

Facharbeit

Kryptographie und Kryptoanalyse der Vigenère-Chiffre

von

Johannes Weitzel

Mathematik GK/12

Herr Bergmann

Schuljahr 2001/2002

18. März 2002

Inhaltsverzeichnis

1. Einführung	1
1.1. Was ist Kryptologie?	1
2. Grundlagen der Kryptologie	2
2.1. Terminologie	2
2.2. Grundlagen klassischer Chiffren	2
2.3. Der Modulo-Operator	3
3. Blaise de Vigenère	3
4. Die Vigenère-Chiffre	4
5. Kryptoanalyse der Vigenère-Chiffre	6
5.1. Der Kasiski-Test	7
5.2. Der Friedman-Test	8
5.3. Auswertung der kryptoanalytischen Tests	11
6. Fazit	12
A. Implementierungen in PASCAL	13
A.1. Die Vigenère-Chiffre	13
A.2. Die Kryptoanalyse nach Kasiski und Friedman	15
A.3. Vigenère-GUI	18
B. Häufigkeiten der Buchstaben der deutschen Sprache	18
Literatur	19

1. Einführung

Diese Facharbeit befasst sich mit der klassischen Vigenère-Chiffre, die 1586 von dem Franzosen **Blaise de Vigenère** entwickelt wurde, und deren Kryptoanalyse nach Verfahren von Kasiski und Friedman. Die Vigenère-Chiffre ist die bekannteste unter allen periodischen polyalphabetischen Algorithmen. Sie ist der Prototyp für viele Algorithmen, die professionell bis in unser Jahrhundert benutzt wurden.

1.1. Was ist Kryptologie?

Kryptologie ist ein Teilgebiet der Mathematik, das sich mit der Kryptographie (dem Verschlüsseln) und der Kryptoanalyse (dem „Knacken“ von Codes) beschäftigt. Das Ver- und Entschlüsseln von Nachrichten übt auch heute noch eine große Faszination auf Menschen aller Bevölkerungsschichten aus. Fachleute aus den Bereichen Mathematik, Informatik und Linguistik beschäftigen sich mit dieser alten Wissenschaft, die bis zur Mitte des zwanzigsten Jahrhunderts hauptsächlich militärisch angewendet wurde.

Doch heute geht Kryptographie alle Menschen an. Pläne, Patente, Verträge und andere vertrauliche Daten werden auf Rechnern gespeichert und weltweit über das Internet ausgetauscht. Ohne Kryptographie wäre es für Firmen leicht, Industriespionage zu betreiben oder für Geheimdienste, Informationen über Personen zu sammeln.

Aber auch für Privatpersonen wird Kryptographie immer wichtiger. Am Bankautomaten, in Telefonzellen, beim Mobilfunk oder beim Homebanking — überall soll Kryptographie den sicheren Datenaustausch garantieren, um die Privatsphäre zu wahren. Wichtig ist, dass die benutzen Algorithmen offen gelegt werden, denn nur so kann gewährleistet werden, dass Kryptologen sich dieser Algorithmen annehmen und dann versuchen, diese zu entschlüsseln. Sind über einen langen Zeitraum alle Attacken erfolglos, so stärkt dies das Vertrauen der Benutzer in die Sicherheit des Algorithmus. Dieses Prinzip wurde auch schon im 19. Jahrhundert von dem belgischen Kryptographen **Auguste Kerckhoffs** (*Kerckhoffs' Maxime*) gefordert:

Die Sicherheit eines Verschlüsselungsverfahrens darf nur von der Geheimhaltung des Schlüssels abhängen, nicht jedoch von der Geheimhaltung des Algorithmus.

2. Grundlagen der Kryptologie

2.1. Terminologie

Die Wissenschaft der Kryptologie enthält einige Termini, die vorab kurz dargestellt werden.

Der lesbare Text einer Nachricht (message) wird **Klartext** genannt und mit M bezeichnet. Man sagt, der Klartext wird über einem **Alphabet** gebildet. Ein Alphabet A ist eine endliche Menge von Zeichen, deren Mächtigkeit mit $n = |A|$ dargestellt wird. Auch **Geheimtext** (chiffre) C und der **Schlüssel** (key) K sind Zeichenketten über dem gleichen Alphabet A . Beispielsweise ist „*DIESERTEXTSOLLGEHEIMBLEIBEN*“ ein Klartext über dem Alphabet $\{A, B, C, \dots, Z\}$.

Die umkehrbare **Verschlüsselungsfunktion** E (encryption) wandelt den Klartext M mit Hilfe des **Schlüssels** K in den **Chiffretext** C . Die Umkehrung von E zur Wiederherstellung wird **Entschlüsselung** genannt und mit D (decryption) bezeichnet.

Nach diesen Definitionen gilt $E_K(M) = C$ und $D_K(C) = M$. Da E umkehrbar ist, gilt

$$D_K(E_K(M)) = M,$$

denn nach dem Entschlüsseln eines Chiffretextes sollte der Klartext herauskommen. Einen solchen Algorithmus nennt man **symmetrischen Algorithmus**, da zum Chiffrieren und zum Dechiffrieren immer der gleiche Schlüssel K benutzt wird. Bei **asymmetrischen Algorithmen**, die in dieser Arbeit jedoch keine Rolle spielen, wird zum Chiffrieren ein Schlüssel K_1 und zum Dechiffrieren ein anderer Schlüssel K_2 benutzt. Es gilt also:

$$\begin{aligned} E_{K_1}(M) &= C \\ D_{K_2}(C) &= M \\ D_{K_2}(E_{K_1}(M)) &= M. \end{aligned}$$

2.2. Grundlagen klassischer Chiffren

Es gibt verschiedene Methoden, die bei klassischen Chiffren (das sind Chiffren, die bis etwa 1950 entwickelt wurden) zum Einsatz kommen. Beispielsweise werden bei einer **Transpositionschiffre** die Zeichen des Klartextes vertauscht (Permutation). Sie bleiben also erhalten, sind im Chiffretext dann aber an anderen Positionen.

Bei **Substitutionschiffren** wird jedes Zeichen des Klartextes durch ein anderes ersetzt. Die Position bleibt jedoch gleich. Eine Chiffre, bei der jedes Klartextzeichen immer auf dasselbe Geheimtextzeichen abgebildet wird, ist mit Hilfe einer Tabelle wie in Anhang B sehr leicht zu entschlüsseln. Solche Chiffren, wie beispielsweise die Chiffre des römischen Kaisers Cäsar (siehe auch Kapitel 4, Fußnote 2), nennt man **monoalphabetisch**. Eine Substitutionschiffre ist **polyalphabetisch**, wenn sie nicht mehr monoalphabetisch ist. Polyalphabetisch heißt also, dass ein Klartextzeichen nicht immer auf dasselbe Geheimtextzeichen abgebildet werden muss. Ein E im Klartext kann im Geheimtext mal einem A, mal einem T oder sonst einem Zeichen des benutzten Alphabets entsprechen.

Die im Folgenden dargestellte Vigenère-Chiffre ist eine solche polyalphabetische Verschlüsselung.

2.3. Der Modulo-Operator

In der Vigenère-Chiffre sowie in vielen anderen kryptographischen Algorithmen spielt der Modulo-Operator eine wichtige Rolle. Mit dem Modulo-Operator berechnet man den Rest einer Division. Beim Dividieren einer natürlichen Zahl a durch eine andere natürliche Zahl b bleibt ein Rest $r \in \{0, 1, \dots, b - 1\}$. Zum Beispiel ist

$$23 : 3 = 7 \text{ Rest } 2 \text{ oder } 23 = 7 \cdot 3 + 2$$

$$27 : 3 = 9 \text{ Rest } 0 \text{ oder } 27 = 9 \cdot 3 + 0.$$

Definition: Seien $a, b \in \mathbb{Z}$ und sei $a = bq + r$. Dann schreibt man:

$$r = a \bmod b.$$

Zwei Zahlen $a, b \in \mathbb{Z}$ heißen **restgleich**, wenn $a \bmod n = b \bmod n$. Man schreibt $a \equiv b \bmod n$ und spricht **a ist kongruent zu b modulo n** .

3. Blaise de Vigenère

Blaise de Vigenère (1523 bis 1596) wurde in Frankreich geboren und genoss die Ausbildung eines Adligen, obwohl er diesem Stand gar nicht angehörte. Er arbeitete in verschie-

denen Berufen und wurde später Sekretär des Herzogs von Nevers. Nachdem dieser starb, wurde Vigenère von einem Gericht beauftragt, diplomatische Aufgaben in Rom zu erfüllen. Dort kam er erstmals in Berührung mit der Kryptographie. Verschiedene Kryptologen besprachen ein Buch des Arztes und Mathematikers **J.B. Porter**. Es enthielt die Beschreibung eines kryptographischen Algorithmus, der zu dieser Zeit als nicht entschlüsselbar galt. Leider war dieser Algorithmus sehr unpraktisch, denn die Tabellen zum Entschlüsseln mussten vom Sender und vom Empfänger der geheimen Nachricht immer mit sich getragen werden. Außerdem war dieser Algorithmus beim Verschlüsseln sehr fehleranfällig. Obwohl dieser Algorithmus nicht viel angewandt wurde, verdankt Porter seinem System die Bezeichnung „Vater der modernen Kryptographie“.

Nachdem Vigenère nach Paris zurückkehrte, begann er seine kryptographischen Studien, die er in seinem Buch „A Treatise on Secret Writing“¹, niederschrieb. Es enthält das Vigenère-Quadrat, welches in Kapitel 4 genauer beschrieben wird. Fachleute hielten die Vigenère-Chiffre für die größte kryptographische Erfindung seit **Julius Cäsar**² (100 bis 44 v.Chr.). Tatsächlich war Vigenères Chiffre eine wesentliche Verbesserung zu Porters System, denn dieses benötigte spezielle Tabellen, während Vigenères Chiffre lediglich ein gut merkbares Schlüsselwort und das Vigenère-Quadrat, das immer und überall zu rekonstruieren ist, benötigt.

4. Die Vigenère-Chiffre

Die Vigenère-Chiffre ist eine Verschiebechiffre, bei der jedes Zeichen im Alphabet verschoben wird, wobei der Betrag der Verschiebung von der Position des Zeichens im Text und dem Schlüsselwort abhängt. Sender und Empfänger müssen bei der Vigenère-Chiffre ein Schlüsselwort vereinbaren. Außerdem müssen beide im Besitz des unten dargestellten Vigenère-Quadrats sein. Die Buchstaben im Vigenère-Quadrat sind durchnummeriert, wobei bei 0 begonnen wird. Der Buchstabe A hat also die Nummer 0, B die Nummer 1 bis Z mit der Nummer 25.

¹sinngemäß: „Eine Abhandlung über Geheimschrift“

²Cäsars Algorithmus beruhte darauf, Buchstaben des Klartextes um eine gewissen Anzahl zu verschieben, woraus sich dann der Chiffretext ergab. Cäsar benutzte dazu den Schlüssel 3, was beispielsweise aus einem A im Klartext ein D im Chiffretext machte.

Das Vigenère-Quadrat

-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Sollte der vereinbarte Schlüssel (im Beispiel „baum“) kürzer als der Klartext sein, muss der Schlüssel so lange hintereinander geschrieben werden, bis er die Länge des Klartextes erreicht.

Beispiel:

Klartext: diesertextsollgeheimbleiben
 Schlüsselwort: baumbaumbaumbaumbaubauba
 Chiffretext: E**I**YEF**R**NQY**T**M**A**M**L**AQ**I**E**C**Y**C**L**Y**U**C**E**H**

Um dieses Ergebnis zu erhalten, geht man wie folgt vor: Der Sender verschlüsselt Buchstabe für Buchstabe. Um den Buchstaben d mit dem Schlüssel b zu kodieren, sucht der Sender einfach den Eintrag in Spalte d und Zeile b im Vigenère-Quadrat. Er findet den Buchstaben E, der den ersten Buchstaben des Chiffretext ergibt (Im Vigenère-Quadrat fett dargestellt).

Auf diese Weise wird nun fortgefahren, bis der gesamte Text verschlüsselt ist.

Zum Entschlüsseln geht man anders vor. Der Empfänger ist im Besitz des Chiffretextes und des Schlüssels. Um den ersten Buchstaben zu entschlüsseln, sucht er in der Spalte *b* den Eintrag *E*. Geht er nun ganz nach oben im Vigenère-Quadrat, findet er dort den Klartextbuchstaben *d*. So kann der Empfänger dem Chiffretext mit Hilfe des Vigenère-Quadrat Buchstabe für Buchstabe entschlüsseln.

Mathematisch kann man das Verschlüsseln mit folgender Formel darstellen:

$$E(z, i) = (z + K_{i \bmod k}) \bmod n = V_{z, K_{i \bmod k}}$$

K ist in dieser Formel der Schlüssel mit der Länge *k*, dargestellt als Vektor mit den Nummern der Buchstaben (baum entspricht 1,0,20,12). *V* ist die Matrix des Vigenère-Quadrats, *z* die Nummer des zu verschlüsselnden Zeichens im Alphabet, *i* die Position von *z* im Klartext und *n* die Länge des Alphabets.

Will man im Beispiel also das *s* von „soll“ mit dem *u* verschlüsseln so setzt man ein:

$$E(z, i) = (18 + K_{11 \bmod 4}) \bmod 26 = V_{18, K_{11 \bmod 4}}$$

$11 \bmod 4$ ergibt 3, und K_3 ist der Buchstabe *u*, der im Vigenère-Quadrat den Zahlenwert 20 hat. Nun ergibt sich:

$$E(z, i) = (18 + 20) \bmod 26 = V_{18, 20}$$

$38 \bmod 26$ ergibt 12, was im Vigenère-Quadrat dem Buchstaben *M* entspricht. Nach dieser Formel kann man nun Texte mit der Vigenère-Chiffre verschlüsseln.

Eine Implementierung dieser Chiffrier-Funktion in der Programmiersprache PASCAL findet sich im Anhang A.1.

5. Kryptoanalyse der Vigenère-Chiffre

Bei der Analyse einer Vigenère-Chiffre ist das wichtigste Ziel die Bestimmung der Schlüssellänge *k*, denn wenn man diese bestimmt hat, so funktioniert der Rest der Entschlüsselung

wie bei einer gewöhnlichen Verschiebechiffre. Die Kryptoanalyse soll an Hand des folgenden Chiffretextes gezeigt werden:

TFNMJ BDCRI TVVAF SRCJI FDPIN NNMKA PELIW TTPYQ
FUIWJ SJBIX ULMGP XRZWN FDQXO PICRS ALAER NVVKJ
HRVKJ OJQIM BKBIS TZKLZ FSMVW PSWXJ SLVXJ SYIPY
FERSW VEVLN FCBHF TDMRX DYTMM IVOIM JIVJZ FIMMS
FESSR QCQDN FIBIS DFUTZ UVZWT GZMAF SJQGM OZKLY
TFAMH IKMVT CJQII BQCWY JDUXJ FZVQJ OJKLR VJAXJ
EFKLR FYZWJ JEIPX FZVIR BJKLN OV

5.1. Der Kasiski-Test

Der Kasiski-Test geht auf den preußischen Infanteriemajor **Friedrich Wilhelm Kasiski** (1805 bis 1881) zurück. Erfunden wurde dieser Test zwar 1854 von dem englischen Mathematiker **Charles Babbage** (1792 bis 1871), jedoch hat Babbage seinen Test nie veröffentlicht. Erst neun Jahre später hat Kasiski dies getan.

Der Kasiski-Test basiert darauf, dass man den Geheimtext nach Wiederholungen von Zeichenfolgen mit mindestens drei Zeichen untersucht und deren Abstand misst. Je länger die gefundenen Zeichenfolgen, desto größer die Wahrscheinlichkeit, dass der Abstand ein Vielfaches der Schlüssellänge ist. Die Erklärung hierfür ist einfach: Wiederholt sich eine Zeichenfolge im Klartext mit einem Abstand als Vielfaches der Schlüssellänge, so wird die Wiederholung (bei einer Vigenère-Chiffre) gleich codiert.

Es kann natürlich sein, dass im Chiffretext Wiederholungen auftreten, die rein zufällig sind und deren Abstand nicht das Vielfache der Schlüssellänge beträgt. Die Wahrscheinlichkeit für zufällige Wiederholungen ist aber viel kleiner als für Wiederholungen des Vielfachen der Schlüssellänge.

Durchsucht man nun den Geheimtext nach solchen Wiederholungen, misst deren Abstand und rechnet dessen Primfaktoren aus, kommt man zu folgendem Ergebnis:

Zeichenfolge	Abstand	Primfaktoren
AFS	175	5 · 5 · 7
VKJ	5	5
JOJ	145	5 · 29
JQI	125	5 · 5 · 5
BIS	80	2 · 2 · 2 · 2 · 5
ZKL	100	2 · 2 · 5 · 5
XJS	5	5
MHI	60	2 · 2 · 3 · 5
FZV	30	2 · 3 · 5
JKL	30	2 · 3 · 5
KLR	10	2 · 5

Auf den ersten Blick könnte man meinen, dass die Schlüsselwortlänge 5 sei. Allerdings kommt auch der Faktor 2 häufig vor. Ein Schlüsselwort der Länge 2 zu wählen wäre aber äußerst unklug, da das Entschlüsseln dann nicht mehr schwer ist. Trotzdem wird jetzt an Hand des Friedman-Tests versucht, die Schlüsselwortlänge genauer zu bestimmen.

5.2. Der Friedman-Test

Der Friedman-Test wurde 1925 vom bedeutenden amerikanischen Kryptologen **William Friedman** (1891 bis 1969) entwickelt. Friedman wird heute zu einem der wichtigsten Kryptologen, die jemals gelebt haben, gezählt. Auch im zweiten Weltkrieg war Friedman maßgeblich an Entschlüsselungen im Auftrag der Amerikaner beteiligt.

Beim Friedman-Test wird untersucht, mit welcher Wahrscheinlichkeit zwei willkürlich aus einem Text herausgegriffene Buchstaben gleich sind. Der **Koinzidenzindex** gibt darauf Antwort. Das ist die Wahrscheinlichkeit dafür, dass zwei zufällig gewählte Buchstaben mit beliebigem Abstand gleich sind. Ausgegangen wird von dem lateinischen Alphabet mit 26 Zeichen, wobei die Häufigkeit der Buchstaben mit n_1 für A, n_2 für B bis n_{26} für Z angegeben wird.

Die Länge n des Textes ergibt sich dann als

$$n = \sum_{i=1}^{26} n_i.$$

Die Gesamtanzahl aller der Paare aus A's ist

$$\frac{n_1(n_1 - 1)}{2}.$$

Denn für die Auswahl des ersten A's gibt es n_1 Möglichkeiten, für die Auswahl des zweiten nur noch $n_1 - 1$ Möglichkeiten.

Also ergibt sich für die Gesamtzahl A aller Paare, bei dem beide Buchstaben gleich sind:

$$A = \frac{n_1(n_1 - 1)}{2} + \frac{n_2(n_2 - 1)}{2} + \dots + \frac{n_{26}(n_{26} - 1)}{2} = \sum_{i=1}^{26} \frac{n_i(n_i - 1)}{2}$$

Die Wahrscheinlichkeit I (Friedmanscher Koinzidenzindex) dafür, dass ein beliebiges Buchstabenpaar aus zwei gleichen Buchstaben besteht, berechnet sich damit zu

$$I = \frac{\sum_{i=1}^{26} n_i(n_i - 1)}{n(n - 1)}.$$

Bei langen deutschen Texten gilt $I \approx I_d = 0,0762$, bei zufällig generierten langen Texten mit $n_1 = n_2 = \dots = n_{26}$ ist die Wahrscheinlichkeit $I_r = \frac{1}{26}$, was 3,846 Prozent entspricht. Es lässt sich erkennen, dass der Wert bei deutschen Texten deutlich höher liegt. Bei dem Beispielchiffretext ergibt sich der Wert $I_b = 0,043423$. Man sieht, dass die Verteilung der Buchstaben im Beispielgeheimtext schon weitaus „zufälliger“ als in durchschnittlichen deutschen Texten ist. So ist — wie bei monoalphabetischer Verschlüsselung üblich — ein Angriff mit einer Tabelle wie in Anhang B nicht mehr möglich. Die Wahrscheinlichkeit, zweimal denselben Buchstaben zu ziehen, wächst mit der ungleichmäßigen Verteilung der Buchstaben.

Die Idee des Friedman-Tests ist nun: Je länger das Schlüsselwort gewählt wurde, desto mehr entfernt sich der Koinzidenzindex I weg vom Maximalwert (0,0762) hin zum Minimalwert (0,03846).

Da es k Teiltex-te gibt, mit je etwa n/k Buchstaben, gibt es insgesamt

$$\frac{1}{2} \cdot \frac{n}{k} \left(\frac{n}{k} - 1 \right) \cdot k = \frac{n(n - k)}{2k}$$

Paare aus gleichen Buchstaben innerhalb der Teiltexthe und

$$n\left(n - \frac{n}{k}\right) \cdot \frac{1}{2} = \frac{n^2(k-1)}{2k}$$

Paare von Buchstaben aus verschiedenen Teiltexthen. Nun können wir (bei bekannter Schlüssellänge k) die Zahl der Paare aus gleichen Buchstaben im Text angeben als

$$\frac{n(n-k)}{2k} \cdot I_d + \frac{n^2(k-1)}{2k} \cdot I_r.$$

Dividiert man nun durch die Zahl der Paare insgesamt $n(n-1)/2$, wird die Wahrscheinlichkeit für Paare aus gleichen Buchstaben geliefert. Diese ist aber ungefähr gleich dem Koinzidenzindex I , also schreiben wir

$$I \approx \frac{n(n-k)}{n(n-1)k} \cdot I_d + \frac{n^2(k-1)}{n(n-1)k} \cdot I_r.$$

Aufgelöst nach k ergibt das die Formel

$$k \approx \frac{(I_d - I_r)n}{(n-1)I_b - I_r n + I_d}.$$

Werden nun die oben berechneten Werte eingesetzt, ergibt sich

$$k \approx \frac{(0,0762 - 0,0385) \cdot 267}{266 \cdot 0,0434 - 0,0385 \cdot 267 + 0,0762} \approx 7,5057.$$

Vergleicht man diesen Wert mit den Überlegungen des Kasiski-Tests, zeigt sich, dass der Wert des Beispiels viel näher an 5 als an 2 ist. Man kann also vermuten, dass die Schlüssellänge tatsächlich 5 beträgt.

5.3. Auswertung der kryptoanalytischen Tests

Da wir die Schlüssellänge ziemlich genau bestimmt haben, ist das komplette Entschlüsseln der geheimen Nachricht einfach, denn jeder fünfte Buchstabe des Geheimtextes wird mit demselben Buchstaben verschlüsselt, d.h. man kann hier dechiffrieren wie bei einer **mono-alphabetischen** Chiffre, was diesen Vorgang stark vereinfacht. Schreiben wir nun jeweils immer den ersten, den zweiten usw. Buchstaben des Geheimtextes untereinander. Es ergibt sich folgende Tabelle:

1. TBTSFNPTFSUXFPANHOBTFPSSFVFTDIJFFQFDUGSOTICBJFOVEFJFBO
2. FDVRDNETUJLRDILVRJKZSSLYEEDYVIEECIFVZJZFKJQDZJJFYEZJV
3. NCVCPMLPIBMZQCAVVQBKMWVIRVBMTOVMSQBUZMQKAMQCUVKAKZIVK
4. MRAJIKIYWIGWXREKKIILVXXPSLHRMIJMSDITWAGLMVIWXQLXLWPIL
5. JIFINAWQJXPNOSRJJMSZWJJYWNFXHMZSRNSZTFMYHTIYJJRJRJXRN

Mit Hilfe der Tabelle der Häufigkeitsverteilung von Buchstaben in deutschen Texten aus Anhang B lässt sich der Geheimtext jetzt entschlüsseln. In Zeile eins kommt F mit 12 Vorkommnissen am häufigsten vor, in Zeile zwei J mit 8 Vorkommnissen, in Zeile 3 V mit 8 und M mit 7, in Zeile 4 I mit 9 und in Zeile 5 J mit 10. Man kann davon ausgehen, dass diese Buchstaben dem Klartext E entsprechen, da es das im Deutschen am häufigsten vorkommende Zeichen ist. Geht man nun im Vigenère-Quadrat in der ersten Zeile zum E, geht dann nach unten zum F, ergibt sich als erster Schlüsselwortbuchstabe das B. Fährt man so nun fort und nimmt in der dritten Zeile den am zweithäufigst vorkommenden Buchstaben M, ergibt sich das Schlüsselwort BRIEF. Vollständig entschlüsselt ergibt der Chiffretext den Klartext:

SOFIEAMUNDSENWARAUFDEMHEIMWEGVONDERSCHULEDASERSTESTUECK
WARSIEMITJORUNNZUSAMMENGEANGANGENSIEHATTENSICHUEBERROBOTE
RUNTERHALTENJORUNNHIELTDASMENSCHLICHEGEGHIRNFUEREINENKOM
PLIZIERTENCOMPUTERSOFIEWARSICHNICHTSOSICHEROBSIEDAZUSTI
MMTEEINMENSCHMUSSTEDOCHMEHRSEINALSEINEMASCHINE ,

welcher dem ersten Absatz von Jostein Gaarders Bestseller „Sofies Welt“ entspricht.

Eine Implementierung der Kryptoanalyse nach Kasiski und Friedman in der Programmiersprache PASCAL findet sich im Anhang A.2.

6. Fazit

Seit hunderten von Jahren gibt es die Notwendigkeit, Informationen und Botschaften verschlüsselt und damit sicher von einem Ort zum anderen zu übermitteln. Insbesondere für Regierungs- und Kriegsinformationen war dies wichtig. Vigenère hat mit seinem Verschlüsselungskonzept die bislang bekannten und praktizierten Methoden ganz entscheidend weiter entwickelt. Die Fortschritte seiner Technik bestehen unter anderem darin, dass Sender und Empfänger keine komplizierten Verschlüsselungstabellen, sondern nur das leicht rekonstruierbare Vigenère-Quadrat, brauchten. Außerdem war die Vigenère-Chiffre bis zur Entwicklung der kryptoanalytischen Tests von Kasiski (1863) und Friedman (1925) ohne Schlüssel praktisch nicht dechiffrierbar. Sie war also mehrere hundert Jahre äußerst sicher. Selbst heute nutzen Programme wie Microsoft Money 98 oder Intuit Quickbooks eine Abwandlung der Vigenère-Chiffre, um Dateien zu verschlüsseln.

Der Bedarf nach Verschlüsselung ist im 20. Jahrhundert, vor allem seit der rasanten Einführung und Nutzung des Internets, z.B. zum Austausch von Daten oder als Geschäftsplattform, erheblich gestiegen. So verwundert es nicht, dass die Bedeutung wirksamer Verschlüsselungstechniken erheblich zugenommen hat. Denn wenn es nicht gelingt, Informationen oder Geschäftsvorgänge über das Internet sicher — das heißt vor dem unberechtigten Zugriff durch Dritte — durchzuführen, dann wird die Idee des Internets und damit die technologische und weltwirtschaftliche Entwicklung einen empfindlichen Rückschlag erleben.

A. Implementierungen in PASCAL

A.1. Die Vigenère-Chiffre

Programm zur Ver- und Entschlüsselung nach Vigenère.

```
program vigenere; {von Johannes Weitzel}

uses crt;

var eingabe, ausgabe, schluessel : string;
    wahl : char;

function ucase (str : string) : string;
var i : integer;
begin
    for i := 1 to length(str) do
        str[i] := upcase(str[i]);
    ucase := str;
end;

function schluesselanpassen(schluessel : string; len_eingabe, len_schluessel : integer) : string;
var i, j : integer;
    zwischenschluessel : string;
begin
    zwischenschluessel := '';
    i := round(len_eingabe/len_schluessel+1);
    j := 0;
    repeat
        zwischenschluessel := zwischenschluessel + schluessel;
        inc(j);
    until j >= i;
    schluesselanpassen := zwischenschluessel;
end;
```

```

function verschluesseln(klartext, schluessel : string) : string;
var i, z : integer;
    chiffretext : string;
begin
    chiffretext := '';
    for i := 1 to length(klartext) do
        begin
            z := Ord(klartext[i]) - Ord('A') + Ord(schluessel[i]) - Ord('A');
            z := z mod 26;
            z := z + Ord('A');
            chiffretext := chiffretext + Chr(z);
        end;
        verschluesseln := chiffretext;
    end;

function entschluesseln(chiffretext, schluessel : string) : string;
var i, z : integer;
    klartext : string;
begin
    klartext := '';
    for i := 1 to length(chiffretext) do
        begin
            z := Ord(chiffretext[i]) + Ord('A') - Ord(schluessel[i]) + Ord('A');
            z := z mod 26;
            z := z + Ord('A');
            klartext := klartext + Chr(z);
        end;
        entschluesseln := klartext;
    end;

begin
    clrscr; writeln('Die Vigenere-Chiffre');
    Write('Klartext bzw. Chiffretext eingeben: '); ReadLn(eingabe);
    Write('Schluessel eingeben: '); ReadLn(schluessel);
    if length(eingabe) > length(schluessel) then
        schluessel := schluesselanpassen(schluessel, length(eingabe), length(schluessel));
    Write('Verschluesseln [1] oder Entschluesseln [2] ?');
    repeat
        wahl := readkey;
    until wahl in ['1', '2'];
    if wahl = '1' then
        begin
            ausgabe := verschluesseln(ucase(eingabe), ucase(schluessel));
            writeln; writeln('Chiffretext: ', ausgabe);
        end
    else
        begin
            ausgabe := entschluesseln(ucase(eingabe), ucase(schluessel));
            writeln; writeln('Klartext: ', ausgabe);
        end;
    readln;
end.

```


A.2. Die Kryptoanalyse nach Kasiski und Friedman

Programm zur Kryptoanalyse nach Kasiski und Friedman.

```
program vig_anaylse; {von Johannes Weitzel}

uses crt;

const N = 160;
      MAX = 10000;
      I_d = 0.0762;
      I_r = 0.0385;

var eingabe : string;
    primzahlen : array [1..N] of integer;
    Buchstabenanzahl : array [1..26] of integer;
    Schluessellaenge, Koinzidenzindex : real;

function BerechneK(m : longint; Koinzidenzindex : real) : real;
begin
    BerechneK := ((I_d - I_r) * m) / ((m - 1) * Koinzidenzindex - I_r * m + I_d);
end;

function BerechneI(m : longint) : real;
var i : integer;
    FriedmannI : real;
begin
    FriedmannI := 0;
    for i := 1 to 26 do
        begin
            FriedmannI := FriedmannI + (Buchstabenanzahl[i] * (Buchstabenanzahl[i] - 1));
        end;
    FriedmannI := FriedmannI / (m * (m - 1));
    BerechneI := FriedmannI;
end;

procedure ZaehleBuchstaben(str:string);
var i:integer;
begin
    for i := 1 to length(str) do
        inc(Buchstabenanzahl[Ord(str[i])-64]);
    end;
end;
```

```

procedure Faktorisierung(zahl : longint);
var i : integer;
    test : boolean;
begin
  i := 1;
  test := true;
  while test = true do
  begin
    while (primzahlen[i] < zahl) do
    begin
      if (zahl mod primzahlen[i] = 0) then
      begin
        zahl := zahl div primzahlen[i];
        write(primzahlen[i]:8);
        while (zahl mod primzahlen[i] = 0) do
        begin
          if (zahl = primzahlen[i]) then
            break;
          zahl := zahl div primzahlen[i];
          write(primzahlen[i]:8);
        end;
        test := true;
      end;
      inc(i);
    end;
    test := false;
    write(zahl:8);
  end;
end;

```

```

procedure ErzeugePrimzahlen;
var a : array[1..MAX] of boolean;
    i, j : longint;
begin
  a[1]:=false;
  for i:=1 to MAX do
    a[i]:=true;
  for i:=2 to MAX div 2 do
    for j:=2 to MAX div i do
      a[i*j]:=false;
  j := 0;
  for i:=2 to MAX do
  begin
    if a[i] = true then
    begin
      inc(j);
      primzahlen[j] := i;
    end;
    if j >= N then break;
  end;
end;

```

```

procedure analyse (str : string);
var i, j : integer;
    such_str, vergleich_str : string;
    pos_such_str, pos_vergleich_str, abstand : integer;
    loesche_anfang : string;
    backup_str: string;
begin
    for i := 1 to (length(str)-3) do
        begin
            such_str := copy(str,i,3);
            for j := i+1 to (length(str)-3) do
                begin
                    vergleich_str := copy(str,j,3);
                    if vergleich_str = such_str then
                        begin
                            backup_str := str;
                            pos_such_str := pos(such_str,str);
                            loesche_anfang := copy(str,1,pos_such_str+2);
                            delete(str,1,pos_such_str+2);
                            pos_vergleich_str := pos(vergleich_str,str) + length(loesche_anfang);
                            abstand := pos_vergleich_str - pos_such_str;
                            write(such_str);
                            write(', Abstand: '); write(abstand:4, ', Faktoren: ');
                            faktorisierung(abstand); writeln;
                            str := backup_str;
                        end;
                    end;
                end;
            end;
        end;
    end;

begin
    clrscr;
    writeln('Programm zur Kryptoanalyse einer Vigenere-Chiffre');
    write('Chiffretext eingeben (in Grossbuchstaben): ');
    readln(ingabe);
    ZaehleBuchstaben(ingabe);
    Koinzidenzindex := BerechneI(length(ingabe));
    Schluessellaenge := BerechneK(length(ingabe),Koinzidenzindex);
    writeln; writeln('Friedman-Test'); writeln('-----');
    writeln('Koinzidenzindex I: ', Koinzidenzindex:3:6);
    writeln('Schluessellaenge k: ', Schluessellaenge:3:6);
    erzeugeprimzahlen;
    writeln; writeln('Kasiski-Test'); writeln('-----');
    analyse(ingabe);
    readln;
end.

```

A.3. Vigenère-GUI

Dieses in Delphi geschriebene Programm verbindet die Algorithmen aus Anhang A.1 und A.2 zu einer leicht bedienbaren grafischen Oberfläche. Der Quellcode, der denen aus Anhang A.1 und A.2 sehr ähnlich ist, befindet sich neben der ausführbaren Datei auf der beigelegten CD-ROM.

B. Häufigkeiten der Buchstaben der deutschen Sprache

Buchstabe	Häufigkeit (in %)	Buchstabe	Häufigkeit (in %)
a	6,51	n	9,78
b	1,89	o	2,51
c	3,06	p	0,79
d	5,08	q	0,02
e	17,40	r	7,00
f	1,66	s	7,27
g	3,01	t	6,15
h	4,76	u	4,35
i	7,55	v	0,67
j	0,27	w	1,89
k	1,21	x	0,03
l	3,44	y	0,04
m	2,53	z	1,13

Literatur

- [1] **Buchmann, Johannes.** *Einführung in die Kryptographie.* Springer-Verlag Berlin Heidelberg 1999
- [2] **Ertel, Wolfgang.** *Angewandte Kryptographie.* Fachbuchverlag Leipzig im Carl Hanser Verlag, München Wien 2001.
- [3] **Kippenhahn, Rudolf.** *Verschlüsselte Botschaften — Geheimschrift, Enigma und Chipkarte.* Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg 1999
- [4] **Kuhlmann, Gregor.** *Programmiersprache TURBO-PASCAL — Eine strukturierte Einführung.* Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg 1987
- [5] **Kuhlmann, Gregor.** *Programmiersprache TURBO-PASCAL für Fortgeschrittene — Eine strukturierte Einführung.* Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg 1988
- [6] **PC Magazin Spezial 5/98.** *Kryptographie und Netzwerksicherheit.*
- [7] **Schneier, Bruce.** *Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C.* Addison-Wesley (Deutschland) GmbH, Bonn 1996.
- [8] **Schüler DUDEN Mathematik I.** Bibliographisches Institut F.A. Brockhaus AG, Mannheim 1990
- [9] **Selke, Gisbert W..** *Kryptographie — Verfahren, Ziele, Einsatzmöglichkeiten.* O'Reilly Verlag, Köln 2000
- [10] <http://raphael.math.uic.edu/~jeremy/crypt/contrib/deepak.html>
(Entnahmedatum 1.3.2002)