Vorwort

Kryptographie war über zwei Tausend Jahre lang fast ausschließlich ein Thema für Militärs und Diplomaten. Als dann die Computer in die Banken einzogen und auch der Zahlungsverkehr zwischen verschiedenen Filialen und später Instituten elektronisch abgewickelt werden sollte, zog damit auch die Kryptographie dort ein, denn natürlich mußte die Kommunikation zwischen verschiedenen Standorten gewöhnliche Leitungen oder gar Richtfunkstrecken benutzen, die sowohl abgehört als auch (schlimmer noch bei Geldtransfers) manipuliert werden können.

Wirklich universell wurde die Nachfrage nach Kryptographie schließlich durch das *world wide web,* wo beispielsweise elektronischer Handel ohne die Übermittlung sensitiver Information nicht möglich ist, wo aber gleichzeitig die Daten über mehrere Zwischenrechner geschickt werden müssen, von denen sie jeder zumindest im Prinzip kopieren und auswerten kann.

Die Rolle der Kryptographie beschränkt sich aber bei weitem nicht mehr auf die bloße Geheimhaltung von Information: Sie soll auch dafür sorgen, daß beispielsweise ein Geldausgabeautomat nur legitimen Kunden Geld gibt oder daß sich auch mit gestohlener Identitätskarte kein Unbefugter Zugang zu sensitiven Bereichen eines Unternehmens verschaffen kann. Außerdem gestattet sie den Abschluß elektronischer Verträge, deren „elektronische Unterschrift“ in Deutschland seit einigen Jahren rechtsgültig ist.

Angesichts dieser großen Bedeutung der Kryptographie bietet sich an, sie auch in der Schule zumindest gelegentlich zu erwähnen, und gerade der Mathematikunterricht liefert zahlreiche Ansatzpunkte, wo der Stoff mit relativ geringem Aufwand durch Beispiele aus der Kryptologie ergänzt werden kann.

Bei der Diskussion zwischen der Fakultät für Mathematik und Informatik der Universität Mannheim und dem Oberschulamt Karlsruhe über ein Thema für das vierte Nachmittagsseminar wurde daher Kryptologie gewählt, der gemeinsame Oberbegriff für Kryptographie und Kryptanalyse.

Da viele spektakuläre Ergebnisse der heutigen Kryptologie wie etwa das RSA-Verfahren sowohl in populärwissenschaftlichen Zeitschriften und Büchern als auch in Fortbildungsveranstaltungen anderer Institutionen schon mehrfach behandelt wurden, sollte sich dieses Seminar ausschließlich auf die Kryptologie der Vergangenheit und (soweit sie sich absehen läßt) Zukunft beschränken; die Kryptologie der Gegenwart sollte keine Rolle spielen.

Dies konnte naturgemäß nicht hundertprozentig gelingen, da die sich abzeichnenden Entwicklungen mit Quantencomputern und molekularem Rechnen natürlich vor allem im Hinblick auf eine mögliche Bedrohung heutiger Kryptoverfahren interessant sind, aber soweit wie möglich geht es heute nur um

Kryptologie – Gestern und Morgen

Wolfgang K. Seiler
Lehrstuhl VI für Mathematik
Universität Mannheim
68131 Mannheim

seiler@math.uni-mannheim.de
http://hilbert.math.uni-mannheim.de/~seiler

Die Maple *worksheets* des Vortrags sind unter
http://hilbert.math.uni-mannheim.de/~seiler/nachmitts
im Netz zu finden.

I. Klassische Verfahren der Kryptologie

Das wichtigste vorab: Abgesehen von einer einzigen Ausnahme ist keines der hier vorgestellten Verfahren auch nur ansatzweise sicher. Die meisten lassen sich ohne große Kenntnisse und ohne großen Aufwand von Hand knacken; einem Experten, der mit Computer arbeitet, bereitet (abgesehen von der einzigen Ausnahme) keines nennenswerte Schwierigkeiten.

Daraus sollte man folgern, daß diese Verfahren heutzutage keine Rolle mehr spielen, aber dem ist leider nicht so: Viele Verschlüsselungsalgorithmen für Festplatten oder in Textverarbeitungsprogrammen arbeiten noch heute mit Algorithmen wie dem VIGENÉRE-Verfahren, das wir gleich als völlig unsicher erkennen werden. Der einzige „Fortschritt“ besteht darin, daß nun nicht mehr die 26 Buchstaben des Alphabets die Bausteine einer Nachricht bilden, sondern die 256 möglichen Bytes. Das ist nicht nur für den Anwender eine große Bequemlichkeit, sondern auch für den Kryptanalytiker, da es die Redundanz der Nachricht deutlich erhöht und somit bessere Ansatzpunkte zur unbefugten Entschlüsselung bietet.

Dieses Kapitel soll daher nicht nur alte Verfahren vorstellen, sondern auch eine Vorstellung davon vermitteln, mit welch geringem Aufwand man gelegentlich Verschlüsselungsverfahren von (auf den ersten Blick) gewaltiger Komplexität knacken kann.

1. Die Anfänge

Wo nur eine verschwindend kleine Minderheit lesen kann, ist *jede* Schrift eine Geheimschrift; viele der frühen Hochkulturen wie etwa die chinesische sahen daher keinen Grund für die Entwicklung von Verschlüsselungstechniken.

Anderswo schützte man Nachrichten dadurch, daß selbst das *Vorhandensein* einer Nachricht verschleiert wurde; von einem berühmten Beispiel etwa berichtet HERODOT in seinen *Historien:*

Ein gewisser HISTIAEUS, der am persischen Hof lebte, wollte seinen Schwiegersohn, den Tyrannen ARISTAGORAS von Miletos, zu einem Aufstand gegen die Perser bewegen. Dazu rasierte er einem zuverlässigen Sklaven die Haare ab, tätowierte seine Botschaft in dessen Kopfhaut und schickte ihn, nachdem die Haare nachgewachsen waren, zu ARISTAGORAS mit dem Auftrag, sich von diesem die Haare schneiden zu lassen.

Heute bezeichnet man dieses Verschleiern der bloßen Existenz einer Nachricht als *Steganographie;* wo sie erfolgreich ist, besteht keine Notwendigkeit mehr, sich um *Kryptographie* zu kümmern.

In vielen Situationen ist Steganographie undurchführbar, da entweder der Gegner weiß, daß Nachrichten ausgetauscht werden, oder aber weil einfach der Umfang der Kommunikation zu groß ist. Typisches Beispiel ist die militärische Kommunikation, für die erstmals die Spartaner um 487 v.Chr. ein kryptographisches Verfahren benutzten, den sogenannten *Skytale:* Ein Lederband wurde um ein Stück Holz gewickelt, die Nachricht in senkrechten Spalten auf das Leder geschrieben und dieses anschließend von einem Soldaten als Gürtel getragen. Der Empfänger mußte ein identisches Stück Holz haben, konnte das Band darum wickeln und sodann die Nachricht lesen.

Auch in Indien war Kryptographie zu politischen und militärischen Zwecken schon früh bekannt; kurz vor 300 v.Chr. schrieb KAUTILYA sein Buch *Artha-sastra,* in dem er (für Diplomaten) nicht nur die Verwendung von Codes erklärt, sondern auch, wie man diese knacken kann – spätestens seit damals gibt es also zusätzlich zur Kryptographie auch die *Kryptanalyse.* Rund 700 Jahre später bezeichnete VATSAYANA in der *Kama sutra* Kryptographie in Schrift und Wort als zwei der 64 Fähigkeiten, über die jede Frau verfügen sollte.

Der älteste heute noch verwendete Code geht auf CAESAR zurück und ist ein deutlicher Rückschritt hinter die Spartaner und Inder: Genau wie die Römer selbst war auch ihre Kryptographie sehr primitiv. Sie bestand im Falle CAESARS einfach darin, das Alphabet zyklisch um drei Stellen nach rechts zu verschieben, d.h.

$$\mathsf{A} \to \mathsf{D}, \quad \mathsf{B} \to \mathsf{E}, \quad ,\ldots, \quad \mathsf{X} \to \mathsf{A}, \quad \mathsf{Y} \to \mathsf{B}, \quad \mathsf{Z} \to \mathsf{C}\,.$$

Heute bezeichnet man jeden Code, der das Alphabet zyklisch um eine feste Anzahl n von Stellen verschiebt, als CAESAR-*Chiffre;* im Internet bedeutsam ist der Fall $n = 13$, die sogenannte `ROT-13`-Chiffre des Usenet. Sie dient natürlich nicht dazu, Informationen geheimzuhalten: Ihr Gebrauch ist lediglich ein Hinweis darauf, daß der verschlüsselte Text nicht nach jedermanns Geschmack ist und jemand, der ihn trotzdem entschlüsselt, sich nicht über den Inhalt beklagen sollte.

2. Arten kryptanalytischer Angriffe

In seiner Originalform hat CAESARs Verfahren einen wesentlichen Nachteil: Da alles eindeutig festgelegt ist, kann jeder, der irgendwann die Methode erlernte, auch künftig alle Nachrichten lesen – angesichts der zeitlich sehr variablen Bündnisse im römischen Reich sicherlich nicht ganz unproblematisch.

Auch ein Gegner, der es irgendwie geschafft hat, *eine* Nachricht zu entschlüsseln, hat künftig die Möglichkeit *jede* Nachricht zu entschlüsseln. Im Rom des ersten Jahrhunderts, wo wohl niemand die *Artha-sastra* kannte, war das vielleicht kein sehr großes Problem; für die heutigen Massenanwendungen der Kryptographie im Internet und im Bankenbereich ist eine solche Chiffre aber wertlos: Selbst die Militärs gehen schon seit mindestens einem halben Jahrhundert davon aus, daß das *grundsätzliche* Verschlüsselungsverfahren über kurz oder lang auch dem Gegner bekannt sein wird – auch wenn sie selbstverständlich weiterhin alles tun, um dies zu verhindern. Die Spartaner hatten hier vorgesorgt: Auch wer ihr Verfahren kannte, hatte ohne das jeweils passende Stück Holz Schwierigkeiten, die Nachricht zu lesen.

Abgesehen von der Original CAESAR-Chiffre hat daher praktisch jedes kryptographische Verfahren zwei Komponenten: Einmal die *Verschlüsselungsmethode,* bei der man realistischerweise davon ausgehen muß, daß man sie nicht unbegrenzt geheimhalten kann, und dann noch einen *Schlüssel,* der bei der Anwendung dieser Methode benutzt wird und der häufig gewechselt werden muß. Als *sicher* darf man ein Kryptoverfahren nur dann betrachten, wenn ein Gegner trotz genauer Kenntnis des Verfahrens nicht in der Lage ist, einen Text ohne Kenntnis des Schlüssels zu dechiffrieren.

Der letzte Satz klingt vernünftig, ist aber unvollständig, da er eine wesentliche Frage ausklammert: *Was darf der Gegner alles wissen, bevor die Sicherheit des Verfahrens* (der Schlüssel) *gefährdet ist?*

Die offensichtliche Antwort scheint zu sein, daß er den chiffrierten Text und das Verfahren kennt. Das ist aber leider viel zu optimistisch: Schon 725 beschrieb Abu Yahmadi, der Autor des ersten arabischen Wörterbuchs, wie er ein altes byzantinisches Kryptogramm löste ausgehend von der Annahme, daß es mit der Formel „Im Namen Gottes“ beginne – für einen Text aus der fraglichen Zeit fast eine Selbstverständlichkeit.

Heute beginnen wir Texte nicht mehr mit der Formel „Im Namen Gottes“, aber wir verwenden Word-Dateien, zip-Dateien *usw.,* die mit *magic bytes* und anderer Vorspann-Information beginnen; einige von uns versenden e-mails, bei denen die ASCII-Kunst oder die Visitenkarte am Ende länger ist als die Nachricht selbst; Militärs beginnen Nachrichten typischerweise mit Worten wie „Wehrmachtskommando XY an ...“; Banken haben sich für elektronische Überweisungen auf ziemlich rigide Formate geeinigt; jeder Besitzer einer EC-Karte kennt seine Geheimzahl und die sonstige Kontoinformation auf der Karte und kann, so er einen Magnetkartenleser hat, auch die verschlüsselte Version auf der Karte lesen: Falls er mit dieser Information (eventuell auch von vielen EC-Karten statt nur von einer) die zehnstellige Schlüsselzahl des Systems ermitteln könnte, müßten sofort alle Geldautomaten abgeschaltet werden.

Ein Gegner kennt daher in vielen Fällen nicht nur den Chiffretext, sondern er kennt auch zumindest einen Teil des zugehörigen Klartexts. Ein gutes Kryptoverfahren muß sicherstellen, daß er trotzdem nicht in der Lage ist, den Schlüssel zu identifizieren und damit auch den ihm unbekannten Rest zu entschlüsseln.

In manchen Situationen ist der Gegner sogar in einer noch besseren Position: Er kann sich zu von ihm frei gewähltem Klartext den Chiffretext verschaffen. Eine solche Situationen könnte beispielsweise auftreten, wenn ein Diplomat ein Schreiben übergibt, bei dem er ziemlich sicher sein kann, daß es anschließend verschlüsselt weitergegeben wird. Kurz vor Ausbruch des zweiten Weltkriegs übergab sogar einmal ein japanischer Diplomat ein Schreiben an das amerikanische Außenministerium mit der Bitte, es nach Japan zu übermitteln, da er den Beamten der eigenen Chiffrierabteilung mißtraue – ob dahinter ein Versuch einer solchen Attacke lag, ist bis heute unbekannt. (Die Amerikaner übermittelten das Dokument.)

Heute bilden auch Paßwortdateien wenig gesicherter Computersysteme einen Ansatz für solche Attacken: Jeder Benutzer kann sein Paßwort beliebig oft ändern und auf jeden beliebigen Wert setzen; das Ergebnis wird verschlüsselt in einer Paßwortdatei gespeichert, die auf manchen Computern für jeden Benutzer lesbar ist.

Auch Smartcards bieten Ansätze für Angriffe mit frei wählbarem Klartext: Sichere Kryptosysteme arbeiten mit Schlüssellängen, die weit jenseits dessen liegen, was sich ein Mensch merken kann; sie können daher nur mit Smartcards oder etwas ähnlichem funktionieren. Diese sind natürlich *(?)* zusätzlich durch eine Geheimzahl oder ein Paßwort gesichert, aber viele unbedarfte Benutzer stellen praktisch sicher, daß ihre Karte nie ohne Paßwort gestohlen werden kann. Damit ist der Dieb in der Lage, die Karte zum Lesen verschlüsselter Nachrichten, zur elektronischen Unterschrift mit dem Namen des Bestohlenen und in Zugangskontrollsystemen verwenden – bis der Diebstahl entdeckt wird.

Eine vielseitig einsetzbare Karte wird sicherlich schnell vermißt werden; danach wird der in die Karte kodierte Schlüssel für ungültig erklärt und somit nutzlos. Ein geschickter Dieb muß also die Karte wieder zurückgeben, *bevor* der Verlust bemerkt wird. Die Schlüsselinformation kann er nicht kopieren: Die steht in einem auslesesicheren Register. Er kann aber beliebig viel Eingabetext seiner Wahl von der Karte bearbeiten lassen, und ein gutes Kryptosystem muß sicherstellen. daß er trotzdem nichts über den Schlüssel erfährt und ohne die Karte später nicht mehr in der Lage ist, verschlüsselte Nachrichten an den Besitzer zu dechiffrieren, in dessen Namen zu unterschreiben oder sich als diesen auszugeben.

3. Kryptanalyse der Caesarchiffre

Es ist klar, daß auch eine CAESAR-Substitution mit unbekannter Verschiebung n keinen Schutz bietet gegen einen Angreifer, der Klartext und zugehörigen Chiffretext kennt: Sobald er nur für einen einzigen Buchstaben sowohl diesen als auch seine Verschlüsselung kennt, kann er n berechnen.

Aber auch ein Angreifer, der allein mit Chiffretext arbeiten muß, hat keine großen Schwierigkeiten: Er kann beispielsweise einfach alle 26 Möglichkeiten $n = 0, \ldots, 25$ durchprobieren; sobald sinnvoller Klartext entsteht, hat er den Schlüssel gefunden. Informationstheoretische Überlegungen zeigen, daß im Allgemeinen schon nach zwei Buchstaben klar ist, ob es sich bei einer Probeentschlüsselung um deutschen Klartext handeln kann oder nicht.

Es geht aber noch einfacher: Bekanntlich kommen die 26 Buchstaben des Alphabets in deutschem Text nicht mit gleicher relativer Häufigkeit vor. Abbildung eins zeigt die Häufigkeitsverteilung der 259 633 Buchstaben aus dem Roman *Dr. Katzenbergers Badereise* von JEAN PAUL. (Wer nachzählen möchte, findet diesen und viele andere Texte im Internet auf

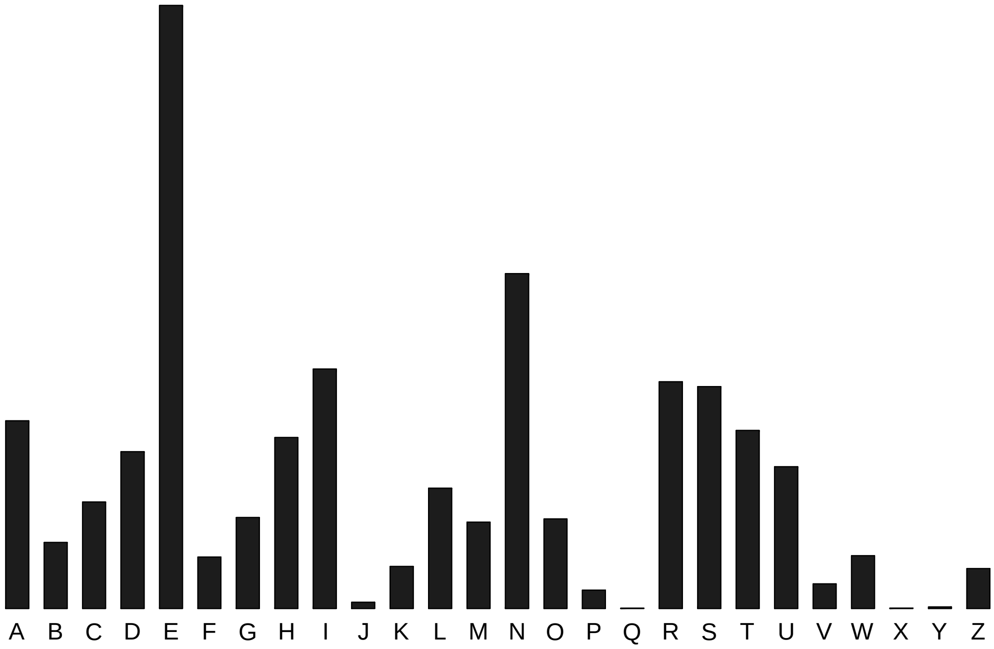


Abb. 1: Buchstabenhäufigkeiten in deutschem Text

den Seiten des Projekts Gutenberg unter http://gutenberg.aol.de/ und bei mehreren mirror sites.)

In Prozentzahlen ausgedrückt sieht das Ergebnis so aus:

A	B	C	D	E	F	G	H	I	J	K	L	M
5,75	2,03	3,27	4,81	18,49	1,58	2,79	5,25	7,35	0,19	1,29	3,69	2,65

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
10,27	2,75	0,57	0,01	6,95	6,81	5,46	4,35	0,76	1,62	0,02	0,05	1,23

Wie zu erwarten war, ist also E mit großem Abstand der häufigste Buchstabe, während Q, X und Y sehr selten vorkommen.

Wenn ich einen anderen Text als *Dr. Katzenbergers Badereise* ausgezählt hätte, wären die Ergebnisse natürlich anders gewesen: Das Spiegel-Interview mit dem Nobelpreisträger Horst Störmer etwa hat 5455 Buchstaben, die sich prozentual folgendermaßen verteilen:

A	B	C	D	E	F	G	H	I	J	K	L	M
6,00	2,07	2,84	4,16	17,23	1,50	2,97	4,68	8,18	0,26	1,20	3,81	3,27

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
10,45	3,04	1,05	0,07	6,93	6,84	5,56	4,42	0,73	1,43	0,02	0,33	1,03

Zumindest in absoluten Prozentzahlen sind die Unterschiede praktisch vernachlässigbar; die *relativen* Unterschiede sind bei den häufigen Buchstaben ebenfalls unbedeutend, bei den seltenen allerdings teilweise recht dramatisch.

Bedenkt man allerdings, daß 0,02% X bei einem Text aus 5455 Buchstaben gleichbedeutend ist mit nur einem X und 0,07% J mit vier J, so sieht man, daß auch diese Abweichungen minimale Zufallsschwankungen sind.

Für sehr kurze Texte spielen solche Zufallsschwankungen eine große Rolle: Falls wir etwa das Kryptogramm

`DQJUL IILPP RUJHQ JUDXH Q`

entschlüsseln wollen, zeigt uns Abbildung zwei, daß es dort gleich drei häufigste Buchstaben gibt: J, Q und U kommen je dreimal vor. Entschlüsselt man unter der Annahme, daß J dem E entspricht, ist $n = 5$ und das Entschlüsselungsergebnis `YLEPG DDGKK MPECL EPYSC L` enthält zwar drei E, hat aber sonst nur wenig mit deutschem Text zu tun. Auch die Entschlüsselungsversuche

`REXIZ WWZDD FIXVE XIRLV E` und `NATEV SSVZZ BETRA TENHR A`,

die von $E \to Q$ (d.h. $n = 12$) und $E \to U$ (d.h. $n = 16$) ausgehen, liefern kein vernünftiges Ergebnis.

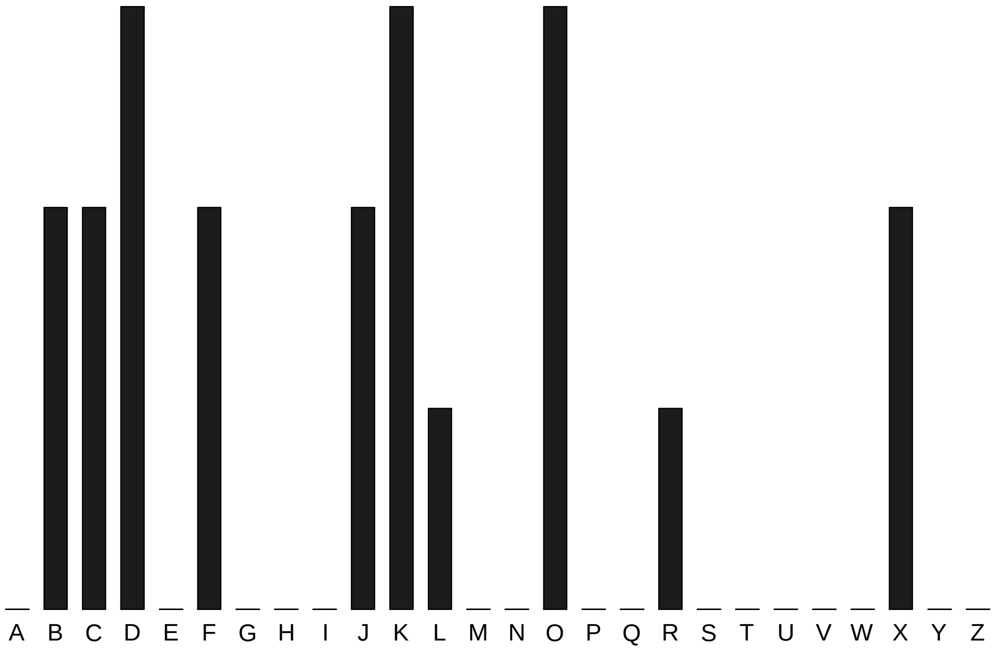


Abb. 2: Buchstabenhäufigkeiten eines kurzen Kryptogramms

Ein klassischer Kryptanalytiker wäre anders vorgegangen: Er wäre sich beim Blick auf Abbildung zwei praktisch sicher gewesen, daß die drei Leerstellen am Ende und die beiden am Anfang wohl nur den fünf seltenen Buchstaben von V bis Z entsprechen können, so daß Abbildung eins um drei Stellen verschoben wurde, d.h. A wurde als D verschlüsselt *usw.,* wie es schon CAESAR praktizierte. Dies führt zum in fast jedem Lehrbuch der klassischen Kryptographie zu findenden Klartext `ANGRIFF IM MORGENGRAUEN`.

Ein moderner Kryptograph möchte sich mit trivialen Kryptogrammen wie dem gerade betrachteten nicht mehr selbst beschäftigen; für solche Dreckarbeit hat er schließlich seine Computer. Denen springt keine Lücke *ins Auge,* und sie haben auch Schwierigkeiten, Lücken *exakt* von Nichtlücken zu unterscheiden. Diese Schwierigkeiten hatte der klassische Kryptanalytiker zwar auch, aber er konnte sich da im Gegensatz zu einem Computer meist auf sein Gefühl verlassen – auch wenn ihn das immer wieder einmal gewaltig in die Irre schickte.

Computer dagegen lieben Zahlen, und auch dazu können ihnen die klassischen Kryptanalytiker verhelfen: Sei p_i die Häufigkeit des i-ten Buchstabens des Alphabets in deutschen Texten, geschätzt etwa durch die Prozentzahlen in obiger Tabelle, und sei q_i die Häufigkeit dieses Buchstabens im zu untersuchenden Kryptogramm. Im obigen Beispiel wäre also

$$q_{10} = q_{17} = q_{21} = \frac{3}{21} = \frac{1}{7},$$

da J, Q und U jeweils dreimal vorkommen;

$$q_4 = q_8 = q_9 = q_{12} = q_{16} = \frac{2}{21},$$

da D, H, I, L und P je zweimal vorkommen,

$$q_{17} = q_{24} = \frac{1}{21}$$

für die beiden einfach vorhandenen Buchstaben R und X, und alle übrigen q_i verschwinden.

Dazu kann man die Zahlen

$$r_n = \sum_{i=1}^{26} p_i q_{i \oplus n}$$

betrachten, wobei $i \oplus n$ die auf den Bereich von 1 bis 26 normierte Summe von i und k modulo 26 ist, d.h.

$$i \oplus n = 1 + \big((i+n-1) \bmod 26\big)\,.$$

Um die Bedeutung der r_n zu verstehen, betrachten wir zunächst

$$r_0 = \sum_{i=1}^{26} p_i q_i\,.$$

Da $(p_i - q_i)^2 = p_i^2 + q_i^2 - 2p_i q_i$ nicht negativ sein kann, ist

$$p_i q_i \le \frac{1}{2}(p_i^2 + q_i^2) \quad \text{und} \quad r_0 = \sum_{i=1}^{26} p_i q_i \le \frac{1}{2}\sum_{i=1}^{26}(p_i^2 + q_i^2)\,.$$

Die beiden Seiten dieser Ungleichung unterscheiden sich dabei umso mehr, je größer $\sum (p_i - q_i)^2$ ist, je mehr sich also die beiden Häufigkeitsverteilungen unterscheiden. Die Ungleichung wird genau dann zur Gleichung, wenn $p_i = q_i$ ist für alle i, wenn also die Buchstaben im Kryptogramm dieselbe Häufigkeitsverteilung haben wie die Buchstaben der deutschen Sprache.

Für die übrigen r_n läßt sich genauso argumentieren: Da die Summe der p_i^2 nicht von der Reihenfolge der Summanden abhängt, ist auch

$$r_n = \sum_{i=1}^{26} p_i q_{i\oplus n} \le \frac{1}{2}\sum_{i=1}^{26}(p_i^2 + q_{i\oplus n}^2) = \frac{1}{2}\sum_{i=1}^{26}(p_i^2 + q_i^2)$$

mit Gleichheit genau dann, wenn $p_i = q_{i\oplus n}$ ist für alle i.

Wir erwarten also einen großen Wert von r_n, wenn die um n verschobene Häufigkeitsverteilung des Kryptogramms gut mit der Häufigkeitsverteilung der deutschen Sprache übereinstimmt oder, anders ausgedrückt, wenn das à la CAESAR um n Buchstaben nach vorne verschobene Kryptogramm eine Häufigkeitsverteilung hat, die gut mit der deutscher Texte übereinstimmt.

Im vorliegenden Fall sind die größten Werte

$$r_{16} = 0{,}0707, \quad r_3 = 0{,}0661, \quad r_{12} = 0{,}0541 \quad \text{und} \quad r_7 = 0{,}0538\,.$$

Der erste Versuch mit $n = 16$ führt, wie wir bereits wissen, zu keinem vernünftigen Ergebnis; der zweite mit $n = 3$ entschlüsselt aber das Kryptogramm.

4. Vigenère-Chiffren

CAESAR-Substitutionen sind so einfach zu entschlüsseln, weil es erstens nur 26 Verschlüsselungsmöglichkeiten gibt und zweitens die Häufigkeitsverteilung der Buchstaben viele Anhaltspunkte gibt. Zur Verbesserung des Verfahrens muß also erstens die Anzahl der Verschlüsselungsmöglichkeiten deutlich erhöht werden, und zweitens sollte man versuchen, die Häufigkeitsverteilung der Buchstaben im Chiffretext möglichst homogen zu machen.

Beide Ziele erreicht die VIGENÈRE-Chiffre, indem sie mehrere CAESAR-Substitutionen miteinander mischt: Schlüssel ist eine (nicht zu kurze) Buchstabenfolge. Diese schreibt man so oft hintereinander, bis eine Buchstabenfolge entsteht, die mindestens genauso lang ist wie der zu verschlüsselnde Text, und man verschlüsselt den i-ten Klartextbuchstaben dadurch, daß man ihn zum i-ten Buchstaben der aus dem Schlüssel entstandenen Buchstabenfolge „addiert". Bei dieser „Addition" wird A mit 1, B mit 2 identifiziert *usw.*, und ein Additionsergebnis größer 26 wird durch Subtraktion von 26 wieder in den richtigen Bereich gebracht. Beispielsweise ist also

$$\mathsf{A}+\mathsf{A}=\mathsf{B}, \quad \mathsf{B}+\mathsf{B}=\mathsf{D}, \quad \mathsf{A}+\mathsf{Y}=\mathsf{Z} \quad \text{und} \quad \mathsf{B}+\mathsf{Y}=\mathsf{A}\,.$$

Verschlüsselung von `Angriff im Morgengrauen` mit dem Schlüssel `Krypto` führt somit zu

```
  ANGRI FFIMM ORGEN GRAUE N
+ KRYPT OKRYP TOKRY PTOKR Y
= LFFHC UQALC IGRWM WLPFW M
```

VIGENÈRE-Chiffren sind benannt nach BLAISSE DE VIGENÈRE, der sie 1586 in seinem Buch *Traictés des chiffres ou secrètes manières d'escrire* beschrieb; sie wurden allerdings bereits im 1518 erschienenen Buch *Polygraphia* des Abbé JEAN TRITHÈME erwähnt und gehen laut VIGENÈRE zurück auf die Hebräer.

L. SACCO, der ehemalige Chef des Chiffrierdienstes der italienischen Armee, schreibt in seinem *Manuel de cryptographie* (Paris, 1951):

> *Vigenère réussit à produire un chiffre plus faible et moins commode que les précédents, qui n'en connut pas moins une fortune imméritée, particulièrement dans les cent dernières années, où il fut adopté par de nombreuses armées, même après la publication d'un moyen de décryptement (1863), indice évident de décadence dans la domaine de la cryptographie.*

Die VIGÈNERE-Chiffre oder ähnliche Verfahren wurden bis Ende des neunzehnten Jahrhunderts unter anderem von der österreichischen, der französischen und der italienischen Armee benutzt; von letzterer sogar bis 1917. In verschiedenen Computerprogrammen ist ähnliches heute noch zu finden.

Um zu verstehen, warum SACCO trotzdem so abfällig darüber spricht, müssen wir ihre Sicherheit etwas genauer betrachten und insbesondere sehen, wann sie mit welchem Aufwand geknackt werden kann.

Für einen Schlüssel mit nur einem Buchstaben haben wir einfach eine CAESAR-Substitution, die praktisch keinerlei Schutz bietet – es sei denn, die Nachricht sei nur einen Buchstaben lang. In diesem Fall bietet die CAESAR-Chiffre perfekte Sicherheit gegen eine Attacke nur mit Chiffretext, denn genau wie der Schlüsselbuchstabe kann auch der Klartextbuchtabe beliebig sein.

Genauso bietet die VIGENÈRE-Chiffre absolute Sicherheit gegen einen Angriff nur mit Chiffretext, wenn der Schlüssel mindestens genauso lang ist wie die Nachricht und zufällig gewählt wird. Verschlüsseln wir die wohlbekannte Nachricht mit einem zufällig gewählten Schlüssel der Länge 21, so erhalten wir etwa

```
  ANGRI FFIMM ORGEN GRAUE N
+ KCHQR OFVFN FVSLA XRQBV E
= LQOIA ULESA UNZQO EJRWA S
```

Nun könnte möglicherweise ein Gegner irgendwie den Schlüssel erraten und damit auf den Klartext stoßen. Das nützt ihm aber nur wenig, denn es gibt noch viele andere Klartexte, die zu diesem Kryptogramm passen: Wenn er andere Schlüssel ausprobiert, erhält er beispielsweise auch die Entschlüsselung

```
  LQOIA ULESA UNZQO EJRWA S
– JYFUT PJSXN PZTVJ MWWCG J
= BRING EBLUM ENFUE RMUTT I
```

und entsprechend läßt sich auch jeder andere Klartext mit 21 Buchstaben produzieren; er kann aus dem Kryptogramm also keine weitere Information ziehen, als daß der Klartext 21 Buchstaben lang war. Wenn in einem Kryptosystem der Schlüssel mindestens genauso komplex ist wie der Chiffretext, läßt sich durch geschickte Wahl von Schlüssel und Verschlüsselungsverfahren fast jede Botschaft in die Chiffre hineinlesen – was von Personen, die

Nachrichten von Außerirdischen bekommen und/oder Verschwörungstheorien beweisen, bewußt oder unbewußt durchaus auch ausgenutzt wird.

Für die Sicherheit eines VIGENÈRE-Verfahrens mit Schlüsserllänge = Nachrichtenlänge ist ganz wesentlich, daß der Schlüssel *zufällig* gewählt wird; er darf auf keinen Fall beispielsweise ein Buch sein, denn die „Summe“ (im VIGENÈRE-Sinne) zweier deutscher Klartexte hat noch so viele statistische Gesetzmäßigkeiten, daß die beiden Summanden relativ einfach ermittelt werden können.

Das gerade vorgestellte perfekt sichere Verfahren bezeichnet man in der Kryptographie als *one time pad,* da es früher meist mit Schreibblöcken realisiert wurde: Man erzeugte durch einen Zufallsprozeß Buchstabenfolgen und notierte diese auf zwei Schreibblöcken in gleicher Weise, bis die Blöcke voll waren. Diese beiden Blöcke gingen dann an die beiden Kommunikationspartner, die für ihre i-te Nachricht die Buchstabenfolge auf Blatt i des Blocks als Schlüssel benutzten; danach wurde dieses Blatt vernichtet. Als erstes benutzte diese Technik wohl der amerikanische General VERNAM im ersten Weltkrieg; im zweiten Weltkrieg kommunizierte London auf diese Weise mit der französischen *Résistance,* und später sicherten auch Fidel Castro und Che Guevara ihre Kommunikation auf diese Weise.

Als *high tech*-Variante davon entstand im Kalten Krieg das *rote Telephon* zwischen dem Weißen Haus und dem Kreml. Dabei handelte es natürlich nicht wirklich um kein Telephon, über das sich etwa Carter und Breschnjew in einen russisch-englischen Kauderwelsch unterhalten konnten, sondern um eine Telexverbindung mit je einem russischen und einem englischen Fernschreiber an den beiden Endpunkten. Zur Sicherung dieser Verbindung dienten Magnetbänder mit Zufallsbits, die über Kuriere ausgetauscht und genau nach Art des *one time pad* benutzt wurden.

In alltäglicheren Anwendungen ist ein solcher Aufwand meist zu hoch; man muß notgedrungen mit Schlüsseln arbeiten, die deutlich kürzer sind als die (Summe der) Nachrichten, die damit verschlüsselt werden müssen. Wir wollen sehen, wie sich das VIGENÈRE-System dann verhält.

Betrachten wir ein Beispiel-Kryptogramm:

```
PDOAA KKMQB LYORJ FTLXM OQGYU XTKCQ LXLVB ATCBU MJEDM SQZJJ
OZPHT AEZZD FFRSK XTYZV MVVZS QTZUO CTGDX ENOGX XGCOI GXHBN
OZXCC COJXJ PBQTV XTDOF ZRZND FHADX LZCQC NPBSL HTDVM ESKSP
YFCDB QHCEV ZDVIA DRKTR PGJDQ RFUUW AKQXP UZQVD AMXLF XGFGC
BTCFT KYRES GSALT VGZWT VSQOP UCALM ZFGMA VNDOZ ZNJOA HVDDQ
CMONK CPPBU ZZZYK PYRAD LZLYV GXNUB IURKX OEBBZ DNAWE UOVKH
TPISF DLIIQ LOXHO PAIZZ CAWEV XYSXU IZVUQ MAICE SKYXL AIZEH
```

```
KVNCR KUXYH IFKMK BJVHO TRSXX ZIEZJ
```

Abbildung drei zeigt, daß die Buchstabenhäufigkeiten sehr viel gleichmäßiger verteilt sind als in deutschem Text; dies deutet darauf hin, daß keine *monoalphabetische,* sondern eine *polyalphabetische* Substitution angewandt wurde, d.h. die verschiedenen Buchstaben des Kryptogramms wurden nicht alle mittels ein und derselben Permutation aus den Klartextbuchstaben berechnet.

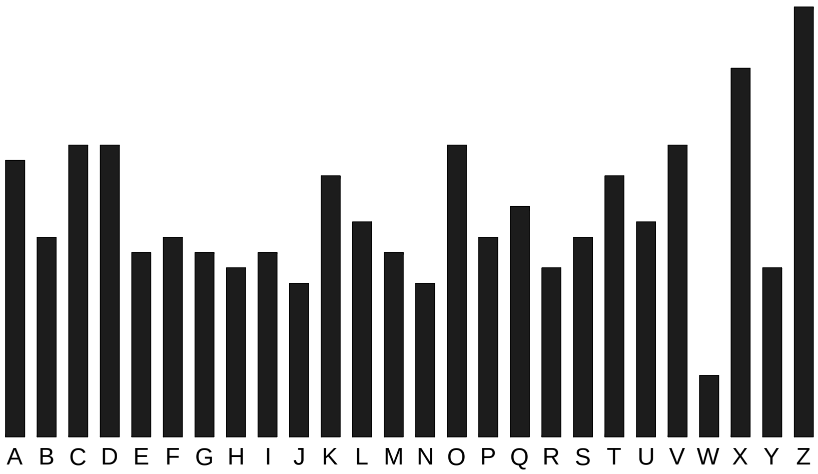


Abb. 3: Buchstabenhäufigkeiten des Beispielkryptogramms

Es gibt polyalphabetische Substitutionen wie etwa den *one time pad,* die ohne Kenntnis von Verfahren und Schlüssel nicht entschlüsselbar sind. Erfolg wird der Kryptanalyst immer dann haben, wenn er die Buchstaben identifizieren kann, die mit derselben Permutation verschlüsselt wurden, und wenn er für (fast) jede Permutation genügend viele Buchstaben hat, um statistische Methoden anzuwenden.

Am einfachsten ist dies bei *periodischen* Verfahren, d.h. wenn die anzuwendende Permutation so wie beim VIGENÈRE-Verfahren periodisch wechselt. In diesem Fall muß er nur die Periode feststellen, um die mit derselben Permutation verschlüsselten Buchstaben zusammenzufassen.

Mit dieser Ermittlung der Periode hatten die Kryptanalytiker im letzten Jahrhundert lange ihre Probleme; den ersten Ansatz fand der preußische Offizier FRIEDRICH W. KASIKI (1805–1881) und veröffentlichte ihn 1863 in einem damals kaum beachteten nur 95 Seiten dicken Buch mit dem Titel *Die Geheimschriften und die Dechiffrierkunst.* Seine Idee war die folgende: Gewisse Wörter und Buchstabenkombinationen wie etwa die bestimmten Artikel sind in fast allen Texten sehr häufig. Wenn nun ein langer

Text mit einem polyalphabetischen Verfahren kurzer Periode verschlüsselt wird, ist es sehr wahrscheinlich, daß solche Buchstabenfolgen mehrfach auf die gleiche Weise verschlüsselt werden. Man suche daher im Kryptogramm nach zweimal vorkommenden Buchstabenfolgen (Heute nennt man so etwas ein KASIKI-Paar) und berechne deren Abstand. Die Periode sollte ein Teiler von relativ vielen dieser Abstände sein, wobei Abstände, die zu langen Buchstabenfolgen gehören, natürlich höher zu gewichten sind als solche, die etwa nur zu Buchstabenpaaren gehören: Bei letzteren ist die Wahrscheinlichkeit, daß es sich um eine zufällige Koinzidenz handelt, erheblich größer.

Die Suche nach KASIKI-Paaren ist recht aufwendig, und sie führen im allgemeinen nur dann zu einer Lösung, wenn das Kryptogramm erheblich länger als ist als der verwendete Schlüssel. Eine Alternative fand um 1920 der wohl bedeutendste der klassischen Kryptologen, WILLIAM FRIEDMAN (1891–1969). Er wurde als WOLFE FRIEDMAN in Russland geboren, aber als seine Eltern 1892 nach Amerika emigrierten, änderten sie seinen Vornamen. Er studierte zunächst Landwirtschaft, spezialisierte sich später auf Genetik und bekam 1915 eine Stelle als Genetiker bei dem Textilkaufmann GEORGE FABYAN, auf dessen Gut Riverbank in Geneva, Illinois. Dieser unterhielt dort Laboratorien für Akustik, Chemie, Genetik und Kryptologie – letztere mit dem Ziel zu beweisen, daß BACON der wahre Autor der SHAKESPEAREschen Schriften sei. Dadurch mußte sich FRIEDMAN zwangsläufig auch für Kryptologie interessieren und war damit so erfolgreich, daß er bald Leiter der Laboratorien sowohl für Genetik als auch für Kryptologie war.

Mit dem Kriegseintritt der Vereinigten Staaten im April 1917 mußte sich auch die amerikanische Armee für Kryptographie interessieren, und da es außer Riverbank kein amerikanisches Zentrum für Kryptologie gab, wurden nicht nur aufgefangene Kryptogramme dorthin geschickt, sondern auch Armee-Offiziere, die dort bei FRIEDMAN in Kryptanalyse ausgebildet werden sollten. Diese Kurse wurden nach Kriegsende fortgesetzt, und 1921 verließ FRIEDMAN Riverbank, um anschließend in verschiedenen militärischen Funktionen sowohl Codes zu entwerfen als auch Codes zu knacken. Von ihm stammt das Wort *Kryptanalyse,* das das unbefugte Dechiffrieren vom legitimen unterscheidet.

Seine 1925 perfektionierte Idee zur Bestimmung der Periode ist folgende: Man betrachte für eine natürliche Zahl n die Wahrscheinlichkeit dafür, daß ein Buchstabe mit seinem n-ten Nachfolger übereinstimmt.

Falls n ein Vielfaches der Periode ist, wurden die beiden Buchstaben mit derselben Permutation verschlüsselt; da Summen nicht von der Reihenfolge der Summanden abhängen, ist die Wahrscheinlichkeit also

$$\sum_{i=1}^{26} p_i^2 ,$$

wobei p_i die Häufigkeit des i-ten Buchstabens in typischem Klartext ist.

Falls n dagegen *kein* Vielfaches der Periode ist, stammen ein Buchstabe und sein n-ter Nachfolger bei hinreichend langer Periode fast immer aus verschiedenen Permutationen, man kann sie also in allererster Näherung als zufallsverteilt ansehen. Dann ist die Wahrscheinlichkeit einer Koinzidenz ungefähr gleich

$$\sum_{i=1}^{26} \left(\frac{1}{26}\right)^2 = \frac{1}{26} .$$

Betrachtet man die deutsche Sprache auf Grundlage von *Dr. Katzenbergers Badereise,* so ist

$$\sum_{i=1}^{26} p_i^2 \approx 0{,}0789 \quad \text{und} \quad \frac{1}{26} \approx 0{,}0385 ,$$

die beiden Werte unterscheiden sich also deutlich.

In einem relativ kurzen Kryptogramm wird man selbstverständlich andere Werte berechnen, aber trotzdem sollte es im allgemeinen für gleiche Alphabete mehr Koinzidenzen geben als für verschiedene.

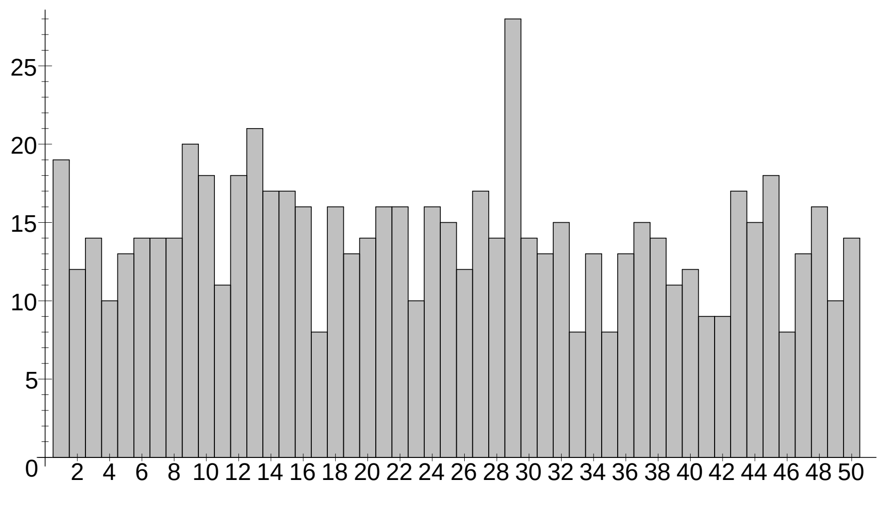


Abb. 4: Koinzidenzen im Beispielkryptogramm für Abstände bis 50

Für das obige Kryptogramm zeigt Abbildung vier die Verteilung der Koinzidenzen bei n Buchstaben Abstand. Das erste Maximum bei $n = 1$ ist natürlich nicht ernstzunehmen; sonst hätten wir eine monoalphabetische Substitution, deren Häufigkeitsverteilung deutlich anders aussieht als Abbildung drei. Auch die Spitzen bei $n = 9$ und $n = 13$ sind wohl eher zufällig, denn bei $n = 18$ und $n = 26$ sind die Anzahlen eher klein. Unübersehbar ist das absolute Maximum bei $n = 29$; um zu sehen, ob dies das „richtige" Maximum ist, müssen wir allerdings etwas mehr Werte berechnen. Abbildung fünf zeigt die entsprechenden Abstände bis einschließlich 120, und man sieht doch recht deutliche Ausschläge bei $n = 58$, 67 und 116. Wir wollen daher als erstes versuchen, das Kryptogramm unter der Annahme zu dechiffrieren, daß $n = 29$ ist.

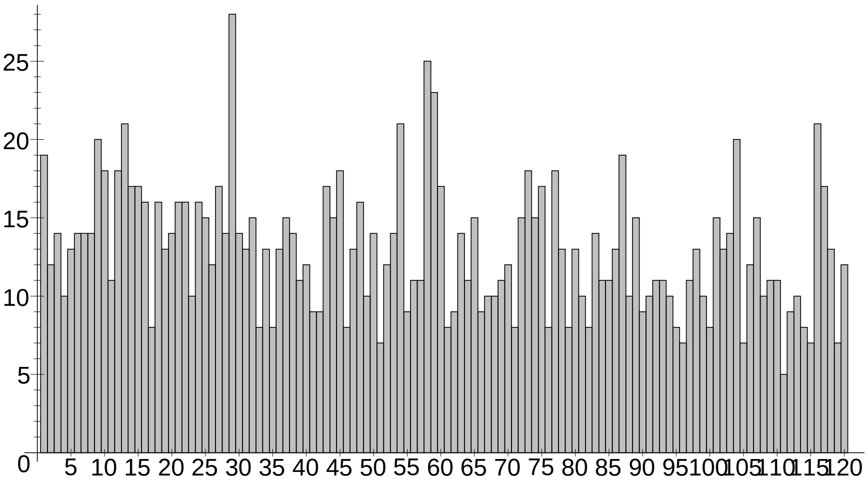


Abb. 5: Koinzidenzen im Beispielkryptogramm für Abstände bis 120

Wir arbeiten also in diesem Ansatz mit der Hypothese, daß zwei Buchstaben genau dann mit derselben Permutation verschlüsselt sind, wenn ihr Abstand ein Vielfaches von 29 ist.

Deshalb teilen wir das Kryptogramm auf in 29 Buchstabenfolgen, die jeweils mit derselben Permutation verschlüsselt sein sollten. Ein Kryptanalytiker würde nun die 29 Häufigkeitsverteilungen dazu betrachten; da sie meisten ungefähr so aussehen wie die von CAESAR-Substitutionen, würde er VIGENÈRE als wahrscheinlichste Möglichkeit ansehen.

In erster Näherung kann man CAESAR-Chiffren über den pq-Test entschlüsseln; wendet man dies hier an, erhält man folgende Kandidaten für Schlüsselbuchstaben, wobei links jeweils der mit der größten $\sum p_i q_i$ steht, danach die vier mit den nächstkleineren:

N	0,085	E	0,057	Z	0,055	O	0,050	A	0,048
A	0,083	E	0,062	N	0,047	Z	0,046	K	0,045
T	0,080	D	0,068	C	0,064	G	0,059	P	0,057
H	0,077	I	0,054	L	0,051	Y	0,049	G	0,049
M	0,077	Z	0,067	Q	0,055	D	0,051	I	0,051
B	0,055	I	0,054	F	0,053	J	0,053	P	0,048
P	0,074	T	0,071	C	0,059	M	0,058	D	0,056
T	0,069	G	0,055	E	0,052	R	0,050	I	0,048
A	0,081	N	0,070	J	0,061	L	0,050	K	0,047
G	0,092	C	0,075	F	0,064	P	0,056	Q	0,048
S	0,079	W	0,066	J	0,060	F	0,055	N	0,054
S	0,061	F	0,060	R	0,053	W	0,052	J	0,047
E	0,080	R	0,056	D	0,054	I	0,052	V	0,049
I	0,070	M	0,062	W	0,056	L	0,052	P	0,051
I	0,089	E	0,059	H	0,053	R	0,053	W	0,051
A	0,080	E	0,079	N	0,074	R	0,055	G	0,052
A	0,067	K	0,061	B	0,054	O	0,050	Z	0,050
R	0,076	N	0,072	W	0,056	B	0,054	I	0,049
K	0,072	B	0,060	E	0,059	V	0,056	U	0,051
R	0,075	E	0,055	N	0,054	Q	0,050	D	0,049
Y	0,069	S	0,059	P	0,058	J	0,054	W	0,054
P	0,099	C	0,061	L	0,053	M	0,051	Y	0,049
T	0,092	X	0,071	G	0,061	K	0,050	H	0,045
O	0,080	K	0,068	B	0,066	F	0,063	S	0,047
L	0,059	P	0,056	W	0,055	M	0,051	X	0,050
O	0,083	K	0,068	B	0,058	G	0,056	N	0,052
G	0,080	C	0,068	K	0,062	D	0,055	R	0,050
I	0,083	E	0,059	Z	0,053	V	0,053	R	0,051
E	0,107	N	0,059	A	0,056	R	0,054	I	0,048

Die Buchstabenfolge in der ersten Spalte legt nahe, daß hier kryptographisch unsorgfältig gearbeitet wurde: Der Schlüssel wurde wohl nicht zufällig gewählt, sondern als sinnvoller Teil der deutschen Sprache, wobei wir aber nicht alle Buchstaben auf Anhieb richtig erraten haben. Wir können nun entweder versuchen, den Schlüssel anhand der Buchstaben aus den folgenden Spalten zu erraten (was hier wohl selbst ohne diese nicht sonderlich schwerfällt), oder aber wir entschlüsseln einfach mit dem wahrscheinlich falschen Schlüssel und korrigieren anhand des entschlüsselten Textes. Wenn wir diesen in Zeilen der Länge 29 aufschreiben, stehen jeweils die Buchstaben, die zum gleichen Alphabet gehören, untereinander, so daß es für jeden Schlüsselbuchstaben mehrere Überprüfungsmöglichkeiten gibt.

Der Entschlüsselungsungsversuch führt auf folgendes Ergebnis: (In der ersten Zeile steht der Schlüssel.)

```
N A T H M B P T A G S S E I I A A R K R Y P T O L O G I E

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

Damit sollte wohl jeder Leser den Klartext rekonstruieren können. Bei einer Schlüssellänge von 29 und einer Textlänge von 480, bei etwa 16,5 Zeichen pro Alphabet also, ist das Vigenère-Verfahren somit schon völlig unsicher. Bei noch mehr Buchstaben pro Alphabet wird das Knacken noch einfacher: Betrachten wir als Beispiel denselben Klartext wie oben, chiffriert mit einem kürzeren Schlüssel:

```
PZWDV AKLNZ ZDMDE MGYNZ VNGSC DVFAD YTFDP PVKOS BFMYT SUDND
JOMAO MJVIQ BMQUQ MZAAV XNAEO UXQFX IDXNM UYHDR AFEHO AQVZN
JPRIQ EBUGC QGRVJ XPLXS IROTX LMMUF ZILPT KKIHM MHWXD NYIIJ
NESVD VTGDX ZZSOA JNJEU ZZLHQ LUXMA CLXJE EZPXQ NXUYJ IIBYW
ETCFN MSXZH FOPLS FPZFG GCUGR JWHIA OPQEY PTLUM MPHCN BKWAZ
IQGCQ KNZNY MUGGO TCXND ELQYN KTVSR WKCQF ZFBWZ WJLLX IEGGA
FHZYA MRVBP QJNNV QAQQG PYJMM YYYAE WQBCQ GEOZY QLTOW YMQLH
ZWMGQ ZDLXF JJOME SGGSZ SBMTK NJJVY
```

Für dieses Kryptogramm zeigt Abbildung sechs die Verteilung der Koinzidenzen bei n Buchstaben Abstand. Die Spitze bei $n = 6$ kann ignoriert werden, da es bei $n = 12$ und $n = 24$ keine Maxima gibt; die Maxima bei $n = 16$, 32 und 48 lassen eine Periodenlänge von 16 als wahrscheinlich erscheinen. Es gibt zwar auch eine ziemlich hohe Spitze bei $n = 40$, was vielleicht doch auf eine Periode acht hindeuten könnte. Da allerdings weder $n = 8$ noch $n = 24$ zu sonderlich großen Häufigkeiten führen, spricht das Diagramm doch eher für $n = 16$; wir sollten also versuchen, das Kryptogramm unter dieser Annahme zu dechiffrieren. Der pq-Test schlägt folgende

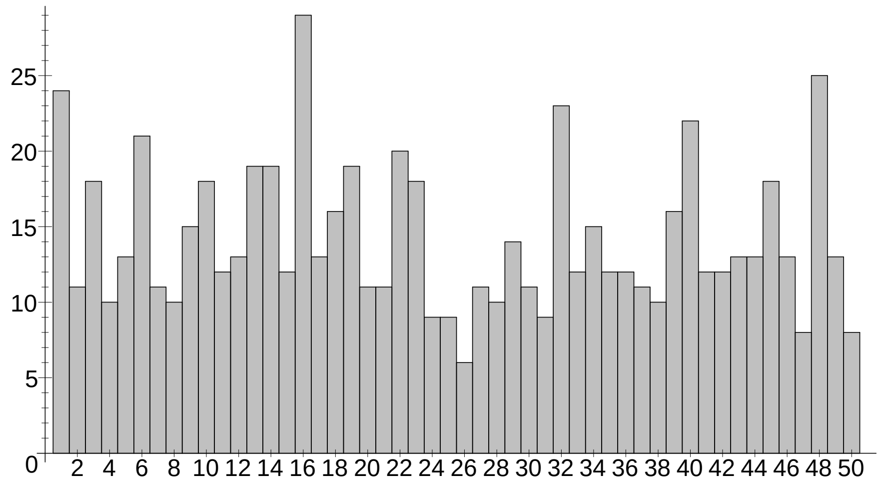


Abb. 6: Koinzidenzen bei kürzerer Schlüssellänge

Schlüsselbuchstaben vor:

N	0,066	M	0,052	Q	0,050	R	0,048	J	0,046
E	0,079	A	0,056	F	0,055	I	0,050	N	0,050
U	0,079	H	0,058	E	0,057	Q	0,056	D	0,051
E	0,064	D	0,055	P	0,049	F	0,048	T	0,048
R	0,083	V	0,059	E	0,050	I	0,046	Q	0,045
S	0,087	B	0,051	O	0,049	R	0,048	D	0,048
T	0,077	P	0,070	G	0,068	F	0,047	K	0,045
U	0,067	H	0,052	L	0,052	Y	0,052	V	0,049
D	0,075	Z	0,060	Q	0,056	U	0,049	H	0,046
I	0,072	Z	0,065	M	0,055	E	0,047	N	0,046
E	0,069	R	0,058	I	0,049	P	0,048	F	0,047
N	0,084	R	0,053	J	0,049	W	0,048	A	0,047
G	0,076	K	0,050	Z	0,049	X	0,047	Y	0,046
A	0,078	W	0,073	B	0,063	F	0,051	N	0,050
N	0,090	J	0,053	W	0,052	A	0,047	R	0,042
G	0,091	X	0,057	T	0,052	K	0,048	F	0,047

Hier steht gleich in der ersten Spalte der sehr wahrscheinlich aussehende Schlüssel

`NEUERSTUDIENGANG` .

der auch in der Tat zu verständlichem Klartext führt (entnommen aus einer Informationsbroschüre der Fakultät für Mathematik und Informatik für Schüler):

Der integrierte Studiengang Mathematik und Informatik bietet Ihnen eine berufsorientierte wissenschaftliche Ausbildung in den

> *Fächern Mathematik und Informatik. Erst nach zwei Jahren entscheiden Sie sich für eine der beiden Ausrichtungen „Mathematik“ und „Informatik“ und damit auch für einen der beiden Abschlüsse „Diplom-Mathematiker“ oder „Diplom-Informatiker“. Während der ersten beiden Studienjahre sind alle Veranstaltungen gemeinsam.*

Informationstheoretische Betrachtungen zeigen, daß schon bei weniger als zwei Buchstaben pro Alphabet der Klartext und der Schlüssel mit hoher Wahrscheinlichkeit eindeutig durch den Chiffretext bestimmt sind. Zu einer auf der Informationstheorie beruhenden Kryptanalyse benötigt man allerdings praktisch *unbeschränkte* Ressourcen, denn die Größen, mit denen hier gerechnet wird, sind definiert als Summen über den Raum *aller* möglicher Schlüssel; im Falle des obigen Schlüssels der Länge 29 wäre das eine Summe über $26^{29} \approx 10^{42}$ Summanden. Kryptanalytiker suchen daher meist nach weniger aufwendigen Verfahren.

Es gibt allerdings zumindest eine berühmte Ausnahme: Der *Data Encryption Standard* DES, der beispielsweise unserem EC-Kartensystem zugrundeliegt, wurde von der *Electronic Frontier Foundation* geknackt mit einer Maschine mit speziell hierfür entwickelten ASICs *(application specific integrated circuits),* die in der Tat alle $2^{56} \approx 7{,}2 \times 10^{16}$ Schlüssel durchprobiert und auf zwei 64-Bit-Blöcke anwendet. Diese werden in einem nötigenfalls mehrstufigen Verfahren daraufhin untersucht, ob sie von ihren statistischen Eigenschaften her mit englischem (oder sonstigem) Klartext übereinstimmen, bis ein entsprechender Block gefunden ist. Die Maschine kostete (einschließlich Entwicklungsarbeit) ungefähr 200 000 \$ und braucht durchschnittlich viereinhalb Tage zur Ermittlung eines Schlüssels; da sie modular aufgebaut ist und mit einem massiv parallelen Algorithmus arbeitet, könnte diese Zeit mit mehr Moduln um praktisch jeden Faktor reduziert werden, den man sich finanziell leisten möchte.

Einzelheiten zum DES Cracker Projekt findet man in

ELECTRONIC FRONTIER FOUNDATION: Cracking DES, *Secrets of Encryption Research, Wiretap Politics & Chip Design,* O'Reilly, 1998;

das Buch ist auch in elektronischer Form in Internet zu finden, beispielsweise unter `http://cryptome.org/cracking-des.htm`.

Auch bei VIGENÈRE-Chiffren muß man realistischerweise davon ausgehen, daß sie bereits anfangen unsicher zu werden, wenn der Klartext nur etwa 30% länger ist als der Schlüssel; wirklich sicher sind sie also praktisch nur in Form des *one time pad.* Wenn man nun bedenkt, daß es Programme gibt, die lange Dokumente oder gar eine gesamte Festplatte verschlüsseln in Abhängigkeit von einem nur wenige Zeichen langen Schlüssel mittels eines byte- statt buchstabenbasierten VIGENÈRE-Systems, wundert es nicht, daß so viele Programme auf dem Markt sind, die „vergessene“ Paßwörter rekonstruieren.

5. Substitutionschiffren

Die Entschlüsselung von CAESAR- und VIGENÈRE-Chiffren wird vor allem dadurch erleichtert, daß es pro Alphabet nur 26 mögliche Schlüssel gibt. Selbstverständlich gibt es keinen Grund, warum man sich auf zyklische Permutationen der Buchstaben des Alphabets beschränken muß; betrachtet man stattdessen beliebige Permutationen, gibt es

$$26! = 403291461126605635584000000$$

Möglichkeiten, also 403 Quadrillionen 291 Trilliarden 461 Trillionen 126 Billiarden 605 Billionen 635 Milliarden und 584 Millionen. Wie schon GIROLOMAO CARDANO (1501–1576, berühmt unter anderem durch die Lösungsformel für die kubische Gleichung, an der er völlig unschuldig ist) bemerkte, ist diese Zahl so groß, daß selbst viele Bücher nicht ausreichten, um alle diese Permutationen zu fassen.

Heutzutage benutzen wir Computer zur Entschlüsselung, und die können deutlich mehr Information verarbeiten als selbst „viele“ Bücher zu fassen vermögen. Ein Computer mit einer Taktfrequenz von 1 GHz führt pro Sekunde eine Milliarde Zykeln aus, pro Stunde also 3,6 Billionen, pro Tag 86,4 Billionen, pro Jahr 31,5 Billiarden. Das Alter unseres Universums wird auf etwa zehn Milliarden Jahre geschätzt; falls ein solcher Computer diese ganze Zeit hindurch gerechnet hätte, hätte er also mittlerweile etwa 300 Quadrillionen Zykeln durchlaufen. Selbst unter der (völlig unrealistischen) Annahme, daß der Computer pro Zykluszeit einen Schlüssel durchprobieren kann, hätte er bis heute noch nicht alle Möglichkeiten erschöpft.

Da Kryptanalytiker nur selten alle möglichen Schlüssel durchprobieren, ist diese Rechnung natürlich ohne jegliche Relevanz für die Sicherheit der Chiffre; in der Tat lassen sich schon relativ kurze Kryptogramme mit Chiffretext allein entschlüsseln.

Dazu reichen nun allerdings nicht mehr die Häufigkeitsverteilungen der einzelnen Buchstaben, denn dazu gibt es doch zu viele Buchstaben mit sehr ähnlicher Häufigkeit. Abbildung sieben zeigt die bereits aus Abbildung eins bekannte Häufigkeitsverteilung bei Anordnung der Buchstaben nach abnehmender Häufigkeit; wie man sieht, sind die Unterschiede zwischen

Abb. 7: Nochmals die Buchstabenhäufigkeiten in deutschem Text

benachbarten Buchstaben oft so minimal, daß eine sichere Identifikation allein aufgrund dieser Unterschiede praktisch ausgeschlossen ist.

Ein Buchstabe ist aber relativ gut charakterisiert durch seine Nachbarschaft: Auf ein „C“ folgt praktisch immer ein „H“, ein „D“ vor einem „E“ ist sehr viel häufiger als eines danach *usw.* Abbildung acht zeigt die in der Tat stark schwankenden Häufigkeiten von Buchstabenpaaren; die Achsen sind dabei der Einfachheit halber mit den Rangummern der Buchstabens im Alphabet beschriftet.

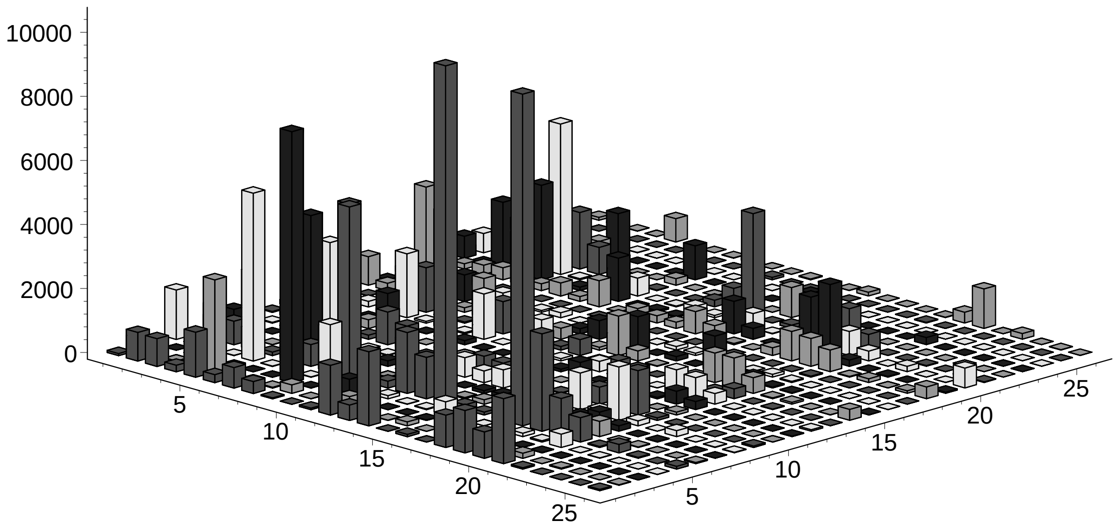


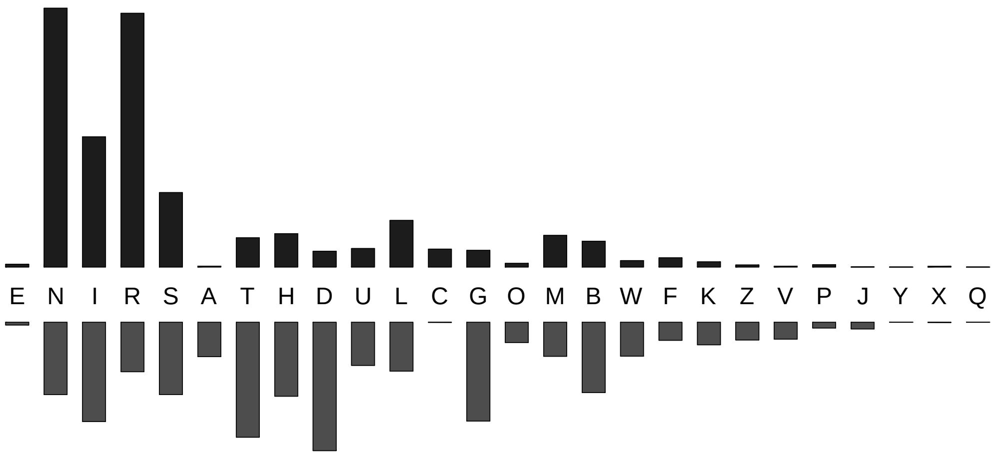
Abb. 8: Häufigkeiten von Buchstabenpaaren in deutschem Text

Das mag nicht besonders nützlich erscheinen, da schließlich die Nachbarn eines Buchstaben genausowenig bekannt sind wie der Buchstabe selbst.

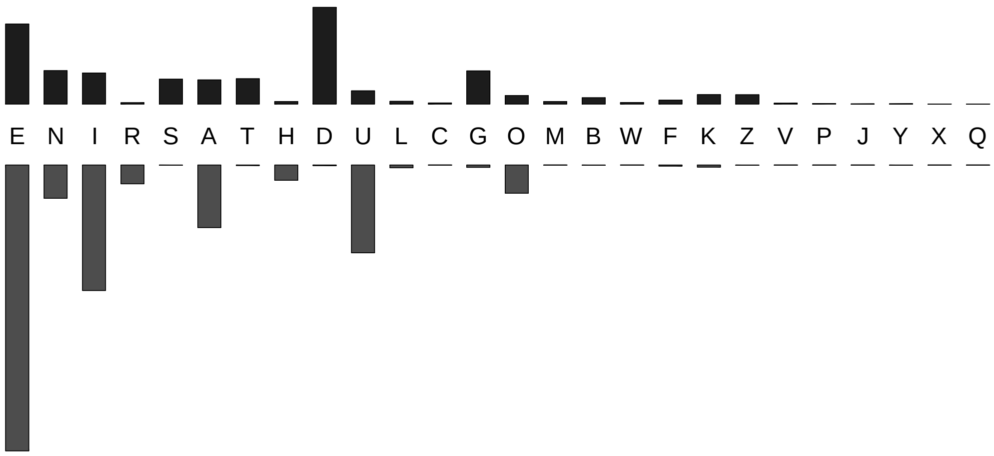
Bekannt ist aber die Rangordnung der Buchstaben in der Häufigkeitsverteilung des Kryptogramms, und diese sollte sich nur wenig von der der zugehörigen Klartextbuchstaben in einem Referenztext unterscheiden. Trägt man daher für jeden Buchstaben nach oben Balken auf, die angeben, wie oft der häufigste, zweithäufigste usw. Buchstabe *hinter* ihm steht und nach unten entsprechende Balken für die Buchstaben vor ihm, erhält man ein Doppelhistogramm, das sich zumindest für einigermaßen häufige Buchstaben nicht allzusehr unterscheiden sollte von dem des zugehörigen Klartextbuchstabens in einem Referenztext.

Für die praktische Anwendung dieser Idee brauchen wir zunächst die entsprechenden Doppelhistogramme für deutschen Klartext. Im folgenden sind für die 26 Buchstaben des Alphabets, nach Häufigkeit geordnet, die entsprechenden Kontaktdiagramme für *Dr. Katzenbergers Badereise* aufgezeichnet:

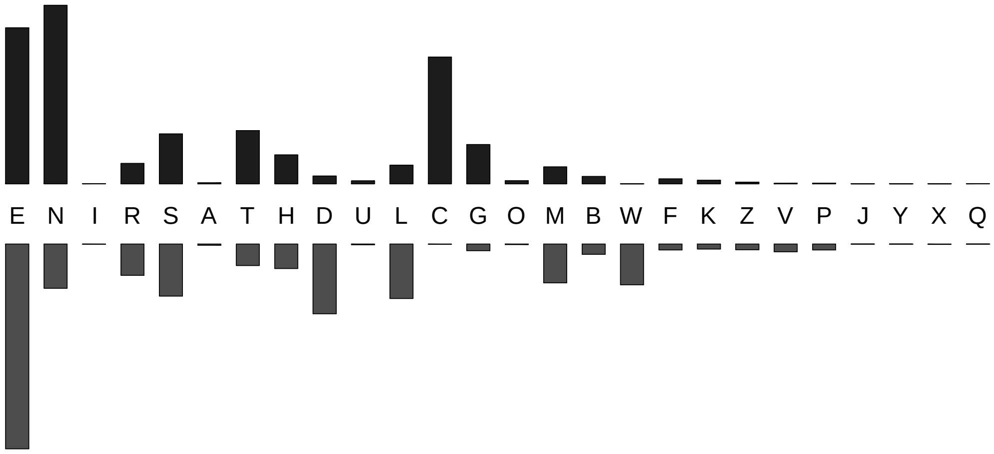
Kontaktdiagramm des Buchstabens E



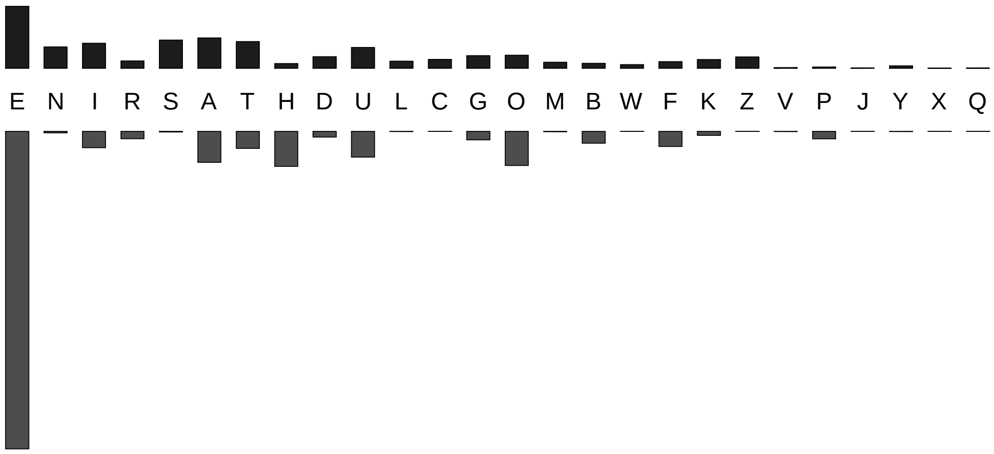
Kontaktdiagramm des Buchstabens N



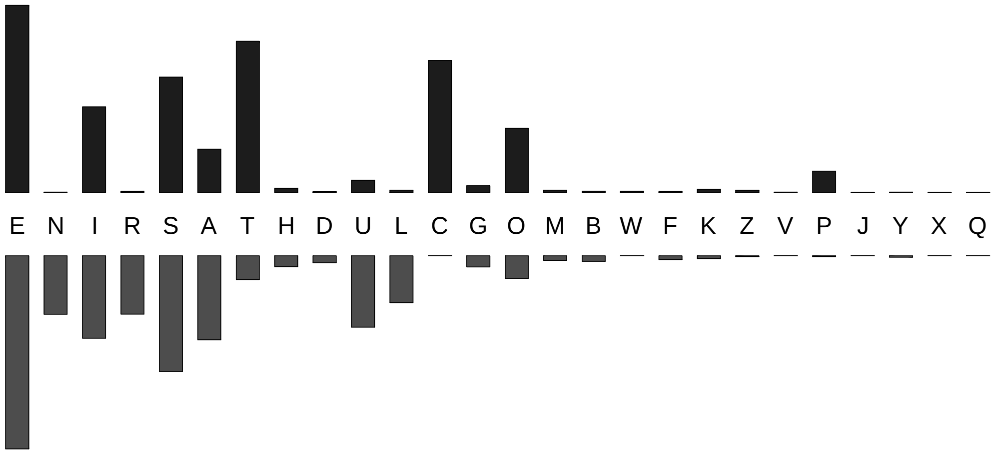
Kontaktdiagramm des Buchstabens I



Kontaktdiagramm des Buchstabens R

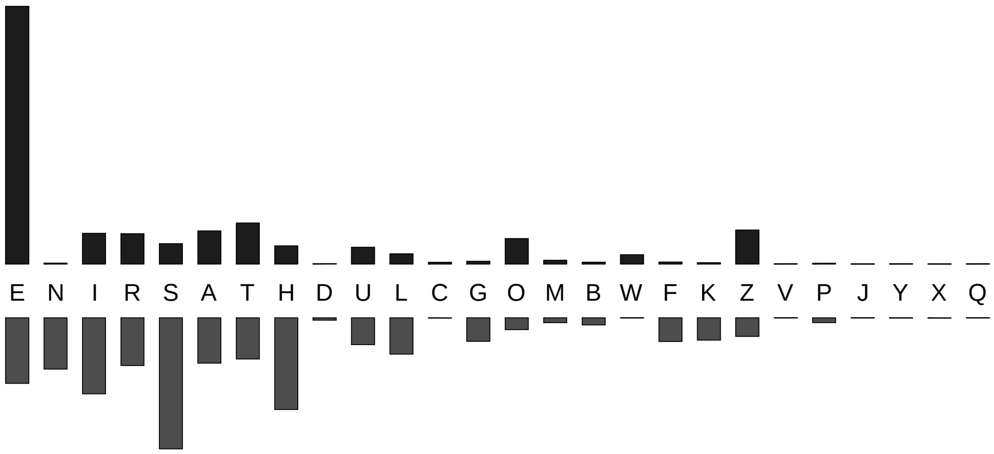


Kontaktdiagramm des Buchstabens S

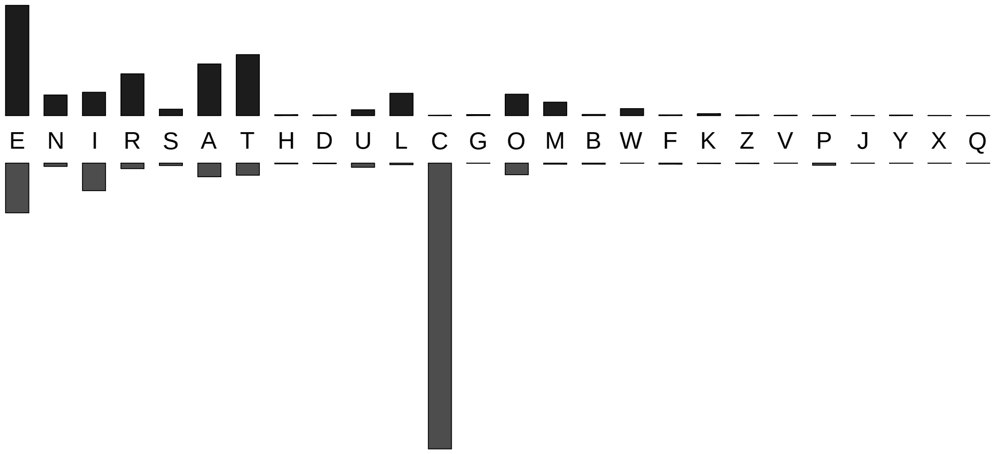


Kontaktdiagramm des Buchstabens A

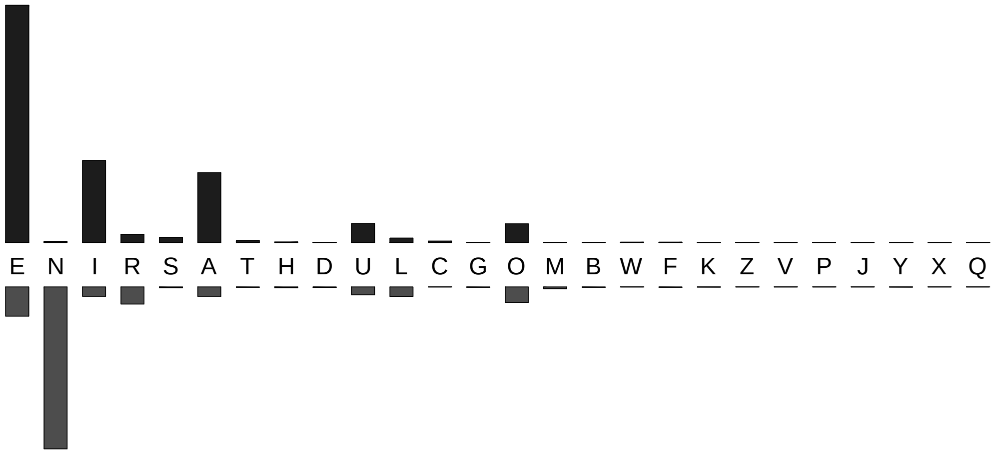
Kontaktdiagramm des Buchstabens T



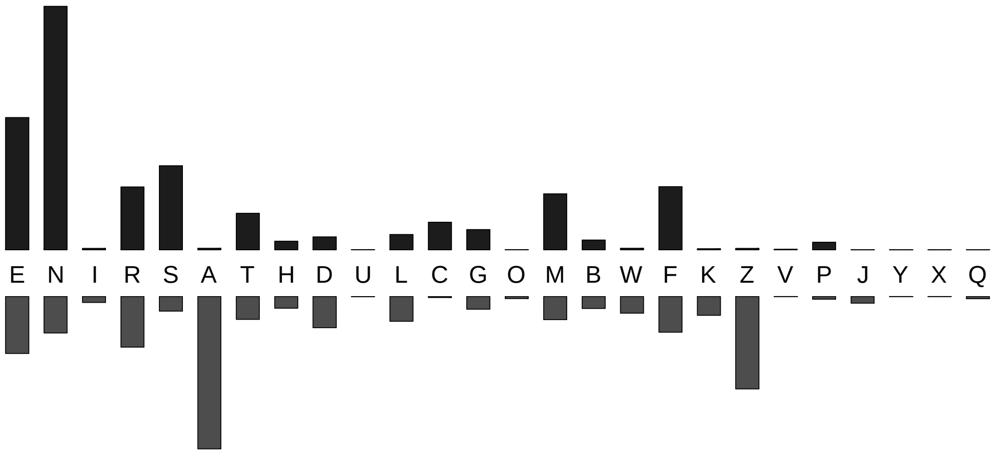
Kontaktdiagramm des Buchstabens H



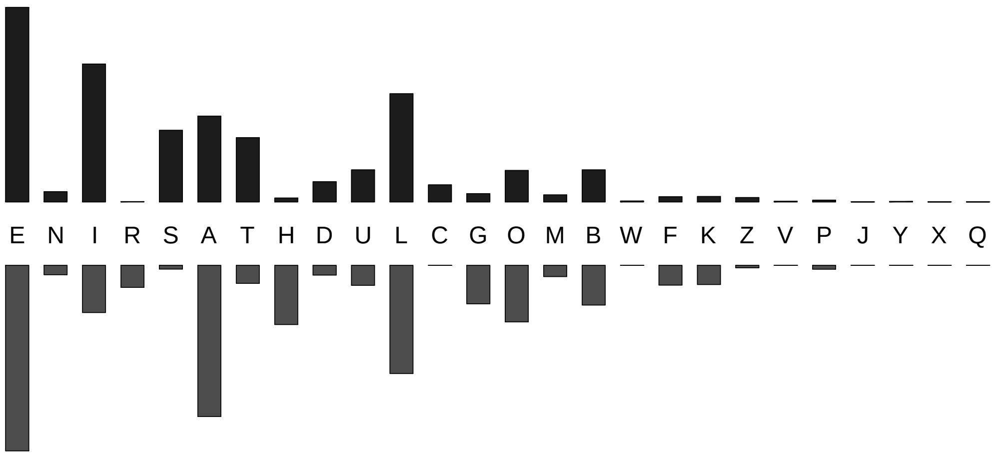
Kontaktdiagramm des Buchstabens D



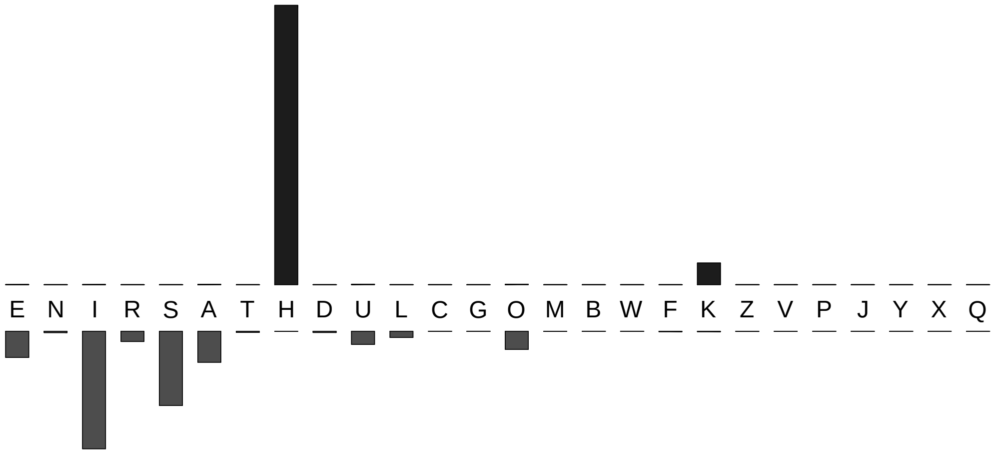
Kontaktdiagramm des Buchstabens U



Kontaktdiagramm des Buchstabens L

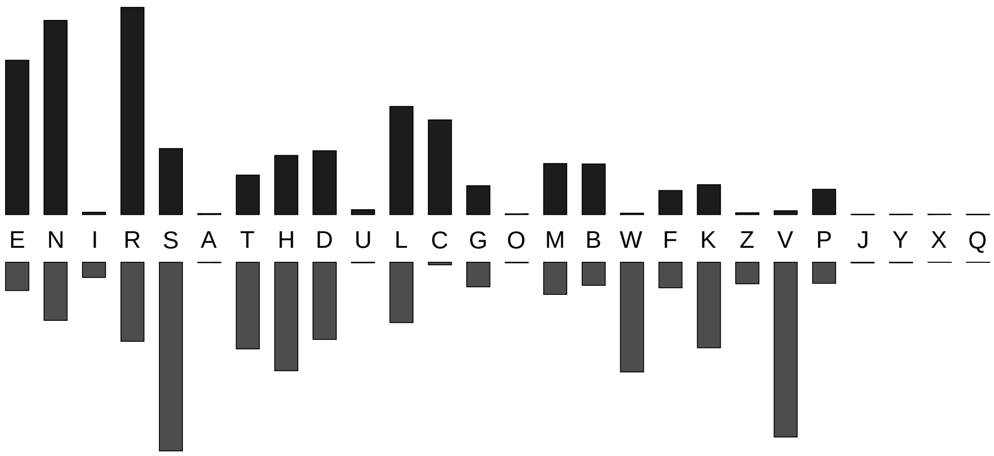


Kontaktdiagramm des Buchstabens C

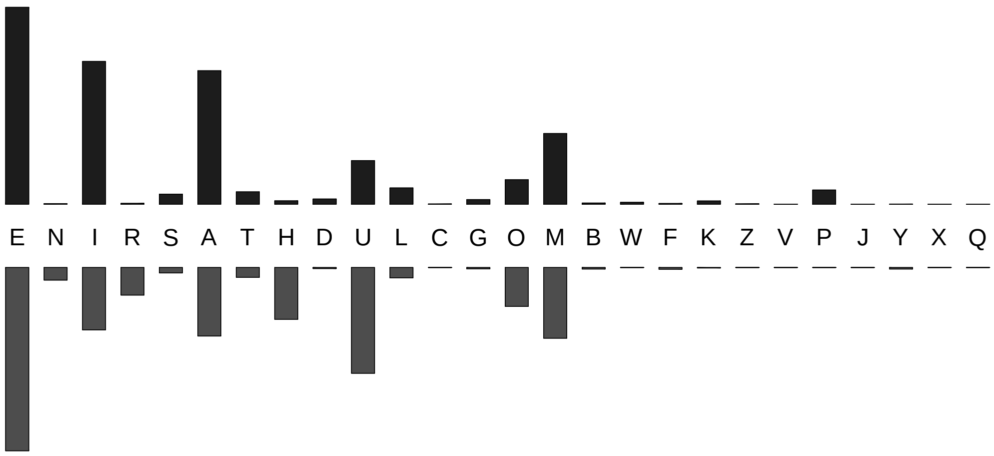


Kontaktdiagramm des Buchstabens G

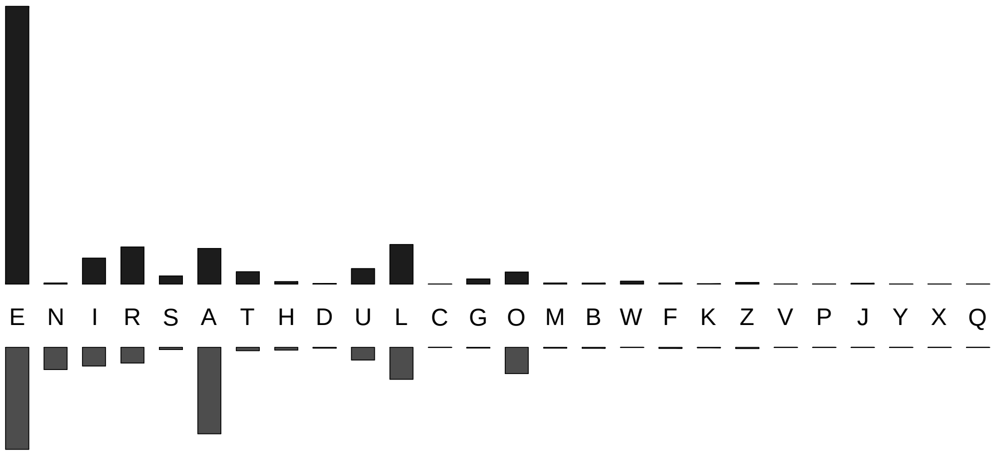
Kontaktdiagramm des Buchstabens O



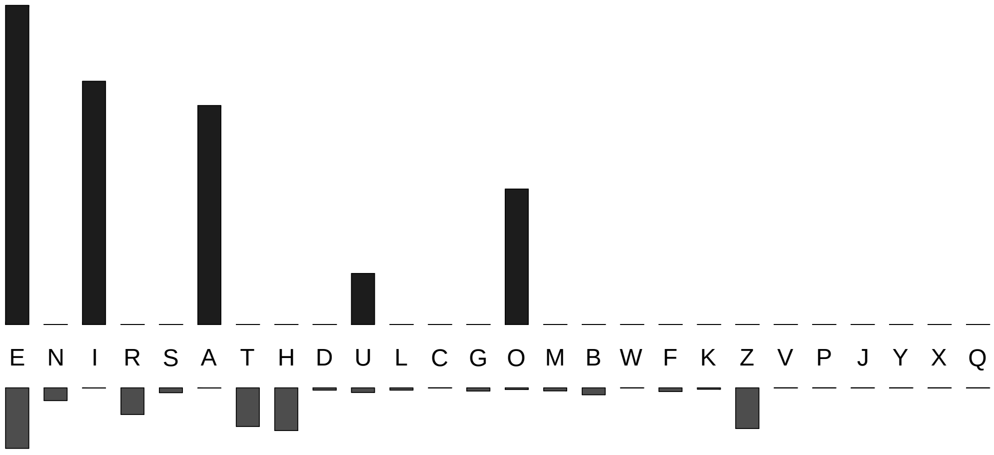
Kontaktdiagramm des Buchstabens M



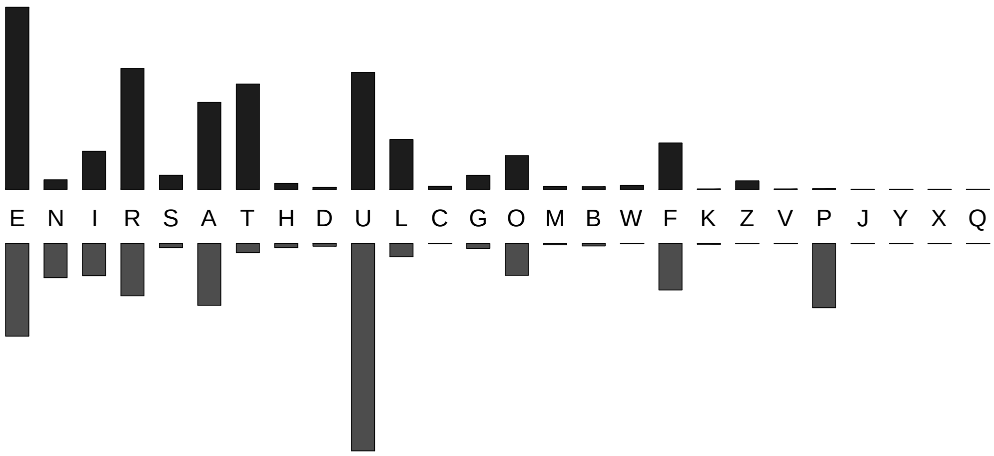
Kontaktdiagramm des Buchstabens B



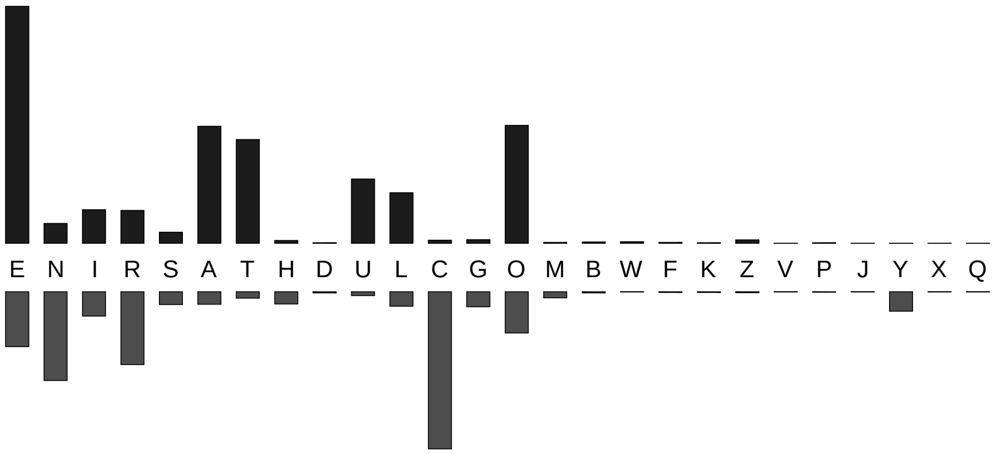
Kontaktdiagramm des Buchstabens W



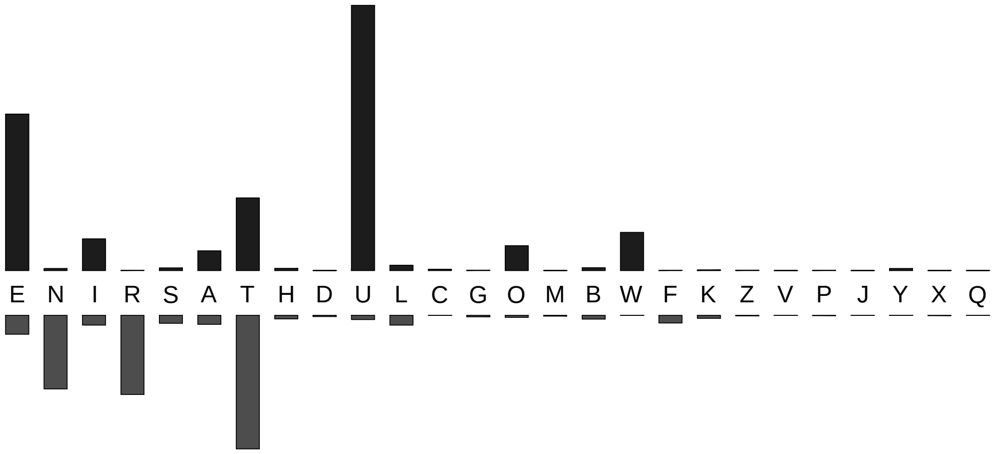
Kontaktdiagramm des Buchstabens F



Kontaktdiagramm des Buchstabens K

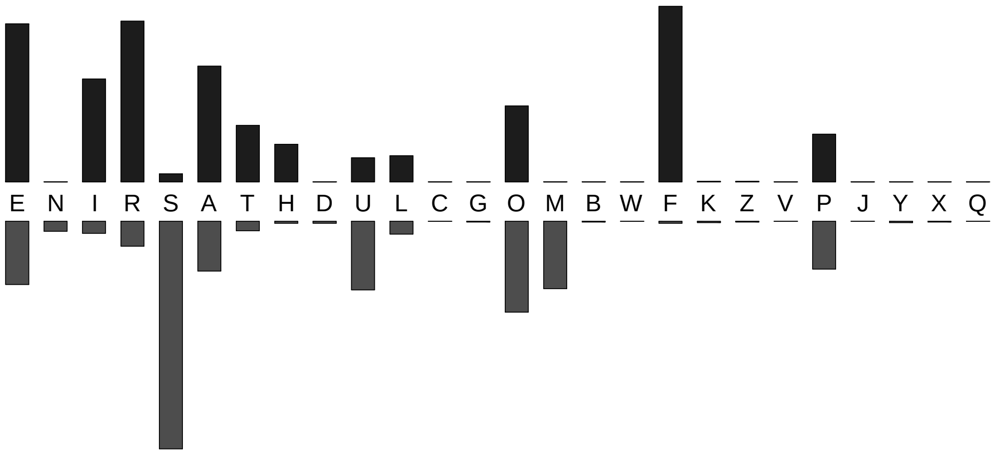


Kontaktdiagramm des Buchstabens Z

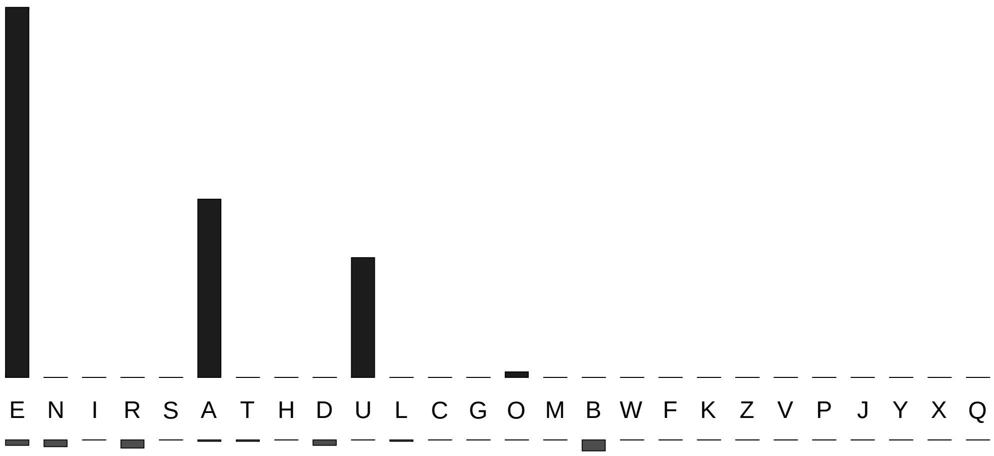


Kontaktdiagramm des Buchstabens V

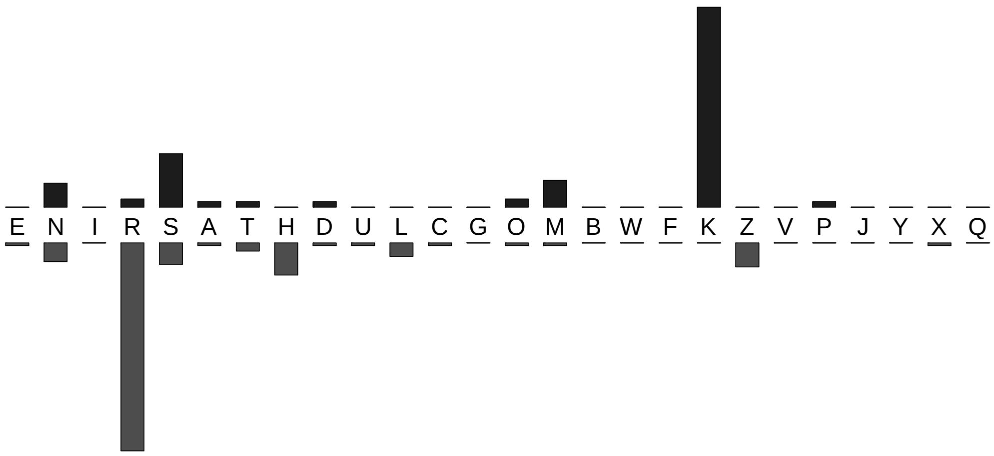
Kontaktdiagramm des Buchstabens P



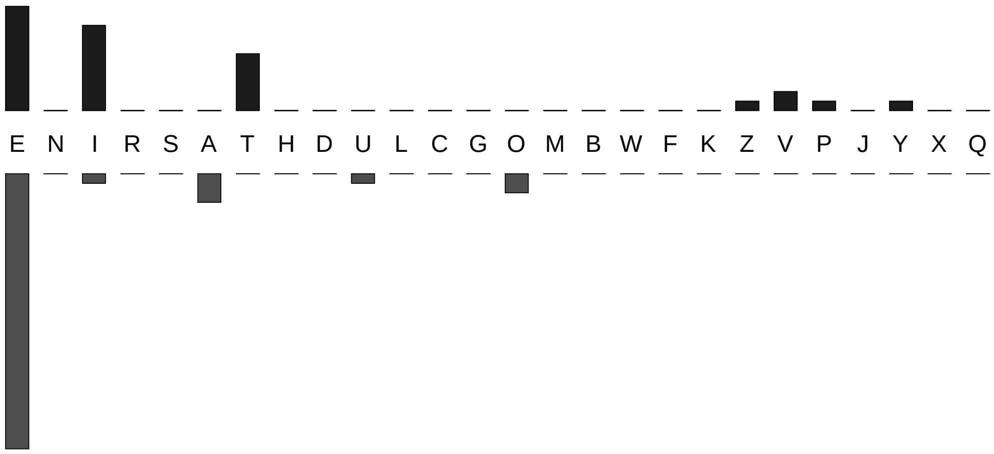
Kontaktdiagramm des Buchstabens J



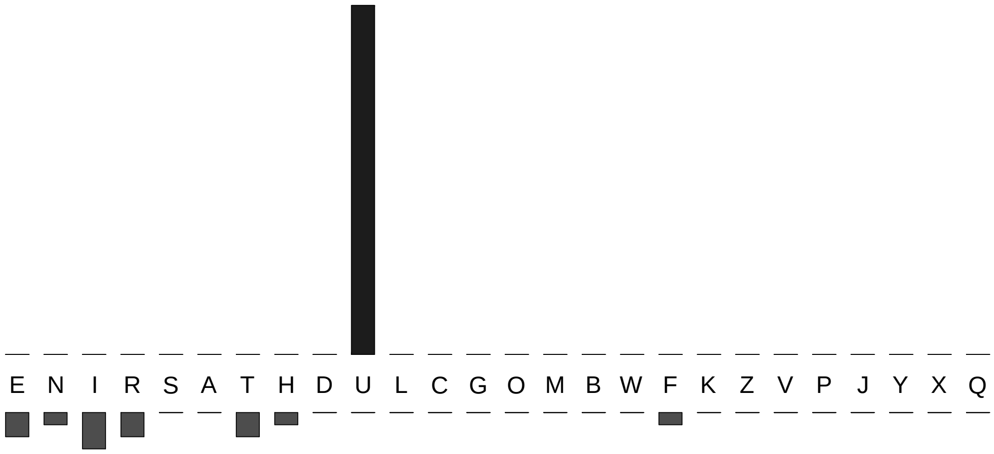
Kontaktdiagramm des Buchstabens Y



Kontaktdiagramm des Buchstabens X



Kontaktdiagramm des Buchstabens Q



Wie man sieht, verhalten sich die Buchstaben in der Tat sehr verschieden: „E“ tritt besonders gerne mit „N“ und “R“ auf, wobei beide eher hinter als vor dem „E“ stehen; „R“ hat in der überwiegenden Anzahl der Fälle ein „E“ vor sich, während die Buchstaben dahinter sehr viel gleichmäßiger verteilt sind; „S“ tritt vor und nach fast allen der häufigen Buchstaben auf, und auch „A“ hat unter diesen nur wenig ausgeprägte Präferenzen; „C“ und „H“ charakterisieren sich gegenseitig wegen der häufigen „CH“ und „SCH“ und so weiter.

Aufgrund seiner Erfahrung mit der deutschen Sprache wird wohl jeder leicht Erklärungen für die genannten Phänomene finden; etwas seltsam erscheint aber, daß beispielsweise „Y“ vorzugsweise auf „R“ folgt und ein „K“ hinter sich stehen hat. Dies ist in der Tat im Bereich der Zufallsschwankungen und nur mit Kenntnis des ausgezählten Textes verständlich: Der Zweck von Dr. Katzenbergers Badereise bestand darin, den Rezensenten seines Buchs zu verprügeln, einen Badearzt namens Strick, der sich St*ryk*ius nennt.

Für seltene Buchstaben sind obige Diagramme also mit Vorsicht zu genießen, wenn auch natürlich der allerseltenste Buchstabe „Q“ in der Tat durch das nachfolgende „U“ charakterisiert ist.

Für die häufigeren Buchstaben aber sind die Diagramme eine gute Charakterisierung; man muß nur bedenken, daß es bei einem kurzen Kryptogramm natürlich Schwankungen nicht nur in der relativen Höhe der Balken gibt, sondern auch bei der Rangfolge: Wie wir bereits bei den Caesar-Chiffren gesehen haben, ist „E“ nicht immer der häufigste Buchstabe.

Betrachten wir als Beispiel das Kryptogramm

```
KAPUY WOYEU NAWON QANSW HEUIE ONGQE BBHWQ OWOON
EEUIW HEGQH LOVFM EOFWY NGQUE BGALG QNFME OFWYV
EKAOI WHWOQ EUHMA TCEOT WNHLO VNAYE CQEUL HAHIE
OYWLH WQIHC EODLW UWHEO HWEIW HIACV NMWCT LOYUE
CYEGH PUSCA GEUIE VWAUE HECEW EIHEY NEQEO HAOKA
PUOLW YLANN MAOYE OYIEG QHKAO NEEUF LNEEU IWHWQ
IQABN HEEHL OVHCA NNKAI QWUBN ECFHL WNFLI ECFHN
HWBHL WYLAN NHCWO THIWU GQLOV NSCWG QHVEF LYCWO
NHLOV UEGQH IEOGQ IEUVE OOLOV MEOOY WOYLW YLANN
LWNFL CNGQU EBNHE HHKAO IIEOO MAHAV TCEOT MECLO
VELBI LUHWI AUAGQ SBWBB
```

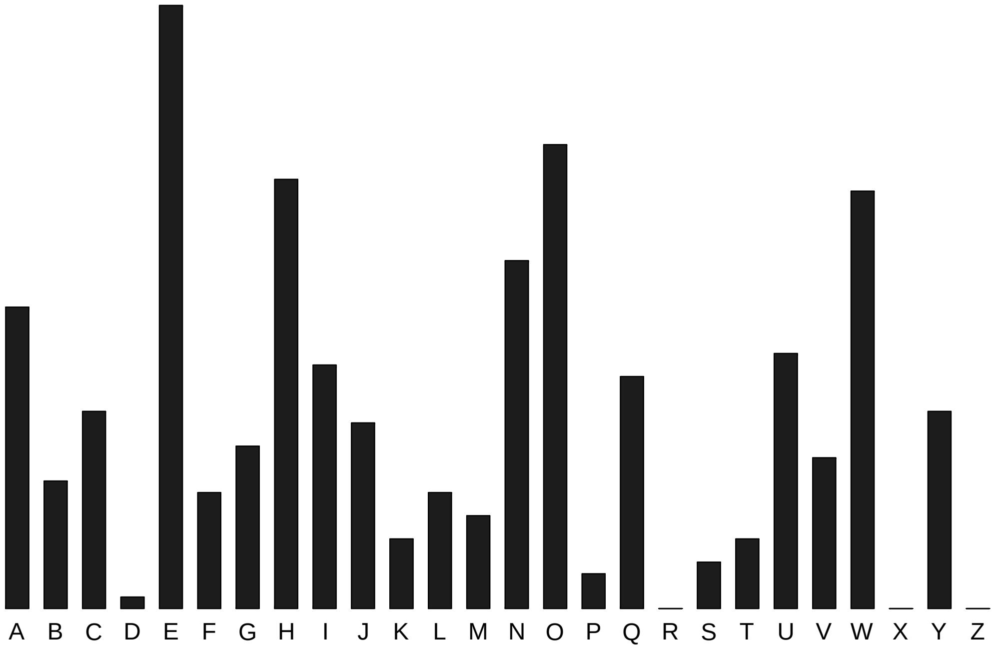


Abb. 9: Buchstabenhäufigkeiten im Kryptogramm

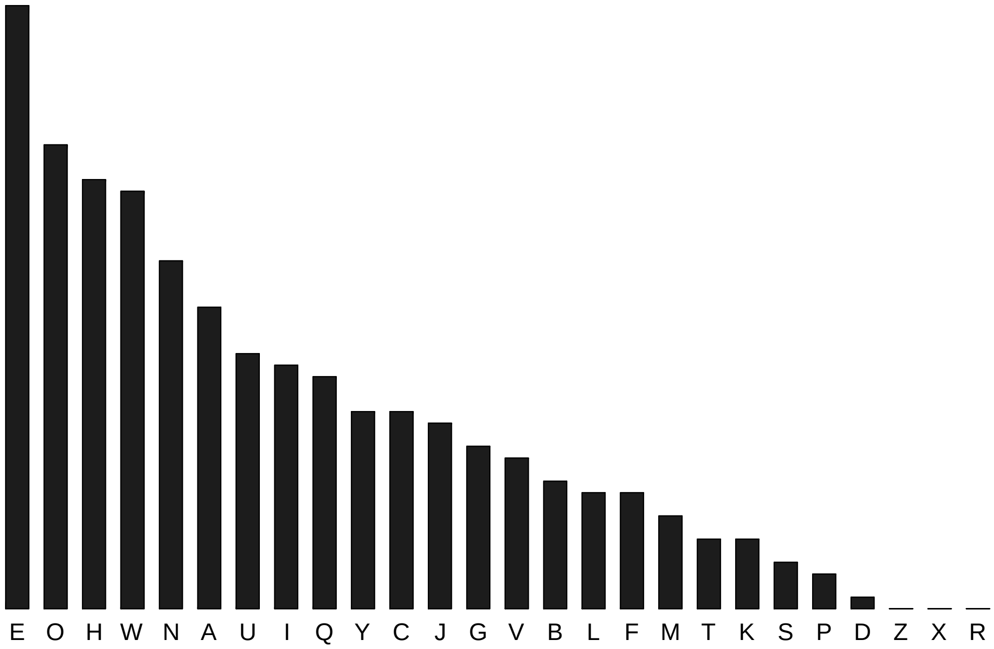
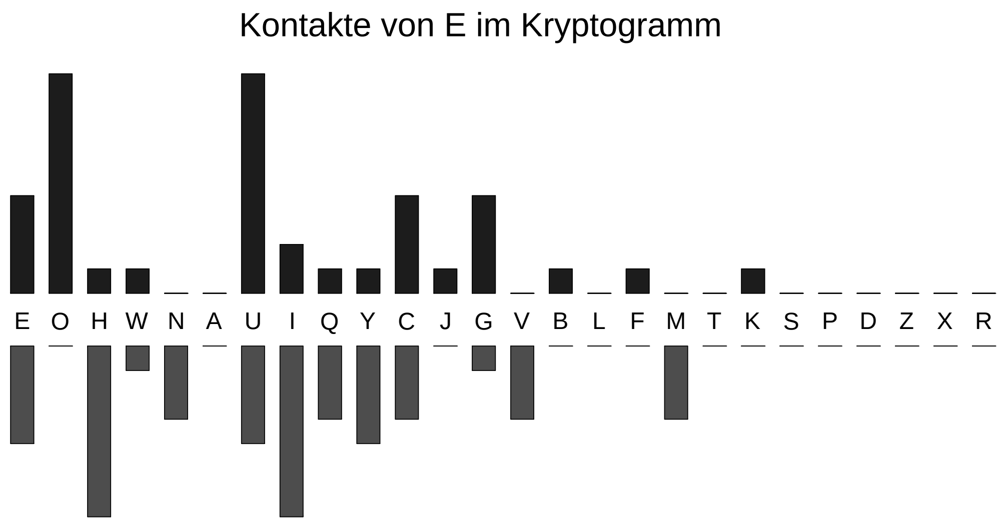


Abb. 10: Sortierte Buchstabenhäufigkeiten im Kryptogramm

Die Häufigkeitsverteilung der Buchstaben ist in Abbildung neun zu sehen; sie ist offensichtlich nicht einfach eine verschobene Häufigkeitsverteilung für Klartext, wie wir sie bei CAESAR-Substitutionen hatten.

Abbildung zehn zeigt aber, daß zumindest die sortierte Häufigkeitsverteilung doch recht gut der in Abbildung sieben dargestellten entspricht; man kann also davon ausgehen, daß es sich bei Kryptogramm um eine monoalphabetische Substitution handelt.

Der häufigste Buchstabe ist, wie wir das auch von deutschem Klartext gewohnt sind, das „E“; betrachten wir sein Kontaktdiagramm:



Schon ein flüchtiger Blick zeigt, daß sich dieses Diagramm in wesentlichen Punkten von dem des Klartext- „E“ unterscheidet, denn dieses kommt nur selten verdoppelt vor, und hat in erster Linie sehr häufige Buchstaben hinter sich stehen.

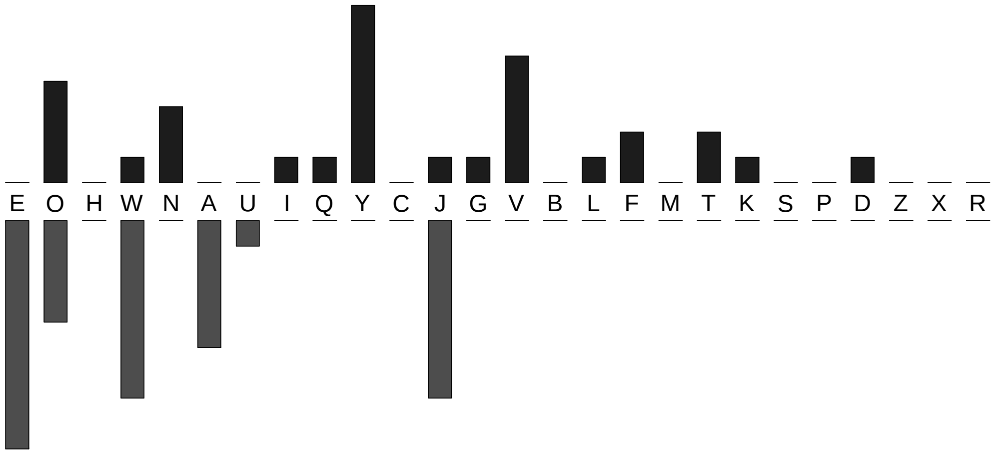
Auch um ein „N“ kann es sich nicht handeln, denn dazu gibt es zu viele Balken nach unten. „I“ ist unmöglich, da Doppel-I praktisch nicht vorkommt, „R“ mit seinem sehr charakteristischen Kontaktdiagramm ist es auch nicht, ebensowenig „S“, das sowohl vor als auch nach sich eher etwas häufigere Buchstaben bevorzugt.

Die Übereinstimmung mit „A“ ist ebenfalls alles andere als optimal, aber immerhin kommt „A“ etwas häufiger als die anderen Vokale verdoppelt vor und hat sehr häufig den zweithäufigsten Buchstaben sowie Buchstaben mittlerer Häufigkeit nach sich bei wenig ausgeprägten Präferenzen für die Vorgänger.

Die Diagramme der nächsten einigermaßen häufigen Buchstaben in deutschem Klartext passen wieder deutlich schlechter. Da der häufigste Buchstabe des Kryptogramms wohl schon unter den zehn häufigsten Buchstaben der deutschen Sprache zu finden sein sollte und mit relativ hoher Wahrscheinlichkeit ein Vokal sein dürfte, spricht alles in allem gesehen doch vieles für „A“. Zumindest falls der Chiffrebuchstabe „O“ für „N“ stehen sollte, paßt das Diagramm einigermaßen zum „A“.

Die (vorläufig) endgültige Entscheidung sollten wir also erst fällen, wenn wir auch das Kontaktdiagramm des zweithäufigsten Chiffrebuchstabens „O“ betrachtet haben.

Kontakte von O im Kryptogramm



Dieses Diagramm entspricht in der Tat recht gut dem des Klartextbuchstabens „N“; es spricht also vieles für die Entschlüsselung

$$E \to A, \quad O \to N .$$

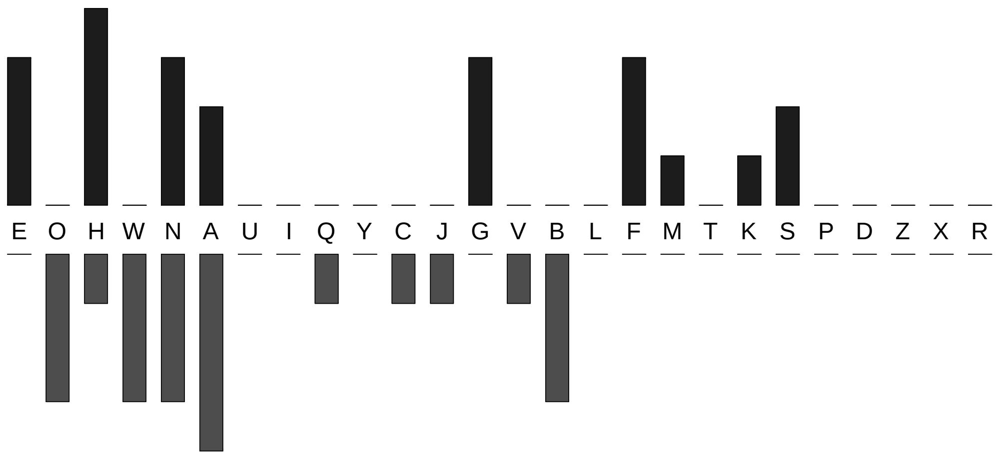
Kontakte von H im Kryptogramm

Der dritthäufigste Kryptogrammbuchstabe „H“ tritt sehr häufig verdoppelt auf, kann also kein „E“, „I“ oder „R“ sein. „S“ und „T“ sind Möglichkeiten, wobei „S“ auf den ersten Blick optisch günstiger aussieht. Bedenkt man aber, daß der erste Balken bei den einen Diagrammen für „E“ und bei den anderen für „A“ stehen, spricht doch vieles eher für „T“: „S“ steht häufiger nach „A“ als davor, bei „T“ ist es eher umgekehrt; auch die relativ scharf ausgeprägten oberen Balken beim „S“ passen nicht richtig. Gegen „T“ spricht vor allem der auffällige „E“-Balken, aber da wir bereits beim dritthäufigsten Buchstaben des Kryptogramms sind und noch immer kein „E“ gefunden haben, ist dieses hier wohl doch etwas unterrepräsentiert.

Kontakte von W im Kryptogramm

Das „W“ im Kryptogramm hat gerne ein Klartext-„N“ nach sich und ein Klartext-„T“ sowohl vor als auch nach sich, tritt praktisch nie verdoppelt auf und hat einen Buchstaben mittlerer Häufigkeit, der oft danach steht; all dies spricht sehr für ein „I“.

Kontakte von N im Kryptogramm

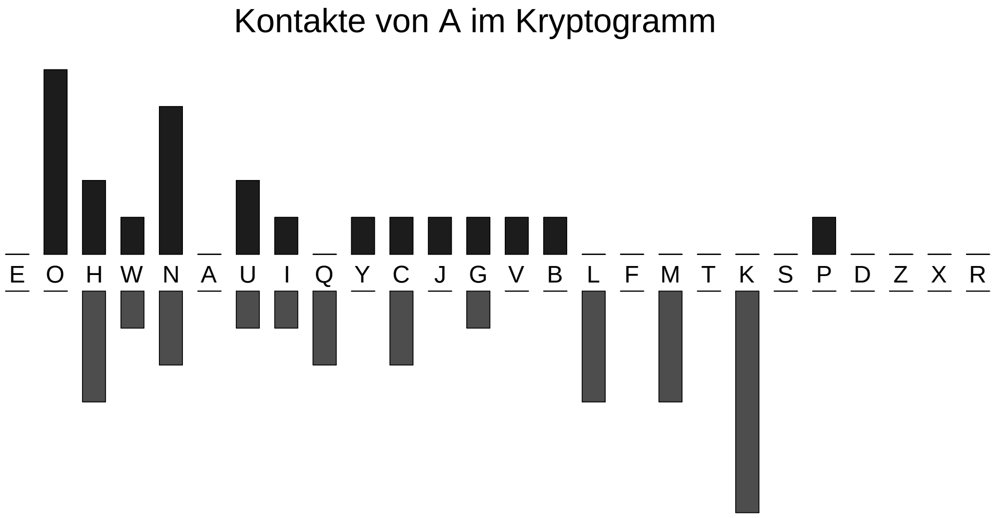


Der Klartextbuchstabe zu „N“ wird oft von „A“, „T“ und sich selbst gefolgt und steht oft nach „N“; damit dürfte fast klar sein, daß es sich um ein „S“ handeln muß.

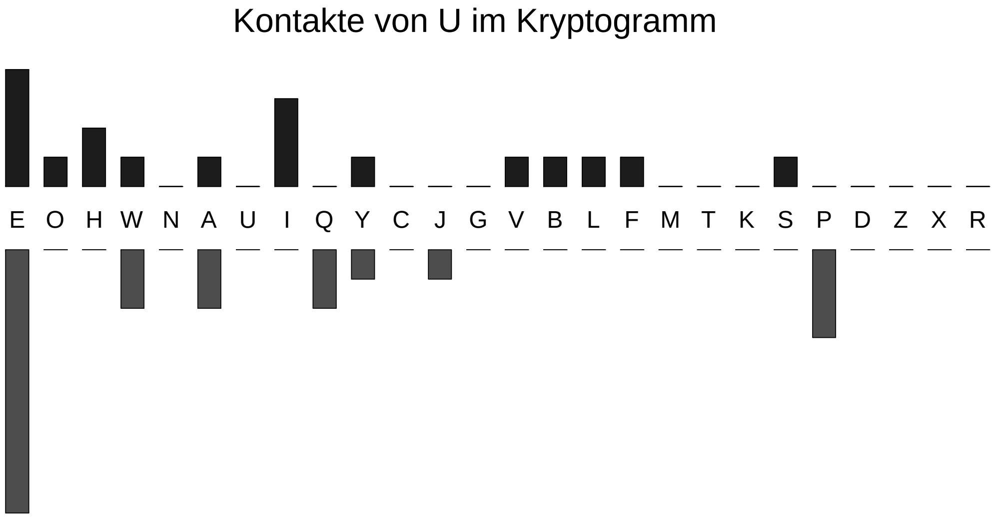
Damit haben wir mit

$$E \rightarrow A, \quad O \rightarrow N, \quad H \rightarrow T, \quad undN \rightarrow S$$

die fünf häufigsten Buchstaben des Kryptogramms entschlüsselt und können zumindest für die mit den jeweils korrekten Balken der Klartext-diagramme vergleichen.



Das Diagramm zum Chiffretext-„A“ gibt uns gleich zwei Identifizierungen: Ein Buchstabe, der weder mit sich selbst noch mit dem Klartext-„A“ zusammen auftritt, dafür aber häufig ein „N“ oder „S“ nach sich führt, „T“ zu beiden Seiten und einen eher seltenen Buchstaben oft vor sich, kann wohl nur „O“ sein, und das häufig davorstehende „K“ ist somit ein „V“.



„U“ wird entschlüsselt durch einen Buchstaben, der die bereits bekannten Vokale gerne sowohl vor als auch hinter sich hat, sich selbst allerdings praktisch nie, die Konsonanten „N“, „S“ und „T“ stehen häufiger dahinter, aber praktisch nie davor, und eine ganze Reihe eher seltener Buchstaben steht immer wieder einmal dahinter – hier fällt eine klare Identifikation schwer.

Wir können aber schauen, ob wir mit den bereits gefundenen acht Klartextbuchstaben schon Wörter im Kryptogramm erkennen können:

```
KAPUY WOYEU NAWON QANSW HEUIE ONGQE BBHWQ OWOON
VO    IN A  SOINS  OS I TA  A NS  A   TI  NINNS
EEUIW HEGQH JOVFM EOFWY NGQUE BGAJG QNFME OFWYV
AA  I TA  T  N    AN I  S   A   O    S  A N I
EKAOI WHWOQ EUHMA TCEOT WNHJO VNAYE CQEUL HAHIE
AVON  ITIN  A T O   AN  IST N  SO A   A   TOT A
OYWLH WQIHC EODJW UWHEO HWEIW HIACV NMWCT JOYUE
N I T I  T  AN  I  ITAN TIA I T O   S I    N  A
CYEGH PUSCA GEUIE VWAUE HECEW EIHEY NEQEO HAOKA
A T     O  A  A  IO A TA AI A TA  SA AN TONVO
PUOLW YLANN MAOYE OYIEG QHKAO NEEUF JNEEU IWHWQ
N I   OSS  ON A N  A   TVON SAA    SAA   ITI
IQABN HEEHJ OVHCA NNKAI QWUBN ECFHL WNFJI ECFHN
O S TAAT  N T O SSVO   I  S A  T  IS    A  TS
HWBHL WYLAN NHCWO THIWU GQJOV NSCWG QHVEF JYCWO
TI T  I  OS ST IN  T I     N  S  I   T A     IN
NHJOV UEGQH IEOGQ IEUVE OOJOV MEOOY WOYLW YLANN
ST N   A  T  AN    A  A NN N   ANN  IN  I   OSS
LWNFJ CNGQU EBNHE HHKAO IIEOO MAHAV TCEOT MECJO
IS    S    A STA TTVON   ANN  OTO    AN   A  N
VEJBI JUHWI AUAGQ SBWBB
A      TI  O O     I
```

Gleich in der ersten Zeile springt „a-so ins“ ins Auge, also steht wohl „U“ für „L“. Damit wir das nächste Wort zu „-os-ital“, d.h. „Q“ steht für „H“ und „S“ für „P“.

Nachdem wir zusätzlich zum schon länger bekannten „S“ auch das „H“ kennen, können wir nach „SCH“ suchen; ein Kandidat dafür ist die fünfte Gruppe der zweiten Zeile, was „G“ als Verschlüsselung von „C“ ergibt.

Damit wird der Text schon stellenweise verständlich: Das zweite Wort muß wohl „ging“ sein, vor dem „avon“ zu Beginn der dritten Zeile kann wohl nur ein „D“ stehen und so weiter. Der Rest ist ein reines Puzzlespiel und führt zum Klartext

> *Voyl ging also ins Hospital. Man schafft ihn in'n Saal mit achtundzwanzig Schlafcouchs, ...zwanzig davon mit Inhalt, wo krank ist und sogar halbtot. Man gibt ihm Tranquilitantia mit Mordswirkung (Largactyl, Procalmadiol, Atarax). Am Tag sah Anton Voyl 'n Big Boss, wo'n Gang macht von Saal zu Saal; mit ihm Hofstaat und Tross vom Hilfsarzt bis zum Arztstift; Big Boss trinkt Milch und spricht dazu, grinst und lacht manchmal ... Dann und wann ging Big Boss bis zur Schlafstatt von'm Mann, wo todkrank war und aufm ultimo Loch pfiff ...*

Der Text stammt aus der deutschen Übersetzung des Romans *La disparaition* von GEORGE PEREC *(Edition Denoël, Paris,* 1969*).* Dieser französische Autor zahlreicher Romane, hatte nach einem feuchtfröhlichen Abend gewettet, daß er einen Roman schreiben könne, ohne den Buchstaben „E" zu verwenden; er gewann. Die deutsche Übersetzung von EUGEN HELMLÉ mit dem Titel *Anton Voyls Fortgang (Zweitausendeins,* ^{3}1998*)* hält sich auch an die Vorgabe, verwendet aber Umlaute – daher die Auslassungspunkte im obigen Text.

Wie man sieht, lassen sich die einzelnen Buchstaben also selbst dann noch recht gut anhand ihrer Umgebung identifizieren, wenn ihre Rangfolge nicht mehr ganz der für deutsche Texte üblichen entspricht.

Eine hier nicht behandelte Möglichkeit, sich Buchstaben zu verschaffen, liegt auch im Erraten von Wörtern aus der Botschaft. Da man beim Abhören meist weiß, wonach man sucht, gibt es oft Wörter, die mit hoher Wahrscheinlichkeit im Text vorkommen sollten. Falls ein solches Wort einen oder gar mehrere Buchstaben mehrfach enthält, kann man im Chiffretext nach demselben Muster suchen. Ein ideales Wort wäre beispielsweise *Substitution:* Es muß von jeder monoalphabetischen Substitution in eine Folge der Art

$$\mathsf{XY}\mathrm{p}\mathsf{XZWZYZW}\mathrm{qr}$$

transformiert werden, und eine Folge dieser Art kann praktisch nur für das Wort *Substitution* stehen. Damit sind sieben Buchstaben identifiziert.

6. Polyalphabetische Substitutionen

Als nächste Stufe der Komplexität bietet sich natürlich an, aufeinanderfolgende Buchstaben mit verschiedenen Permutationen zu verschlüsseln, d.h. also eine Art von VIGENÈRE-Verfahren mit allgemeinen monoalphabetischen Substitutionen anstelle der CAESAR-Substitutionen.

Leider steigt dadurch, sofern die Anzahl der Permutationen nicht in die Größenordnung der Nachrichtenlänge kommt, die Sicherheit nur unwesentlich an, denn die Verfahren aus dem vorigen Paragraphen funktionieren auch für diesen Fall: Man stellt zunächst, genau wie bei den VIGENÈRE-Chiffren, die Periode fest; danach weiß man, welche Buchstaben mit derselben Permutation verschlüsselt wurden. Jede dieser Buchstabenmengen wird nach Häufigkeit geordnet, und man kann wieder Kontaktdiagramme aufstellen. Dabei muß nur beachtet werden, daß die rechten und linken Nachbarn nicht aus derselben Gruppe kommen, sondern aus je einer anderen festen Gruppe, so daß Balken nach oben und nach unten nicht mehr für denselben Buchstaben stehen müssen. Abgesehen von dieser kleinen Erschwernis ändert sich nichts, und ein erfahrener Kryptanalytiker kann ab etwa 30–40 Buchstaben pro Alphabet praktisch sicher sein, daß er die Nachricht entschlüsseln kann. Wie die Informationstheorie lehrt, legen bereits deutlich weniger Buchstaben den Klartext fest, allerdings lassen sich deren Methoden meist nicht in effiziente Analyseverfahren übersetzen.

Eine Ausnahme bildet der bereits erwähnte DES-Cracker: Er probiert alle 2^{56} Entschlüsselungsmöglichkeiten des Data Encryption Standard DES durch und verwirft die meisten allein schon deswegen, weil das erste Byte ein nichtdruckbares Zeichen ist. Die meisten verbliebenen Möglichkeiten können aufgrund des zweiten Bytes eliminiert werden, und nur für den geringen verbleibenden Rest macht ein PC Häufigkeitsanalysen.

Die Möglichkeit einer solchen Maschine beruht aber ausschließlich auf der sehr geringen Schlüssellänge des DES; sein am 2. Oktober 2000 gekürter Nachfolger AES (Advanced Encryption Standard) arbeitet wahlweise mit drei Schlüssellängen, von denen die kürzeste 128 ist. Hierfür ist bei heutigem Stand der Technik keine entsprechende Maschine denkbar; schließlich steigt der Aufwand eines solchen Verfahrens schon bei Schlüssellänge 128 um den Faktor 2^{72} oder rund $5 \cdot 10^{21}$.

Zurück in die Vergangenheit!

Polyalphabetische Chiffren mit sehr großen Periodenlängen erzeugten die sogenannten *Rotormaschinen,* die in den Zwanzigerjahren aufkamen und die Kryptographie des zweiten Weltkriegs dominierten. Ein Rotor ist eine Scheibe, die auf jeder Seite in gleichmäßigen Abständen 26 Kontakte hat. Jeder Kontakt auf der Vorderseite ist mit einem der Kontakte auf der Rückseite verbunden, wobei nur darauf zu achten ist, daß diese Verbindungen spiegelsymmetrisch angebracht werden, so daß die so realisierte Permutation ein Produkt von dreizehn Transpositionen ist.

Zum Einsatz der Maschine wählt man, je nach Maschine, drei bis fünf von mehreren verfügbaren Rotoren aus und setzt diese in einer festzulegenden Reihenfolge und Anfangsstellung in die Maschine ein. Der erste Buchstabe wird dann von einem festen Kontakt auf der einen Seite der Maschine durch die Rotoren in dieser Stellung an einen der 26 festen Kontakte auf der anderen Seite der Maschine übertragen; das Chiffrierergebnis ist der diesem Kontakt zugeordnete Buchstabe. Danach bewegen sich die Rotoren automatisch in einer nicht ganz regelmäßigen Weise weiter, so daß für den nächsten Buchstaben eine andere Permutation realisiert wird und so weiter. Zur Entschlüsselung wird der Chiffretext einfach in eine identisch aufgebaute Maschine mit gleicher Anfangsstellung aber umgekehrter Rotorreihenfolge gegeben.

(Unter http://library.thinkquest.org/28005/flashed/timemachine/courseofhistory/enigmaapp.shtml kann man ausprobieren, wie das bei er deutschen Enigma-Maschine funktionierte.)

Die mit solchen Maschinen erzielbaren Perioden liegen im Bereich über 100 000; trotzdem wurden sie schon im zweiten Weltkrieg häufig geknackt.

Ein Grund dafür ist, daß während der Gültigkeitsperiode eines Schlüssels alle Nachrichten mit derselben Anfangsstellung verschlüsselt werden müssen, so daß damit zumindest für die ersten Buchstaben relativ viele Chiffrebuchstaben zusammenkommen. Das weitere Problem, daraus die Anfangsstellung der Maschine zu ermitteln, lösten die Engländer im zweiten Weltkrieg mit speziell gebauten Maschinen, den sogenannten „Bomben“, die mechanisch alle möglichen Anfangsstellungen durchprobierten. Die Verdrahtung der einzelnen Rotoren war schon vor dem Krieg von den Polen geknackt worden.

Seit etwa 1970 werden Rotormaschinen nicht mehr eingesetzt – außer in UNIX. Das dortige `crypt(1)`-Kommando simuliert eine Rotormaschine mit *einem* Rotor, der 256 Ein- und Ausgänge hat. Sehr sicher ist das nicht, und auch UNIX selbst verwendet beispielsweise für Paßwörter das auf einem „gesalzenen“ DES beruhende deutlich bessere `crypt(3)`-Unterprogramm.

7. Literaturhinweise

Das (dickleibige) Standardwerk zur alten Kryptographie mit Schwerpunkt auf der geschichtlichen Darstellung ist

DAVID KAHN: *The Codebreakers – the comprehensive history of secret communication from ancient time to the internet,*Scribner, New York, ^{2}1996

Deutlich kürzer und billiger, mit Schwerpunkt auf der Darstellung der Verfahren, ist

HELEN FOUCHÉ GAINES: *Cryptanalysis – a study of ciphers and their solution,* Dover, New York, 1956 *(Nachdruck einer Originalausgabe von 1939)*

Rotormaschinen und ihrer Kryptanalyse gewidmet ist das Buch

CIPHER A. DEAVOURS, LOUIS KRUH: *Machine cryptography and modern cryptanalysis,* Artech House, Dedham MA, 1985

Das Wort „modern“ bezeichnet hier den Zeitraum von etwa 1920–1970.

Ein wirklich modernes Lehrbuch mit Kapitel über klassische Kryptographie sowie über statistische und informationstheoretische Ansätze zur Kryptanalyse ist

JAN C.A. VAN DER LUBBE: *Basic Methods of cryptography,* Cambridge University Press, 1998

Eine sehr ausführliche Darstellung der statistischen und informationstheoretischen Ansätze findet man in

ALAN G. KONHEIM: *Cryptography: a primer,* Wiley, New York, 1981

II. Quantenkryptographie

Von allen Kryptosystemen, die wir bislang kennengelernt haben, hat nur der *one time pad* beweisbare Sicherheit. Das liegt natürlich teilweise einfach daran, daß im letzten Vortrag nur wenige und zudem sehr alte Systeme vorgestellt wurden. Wenn man allerdings Sicherheit im Sinne von *absoluter* Sicherheit betrachtet, wenn man also fordert, daß ein Gegner aus dem Kryptogramm *keinerlei* Information ziehen kann außer der Tatsache, daß eine Nachricht einer bestimmten Länge versandt wurde, zeigt ein Satz von SHANNON (s. etwa [**W**], §7.3), daß ein in diesem Sinne sicheres Kryptosystem notwendigerweise mit einem Schlüssel arbeiten muß, der *mindestens* so lang ist wie die zu verschlüsselnde Nachricht; er kann also nicht kürzer sein als beim *one time pad.*

Perfekte Sicherheit von Kryptosystemen erfordert somit einen sehr hohen Aufwand an Schlüsselinformation; hinzu kommt, daß der Schlüsselaustausch ohnehin ein großes Problem aller symmetrischer Kryptoverfahren ist: Jede Schlüsselvereinbarung erfordert entweder ein persönliches Treffen der Beteiligten oder einen vertrauenswürdigen Boten oder aber ein anderes Kryptosystem, von dessen Sicherheit dann *jede* künftige Kommunikation abhängt.

Die Quantenkryptographie gestattet es zwei räumlich getrennten Partnern, einen Schlüssel (beispielsweise für den *one time pad*) so zu vereinbaren, daß ein Gegner mit hoher Wahrscheinlichkeit kein einziges Bit an Information erhält.

1. Informationsübertragung mit einzelnen Photonen

Die Grundidee des Verfahrens besteht darin, die Bits durch einzelne Photonen zu kodieren. Genau wie einen ganzen Lichtstrahl kann man auch ein einzelnen Photon *polarisieren,* d.h. man kann seine Schwingungsebene festlegen und beispielsweise vereinbaren, daß eine horizontale Schwingungsebene für eine Eins und eine vertikale für eine Null stehen soll. Der Empfänger stellt dann sein Polarisationsfilter horizontal; falls er dahinter ein Photon mißt, wurde eine Null gesendet, ansonsten eine Eins.

Praktisch werden die Photonen meist approximiert durch sehr kurze Lichtblitze, die mit hoher Wahrscheinlichkeit nicht mehr als ein Photon enthalten, weil sie beispielsweise so kurz sind, daß nur mir Wahrscheinlichkeit 1/10 überhaupt ein Photon enthalten ist. Die Wahrscheinlichkeit für zwei oder mehr Photonen in einem nichtleeren Lichtblitz liegt dann bei etwa 6%, was tolerierbar ist.

Da drehbare Polarisationsfilter nur langsam auf eine neue Richtung eingestellt werden können, verwendet man auch hier andere Methoden. Beispielsweise läßt sich der POCKELS-Effekt aus der nichtlinearen Optik ausnutzen, wonach gewisse anisotrope Kristalle beim Anlegen einer Spannung ihren Brechungsindex ändern, oder aber man verwendet für jede gewünschte Polarisationsrichtung eine eigene Laserdiode oder sonstige geeeignete Lichtquelle mit dahinterstehendem Polarisationsfilter und vereinigt anschließend die Strahlengänge. Auf diese Weise lassen sich Lichtblitze mit einer Frequenz von etwa 1 MHz erzeugen.

Da nicht jeder Puls ein Photon enthält und auch während der Übertragung Photonen verloren gehen, muß der Empfänger zunächst wissen, *ob* ein Photon angekommen ist; außerdem muß er dann dessen Polarisationsrichtung bestimmen. Dazu läßt er das Photon durch einen doppelbrechenden Kristall (z.B. einen Kalkspat) gehen; je nachdem ob es parallel oder senkrecht zu dessen optischer Achse polarisiert ist, verläßt es den Kristall an einer von zwei wohldefinierten Stellen, hinter denen Photomultiplikatoren und Detektoren stehen, so daß für jede der beiden Polarisationsrichtungen ein Detektor anspricht. Natürlich müssen Sender und Empfänger synchronisiert werden; dies geschieht beispielsweise dadurch, daß eine festgelegte Zeitspanne vor jedem Puls ein heller Lichtblitz gesendet wird, der garantiert ankommt.

Entsprechende Apparaturen wurden experimentell getestet, erstmalig im Oktober 1989 über eine Entfernung von damals nur 32 cm. Inzwischen ist man bei Glasfaserkabeln einer Länge von knapp 50 km angelangt, sowohl bei aufgerollter Glasfaser im Labor als auch bei „echten“ Glasfaserverbindungen wie etwa der der Swiss Telekom von Genf nach Nyon (22,8 km) unter dem Genfer See. Für eine Vernetzung innerhalb einer Stadt ist Quantenkryptographie also bereits heute praktikabel. Entsprechende Strecken über Tausende von Kilometern erscheinen nach derzeitigem Stand der Technik unwahrscheinlich: In [**ZGHMT**] wird die maximal erreichbare Distanz auf etwa 100 km geschätzt. Der Grund ist vor allem die Absorption in Glasfaserkabeln: Bei den für Telekommunikation typischen Wellenlängen um 1300 und 1550 nm hat man eine Dämpfung von 0,35 beziehungsweise 0,2 dB/km, so daß nach knapp 30 beziehungsweise 50 km nur noch ein Zehntel der abgeschickten Photonen übriggeblieben ist. Für Photonen mit nur 800,nm Wellenlänge, die sich einfacher detektieren lassen, beträgt die Dämpfung sogar 2 dB/km, man hat also schon nach fünf Kilometern neun von zehn Photonen verloren.

Verglichen mit anderen Kryptosystemen, die auch für die drahtlose Übermittlung von Nachrichten benutzt werden können, ist die Notwendigkeit einer Glasfaserverbindung überhaupt ein Nachteil: Besser wäre es, wenn man die Photonen kabellos übertragen könnte.

Das Problem hierbei ist natürlich, daß es in der Luft, vor allem bei Tageslicht, bereits ziemlich viele Photonen gibt. Das ist allerdings nicht ganz so schlimm, wie es auf den ersten Blick aussieht, denn bei hinreichender Sorgfalt kann man den Ort, die Zeit und die Wellenlänge der übermittelten Photonen sehr genau festlegen. Bei einer Wellenlänge von etwa 770 nm (im infraroten Bereich also) ist die Atmosphäre auch so durchlässig, daß sich die Verlustrate in Grenzen hält.

Wissenschaftler der National Laboratories in Los Alomos berichteten in *Nature* im Juli 2000 [**BHLMNP**], daß sie am 13. August 1999 von 9^{30} bis 11^{30} Uhr unter dem wolkenlosen blauen Himmel von New Mexico drahtlose Quantenkryptographie über eine Entfernung von 1,6 km erfolgreich testen konnten. Fernziel ist die magische Grenze von 30 km, die es erlauben würde, Photonen an einen Satelliten zu schicken. Ein solcher Satellit ist zwar von einem festen Punkt der Erde aus nur etwa acht Minuten pro Tag in direkter Sichtverbindung; diese Zeit würde aber, bei den angestrebten Datenraten, ausreichen, um einen Schlüssel zu vereinbaren, mit dem sich deutlich mehr als die typischerweise in einem Unternehmen innerhalb von 24 Stunden anfallenden Nachrichten vom und zum Satelliten verschlüsseln lassen.

2. Protokolle zur Quantenkryptographie

Natürlich wurde bei keinem der vorgestellten Experimente genau das oben skizzierte Verfahren getestet, denn so wie beschrieben bietet es keinerlei kryptographische Sicherheit: Ein Gegner könnte einfach alle Photonen abfangen und neue auf die Reise schicken; falls er diese genauso polarisiert, wie die abgefangenen, wird weder der Sender noch der Empfänger etwas bemerken.

Hier kommt nun die Quantennatur des Lichts ins Spiel: Licht kann nicht nur vertikal oder horizontal polarisiert werden, sondern in jeder beliebigen Richtung (oder auch zirkulär). Geht es dann durch ein auf eine andere Richtung eingestelltes Polarisationsfilter, nimmt die Intensität mit dem Cosinusquadrat der Winkeldifferenz zwischen den beiden Richtungen ab, bis sie bei aufeinander senkrecht stehenden Richtungen verschwindet. Das austretende Licht ist jeweils in Richtung des analysierenden Filters polarisiert.

Im Falle eines einzigen Photons kann die Intensität natürlich nicht abnehmen; das Photon wird entweder durchgelassen oder nicht. Falls die beiden Richtungen weder gleich sind noch aufeinander senkrecht stehen, ist allerdings nicht mehr deterministisch festgelegt, welcher der beiden Fälle eintritt: Bei einer Winkeldifferenz δ zwischen den beiden Polarisationsrichtungen wird das Photon mit Wahrscheinlichkeit $\cos^2 \delta$ durchgelassen und mit Wahrscheinlichkeit $\sin^2 \delta$ absorbiert. Speziell für $\delta = 45°$ wird sind beide Wahrscheinlichkeiten gleich ein halb.

Der erste Ansatz, dies für kryptographische Zwecke auszunutzen, stammt von Charles Bennet und Gilles Brassard aus dem Jahr 1984; er wird heute kurz als das BB84-Protokoll bezeichnet.

Bei diesem Protokoll entscheidet sich der Sender vor dem Senden eines jeden Photons zufällig für ein Referenzsystem, bestehend entweder aus den Richtungen 0° und 90°, oder aus den Richtungen 45° und 135°. Danach entscheidet er sich, wiederum zufällig, für einen der beiden Bitwerte 0 oder 1 und kodiert beispielsweise die Eins durch Polarisationsrichtung 0° oder 45°, die Null durch 90° oder 135°. Dazu braucht er entweder zwei Pockels-Zellen oder vier Lichtquellen mit entsprechend eingestellten Polarisationsfiltern dahinter.

Praktisch würden die Zufallsentscheidungen beispielsweise durch Digitalisieren von weißem Rauschen erfolgen oder aber indem man polarisierte Photonen durch ein um 45° gedrehtes Filter schießt und mißt, welche transmittiert werden.

Auch der Empfänger wählt für jedes Bit zufällig eines der beiden Referenzsysteme; falls er dasselbe System gewählt hat wie der Absender, kann er das von diesem abgeschickte Bit messen, andernfalls erhält er ein Zufallsergebnis, was im Mittel in der Hälfte aller Fälle vorkommen wird.

Die folgende Tabelle faßt die verschiedenen Möglichkeiten für Sender und Empfänger noch einmal zusammen:

gesendet wird:	0°	0°	45°	45°	90°	90°	135°	135°
empfangen mit:	+	×	+	×	+	×	+	×
Meßergebnis:	0°	?	?	45°	90°	?	?	135°

Hier bedeutet „+“, daß der Empfänger seinen Doppelspat so orientiert, daß er 0° und 90°-Polarisierung messen kann, während „×“ entsprechend für das um 45° gedrehte Bezugssystem steht. Ein „?“ in der letzten Zeile soll besagen, daß hier das Meßergebnis nur vom Zufall abhängt und somit keine sinnvolle Information enthält.

Um festzustellen, welche Hälfte der gemessenen Bits sinnvolle Information enthält, informiert der Empfänger den Sender anschließend über einen gewöhnlichen, nicht abhörsicheren Kanal, welche Photonen angekommen sind und mit welchem Referenzsystem er sie gemessen hat. Der Sender teilt ihm dazu jeweils mit, ob dies die richtige Wahl war oder nicht. Dieser Austausch soll im folgenden kurz als die „öffentliche“ Diskussion bezeichnet werden.

Für die Photonen, bei denen beide mit demselben Referenzsystem gearbeitet haben, notiert sich der Sender den gesendeten und der Empfänger den gemessenen Bitwert; bis auf allfällige Übertragungsfehler sollten diese somit beide dieselbe Bitfolge notieren. Der Lauscher, der nachträglich zwar das korrekte Referenzsystem erfahren hat, nicht aber den übertragenen Bitwert, kann zumindest aus der „öffentlichen“ Diskussion nichts darüber erfahren. Seine sonstigen Möglichkeiten sollen weiter unten erörtert werden.

Zunächst aber soll hier noch das 1992 von Bennet vorgeschlagene logisch und technisch etwas einfachere B92-Protokoll vorgestellt werden. Hier verwendet der Sender nur die beiden Polarisationsrichtungen 0° für Null und 45° für Eins; der Empfänger nur 135° für Null und 90° für Eins. Insbesondere braucht der Empfänger also keinen doppelbrechenden Kristall mehr mit zwei Detektoren, sondern nur noch eine Pockels-Zelle mit *einem* nachgeschalteten Detektor. Die möglichen Meßergebnisse sind in folgender Tabelle zusammengefaßt:

gesendet wird	0°	0°	45°	45°
gemessen wird mit	90°	135°	90°	135°
Photon wird detektiert?	nein	?	?	nein

Das „?“ in der letzten Zeile soll dabei bedeuten, daß der Empfänger mit gleicher Wahrscheinlichkeit ein Photon mißt oder auch nicht.

Es gibt also nur zwei Kombinationen, bei denen der Empfänger überhaupt ein Photon finden kann: Falls der Empfänger mit 0° gesendet und der Empfänger mit 135° gemessen hat, oder wenn der Empfänger mit 45° gesendet und der Empfänger mit 90° gemessen hat. Falls ein Photon gemessen wird, weiß der Empfänger also, mit welcher Polarisationsrichtung gesendet wurde und kann je nachdem eine Null oder Eins notieren.

Im anschließenden „öffentlichen“ Dialog muß er dem Sender dann nur mitteilen, in welchen Positionen er Photonen gemessen hat, so daß auch der sich die Bitwerte dafür notieren kann. Man beachte, daß bei diesem Protokoll nur jedes vierte übermittelte Photon zu einem Bitwert führt.

3. Elimination der gegnerischen Information

Sowohl beim BB84- als auch beim B92-Protokoll kennen Sender und Empfänger am Ende eine Bitfolge, über die ein Gegner zumindest durch Abhören der „öffentlichen“ Diskussion nichts in Erfahrung bringen kann. Er hat aber natürlich auch noch die Möglichkeit, sich in den anfänglichen Photonenaustausch einzuschalten.

Hier wird die Quantentheorie wichtig: Ein Lauscher hat keine Möglichkeit, ein Photon zu messen, ohne es zu verändern. Falls er beispielsweise ein mit 0°-Polarisierung gesendetes Photon mit einem Doppelspat im schrägen Referenzsystem mißt, hat er anschließend mit gleicher Wahrscheinlichkeit ein mit 45° oder 135° polarisiertes Photon; er weiß aber nicht, ob das Photon wirklich mit dieser Polarisierung ankam, oder ob es eine der beiden Polarisierungsrichtungen 0° und 90° hatte.

Hier ist extrem wichtig, daß mit einzelnen Photonen gearbeitet wird: Falls zwei Photonen im selben Zustand übertragen werden, kann man sie durch einen Strahlteiler trennen und jedes in einem anderen Referenzsystem messen; beim BB84-Protokoll wird dann in der „öffentlichen“ Diskussion klar, welche der Messungen zum richtigen Bitwert führte, beim B92-Protokoll ist er in drei Viertel aller Fälle sofort klar, da eine Polarisationsrichtung von 90° im $+$-System nur gemessen werden kann, wenn das Photon im $\times$-System übertragen wurde, d.h. mit Polarisationsrichtung 45°. Entsprechend kann 135° im $\times$-System nur gemessen werden kann, wenn das Photon mit $0°$ polarisiert war. Lediglich bei der Kombination $(0°, 45°)$ ist nicht klar, welches Bit der Sender übermitteln wollte.

Mit nur einem Photon kann der Gegner jedoch nie sicher sein, welchen Zustand das gesendete Photon hatte und muß sich überlegen, was er als nächster tun will. Er hat zwei Möglichkeiten:

Als erstes könnte er das Photon einfach verschwinden lassen. Das ist aus seiner Sicht nicht sinnvoll: Falls es ihm nur darum geht, die Kommunikation unmöglich zu machen, hat er einfachere Möglichkeiten. Er muß also wieder irgendein Photon senden.

Am sichersten wäre es, ein Photon zu senden, das genau dieselbe Polarisationsrichtung hat wie das abgefangene; in diesem Fall bliebe sein Lauschen unbemerkt. Diese Möglichkeit wird aber durch die Quantentheorie ausgeschlossen, da er den Zustand des Photons nicht vollständig bestimmen kann.

Am wenigsten verfälscht er beim BB84-Protokoll, wenn er ein Photon losschickt, das die von ihm gemessene Polarisationsrichtung hat: Falls er im richtigen Referenzsystem maß, bleibt sein Eingriff bei diesem Photon unbemerkt, andernfalls gibt es immerhin noch eine 50%-ige Wahrscheinlichkeit, daß der Empfänger, falls er im richtigen System mißt, den korrekten gesendeten Wert mißt. (Falls auch der Empfänger im falschen System mißt, wird das Bit nicht verwendet, so daß es auf diesen Fall nicht ankommt.) In etwa einem Viertel aller Fälle sorgt der Lauscher durch seinen Eingriff dafür, daß der Empfänger trotz gleichen Referenzsystems einen anderen Bitwert mißt als den ursprünglich gesendeten.

Beim B92-Protokoll kann ein Lauscher, der wie der Empfänger beim B84-Protokoll mit Doppelbrechung arbeitet, jedes zweite Bit messen: Falls er im $+$-System $90°$ mißt, muß das Photon mit $45°$ polarisiert gewesen sein, wenn er im $\times$-System $135°$ mißt, mit $45°$. In diesem Fall kann er ein entsprechendes Ersatzphoton schicken. In der anderen Hälfte der Fälle muß er aber raten und verfälscht somit auch hier wieder insgesamt rund ein Viertel aller übertragener Bitwerte.

Sender und Empfänger einigen sich daher in der „öffentlichen“ Diskussion auf eine Zufallsstichprobe der notierten Bits und vergleichen diese; auf diese Weise können sie die Fehlerrate der Übertragung feststellen und durch Vergleich mit der (bekannten) Fehlerrate bei ungestörter Kommunikation abschätzen, wieviel Information der Gegner in Erfahrung bringen konnte.

Falls sie einen Abhörversuch feststellen, haben sie die Möglichkeit, die Kommunikation abzubrechen und zu einem späteren Zeitpunkt, eventuell nach genauer Untersuchung der Verbindungsstrecke, einen neuen Versuch zu wagen. Sie können aber auch versuchen, die vom Gegner gewonnene Information aus ihrer gemeinsamen Bitfolge zu eliminieren. Dies geschieht in drei Schritten:

Zunächst wird natürlich die Stichprobe aus der Bitfolge eliminiert: Diese Bits wurden in der „öffentlichen“ Diskussion verglichen und sind somit dem Gegner bekannt.

Im zweiten geht es darum, überhaupt eine gemeinsame Bitfolge zu bekommen: Durch den Vergleich einer Stichprobe kann schließlich nur die Fehler*rate* ermittelt werden, nicht aber die Stellen, an denen sich außerhalb der Stichprobe Fehler befinden. Dieser zweite Schritt ist unabhängig von der Aktivität eines Gegners notwendig, da auch bei einer nicht abgehörten Übertragung Photonen ihren Zustand ändern können, Detektoren gelegentlich auch nicht existente Photonen melden *usw.*

Die Grundidee ist dieselbe wir in der klassischen Kodierungstheorie: Bei jeder Übertragung von Information über einen Kanal gibt es Störungen, die zur Verfälschung eines Teils der Information führen. Um diese zu eliminieren, überträgt man zusätzlich zur eigentlichen Information noch zusätzliche Prüfbits, mit deren Hilfe man eine (vom verwendeten Code abhängige) Anzahl von Bitfehlern pro Codewort korrigieren kann.

Nehmen wir etwa an, die vereinbarte Bitfolge sei aus Sicht des des Senders der Vektor $\mathfrak{v}$, aus Sicht des Empfängers aber $\mathfrak{v}' = \mathfrak{v} + \mathfrak{u}$, wobei $\mathfrak{u}$ der Fehlervektor ist.

Falls sich $\mathfrak{v}$ als Folge von Worten aus einem fehlerkorrigierenden Code auffassen ließe, könnte der Empfänger einfach die Dekodierungsabbildung dieses Codes auf $\mathfrak{v}'$ anwenden, um $\mathfrak{v}$ zu erhalten. Da aber nur ein Bruchteil der Blitze zu Komponenten von $\mathfrak{v}$ führt, hat der Sender keine Möglichkeit, $\mathfrak{v}$ entsprechend zu präparieren. Er behilft sich daher mit einem Trick: Er wählt einen zufälligen Bitvektor und kodiert diesen mit einem fehlerkorrigierenden Code, der für die ermittelte Fehlerrate ausreicht; die Länge dieses Vektors sei so gewählt, daß der enstehende Codevektor $\mathfrak{w}$ dieselbe Länge wie $\mathfrak{v}$ hat. Sodann berechnet er $\mathfrak{v} + \mathfrak{w}$ und überträgt diesen Vektor über die ungesicherte Leitung – je nach deren Qualität eventuell wiederum gesichert durch einen fehlerkorrigierenden Code. Auch die Art der verwendeten Codes wird dem Empfänger über die ungesicherte Leitung mitgeteilt.

Der Empfänger kennt dann sowohl $\mathfrak{v} + \mathfrak{w}$ als auch $\mathfrak{v}' = \mathfrak{v} + \mathfrak{u}$, er kann also deren Differenz $\mathfrak{v} + \mathfrak{w} - \mathfrak{v}' = \mathfrak{w} - \mathfrak{u} = \mathfrak{w} + \mathfrak{u}$ berechnen. (Da wir mit Bits rechnen, gibt es keinen Unterschied zwischen plus und minus.) Das ist aber das mit dem Fehler $\mathfrak{u}$ gestörte Codewort $\mathfrak{w}$, die Dekodierungsfunktion des fehlerkorrigierenden Codes erlaubt also die Rekonstruktion von $\mathfrak{w}$. Damit ist dann aber auch $\mathfrak{u}$ berechenbar und somit auch $\mathfrak{v}$.

Der Lauscher weiß weniger über $\mathfrak{v}$ als der Empfänger; er kann zwar eventuell auch etwas von der Fehlerkorrektur im zweiten Schritt profitieren, hat aber immer noch keine vollständige Information über $\mathfrak{v}$.

Dieses unterschiedliche Wissen über $\mathfrak{v}$ wird nun im dritten Schritt, der sogenannten *privacy amplification*, ausgenutzt, um $\mathfrak{v}$ auf einen Vektor $\mathfrak{z}$ zu verkürzen, über den der Lauscher mit hoher Wahrscheinlichkeit so gut wie nichts weiß. Die Längendifferenz zwischen $\mathfrak{v}$ und $\mathfrak{z}$ entspricht dabei der Information, die der Lauscher über $\mathfrak{v}$ gewonnen hat; diese kann leicht aus der Fehlerrate berechnet werden.

Zur Berechnung von $\mathfrak{z}$ einigen sich Sender und Empfänger wieder in „öffentlicher“ Diskussion, über die (zufällige) Auswahl einer Hashfunktion φ aus einer großen Anzahl vorher vereinbarter solcher Funktionen. Dies darf erst *nach* Übertragung der Photonen geschehen, denn wenn der Lauscher diese Funktion kennt, kann er seine Abhörstrategie darauf abstimmen und sie praktisch wirkungslos machen; insbesondere gibt es also immer ein kleines Restrisiko, daß der Lauscher durch vorheriges Erraten der richtigen Funktion durch deren Anwendung keine Information verliert.

Bei einem hinreichend großen Raum von Hashfunktionen ist dieses Restrisiko jedoch verschwindend klein, und eine genauere informationstheoretische Analyse zeigt, daß die Information, die der Lauscher über $\varphi(\mathfrak{v})$ hat, mit sehr hoher Wahrscheinlichkeit sehr nahe bei Null liegt. Die Behandlung der Einzelheiten dieser Analyse und des dahinterstehenden Begriffsapparats würde hier zu weit führen; interessierte Leser seien auf die Originalarbeit [**BBR**] verwiesen, vor allen den dortigen Paragraphen 4.2.

Zum Schluß sei nach angemerkt, daß auch die Sätze aus [**BBR**] noch nicht beweisen, daß Quantenkryptographie wirklich sicher ist – auch nicht bis auf das erwähnte Restrisiko. Wir sind nämlich immer nur davon ausgegangen, daß der Lauscher sich darauf beschränkt, einzelne Photonen zu messen und in geeigneter Weise zu ersetzen. Tatsächlich könnte er stattdessen auch irgendwelche Interferenzerscheinungen oder andere Daten über aus vielen Photonen zusammengesetzte Zustände messen und daraus Schlußfolgerungen ziehen.

Realistischerweise muß man zur Abschätzung der Sicherheit des Verfahrens dem Gegner erlauben, alle Messungen durchzuführen, die die Quantentheorie nicht ausdrücklich verbietet, darunter auch solche, an die bislang noch niemand gedacht hat.

Unter diesen Umständen erfordert die Analyse seiner Möglichkeiten erheblich tiefere Methoden aus der Quantentheorie, als die einfache Darstellung in diesem Seminar bieten kann. Mit solchen Methoden kann man dann allerdings beweisen, daß Quantenkryptographie auch gegenüber einem Gegner mit unbeschränkten Ressourcen im erwähnten Sinne sicher ist; siehe dazu etwa [**SP**].

4. Literaturhinweise

Eine erste Übersicht gibt der Artikel

CHARLES H. BENNET, GILLES BRASSARD, ARTHUR K. EKERT: Quantenkryptographie, *Spektrum der Wissenschaft,* Dezember 1992, S. 96–104

Neuere Entwicklungen findet man in den Kapiteln über Quantenkryptographie der Bücher

MICHAEL BROOKS[Hrsg.]: *Quantum computing and communications,* Springer, 1999 *und*

COLIN P. WILLIAMS, SCOTT H. CLEARWATER: *Explorations in quantum computing,* Springer Telos, 1997

Das erste der beiden Bücher besteht aus ziemlich kurzen und wenig technischen Darstellungen, die sich vor allem an Laien wenden; im zweiten wird auch auf die Mathematik und die Physik hinter den Verfahren eingegangen. Außerdem enthält dieses Buch eine CD mit Mathematica *notebooks* zur Simulation von Quantenprozessen.

Hier im Text zitiert wurden

[**BBR**] CHARLSES BENNETT, GILLES BRASSARD, JEAN MARC ROBERT: Privacy amplification by public discussion, *SIAM J. Comp.* **17** (1988), 210–229

[**BHLMNP**] W.T. BUTTLER, R.J. HUGHES, S.K. LAMOREAUX, G.L. MORGAN, J.E. NORDHOLT, C.G. PETERSON: Daylight quantum key distribution over 1.6 km, *Phys. Rev. Letters* **84**, 24 *(12. Juni 2000),* 5652–5255

[**SP**] PETER W. SHOR, JOHN PRESKILL: Simple proof of security of the BB84 quantum key distribution protocol, *Phys. Rev. Letters* **85**,2 *(10. Juli 2000),*441–444

[**W**] DOMINIC WELSH: *Codes und Kryptographie,* VCH Weinheim, 1991

[**ZGHMT**] H. ZBINDEN, N. GISIN, B. HUTTNER, A. MULLER, W. TITTEL: Practical Aspects of Quantum Cryptographic Key Distribution, *J. Cryptology* **13** (2000), S. 207–220

III. Quantencomputer

Die Quantentheorie ist nicht nur in der Lage, Kryptographie sicherer zu machen, sie stellt potentiell auch eine ernste Bedrohung existierender Kryptoverfahren dar, die mit einem sogenannten Quantencomputer möglicherweise leicht gebrochen werden können. Besonders gefährdet sind die sogenannten asymmetrischen oder *public key*-Verfahren, die wir daher (auch wenn sie zur Gegenwart gehören) zumindest kurz rekapitulieren müssen.

1. Kryptographie mit öffentlichen Schlüsseln

In der klassischen Kryptographie gibt es zu jedem Kryptoverfahren einen Schlüssel, dessen Kenntnis sowohl zur Verschlüsselung als auch zur Entschlüsselung notwendig ist. 1976 veröffentlichten MARTIN HELLMANN, Elektrotechniker an der Stanford University, und WHITFIELD DIFFIE, ein *graduate student,* in den *IEEE Transactions on Information Theory* eine Arbeit mit dem Titel *New directions in cryptography,* in der sie behaupteten, man könne auch Verfahren entwickeln, bei denen zwar jedermann eine Nachricht verschlüsseln könne, aber trotzdem nur der legitime Empfänger in der Lage sei, sie zu *ent*schlüsseln. Inzwischen ist bekannt, daß in Großbritannien COCKS und ELLIS bereits zwei Jahre vorher ein entsprechendes System entwickelten, das aber aus Gründen der „nationalen Sicherheit“ nie veröffentlicht wurde.

DIFFIE und HELLMANN stellten noch kein explizites derartiges Verfahren vor, aber zwei Jahre später publizierten RON RIVEST, ADI SHAMIR und LEONARD ADLEMAN vom MIT in den *Communications of the ACM* ihre Arbeit *A Method for Obtaining Digital Signatures and Public Key*

Cryptosystems, die das heute nach ihren Initialen als RSA bekannte und kommerziell sehr erfolgreiche Kryptosystem vorstellte.

Ausgangspunkt ist die Beobachtung, daß es zwar sehr leicht ist, zwei Zahlen miteinander zu multiplizieren, daß aber die inverse Operation, die Faktorzerlegung, erheblichen Aufwand erfordert.

Man wähle sich daher zwei große Primzahlen p und q und berechne deren Produkt N; außerdem wähle man einen zu $(p-1)(q-1)$ primen Exponenten e zwischen 2 und $N-2$. Eine zu übermittelnde Nachricht wird aufgespalten in Blöcke, die sich durch Zahlen zwischen 0 und $N-1$ kodieren lassen, und ein solcher Block a wird chiffriert als

$$b = a^e \bmod N\,,$$

was sich auch für große Werte von N und e noch ziemlich effizient berechnen läßt.

Zur Berechnung von a aus b beachte man, daß für die beiden Primzahlen p und q

$$x^p \equiv x \bmod p \quad \text{und} \quad x^q \equiv x \bmod q \quad \text{für alle } x$$

gilt: Nach dem binomischen Lehrsatz ist $(u+v)^p$ gleich

$$u^p + \binom{p}{1}u^{p-1}v + \binom{p}{2}u^{p-2}v^2 + \cdots + \binom{p}{p-2}u^2v^{p-2} + \binom{p}{p-1}uv^{p-1} + v^p\,,$$

und hierin sind wegen

$$\binom{p}{k} = \frac{p!}{k!(p-k)!}$$

alle Binomialkoeffizienten mit $1 \le k \le p-1$ durch p teilbar. Somit ist

$$(u+v)^p \equiv u^p + v^p \bmod p\,,$$

woraus die Behauptung leicht durch vollständige Induktion folgt.

Falls x nicht durch p oder q teilbar ist, ist dann aber auch

$$x^{p-1} \equiv 1 \bmod p \quad \text{und} \quad x^{q-1} \equiv 1 \bmod q\,,$$

also auch

$$x^{(p-1)(q-1)} \equiv 1 \bmod p \quad \text{und} \quad x^{(p-1)(q-1)} \equiv 1 \bmod q$$

und somit

$$x^{(p-1)(q-1)} \equiv 1 \bmod N\,.$$

Allgemeiner gilt für jede ganze Zahl k

$$x^{k(p-1)(q-1)} \equiv 1 \bmod N .$$

Multiplikation mit x führt auf

$$x^{k(p-1)(q-1)+1} \equiv x \bmod N ,$$

wobei diese Formel wieder für *alle* x gilt.

Nach Voraussetzung ist e teilerfremd zu $(p-1)(q-1)$; mit dem EUKLIDischen Algorithmus lassen sich also ganze Zahlen d und k finden, so daß

$$de + k(p-1)(q-1) = 1$$

ist. Da sich diese Gleichung nicht ändert, wenn man ein Vielfaches der Gleichung

$$(p-1)(q-1)e - e(p-1)(q-1) = 0$$

addiert, läßt sich leicht erreichen, daß dabei $d > 0$ ist. Alsdann ist

$$b^d = a^{ed} = a^{1-k(p-1)(q-1)} = a \cdot a^{-k(p-1)(q-1)} \equiv a \cdot 1 = a \bmod N .$$

Veröffentlicht man also N und e, so kann damit jedermann Nachrichten verschlüsseln; zum Entschlüsseln benötigt man aber d, in dessen Berechnung wiederum die Zahl $(p-1)(q-1)$ einging, deren Kenntnis gleichbedeutend mit der von p und q ist. d wäre hier also ein zusätzlicher *geheimer* Schlüssel.

Dieser kann auch zur Erzeugung elektronischer Unterschriften benutzt werden: Als elektronische Unterschrift zur Nachricht a mit $2 \le a \le N-2$ dient

$$u = a^d \bmod N .$$

Jedermann kann nachrechnen, daß

$$u^e = a^{de} \equiv a \bmod N$$

ist und somit die Unterschrift überprüfen; erzeugen läßt sie sich nur mit Kenntnis von d.

Von DIFFIE und HELLMANN stammt ein Verfahren zur Schlüsselvereinbarung über eine unsichere Leitung. Auch hier geht es um eine Umkehrung der Funktion $a^e \bmod N$, nun aber bezüglich des Exponenten: Für eine Primzahl p finde man für festes a zwischen zwei und $p-2$ und gegebenes b

einen Exponenten e, für den $b \equiv a^e \bmod p$ ist, den sogenannten diskreten Logarithmus von b zur Basis a modulo p. Es gibt eine ganze Reihe von Algorithmen, mit denen sich e berechnen läßt; sie sind aber allesamt deutlich aufwendiger als die Berechnung von $a^e \bmod p$.

Davon ausgehend, schlagen DIFFIE und HELLMANN folgendes Verfahren vor: Die beiden Partner A und B einigen sich auf eine Primzahl p und auf $2 \leq a \leq p-2$; sodann erzeugt A eine Zufallszahl u, berechnet die Potenz $x = a^u \bmod p$ und schickt diese an B; dieser erzeugt eine Zufallszahl v, berechnet $y = a^v \bmod p$ und schickt dies an A. Mit dieser Information kann jeder der beiden die Zahl

$$S = a^{uv} \bmod p = x^v \bmod p = y^u \bmod p$$

berechnen, ein Lauscher, der nur p, a, x und y kennt, aber weder u noch v, kann dies nicht. S kann also als Schlüssel dienen.

Man beachte den großen Unterschied zwischen dieser Form der Schlüsselvereinbarung und dem entsprechenden Algorithmus auf Grundlage der Quantenkryptographie: Letzterer ist sicher auch gegenüber einem Gegner mit unbegrenzten rechnerischen Ressourcen; das hier vorgestellte Verfahren aber bietet gegenüber einem solchen Gegner keinerlei Schutz, da er beispielsweise aus a und x problemlos u durch systematisches Ausprobieren berechnen könnte.

Auch das RSA-Verfahren bietet keinerlei Schutz gegen einen solchen Gegner, denn für ihn ist auch Faktorisieren kein Problem. Der Schutz der hier vorgestellten Verfahren beruht also ausschließlich darauf, daß ein Gegner mit *beschränkten* rechnerischen Möglichkeiten eine prinzipiell lösbare mathematische Aufgabe nicht in akzeptabler Zeit lösen kann.

Quantencomputer, so sie je existieren, können die rechnerischen Möglichkeiten eines Gegners so dramatisch erweitern, daß alle diese Verfahren wertlos werden könnten.

2. Quantenmechanische Grundlagen

Zum Verständnis eines Quantencomputers müssen wir uns etwas stärker mit dem formalen Apparat der Quantenmechanik befassen als dies bei der Quantenkryptographie notwendig war.

Grundlage der Beschreibung eines quantenmechanischen Systems ist immer sein *Zustandsraum,* ein Vektorraum über dem Körper $\mathbb{C}$ der komplexen Zahlen mit einem HERMITEschen Skalarprodukt. Dieses Produkt wird in der Form $\langle u|v\rangle$ geschrieben, und von daher hat sich die DIRACsche Konvention eingebürgert, die Elemente von V in der Form $|v\rangle$ zu schreiben. Da $\langle u|v\rangle$ eine *bracket* ist, wird $|v\rangle$ allein meist kurz als ein *ket* bezeichnet.

Diese Schreibweise mag einem Mathematiker auf den ersten Blick verwirrend erscheinen; er sollte sich aber klarmachen, daß es hier nur um ein Symbol geht und sich an der Mathematik natürlich nichts ändert, egal ob man einen Vektor als $\mathfrak{v}$, $\vec{v}$ oder eben $|v\rangle$ schreibt.

Die Vektoren aus V beschreiben die möglichen Zustände des Systems, wobei zueinander proportionale Vektoren denselben Zustand beschreiben; der tatsächliche Zustandsraum ist also der projektive Raum $\mathbb{P}(V)$. Da es somit auf die Länge des Zustandsvektors nicht ankommt, wird diese vielfach auf eins normiert.

Meßbare physikalische Größen werden durch HERMITEsche Operatoren $A\colon V \to V$ beschrieben. Ist das System vor der Messung im Zustand, der durch den Einheitsvektor $|v\rangle$ beschrieben wird, so ist sein Zustand nach der Messung ein Eigenvektor von A und der Meßwert ist der zugehörige Eigenwert. Zu einem HERMITEschen Operator gibt es bekanntlich immer eine Orthonormalbasis $\mathfrak{B}$ von V aus Eigenvektoren von A; falls die zugehörigen Eigenwerte alle verschieden sind, ist die Wahrscheinlichkeit für den Eigenwert λ und den zugehörigen Eigenvektor $|w\rangle$ der Länge eins gleich dem Betragsquadrat des Produkts von $|v\rangle$ und $|w\rangle$.

Betrachten wir als Beispiel die Photonen aus dem vorigen Kapitel. Der Zustandsraum für ein Photon ist ein zweidimensionaler Vektorraum, in dem die Polarisationsrichtung φ etwa durch den Vektor $\binom{\cos\varphi}{\sin\varphi}$ repräsentiert wird; im Falle eines reellen Vektorraums wäre jeder Vektor proportional zu einem solchen Einheitsvektor. Daß wir komplexe Vektorräume betrachten, bedeutet, daß zum Zustand auch noch ein Phasenfaktor gehört, der für die vorgestellten Protokolle zur Quantenkryptographie aber keine Rolle spielt.

Die Messung der Polarisationsrichtung mit einem Polarisationsfilter mit Durchlaßrichtung ψ wird durch den Operator

$$A_\psi = \begin{pmatrix} \cos 2\psi & \sin 2\psi \\ \sin 2\psi & -\cos 2\psi \end{pmatrix} = \begin{pmatrix} \cos^2\psi - \sin^2\psi & 2\sin\psi\cos\psi \\ 2\sin\psi\cos\psi & \sin^2\psi - \cos^2\psi \end{pmatrix}$$

modelliert; seine Eigenwerte sind ± 1 mit Eigenvektoren

$$|v_1\rangle = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} \quad \text{und} \quad |v_{-1}\rangle = \begin{pmatrix} -\sin\psi \\ \cos\psi \end{pmatrix}.$$

Bei einem in Richtung φ polarisierten Photon liegt also die Wahrscheinlichkeit für $+1$ beim Quadrat von

$$\begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix}\begin{pmatrix}\cos\psi\\ \sin\psi\end{pmatrix} = \cos\varphi\cos\psi + \sin\varphi\sin\psi = \cos(\varphi-\psi)\,,$$

die für -1 entsprechend bei $\sin^2(\varphi-\psi)$.

3. Quantenregister und QBits

In einem klassischen Computer werden Bits durch die beiden Zustände eines bistabilen Schaltelements wie etwa eines Flipflops oder durch eine Magnetisierungsrichtung oder etwas ähnliches realisiert. Unabhängig von der physikalisch-technischen Realisierung hat man immer ein Element, das sich stets in einem von zwei wohldefinierten Zuständen befindet; diese werden traditionell als 0 und 1 bezeichnet.

Angenommen, wir verwenden stattdessen ein Photon. Das ist beim heutigen Stand der Technologie zwar völlig unrealistisch, aber es ist wohl die anschaulichste Art, das Prinzip eines Quantencomputers zu verstehen; weiter unten werden auch realitätsnähere Ansätze diskutiert werden.

Zur Kodierung eines Bits können wir etwa vereinbaren, daß die horizontale Polarisation einer Null und die vertikale einer Eins entsprechen soll; mit einem Polarisationsfilter lassen sich Photonen in jedem der beiden Zustände unschwer produzieren.

Wir können aber auch anders vorgehen: Wir produzieren ein Photon mit Polarisationsrichtung 45° und lassen es so auf einen Doppelspat fallen, daß der eine der beiden austretenden Strahlen horizontal polarisiert ist, der andere vertikal.

Da nur ein Photon in den Kristall eintritt, kann auch nur eines austreten; indem wir zwei Detektoren in die beiden austretenden Strahlengänge stellen, können wir leicht feststellen, in welchem der beiden Strahlen sich das Photon befindet.

Dieses Aufstellen von Detektoren ist allerdings eine Messung, der Zustandsvektor des Photons wird also auf einen Eigenraum eines Operators projiziert. Was wir messen, ist daher nicht wirklich das aus dem Kristall austretende Photon. Dessen Zustand ist nach der experimentell sehr gut bestätigten Lehre der Quantentheorie, solange es keiner Messung unterworfen wird, eine Überlagerung der beiden möglichen Meßergebnisse: Bezeichnen wir die beiden möglichen Zustände *nach* der Messung mit $|0\rangle$ und $|1\rangle$, so befindet sich das Photon *vor* der Messung in einem Zustand, der durch den (auf Länge eins normierten) Vektor

$$\frac{\sqrt{2}}{2}\,|0\rangle + \frac{\sqrt{2}}{2}\,|1\rangle$$

beschrieben wird. Hier sind also die beiden Zustände $|0\rangle$ und $|1\rangle$ überlagert, und falls der Winkel 45° durch einen anderen ersetzt wird, lassen sich auch die Koeffizienten vor $|0\rangle$ und $|1\rangle$ beliebig verändern.

Ein solches quantenmechanisches System mit einem zweidimensionalen Zustandsraum bezeichnen wir als ein *QBit.* Im Gegensatz zu einem gewöhnlichen Bit, das nur die Werte 0 und 1 annehmen kann, sind für ein QBit also beliebige Werte der Form

$$\alpha\,|0\rangle + \beta\,|1\rangle \quad \text{mit} \quad \alpha, \beta \in \mathbb{C}$$

möglich.

Setzt man mehrere QBits zusammen, entsteht ein Quantenregister. Wichtig ist dabei, daß dieses Quantenregister ein einziges quantenmechanisches System ist, es ist also beispielsweise durchaus möglich, daß der Zustand „alle QBits sind $|0\rangle$“ mit dem Zustand „alle QBits sind $|1\rangle$“ überlagert ist. In diesem Fall wüßten wir zwar nicht im voraus, welches Ergebnis eine Messung eines QBits aus dem Register hätte, wir könnten uns aber sicher sein, daß nach der Messung des ersten QBits jede Messung eines weiteren QBits zum selben Ergebnis führte.

In Dirac-Notation schreibt man

$$|\alpha_1\alpha_2\ldots\alpha_n\rangle$$

für den Zustand, in dem das i-te QBit den Wert $\alpha_i \in \{0, 1\}$ hat; die 2^n Vektoren, die man auf diese Weise erhält, bilden offenbar eine Basis des Zustandsraums für das Quantenregister. (Mathematisch betrachtet ist dieser das Tensorprodukt $V_1 \otimes V_2 \otimes \cdots \otimes V_n$ der Zustandsräume V_i der einzelnen QBits.) Der Zustand eines Quantenregisters wird also beschrieben durch eine Linearkombination von Vektoren der obigen Form wie etwa

$$\frac{\sqrt{2}}{2}\,|00\ldots0\rangle\,\frac{\sqrt{2}}{2}\,|11\ldots1\rangle$$

für das Beispiel aus dem vorigen Absatz.

4. Quantencomputer

Genau wie ein klassischer Computer den Inhalt klassischer Register manipuliert, manipuliert ein Quantencomputer den Inhalt von Quantenregistern.

Das fundamentale Grundgesetz der Quantenmechanik, die SCHRÖDINGER-Gleichung, sagt uns, wie sich deren Inhalt verändern kann: Ist $|\psi(t)\rangle$ der Zustandsvektor zur Zeit t und wird das System durch keine äußeren Einflüsse gestört, so ist

$$i\hbar\frac{\partial\,|\psi(t)\rangle}{\partial t} = H\,|\psi(t)\rangle\ ,$$

wobei H den HAMILTONschen Operator des Systems bezeichnet, jenen HERMITEschen Operator also, der die Gesamtenergie des Systems beschreibt, und $\hbar = h/2\pi \approx 1{,}054589 \times 10^{-34}$ Js ist das durch 2π dividierte PLANCKsche Wirkungsquantum.

Die SCHRÖDINGER-Gleichung ist ein System linearer Differentialgleichungen mit konstanten Koeffizienten; seine Lösung läßt sich mit Hilfe der Matrixexponentialfunktion sofort hinschreiben:

$$|\psi(t)\rangle = e^{-iHt/\hbar}\,|\psi(0)\rangle = U(t)\,|\psi(0)\rangle \quad \text{mit} \quad U(t) = e^{-iHt/\hbar}\,.$$

Dabei ist

$$U(t)\cdot\overline{U(t)}^{T} = e^{-iHt/\hbar}\cdot e^{i\overline{H}^{T}t/\hbar} = E$$

die Einheitsmatrix, denn $\overline{H}^{T} = H$ nach Definition eines HERMITEschen Operators. Somit ist der Operator $U(t)$, der die zeitliche Entwicklung des Systems beschreibt, unitär und insbesondere auch invertierbar.

Diese Invertierbarkeit ist ein großer Unterschied zu klassischen Computern: Die Addition $3+5=8$ etwa läßt sich nicht invertieren, denn das Ergebnis „8“ enthält keine Information mehr darüber, wie es zustandegekommen ist. Langfristig werden aber möglicherweise auch klassische Computer bei immer weitergehender Miniaturisierung der Bauelemente mit reversibler Logik rechnen müssen, denn die einzige Stelle beim Rechnen, bei der Energieverbrauch aus physikalischen Gründen nicht vermieden werden kann, ist die Vernichtung von Information.

Von den klassischen Logikoperationen, mit denen heutige Computer arbeiten, ist nur die Negation reversibel und natürlich auch unitär: Bezüglich der Basis $\{|0\rangle, |1\rangle\}$ des Zustandsraums eines Photons wird sie durch die Matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

beschrieben. Konjunktion und Disjunktion sind aus demselben Grund wie die Addition nicht reversibel: Für eine reversible Logikoperation muß die Anzahl der Ausgangsbits gleich der der Eingangsbits sein. Ein Beispiel mit zwei Eingangsbits ist die kontrollierte Negation

$$(x, y) \mapsto \begin{cases} (x, \neg y) & \text{falls } x = 1 \\ (x, y) & \text{falls } x = 0 \end{cases} = (x, x \oplus y) ,$$

wobei $\oplus$ die Addition modulo 2 bezeichnet.

Bezüglich der Basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ wird die kontrollierte Negation durch die unitäre Matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

beschrieben, und sie ist auch quantenmechanisch realisierbar.

Wichtiger ist das TOFFOLI-Gate, das auf drei QBits operiert:

$$(x, y, z) \mapsto T(x, y, z) \underset{\text{def}}{=} \big((x, y, z \oplus (x \wedge y)\big) ,$$

denn mit seiner Hilfe lassen sich auf Kosten eines zusätzlichen Bits „und“ und „oder“ realisieren:

$$T(x, y, |0\rangle) = (x, y, x \wedge y) \quad \text{und} \quad T(\neg x, \neg y, |1\rangle) = (\neg x, \neg y, x \vee y) .$$

In der Standardbasis des Zustandsraums für drei QBits vertauscht das TOFFOLI-Gate einfach die beiden Vektoren $|110\rangle$ und $|111\rangle$, es wird also durch eine unitäre Matrix beschrieben. Seine quantenmechanische Realisierung *ohne* Phasenverschiebung ist etwas trickreich, aber möglich. Insbesondere kann es als eine Folge von Operationen mit jeweils höchstens zwei Eingangsvariablen realisiert werden, was für die technische Realisierung von großer Bedeutung ist: Interaktionen zwischen drei Quantenbits gleichzeitig wären zu weit jenseits des derzeit Vorstellbaren.

Ein Quantencomputer kann also mit der Negation und dem TOFFOLI-Gate alle logischen Berechnungen durchführen. Arithmetische Operationen können nach den klassischen Regeln der Schaltalgebra auf logische zurückgeführt werden; auch sie sind somit in einem Quantencomputer realisierbar.

5. Der Algorithmus von Shor

Als Beispiel für die kryptographische Relevanz eines Quantencomputers wollen wir die Faktorisierung einer ganzen Zahl N betrachten.

Der dümmste Ansatz zur Faktorisierung von Hand beseht darin, daß wir x und y unabhängig voneinander die Zahlen von 2 bis N durchlaufen lassen und jeweils das Produkt xy berechnen; falls dieses gleich N ist, haben wir eine Zerlegung von N gefunden.

Für eine etwa hundertstellige Zahl N (mit deren Faktorisierung ein heutiger Computer keine größeren Schwierigkeiten hat) wären hierzu rund 10^{200} Multiplikationen notwendig, mit klassischen Computern ein Ding der Unmöglichkeit: Selbst wenn alle heutigen Computer seit Beginn des Universums daran gerechnet hätten, wäre erst ein verschwindend kleiner Bruchteil der Produkte berechnet.

Für einen Quantencomputer dagegen sind diese 10^{200} Multiplikationen überhaupt kein Problem: Da $10^{100} \approx 2^{332}$ ist, nehmen wir zwei Quantenregister x, y aus etwa 350 QBits und bringen beide in den Zustand, in dem alle Basisvektoren denselben Koeffizienten haben. Sodann berechnen wir das Produkt der beiden Registerinhalte auf reversible Weise, d.h. wir berechnen das Tripel (x, y, xy). Dieses ist in einem Zustand, in dem alle Kombinationen (x, y, xy) mit $0 \le x, y < 2^{350}$ überlagert sind, insbesondere also auch die, für die $xy = N$ ist. Der Quantencomputer kann also all diese Multiplikationen gleichzeitig durchführen.

Damit ist allerdings leider das Faktorisierungsproblem noch nicht gelöst, denn wir müssen das Tripel ja auch noch messen. Dabei kollabiert der überlagerte Zustand und wir erhalten als Ergebnis ein Tripel (x, y, xy) aus natürlichen Zahlen, über das wir keinerlei Kontrolle haben. Insbesondere ist die Wahrscheinlichkeit, daß an dritter Stelle die Zahl N steht, verschwindend gering, so daß dieser sehr einfache Ansatz definitiv nicht zum Ziel führt.

Ein Quantencomputer kann also zwar sehr viele Operationen gleichzeitig durchführen, aber dies läßt sich nur ausnutzen, wenn man das Ziel der Rechnung anschließend auch wirklich messen kann.

Deshalb geht der Algorithmus von Shor das Faktorisierungsproblem völlig anders an: Wie von Fermat bis hin zu den modernsten Siebalgorithmen immer wieder ausgenutzt wird, hat man dann eine gute Chancen, eine Zahl zu faktorisieren, wenn man sie selbst oder ein Vielfaches als Differenz zweier Quadrate darstellen kann: Ist

$$kN = x^2 - y^2 = (x - y)(x + y) ,$$

so kann man hoffen, daß $\mathrm{ggT}(x-y, N)$ und $\mathrm{ggT}(x+y, N)$ echte Teiler von N sind.

Der Algorithmus von SHOR nutzt dies aus, indem er für eine zufällig gewählte Zahl x zwischen 2 und $N-2$ ihre Ordnung modulo N berechnet, d.h. die kleinste natürliche Zahl r, für die

$$x^r \equiv 1 \bmod N$$

ist. Eine solche Zahl r muß es nicht geben; für $N=4$ und $x=2$ beispielsweise gibt es keine. Elementare zahlentheoretische Betrachtungen zeigen, daß die zu N teilerfremde Zahlen bezüglich der Multiplikation modulo N eine zyklische Gruppe bilden; genau für diese Zahlen gibt es also ein solches r, und für alle anderen ist der ggT von x und N ein echter Teiler von N.

Falls r existiert und ungerade ist, hat man Pech gehabt und beginnt noch einmal mit einem neuen x; andernfalls ist

$$(x^{r/2+1}+1)(x^{r/2}-1) \equiv 0 \bmod N\ ,$$

wir sind also genau in der obigen Situation und können ggTs berechnen. Falls dies keine echten Teiler sind, haben wir ebenfalls Pech gehabt und müssen mit einem neuen x von vorne anfangen. Da r die kleinste Zahl ist, für die x^r-1 durch N teilbar ist, passiert das genau dann, wenn $x^{r/2}+1$ durch N teilbar ist, also im Fall, daß $x^{r/2} \equiv -1 \bmod N$. Der Algorithmus ist somit genau dann nützlich, wenn dieser Fall nicht allzu oft (oder gar immer) eintritt.

Ist etwa p eine ungerade Primzahl und $N=p^n$, so kann bei geraden r die Primzahl p keinesfalls Teiler *beider* Faktoren

$$x^{r/2+1}+1 \quad \text{und} \quad x^{r/2}-1$$

sein, da sich die beiden nur um zwei unterscheiden. Also ist einer der beiden durch p^n teilbar, was natürlich nur der erste sein kann. Somit versagt der Algorithmus von SHOR in diesem Fall für *jedes* x.

Im kryptographisch besonders interessanten Fall, daß $N=pq$ Produkt zweier ungerader Primzahlen ist, sieht es aber – je nach Standpunkt leider oder zum Glück – ganz anders aus: Da modulo einer Primzahl $y \equiv \pm 1$ die beiden einzigen Lösungen der Gleichung $y^2 \equiv 1$ sind, zeigt eine einfache, wenn auch etwas langwierige Argumentation über den chinesischen Restesatz, daß für mindestens die Hälfte aller zu N primen Zahlen die Ordnung r gerade und $x^{r/2}$ nicht kongruent -1 modulo N ist. Hier liegt also die Erfolgswahrscheinlichkeit bei über 50%; wiederholt man die die Rechnung mit einem anderen x, kommt man bereits auf 75%, nach sieben zufällig gewählten Werten von x auf über 99%, nach zehn auf über 99,9%. Falls es also gelingt, r effizient zu berechnen, ist dieses Verfahren extrem gefährlich für das RSA-System.

Die Berechnung von r stellt uns vor ein ähnliches Problem wie die Berechnung eines diskreten Logarithmus, und in der Tat führt der Algorithmus von SHOR zur Bestimmung von r auch auf einen Algorithmus zur Berechnung diskreter Logarithmen; auch die Verallgemeinerung auf entsprechende Probleme für elliptische Kurven stellt keine prinzipiellen Probleme.

SHOR geht folgendermaßen vor: Für $N < 2^L$ braucht er einen Quantencomputer mit zwei Quantenregistern der Längen $2L$ beziehungsweise L. Im ersten speichert er die Überlagerung aller Zahlen a von 0 bis $2^{2L} - 1$, für das zweite berechnet er den Zustand x hoch erstes Register modulo N. Der Aufwand hierfür liegt für die klassische Berechnungsmethode durch fortgesetztes Quadrieren in der Größenordnung von L^3 Operationen.

Der Computer befindet sich dann in einem Zustand, in dem alle Paare $(a, x^a \bmod N)$ mit $0 \leq a < 2^{2L}$ überlagert sind. Wird nun der Inhalt des zweiten Registers gemessen, ist die Chance für das Messen der Zahl eins natürlich verschwindend gering; das Meßergebnis wird *irgendeine* Zahl b zwischen 0 und $N - 1$ sein.

Durch die Messung des zweiten Registers hat sich aber der Inhalt des ersten verändert: Der *gesamte* Computer ist *ein* quantenmechanisches System, das in einen Zustand gebracht wurde, in dem der Inhalt des zweiten Registers gleich x hoch dem Inhalt des ersten Registers ist. Die Messung des zweiten Registers kann an dieser Tatsache nichts ändern, und da das zweite Register nach der Messung die Zahl b enthält, kann eine Messung des ersten Registers nun nur noch ein Ergebnis a liefern, für das $x^a \bmod N = b$ ist. Ein solches Ergebnis nützt uns aber nichts, und deshalb dürfen wir das erste Register keinesfalls messen.

Da das erste Register doppelt so lang ist wie das zweite, befindet es sich immer noch in einem überlagerten Zustand, nämlich in der Überlagerung aller Zahlen $0 \leq a < 2^{2L}$, für die $x^a \equiv b \bmod N$ ist. Ist a_0 die kleinste solche Zahl, sind dies genau diejenigen Zahlen der Form $a_0 + kr$ mit $k \in \mathbb{N}_0$, die kleiner sind als 2^{2L}; hierbei ist r die gesuchte Ordnung.

Bei den Zahlen im überlagerten Zustand im Register handelt es sich also um eine periodische Folge von $m = \left[\frac{2^{2L} - a_0}{r}\right]$ Zahlen; was uns interessiert, ist die Periode r.

Die Periode eines Gitters läßt sich optisch aus dem Beugungsspektrum des Gitters bestimmen; genauso wollen wir auch hier vorgehen und nicht den *Inhalt* des ersten Registers messen, sondern so etwas wie sein *Beugungsspektrum.* Ein wesentlicher Aspekt der Quantentheorie ist schließlich, daß auch einander ausschließende Zustände eines Teilchens miteinander interferieren: Falls etwa ein Photon zwei mögliche Wege zu einem Punkt zurücklegen kann, tritt auch im Falle eines einzigen Photons Interferenz auf.

Das rechnerische Analogon zum Beugungsspektrum ist die *Quanten-*FOURIER-*Transformation.* Ersteres leitet man bekanntlich aus dem HUYGENSschen Prinzip ab, wonach jeder Punkt, in dem das Gitter durchlässig ist, Ausgangspunkt einer neuen Welle ist; diese wird beschrieben durch eine komplexe Exponentialfunktion, und das Beugungsspektrum ist gegeben durch die Summe aller dieser Exponentialfunktionen.

Ganz entsprechend ordnet die Quanten-FOURIER-Transformation eines Registers der Länge $2L$ ordnet dem Inhalt $|x\rangle$ dieses Registers (aufgefaßt als Zahl zwischen 0 und $2^{2L}-1$ oder als Überlagerung mehrerer solcher Zahlen) den Zustand

$$2^{-L}\sum_{\nu=0}^{2^{2L}-1} e^{2\pi|x\rangle\nu/2^{2L}}\,|\nu\rangle$$

zu. Man überzeugt sich leicht, daß dies eine unitäre Abbildung definiert; dies liegt einfach an der wohlbekannten Formel

$$\sum_{\nu=1}^{m} e^{k\nu/m} = \begin{cases} n & \text{falls } n|k \\ 0 & \text{sonst} \end{cases},$$

hinter der die Symmetrie der n-ten Einheitswurzeln zum Nullpunkt der komplexen Zahlenebene steckt – falls $n = m/\mathrm{ggT}(m,k)$ größer als eins ist.

Schwieriger ist es, diese Abbildung quantenmechanisch zu realisieren; dazu braucht man die sogenannte schnelle FOURIER-Transformation, die durch geschickte Gruppierung der Summanden nach dem Prinzip *Teile und herrsche* eine deutlich effizientere Berechnung der Abbildung erlaubt und deren einzelne Rechenschritte sich leicht in Quantenprozesse übersetzen lassen, deren Ergebnis jeweils nur von zwei QBits abhängt.

Diese Quanten-FOURIER-Transformation wird nun also auf das erste Register angewandt; dadurch kommt dieses in den Zustand

$$\begin{aligned} &2^{-L}\sum_{\nu=0}^{2^{2L}-1}\frac{1}{\sqrt{m}}\sum_{\mu=1}^{m} e^{2\pi i(a_0+\mu r)\nu/2^{2L}}\,|\nu\rangle \\ &= \frac{1}{2^L\sqrt{m}}\sum_{\nu=0}^{2^{2L}-1} e^{2\pi i a_0}\left(\sum_{\mu=1}^{m} e^{2\pi i r\mu\nu/2^{2L}}\right)|\nu\rangle \end{aligned} .$$

Da $m = \left[\frac{2^{2L}-a_0}{r}\right] \approx 2^{2L}/r$ ist, läßt sich die Klammer approximieren durch

$$\sum_{\mu=1}^{m} e^{2\pi i r\mu\nu/2^{2L}} \approx \sum_{\mu=1}^{m} e^{2\pi i\mu\nu/m} = \begin{cases} m & \text{falls } m|\nu \\ 0 & \text{sonst} \end{cases} .$$

Der Zustand des ersten Registers ist also ungefähr gleich

$$\frac{1}{2^L\sqrt{m}}\sum_{\nu=0}^{m-1} e^{2\pi i a_0} m\,|r\nu\rangle = \frac{e^{2\pi i a_0}\sqrt{m}}{2^L}\sum_{\nu=0}^{m-1}|r\nu\rangle \ .$$

(Durch Aufsummieren der geometrischen Reihe kann man natürlich auch leicht den genauen Wert der geometrischen Reihe berechnen.)

Falls wir ein Register messen, dessen Inhalt durch die approximative Zustandsfunktion beschrieben wird, erhalten wir mit *Sicherheit* ein Vielfaches von r als Ergebnis, wobei die verschiedenen Vielfachen allerdings gleiche Wahrscheinlichkeit haben.

Die exakte Zustandsfunktion führt zu leichten Abweichungen davon: Die Betragsmaxima der Koeffizienten liegen nun bei den Vielfachen der wirklichen Periode $2^{2L}/r$, aber weiter davon entfernt liegende Werte sind nicht mehr unmöglich, sondern nur noch unwahrscheinlich.

Wir wiederholen deshalb das Experiment mit demselben x mehrfach (eine genauere Analyse zeigt, daß etwa L Wiederholungen mit hoher Wahrscheinlichkeit genügen) und haben dann verschiedene Meßwerte für ν. Für die meisten, wenn nicht gar alle dieser Werte gibt es eine Zahl λ, so daß

$$\nu \approx \lambda\frac{2^{2L}}{r} \quad \text{oder} \quad \frac{\nu}{2^{2L}} \approx \frac{\lambda}{r}$$

ist.

In der hinteren Gleichung ist die linke Seite bekannt; von der rechten kennen wir weder Zähler noch Nenner. Wir wissen aber, daß der Nenner r

als Ordnung von x modulo N kleiner als N ist und damit sehr viel kleiner als der Nenner 2^{2L} auf der linken Seite, der ja größer als N^2 ist. Zur Bestimmung von r müssen wir also für die verschiedenen Brüche $\lambda/2^{2L}$ Approximationen finden, deren Nenner kleiner ist als N.

Die besten rationalen Approximationen einer reellen Zahl liefert deren Kettenbruchentwicklung. Was das ist und wie man es berechnet, sieht man am besten an einem Beispiel: Angenommen, wir wollen die Zahl π durch eine rationale Zahl approximieren. Als ganzzahlige Approximation können wir $[\pi] = 3$ nehmen; als erste echt gebrochene Approximation versuchen wir es mit einem Ansatz der Form

$$\pi \approx 3 + \frac{1}{a_1} \quad \text{oder} \quad a_1 \approx \frac{1}{\pi - 3}\,;$$

der Kettenbruchalgorithmus wählt

$$a_1 = \left[\frac{1}{\pi - 3}\right] = [7{,}0625\ldots] = 7\,.$$

Also ist $\pi \approx 3\frac{1}{7} = \frac{22}{7}$, eine seit langem bekannte für den kleinen Nenner sehr gute Approximation.

Als nächstbessere Approximation versuchen wir es mit

$$\pi \approx 3 + \cfrac{1}{7 + \cfrac{1}{a_2}} \quad \text{oder} \quad \frac{1}{\pi - 3} \approx 7 + \frac{1}{\pi}\,;$$

wir wählen

$$a_2 = \left[\cfrac{1}{\cfrac{1}{\pi - 3} - 7}\right] = [15{,}996599\ldots] = 15$$

und so weiter. Allgemein setzen wir also zur Approximation einer reellen Zahl x in einem nullten Schritt

$$b_0 = x \quad \text{und} \quad a_0 = [b_0]\,.$$

Im i-ten Schritt, für $i \geq 1$, testen wir zunächst, ob $a_{i-1} = b_{i-1}$ ist; wenn ja, sind wir fertig (und x war eine rationale Zahl). Andernfalls setzen wir

$$b_i = \frac{1}{b_{i-1} - a_{i-1}} \quad \text{und} \quad a_i = [b_i]\,.$$

Als n-te Konvergente des Kettenbruchs bezeichnen wir die rationale Zahl

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \cfrac{1}{\ddots\, a_n}}}}} \, .$$

Für uns interessant ist dieser Kettenbruchalgorithmus wegen des folgenden Satzes:

Ist für eine reelle Zahl x *und eine rationale Zahl* p/q

$$\left| x - \frac{p}{q} \right| < \frac{1}{2q^2} \, ,$$

so ist p/q *ein nicht weiter kürzbarer Bruch und eine Konvergente des Kettenbruchs von* x*.*

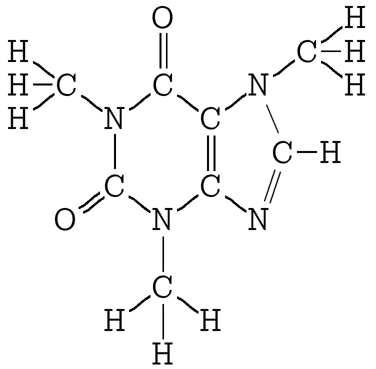
Der Kettenbruchalgorithmus liefert also die besten rationalen Approximationen mit Nennern einer vorgegebenen Größenordnung, und deshalb berechnen wir für jeden Meßwert μ alle Konvergenten des Kettenbruchs zu $\frac{\mu}{2^{2l}}$ mit Nenner kleiner N und suchen dann nach einem Nenner r, der bei möglichst vielen Meßwerten auftritt. Dieser ist ein Kandidat für die gesuchte Ordnung von x, und wir verfahren damit wie im SHORschen Algorithmus angegeben.

6. Experimentelle Realisierung

Das große Problem beim Bau von Quantencomputern ist die *Dekohärenz:* Ein überlagerter Zustand bleibt nur erhalten, wenn keinerlei Wechselwirkung mit äußeren Systemen eintritt; dies läßt sich auch mit großem Aufwand nur für sehr kurze Zeitspannen sicherstellen. SHOR entwickelte *Quantencodes,* die nach dem Vorbild klassischer fehlerkorrigierender Codes ein gewisses Maß an Dekohärenz verkraften können, aber auch damit läßt sich diese Zeitspanne nur begrenzt ausdehnen. Ein Quantencomputer auf der Basis von Photonen mit ihren extrem kurzen Dekohärenzzeiten erscheint deshalb im Augenblick jenseits jeder technischen Realisierbarkeit.

Am vielversprechendsten ist im Augenblick ein Ansatz, der mit NMR-Spektroskopie arbeitet. Bei dieser handelt es sich um eine schon seit über dreißig Jahren gebräuchliche chemische Analysemethode, die inzwischen auch für bildgebende Verfahren in der Medizintechnik eingesetzt wird. Sie beruht auf der magnetischen Resonanz gewisser Atomkerne. Zwar zeigen nur wenige Isotopen Kernresonanz, aber es handelt sich dabei um Isotope häufiger und wichtiger Atome: ^{1}H,^{2}H,^{10}B,^{11}B,^{13}C,^{14}N,^{15}N,^{17}O,^{19}F,^{29}Si und ^{31}Si.

Als Beispiel eines für einen ersten Quantencomputer geeigneten Moleküls wurde (nicht unbedingt ganz ernsthaft) das *Koffein* vorgeschlagen, das in der Tat genug solche Atome enthält, um einen kleinen „Supercomputer in der Kaffeetasse" zu realisieren:



Protonen und Neutronen in einem Atom haben jeweils Spin $\pm 1/2$, wobei sich benachbarte Protonen sowie benachbarte Neutronen jeweils antiparallel ausrichten. Bleibt dabei ein Gesamtspin übrig, richtet sich das Atom in einem umgebenden Magnetfeld der Stärke von etwa 9 bis 15 Tesla in einer von wenigen wohldefinierten Richtungen aus, zwischen denen es durch Aufnahme beziehungsweise Abgabe von Strahlungsquanten wechseln kann. Mit geeigneten Radiowellen läßt sich also zwischen verschiedenen Zuständen hin- und herschalten; durch Aufnahme eines Resonanzspektrums lassen sich die (über viele Atome gemittelten) Zustände ablesen. Dadurch, daß man hier nicht mit Quantenzuständen einzelner Partikel arbeitet, sondern über viele Partikel mittelt, wird auch das Problem der Dekohärenz entschärft. Man rechnet damit, daß ein solcher Computer schon bald die Zahl fünfzehn faktorisieren kann.

QBits in einzelnen Atomen benutzt die um 1995 in Innsbruck konzipierte *Ionenfalle.* Hier werden Ionen durch elektromagnetische Felder in einer linearen Anordnung gehalten; die QBits werden kodiert durch Anregungszustände der Ionen (von denen jedes durch einen eigenen Laser kontrolliert wird) und der Gitterschwingungen (Phononen) zwischen den einzelnen Ionen. Überlagerte Zustände, die mehrere QBits umfassen, werden über die Vibrationsenergie ihres Schwerpunkts kontrolliert.

Am National Institut of Standards in Boulder, Colorado, wurde so die kontrollierte Negation implementiert; Faktorisierung kleiner Zahlen erscheint durch Weiterentwicklung und Vergrößerung der Apparatur möglich, allerdings müssen dazu noch einige technische Probleme gelöst werden.

Für weiteres und insbesondere auch andere physikalische Ansätze sei auf die zitierte Literatur verwiesen.

7. Literaturhinweise

Auch hier sind als erstes die beiden schon im vorigen Kapitel erwähnten und kurz charakterisierten Bücher

MICHAEL BROOKS[Hrsg.]: *Quantum computing and communications,* Springer, 1999 *und*

COLIN P. WILLIAMS, SCOTT H. CLEARWATER: *Explorations in quantum computing,* Springer Telos, 1997

zu nennen. Ein Buch, das speziell auf die algorithmische Seite Bezug nimmt (oft eher anhand von Beispielen als einer klaren Darstellung des Algorithmus) ist

ARTHUR O. PITTENGER: *An introduction to quantum computing algorithms,* Birkhäuser, Boston, 1999

Die Originalarbeit zum Algorithmus von SHOR wurde kürzlich in überarbeiteter Form nachgedruckt:

PETER W. SHOR: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41** (1999), S. 303–332

Einen weiteren Übersichtsartikel, in dem auch auf das hier nicht behandelte Problem der Fehlerkorrektur in Quantencomputern eingegangen wird, ist SHORs Vortrag auf dem internationalen Mathematikerkongreß 1998 in Berlin:

PETER W. SHOR: Quantum Computing, *Proceedings of the international congress of mathematicians Berlin 1998, Documenta Mathematica, Extra volume ICM* 1998 · I, DEUTSCHE MATHEMATIKER-VEREINIGUNG, 1998, S. 467–486 *oder*
www.mathematik.uni-bielefeld.de/documenta/xvol-icm/00/00.html

Eine Darstellung des Kettenbruchalgorithmus für Schüler, die ab etwa Klasse elf lesbar sein sollte, ist

N.M. BESKIN: *Fascinating Fractions,* Mir, Moskau, 1986.

IV. DNS-Computer

Alles Leben beruht auf der Informationsverarbeitung in den Zellen eines Organismus. Diese arbeitet mit komplexen chemischen Reaktionszykeln, die bei weitem noch nicht alle verstanden sind. Das wichtigste Speichermedium ist die Desoxyribonucleinsäure, kurz DNS, der Zelle.

Die Informationsdichte dort ist ungeheuer hoch: In trockenem Zustand benötigt ein Bit gerade einmal ein Volumen von $1\,\mathrm{nm}^3$, in der Zelle etwa $100\,\mathrm{nm}^3$. Für einen Kubikzentimeter trockener DNS kommt man somit auf eine Speicherkapazität von 10^{21} Bit, das ist eine achtel Billion Gigabyte; für DNS in der Zelle kommt man mit einem Kubikzentimeter immerhin noch auch 125 Milliarden Gigabyte. In einem Computer faßt ein Speicherriegel der Größe $15\,\mathrm{cm} \times 3\,\mathrm{cm} \times 1\,\mathrm{cm}$ gerade einmal ein halbes Gigabyte, und auch bei CDs braucht man größenordnungsmäßig fast eine Billion Stück, um die in einem Kubikzentimeter DNS enthaltene Information zu speichern.

Hinzu kommt, daß die Information in der DNS massiv parallel verarbeitet werden kann mit einem Energieaufwand pro Operation, der ungefähr zehn Größenordnungen unter dem heutiger Supercomputer liegt.

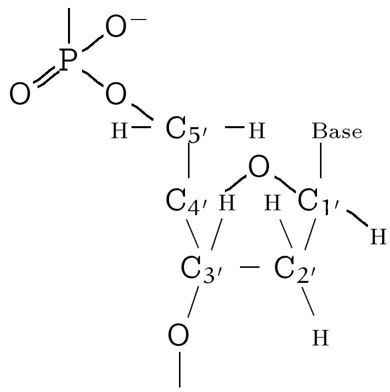
Wenn man dieses Potential für klassische Probleme der wissenschaftlichen oder gewerblichen Informationsverarbeitung nutzbar machen könnte, wäre dies eine Revolution, die auch die Kryptographie massiv verändern würde.

Seit 1994 gibt es einen ersten Hinweis darauf, daß dies vielleicht nicht völlig unrealistisch ist: Damals löste der Mathematiker und Informatiker Leonard M. Adleman (der hier bereits im Zusammenhang mit dem RSA-Verfahren erwähnt wurde) in einem molekularbiologischen Labor ein (ziemlich einfaches) graphentheoretisches Problem mit Hilfe von DNS.

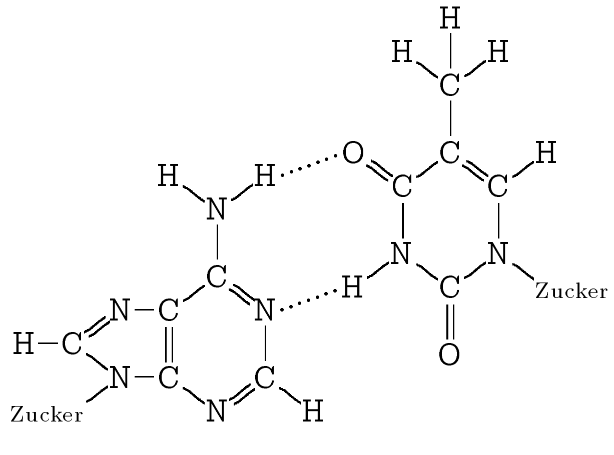
Um seine Vorgehensweise und das Potential der Methode zu verstehen, müssen wir zunächst kurz den Aufbau der DNS betrachten.

1. Die Desoxyribonucleinsäure

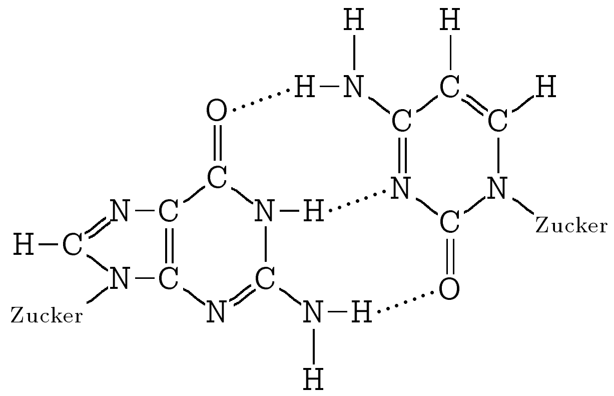
Die Desoxyribonucleinsäure hat ihren Namen vom hier abgebildeten Zuckermolekül Desoxyribose, aus dem ihr Gerüst aufgebaut ist.



Die fünf Kohlenstoffatome werden mit $1'$ bis $5'$ bezeichnet; in der Abbildung ist dies als Index eingetragen. An der Phosphatgruppe von $5'$ und der Hydroxygruppe von $3'$ fehlt jeweils ein Wasserstoffatom: Hier werden die Zucker zu einer Kette zusammengesetzt, wobei stets auf die $5'$-Phosphatgruppe eine $3'$-Hydroxygruppe folgt. Am Kohlenstoffatom $1'$ steht *Base;* hier wird eine jener vier Basen eingebaut, die die eigentlichen Informationsträger der DNS sind: Adenin, Guanin, Thymin und Cytosin.



Adenin Thymin



Guanin Cytosin

Wie man an den Abbildungen sieht, können sich zwischen Adenin und Thymin sowie zwischen Guanin und Cytosin die gestrichelt eingezeichneten Wasserstoffbrücken ausbilden; diese sorgen dafür, daß DNS nur selten in einzelnen Strängen vorkommt: Meist kombinieren sich zwei Stränge zu einem Doppelstrang, wobei jedem Adenin ein Thymin und jedem Guanin ein Cytosin gegenübersteht. Man bezeichnet diese Basen deshalb als zueinander WATSON-CRICK-komplementär.

Aus geometrischen Gründen hat ein DNS-Doppelstrang die berühmte Form der Doppelhelix, die wiederum selbst weitere geometrische Strukturen bildet. Für die Informationsverarbeitung in der Zelle sind alle diese geometrischen Strukturen sehr wichtig; die entsprechenden Mechanismen sind aber noch nicht so gut verstanden, daß man sie auch in Laborexperimenten ausnutzen könnte, so daß diese Strukturen für dieses Seminar irrelevant sind.

2. Die Polymerase-Kettenreaktion

Ein wesentlicher Bestandteil des molekularen Rechnens ist der Aufbau von DNS-Sequenzen sowie die Vervielfältigung von Sequenzen mit erwünschten Eigenschaften. Das wichtigste Hilfsmittel dafür ist die 1987 von KARY B. MULLIS entwickelte Polymerase-Kettenreaktion.

Polymerasen sind Enzyme, die in der Zelle die Duplizierung der Information aus DNS-Strängen steuern; sie sind je nach Lebewesen verschieden, und auch innerhalb derselben Zelle gibt es oft mehrere Polymerasen mit verschiedenen Aufgaben. Das besonders gut erforschte Bakterium *Escherichia coli* etwa enthält drei DNS-Polymerasen; die am stärksten vertretene DNS-Polymerase I ist eine Kette mit 928 Aminosäuren. Es handelt sich hier also um sehr komplexe Enzyme, die deshalb auch heute noch nicht synthetisiert werden, sondern auch für die Laborarbeit von Bakterien produziert werden.

Polymerasen werden zusammen mit zwei sogenannten *Primern* eingesetzt, d.h. mit zwei Folgen von DNS-Basen; diese sorgen dafür, daß selektiv ein DNS-Strang produziert wird, der WATSON-CRICK-komplementär ist zum Stück zwischen den beiden Primern auf einem vorhandenen DNS-Strang. Insbesondere muß DNS also einsträngig vorliegen, bevor eine Polymerase aktiv werden kann. In der lebenden Zelle sorgen Enzyme dafür, daß benötigte Teilstränge zur Verfügung stehen; im Labor geht man brutaler vor und teilt den ganzen Doppelstrang durch Erhitzen auf eine Temperatur von $90° - 95°$ C.

Da diese Temperatur praktisch alle Enzyme zerstört, arbeitet man heute nicht mehr mit der Polymerase von *Escherichia coli,* sondern meist mit der sogenannten Taq-Polymerase des Bakteriums *Thermus aquaticus,* das in heißem Wasser lebt und daher auch eine hitzebeständige Polymerase hat, die ihre optimale katalytische Wirkung bei etwa 75° C entfaltet.

Zur Vervielfältigung von DNS geht man folgendermaßen vor:

Man gibt die DNS zusammen mit den beiden Primern, der Taq-Polymerase und den vier Nucleotiden, aus denen die DNS aufgebaut ist, in ein Gefäß und denaturiert zunächst die DNS durch Erhitzen, d.h. man zerlegt sie in ihre Einzelstränge. Sodann kühlt man ab auf etwa 55° C; dies führt dazu, daß sich die Primer an die passenden Stellen der DNS-Stränge angliedern. Eine Erhöhung der Temperatur auf etwa 75° C aktiviert die Polymerase, die nun dafür sorgt, daß (ausgehend von den $3'$-Enden der Primer) WATSON-CRICK-komplementäre Nucleotide an die DNS-Stränge angelegt werden.

Damit hat man die gewünschten DNS-Sequenzen verdoppelt, was im allgemeinen nicht ausreicht; deshalb wird das Verfahren etwa 25–35 Mal wiederholt, was zu einer etwa millionenfachen Vermehrung führt. Das Verfahren kann in sogenannten Thermocyclern automatisch durchgeführt werden, da es nur auf die zyklische Temperatursteuerung ankommt.

3. Adlemans Experiment

Gegeben seien eine gewisse Anzahl von Städten sowie Straßen, die Städte gewissen anderen Städten verbinden. Man finde eine Straßenverbindung, die eine vorgegebene Stadt mit einem vorgegebenen Ziel so verbindet, daß jede Stadt genau einmal besucht wird.

Dies ist eine von vielen Varianten eines graphentheoretischen Problems, das für große Anzahlen von Städten und Verbindungsstraßen rechnerisch nur mit großem Aufwand gelöst werden kann. (Es ist NP-vollständig.) ADLEMAN beschränkte sich allerdings auf nur sieben Städte und fand heraus, daß der durchschnittliche menschliche Betrachter etwa 54 Sekunden braucht um sein spezielles Problem zu lösen. Interessanter ist aber die Art und Weise, wie er diese Lösung statt durch Hinschauen durch sieben Tage Arbeit im Labor produzierte.

Die Grundidee ist folgende: Jede Stadt wird durch eine Folge von zwanzig DNS-Basen kodiert; als Lösung soll jene Folge von 140 DNS-Basen produziert werden, die den (in ADLEMANs Beispiel einzigen) Lösungsweg angibt.

Die Herstellung kurzer DNS-Sequenzen mit vorgegebener Basenfolge gehört heutzutage zu den Standardtechniken der Molekularbiologie; große Labors haben Automaten, die solche Sequenzen (über die Polymerase-Kettenreaktion) erzeugen, kleinere kaufen sie von den großen: Für 25$ kann man innerhalb von wenigen Tagen ein Reagenzglas bekommen, das etwa 10^{18} DNS-Stränge mit einer vorgegebenen Folge von zwanzig Basen enthält.

Für jede der sieben Städte wurde also ein solches Reagenzglas benötigt, dazu aber auch noch für jede Straße zwischen zwei Städten. Natürlich müssen die Basenfolgen für Städte und für Straßen aufeinander abgestimmt sein: ADLEMAN faßte die Basenfolge für die i-te Stadt auf als ein Paar (x_i, y_i) aus zwei Basenfolgen der Länge zehn, und wenn es eine Straßenverbindung zwischen i-ter und j-ter Stadt gibt, kodierte er diese durch die Basenfolge $(\overline{y_i}, \overline{x_j})$, wobei die Überstreichung hier für das WATSON-CRICK-Komplement stehen soll.

Der erste Schritt der eigentlichen Berechnung bestand darin, alle diese Sequenzen (jeweils etwa 10^{14} Moleküle) in Wasser aufzulösen und Ligase dazuzugeben. (Ligasen sind Enzyme, die kurze DNS-Sequenzen zu langen zusammenfügen.) Dazu kommt noch etwas Puffer und ähnliches, und ungefähr eine Sekunde später sind im Reagenzglas viele doppelsträngige DNS-Sequenzen zu finden, die möglichen Wegen durch den Graph entsprechen.

Unter diesen Sequenzen sind mit praktisch 100%-iger Sicherheit auch solche, die der Lösung des Problems entsprechen; das Problem besteht genau wie bei den überlagerten Zuständen in Quantencomputern darin, diese herauszufiltern. Ein Vorteil gegenüber Quantencomputern ist allerdings, daß man beim molekularen Rechnen durch Messungen üblicherweise nichts zerstört.

Im zweiten Schritt ging es darum, Sequenzen zu finden, die zu Wegen mit dem Richtigen Anfangs- und Zielort gehören. Finden bedeutete in diesem Fall, daß diese Sequenzen stark vermehrt werden sollten, und das ist ein klarer Fall für die Polymerase-Kettenreaktion: Man nehme die WATSON-CRICK-Komplemente des Start- und des Zielorts als Primer und lasse die Reaktion laufen. Danach sind alle Sequenzen, bei denen Anfangs- und Zielort stimmen, etwa millionenfach verstärkt; die, bei denen nur einer der beiden Orte stimmt, etwa dreißigfach, und der Rest überhaupt nicht. Entnimmt man also eine kleine Probe zur Weiterverarbeitung, so enthält diese kaum noch Sequenzen, bei denen einer der beiden Orte falsch ist.

Im dritten Schritt ging es darum, alle Sequenzen auszusondern, die eine falsche Länge haben. Ein Weg, der jede Stadt genau einmal berührt, entspricht einer Folge von sieben Städten und damit einem DNS-Strang der Länge 140. Zur Isolation dieser Stränge führt eine Art chromatographisches Verfahren, die *Gel-Elektrophorese.*

Hierbei wird ausgenutzt, daß die betrachteten Moleküle negativ geladen sind (man achte auf das O^- in der Desoxyribose). Bringt man die Lösung also auf einen Gel auf, meist Agarose oder Polyacrylamid, und legt ein elektrisches Feld an, so hängt die Wanderungsgeschwindigkeit ab sowohl von der elektrischen Ladung als auch der Molekülgröse und (in geringerem Ausmaß) einigen anderen Gegebenheiten. Die Methode ist aber jedenfalls trennscharf genug, um Sequenzen, deren Länge sich um zwanzig Nucleotide unterscheidet, zuverlässig zu trennen; nachdem die Moleküle einige Zeit gewandert sind, entnimmt man die aus der 140er-Region und verwirft den Rest.

Unter den noch verbleibenden Sequenzen befinden sich immer noch zahlreiche Nichtlösungen, denn eine Folge von sieben Städten erhält man auch, wenn man einige Städte ausläßt und andere dafür mehrfach besucht. Es muß also entweder sichergestellt werden, daß keine Stadt mehrfach besucht wurde, was mit molekularbiologischen Methoden sehr schwer sein dürfte, oder aber, daß jede Stadt mindestens einmal besucht wurde.

Letzteres ist für den Anfangs- und den Zielort bereits bekannt; bleiben also noch fünf Städte. Für diese verwendete ADLEMAN eine Kombination aus molekularbiologischer und physikalischer Vorgehensweise: Um diejenigen Moleküle auszusondern, die die Sequenz für eine gegebene Stadt nicht enthalten, heftete er WATSON-CRICK-Komplemente dieser Stadt an kleine Eisenkügelchen mit einem Durchmesser von etwa $1\,\mu m$, gab dies zu der denaturierten (d.h. wieder durch Erhitzen in Einzelstränge zerlegten) noch verbliebenen Lösung und lies die Stränge sich wieder kombinieren. Dabei paarten sich die Sequenzen an Eisenkugeln mit Teilsequenzen jener

Moleküle, die einen Weg durch die betrachtete Stadt beschreiben. Die so entstandenen DNS-Sequenzen waren also mit Eisenkugeln verbunden und konnten so mit einem Magneten am Rand des Reagenzglases festgehalten werden, während der Rest des Glases ausgeschüttet wurde.

Danach wurde frisches Wasser zugegeben und das ganze wieder erhitzt zur Denaturierung; jetzt aber wurde der Magnet an das erhitzte Reagenzglas gehalten, so daß der die Sequenzen mit Eisenkügelchen festhielt. Der Rest wurde umgegossen in ein anderes Reagenzglas, wo dann das gleiche Spiel für die nächste Stadt wiederholt werden konnte, bis alle fünf Städte abgearbeitet waren.

Falls danach noch Moleküle übrig waren, konnte es sich nur um Lösungen handeln. Da es aber wahrscheinlich nur noch relativ wenige waren, vermehrte sie ADLEMAN zunächst durch eine Polymerase-Kettenreaktion und überzeugte sich durch Gel-Elektrophorese davon, daß seine Lösung wirklich Sequenzen der Länge 140 enthielt. Zur vollständigen Lösung des Problems mußten diese dann nur noch analysiert werden.

Auch hierzu dient wieder die Polymerase-Kettenreaktion in Verbindung mit Gel-Elektrophorese: Für jeden Zwischenort führte ADLEMAN eine Polymerase-Kettenreaktion durch, wobei er als Primer die WATSON-CRICK-Komplemente von Anfangsstadt und Zwischenort nahm. Dadurch wurde die Teilsequenz bis zu diesem Zwischenort stark vermehrt, und durch Gel-Elektrophorese läßt sich feststellen, wie lange sie ist. Diese Länge, geteilt durch zwanzig, ist die Position der Stadt im Lösungsweg.

4. Wie geht es weiter?

ADLEMAN hat mit seinem Experiment zwar kein neues Problem gelöst, aber er hat gezeigt, daß molekularbiologische Verfahren zumindest grundsätzlich zur Lösung rechnerischer Probleme benutzt werden können – bis zu ihrem effizienten Einsatz ist es sicherlich noch ein langer Weg.

Das durchgeführte Experiment überprüft mit brutaler Gewalt (fast) *alle* möglichen Wege durch den Graphen, was schon bei 200 Städten eine DNS-Menge verlangen würde, die mehr wiegt als die Erde. Mit klassischen Computern und den besten derzeit bekannten Algorithmen lassen sich dagegen auch Probleme mit einigen Tausend Städten behandeln.

Auch der letzte Schritt des Experiments funktionierte nur deshalb, weil das Problem eine eindeutige Lösung hatte. Dies ist allerdings kein großes Problem, denn alternative Methoden zur Bestimmung der Basenfolge in einem DNS-Strang gibt es natürlich: Schließlich wurde inzwischen sogar das gesamte menschliche Genom entschlüsselt.

Um ein Gefühl für die mögliche Relevanz des molekularen Rechnens in der Zukunft zu bekommen, insbesondere auch in Bezug auf die Kryptologie, ist es vielleicht ganz nützlich, zum Vergleich ein völlig anderes Thema zu betrachten, die bisherige Geschichte der Faktorisierung ganzer Zahlen.

Das einfachste Verfahren zur Faktorisierung einer ganzen Zahl N ist das systematische Durchprobieren aller Zahlen $d \leq \sqrt{N}$; dieser Algorithmus ist vergleichbar mit ADLEMANs oben beschriebener Methode.

Wie wir bei der Diskussion von SHORs Algorithmus gesehen haben, liefert der Ansatz von FERMAT ein alternatives Verfahren; berechnet man, wie es FERMAT tat, für $a = 1, 2, 3, \ldots$ systematisch die Zahlen $N + a^2$ in der Hoffnung, darunter ein Quadrat zu finden, so lassen sich zumindest Zahlen mit zwei nahe beieinanderliegenden Faktoren deutlich schneller zerlegen.

Das neunzehnte Jahrhundert war von der Mechanisierung geprägt; in der ersten Hälfte des zwanzigsten Jahrhunderts erreichte diese auch die Zahlentheorie, als D.H. LEHMER eine Siebversion der FERMAT-Methode mit Zahnrädern und Fahrradketten zur Faktorisierung verwendete.

Bei den elektrischen und elektronischen Rechner griff die neue Technik schneller auf die Zahlentheorie über: D.H. LEHMER hatte bereits auf den ersten Computern seiner Universität Hintergrundprogramme laufen, die sich mit der Faktorisierung von Zahlen beschäftigten, wenn es sonst nichts zu tun gab.

In den Siebzigerjahren, als Computer leistungsfähig genug waren, um in Zahlbereiche vorzudringen, die vorher jenseits praktisch durchführbarer Rechnungen lagen, war wieder die Mathematik gefragt, die in den letzten dreißig Jahren immer neue Faktorisierungsalgorithmen entwickelte. Diese mußten auf den jeweils aktuellen Computern implementiert werden, wobei in den Achtzigerjahren vor allem die CRAY-Computer eine sehr große Rolle spielten und aufgrund ihrer Pipeline-Architektur zur Entwicklung und Optimierung neuer Programmiertechniken zwangen.

In den Neunzigerjahren brachte das Internet die Möglichkeit zum verteilten Rechnen; 1994 wurde dadurch jene berühmte 129-stellige Zahl faktorisiert, die 1978 bei der Vorstellung des RSA-Systems als Beispiel diente und von der man damals überzeugt war, daß sie auch in hundert Jahren noch sicher sei. (Heute hält das Bundesamt für Sicherheit in der Informationstechnik Zahlen mit mindestens 1024 Bit für sicher; ab 2005 verlangt es die doppelte Länge.)

Für Fortschritte bei der Faktorisierung waren also jeweils drei Aspekte maßgeblich:

- Bessere mathematische Algorithmen
- Bessere Maschinen
- Besseres Verständnis des Umgangs mit diesen Maschinen.

Auch die weitere Entwicklung des molekularen Rechnens wird wohl von drei ähnlichen Aspekten abhängen.

Sechs Jahre nach ADLEMANs Experiment kann man versuchen, zumindest ein bißchen über jeden dieser drei Aspekte zu spekulieren.

Die weitaus größte Zahl von Publikationen im Umkreis der DNS-Computer beschäftigt sich mit theoretischen Algorithmen sowie mit abstrakten Maschinen und formalen Sprachen zur Modellierung des molekularen Rechnens. Viele dieser Arbeiten kommen von theoretischen Informatikern ohne Chemieausbildung und dürften wohl selbst innerhalb der Informatik schon in wenigen Jahren vergessen sein; bis zu einer Erprobung im Labor dürften es kurzfristig auch vom verbleibenden Rest kaum eines bringen. Langfristig allerdings könnten einige wenige dieser Ansätze durchaus zu brauchbaren Verfahren weiterentwickelt werden.

Von besseren *Maschinen* dürfte der Fortschritt des molekularen Rechnens angesichts der hochentwickelten verfügbaren chemischen Labortechnik in den nächsten Jahren wohl kaum abhängen; vielmehr geht es darum, zunächst zu erforschen, welche primitiven Grundoperationen wie durchgeführt werden können, um effizient molekular zu rechnen. Mit dieser Frage beschäftigen sich eine ganze Reihe molekularbiologischer Laboratorien, die beispielsweise mit an Oberflächen angehefteten Molekülen und ähnlichem arbeiten und teilweise auch schon erste Erfolge melden können: So gelang inzwischen die molekulare Addition zweier (kleiner) Binärzahlen. Im Augenblick ist noch nicht abzusehen, ob sich mehrere Grundparadigmen durchsetzen werden, oder ob die Entwicklung auf *den* DNS-Computer zusteuert.

Der dritte Aspekt, der bessere Umgang mit der Maschinerie, kann wohl erst dann zum Tragen kommen, wenn die Entwicklungen beim Algorithmenentwurf und bei den molekularbiologischen Ansätzen anfangen, gegeneinander zu konvergieren, wenn also die entsprechenden Experten anfangen, Notiz voneinander zu nehmen. Dies setzt einerseits voraus, daß die Labors stabile und zuverlässige Verfahren entwickeln. Vor allem aber wird ein wirklicher Fortschritt erst dann möglich sein, wenn es eine nennenswerte Anzahl von Wissenschaftlern gibt, die sowohl in der Molekularbiologie als auch in der Algorithmik zumindest eine gewisse Mindestqualifikation haben.

Der Zeitrahmen dafür hängt wesentlich davon ab, ob es gelingt, ausreichend Studenten für eine solche duale Qualifikation zu begeistern und sie ihnen auch zu ermöglichen, oder ob aber die Studenten angesichts des Booms sowohl in der Informationstechnik als auch in der Molekularbiologie vollauf mit der Qualifikation in einem dieser Gebiete zufrieden sind.

An diesem Punkt geraten die Spekulationen ins Gebiet der Politik, wo Spekulationen nur selten zu etwas vernünftigem führen; daher soll dieses Seminar besser hier enden.

4. Literaturhinweise

Die Originalarbeit von Adleman ist

Leonard M. Adleman: Molecular computation of solutions to combinatorial problems, *Science* **266** (1994), S. 1021–1024

Eine erste Diskussion, bestehend aus mehreren Leserbriefen und einer Antwort Adlemans ist in *Science* **268** (1995), S. 481–484 zu finden. Eine gut lesbare Darstellung der wesentlichen Punkte des Experiments ist

Leonard M. Adleman: Rechnen mit DNA, *Spektrum der Wissenschaft,* November 1998, S. 70–77

Einen Überblick über die weitere Entwicklung geben beispielsweise die Tagungsbände *DNA Based Computers* I-V; die ersten drei wurden von der American Mathematical Society als Bände 27 (1996), 44 und 48 (beide 1999) der *Series in Discrete Mathematics and Theoretical Computer Science* veröffentlicht, die beiden weiteren habe ich noch nicht im Druck gesehen. Ein ähnlicher Tagungsband ist

Gheorghe Răun [Hrsg:] *Computing with bio-molecules,* Springer, 1998

Eine Übersicht über die molekularbiologischen Methoden, die dem molekularen Rechnen zugrunde liegen, findet man beispielsweise in den Büchern

Bernard L. Glick, Jack J. Pasternak: *Molekulare Biotechnologie,* Spektrum Akademischer Verlag, 1995 *und*

Rolf Knippers: *Molekulare Genetik,* Thieme, ^{7}1997,

von denen das erste mit etwas anschaulicheren Darstellungen arbeitet und das zweite stärker auf den biochemischen Hintergrund eingeht. Detailierte Laboranweisungen zu den Grundtechniken findet man etwa in

Hans Günther Gassen, Gangolf Schrimpf [Hrsg.]: *Gentechnische Methoden,* Spektrum Akademischer Verlag, ^{2}1999