Skript zur Schnuppervorlesung 2013

Nicht zu knacken? - Mathematische Werkzeuge zur Geheimniswahrung

Felix Wellen, Lena Martin
Karlsruher Institut für Technologie
Fakultät für Mathematik

27. Juni 2013

Inhaltsverzeichnis

Einleitung

Kryptologie und Steganographie

Es gibt verschiedene Möglichkeiten zur Geheimhaltung einer Nachricht. Grundsätzlich kann man Maßnahmen bezüglich ihres Ansatzes unterscheiden: Zum einen kann man versuchen, die Existenz einer Nachricht zu verbergen. Ein Dritter soll beim Betrachten des Trägermediums gar nicht bemerken, dass eine geheime Nachricht übermittelt wird. Ein altbekannter Trick ist z.B. das Schreiben mit Zitronensaft oder das Verstecken einer Nachricht innerhalb eines anderen Textes.

Heutzutage werden Infomationen z.B. in Bild- oder Audiodateien versteckt. Die Wissenschaft, die sich hiermit auseinandersetzt, ist die Steganographie (*steganós* griech. *bedeckt*). Zum anderen kann man aber auch eine Nachricht so verändern, dass ein unberechtigter Leser ihr keinen Sinn entnehmen kann. Hiermit beschäftigt sich die *Kryptologie*. Das Wort *kryptos* stammt aus dem Griechischen und bedeutet *verborgen, geheim*, *Krypotgraphie* heißt soviel wie *verborgenes Schreiben* und *Kryptologie Lehre vom Geheimen*. Die Begriffe Kryptographie und Kryptologie werden heutzutage oft synonym verwendet, manchmal wird jedoch auch die Kryptographie als das Teilgebiet der Kryptologie aufgefasst, welches sich mit der Entwicklung von Verschlüsselungstechniken beschäftigt. In der Kryptoanalyse untersucht man, ob und wie man bestimmte Verschlüsselungen knacken kann.
Mathematik spielt in allen Teilgebieten der Kryptologie eine wesentliche Rolle. Durch sie erhält man z.B. die Möglichkeit, die Sicherheit eines Verfahrens abzuschätzen. Zudem kann man mit ihrer Hilfe sicherer Verfahren entwickeln und verbessern. Welche mathematischen Methoden und Werkzeuge dabei in der Kryptologie zum Einsatz kommen, wollen wir in der Schnuppervorlesung vorstellen. Dabei werden wir ganz unterschiedliche Gebiete aus der Mathematik kennenlernen.

Kapitel 1

Symmetrische Verschlüsselungsverfahren

1.1 Wichtige Begriffe

- *Klartext*: Buchstaben- oder Zeichenfolge, die man übermitteln möchte.
- *Geheimtext*: Verschlüsselte Nachricht.
- Die Wörter *chiffrieren* und *verschlüsseln* bedeuten, dass man den Klartext in einen Geheimtext umwandelt.
- *Dechiffrieren* heißt dann den Verschlüsselungsvorgang wieder rückgängig zu machen. Man sagt auch, den Geheimtext wieder in den Klartext *entschlüsseln*.
- Als *Chiffrieralgorithmus* bezeichnet man die allgemeine Vorschrift des Vorgangs der Ver- und Entschlüsselung.
- Der *Schlüssel* stellt jeweils eine konkrete Verschlüsselungsanweisung dar.
- *Symmetrisches Verschlüsselungsverfahren*: Sender und Empfänger verwenden denselben Schlüssel zum Ver- und Entschlüsseln.

1.1.1 Alphabet

Nachrichten, die wir verschlüsseln wollen, bestehen aus Zeichen. Die Menge der Zeichen, die in einem Text auftreten, bezeichnen wir als Alphabet. Meist wird unser Klartextalphabet das uns wohlbekannte Alphabet $\{$a,b,c, . . . , x,y,z$\}$ sein, wobei Umlaute wie ä als ae geschrieben werden, ß wird durch ss ersetzt. Das Geheimtextalphabet kann dasselbe sein, man kann aber Zahlen verwenden oder eigene Zeichen erfinden z.B. $\{0, 1, 2, \ldots, 25\}$ oder $\{* \sharp \circ \star \divideontimes \bigtriangledown \rtimes \ltimes \amalg \otimes \odot \circledcirc \circledast \bigtriangleup \cup \wr \sqcap \vee \infty \lhd \flat \heartsuit \diamond \perp \ni \blacksquare\}$.

Bei einer monoalphabetischen Verschlüsselung (*monos*, griech.: einzig, allein) wird jeder Klartextbuchstabe durch einen Geheimtextbuchstabe ersetzt, wobei jedem Buchstaben des Klartextalphabets genau ein fester Buchstabe des Geheimalphabets zugeordnet wird. Deshalb spricht man auch von monoalphabetischer Substitution (*substituere*, lat.: ersetzen) .

1.1.2 Cäsar-Chiffre

Eines der ältesten Beispiele für eine monoalphabetische Chiffrierung ist die Cäsar-Chiffre, die von Julius Cäsar (100 bis 44. v. Chr.) verwendet wurde. Die Zeichen des Geheimtextalphabets sind dabei dieselben wie die des Klartextalphabet. Man schreibt das Geheimtextalphabet einfach um einige

Abbildung 1.1: Cäsarscheibe

Stellen verschoben unter das Klartextalphabet - bei Cäsar waren es drei Stellen.

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

Insgesamt kann man aus 26 Verschiebe-Chiffren wählen, wobei die Verschiebungen mit dem Schlüsselbuchstaben A trivial ist, es gibt also 25 sinnvolle Schlüssel. Im Jahre 1470 erfand Leon Battista Alberti ein simples Hilfsmittel für die Cäsar-Verschlüsselung: Die Cäsarscheiben.

Die Cäsar-Verschlüsselung ist sehr schnell zu knacken. Es genügt, einfach systematisch die Schlüssel durchzuprobieren. Spätestens im 25. Versuch ist man erfolgreich. Man könnte nun also einfach versuchen, eine größere Anzahl an möglichen Schlüsseln zu erzeugen, um eine höhere Sicherheit zu gewährleisten. Dies könnte man z.B. dadurch erreichen, dass man das Alphabet nicht nur verschiebt, sondern die 26 Buchstaben des Geheimtextalphabets in beliebiger Reihenfolge unter das Klartextalphabet schreibt:

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

In der Mathematik nennt man solch eine Umordnung von Objekten eine Permutation. Ordnet man n Dinge in einer Reihe an, so kann man sich überlegen, wie viele Möglichkeiten es dafür gibt: $n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$. Man schreibt für das Produkt kurz $n!$ und sagt dazu n-Fakultät. In unserem Falle besteht die Menge, das Alphabet, aus 26 Buchstaben. Der Schlüssel besteht aus der Reihenfolge in der wir die Buchstaben des Geheimtextalphabets unter das Klartextalphabet schreiben. Nach obiger Überlegung gibt also $26! = 403291461126605635584000000 \approx 4 \cdot 10^{26}$ mögliche Anordnungen der Buchstaben im Geheimtextalphabet und damit ebensoviele Schlüssel. Einfaches Ausprobieren von Hand wie im Falle der Cäsarverschlüsselung ist hier also nicht mehr zu realisieren. Dass man diese Art der Verschlüsselung mit Methoden der Statistik dennoch leicht knacken kann, werden wir im Folgenden sehen.

1.1.3 Permutationen

Definition 1.1 Eine *Permutation* von n verschiedenen Elementen ist eine Abbildung

$$\pi : \{1, \ldots, n\} \to \{1, \ldots, n\},$$

die keine zwei Elemente aus $\{1, \ldots, n\}$ auf das gleiche Element abbildet. Daher können Permutationen auf die folgende Art notiert werden

$$\pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ \pi(1) & \pi(2) & \ldots & \pi(n) \end{pmatrix}$$

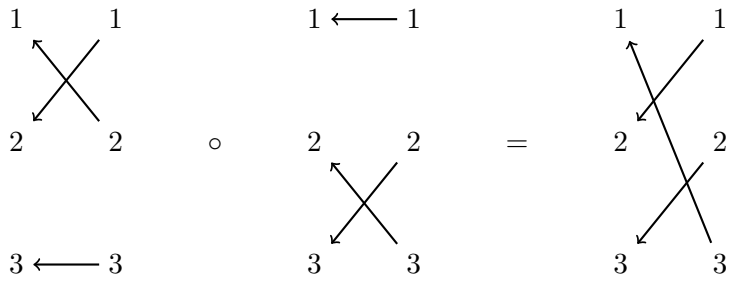
wobei in der unteren Zeile jede der Zahlen $1, \ldots, n$ genau einmal vorkommt. Es gibt noch andere Schreibweisen, um Permutationen aufzuschreiben. Wir wollen es aber der Einfachheit halber bei dieser belassen. Die allereinfachste Form der Permutation ist die sogenannte Identität, bei ihr wird jedes Element auf sich selbst abgebildet:

$$\mathrm{Id} = \begin{pmatrix} 1 & 2 & \ldots & n \\ 1 & 2 & \ldots & n \end{pmatrix}.$$

Die nächstkomplziertere Stufe sind die *Transpositionen*, bei denen jeweils nur zwei Elemente i und j vertauscht werden, während alle anderen Einträge fest bleiben.

$$\tau_{ij} = \begin{pmatrix} 1 & \ldots & i & \ldots & j & \ldots & n \\ 1 & \ldots & j & \ldots & i & \ldots & n \end{pmatrix}.$$

Man kann Permutationen auch miteinander verknüpfen, indem man sie nacheinander ausführt. Verknüpft man zwei Permutationen, so konstruiert man aus ihnen durch Hintereinanderausführung eine dritte. Dieses Hintereinanderausführen ist wörtlich zu verstehen – es werden n Elemente zunächst gemäß der ersten Permutation und anschließend entsprechend der zweiten Permutation vertauscht. Insgesamt wurden die n Dinge dann auf eine neue, dritte Art vertauscht. Das Schöne daran ist, dass man durch das Verknüpfen von Transpositionen alle anderen Permutationen konstruieren kann. Das folgende Bild zeigt, wie man aus zwei Transpositionen eine Permutation gewinnt, die keine Transposition mehr ist:



In der von uns verwendeten Schreibweise für Permutationen entspricht dies:

$$\pi = \tau_2 \circ \tau_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.$$

Dies wird auch in der Sprechweise deutlich, denn man liest das Zeichen $\circ$ als „nach“. Die Verknüpfung von Permutationen ist im Allgemeinen nicht kommutativ, d.h. $\pi_1 \circ \pi_2 \neq \pi_2 \circ \pi_1$.

1.1.4 Kryptoanalyse Monoalphabetischer Verschlüsselungen

In Texten, die in natürlichen Sprachen verfasst sind, kommen die Buchstaben des zugrundeliegenden Alphabets unterschiedlich häufig vor. Tatsächlich hat jede Sprache eine eigene Charakteristik: Zwar kommt sowohl im Deutschen, Englischen, Französischen, Spanischen, Italienischen und Türkischen der Buchstabe E am häufgsten vor, jedoch findet man bei den zweithäufigsten Buchstaben stärkere Unterschiede zwischen den Sprachen. Hat man einen Geheimtext vorliegen und vermutet, dass dieser monoalphabetisch verschlüsselt ist, zählt man, wie oft ein bestimmtes Geheimtextzeichen vorkommt, notiert die Anzahl und setzt sie in Verhältnis zur Gesamtanzahl der Buchstaben des vorliegenden Textes. So erhält man relative Häufigkeiten für das Vorkommen der verschiedenen Buchstaben. Nun kann man diese mit bekannten Werten vergleichen: Im Deutschen sticht z.B. der Buchstabe E mit einer besonders hohen relativen Häufigkeit von ca. 17 % heraus, er ist damit mit großem Abstand der häufigste Buchstabe. Der zweithäufigste Buchstabe im Deutschen ist – mit ca. 9 % – das N. Die für die deutsche Sprache typische Häufigkeitsverteilung ist in Abbildung 1.2 zu sehen. Dieser Häufigkeitsanalyse liegt ein Text mit 1000 Zeichen zugrunde.

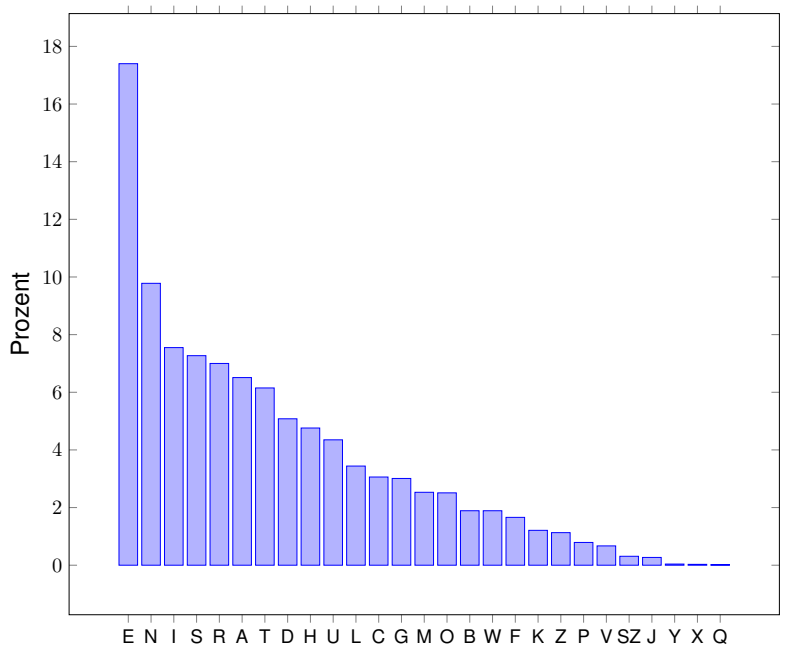


Abbildung 1.2: Relative Häufigkeiten der Buchstaben im Deutschen

Die Häufigkeitsanalyse ist ein nützliches Werkzeug beim Knacken einer Chiffrierung. Geht man davon aus, dass ein Geheimtext in in einer bestimmten Sprache z.B. in deutsch verfasst und dann monoalphabetisch verschlüsselt wurde, so kann man durch einen Vergleich der Häufigkeitsanalyse des Geheimtextes mit der typischen Häufigkeitsverteilung im Deutschen, einzelne Buchstaben, wie z.B. e. n. i , s , r, a und t identifizieren. Auch sehr hilfreich ist die Häufigkeitsanalyse von bestimmten Buchstabenfolgen. Eine Buchstabenfolge der Länge n wird als N-Gramm bezeichnet, oft werden als Vorsilben auch die griechischen Zahlwörter verwendet. Buchstabenpaare werden demnach Bigramme genannt. Im Deutschen besonders häufig auftretende Bigramme sind z.B. ER, EN. Die Bigramme IE und EI haben eine Ausnahmestellung, da I und E das einzige Buchstabenpaar ist, für die beide Kombinationen in etwa gleich oft zu finden sind. Während der Buchstabe C alleine sehr selten bis gar nicht auftritt, kommt er als CH in der Kombination mit H, einem etwas häufiger auftretenden Buchstaben, sehr oft vor, vergleiche hierzu auch Abbildung 1.3.

In der Regel kommt man beim Knacken einer monoalphabetischen Verschlüsselung schon sehr weit, wenn man die relativen Häufigkeiten der einzelnen Buchstaben und der wichtigen Bigramme kennt.

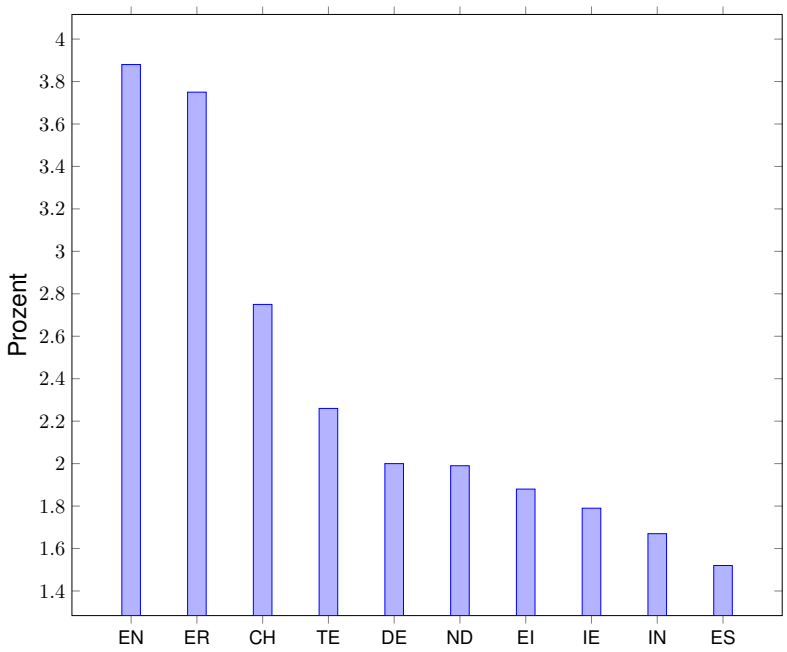


Abbildung 1.3: Relative Häufigkeiten der meistauftretenden Bigramme im Deutschen

Man sollte aber immer im Hinterkopf behalten, dass relative Häufigkeiten von der Länge und Art eines Textes abhängig sind. Die Nachricht „Tarnung intakt. Missionsstart nach Plan am Montag acht Uhr.“ kommt ohne ein einziges E aus. Der französische Schriftsteller Georges Perec schrieb sogar seinen 300-seitigen Roman „La Disparition“, ohne einmal den Buchstaben E zu verwenden, der im Französischen, genau wie im Deutschen statistisch gesehen der häufigste ist.

1.2 Polyalphabetische Chiffrierung

1.2.1 Vigenère-Verschlüsselung

Die Häufigkeitsanalyse soll schon im 7. Jahrhundert benutzt worden sein, um monoalphabetische Verschlüsselungen zu knacken. Man suchte also eine Verschlüsselungstechnik, bei der ein Angriff mittels Häufigkeitsanalyse nicht mehr funktionieren sollte. Eine naheliegende Idee wäre z.B. den häufiger auftretenden Buchstaben mehrere Geheimtextbuchstaben zuzuordnen. Eine andere Idee hatte Blaise de Vigenère, ein französischer Diplomat, im Jahre 1586. Er schlug eine Verschlüsselung vor, die mehrere monoalphabetische Verschlüsselungen miteinander kombiniert.

Bei der Vigenère-Verschlüsselung werden mehrere Cäsar-Verschiebungen angewendet. Ein wiederholt über den Klarext geschriebenes Schlüsselwort gibt an, welcher Cäsar-Schlüssel jeweils verwendet werden soll:

Schlüsselwort:	geh	eimgehe	img	eh	eim	geheim
Klartext:	Wir	treffen	uns	an	der	Kirche.
Geheimtext:	Cmy	xzqljlr	czy	eu	hmd	Qmygpq.

Beim Ver-und Entschlüsseln ist das Vigenère-Quadrat sehr hilfreich:

Dass hier die Häufigkeitsanalyse nicht mehr funktioniert, kann man im Beispiel unten sehen . Links sieht man die Häufigkeitsverteilung der Buchstaben im Klartext, der in deutscher Sprache verfasst

Schlüsselwortbuchstabe \ Klartextbuchstabe	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Tabelle 1.1: Vigenère-Quadrat

wurde, rechts wurde ein Text mit der Technik von Vigenère und dem Schlüsselwort *Schokolade* verschlüsselt.

Ungefähr 300 Jahre lang wurde der Vigenère-Code nicht geknackt. Im Jahre 1854 schließlich fand der Mathematiker Charles Babbage eine Methode, die er allerdings geheim hielt. Neun Jahre später kam der preusische Offizier Friedrich Kasiski auf dieselbe Idee, er veröffentlichte seine Methode, die deshalb den Namen *Kasiski-Test* trägt.
Die wesentliche Aufgabe besteht also darin, die richtige Schlüsselwortlänge zu ermitteln. Kennt man die Schlüsselwortlänge, so kann man in vielen – aber nicht allen Fällen – mit Hilfe von Häufigkeitsanalysen auch das Schlüsselwort herausbekommen und so den Code knacken.

Bestimmung der Schlüsselwortlänge
Ist der Geheimtext im Vergleich zum Schlüsselwort lang, so kann es passieren, dass ein Wort oder N-Gramm mehrmals mit denselben Buchstaben verschlüsselt wird. Bei einem ungünstigen Zusammenspiel von Schlüsselwortlänge und Klartext, kann dies sogar in einem relativ kurzen Text passieren:

```
WIRTREFFENUNSANDERKIRCHEDIRTRAUEICHZUNICHTSANDERENZUVERRATEN
CMYXZQLJLRCZYEUHMDQMYGPQJMYXZMAIPGPLARPGPFYEUHMDKRGYDQXVHXMZ
```

Die Klartextbuchstaben IRTR stehen hier jeweils an denselben Stellen des Schlüsselwortes:

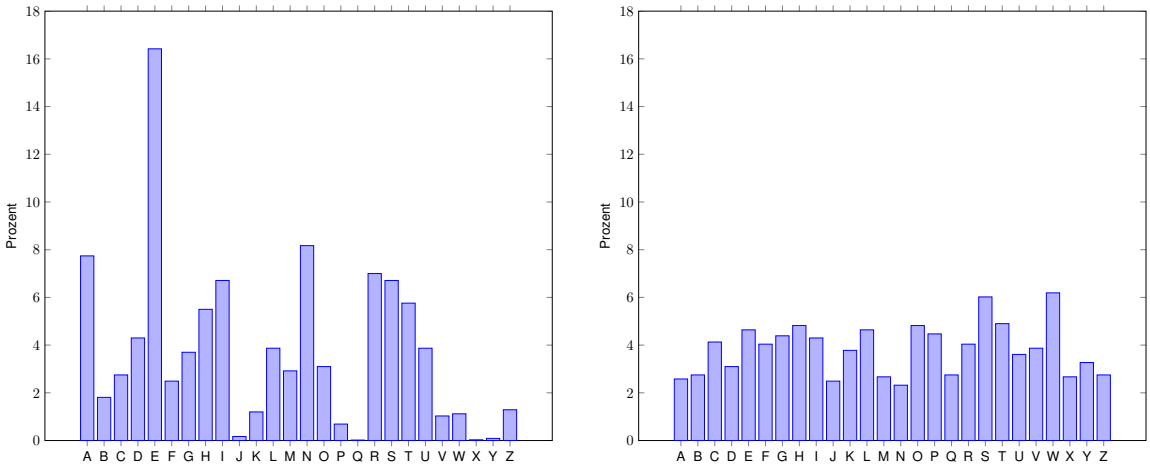


Abbildung 1.4: Vergleich der relativen Buchstabenhäufigkeiten, links Häufigkeitsverteilung im Klartext, rechts Häufigkeitsverteilung des Vigenère-verschlüsselten Textes.

GEHEIMGEHEIMGEHEIMGEHEIMGEHEIMGEHEIMGEHEIMGEHEIMGEHEIMGEHEIM
WIRTREFFENUNSANDERKIRCHEDIRTRAUEICHZUNICHTSANDERENZUVERRATEN
CMYXZQLJLRCZYEUHMDQMYGPQJMYXZMAIPGPLARPGPFYEUHMDKRGYDQXVHXMZ

Das passiert dann, wenn der Abstand dieser Buchstabengruppe ein Vielfaches der Schlüsselwortlänge ist.
Der Kasiski-Test macht sich nun genau dies zu Nutze: Man sucht nach bestimmten Zeichenfolgen im Geheimtext, die sich wiederholen und zählt jeweils ihren Abstand zwischen den Anfängen. In unserem Beispiel beträgt der Abstand der Folge MYXZ $24 = 2 \cdot 2 \cdot 2 \cdot 3$ Zeichen, für EUHMD zählt man $30 = 2 \cdot 3 \cdot 5$ Zeichen und für PGP sind es $6 = 2 \cdot 3$ Zeichen. Man nimmt dann an, dass die Abstände jeweils ein Vielfaches der Schlüsselwortlänge sind. Mit großer Wahrscheinlichkeit ist dann die Schlüsselwortlänge ein gemeinsamer Teiler der Abstände. In unserem Fall ist es sogar der größte gemeinsame Teiler: $ggT(6, 24, 30) = 6$.

Bestimmung des Schlüsselworts
Kennt man die Länge des Schlüsselworts und ist dieses im Verhältnis zum Text nicht zu lang, so kann man nun das Schlüsselwort selbst herausbekommen. In unserem Fall wissen wir, dass die Buchstaben Nr. 1,7,13,19,25,... immer mit demselben Cäsar-Code verschlüsselt wurden. Gleiches gilt für die Buchstaben Nr. 2,8,14,20,26,.... Wir können nun also die Buchstaben in Gruppen zusammenfassen und auf diese Häufigkeitsanalysen anwenden, um jeweils die zugrundeliegende Cäsar-Substitution zu bestimmen. Ist das Schlüsselwort ein lexikalisches und nicht zu lang, hat man in der Regel mit dieser Methode Erfolg. Schwieriger wird es schon, wenn das Schlüsselwort aus einer zufälligen Buchstabenfolge besteht. Wir werden im nächsten Abschnitt hieraus eine unknackbare Verschlüsselungsmethode herleiten.

1.2.2 One-Time-Pad - eine unknackbare Verschlüsselung

Wählt man bei der Vigenère-Verschlüsselung das Schlüsselwort nach drei Regeln, bekommt man eine unknackbare Verschlüsselung:

1. Das Schlüsselwort muss aus genauso vielen Zeichen wie der Klartext bestehen.
2. Die Zeichenfolge des Schlüsselworts muss zufällig sein, sie darf nicht vorhergesagt werden können.

3. Jedes Schlüsselwort darf nur ein einziges Mal verwendet werden.

Hat man z.B. ein vier Zeichen langes Geheimtextwort, so kann man zu jedem vierbuchstabigen Wort einen passenden Schlüssel finden, damit der Geheimtext in dieses Klartextwort entschlüsseln kann. z.B. kann man das Wort SMWH mit dem Schlüssel MYLE zu GOLD entschlüsseln, benutzt man hingegen den Schlüssel MESO , so bedeutet SMWH GIFT.

Doch diese unknackbare Verschlüsselung hat auch einen Nachteil: Sie ist wenig praktikabel. Sender und Empfänger müssen irgendwann zuvor den eventuell sehr langen Schlüssel ausgetauscht haben. Das One-Time-Pad wird vorallem dann eingesetzt, wenn allerhöchste Sicherheit gefordert ist, wie z.B. beim sogennanten roten Telefon, dass 1962 als Verbindung zwischen den obersten Regierungsstellen der USA un der UdSSr eingerichtet wurde.

1.2.3 Mehr zu Permutationen

Wir haben in Abschnitt 1.1.3 Permutationen kennengelernt. Wir wissen, dass man Permutationen miteinander verknüpfen kann und dass alle Permutationen als Verknüpfung von Transpositionen darstellbar sind. Wir wollen nun noch einige wichtige Eigenschaften von Permutationen festhalten, die wir im folgenden Kapitel verwenden werden.

1. Permutationen sind umkehrbar: Zu jeder Permutation π findet man genau eine Permutation π^{-1}, sodass $\pi \circ \pi^{-1} = \pi^{-1} \circ \pi = \mathrm{Id}$ gilt. Man nennt π^{-1} dann auch die zu π *inverse* Permutation.

2. Ein Element, das durch eine Permutation π auf sich selbst abgebildet wird, nennt man einen *Fixpunkt* bezüglich π. Eine Permutation heißt *fixpunktfrei*, wenn kein Element auf sich abgebildet wird.

3. Eine Permutation π heißt *selbstinvers*, wenn $\pi \circ \pi = \mathrm{Id}$ gilt.

4. Transpositionen sind selbstinvers.

1.3 Verschlüsselungsmaschinen - Enigma

Enigma (griech. für Rätsel) ist ein Markenname für elektrische Rotor-Verschlüsselungsmaschinen, deren Verschlüsselungstechnik auf Polyalphabetischer Substitution basiert. Es wurden damals auch andere Verschlüsselungsmaschinen entwickelt, die wie die *Enigma* mit mehreren Rotoren versehen waren. Die besonderen Rolle, die sie im Zweiten Weltkrieg spielte, ist sicherlich ein Grund, warum unter einer Vielzahl solcher Rotormaschinen gerade die *Enigma* so große Berühmtheit erlangte: Von den Deutschen wurde die *Enigma* für unknackbar gehalten. Jedoch gelang es Polen und Engländern gemeinsam, einen großen Teil der von den Deutschen mithilfe einer *Enigma* verschlüsselten Nachrichten zu knacken. Während also die Deutschen davon ausgingen, dass sie ihre Nachrichten erfolgreich geheim hielten, konnten die Alliierten die Funksprüche sehrwohl entziffern. Dies war ein enormer taktischer Vorteil.

Entwickelt wurde die *Enigma* ursprünglich nicht nur für militärische, sondern auch für kommerzielle Zwecke. Der Ingenieur Arthur Scherbius (1878 - 1929) wollte die *Enigma* sowohl im militärischen Bereich als auch in der Industrie vermarkten. 1918 meldete Scherbius das Rotorprinzip, das seinen *Enigma*-Verschlüsselungsmaschinen zugrunde lag, zum Patent an. Es wurde eine Vielzahl verschiedener Modelle gebaut, die sich in der Anzahl der Rotoren und anderen technischen Feinheiten unterschieden.

1.3.1 Aufbau, Funktionsweise

Äußerlich ähnelt die *Enigma* einer Schreibmaschine. Die meisten besitzen eine Tastatur mit 26 Tasten für die Eingabe und ein Lampenfeld mit 26 Lämpchen für die Ausgabe. Betätigt man eine Taste, so fließt über die Verdrahtungen innerhalb der *Enigma* Strom und eines der Lämpchen leuchtet auf. Dabei ist die *Enigma* so konzipiert, dass sich die Verdrahtung ständig ändert. Wir gehen zunächst von einer *Enigma* mit einer Eintrittswalze, drei rotierenden Walzen und einem Reflektor, der oft auch Umkehrwalze genannt wird, aus. Die Verdrahtung innerhalb einer Walze war durch das verwendete *Enigma*-Modell festgelegt. Jedoch wurden oft mehr als fünf verschiedene Walzen mitgeliefert. Der jeweilige Tagesschlüssel bestimmte dann unter anderem, welche drei Walzen in welcher Reihenfolge in die *Enigma* eingesetzt werden sollten. Das Schema (1.1) zeigt den Weg, auf dem der Strom bei gedrückter Taste durch die Enigma fließt:

$$\begin{array}{ccccccccc} & & \text{Walze 3} & \leftarrow & \text{Walze 2} & \rightarrow & \text{Walze 1} & \leftarrow & \text{Eintrittswalze} & \leftarrow & \text{Tastatur} \\ & \swarrow & & & & & & & & & \\ \text{Reflektor} & & & & & & & & & & \\ & \searrow & & & & & & & & & \\ & & \text{Walze 3} & \rightarrow & \text{Walze 2} & \rightarrow & \text{Walze 1} & \rightarrow & \text{Eintrittswalze} & \rightarrow & \text{Lampenfeld} \end{array} \tag{1.1}$$

Die Eintrittzwalze und der Reflektor sind fest, während die drei mittleren Walzen rotieren: Walze 1 dreht sich nach jeder Tastenbetätigung um eine Position weiter. Die Walze 2 und 3 bleiben die meiste Zeit fest. Jedoch ist an der Walze 1 ein Ring mit einem Zahn angebracht, der bei einer bestimmten Position der Walze 1 eine Rotation der Walze 2 auslöst. Man kann sich dies wie beim Kilometerzähler im Auto vorstellen. Auch die zweite Walze ist mit einem solchen Ring ausgestattet, der wiederum die Bewegung von Walze 3 beeinflusst. Die Position dieses Zahns, die sogenannte Ringstellung, kann durch den Anwender beeinflusst werden. Der Reflektor sorgt dafür, dass der Strom zweimal durch jede der Walzen fließt. Durch ihn wird es möglich, dass mit der *Enigma* sowohl ver- als auch entschlüsselt werden kann.

Mathematisch können wir den Verschlüsselungsvorgang mithilfe von Permutationen beschreiben. Wir betrachten dafür eine feste Walzenstellung. Es seien ϵ die Permutation des Klartextalphabets, die durch die Eintrittswalze entsteht, π_1 die Permutation, die durch Walze 1, π_2 die Permutation, die durch Walze 2, und π_3 die Permutation, die durch Walze 3 entsteht. Hinzu kommt noch die Transposition τ durch die Umkehrwalze. Die insgesamt entstehende Permutation des Klartextalphabets kann man dann als Hintereinanderausführung der einzelnen Permutationen gemäß Schema (1.1) schreiben:

$$\pi = \epsilon^{-1} \circ \pi_1^{-1} \circ \pi_2^{-1} \circ \pi_3^{-1} \circ \tau \circ \pi_3 \circ \pi_2 \circ \pi_1 \circ \epsilon$$

Diese Permutation weist zwei Besonderheiten auf. Erstens ist sie selbstinvers:

$$\pi \circ \pi = \epsilon^{-1} \circ \pi_1^{-1} \circ \pi_2^{-1} \circ \pi_3^{-1} \circ \tau \circ \pi_3 \circ \pi_2 \circ \pi_1 \circ \epsilon \circ \epsilon^{-1} \circ \pi_1^{-1} \circ \pi_2^{-1} \circ \pi_3^{-1} \circ \tau \circ \pi_3 \circ \pi_2 \circ \pi_1 = \mathrm{Id}$$

und damit $\pi = \pi^{-1}$. Dies bedeutet aber auch, dass Buchstaben immer paarweise verschlüsselt werden, d.h. drückt man die Taste A und das Lämpchen E leuchtet auf, so hätte das Lämpchen A leuchten müssen, hätte man statt A die Taste E gedrückt. In Permutationsschreibweise ausgedrückt: π(A)=E, so ist π(E)=A. Die zweite Besonderheit ist, dass kein Buchstabe in sich selbst verschlüsselt werden kann, die Permutationen sind fixpunktfrei. Diese Tatsache wurde von den Kryptoanalysten, wie Alan Turing, ausgenutzt. Auch hierzu findet ihr eine Aufgabe auf dem Übungsblatt.

Wie bereits oben erwähnt, gab es verschiedene *Enigma*-Modelle, die während des Zweiten Weltkriegs zum Einsatz kamen. Allerdings mussten jeweils beim Ver- und Entschlüsselungsvorgang baugleiche

Modelle eingesetzt werden, die verschiedenen Modelle waren nicht immer kompatibel. Einige von diesen Modellen waren zusätzlich zu den Walzen mit einem Steckerbrett versehen. Über dieses konnte zusätzlich eine paarweise Vertauschung einzelner Buchstaben erzeugt werden.

Über den jeweiligen Tagesschlüssel wurden die folgenden Einstellungen festgelegt:

- Walzenlage – Auswahl und Anordnung der Walzen
- Grundstellung der Walzen
- Ringstellung
- Verkabelung im Steckerbrett

1.3.2 Mächtigkeit des Schlüsselraums

Wir gehen zunächst von einer *Enigma* ohne Steckerbrett mit drei rotierenden Walzen, drei Ringen und einer Umkehrwalze aus, für die ein Set mit fünf Walzen bereitsteht, aus.

- Man wählt 3 aus 5 Walzen aus, hierfür gibt es also $60 = 5 \cdot 4 \cdot 3$ Möglichkeiten.
- An jeder Walze kann nochmal eine Ringstellung vorgenommen werden, wobei die Stellung des dritten Rings kryptographisch nicht relevant ist. Hier gibt es $26^2 = 676$ Möglichkeiten,
- Grundstellung der Walzen: nochmal $26^3 = 17\,576$ Möglichkeiten.

Hieraus ergeben sich für eine Enigma ohne Steckerbrett: $60 \cdot 26^5 = 60 \cdot 11\,881\,376 = 712\ 882\ 560$ mögliche Schlüssel.

Ist die *Enigma* zusätzlich mit einem Steckbrett versehen, können über Steckerkabel nochmals Buchstaben paarweise vertauscht werden. Man kann dabei $n = 0, ..., 13$ Kabel verwenden. Es gibt 26 Steckbuchsen. Für das erste Buchstabenpaar gibt es daher $26 \cdot 25$ Steckmöglichkeiten, für das zweite $24 \cdot 23$ usw. Beim Einsatz von n Steckerkabeln sind $2n$ Buchstaben betroffen, für das n-te Kabel bleiben noch $(26-2n+2) \cdot (26-2n+1)$ Steckmöglichkeiten übrig. Allerdings muss man noch berücksichtigen, dass aus kryptographischer Sicht einige dieser Möglichkeiten dasselbe Ergebnis liefern: Erstens ist es gleichgültig, in welcher Reihenfolge man die Kabel in die ausgewählten Steckerbuchsen steckt, d.h. ob ich zuerst Kabel 1, dann Kabel 2 oder erst das n-te Kabel setze: Hier entfallen $n!$ Möglichkeiten. Zweitens ist es kryptographisch auch nicht relevant, welches Ende des Steckers in welche Buchse gesteckt wird, d.h. ob ich z.B. (AC) oder (CA) bilde. Dies ergibt 2^n gleichwertige Kombinationen.

Verwendet man also genau n Kabel, so kommt man insgesamt auf

$$\frac{1}{2^n} \cdot \frac{1}{n!} \cdot \frac{26!}{(26-2n)!}$$

kryptographisch relevante Steckmöglichkeiten.

Die Deutschen verwendeten meist eine feste Anzahl an Steckerkabeln, zunächst sechs, ab 1939 in der Regel zehn Steckerkabel. Bei einer Verwendung von zehn Steckerkabeln bedeutet dies eine Vergrößerung des Schlüsselraums um den Faktor $150\,738\,274\,937\,250$ auf

$$60 \cdot 26^5 \cdot 150\,738\,274\,937\,250 = 107\,458\,687\,327\,250\,619\,360\,000 \approx 1.5 \cdot 10^{23}$$

mögliche Schlüssel. Tatsächlich verkleinerten die Deutschen durch diese Festlegung auf eine bestimmte Anzahl an Steckerkabeln den Schlüsselraum der *Enigma*.

1.3.3 Kryptographische Schwächen

Wie wir bereits bei der Monoalphabetischen Substitution – dort war der Schlüsselraum mit $4 \cdot 10^{26}$ sogar noch größer – gesehen haben, ist die Größe des Schlüsselraums nicht ausreichend, um die Sicherheit eines Verschlüsselungsverfahrens gewährleisten zu können.

Im Zusammenhang mit der *Enigma* setzten Fehler in verschiedenen Bereichen die Sicherheit deutlich herab:

- Konstruktion der Maschine
- Vorschriften zur Verwendung
- Faktor Mensch: Spionage, Anwendungsfehler

Konstruktion der Maschine

Die Bauweise der Maschine hatte wichtige Auswirkungen auf das von ihr durchgeführte Verschlüsselungsverfahren.

Schon vor Kriegsbeginn waren kommerzielle Maschinen der Marke *Enigma* auf dem freien Markt erhältlich gewesen. Zwar wurden im militärischen Bereich Rotoren mit einer anderen Verdrahtung verwendet, die grundsätzliche Funktionsweise konnte jedoch untersucht werden. Im Jahre 1931 verkaufte Hans-Thilo Schmidt, ein Mitarbeiter der Chiffrierstelle des Reichswehrministeriums, Informationen über die *Enigmamodelle* an den französischen Geheimdienst, darunter eine „Gebrauchsanweisunge für die Chiffriermaschine *Enigma*“ und eine „Schlüsselanleitung für die Chiffriermaschine *Enigma*“. Dank eines Abkommens zwischen Frankreich und Polen, wurden diese Informationen an Polen weitergegeben. Nach den darin enthaltenen Plänen konnte ein Nachbau der darin beschriebenen *Enigma* angefertigt werden. Dies half den polnischen Kryptologen, den Verschlüsselungsvorgang genauer zu untersuchen und Schwächen darin zu finden. Zusätzlich hatten sie einige der Schlüsselbücher erhalten. Eine dieser Schwächen bestand in der Umkehrwalze: Wie oben bereits erwähnt, sorgt sie dafür, dass alle Permutationen des Klartextes, die durch die Enigma entstehen, selbstinvers sind. Zudem sind die Verdrahtungen in der Enigma so gelegt, dass kein Buchstabe in sich selbst verschlüsselt werden kann, die Permutationen sind nicht nur selbstinvers, sondern zudem fixpunktfrei. Dies reduziert die Anzahl der möglichen Permutationen deutlich. Wie stark diese Einschränkung ist, dürft ihr euch auf dem Übungsblatt überlegen.

Hätten die polnischen und englischen Kryptoanalytiker keine Möglichkeit gehabt, den Verschlüsselungsvorgang der *Enigma* genauer zu untersuchen, so hätten sie wahrscheinlich keine Methoden gefunden, die Verschlüsselung zu knacken. Die Verschlüsselung durch die *Enigma* verstößt damit gegen ein wichtiges Prinzip, dessen Gültigkeit für die heutzutage verwendeten Verschlüsselungsverfahren gefordert wird. Formuliert wurde dieses Prinzip allerdings schon lange vor dem Zweiten Weltkrieg im Jahre 1883 von Auguste Kerckhoff, einem niederländischen Linguist und Kryptologen:

Kerckhoffs Prinzip
Die Sicherheit eines Verschlüsselungsverfahrens darf nicht auf der Geheimhaltung des Verfahrens beruhen, sondern ausschließlich auf der Geheimhaltung des Schlüssels.

Vorschriften zur Verwendung

Um die jeweiligen Tagesschlüssel zu vereinbaren, wurden monatlich Schlüsselbücher an die einzelnen Stellen verteilt. Um aber ein häufiges Senden mit demselben Schlüssel zu verhindern, wurden am Anfang jeder Nachricht ein spezieller Nachrichtenschlüssel übermittelt,der die genaue Walzenstellung

angab. Dazu wurde dieser mit dem jeweiligen Tagesschlüssel verschlüsselt. Dabei machten die Deutschen jedoch den Fehler, diesen Nachrichtenschlüssel sicherheitshalber immer zweimal hintereinander zu schreiben. Dies fiel dem polnischen Mathematiker Marian Rejewski auf, der bereits vor Ausbruch des Zweiten Weltkriegs daraus eine Methode entwickelte, den Tagesschlüssel zu ermitteln. Wir werden später noch etwas genauer darauf eingehen. Weitere Fehler machten die Deutschen, wie oben bereits erwähnt, bei der Verwendung des Steckerbretts, indem sie meist eine feste Zahl (10) an Steckern verwendeten und bestimmten, dass kein Buchstabe mit seinem Nachfolger verbunden werden durfte. Weiter machten sie Vorschriften zu den Walzen: keine Walzenlage durfte zweimal im Monat und keine Walze an zwei aufeinander folgenden Tagen in derselben Position verwendet werden. Hier konnten die Kryptoanalytiker also bereits verwendete Walzenlagen aus ihren Untersuchungen ausschließen. Auch gab es spezielle Sprachregelungen und Sendezeiten für gewisse militärische Nachrichten, die teilweise auch noch auf unterschiedliche Art und Weisen übermittelt wurden. Die Kryptoanalytiker konnten also ein Repertoire an wahrscheinlichen Klartextwörtern aufstellen, die ihnen dann beim Entschlüsseln halfen.

Faktor Mensch

Ein wesentlicher Faktor war sicherlich die Weitergabe der Informationen durch Hans-Thilo Schmidt sowie die Gefahr, dass Schlüsselbücher in die falschen Hände gerieten. Tatsächlich war es aber auch die Nachlässigkeit einzelner Anwender, die den Kryptoanalytiker zusätzliche Einsichten und Angriffspunkte bot: Häufig wurden einfache Tastenkombinationen wie AAA oder nebeneinanderliegende Tasten als Nachrichtenschlüssel oder gleich mehrmals ein und derselbe Schlüssel verwendet.

1.3.4 Ermittlung des Tagesschlüssels

Der Tagesschlüssel hatte drei wesentliche Bestandteile: Die Angabe der Steckverbindungen, die Auswahl und Anordnung der Walzen sowie die Anfangsstellung der Walzen. z.B. (A,L), (K,D), (P,E) ..., II-III–I und L–S–M. Wurde nun eine Nachricht gesendet, so wurden an der *Enigma* zunächst die Einstellungen gemäß dem Tagesschlüssel vorgenommen. Der Sender dachte sich dann wiederum eine neue Anfangsstellung für die Walzen aus z.B. T–F–G. Dies ergab den spezifischen Nachrichtenschlüssel, welchen der Sender zu Beginn einer Nachricht zweimal hintereinander noch mit der ursprünglichen Tagesschlüssel-Einstellungen in die *Enigma* eingab. Anschließend stellte er dann die Walzenstellung auf T–F–G um und begann mit der Verschlüsselung der eigentlichen Nachricht.

Marian Rejewski wusste also dass jeweils der 1. und der 4., der 2. und der 5. sowie der 3. und der 6. Buchstabe einer Nachricht im Klartext übereinstimmten. Zudem wusste er, dass die ersten sechs Buchstaben der Nachrichten, die er an einem Tag abfing, mit demselben Tagesschlüssel verschlüsselt waren:

Nachricht 1	L	O	K	R	G	M
Nachricht 2	M	V	T	X	Z	E
Nachricht 3	J	K	T	M	P	E
Nachricht 4	D	V	Y	P	Z	X

Unbekannt hingegen waren ihm die Walzenlage, die Walzenstellung und die Steckbrettverbindung. Er stellte zunächst eine Beziehungstabelle auf, in der er immer den ersten und vierten Buchstaben eines Nachrichtenbeginns in einen Zusammenhang brachte:

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

Da an einem Tag relativ viele Nachrichten gesendet wurden, gelang es ihm meist, die Beziehungstabelle aufzufüllen:

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

Genauso konnte er eine Beziehungstabelle für den zweiten und den fünften sowie den dritten und den sechsten Buchstaben aufstellen. Er untersuchte dann einzelne Ketten, jeweils innerhalb einer Beziehungstabelle:

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

und hielt deren Länge fest. In diesem Beispiel wären es die Längen 4, 9, 7, 7. Tauscht man z.B. über das Steckerbrett die Buchstaben Q und H, so sieht die neue Beziehungstabelle wie folgt aus:

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

In den Ketten wirkt sich dies folgendermaßen aus:

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

Rejewski stellte fest, dass die Länge dieser Ketten ausschließlich von der Walzenlage und -stellung beeinflusst wurde, jedoch vom Steckerbrett unabhängig waren. Gerade aber die Walzenlage und Anfangsstellung zu kennen, war das Entscheidende. Ein ganzes Jahr lang katalogisierten Rejewski und seine Kollegen die Walzeneinstellungen und die resultierenden Kettenlängen, die sie mit ihrer nachgebauten *Enigma* ermittelten. Mithilfe des Katalogs konnten sie dann die Walzenlage und -stellung des jeweiligen Tagesschlüssel ermitteln und die Nachrichten soweit in einen Text umwandeln, dass dieser sich vom Klartext nur noch durch vertauschte Buchstaben unterschied. Später, als die deutschen zusätzliche Walzen zum Auswechseln hinzufügten, entwickelte Rejewski die Bombe: ein Verbund von mehreren *Enigma*-Adaptionen, die automatisiert nach dem jeweiligen Tagesschlüssel suchten. 1938 wurden die Walzensätze mit 5 Walzen eingeführt und die Anzahl der Kabel im Steckbrett erhöht, die polnischen Kryptologen gaben ihr Wissen an englische und französische weiter, die auf größere Resourcen zurückgreifen konnten.

Ermuntert von den polnischen Erfolgen beim Knacken der *Enigma*-Verschlüsselung, wurde auch in Großbritannien, genauer in Bletchley-Park, fleißig an der Entschlüsselung deutscher Nachrichten gearbeitet. Außerdem wurden neue Methoden gesucht, die Verschlüsselung zu brechen. Zu Recht befürchtete man, dass die Deutschen auf ihren Fehler aufmerksam werden und es in Zukunft unterlassen könnten, den Nachrichtenschlüssel mehrfach zu senden. Alan Turing kam auf die Idee, die Fixpunktfreiheit der Verschlüsselung auszunutzen. Auch fiel ihm beim Vergleich bereits entschlüsselter Nachrichten auf, dass sehr ähnliche Nachrichten z.B. der Wetterbericht oft zu festgelegten Zeiten gesendet wurden und typische Wörter wie „Wetter“ enthielten. Diese Wörter bezeichnete man als „Cribs“. Da ein Buchstabe nie in sich selbst verschlüsselt wurde, konnte man ein erwartetes Wort einfach über

den Geheimtext schieben und für dessen Position schon einmal alle Stellen ausschließen, an denen ein Klartextbuchstabe in sich selbst verschlüsselt werden würde. So konnte man schnell einen Großteil der möglichen Tagesschlüssel streichen. Turing verfeinerte seine Suche zudem auf spezielle Cribs: Auch er suchte nach Zeichenketten, jedoch standen diese nicht wie bei Rejewski in Zusammenhang mit dem wiederholten Senden des Nachrichtenschlüssels. Ihm gelang es einen Zusammenhang zwischen Steckerverbindungen und Zeichenketten herzustellen. Er entwickelte dann die sogenannte *Turing-Bombe*, die alle noch verbleibenden möglichen Walzenstellungen- und lagen durchprobieren konnte. Dafür schaltete er zunächst drei *Enigma*-Machinen zusammen, deren Grundstellung versetzt eingestellt war. Um alle 60 Möglichkeiten, die durch die Auswahl von drei aus fünf Rotoren zustande kommen, gleichzeitig testen zu können, kombinierte Turing 60 solcher Dreier-Enigmas zu einer großen Maschine: die *Turing-Bombe*. Hatte man den richtigen Crib, so fand die Maschine innerhalb einiger Stunden den Tagesschlüssel.

Kapitel 2

Moderne Kryptographie

Durch den alltäglichen Einsatz von Computern und mit dem Einzug des Internets in unseren Alltag ist die Sicherheit und der Schutz der Privatsphäre Einzelner zu einem Brennpunktthema geworden. Die Zahl der täglich gesendeten Nachrichten und die Menge an gespeicherten Daten nimmt ständig zu. Wir sind an E-Mails, Onlinebanking, E-Commerce etc. gewöhnt – in den meisten Fällen bemerken wir als Anwender dabei gar nicht mehr, dass im Hintergrund ständig Ver- und Entschlüsselungsprozesse ablaufen. Wir erwarten heutzutage, dass wir gleichzeitig spontan und sicher kommunizieren können, wenn wir z.B. bei einem Online-Handel einkaufen möchten. Dass heute solch eine spontane Kommunikation mit gleichzeitigem Schutz der Privatsphäre möglich ist, haben wir Menschen wie z.B. Ralph Merkle, Martin Hellman und Whitfield Diffie zu verdanken, die bereits in den 70er Jahren – das Internet steckte noch in den Kinderschuhen – die digitale Revolution vorausahnten. Die damals bekannten Verschlüsselungsverfahren waren – wie alle Verfahren, die auch wir bis jetzt behandelt haben – symmetrische Verschlüsselungsverfahren, d.h. Verfahren, bei denen für Ver- und Entschlüsselung derselbe Schlüssel verwendet wird. Dies setzt jedoch voraus, dass Sender und Empfänger zuvor einen Schlüssel vereinbart und vor Dritten geheimgehalten haben. Gerade die Schlüsselübermittlung bzw. - vereinbarung stellt jedoch eine große Schwachstelle aller symmetrischen Verfahren dar: Codebücher geraten leicht in Feindeshand, ein persönliches Treffen ist nicht immer möglich oder einfach zu umständlich. In den 70-er Jahren glaubten einige Visionäre daran, dass Computer in der Zukunft auch im privaten Bereich zum Einsatz kommen würden und erahnten, dass es irgendwann eine weltweite Vernetzung geben würde. Die Kryptologen unter ihnen erkannten allerdings ein schwerwiegendes Problem: Wenn schon für Militär und Firmen der Schlüsselaustausch ein großes Problem darstellte, wie sollte dies sogar für Privatleute gelöst werden? Auch die Zahl der Nachrichten und die Menge an Daten würde sicherlich zunehmen. Über 2000 Jahre war das Problem des Schlüsselaustauschs nicht gelöst worden – welche unterschiedlichen Ideen und Ansätze Diffie, Hellman und andere Kollegen im 20. Jahrhundert entwickelten, um das Problem des Schlüsselaustausches zu lösen, soll Thema des folgenden Kapitels sein.

2.1 Das Schlüssel-Problem

Wendet man symmetrischen Verschlüsselungsverfahren an, entsteht in großen Netzwerken mit vielen Teilnehmern ein riesiger Bedarf an Schlüsseln: Möchten N Teilnehmer paarweise untereinander vertrauliche Botschaften austauschen können, werden schon $\frac{N(N-1)}{2}$ Schlüssel benötigt. Kommt ein neues Mitglied im Netzwerk hinzu, muss dieses wiederum mit allen N Teilnehmern jeweils einen neuen Schlüssel vereinbaren. Dies ist nicht nur aufwendig, sondern es bringt ein neues Problem mit sich: Wie kann der neue Teilnehmer mit jedem einzelnen der anderen Teilnehmer einen Schlüssel verabreden und dies am besten möglichst schnell und sicher?

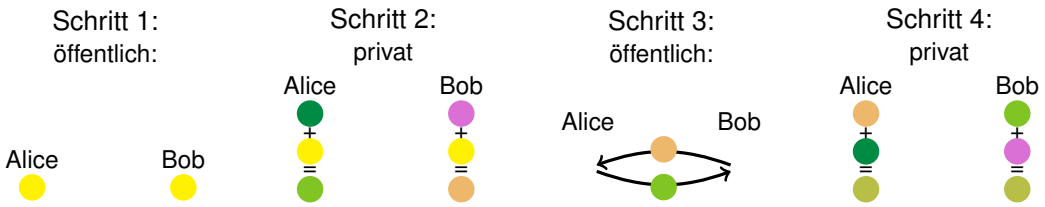
Auf der Suche nach einer Lösung des Problems, bildeten sich zunächst zwei zentrale Fragen heraus:

- Können zwei Personen in aller Öffentlichkeit ein gemeinsames Geheimnis erzeugen?
- Ist es möglich, ein sicheres Verschlüsselungsverfahren zu finden, welches nicht symmetrisch ist?

Wir werden sehen, dass beide Fragen mit „ja“ beantwortet werden können.

2.1.1 Ein Geheimnis entsteht

Alice und Bob wollen ihr Zimmer in genau demselben Farbton streichen, sonst soll vorher niemand erfahren, welcher Farbton dies ist. In aller Öffentlichkeit machen sie aus, dass sie zunächst beide einen Liter derselben gelben Farbe in einen Eimer geben. Anschließend mischt jeder der beiden einen Liter eines weiteren Farbtons hinzu, den er vor allen geheim hält, Alice wählt Frühlingsgrün und Bob entscheidet sich für Orchidee. Daraufhin tauschen Alice und Bob die entstandenen Farbgemische aus, was für jeden zu sehen ist. Zurück in ihrem geheimen Kämmerchen gibt wieder jeder einen Liter seiner Farbe, die er niemandem preisgegeben hat, in den erhaltenen Farbeimer und mischt diese gut durch. Bob weiß nur, dass er Orchidee verwendet hat, aber nicht, welche Farbe Alice verwendet hat, genauso geht es Alice. Beide haben aber nun drei Liter Farbe in exakt demselben Farbton. Ein möglicher Angreifer kennt zwar die urspüngliche Farbe gelb und die Farbtöne der ausgetauschten Farbe, er kann davon aber nicht auf die beiden im Geheimen hinzugefügten Farben und somit auf den entgültigen Farbton schließen.



Es ist also grundsätzlich möglich, in aller Öffentlichkeit ein Geheimnis zu erschaffen. Die Frage ist nun, wie man dies in ein kryptographisches Verfahren übertragen kann.

2.1.2 Einwegfunktionen

Von großer Bedeutung im obigen Beispiel ist, dass der Vorgang des Farbemischens sehr einfach ist, während es unmöglich ist, diesen wieder rückgängig zu machen. So eine Vorschrift, die leicht durchzuführen, aber nicht mehr rückgängig zu machen ist, nennt man auch eine Einwegfunktion. Man kann aus kryptographischer Sicht auch dann von einer Einwegfunktion sprechen, wenn es nur ausreichend lange braucht, um die Wirkung einer Funktion rückgängig zu machen, da dies in vielen Fällen bereits ausreicht, um ein gewünschtes Sicherheitsniveau zu gewährleisten. Weiter ist der Vorgang des Farbemischens *kommutativ*: Das Ergebnis der Farbmischung hängt nicht davon ab, ob zuerst Alice und dann Bob, oder erst Bob und dann Alice jeweils ihre geheime Farbe zur gelben Farbe dazumischen.

Auf der Suche nach geeigneten Einwegfunktionen wurde plötzlich ein mathematisches Gebiet interessant, das bisher zwar viele Mathematiker und Mathematikinteressierten besonders fasziniert hatte, jedoch immer als sehr anwendungsfern gegolten hatte: Die Zahlentheorie.

2.2 Zahlentheoretische Grundlagen

2.2.1 Teilbarkeit, Größter gemeinsamer Teiler und der euklidische Algorithmus

Definition 2.1
Seien $n, d \in \mathbb{N}$. Existiert ein $t \in \mathbb{N}$ mit $d \cdot t = n$, dann heißt d ein Teiler von n. Kurzschreibweise: $d|n$. Die Zahl n heißt dann ein Vielfaches von d.

Für alle Teiler d von n gilt: $d \leq n$. Somit ist die Menge aller Teiler einer Zahl endlich. Außerdem ist sie nichtleer, denn die 1 ist Teiler jeder natürlichen Zahl.

Definition 2.2 Für zwei natürliche Zahlen n, m nennt man $\{t \in \mathbb{N} : t|n \text{ und } t|m\}$, die Menge der gemeinsamen Teiler von n und m. Das größte Element dieser Menge bezeichnet man als den *größten gemeinsamen Teiler* von m und n und notiert dafür $ggT(n, m)$. Zwei natürliche Zahlen heißen teilerfremd, wenn der einzige gemeinsame Teiler die Zahl 1 ist.

Bemerkung 2.3 Die 1 ist immer ein gemeinsamer Teiler zweier natürlicher Zahlen. Außerdem besitzt die Menge aller gemeinsamen Teiler immer ein größtes Element, d.h. der ggT existiert immer. Somit könnte man in Definition 2.2 auch schreiben: Zwei natürliche Zahlen a, b heißen teilerfremd, wenn $\mathrm{ggT}(a, b) = 1$ ist.
Außerdem sei angemerkt, dass sich die Sätze hier alle auf die ganzen Zahlen verallgemeinern lassen. ◇

Wir wollen noch eine wichtige Eigenschaft von Teilern festhalten:

Satz 2.4 *Sei $a > b$ und d ein gemeinsamer Teiler von a und b, dann teilt d auch die Differenz $a - b$.*

Beweis: Aus $d|a$ und $d|b$ folgt, dass es $s, t \in \mathbb{N}$ gibt, sodass $d \cdot s = a$ und $d \cdot t = b$ gilt. Es ist dann $a - b = d \cdot s - d \cdot t = d \cdot (s - t)$. Damit folgt $d|(a - b)$.

Satz 2.5 *Seien $a, b \in \mathbb{N}$, $a > b$. Dann gibt es eindeutig bestimmte Zahlen $q, r \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ mit $r < b$, sodass gilt:*

$$a = q \cdot b + r.$$

Beweis: Den Beweis kann man z.B. auf S.19 in [Bur05] nachlesen.

Euklidischer Algorithmus

Diese beiden Sätze kann man benutzen, um einen Algorithmus zu finden, mit dessen Hilfe man den größten gemeinsamen Teiler zweier natürlicher Zahlen a und b bestimmen kann. Wir stellen diesen zuerst an zwei Beispielen vor:

Beispiel 2.6

(a) Gesucht ist der ggT(8,12):

$$\begin{aligned} a_0 &:= 12, \quad a_1 := 8 \\ a_2 = a_0 - k_1 \cdot a_1 &= 12 \quad -1 \cdot 8 = 4 \\ a_3 = a_1 - k_2 \cdot a_2 &= 8 \quad -2 \cdot 4 = 0 \end{aligned}$$

Es ist ggT(12, 8)=4.

(b) Gesucht ist der ggT(263,124):

$$a_0 := 263, \quad a_1 := 124$$

$$\begin{aligned}
a_2 &= a_0 - k_1 \cdot a_1 &&= 263 && -2 \cdot 124 = 15 \\
a_3 &= a_1 - k_2 \cdot a_2 &&= 124 && -8 \cdot 15 \;\; = 4 \\
a_4 &= a_2 - k_3 \cdot a_3 &&= 15 && -3 \cdot 4 \;\; = 3 \\
a_5 &= a_3 - k_4 \cdot a_4 &&= 4 && -1 \cdot 3 \;\; = 1 \\
a_6 &= a_4 - k_5 \cdot a_5 &&= 3 && -3 \cdot 1 \;\; = 0
\end{aligned}$$

Also gilt ggT(263, 124)=1.

△

Etwas allgemeiner formuliert sieht das Verfahren zur Bestimmung des ggT(a, b) mit $a > b$ so aus:

Setze $a_0 := a$, und $a_1 := b$. Suche nun die größtmögliche Zahl $k_1 \in \mathbb{N}$, sodass gilt:

$$0 \leq a_2 := a_0 - k_1 a_1 < a_1.$$

Nun muss man zwei Fälle unterscheiden:

1. Fall: $a_2 = 0$. Dann ist $a_0 - k_1 a_1 = 0$, also ist a_1 ein Teiler von a_0. Somit ist $\mathrm{ggT}(a_0, a_1) = a_1$. und man hat den ggT schon gefunden.
2. Fall: $a_2 \neq 0$. Dann wählt man $k_2 \in \mathbb{N}$ analog zu k_1 derart, dass gilt:

   $$0 \leq a_3 := a_1 - k_2 a_2 < a_2.$$

   und macht wieder dieselbe Fallunterscheidung für a_3. Solange dabei der zweite Fall eintritt, berechnet man $a_4, a_5, \ldots a_i$ gemäß

   $$0 \leq a_i := a_{i-2} - k_{i-1} a_{i-1} < a_{i-1}.$$

bis schließlich $a_{i+1} = 0$ gilt. Dann ist a_i ein Teiler von a_{i-1} und

$$a_i = \mathrm{ggT}(a_i, a_{i-1}) = \mathrm{ggT}(a, b).$$

Satz 2.7 *Seien $a, b \in \mathbb{N}$, dann gibt es zwei Zahlen $c, d, \in \mathbb{Z}$, sodass gilt:*

$$\mathrm{ggT}(a, b) = c \cdot a + d \cdot b.$$

Beweis: Den Beweis kann man z.B. auf S.25 in [Bur05] nachlesen.

Erweiterter Euklidischer Algorithmus

Indem wir ihn etwas erweitern, können wir mit dem Euklidischen Algorithmus auch die ganzen Zahlen c und d aus Satz 2.7 finden:

Beispiel 2.8 Wir wissen aus Beispiel 2.6, dass ggT$(263, 124) = 1$ ist. Jetzt rechnen wir sozusagen rückwärts:

$$\begin{aligned}
1 &= 4 - 1 \cdot 3 && = 4 - 1 \cdot (15 - 3 \cdot 4) \\
&= 4 \cdot 4 - 1 \cdot 15 && = 4 \cdot (124 - 8 \cdot 15) - 1 \cdot 15 \\
&= 4 \cdot 124 - 33 \cdot 15 && = 4 \cdot 124 - 33 \cdot (263 - 2 \cdot 124) \\
&= -33 \cdot 263 + 70 \cdot 124
\end{aligned}$$

Somit gilt $1 =$ ggT$(263, 124) = -33 \cdot 263 + 70 \cdot 124$. In diesem Fall wäre also $c = -33$ und $d = 70$. △

2.2.2 Primzahlen

Definition 2.9
Eine Primzahl ist eine natürliche Zahl $p > 1$, die sich nicht als Produkt zweier kleinerer natürlicher Zahlen schreiben lässt.

Im Folgenden schreiben wir für die Menge der Primzahlen kurz $\mathbb{P}$.

$$\mathbb{P} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...\}.$$

Bisher kennt man keine explizite Formel, mit der man Primzahlen berechnen könnte. Effiziente Primzahltests, die schnell feststellen können, ob sehr große Zahlen prim sind, funktionieren bisher nur indirekt, d.h. sie können höchstens ausschließen, dass die getestete Zahl eine Primzahl ist. Beispielsweise weiß man, dass 2 die einzige *gerade* Primzahl ist, so kann man alle geraden Zahlen, die größer als zwei sind, schon einmal als Primzahlkandidaten ausschließen.

Satz 2.10 *Fundamentalsatz der Zahlentheorie*
Jede natürliche Zahl n lässt sich als Produkt von Primzahlen schreiben. Sortiert man die Primfaktoren der Größe nach, so ist die Darstellung eindeutig.

So ist z.B. $1050 = 2 \cdot 3 \cdot 5 \cdot 5 \cdot 7$ oder $15283 = 17 \cdot 29 \cdot 31$.

Beweis: Den Beweis kann man u.a. in [Bur05], S. 49 nachlesen.

□

Euklid lebte ca. 300 Jahre vor Christus und lieferte einen Beweis des folgenden Satzes:

Satz 2.11 *(Euklid)*
Es gibt unendlich viele Primzahlen.

Beweis:
Wir betrachten eine endliche Menge von Primzahlen $P = \{p_1, p_2, \ldots, p_k\},\ k \in \mathbb{N}$ und die Zahl

$$N = p_1 \cdot p_2 \cdots p_k + 1.$$

Wegen $N - p_1 \cdot p_2 \cdots p_k = 1$ ist keines der Elemente aus P ein Teiler von N. Nach dem Fundamentalsatz der Zahlentheorie ist N jedoch als Produkt von Primzahlen darstellbar. Damit folgt, dass es mindestens $k + 1$ Primzahlen geben muss. Da wir $k \in \mathbb{N}$ beliebig groß wählen können, folgt die Behauptung. □

Im Laufe der Zeit hat man viele sehr unterschiedliche Beweise für Euklids Primzahlsatz gefunden.

Große Primzahlen spielen in modernen kryptographischen Verfahren eine zentrale Rolle, so ist z.B. das Multiplizieren zweier großer Primzahlen – man denkt an Zahlen mit ca. 300 Stellen – eine Einwegfunktion. Denn es ist sehr schwer, aus dem Produkt die einzelnen Faktoren zu finden. Nach bisherigem Kenntnisstand wird das Faktorisieren großer Zahlen als ein *schwer lösbares Problem* angesehen.

2.2.3 Gruppen

In diesem Abschnitt wollen wir mit Gruppen einen außerordentlich erfolgreichen Begriff der modernen, abstrakten Mathematik verstehen. Neben aus dem Mathematikunterricht bekannten Gruppen haben wir in den vorangegangenen Abschnitten bereits mit den Permutationen und dem Rechnen mit Resten zwei wichtige Beispiele für Gruppen gesehen. Um zu erklären, was eine Gruppe ist, bietet es sich an, ein wenig auszuholen. Daher ist der Definition der folgende Abschnitt über Verknüpfungen vorangestellt.
Eine Gruppe besteht aus einer Menge mit einer Verknüpfung, die gewisse Eigenschaften erfüllt. Beispielsweise sind die ganzen Zahlen (als Menge: $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$) mit „$+$“ (das ist die Verknüpfung!) eine Gruppe $(\mathbb{Z}, +)$.

2.2.4 Verknüpfungen

Verknüpfungen gibt es sehr viele und nur wenige davon sind die Verknüpfung einer Gruppe. Die einzige Anforderung an eine Verknüpfung ist, dass sie zu je zwei Elementen a, b einer Menge M ein drittes Element c der gleichen Menge M produziert:

So wie Plus mit „$+$“ bezeichnet wird, gibt es auch für andere Verknüpfungen Symbole, zum Beispiel $*$, $\circ$, $\oplus$, $\cdot$, $\times$ (und viele mehr). Im Folgenden wollen wir für eine nicht genauer festgelegte Verknüpfung beispielhaft $\circ$ schreiben. Wenn nun a und b mittels $\circ$ verknüpft werden, dann wird das Ergebnis mit $a \circ b$ bezeichnet.
Um eine Verknüpfung formal korrekt definieren zu können, braucht man das sogenannte *kartesische Produkt*. Für zwei Mengen A, B ist dies die Menge der Paare:

$$\{(a, b) \colon a \in A, b \in B\} = A \times B.$$

Das kartesische Produkt der Mengen A und B wird – wie oben bereits angedeutet – mit $A \times B$ bezeichnet.

Definition 2.12 (Verknüpfung)
Sei M eine Menge. Eine *Verknüpfung* auf M ist eine Zuordnung $\circ$, die jedem Element (a, b) der Menge $M \times M$ genau ein Element $a \circ b$ der Menge M zuordnet.

Beispiel 2.13

1. Alle Grundrechenarten auf passenden Zahlbereichen sind Beispiele für Verknüpfungen. So ist „$+$“ zwar eine Verknüpfung auf $\mathbb{N}$, aber „$-$“ nicht, weil etwa $2 - 3$ kein Element von $\mathbb{N}$ ist.

2. Sei $\mathcal{T}$ die Menge der Wörter, das heißt, die Elemente von $\mathcal{T}$ sind Folgen von Buchstaben wie zum Beispiel *djwhwue* oder *abababa*. Auf $\mathcal{T}$ ist durch hintereinander schreiben eine Verknüpfung gegeben.

3. Sei $\mathcal{F}$ die Menge der Farben. Wie man Farben konkret aufschreiben kann, soll uns hier jetzt nicht interessieren. Zwei Farben können nun zu einer dritten Verknüpft werden, indem man sie zu gleichen Teilen mischt.

4. Sei $\mathcal{A}$ die Menge der Aussagen. Das ist etwas vage. Elemente von $\mathcal{A}$ sollen zum Beispiel sein:

   Das Leben ist schön
   Wenn es regnet ist die Straße nass
   Nachts sind alle Katzen grau und Schafe machen mäh

   Auf dieser Menge gibt es sehr viele interessante Verknüpfungen*, zum Beispiel „und“ und „oder“, wobei „und“ zwei Aussagen A und B verknüpft, indem „und“ dazwischen geschrieben wird. Wenn man die beiden ersten Aussagen von oben mit „und“ verknüpft, erhält man also die Aussage: „Das Leben ist schön und wenn es regnet ist die Straße nass. “

△

2.2.5 Eigenschaften von Verknüpfungen

Verknüpfungen können sehr viele Eigenschaften haben. So ist etwa aus der Schule die Rechenregel $a + b = b + a$ bekannt – die Gleichung kann auch für eine beliebige Verknüpfung $\circ$ auf einer Menge M hingeschrieben werden: $a \circ b = b \circ a$. Wenn die Gleichung für alle Elemente a, b von M gilt, dann sagt man, dass die Verknüpfung $\circ$ kommutativ ist.

Definition 2.14 Sei $\circ$ eine Verknüpfung auf einer Menge M.

1. $\circ$ heißt *assoziativ*, wenn für alle a, b, c aus M gilt:

   $$a \circ (b \circ c) = (a \circ b) \circ c$$

2. $\circ$ heißt *kommutativ*, wenn für alle a, b aus M gilt:

   $$b \circ a = a \circ b$$

3. Ein Element $e \in M$ heißt *Neutralelement* bzgl. $\circ$, wenn für alle $a \in M$ gilt:

   $$a \circ e = a = e \circ a$$

   (Ein Neutralelement einer Verknüpfung auf M wird auch oft e_M genannt)

4. Falls es ein Neutralelement bzgl. $\circ$ gibt, heißt ein Element $b \in M$ *invers* zu $a \in M$, wenn

   $$a \circ b = e = b \circ a$$

5. Das Paar $(M, \circ)$ heißt *Gruppe*, wenn $\circ$ assoziativ ist, es ein Neutralelement in M gibt und es zu jedem Element von M ein Inverses gibt.

An bereits bekannten Beispielen gibt es neben $(\mathbb{Z}, +)$ noch die Erweiterungen auf größere Zahlbereiche: $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$ und $(\mathbb{C}, +)$[†]. Auch die Multiplikation liefert bei entsprechender Einschränkung der zugrundeliegenden Mengen Gruppen: $(\{-1, 1\}, \cdot)$, $(\mathbb{Q} \setminus \{0\}, \cdot)$ und $(\mathbb{R} \setminus \{0\}, \cdot)$.

*Die beiden folgenden Verknüpfungen sind leider nur welche, wenn man grammatikalisch eher fragwürdige Sätze noch in die Menge $\mathcal{A}$ aufnimmt.

†Falls man damit was anfangen kann. Wird im Folgenden nicht benötigt.

2.2.6 Beweise

Wir werden nun Aussagen über Gruppen beweisen. Das heißt, es werden in den folgenden Beweisen nur die abstrakten Eigenschaften verwendet, die die Verknüpfung einer Gruppe haben muss. Die Konsequenz ist, dass die so bewiesenen Sätze für alle Gruppen gelten.

Satz 2.15 *Sei $(G, \circ)$ eine Gruppe. Dann gibt es genau ein Neutralelement.*

D.h. es gibt neben dem durch die Axiome geforderten Neutralelement kein weiteres Element, das auch die Eingeschaften eines Neutralelements hat. Um diese Aussage zu beweisen darf man jetzt nur die Verknüpfung und die drei Axiome, also Assoziativität, Existenz des Neutralelements und Existenz von Inversen verwenden. Wichtig für das Verständnis des folgenden Beweises ist noch, dass man in der Mathematik oft so tut, als wären Dinge verschieden, obwohl man das nicht weiß und es vielleicht auch überhaupt nicht so ist. Das ist etwas gewöhnungsbedürftig, funktioniert aber auch einfach richtig gut. Speziell für unseren Beweis werden wir gleich so tun, als hätten wir zwei Neutralelemente und anschließend zeigen, dass sie gleich sind. Das reicht, weil damit alle Neutralelemente in G gleich sind und es damit genau eins gibt.

Beweis: Es reicht zu zeigen, dass je zwei Neutralelemente in G gleich sind. Seien also e_1 und e_2 Neutralelemente in G (bzgl. $\circ$ natürlich). Dann gilt

$$e_1 = e_1 \circ e_2 = e_2$$

wobei die erste Gleichheit gilt, weil e_2 ein Neutralelement ist und die zweite, weil e_1 ein Neutralelement ist. □

Weil nun nur ein Element von G gemeint sein kann, wenn von dem Neutralelement in G die Rede ist, können wir eine Konvention machen: Im Folgenden soll stets e_G das Neutralelement einer Gruppe G bezeichnen.
Wir wollen gleich weitere Aussage beweisen:

Satz 2.16 *In einer Gruppe $(G, \circ)$ gibt es zu jedem Element genau ein inverses Element.*

Der grobe Rahmen ist der gleiche wie im ersten Beweis.

Beweis: Sei zunächst $a \in G$ und seien weiter $b_1, b_2 \in G$ beide invers zu a. Dann gilt:

$$b_1 \circ a = e_G = b_2 \circ a$$

Diese Gleichung kann man jetzt von rechts mit b_1 (oder einem beliebigen anderen Inversen von a) verknüpfen und man erhält:

$$b_1 \circ a \circ b_1 = b_2 \circ a \circ b_1$$

Und wenn man auf jeder Seite die Inversität des rechten b_1 benutzt wird das zu:

$$b_1 \circ e_G = b_2 \circ e_G$$

Also:

$$b_1 = b_2$$

Und genau das wollten wir haben! □

Genau wie wir vorher das Neutralelement e_G nennen konnten, können wir jetzt das Inverse von a mit a^{-1} bezeichnen.
Als nächstes Ziel wollen wir uns davon überzeugen, dass die Addition modulo n auf der Menge der natürlichen Zahlen, die kleiner als n sind, eine Gruppenverknüpfung ist. Dazu führen wir erstmal die folgende Bezeichnung für die Grundmenge ein:

$$\mathbb{Z}/n\mathbb{Z} = \{0, 1, 2, \ldots, n-2, n-1\}$$

Die Verknüpfung auf $\mathbb{Z}/n\mathbb{Z}$ nennen wir $\bar{+}$ und sie ist durch die normale Addition und anschließendes Restbilden (bei Division durch n) gegeben. Das versteht man vielleicht am Besten anhand eines Beispiels:

Beispiel 2.17 Sei $n = 4$, dann ist

$$\mathbb{Z}/4\mathbb{Z} = \{0, 1, 2, 3\}$$

Zwei Elemente, etwa 2 und 3, werden nun verknüpft, indem man erst addiert: $2 + 3 = 5$ und anschließend den Rest bildet: $5 \bmod 4 = 1$. In $\mathbb{Z}/4\mathbb{Z}$ gilt also:

$$2 \bar{+} 3 = 1$$

△

Satz 2.18 $(\mathbb{Z}/n\mathbb{Z}, \bar{+})$ *ist eine Gruppe.*

Beweis: Das Neutralelement ist 0, das Inverse zu $k \in \mathbb{Z}/n\mathbb{Z}$ ist $n - k$, denn: $(n - k + k) \bmod n = 0$. Schließlich ist $\bar{+}$ assoziativ, weil $+$ assoziativ ist: $(a + (b + c)) \bmod n = ((a + b) + c) \bmod n$. □

2.2.7 Die Gruppe $(\mathbb{Z}/n\mathbb{Z})^\times$

Wir haben bereits gesehen, dass die Multiplikation gelegentlich auch Gruppen liefert. Im Fall der Ganzen Zahlen $(\mathbb{Z}, +)$ ist diese jedoch etwas langweilig. Es gibt nämlich an multiplikativ invertierbaren Elementen nur 1 und -1. Eine vollkommen andere Situation ergibt sich für $\mathbb{Z}/n\mathbb{Z}$. Hier kann es nämlich durchaus sein, dass Zahlen beim Multiplizieren kleiner werden.
Zunächst liefert die bekannte Multiplikation „$\cdot$“ auf $\mathbb{Z}/n\mathbb{Z}$ eine Verknüpfung $\bar{\cdot}$ durch:

$$a \bar{\cdot} b = (a \cdot b) \bmod n$$

Hier helfen vielleicht wieder Beispiele:
In $\mathbb{Z}/5\mathbb{Z}$ gilt zum Beispiel: $3 \bar{\cdot} 4 = 12 \bmod 5 = 2$.
In $\mathbb{Z}/6\mathbb{Z}$ gilt: $2 \bar{\cdot} 3 = 0$.
Es stellt sich die Frage, auf welche Elemente von $\mathbb{Z}/n\mathbb{Z}$ man verzichten muss, um eine bzgl. $\bar{\cdot}$ Inverse zu erhalten. Für das Neutralelement kommt für $n > 1$ nur die 1 in Frage. Damit ist geklärt, was es für Element von $\mathbb{Z}/n\mathbb{Z}$ heißt, invertierbar bezüglich $\bar{\cdot}$ zu sein. Nämlich muss es für ein invertierbares $a \in \mathbb{Z}/n\mathbb{Z}$ ein $b \in \mathbb{Z}/n\mathbb{Z}$ geben, sodass $a \bar{\cdot} b = (a \cdot b) \bmod n = 1$ gilt. Das lässt sich umschreiben zu:

$$a \cdot b + k \cdot n = 1, k \in \mathbb{Z}$$

Hier muss man sich an die Darstellung des ggT aus Satz 2.7 erinnern – es gibt also eine Inverse b, wenn $\mathrm{ggT}(a, n) = 1$ gilt. Andererseits gilt $\mathrm{ggT}(a, n) = 1$, wenn man eine Linearkombination wie oben hat, weil gemeinsame Teiler von a und n die linke Seite teilen, und damit auch die rechte Seite, also 1. Letzteres bedeutet, dass alle invertierbaren Elemente auch teilerfremd zu n sind und führt dazu, dass die folgende Gruppe „größtmöglich“ gewählt ist.

Satz 2.19 *Sei* $(\mathbb{Z}/n\mathbb{Z})^\times = \{a \in \mathbb{Z}/n\mathbb{Z} | \, \mathrm{ggT}(a, n) = 1\}$*, dann ist* $((\mathbb{Z}/n\mathbb{Z})^\times, \bar{\cdot})$ *eine Gruppe.*

Beweis: Es bleibt nur noch die Assoziativität, welche sich auf die gleiche Art wie oben bei $\bar{+}$ von $(\mathbb{Z}, +)$ nun von $(\mathbb{Z}, \cdot)$ überträgt. □

Definition 2.20 Sei $(G, \circ)$ eine Gruppe. Für die k-fache Verknüpfung eines Elements g mit sich selbst, schreibt man auch g^k. In Formeln:

$$\underbrace{g \circ g \circ \cdots \circ g}_{k-\text{mal}} = g^k$$

Das Element g^k wird auch die k-te *Potenz* von g genannt.

In $(\mathbb{Z}/n\mathbb{Z}, \bar{+})$ lässt sich die eben definierte Operation leicht umkehren. Dies ist Gegenstand von Aufgabe 4 auf dem 4. Übungsblatt.
Sehr viel schwieriger ist es, diese Operation in $((\mathbb{Z}/n\mathbb{Z})^\times, \bar{\cdot})$ umzukehren – es ist sogar so schwierig, dass das Potenzieren für die Kryptographie taugt. Wie das genau funktioniert, ist unter anderem Thema des nächsten Abschnitts.

2.3 Einwegfunktionen und Schlüsselaustausch

Die Erkenntnisse des vorangegagenen Abschnitts wollen wir nun in der Kryptologie einsetzen. Zunächst werden wir uns damit beschäftigen, wie Passwörter möglichst sicher auf Computern gespeichert werden können und anschließend dem Hintergrund des Farbenbeispiels vom Anfang des Kapitels nachgehen und ein Verfahren untersuchen, das es zwei Parteien erlaubt, einen geheimen Schlüssel zu vereinbaren, den kein dritter kennt, ohne dass sie dabei Gelegenheit haben, einen solchen Schlüssel im Geheimen auszutauschen.

2.3.1 Passwortabfragen

Manch einer wird sicher schon beim Anlegen von Benutzerkonten für Internetdienste mehrfach das gleiche Passwort verwendet haben. Ein Problem dabei ist, dass ein Internetdienst – zumindest während der Passwortabfrage – irgendwo dieses Passwort speichert. Ähnliche Probleme enstehen etwa in Firmennetzwerken – wenn Passwörter zentral gespeichert werden, kann sie jeder, der Zugang zum Speicher hat, lesen und und diese Erkenntnis mißbrauchen. Tatsächlich gibt es einen interessanten Trick, der es erlaubt, die Korrektheit eines Passworts zu prüfen ohne es zu kennen. Dies geschiet mit den bereits in 2.1.2 erwähnten *Einwegfunktionen*. Was eine Einwegfunktion ist, soll hier nur recht vage definiert werden.

Definition 2.21 Eine *Einwegfunktion* ist eine Funktion f, sodass $f(x)$ sehr schnell ausgerechnet werden kann, es aber sehr lange dauert, zu gegebenem y ein x mit $f(x) = y$ auszurechnen.

Funktionen, auf die die obige Definition im weitesten Sinne zutrifft, sieht man sehr häufig. So ist zum Beispiel erfahrungsgemäß Dividieren anstrengender als Multiplizieren (jeweils mit einer gegebenen Zahl) und Logarithmen ziehen sehr viel aufwändiger als Potenzen bilden. Ein weiteres Beispiel aus dem Mathematikunterricht ist wohl noch Differenzieren und bilden von Stammfunktionen. Echte Einwegfunktionen sind diesen Beispielen ähnlich, ein sehr wichtiges Beispiel haben wir im letzten Abschnitt kennen gelernt.

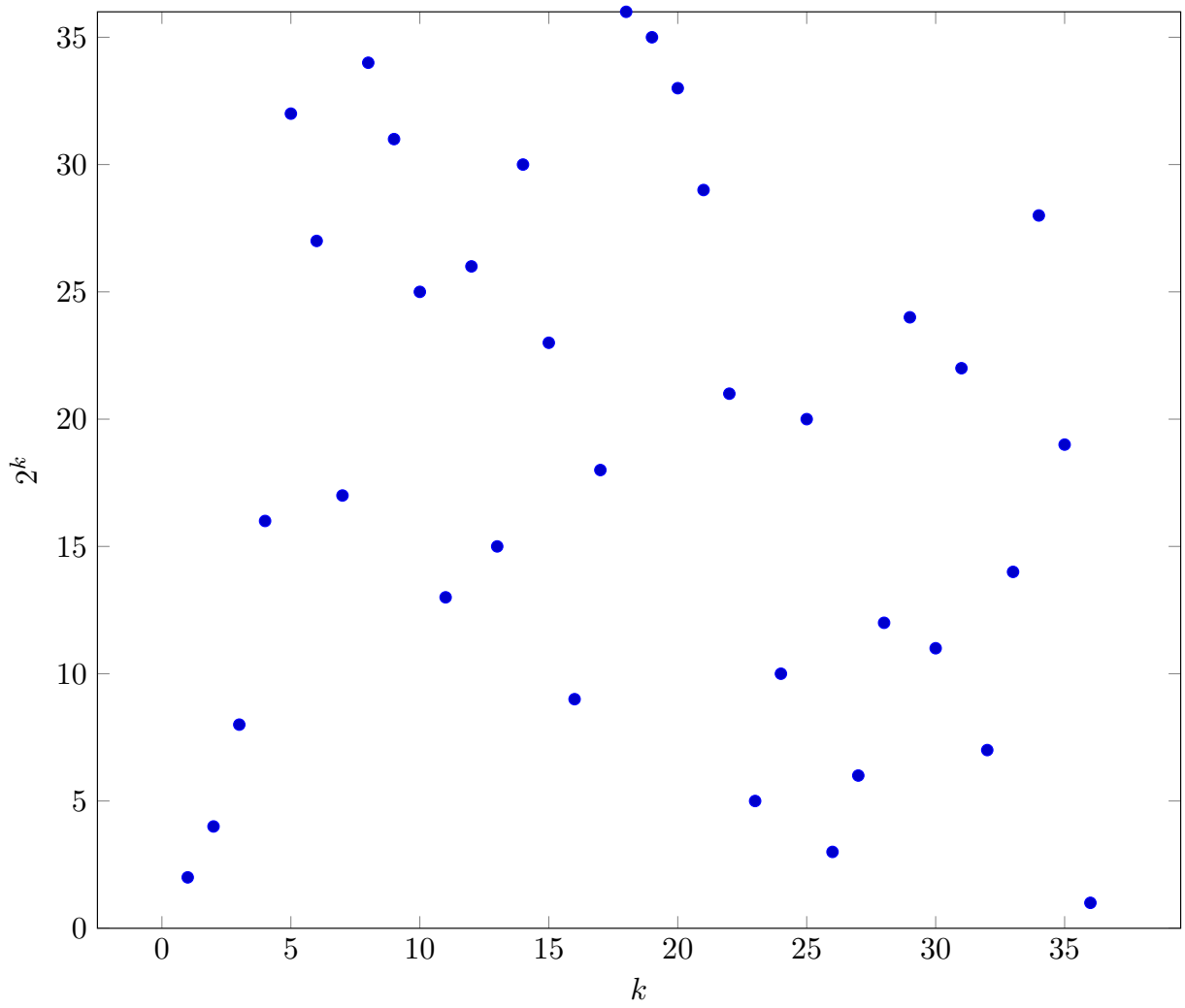


Abbildung 2.1: Diskrete Potenzen von 2 in $\mathbb{Z}/37\mathbb{Z}$

Definition 2.22

1. Sei p eine Primzahl und $g \in (\mathbb{Z}/p\mathbb{Z})^\times$. Die *diskrete Potenzfunktion* (zur Basis g) ordnet einer Zahl k mit $0 < k < p$ das Gruppenelement

$$g^k = \underbrace{g \circ g \circ \cdots \circ g}_{k\text{-mal}}$$

   zu.

2. Falls jedes Element von $(\mathbb{Z}/p\mathbb{Z})^\times$ eine Potenz von g ist, gibt es den *diskreten Logarithmus* $\log_g$, der jedem $h \in (\mathbb{Z}/p\mathbb{Z})^\times$ ein $l = \log_g(h)$ mit $0 < l < p$ zuordnet, sodass $g^l = h$ gilt.

Tatsächlich gibt es in $(\mathbb{Z}/p\mathbb{Z})^\times$ immer ein Element g, sodass jedes Element eine Potenz von g ist. Der Beweis sprengt jedoch leider den Rahmen unserer Vorlesung. Andererseits gibt es auch Elemente auf die das nicht zutrifft, wie zum Beispiel 1 im Fall $p > 2$ oder etwa 5 im Fall $p = 19$. Abbildung 2.1 überzeugt vielleicht, dass es sehr schwierig ist, den diskreten Logarithmus zu berechnen. Wenn Informatiker untersuchen, wie schnell etwas ausgerechnet werden kann, wird die benötigte Rechenzeit oft in Abhängigkeit der Größe der Eingangsdaten – zum Beispiel der Größenordnung von Zahlen, die verrechnet werden sollen – angegeben. Anschaulich weiß man etwa, dass es mehr Aufwand ist

zwei vierstellige Zahlen schriftlich zu multiplizieren als das gleiche mit zwei zweistelligen Zahlen zu tun (günstige Spezialfälle ausgenommen). Hier wird das Wachstum sehr grob durch die Funktion $n \cdot m$ beschrieben, wenn n die Anzahl der Ziffern der ersten Zahl und m die der zweiten Zahl ist.
Es stellt sich heraus, dass die Zeit, die gebraucht wird um einen diskreten Logarithmus in $(\mathbb{Z}/p\mathbb{Z})^\times$ zu berechnen, grob gemäß $\sqrt{p}$ wächst. D.h., wenn p ungefähr viermal so groß gewählt wird, brauchen gute Verfahren etwa doppelt so lange um diskrete Logarithmen zu berechnen. Ganz anders verhält es sich mit Potenzen, hier wächst die benötigte Zahl etwa gemäß $\log(p)$ – und damit sehr viel langsamer, als $\sqrt{p}$: Hier verdoppelt sich die Rechenzeit nur, wenn man p quadriert!
Dieser dramatische Unterschied wird vielleicht durch folgendes Zahlenbeispiel am besten verdeutlicht. Wir wollen für die Rechnung einen Moment ignorieren, dass p eigentlich eine Primzahl sein muss. Sei also $p = 1000000 = 10^6$. Wir wollen nun in beiden Fällen sehen, wie die Rechenzeit wächst, wenn wir zu $q = p^2 = 10^{12}$ übergehen. Im Fall des Potenzierens verdoppelt sich die Rechenzeit, im Fall der Wurzel wird sie dagegen vertausendfacht! Für große Primzahlen p ist es also praktisch unmöglich den diskreten Logarithmus zu berechnen. Doch was hilft uns das nun bei dem Problem der Passwotrkontrolle?
Wenn man eine Einwegfunktion zur Verfügung hat, kann man statt eines Passworts das Bild des Passworts[‡] unter der Einwegfunktion speichern. Da es praktisch unmöglich[§] ist, aus dem gespeicherten Wert das Passwort zu berechnen, kann man es so sehr viel sicherer speichern und es ist trotzdem noch möglich rauszufinden, ob jemand das Passwort kennt – dazu muss man einfach nur das Passwort abfragen und wieder die Einwegfunktion darauf anwenden, um schließlich das Ergebnis mit dem gespeicherten Wert zu vergleichen. Neben dieser Anwendung gibt es noch eine weitere äußerst verblüffende, die Gegenstand des nächsten Abschnitts ist.

2.3.2 Schlüsselaustausch

Wir untersuchen nun ein weiteres Problem, das zu Beginn dieses Kapitels bereits erwähnt wurde. Im Fall der Enigma verteilte die Wehrmacht für jeden Monat jeweils neue Schlüsselbücher, die neben anderen Informationen für jeden Tag einen Schlüssel enthielten. Neben dem logistischen Aufwand bei der Schlüsselverteilung, war dies natürlich eine empfindliche Schwachstelle dieser Verschlüsselung, die auch mehrfach erfolgreich angegriffen wurde. Eine moderne Anwendung, der Handel via Internet, wäre in der heutigen Form kaum vorzustellen, wenn jeder, der etwas kauft, sich etwa vor dem Kauf per Post einen geheimen Schlüssel vom entsprechenden Händler zuschicken lassen müsste. Alle evtl. benötigten Schlüssel vorab zu verteilen, ist auch undenkbar, wenn man bedenkt, dass – wie zu Beginn des Kapitels diskutiert wurde – für N Teilnehmer schon $N(N-1)/2$ Schlüssel notwendig wären, wenn jeder mit jedem sicher kommunizieren können möchte.
Tatsächlich ist es möglich, die nötigen Schlüssel „aus dem Nichts“ entstehen zu lassen. Der Diffie-Hellman-Schlüsselaustausch ist ein Verfahren, das dies ermöglicht und dessen Sicherheit durch die Komplexität des diskreten Logarithmus gestützt wird. Man benötigt hierzu wieder eine Primzahl p und ein Element $g \in (\mathbb{Z}/p\mathbb{Z})^\times$. Dass alle Elemente von $(\mathbb{Z}/p\mathbb{Z})^\times$ Potenzen von g sind, wird hier nicht gefordert – es wäre allerdings schlecht hier ein Element zu wählen, das nur wenige verschiedene Potenzen hat. Wenn das Verfahren ausgeführt wird, werden p und g nicht geheim gehalten. Nun wollen zwei Parteien, nennen wir sie Alice und Bob verschlüsselt kommunizieren und benötigen einen Schlüssel. Zunächst wählt Alice eine zufällige Zahl a mit $0 < a < p$ und behält diese für sich und Bob verfährt genauso mit einer Zahl b. Nun teilen Alice und Bob dem jeweils anderen die Zahlen g^a und g^b mit. Ein Zuhörer müsste nun einen diskreten Logarithmus berechnen um a oder b herauszufinden. Alice kennt nun insgesamt a und g^b, kann also g^{ab} berechnen. Bob kennt b und g^a und kann damit g^{ba}. Es gilt $g^{ab} = g^{ba}$, womit Alice und Bob nun eine Zahl kennen, die niemand sonst so leicht berechnen kann, also im Idealfall niemandem bekannt ist.

[‡] Dazu muss man aus dem Passwort noch eine natürliche Zahl machen.
[§] Leider ist das in der Realität oft doch möglich. Näheres gibt es zum Beispiel hier: [**hash˙hack**]

Diese Zahl kann nun als Schlüssel für ein symmetrisches Verschlüsselungsverfahren dienen. Man könnte etwa die Vigenère-Verschlüsselung verwenden, wenn man die Schlüsselzahl in eine Zeichenfolge umcodiert – es gibt jedoch sehr viel bessere moderne Verfahren wie zum Beispiel AES, die wir hier aber leider nicht vorstellen können.

Beispiel 2.23 Wir wollen das Diffie-Hellman-Verfahren nun mit kleinen[¶] Zahlen einmal komplett durchführen.
Seien dazu $p = 29$ und $g = 11$. Die beiden Parteien heißen wie üblich Alice und Bob. Als erstes wählt Alice im Geheimen die Zahl $a = 5$ und Bob entsprechend $b = 12$. Nun berechnet Alice $g^a = 11^5 = (11^2)^2 \cdot 11 = 5^2 \cdot 11 = 25 \cdot 11 = 14$ und teilt das Ergebnis Bob – und eventuell auch dem Rest der Welt – mit. Genauso berechnet Bob $g^b = 11^{12} = (11^2)^6 = (5^2)^3 = 25 \cdot 25^2 = 25 \cdot 16 = 23$ und teilt dieses Ergebnis Alice mit.
Nun kennt Alice sowohl g^b als auch a und kann $(g^b)^a = 23^5 = \cdots = 25$ berechnen. Und Bob berechnet $(g^a)^b = 14^{12} = 25$. Alice und Bob haben nun also den gemeinsamen, geheimen Schlüssel 25 und können damit ihre Kommunikation verschlüsseln. △

Anstelle von $((\mathbb{Z}/p\mathbb{Z})^\times, \cdot)$ können auch andere endliche Gruppen für das Diffie-Hellman-Verfahren verwendet werden. In der Praxis werden für p und g sehr spezielle Werte verwendet und natürlich gibt es noch viele technische Details drumherum, aber trotzdem wird im Kern tatsächlich genau das oben beschriebene Verfahren in Alltagssituationen von Computern durchgeführt.

2.3.3 Elliptische Kurven

Auch wenn die Gruppe $((\mathbb{Z}/p\mathbb{Z})^\times, \cdot)$ in der Praxis verwendet wird, ist sie, wie von vielen Experten vermutet wird, nicht die beste Wahl. Zum Beispiel hält die National Security Agency[‖] in [**nsa˙ecm**] ein Plädoyer dafür, die sogenannte Punktegruppe einer elliptischen Kurve für das Diffie-Hellman-Verfahren zu verwenden. Zusammengefasst ist die behauptete Verbesserung, dass mit sehr viel kleineren Schlüsseln das gleiche Maß an Sicherheit erreicht wird – ohne irgendwelche Einbußen.
Hier soll nur exemplarisch die folgende Elliptische Kurve (schwarz in Abbildung 2.2) gezeigt werden, ohne im Allgemeinen zu klären, was eine Elliptische Kurve überhaupt ist.

Auf der Menge der Punkte einer elliptischen Kurve lässt sich eine Verknüpfung $+$ definieren. Um zwei Punkte P, Q zu addieren, zieht man – wie in Abbildung 2.2 dargestellt – eine Gerade PQ durch P und Q. Wir wollen zunächst davon ausgehen, dass $P \neq Q$ gilt und es genau einen Schnittpunkt S mit der Elliptischen Kurve gibt. Wird der Punkt S an der x-Achse gespiegelt, dann ist das Spiegelbild S' wieder ein Punkt auf der Elliptischen Kurve, weil die x-Achse eine Symmetrieachse der Kurve ist. Der so erhaltene Punkt S' ist das Ergebnis der Verknüpfung von P und Q, in Formeln: $S' = P + Q$.
Falls $P = Q$ gilt, soll „die Gerade durch P und Q“ eine Tangente sein – auch dieser Fall geht durch die speziellen Eigentschaften Elliptischer Kurven gut: Eine Tangente (die nicht senkrecht ist) hat immer genau einen weiteren Schnittpunkt mit der Kurve. Nun bleibt der Fall zu klären, dass der Schnittpunkt S nicht existiert – interessanterweise hat nämlich keine Gerade mehr als drei Schnittpunkte mit der Kurve, daher ist immer klar, was S sein soll. Dieser Fall tritt nur ein, wenn PQ senkrecht, also parallel zur y-Achse ist. Um dieses Problem zu lösen, erfindet man einen Punkt ∞, der einfach per Definition zur Kurve gehört und der nötige Schnittpunkt mit allen senkrechten Geraden ist. Damit alles gut geht, muss man noch per Definition festlegen, dass ∞ gespiegelt an der x-Achse wieder ∞ ist.
Man kann Beweisen, dass nun die Punkte der Elliptischen Kurve mit nun definierten Verknüpfung $+$ eine Gruppe sind. Allerdings ist diese Gruppe alles anderen als endlich – es liegen mit Sicherheit

[¶] Welche Zahlen in der Praxis gewählt werden, kann man in [**rfc˙2412**] nachlesen. p wird unter anderem als 1024 Bit Zahl gewählt – also eine etwa 300-stellige Dezimalzahl.

[‖] Ein Geheimdienst der USA, der eine ähnliche Funktion wie der Bundesnachrichtendienst in der Bundesrepublik hat und unter anderem auf Verschlüsselung spezialisiert ist.

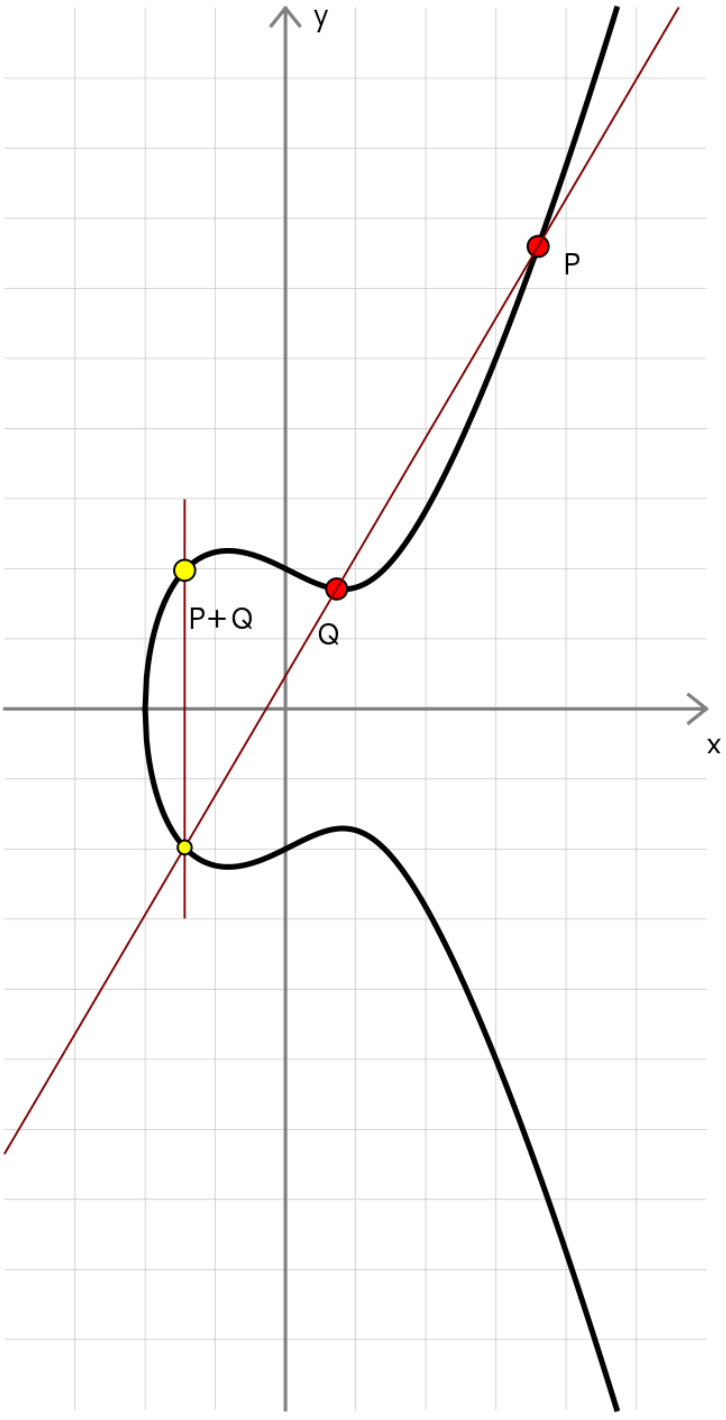


Abbildung 2.2: Verknüpfung der Punkte P und Q auf einer Elliptischen Kurve

unendlich viele Punkte auf der abgebildeten Kurve! Um dieses „Problem“ zu beheben muss man noch zu Koordinaten übergehen, die Elemente von $\mathbb{Z}/n\mathbb{Z}$ sind. Ganz richtig ist das so leider nicht – aber um zu sehen, wie das genau funktioniert, müsste man nun sehr weit ausholen, was hier nicht möglich ist**. Für den Einsatz im Diffie-Hellman-Verfahren schaut man sich also Potenzen eines Punktes auf dieser Kurve bezüglich dieser Verknüpfung an.
Wer noch mit der Verknüpfung experimentieren möchte, kann das mit dem Programm [**ec˙plotter**] tun – wir wollen nicht weiter in das umfangreiche Gebiet der Elliptische-Kurven-Kryptologie eintauchen.

2.4 Asymmetrische Verschlüsselung

Das im vorangegangenen Abschnitt diskutierte Diffie-Hellman-Verfahren bietet leider keine Möglichkeit zu wissen, ob am anderen Ende einer Kommunikationsleitung auch wirklich die Person ist, mit der man Vertrauliches austauschen möchte. Selbst wenn es wirklich nur eine Leitung gäbe, die an einem bekannten Ort endet, gibt es den sogenannten Man-in-the-middle-Angriff, bei dem ein Angreifer die gesamte Kommunikation belauschen kann.

2.4.1 Man-in-the-middle

Der „Man-in-the-middle“ soll im Folgenden Mallory heißen und in der Mitte einer Leitung zwischen Alice und Bob die Möglichkeit haben, gesendete Zahlen abzufangen und durch eigene zu ersetzen.
Wollen Alice und Bob nun einen Schlüssel mittels des Diffie-Hellman-Verfahrens erzeugen, so geht Mallory wie folgt vor:
Wenn Alice sich ihre geheime Zahl a ausdenkt und g^a an Bob senden will, so fängt Mallory g^a ab, denkt sich eine eigene geheime Zahl m aus und schickt anstelle von g^a einfach g^m an Bob weiter. Letzterer denkt sich nichtsahnend seinen Exponenten b aus und sendet g^b. Diese Zahl wird wieder von Mallory abgefangen und durch g^m ersetzt. Nun berechnet Alice den Schlüssel g^{am} und Bob den Schlüssel g^{bm}. Mallory kennt beide Schlüssel und kann alles Verschlüsselte, was Alice sendet mit g^{am} entschlüsseln und mit g^{bm} verschlüsselt an Bob weitersenden. So bemerken Alice und Bob nichts und Mallory kann alles mitlesen.

2.4.2 Öffentliche und private Schlüssel

Das ist natürlich ein ernsthaftes Problem – mit heute verwendeter Technik kann man nicht verhindern, dass Leitungen abgehört und Gesendetes ersetzt wird. Allerdings wäre dieses Problem gelöst, sobald etwa Alice weiß, dass sie direkt mit Bob redet. Eine Möglichkeit dies zu gewährleisten ist die sogenannte *asymmetrische Verschlüsselung*.
Dabei hat jeder Kommunikationsteilnehmer zwei Schlüssel. Einer der Schlüssel heißt *öffentlicher Schlüssel* und ist allen bekannt, der zweite heißt *privater Schlüssel* und darf zu keinem Zeitpunkt einem anderen als dem Besitzer bekannt sein. Mit dem öffentlichen Schlüssel kann nun jeder Zahlen verschlüsseln, die nur mit dem zugehörigen privaten Schlüssel wieder entschlüsselt werden können. Realisiert wird dies über eine Einwegfunktion, die vom öffentlichen Schlüssel abhängt und mit dem privaten Schlüssel ohne lange Rechenzeit umgekehrt werden kann. Man nennt eine solche Einwegfunktion, die mit einer Zusatzinformation (hier: dem privaten Schlüssel) umgekehrt werden können, *Falltürfunktion* bzw. *Trapdoor-function*.

** Es gibt meines Wissens zu diesem Thema keine Quelle die man guten Gewissens Interessierten ohne Grundstudium in Mathematik empfehlen kann.

2.4.3 Das RSA-Kryptosystem

RSA ist ein asymmetrisches Verfahren, das die oben diskutierten Probleme löst. Um das Verfahren verstehen zu können, müssen wir uns vorab noch einwenig mit der Gruppe $(\mathbb{Z}/n\mathbb{Z})^\times$ beschäftigen. Der folgende Satz ist vielleicht mit den abstrakten Gruppeneigenschaften leichter nachzuvollziehen und wird daher für beliebige Gruppen gezeigt.

Satz 2.24 *Sei $(G, \circ)$ eine Gruppe mit $n \in \mathbb{N}$ Elementen. Sei weiter e_G das Neutralelement von $(G, \circ)$ und $g \in G$ ein beliebiges Element, dann gilt: $g^n = e_G$.*

Beispiel 2.25 Für eine Primzahl p und $k \in (\mathbb{Z}/p\mathbb{Z})^\times$ gilt stets: $k^{p-1} = 1$. Multipliziert man noch beide Seiten mit k, so erhält man die seltsame, aber hier tatsächlich korrekte, Gleichung $k^p = k$. △

Beweis: Wir wollen uns zuerst einen Spezialfall anschauen, der sich leichter beweisen lässt: Sei $g \in G$ so, dass jedes Element von G eine Potenz von g ist. Das gilt zum Beispiel für $1 \in \mathbb{Z}/n\mathbb{Z}$ mit $G = \mathbb{Z}/n\mathbb{Z}$ und $\circ = \bar{+}$.
Nun muss es $i, k \in \mathbb{N}$ mit $g^i = g^{i+k}$ geben, sonst wären alle Potenzen von g verschieden, was im Widerspruch zur Endlichkeit von G stünde. Damit gilt auch $g^k = e_G$, $g^{k+1} = g$, $g^{k+2} = g^2$, …, $g^{k+k} = e_G$. Die ersten k Potenzen wiederholen sich also immer weiter und es gibt insgesamt höchstens k verschiedene Potenzen, was nach Voraussetzung nur für $k = n$ der Fall sein kann. Also gilt $g^n = e_G$. Damit ist der Spezialfall bewiesen.
Im allgemeinen Fall muss zwar nicht jedes Element von G eine Potenz von g sein, dennoch gibt es immer ein kleinstes k, sodass g^k das Neutralelement ist – das ist genau die Aussage der dritten Aufgabe des vierten Übungsblatts. Nun stellt sich heraus, dass k ein Teiler von n ist. Um das einzusehen, schöpft man G sukzessive mit Teilmengen der Form

$$\{b \circ g^1, b \circ g^2, \ldots, b \circ g^{k-1}, b \circ g^k\}$$

aus. Das heißt, man sucht sich zunächst unter den Elementen von G, die keine Potenz von g sind eins aus – sagen wir b – und multipliziert alle Potenzen von g damit. Unter den so erhaltenen Elementen ist kein einziges, das eine Potenz von g ist, weil wenn dem so wäre, gälte $g^i = b \circ g^j$, womit entweder $b = g^{i-j}$ gilt oder b invers zu g^{j-i} (und damit auch selbst eine Potenz von g) wäre. Dieses Verfahren lässt sich fortführen, indem man immer ein noch nicht erwischtes Element von links anmultipliziert. Schließlich gibt es aufgrund der Endlichkeit von G keine solchen Elemente mehr. Wenn l die Anzahl der Schritte ist, bis letzteres eintritt, dann gilt $k \cdot l = n$, womit gezeigt ist, dass k ein Teiler von n ist. Dami gilt:

$$g^n = g^{k \cdot l} = \left(g^k\right)^l = e_G^l = e_G$$

□

Das für unser Ziel interessante an dieser Aussage ist, dass sich Vielfache der Gruppengröße im Exponenten wegkürzen. Das wird es uns später erlauben, geeignete Falltürfunktionen zu konstruieren – findet man nämlich zu einem Exponenten e eine Zahl a, sodass $e \cdot a$ modulo der Gruppengröße 1 ist, dann kann man Potenzieren mit e durch Potenzieren mit a rückgängig machen: Wenn n die Gruppengröße und $k \in \mathbb{Z}$ so ist, dass $k \cdot n + 1 = e \cdot a$ ist, dann gilt

$$(x^e)^a = x^{e \cdot a} = x^{k \cdot n + 1} = x \quad \text{für alle } x$$

Das heißt eine Nachricht kann erst durch Potenzieren mit e verschlüsselt und anschließend durch Potenzieren mit a wieder entschlüsselt werden. Das ist bereits das Grundprinzip des RSA-Verfahrens. Bevor wir dieses näher betrachten, müssen wir uns allerdings noch mit der Größe der Gruppe $(\mathbb{Z}/N\mathbb{Z})^\times$

beschäftigen – allerdings genügt uns der Spezialfall, dass N ein Produkt von Primzahlen p, q ist.
Zur Erinnerung:

$$(\mathbb{Z}/N\mathbb{Z})^\times = \{k \in \mathbb{N} | \quad k < N \text{ und } \mathrm{ggT}(k, N) = 1\}$$

Das heißt die Anzahl der Elemente von $(\mathbb{Z}/N\mathbb{Z})^\times$ ist die Anzahl der zu $N = p \cdot q$ teilerfremden natürlichen Zahlen, die kleiner als N sind. Diese Anzahl wollen wir von nun an mit $\varphi(N)$ bezeichnen und im folgenden Satz für unseren Spezialfall bestimmen.

Satz 2.26 *Für zwei Primzahlen p, q gilt:*

$$\varphi(p \cdot q) = (p-1) \cdot (q-1)$$

Beweis: Eine natürliche Zahl $k \in \mathbb{N}$ mit $0 < k < pq$ ist genau dann teilerfremd zu pq, wenn ihre Primfaktorzerlegung weder p noch q enthält. *Nicht* teilerfremd zu pq sind die folgenden Zahlen

$$p \cdot 1, p \cdot 2, \ldots, p \cdot (q-1), p \cdot q$$
$$q \cdot 1, q \cdot 2, \ldots, q \cdot (p-1), q \cdot p$$

wobei hier pq als einzige Zahl doppelt aufgeführt wurde. In der oberen Reihe stehen q Zahlen und in der unteren p – insgesamt gibt es also $p + q + 1$. Das heißt, dass es

$$pq - (p + q - 1) = pq - p - q + 1 = (p-1) \cdot (q-1)$$

zu pq teilerfremde Zahlen gibt. □

Interessant ist hier für das RSA-Verfahren, dass man durch Faktorisieren von N die Anzahl der Elemente von $(\mathbb{Z}/N\mathbb{Z})^\times$ bestimmen kann und man geht, ähnlich wie beim Diffie-Hellman-Verfahren und dem diskreten Logarithmus, davon aus, dass es nicht wesentlich leichter ist, diese Anzahl, also $\varphi(N)$ zu bestimmen, als N zu faktorisieren. Man benutzt im RSA-Verfahren also neben diskreten Logarithmen indirekt die bereits vorgestellte Einwegfunktion, die zwei Primzahlen auf ihr Produkt abbildet.
Wir haben nun alles beisammen, um das Verfahren vorstellen zu können. Unsere Kommunikationparteien heißen wieder Alice und Bob. Als erstes bestimmt Alice zwei große Primzahlen p, q und berechnet $N = pq$ und $\varphi(N) = (p-1)(q-1)$. Nun wählt sie einen Exponenten e mit $\mathrm{ggT}(e, \varphi(N)) = 1$, sodass also Potenzieren von Elementen von $(\mathbb{Z}/N\mathbb{Z})^\times$ rückgängig gemacht werden kann. Um zu letzterem in der Lage zu sein, berechnet Alice noch mit dem erweiterten euklidischen Algorithmus ein a mit $e \cdot a = k \cdot \varphi(N) + 1$. Die Zahlen a und $\varphi(N)$ behält Alice für sich – sie bilden ihren privaten Schlüssel. Die Zahlen N und e sind Alice öffentlicher Schlüssel, den sie nun allen die es interessiert mitteilt.
Nun kann Bob eine Nachricht an Alice in Form eines Elementes $m \in (\mathbb{Z}/N\mathbb{Z})^\times$ derart verschlüsseln, dass es nur Alice lesen kann. Dazu berechnet er einfach m^e. Wenn Alice diese Zahl empfängt, kann sie $(m^e)^a = m^{e \cdot a} = m^{k \cdot \varphi(N)+1} = m$ berechnen und die Nachricht lesen. Andererseits könnte Alice für eine Nachricht m die Zahl m^a berechnen und veröffentlichen. Aus dieser kann nun jeder $(m^a)^e = m$ berechnen und wissen, dass nur Alice die Nachricht m verschickt haben kann. Wenn man beides kombiniert und Bob auch einen öffentlichen Schlüssel (N_B, e_B) und privaten Schlüssel $(b, \varphi(N_B))$ hat, kann er $\left(m^b\right)^e$ berechnen, womit nur Alice die Nachricht lesen könnte und zusätzlich noch wüsste, dass sie von Bob verschickt wurde.
Wem das alles zu abstrakt war, hilft vielleicht das folgende Zahlenbeispiel.

Beispiel 2.27 Seien $p = 17$ und $q = 11$. Dann ist $N = 187$ und $\varphi(N) = 16 \cdot 10 = 160$. Als Exponent e wählt Alice 3 (es gilt $\mathrm{ggT}(3, 160) = 1$) und veröffentlicht $(187, 3)$. Mit dem erweiterten euklidischen Algorithmus (der für diese Zahlen lediglich einer Division mit Rest entspricht) erhält Alice folgende Gleichung:

$$160 - 3 \cdot 53 = 1$$

Ihr geheimes a ist also[††] $-53 \bmod 160 = 107$. Nehmen wir nun an, Bob möchte Alice die Nachricht $m = 43$ schicken. Dazu berechnet er

$$m^e = 43^3 \bmod 187 = 32$$

und schickt das Ergebnis an Alice. Diese berechnet nun

$$32^a = 32^{107} \bmod 187 = 43$$

△

2.4.4 RSA in der Praxis

Wie schon beim Diffie-Hellman-Verfahren gäbe es auch im Fall von RSA sehr viel zu beachteten, wenn man mittels dieses Verfahrens wirklich sicher kommunizieren will. Die folgende Erkärung ist daher eine sehr grobe Vereinfachung.
Wir wollen noch verstehen, wie uns öffentliche und private Schlüssel helfen, Man-in-the-middle Angriffe zu verhindern. Eine schlechte Lösung wäre es, wenn die öffentlichen Schlüssel aller n Teilnehmer vorab verteilt würden – auch wenn das hier schon deutlich besser ginge als mit den im symmetrischen Fall nötigen $n(n-1)/2$ Schlüsseln. Man geht anders vor: Wenn Bob und Alice kommunizieren möchten, lässt Alice sich Bobs öffentlichen Schlüssel von einem Dritten, dem beide vertrauen, bestätigen und dessen öffentlicher Schlüssel verbreitet ist. Wenn Alice den öffentlichen Schlüssel (N, e) hat, dann lässt sie sich das sozusagen von dem vetrauenswürdigen Dritten bestätigen. Dieses bestätigen ist eine Nachricht der Form „Alice hat den öffentlichen Schlüssel (N, e)“, die vom vertraunswürdigen Dritten mit seinem privaten Schlüssel verschlüsselt wird. Diesen Vorgang nennt man *signieren*. Wenn nun Bob dem Dritten vertraut und die derart verschlüsselte Nachricht erhält, weiß er, dass Alice den öffentlichen Schlüssel (N, e) hat und kann Nachrichten verschicken, die nur Alice lesen kann. Genauso kann sich Bob seinen öffentlichen Schlüssel (N_B, e_B) signieren lassen. An diesem Punkt könnten Bob und Alice natürlich auch auf das Diffie-Hellman-Verfahren verzichten und sich direkt einen Schlüssel zuschicken. Allerdings gibt es trotzdem noch einen guten Grund, innerhalb von RSA einen Schlüssel mit dem Diffie-Hellman-Verfahren zu erzeugen: *forward security*. Dabei geht es darum, was passieren kann, wenn irgendwann jemand Alice oder Bobs Geheimnisse entdeckt. Öffentliche und private Schlüssel werden typischerweise sehr lange aufgehoben. Ein mit dem Diffie-Hellman-Verfahren erzeugter Schlüssel kann dagegen nach der Kommunikation gelöscht werden (ebenso die geheimen Exponenten a und b). Das bedeutet, dass niemand, der die verschlüsselte Kommunikation zwischen Alice und Bob mitgeschnitten hat, nachträglich den Mitschnitt entschlüsseln kann, wenn er Zugang zu Alice oder Bobs Geheimnissen erlangt. Ein großer Vorteil des Diffie-Hellman-Verfahrens gegenüber dem RSA-Verfahren ist außerdem, dass der gemeinsame Schlüssel mit sehr viel weniger Rechenleistung erzeugt werden kann, als für das erstellen von zwei Schlüsselpaaren nötig ist.

[††] Die Definition des Rests kann wörtlich für negative Zahlen übernommen werden. In diesem Beispiel: $-53 = (-1) \cdot 160 + 107$.

Bücher

[Beu10a] Albrecht Beutelspacher. *Diskrete Mathematik für Einsteiger: Mit Anwendungen in Technik und Informatik*. Ed. by Marc-Alexander Zschiegner. Wiesbaden: Vieweg+Teubner—Springer, 2010. ISBN: 978-3-8348-9941-5.

[Beu10b] Albrecht Beutelspacher. *Kryptografie in Theorie und Praxis: Mathematische Grundlagen für Internetsicherheit, Mobilfunk und elektronisches Geld*. Ed. by Heike B. Neumann and Thomas Schwarzpaul. Wiesbaden: Vieweg+Teubner—Springer, 2010. ISBN: 978-3-8348-9631-5.

[Bur05] Davin M. Burton und Heinz Dalkowski. *Handbuch der elementaren Zahlentheorie*. Lemgo: heldermann Verlag, 2005. ISBN: 3-88538-112-5.

[Haf11] Dörte Haftendorn. *Mathematik sehen und verstehen: Schlüssel zur Welt*. Heidelberg: Spektrum Akademischer Verl., 2011. ISBN: 978-3-8274-2044-2.

[Hen11] Norbert Henze. *Stochastik für Einsteiger: Eine Einführung in die faszinierende Welt des Zufalls*. 9., erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag—Springer, 2011. ISBN: 978-3-8348-8649-1.

[Sin00] Simon Singh. *The code book : the secret history of codes and codebreaking*. London [u.a.]: Fourth Estate, 2000. ISBN: 1-85702-889-9.

Online

[al02] Katrin Schäfer et al. *Enigma - jojhp tvnnu qmmtf ppo (I IV III B G B)*. http://www.matheprisma.uni-wuppertal.de/Module/Enigma/index.htm. 2002.

[Com] Wikipedia Community. *Wikipedia-Eintrag zur Buchstabenhäufigkeit*. http://de.wikipedia.org/wiki/Buchstabenh\protect{\grmnU@D1ex{\setbox\z@\hbox{\char127}\dimen@-.45ex\advance\dimen@\ht\z@}\accent127\fontdimen5\font\grmnU@Da}\penalty\@M\hskip\z@skipufigkeit.

[Con] Contributers. *CrypTool-Online*. http://www.cryptool-online.org/.

[Cry07] Cryptomuseum. *Enigma Cipher Machine*. http://cryptomuseum.com/crypto/enigma/. 2007.

[Ell07] Graham Ellsbury. *The Enigma and the Bombe*. http://www.ellsbury.com/enigmabombe.htm. 2007.

[eta07] Bernhard Esslinger et.al. *Kryptologie für Jedermann. Einführung in sichere Ver-und Entschlüsselungsverfahren*. https://www.sicher-im-netz.de/files/images/fibel_kryptologie.pdf. 2007.

[Kne07] Karsten Knetsch. *Die Enigma - Funktionsweise und Implementierung innerhalb einer webbasierten Lernumgebung*. http://www.ostfalia.de/cms/de/pws/seutter/kryptologie/enigma/. 2007.

[Küh11] Stefan Kühnlein. *Einfuehrung in die Algebra und Zahlentheorie, Skript*. http://www.math.kit.edu/iag3/lehre/einfalgzahl2011s/media/einfalgzahl.pdf. 2011.

[Pet] Hartmut Petzold. *Auszug aus: Meisterwerke aus dem Deutschen Museum Band II*. http://www.deutsches-museum.de/sammlungen/ausgewaehlte-objekte/meisterwerke-ii/enigma/.

[Pom07] Klaus Pommerening. *Kryptologie - Zeichenhäufigkeiten in Deutsch*. http://www.staff.uni-mainz.de/pommeren/Kryptologie/Klassisch/1_Monoalph/deutsch.html. 2007.