perfekt sicher zu bezeichnen. Bei der Kryptoanalyse der monoalphabetischen Chiffrierung wurde trotz einer sehr großen Zahl von Schlüsseln klar, daß diese Chiffrierung unsicher ist, denn die Struktur der deutschen Sprache, die natürlich im Klartext zum Ausdruck kommt, wurde auf den Geheimtext übertragen. Dadurch waren Rückschlüsse vom Geheimtext auf den Klartext möglich. Um nun Abhängigkeiten der inneren Struktur der

Klar- und Geheimtexte ausschließen zu können, benötigt man ein formales Werkzeug. Dieses Werkzeug ist die Wahrscheinlichkeitstheorie. Dabei gehen wir von gegebenen Wahrscheinlichkeitsverteilungen

$$\mathbf{P}_{\mathcal{P}} : \mathcal{P} \rightarrow [0, 1], p \mapsto \mathbf{P}_{\mathcal{P}}(p) > 0$$

und

$$\mathbf{P}_{\mathcal{K}} : \mathcal{K} \rightarrow [0, 1], k \mapsto \mathbf{P}_{\mathcal{K}}(k) > 0$$

aus.  $\mathbf{P}_{\mathcal{P}}(p)$  ist also die Wahrscheinlichkeit, daß der Klartext  $p$  übertragen werden soll, und  $\mathbf{P}_{\mathcal{K}}(k)$  ist die Wahrscheinlichkeit dafür, daß für die Übertragung der Schlüssel  $k$  verwendet wird. Ferner nehmen wir an, daß die Wahl des Schlüssels nicht vom Klartext abhängt, d. h. wir haben eine Wahrscheinlichkeitsverteilung

$$\mathbf{P}_{\mathcal{P} \times \mathcal{K}} : \mathcal{P} \times \mathcal{K} \rightarrow [0, 1], (p, k) \mapsto \mathbf{P}_{\mathcal{P} \times \mathcal{K}}(p, k) = \mathbf{P}_{\mathcal{P}}(p) \cdot \mathbf{P}_{\mathcal{K}}(k) \quad \forall p \in \mathcal{P}, k \in \mathcal{K}$$

(stochastische Unabhängigkeit).  $\mathbf{P}_{\mathcal{P} \times \mathcal{K}}(p, k)$  ist also die Wahrscheinlichkeit, daß der Klartext  $p$  mit dem Schlüssel  $k$  übertragen wird.

Nehmen wir an, wir haben den Geheimtext  $c \in \mathcal{C}$  abgehört, wir kennen das verwendete Kryptosystem und wir kennen die Wahrscheinlichkeitsverteilungen  $\mathbf{P}_{\mathcal{P}}, \mathbf{P}_{\mathcal{K}}$ . Unsere Aufgabe besteht nun darin, das zu  $c \in \mathcal{C}$  gehörige  $p \in \mathcal{P}$  zu finden. Hätten wir  $c$  nicht, so stünde nur  $\mathbf{P}_{\mathcal{P}}(p)$  und  $\mathbf{P}_{\mathcal{K}}(k)$  zur Verfügung. Wenn es möglich ist, mit Hilfe von  $c$  eine neue Wahrscheinlichkeitsverteilung

$$\mathbf{P}_{\mathcal{P}|c} : \mathcal{P} \rightarrow [0, 1]$$

auf  $\mathcal{P}$  anzugeben, die nicht gleich  $\mathbf{P}_{\mathcal{P}}$  ist und somit die Aufgabe erleichtert, das zu  $c$  gehörige  $p$  zu finden, dann hat uns das Abhören von  $c$  geholfen, ansonsten nicht. Aufgrund von  $\mathbf{P}_{\mathcal{P}}, \mathbf{P}_{\mathcal{K}}$  und der stochastischen Unabhängigkeit sind wir in der Lage, eine neue Wahrscheinlichkeitsverteilung auf  $\mathcal{C}$  anzugeben:

$$\mathbf{P}_{\mathcal{C}} : \mathcal{C} \rightarrow [0, 1], \xi \mapsto \sum_{p \in \mathcal{P}, k \in \mathcal{K}, e_k(p) = \xi} \mathbf{P}_{\mathcal{P}}(p) \cdot \mathbf{P}_{\mathcal{K}}(k)$$

Summiert werden hier also alle  $p \in \mathcal{P}$  und  $k \in \mathcal{K}$ , für die die Bedingung  $e_k(p) = \xi$  gilt.

Ferner können wir eine Wahrscheinlichkeitsverteilung  $\mathbf{P}_{\mathcal{C}|p}$  auf  $\mathcal{C}$  angeben, die davon ausgeht, daß der Klartext  $p \in \mathcal{P}$  gesendet wurde.

$$\mathbf{P}_{\mathcal{C}|p} : \mathcal{C} \rightarrow [0, 1], \xi \mapsto \sum_{k \in \mathcal{K}, e_k(p) = \xi} \mathbf{P}_{\mathcal{K}}(k)$$

Mit diesen Zutaten ist es nun unter Verwendung des **Satzes von Bayes**<sup>1</sup> möglich,  $\mathbf{P}_{\mathcal{P}|c}$  zu berechnen:

$$\mathbf{P}_{\mathcal{P}|c} : \mathcal{P} \rightarrow [0, 1], p \mapsto \frac{\mathbf{P}_{\mathcal{C}|p}(c)}{\mathbf{P}_{\mathcal{C}}(c)} \cdot \mathbf{P}_{\mathcal{P}}(p)$$

Ist somit der Faktor  $\frac{\mathbf{P}_{\mathcal{C}|p}(c)}{\mathbf{P}_{\mathcal{C}}(c)}$  gleich 1, so hat sich unser Wissen über die Wahrscheinlichkeitsverteilung durch das Abhören von  $c$  nicht verbessert. Ein solches Kryptosystem bezeichnen wir im folgenden als „perfekt sicher“.

### 5.6.1 Definition perfekt sicherer Kryptosysteme

Sei  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$  ein Kryptosystem mit den entsprechenden Wahrscheinlichkeitsverteilungen  $\mathbf{P}_{\mathcal{P}}, \mathbf{P}_{\mathcal{K}}$  und der stochastischen Unabhängigkeit von Klartext- und Schlüsselwahl. Dieses Kryptosystem heißt perfekt sicher, falls

$$\mathbf{P}_{\mathcal{P}}(p) = \mathbf{P}_{\mathcal{P}|c}(p) \quad \forall p \in \mathcal{P}, c \in \mathcal{C}$$

Das in der obigen Definition für perfekte Sicherheit gegebene Kryptosystem ist im allgemeinen nicht praktikabel (aber suggestiv), da in dieser Form kaum überprüfbar. Daher ist der folgende Satz von Claude Shannon von zentraler Bedeutung.

### 5.6.2 Kriterium für perfekte Sicherheit von Claude Shannon

Sei  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$  ein Kryptosystem und die Anzahl der Elemente von  $\mathcal{C}$  gleich der Anzahl der Elemente von  $\mathcal{K}$  und  $\mathcal{P}$ . (in Zeichen:  $|\mathcal{C}| = |\mathcal{K}| = |\mathcal{P}|$ ). Das Kryptosystem ist genau dann perfekt sicher, wenn

$$\mathbf{P}_{\mathcal{K}}(k) = \frac{1}{|\mathcal{K}|} \quad \forall k \in \mathcal{K}$$

(Gleichverteilung) und wenn es zu jedem  $p \in \mathcal{P}$  und  $c \in \mathcal{C}$  genau einen Schlüssel  $k \in \mathcal{K}$  mit  $e_k(p) = c$  gibt.

Entscheidend an diesem Satz ist die „genau dann“-Aussage, d. h.

- gilt für ein Kryptosystem  $|\mathcal{C}| = |\mathcal{K}| = |\mathcal{P}|, \mathbf{P}_{\mathcal{K}}(k) = \frac{1}{|\mathcal{K}|}$  und existiert zu jedem  $p \in \mathcal{P}, c \in \mathcal{C}$  genau ein  $k \in \mathcal{K}$  mit  $e_k(p) = c$ , so ist dieses Kryptosystem perfekt sicher.

---

<sup>1</sup>Beispiel zum Satz von Bayes:

gegeben seien die Ereignisse  $F$ : „Person ist eine Frau“ und  $R$ : „Person ist Raucher“. Die Wahrscheinlichkeiten  $\mathbf{P}(F), \mathbf{P}(R), \mathbf{P}(R|F)$  seien bekannt. Die Wahrscheinlichkeit  $\mathbf{P}(F|R)$  läßt sich folgendermaßen berechnen:  $\mathbf{P}(F|R) = \frac{\mathbf{P}(R|F)}{\mathbf{P}(R)} \cdot \mathbf{P}(F)$ .

- ist das Tupel  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$  ein perfekt sicheres Kryptosystem mit  $|\mathcal{C}| = |\mathcal{K}| = |\mathcal{P}|$ , so gilt:  $\mathbf{P}_{\mathcal{K}}(k) = \frac{1}{|\mathcal{K}|}$  und zu jedem  $p \in \mathcal{P}, c \in \mathcal{C}$  gibt es genau ein  $k \in \mathcal{K}$  mit  $e_k(p) = c$ .

Das bekannteste Kryptosystem, dessen perfekte Sicherheit man mit Satz 2.2.2 beweisen kann, ist das 1917 von Gilbert Vernam erfundene und patentierte **One-Time-Pad**.

Dabei betrachtet man Bitstrings der Länge  $n$ , also  $\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0, 1\}^n, n \in \mathbb{N}$ . Die Verschlüsselungsfunktion  $e_k$  zum Schlüssel  $k \in \{0, 1\}^n$  ist gegeben durch

$$e_k : \{0, 1\}^n \rightarrow \{0, 1\}^n, (p_1, \dots, p_n) \mapsto (p_1 \oplus k_1, \dots, p_n \oplus k_n)$$

(Addition ohne Übertrag). Will man einen Klartext  $p$  verschlüsseln, so wählt man gemäß einer Gleichverteilung einen Schlüssel  $k \in \{0, 1\}^n$  und berechnet  $c = e_k(p)$ . Da es zu jedem  $p \in \{0, 1\}^n, c \in \{0, 1\}^n$  genau einen Schlüssel  $k \in \{0, 1\}^n$  gibt mit  $c_i = p_i \oplus k_i, i = 1, \dots, n$ , nämlich  $k_i = p_i \oplus c_i$ , ist dieses Kryptosystem perfekt sicher. Da auch  $p_i = c_i \oplus k_i, i = 1, \dots, n$  gilt, ist dieses Kryptosystem zudem symmetrisch. Die perfekte Sicherheit ist dadurch erkauft, daß zu jeder Übertragung eines Klartextes ein neuer gemäß Gleichverteilung erzeugter Schlüssel zu verwenden ist. Dieser Schlüssel muß der Senke bekannt sein. Entweder legt man die Schlüssel a priori fest, dann muß man sie speichern und geheimhalten, oder man überträgt den Schlüssel gesondert.

## 6 Mathematische Grundlagen

Die modernen Verfahren der Kryptographie nutzen im Wesentlichen Strukturen der diskreten Mathematik. Die mathematischen Gebiete der Algebra und der Zahlentheorie finden somit in der Kryptographie eine Anwendung. Algebra beschäftigt sich mit der Theorie algebraischer Strukturen, also mit Mengen und Verknüpfungen und gewissen Axiomen, die diese Verknüpfungen betreffen. Die Zahlentheorie hat die Menge der ganzen Zahlen  $\mathbb{Z}$  und ihre Strukturen zum Gegenstand. In diesem Kapitel werden einige grundlegende Tatsachen aus der Algebra und der Zahlentheorie zusammengestellt, die für die Kryptographie gebraucht werden.

### 6.1 Gruppen, erzeugende Elemente

Ein zentraler Begriff der Algebra ist der Begriff einer Gruppe:

**Definition 6.1.** Es sei  $G \neq \emptyset$  eine Menge und  $\circ : G \times G \rightarrow G$  eine Verknüpfung.  $(G, \circ)$  heißt Gruppe, wenn die folgenden Bedingungen erfüllt sind:

- (i)  $\circ$  ist assoziativ, d.h. für  $a, b, c \in G$  gilt stets  $(a \circ b) \circ c = a \circ (b \circ c)$ .
- (ii) Es gibt ein neutrales Element  $e \in G$  mit  $e \circ a = a \circ e = a$  für alle  $a \in G$ .
- (iii) Zu jedem  $a \in G$  gibt es ein  $a^{-1} \in G$  mit  $a \circ a^{-1} = a^{-1} \circ a = e$ ;  $a^{-1}$  heißt inverses Element.

Gilt zusätzlich für alle  $a, b \in G$   $a \circ b = b \circ a$ , so heißt die Gruppe  $G$  kommutativ oder abelsch.

Eine Menge  $H \neq \emptyset$  mit einer Verknüpfung  $\circ$  bezeichnet man als Halbgruppe, falls  $\circ$  assoziativ ist.  $(H, \circ)$  kann ein neutrales Element  $e$  enthalten. Wegen  $e \circ e = e$  enthält  $H$  dann auch ein zu  $e$  inverses Element. Allgemein heißt ein Element  $x \in H$  Einheit, wenn  $H$  ein neutrales Element  $e$  und ein zu  $x$  inverses Element  $y$  enthält. Die Einheiten einer Halbgruppe  $H$  bilden sogar eine Gruppe  $E(H)$ . Wegen  $e \in E(H)$  gilt  $E(H) \neq \emptyset$ .

Die Verknüpfung  $\circ$  einer Gruppe wird meist als Addition  $+$  oder Multiplikation  $\cdot$  geschrieben. Beispiele kommutativer Gruppen sind  $(\mathbb{R}, +)$ ,  $(\mathbb{R} \setminus \{0\}, \cdot)$  und  $(\mathbb{Z}, +)$ .  $(\mathbb{Z}, \cdot)$  ist eine kommutative Halbgruppe mit neutralem Element 1, aber keine Gruppe.

Eine Teilmenge  $U$  einer Gruppe  $(G, \circ)$  heißt Untergruppe von  $G$ , falls  $U$  mit der auf  $U$  eingeschränkten Verknüpfung  $\circ$  selbst eine Gruppe ist. Man schreibt dann  $U < G$ . Ein leicht überprüfbares Kriterium, ob eine gegebene Teilmenge  $U \neq \emptyset$  von  $G$  eine Untergruppe ist, ist das folgende:

$$U < G \iff a, b \in U \Rightarrow a \circ b^{-1} \in U.$$

Offensichtlich sind also  $G$  und  $\{e\}$  Untergruppen von  $(G, \circ)$ . Die Menge aller Untergruppen wird mit  $\mathcal{U}(G)$  bezeichnet. Mit zwei Untergruppen  $U, V \in \mathcal{U}(G)$  ist auch ihr Durchschnitt eine Untergruppe von  $G$ :

$$U, V \in \mathcal{U}(G) \Rightarrow U \cap V \in \mathcal{U}(G).$$

Damit ist die folgende Definition sinnvoll:

**Definition 6.2.** Seien  $G$  eine Gruppe und  $X \subseteq G$ . Der Durchschnitt aller Untergruppen von  $G$ , die  $X$  enthalten, wird als die von  $X$  erzeugte Untergruppe  $\langle X \rangle$  bezeichnet:

$$\langle X \rangle = \bigcap_{\substack{U \in \mathcal{U}(G) \\ X \subseteq U}} U.$$

Eine Gruppe  $G$  heißt zyklisch, falls  $G$  von einem Element erzeugt wird, falls also ein  $x \in G$  existiert mit  $\langle x \rangle = G$ .

Es sei nun  $(H, \cdot)$  eine multiplikative Halbgruppe mit neutralem Element  $e$ . Dann definiert man die Potenz  $a^n$  eines Elementes  $a \in H$  für alle  $n \in \mathbb{N}$  rekursiv durch

$$a^0 := e, \quad a^n := a \cdot a^{n-1}.$$

Entsprechend definiert man für eine additive Halbgruppe  $(H, +)$  das Vielfache  $na$  für alle  $n \in \mathbb{N}$ :

$$1a := a, \quad na := (n-1)a + a.$$

Sei  $G$  eine endliche Gruppe. Für die von einem Element  $x \in G$  erzeugte Untergruppe in der Gruppe  $G$  gilt in multiplikativer Schreibweise

$$\langle x \rangle = \{x^n | n \in \mathbb{N}\}$$

und in additiver Schreibweise

$$\langle x \rangle = \{nx | n \in \mathbb{N}\}.$$

Damit ist es leicht einzusehen, daß jede zyklische Gruppe kommutativ ist.

## 6.2 Rechnen in Restklassengruppen

Ein wichtiges Beispiel einer zyklischen Gruppe ist die Menge  $\mathbb{Z}$  der ganzen Zahlen mit der üblichen Addition. Es gilt

$$(\mathbb{Z}, +) = \langle 1 \rangle.$$

Zu jedem  $n \in \mathbb{N}_0$  gibt es eine von  $n$  erzeugte Untergruppe:

$$n\mathbb{Z} := \{nz | z \in \mathbb{Z}\}.$$

Es kann gezeigt werden, daß dies bereits alle Untergruppen von  $(\mathbb{Z}, +)$  sind. Mit diesen Untergruppen definiert man eine neue Verknüpfung. Für  $a, b \in \mathbb{Z}$  gelte

$$a \equiv b \pmod{n} : \Longleftrightarrow a - b \in n\mathbb{Z}.$$

Die rechte Seite besagt, daß es eine ganze Zahl  $z \in \mathbb{Z}$  gibt mit  $a - b = nz$ ; damit folgt, daß  $n$  die Differenz  $a - b$  teilt, man schreibt auch  $n | a - b$ . Da  $\equiv$  mit der Addition  $+$  verträglich ist, kann gezeigt werden, daß

$$\mathbb{Z}_n := \{n\mathbb{Z}, 1 + n\mathbb{Z}, \dots, (n-1) + n\mathbb{Z}\}$$

eine Gruppe ist mit dem neutralen Element  $n\mathbb{Z}$ , die Restklassengruppe modulo  $n$ . Dabei ist für  $a \in \{1, \dots, n-1\}$  die Nebenklasse  $a + n\mathbb{Z}$  definiert als  $\{a + nz | z \in \mathbb{Z}\}$ .

$\mathbb{Z}_n$  enthält  $n$  Elemente und wird von  $1+n\mathbb{Z}$  erzeugt. Insbesondere ist  $\mathbb{Z}_n$  eine zyklische Gruppe. Häufig wird  $\mathbb{Z}_n$  kürzer dargestellt als

$$\mathbb{Z}_n = \{0, 1, 2, \dots, n-1\}.$$

Dies sind die kleinsten nichtnegativen Reste modulo  $n$ . Für nichtnegative ganze Zahlen  $m \in \mathbb{N}_0$  erhält man den zugehörigen kleinsten Rest modulo  $n$  durch ganzzahlige Division durch  $n$ . Um einer negativen ganzen Zahl  $-m, m \in \mathbb{N}$ , ihren kleinsten nichtnegativen Rest modulo  $n$  zuzuordnen, kann man zunächst ein geeignetes Vielfaches von  $n$  addieren, so daß eine positive ganze Zahl entsteht, und anschließend durch  $n$  dividieren.

Es läßt sich nun zeigen, daß  $\equiv$  auch mit der Multiplikation  $\cdot$  von  $\mathbb{Z}$  verträglich ist, daß also durch

$$(a + n\mathbb{Z}) \cdot (b + n\mathbb{Z}) := ab + n\mathbb{Z}$$

eine innere Verknüpfung in  $\mathbb{Z}_n$  definiert wird.  $(\mathbb{Z}_n, \cdot)$  ist eine kommutative Halbgruppe mit dem neutralen Element  $1 + n\mathbb{Z}$ . Die Einheitengruppe von  $(\mathbb{Z}_n, \cdot)$  ist

$$E(\mathbb{Z}_n) = \{a + n\mathbb{Z} \mid \text{ggT}(a, n) = 1\}.$$

Sie wird als prime Restklassengruppe modulo  $n$  bezeichnet; man schreibt  $E(\mathbb{Z}_n) = \mathbb{Z}_n^*$ . Für die Anzahl der Elemente in  $\mathbb{Z}_n^*$  gilt also

$$|\mathbb{Z}_n^*| = |\{k \in \mathbb{N} \mid k < n, \text{ggT}(n, k) = 1\}| =: \varphi(n).$$

Dabei heißt die Funktion  $\varphi : \mathbb{N} \rightarrow \mathbb{N}$  Eulersche  $\varphi$ -Funktion.

**Beispiel 6.3.** Für eine Primzahl  $p \in \mathbb{N}$  läßt sich die Eulersche  $\varphi$ -Funktion leicht auswerten. Offensichtlich gilt  $\varphi(p) = p - 1$ . Für das Produkt zweier Primzahlen  $p, q \in \mathbb{N}$  gilt  $\varphi(pq) = (p - 1)(q - 1)$ .

### 6.3 Der größte gemeinsame Teiler

Es seien  $a, b \in \mathbb{Z} \setminus \{0\}$ . Da die Untergruppen von  $(\mathbb{Z}, +)$  bekannt sind, weiß man, daß es zu der von  $a, b$  erzeugten Untergruppe ein  $d \in \mathbb{N}$  gibt mit  $\langle a, b \rangle = d\mathbb{Z}$ . Da außerdem  $a$  und  $b$  in  $\langle a, b \rangle$  enthalten sind, folgt  $d \mid a, b$ , d. h.  $d$  ist ein gemeinsamer Teiler von  $a$  und  $b$ . Es gilt

$$d\mathbb{Z} = \langle a, b \rangle = \{ra + sb \mid r, s \in \mathbb{Z}\}.$$

Also gibt es Zahlen  $x, y \in \mathbb{Z}$  für die gilt  $d = xa + yb$ . Für alle  $t \in \mathbb{Z}$  mit  $t \mid a, b$  gilt somit  $t \mid d$ , d. h.  $d$  ist der größte gemeinsame Teiler von  $a$  und  $b$ ; man schreibt  $d = \text{ggT}(a, b)$ . Es stellt sich nun die Frage, ob man den größten gemeinsamen Teiler zweier ganzer Zahlen leicht berechnen kann. Eine Antwort ist der euklidische Algorithmus (Euklid von Alexandria, um 365-300 v. Chr.).

**Proposition 1.** (Euklidischer Algorithmus) Seien  $a, b \in \mathbb{Z}, b \neq 0$ . Mit  $r_1 := a, r_2 := b$  bricht die Division mit Rest

$$r_i := q_i r_{i+1} + r_{i+2}$$

mit  $|r_{i+2}| < |r_{i+1}|$  falls  $r_{i+2} \neq 0$  nach höchstens  $n \leq |b|$  Schritten ab, d. h. es gibt ein  $n \leq |b|$  mit  $r_{i+1} \neq 0$  für  $i < n$  und  $r_n = q_n r_{n+1} + 0$ . Dann gilt:  $d := r_{n+1} = \text{ggT}(a, b)$  und es können  $x, y \in \mathbb{Z}$  bestimmt werden mit  $d = xa + yb$ .

Zur Begründung überlegt man sich, daß folgendes gilt:

$$\left. \begin{array}{l} r_{n+1} | r_n \\ r_{n-1} = q_{n-1} r_n + r_{n+1} \end{array} \right\} \Rightarrow r_{n+1} | r_{n-1} \Rightarrow \dots \Rightarrow r_{n+1} | b \Rightarrow r_{n+1} | a.$$

Die Schranke für die Anzahl der Iterationen kann wesentlich verbessert werden: Seien dazu  $a > b > 0$  und  $\theta := \frac{1+\sqrt{5}}{2}$ . Dann ist die Anzahl der Iterationen im euklidischen Algorithmus höchstens  $\frac{\log b}{\log \theta} + 1$ . Den Ablauf des Algorithmus verdeutlicht man sich am besten anhand eines Beispiels.

**Beispiel 6.4.** Gesucht ist  $d = \text{ggT}(36, 21)$ . Der euklidische Algorithmus liefert

$$\begin{array}{rcl} 36 & = & 1 \cdot 21 + 15 \\ 21 & = & 1 \cdot 15 + 6 \\ 15 & = & 2 \cdot 6 + 3 \\ 6 & = & 2 \cdot 3 \end{array} \qquad \begin{array}{rcl} 15 & = & 36 - 21 \\ 6 & = & 21 - 15 \\ 3 & = & 15 - 2 \cdot 6. \end{array}$$

Somit gilt  $3 = \text{ggT}(36, 21)$  und  $3 = 36 - 21 - 2(21 - (36 - 21)) = 3 \cdot 36 - 5 \cdot 21$ .

Im letzten Abschnitt wurde die Einheitengruppe von  $\mathbb{Z}_n$  angegeben, die prime Restklassengruppe modulo  $n$ . Um zu einem gegebenen Element aus  $\mathbb{Z}_n^*$  ein inverses Element zu bestimmen, kann der euklidische Algorithmus verwendet werden. Sei also  $c \in \mathbb{Z}_n^*$ , d. h.  $\text{ggT}(c, n) = 1$ . Dann können Elemente  $d$  und  $m$  bestimmt werden mit  $1 = cd + mn$ ; folglich gilt  $cd \equiv 1 \pmod{n}$ , d. h.  $d$  ist in  $\mathbb{Z}_n^*$  das zu  $c$  inverse Element.

## 6.4 Primzahlen

Eine ganze Zahl  $p \in \mathbb{Z}$  heißt Primzahl, falls aus  $p = ab$  für  $a, b \in \mathbb{Z}$  stets folgt  $a \in E(\mathbb{Z}, \cdot) = \{1, -1\}$  oder  $b \in E(\mathbb{Z}, \cdot)$ . Eine Primzahl ist in diesem Sinne also unzerlegbar. Primzahlen spielen in  $\mathbb{Z}$  eine fundamentale Rolle, wie der folgende Satz zeigt.

**Proposition 2.** Es sei  $a \in \mathbb{Z} \setminus \{1, -1, 0\}$ . Dann gibt es Primzahlen  $p_1, \dots, p_n$  und eine Einheit  $e$  mit  $a = ep_1 \cdot \dots \cdot p_n$ . Diese Produktdarstellung ist bis auf Einheiten und die Reihenfolge der Faktoren eindeutig.

Die Existenz der Produktdarstellung liefert noch kein effizientes Verfahren, wie man zu einer gegebenen ganzen Zahl ihre Primfaktoren bestimmt. Tatsächlich ist ein solches Verfahren bis heute nicht bekannt und im nächsten Kapitel wird ein kryptographisches Verfahren vorgestellt, das auf der Schwierigkeit des Faktorisierungsproblems beruht.

## 6.5 Körper

Bisher wurden Gruppen betrachtet, also Mengen mit einer inneren Verknüpfung, die meist als Addition oder Multiplikation interpretiert wurde. Körper sind Mengen mit zwei inneren Verknüpfungen.

**Definition 6.5.** Eine Menge  $(K, +, \cdot)$  mit zwei inneren Verknüpfungen  $+$  und  $\cdot$  heißt Körper, falls die folgenden Bedingungen erfüllt sind:

- (i)  $(K, +)$  ist eine abelsche Gruppe (mit neutralem Element 0).
- (ii)  $(K \setminus \{0\}, \cdot)$  ist eine abelsche Gruppe (mit neutralem Element 1).
- (iii) Für alle  $x, y, z \in K$  gilt das Distributivgesetz  $x \cdot (y + z) = (x \cdot y) + (x \cdot z) := x \cdot y + x \cdot z$  (Punktrechnung geht vor Strichrechnung).

Man beachte, daß aus der Definition folgt, daß  $0 \cdot x = 0$  für alle  $x \in K$  gilt. Wichtige (unendliche) Körper sind der Körper der rationalen Zahlen  $(\mathbb{Q}, +, \cdot)$ , der Körper der reellen Zahlen  $(\mathbb{R}, +, \cdot)$  und der Körper der komplexen Zahlen  $(\mathbb{C}, +, \cdot)$ . In der Kryptographie spielen dagegen endliche Körper eine wichtige Rolle, beispielsweise bei der Definition des diskreten Logarithmus und bei der Kryptographie mit elliptischen Kurven. Das folgende Beispiel ist von fundamentaler Bedeutung.

**Beispiel 6.6.** Sei  $p$  eine Primzahl. Wegen  $\mathbb{Z}_p^* = \{a + p\mathbb{Z} \mid \text{ggT}(a, p) = 1\} = \{1, 2, \dots, p-1\} = \mathbb{Z}_p \setminus \{0\}$  ist  $(\mathbb{Z}_p \setminus \{0\}, \cdot)$  eine (multiplikative) abelsche Gruppe.  $\mathbb{Z}_p^*$  ist sogar zyklisch. Da  $(\mathbb{Z}_p, +)$  eine (additive) abelsche Gruppe ist und da das Distributivgesetz erfüllt ist, ist somit  $(\mathbb{Z}_p, +, \cdot)$  ein Körper. Für die Anzahl der Elemente dieses Körpers gilt  $|\mathbb{Z}_p| = p$ . Wegen  $p \cdot x = 0 \cdot x = 0$  für alle  $x \in \mathbb{Z}_p$  sagt man, daß  $\mathbb{Z}_p$  die Charakteristik  $p$  hat.

## 7 Public-Key-Kryptographie

### 7.1 Trapdoor-Einwegfunktionen

Im Jahr 1976 veröffentlichten Whitfield Diffie und Martin Hellman das Prinzip der Public-Key-Kryptographie. Bis zu diesem Zeitpunkt mussten zwei Teilnehmer (Quelle, Senke), die geheim kommunizieren wollten, schon vorher ein gemeinsames Geheimnis haben, den geheimen Schlüssel (Verwendung symmetrischer Kryptosysteme). Ab 1976 war es möglich, ohne jeden vorherigen Kontakt einer Person eine verschlüsselte Nachricht zu schicken, die nur sie selbst entschlüsseln kann. Grundlage dafür sind Trapdoor-Einwegfunktionen.

Betrachtet man ein Kryptosystem  $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ , so ist eine Verschlüsselungsfunktion  $e_{k_1} : \mathcal{P} \rightarrow \mathcal{C}$  eine Trapdoor-Einwegfunktion, falls  $e_{k_1}$  trotz Kenntnis des Schlüssels

$k_1$  (und damit Kenntnis der kompletten Funktion  $e_{k_1}$ ) praktisch (im Sinne der Komplexitätstheorie) nicht umkehrbar ist, also aus  $c = e_{k_1}(p)$  das entsprechende  $p$  nicht berechnet werden kann (Einwegfunktionen), auf der anderen Seite mit Hilfe einer Zusatzinformation  $k_2 \in \mathcal{K}$  eine einfach auszuwertende Funktion  $d_{k_2} : \mathcal{C} \rightarrow \mathcal{P}$  angegeben werden kann mit  $d_{k_2}(e_{k_1}(p)) = p$  für alle  $p \in \mathcal{P}$  (Trapdoor-Eigenschaft). Ist nun für jedes  $k \in \mathcal{K}$  die Funktion  $e_k$  eine Trapdoor-Einwegfunktion, so kann man  $|\mathcal{K}|$  Teilnehmern jeweils einen Schlüssel  $k_i$  zuordnen und sowohl den Teilnehmer als auch den entsprechenden Schlüssel publizieren („Telefonbuch“).

Die Frage, ob es überhaupt Trapdoor-Einwegfunktionen gibt, ist bis heute unbeantwortet. Man kann allerdings Funktionen angeben, die nach dem jetzigen Stand der Technik Trapdoor-Einwegfunktionen sind. Dazu betrachten wir die Menge  $(\mathbb{Z}_n, \cdot)$  mit der Multiplikation  $x \cdot y := xy \bmod n$ . Für jedes  $k \in \mathbb{N}$  kann durch die Multiplikation auch das Potenzieren in  $\mathbb{Z}_n$  definiert werden:

$$x^0 := 1, \quad x^k := x \cdot x^{k-1}$$

Sei nun  $n = p \cdot q$  das Produkt zweier Primzahlen  $p \neq q$ , so betrachten wir die Funktionen

$$e_k : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \quad x \mapsto x^k, \quad k \in \mathbb{N}$$

für alle  $k$ , die teilerfremd zu  $\varphi(n) = (p-1)(q-1)$  sind. Es läßt sich zeigen, daß für jedes dieser  $k$  die jeweilige Funktion  $e_k$  bijektiv ist und daß es ein  $\bar{k} \in \mathbb{N}$  gibt mit

$$d_{\bar{k}} : \mathbb{Z}_n \rightarrow \mathbb{Z}_n : x \mapsto x^{\bar{k}} \text{ und } d_{\bar{k}}(e_k(x)) = e_k(d_{\bar{k}}(x)) = x \quad \forall x \in \mathbb{Z}_n$$

Die natürliche Zahl  $\bar{k}$  ist charakterisiert durch

$$k \cdot \bar{k} = 1 \bmod \varphi(n)$$

Kennt man die Primfaktorzerlegung  $n = p \cdot q$  von  $n$ , so läßt sich  $\bar{k}$  aus  $p$  und  $q$  relativ einfach mit dem erweiterten euklidischen Algorithmus berechnen. Kennt man aber nur die Zahlen  $n$  und  $k$ , die die Funktion  $e_k$  vollständig charakterisieren, so ist zwar  $e_k$  leicht auszuwerten aber bis heute  $p$  und  $q$  (und damit  $\bar{k}$ ) im allgemeinen nicht in adäquater Zeit zu berechnen.

## 7.2 Das RSA-Verfahren

Das von R. Rivest, A. Shamir und L. Adleman 1978 erfundene und nach ihnen benannte RSA-Verfahren verwendet die im letzten Abschnitt angegebenen Trapdoor-Einwegfunktionen

$$e_k : \mathbb{Z}_n \rightarrow \mathbb{Z}_n, \quad x \mapsto x^k$$

mit öffentlichem Schlüssel  $(n, k)$ , der Primfaktorzerlegung  $n = pq$  mit  $p \neq q$  und dem geheimen Schlüssel  $\bar{k}$  mit

$$k \cdot \bar{k} = 1 \bmod \varphi(n)$$

und  $(x^k)^{\bar{k}} = (x^{\bar{k}})^k = x$  für alle  $x \in \mathbb{Z}_n$ . Entscheidend für die Sicherheit des RSA-Verfahrens ist die Größe der beiden Primzahlen  $p$  und  $q$ , die in Binärdarstellung durch die Anzahl der nötigen Bits charakterisiert werden. Heute werden für  $p$  und  $q$  256 Bits (entspricht etwa 100 Dezimalstellen) verwendet. Somit entspricht  $n$  einer ganzen Zahl aus 512 Bits. Nehmen wir nun an, unser Klartext bestehe aus Buchstaben, Ziffern und Sonderzeichen, so wird zum Beispiel im ASCII-Code jedes der möglichen 256 Zeichen durch 8 Bits dargestellt (American Standard Code for Information Interchange)

| ASCII-Zeichen | Binäre Form |
|---------------|-------------|
| Zwischenraum  | 00100001    |
| 0             | 00110000    |
| 1             | 00110001    |
| ⋮             |             |
| 9             | 00111000    |
| A             | 01000001    |
| Z             | 01011010    |
| ⋮             |             |

Somit entspricht einer Zahl  $m \in \{0, 1, \dots, 2^{512} - 1\}$  (darstellbar durch 512 Bits) ein Text bestehend aus maximal 64 Zeichen ( $64 \cdot 8 = 512$ ). Dem Klartext "ALLES KLAR!" entspricht eine natürliche Zahl  $m \approx 1,57 \cdot 10^{26}$ . Zwei entscheidende Nachteile des RSA-Verfahrens führten auf die Suche nach anderen Trapdoor-Einwegfunktionen und damit auch alternativer kryptographischer Verfahren.

- (i) Sobald es jemandem gelingt, die Primfaktorzerlegung natürlicher Zahlen effizient durchführen zu können, muß das RSA-Verfahren sofort ersetzt werden. Es ist dann zu spät, mit der Suche nach Alternativen zu beginnen.
- (ii) Die bisherigen Algorithmen zur Primfaktorzerlegung zwingen den Kryptographen, Primzahlen mit mindestens 512 Bits zu verwenden. Diese Größen machen die Ver- und Entschlüsselung zu sehr komplexen Vorgängen. Gesucht sind Public-Key-Verfahren, die mit kleineren Zahlen auskommen.

### 7.3 Der diskrete Logarithmus

Für jede Primzahl  $p$  ist die Einheitengruppe von  $(\mathbb{Z}_p, \cdot)$  die multiplikative Gruppe

$$\mathbb{Z}_p^* = \{a + p\mathbb{Z} \mid \text{ggT}(a, p) = 1\} = \{1, 2, \dots, p-1\}.$$

Man kann zeigen, daß diese Gruppe zyklisch ist, d. h. es gibt in  $\mathbb{Z}_p$  mindestens ein erzeugendes Element  $g$  mit  $\mathbb{Z}_p^* = \langle g \rangle = \{g^k | k = 1, 2, \dots, p-1\}$ . Nun betrachten wir die Abbildung

$$\exp_g : \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^* : x \mapsto g^x$$

für jedes erzeugende Element  $g$  von  $(\mathbb{Z}_p^*, \cdot)$ . Die Abbildung  $\exp_g$  ist bijektiv. Analog zum Vorgehen in der Analysis wird nun die Umkehrabbildung zu  $\exp_g$  als diskreter Logarithmus bezeichnet:

$$\log_g : \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^*, \quad x \mapsto \log_g(x) \text{ mit } g^{\log_g(x)} = x.$$

Das Problem des diskreten Logarithmus in  $(\mathbb{Z}_p^*, \cdot)$  lautet also folgendermaßen: Finde zur Basis  $g$  mit  $\langle g \rangle = \mathbb{Z}_p^*$  und zu  $b \in \mathbb{Z}_p^*$  ein  $x \in \mathbb{Z}_p^*$  mit  $g^x = b$ . Man schreibt  $\log_g b = x$ . Da diese Funktion im allgemeinen nur mit größtem Aufwand auszuwerten ist (im Gegensatz zu  $\exp_g$ ) ist die Exponentialfunktion zu speziellem erzeugendem Element  $g$  für spezielles  $p$  eine Einwegfunktion. Allerdings sind diese Funktionen zunächst für die Kryptographie nicht geeignet, da keine Trapdoor-Information in Sicht ist. Dennoch kann man die diskreten Exponentialfunktionen wie Trapdoor-Einwegfunktionen verwenden, wenn man dem zu verschlüsselnden Geheimtext noch gewisse Zusatzinformationen beifügt. Dies geschieht im Verschlüsselungsverfahren von El Gamal.

## 7.4 Das Verschlüsselungsverfahren von El Gamal

Ausgangspunkt ist die multiplikative Gruppe  $\mathbb{Z}_p^*$  mit einem erzeugenden Element  $g$ . Ein Teilnehmer, der mit einem öffentlichen Schlüssel in ein Verzeichnis eingetragen werden will, wählt zufällig ein  $t \in \mathbb{N}$  und berechnet seinen öffentlichen Schlüssel  $(\tau, g)$  aus  $\tau := g^t \bmod p$ . Eine Quelle, die dem Teilnehmer mit öffentlichem Schlüssel  $\tau$  eine Nachricht senden will, wählt sich zufällig eine Zahl  $a \in \mathbb{N}$  und berechnet die Zahlen  $\alpha := g^a \bmod p$  und  $k := \tau^a \bmod p$ . Dann verwendet er ein vorher festgelegtes, symmetrisches Kryptosystem mit Schlüsselmenge  $\mathcal{K} = \mathbb{Z}_p^*$  und Verschlüsselungsfunktion  $e_k$  für  $k \in \mathcal{K}$  und sendet seine geheime Nachricht  $c = e_k(p)$  und die Information  $\alpha = g^a \bmod p$  mit dem berechneten Schlüssel an den Empfänger. Da nun

$$k = \tau^a \bmod p = (g^t)^a \bmod p = (g^a)^t \bmod p = \alpha^t \bmod p$$

gilt, kann der Empfänger mit dem geheimen Schlüssel  $t$  den Schlüssel  $k$  unter Verwendung von  $\alpha$  berechnen und somit dechiffrieren. Ein Angreifer, der nur  $\tau$ ,  $\alpha$  und  $g$  kennt, hat somit  $t = \log_g(\tau)$  zu berechnen, um  $k$  zu erhalten.

## 7.5 Elliptische Kurven

Bisher haben wir den diskreten Logarithmus für die multiplikativen Gruppen  $(\mathbb{Z}_p^*, \cdot)$  betrachtet. Analog dazu kann man auch für andere zyklische Gruppen den diskreten

Logarithmus definieren. In der additiven zyklischen Gruppe  $(\mathbb{Z}_p, +)$  mit einer Primzahl  $p$  lautet die der Exponentialfunktion entsprechende Abbildung

$$e_g : \mathbb{Z}_p \rightarrow \mathbb{Z}_p, \quad x \mapsto \begin{cases} \underbrace{g + \dots + g}_{x\text{-mal}} =: x \cdot g, & x \neq 0 \\ 0 & x = 0. \end{cases}$$

für ein erzeugendes Element  $g$ , d. h. für ein Element mit  $\langle g \rangle = \{k \cdot g \mid k \in \{0, \dots, p-1\}\} = \mathbb{Z}_p$ . Es läßt sich zeigen, daß in  $(\mathbb{Z}_p, +)$  alle von Null verschiedenen Elemente  $g \in \mathbb{Z}_p \setminus \{0\}$  erzeugende Elemente sind. Die Abbildung  $e_g$  mit  $g \in \mathbb{Z}_p \setminus \{0\}$  ist bijektiv. Daher existiert eine dem diskreten Logarithmus entsprechende Umkehrabbildung

$$d_g : \mathbb{Z}_p \rightarrow \mathbb{Z}_p, \quad x \mapsto d_g(x) \text{ mit } e_g(d_g(x)) = x.$$

Das Problem des diskreten Logarithmus in  $(\mathbb{Z}_p, +)$  lautet also: Finde zur Basis  $g \in \mathbb{Z}_p \setminus \{0\}$  und zu  $b \in \mathbb{Z}_p$  ein  $x$  mit  $x \cdot g = b$ . Die Lösung ist  $x = g^{-1}b$ , wobei  $g^{-1}$  das multiplikative Inverse von  $g$  in  $\mathbb{Z}_p^*$  bezeichnet. Da das inverse Element in  $\mathbb{Z}_p^*$  mit dem erweiterten euklidischen Algorithmus effizient berechenbar ist, muß man nach anderen endlichen zyklischen oder allgemeiner abelschen (additiven) Gruppen suchen, in denen der diskrete Logarithmus praktisch nicht auszuwerten ist. Diese abelschen Gruppen erhält man unter anderem durch elliptische Kurven.

Seien  $p > 3$  eine Primzahl und  $a, b \in \mathbb{Z}_p$ . Man betrachtet folgende Gleichungen in drei Unbekannten  $x, y, z \in \mathbb{Z}_p$ , wobei jetzt  $\mathbb{Z}_p$  bezüglich  $+$  und  $\cdot$  (als endlicher Körper) gegeben ist:

$$y^2 \cdot z = x^3 + a \cdot x \cdot z^2 + b \cdot z^3.$$

Man interessiert sich nun für alle  $(x', y', z') \in \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p$ , die für feste  $a, b \in \mathbb{Z}_p$  die obige Gleichung erfüllen. Dabei heißen zwei Lösungen  $(x', y', z')$  und  $(x'', y'', z'')$  äquivalent, falls es ein  $c \in \mathbb{Z}_p$  mit  $c \neq 0$  gibt, so daß

$$(x'', y'', z'') = (c \cdot x', c \cdot y', c \cdot z').$$

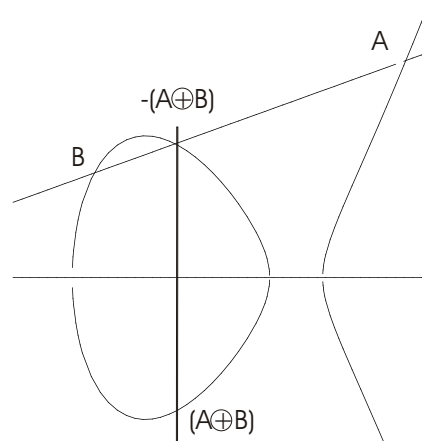
Interessant sind alle nichtäquivalenten Lösungen, von denen man sich immer genau eine in einer Menge  $E(p, a, b)$  notiert. Es läßt sich zeigen, daß man alle nichtäquivalenten Lösungen der obigen Gleichung folgendermaßen darstellen kann (die Lösung  $(0, 0, 0)$  wird nicht berücksichtigt):

$$E(p, a, b) = \{(x', y', 1) \mid y'^2 = x'^3 + a \cdot x' + b, x', y' \in \mathbb{Z}_p\} \cup \{0, 1, 0\}.$$

Nun soll eine Addition  $\oplus$  auf  $E(p, a, b)$  definiert werden. Dazu betrachten wir elliptische Kurven zunächst über den reellen Zahlen. Fordert man als Bedingung an die Parameter  $a, b \in \mathbb{R}$ , daß für die Diskriminante  $D$  der rechten Seite

$$D := \frac{a^3}{27} + \frac{b^2}{4} \neq 0$$

gilt, so kann man zeigen, daß für die zugehörige implizite Gleichung  $F(x', y') = 0$  für  $y'$  als Funktion von  $x'$ , also für  $F(x', y') = y'^2 - x'^3 - ax' - b$ , in jedem Punkt  $(x', y')$  mindestens eine der partiellen Ableitungen  $\frac{\partial F}{\partial x'}$ ,  $\frac{\partial F}{\partial y'}$  ungleich Null ist. Insbesondere gibt es dann immer einen (falls  $D > 0$ ) oder drei (falls  $D < 0$ ) verschiedene Schnittpunkte mit der  $x$ -Achse. Es ergeben sich Lösungspunkte der folgenden Art:



Die Addition  $\oplus$  wird nun folgendermaßen definiert: Sei  $E(a, b)$  eine elliptische Kurve und seien  $A = (x_1, y_1, 1) \in E(a, b)$ .

$$A \oplus (0, 1, 0) := (0, 1, 0) \oplus A = A,$$

also ist  $(0, 1, 0) =: 0$  das neutrale Element bezüglich  $\oplus$ . Das zu  $A$  inverse Element  $-A$  wird definiert als

$$-A := (x_1, -y_1, 1).$$

Sei  $B = (x_2, y_2, 1) \in E(a, b)$ . Haben  $A$  und  $B$  verschiedene  $x'$ -Koordinaten, so schneidet die Gerade  $\overline{AB}$  den  $(x', y')$ -Graphen der Menge  $E(a, b)$  in einem weiteren Punkt  $C = (\hat{x}, \hat{y}, 1)$ . Es gilt dann

$$A \oplus B := -C \quad \text{und} \quad -C = (\hat{x}, -\hat{y}, 1).$$

Sind die  $x'$ -Koordinaten gleich, die  $y'$ -Koordinaten jedoch nicht, so setzt man  $A \oplus B = 0$ , da dann  $B = -A$  gilt. Für  $B = A$  verwendet man die Tangente an den  $(x', y')$ -Graphen, um  $C$  zu erhalten. Ist die Tangente parallel zur  $y'$ -Achse, so ist  $A \oplus A = 0$ . Ist  $A$  ein Wendepunkt, so ist  $A \oplus A = -A$ .

Analog dazu führt man nun  $\oplus$  in  $E(p, a, b)$  über  $\mathbb{Z}_p$  ein. Man erhält eine additive abelsche Gruppe  $(E(p, a, b), \oplus)$ . Die Anzahl der Elemente  $N$  dieser Gruppe läßt sich abschätzen durch

$$p + 1 - 2\sqrt{p} \leq N \leq 2\sqrt{p} + p + 1$$

(Theorem von Hasse). Durch Variation von  $a, b$  lassen sich alle in diesem Intervall liegenden  $N$  erzeugen.

Man verwendet nun Gruppen  $(E(p, a, b), \oplus)$ , um durch die Funktionen  $e_g$  und ihre Umkehrfunktionen  $d_g$  zu einer ElGamal-Verschlüsselung zu kommen. In der additiven Schreibweise ist die Exponentialfunktion  $e_g$  dabei gegeben als Multiplikation mit einer ganzen Zahl. Diese Funktion ist genau dann bijektiv, wenn  $g$  ein erzeugendes Element ist. Da die hier betrachteten abelschen Gruppen jedoch nicht zyklisch sein müssen, sucht man nur einen Punkt  $B \in E(p, a, b)$ , der eine möglichst große Untergruppe erzeugt. In der von  $B$  erzeugten Untergruppe lautet das Problem des diskreten Logarithmus zur Basis  $B$ : Finde zu einem gegebenen Punkt  $P \in E$  ein  $x \in \mathbb{Z}$  mit  $xB = P$ . Die Multiplikation mit einer ganzen Zahl ist dabei wie in  $(\mathbb{Z}_p, +)$  mit Hilfe der Addition definiert. Bei einer ElGamal-Verschlüsselung mit Elliptischen Kurven kommt man im Gegensatz zum RSA-Algorithmus mit etwa 163-Bit-Zahlen für  $p$  aus. Bereits dann ist der diskrete Logarithmus sehr komplex zu berechnen. Dies rechtfertigt den hohen Aufwand zur Berechnung von  $E(p, a, b)$  für geeignet gewählte  $a, b$  und die etwas kompliziertere Arithmetik. Es ist wichtig festzuhalten, daß die Gruppe  $E(p, a, b)$  a priori gewählt werden kann. Der Aufwand dafür ist zwar im allgemeinen sehr hoch, ist aber vor der Implementierung eines Kryptosystems zu leisten.

## 8 Zero-Knowledge-Verfahren

### 8.1 Authentifikationssysteme

In diesem Abschnitt betrachten wir spezielle Authentifikationssysteme, bei denen sich eine Person  $A$  gegenüber einer Person  $B$  durch den Nachweis eines bestimmten Geheimnisses identifizieren muß. Idealerweise sollte das so erfolgen, daß

- $B$  das Geheimnis von  $A$  nicht vorher kennen muß,
- $B$  während der Identifikation von  $A$  nichts über dieses Geheimnis erfährt.

Somit kann sich  $B$  Dritten gegenüber nicht als  $A$  ausgeben. Ziel ist es, daß sich  $B$  von  $A$ 's Identität mit beliebig großer Sicherheit überzeugt, ohne irgendwelche Informationen über das Geheimnis von  $A$  zu erhalten. Dies wird als Zero-Knowledge-Eigenschaft bezeichnet.

Wir wollen uns anhand eines Beispiels klarmachen, daß ein Authentifikationssystem mit den oben genannten Eigenschaften möglich ist. Eine Person  $A$  soll gegenüber einer Person  $B$  nachweisen, daß sie quadratische Gleichungen

$$ax^2 + bx + c = 0$$

für beliebige Koeffizienten  $a, b, c \in \mathbb{R}$  lösen kann. Das Verfahren läuft folgendermaßen ab:

- $B$  nennt drei reelle Zahlen  $a, b, c$ .
- $A$  berechnet die beiden Lösungen der Gleichung  $ax^2 + bx + c = 0$  und teilt diese  $B$  mit.
- $B$  überprüft die Richtigkeit von  $A$ 's Lösung durch Einsetzen in die Gleichung.

Dieses Verfahren kann beliebig oft wiederholt werden und somit kann sich  $B$  beliebig sicher davon überzeugen, daß  $A$  das Geheimnis (die Lösung der quadratischen Gleichung) kennt, ohne daß  $B$  etwas von diesem Geheimnis erfährt.

In der Public-Key-Kryptographie werden Zero-Knowledge-Verfahren verwendet, für die sich beweisen läßt, daß  $B$  während der Authentikation nichts über  $A$ 's Geheimnis erfährt.

## 8.2 Der Fiat-Shamir-Algorithmus

Das bekannteste und in der Praxis wichtigste Zero-Knowledge-Verfahren ist der Fiat-Shamir-Algorithmus. Die Idee dieses Algorithmus beruht darauf, daß es praktisch unmöglich ist, Quadratwurzeln in  $(\mathbb{Z}_n^*, \cdot)$  zu berechnen. Es läßt sich zeigen, daß dieses Problem genauso schwer zu lösen ist wie das Problem, die Primfaktoren von  $n$  zu finden. Der Fiat-Shamir-Algorithmus besteht aus zwei Phasen, der Schlüsselerzeugungsphase und der Anwendungsphase. In der ersten Phase erzeugt  $A$  zunächst zwei große Primzahlen  $p$  und  $q$  und bildet ihr Produkt  $n = pq$ . Die Zahl  $n$  ist öffentlich, während  $p$  und  $q$  nur  $A$  bekannt sein dürfen. Dann wählt  $A$  eine Zahl  $s$  und berechnet  $v := s^2 \bmod n$ . Die Zahl  $s$  ist das individuelle Geheimnis des Teilnehmers  $A$ , während man mit Hilfe von  $v$  verifizieren kann, ob eine Person das Geheimnis kennt oder nicht. Das heißt, daß  $s$  geheim bleiben muß und  $v$  publiziert wird.

In der Anwendungsphase muß  $A$  eine Person  $B$  davon überzeugen, daß sie das Geheimnis  $s$  kennt, ohne daß  $B$  dieses Geheimnis mitgeteilt wird. Dazu führen  $A$  und  $B$  folgendes Protokoll durch:

- $A$  wählt zufällig ein Element  $r$  aus  $\mathbb{Z}_n^*$  und quadriert dieses:  $x := r^2 \bmod n$ . Danach sendet  $A$  den Wert  $x$  an  $B$ ;
- $B$  wählt zufällig ein Bit  $b$  und sendet dieses an  $A$ ;
- ist  $b = 0$ , so sendet  $A$  den Wert  $y := r$  an  $B$ ;  
ist  $b = 1$ , so sendet  $A$  den Wert  $y := r \cdot s \bmod n$  an  $B$ .
- $B$  verifiziert diese Antworten: Im Falle  $b = 0$  überprüft sie, ob  $y^2 \bmod n = x$  ist; im Fall  $b = 1$  testet sie die Richtigkeit der Gleichung  $y^2 \bmod n = x \cdot v \bmod n$ .

Durchführbarkeit: Wenn  $A$  das Geheimnis  $s$  kennt, so kann sich  $B$  davon überzeugen, denn es gilt

$$y^2 = (r \cdot s^b)^2 \bmod n = r^2 \cdot s^{2b} \bmod n = r^2 v^b \bmod n = x \cdot v^b \bmod n.$$

Korrektheit: Eine betrügerische Person  $\tilde{A}$ , die  $s$  nicht kennt, kann nicht beide Fragen  $b = 0$  und  $b = 1$  korrekt beantworten, denn sonst könnte sie aus der korrekten Antwort  $y_0 = r$  und  $y_1 = r \cdot s$  den Schlüssel  $s = y_1 y_0^{-1} \bmod n$  berechnen.  $\tilde{A}$  kann also höchstens mit der Wahrscheinlichkeit  $\frac{1}{2}$  betrügen. Diese Wahrscheinlichkeit  $\frac{1}{2}$  kann  $\tilde{A}$  erzwingen: Wenn sie vermutet, daß  $B$  "0" sendet, so wählt sie  $x := r^2 \bmod n, y = r$ . Wenn sie vermutet, daß "1" von  $B$  gesendet wird, so wählt sie  $x = r^2 \cdot v^{-1} \bmod n$  und  $y = r$ , denn dann ist  $y^2 = xv \bmod n$ .

## Literatur

- [Ba94] F. L. Bauer, *Kryptologie*. Springer-Verlag, 2. Auflage 1994.
- [Bau90] H. Bauer, *Maß- und Integrationstheorie*. de Gruyter-Verlag, 1990.
- [Bau91] H. Bauer, *Wahrscheinlichkeitstheorie*. de Gruyter-Verlag, 4. Auflage 1991.
- [Be93] A. Beutelspacher, *Kryptologie*. Vieweg-Verlag, 3. Auflage 1993.
- [Bi86] P. Billingsley, *Probability and Measure*. Wiley, 2. Auflage 1986.
- [Boe93] J. F. Böhme, *Stochastische Signale*. Teubner-Verlag, 1993.
- [Bo92] M. Bossert, *Kanalcodierung*. Teubner-Verlag, 1992.
- [Fr96] B. Friedrichs, *Kanalcodierung*. Springer-Verlag, 1996.
- [FuRi94] W. Fumy und H. P. Rieß, *Kryptographie*. Oldenbourg-Verlag, 2. Auflage 1994.
- [Ha91] E. Hänsler, *Statistische Signale*. Springer-Verlag, 1991.
- [HeQu95] W. Heise und P. Quattrocchi, *Informations- und Codierungstheorie*. Springer-Verlag, 3. Auflage 1995.
- [Hi2001] C. Hillermeier, *Informationstheorie*, Skriptum, Erlangen 2001.
- [Ju95] D. Jungnickel, *Codierungstheorie*. Spektrum Akademischer Verlag, 1995.
- [Pr95] J. G. Proakis, *Digital Communications*. McGraw-Hill, 3. Auflage 1995.
- [Ro95] H. Rohling, *Einführung in die Informations- und Codierungstheorie*. Teubner-Verlag, 1995.
- [Ro91] S. Roman, *Coding and Information Theory*. Springer, 1991.
- [ScSt94] S. Schäffler und T. F. Sturm, *Wahrscheinlichkeitstheorie und Statistik I für Mathematiker*. Skriptenreihe des IAMS, Band 5, TU München, 1994.
- [ScSt95] S. Schäffler und T. F. Sturm, *Wahrscheinlichkeitstheorie und Statistik II für Mathematiker*. Skriptenreihe des IAMS, Band 6, TU München, 1995.
- [ScSu02] S. Schäffler und A. Sutor, *Kryptographie*, Skriptum, München 2002.
- [Ver69] *Error detecting decimal codes*. Math. Centre Tracts. 29, Amsterdam, 1969.