Kryptologie

Von Caesar bis RSA

Inhaltsverzeichnis

1. Geschichtliches zur Kryptologie **1**
1.1. Skytale . . . 1
1.2. Steganographie . . . 1
1.3. Caesar Cipher . . . 3
1.4. Polyalphabetische Verschlüsselung . . . 3

2. Monoalphabetische Verfahren **5**
2.1. CAESAR-Cipher . . . 5
2.2. Tauschchiffre . . . 6
2.2.1. Ein praktisches Beispiel . . . 7
2.3. Das Entschlüsseln monoalphabetischer Texte . . . 8
2.3.1. Ein weiteres Beispiel . . . 8
2.4. Hilfsmittel zur Ver- & Entschlüsselung . . . 11
2.4.1. Die Muster einer Sprache . . . 11
2.4.2. Die Abhängigkeit der Häufigkeiten vom Klartext . . . 11
2.4.3. Der Wortzwischenraum . . . 12
2.4.4. Häufigkeiten von n-Grammen . . . 12
2.5. Die Abhängigkeit von der Verfassersprache . . . 13
2.5.1. Häufigkeitsgebirge . . . 13
2.5.2. Der Koinzidenzindex kappa einer Sprache . . . 14

3. Die Idee der polyalphabetischen Chiffrierung **17**
3.1. Vigenère-Chiffrierung . . . 17
3.2. Ein praktisches Beispiel . . . 18
3.3. Die Theorie der Entschlüsselung . . . 19
3.3.1. Der Kasiski-Test . . . 19
3.3.2. Der Friedmann-Test . . . 20
3.3.3. Die Bestimmung des Schlüsselwortes . . . 21
3.4. Ein weiteres Beispiel . . . 21
3.5. Vigenère knacken . . . 24

4. Die Enigma **25**
4.1. Geschichte . . . 25
4.2. Prinzip . . . 26
4.3. Aufbau . . . 27
4.4. Schlüsselraum . . . 28
4.5. Entzifferung . . . 28
4.6. Lernprogramm . . . 30

5. Data Encryption Standard **31**
5.1. Binärcodes . . . 31

5.2. Erste DES Schicht 35
5.3. F-Modul und S-Module 39
5.4. DES Schlüssel 41
5.5. Sicherheit 42
5.5.1. Bedeutung der Blocklänge 42

6. RSA **47**
6.1. Asymmetrie 47
6.2. Einwegfunktionen 47
6.3. Idee von RSA 49
6.4. RSA knacken 54
6.5. übung zu RSA mit Mathematica 55

7. Pretty Good Privacy **57**

A. Memorandum Bombes **58**

1. Geschichtliches zur Kryptologie

> Kryptologie ist die Kunst und Wissenschaft, Methoden zur Verheimlichung von Nachrichten zu entwickeln.

Häufig
wird dabei noch zwischen Kryptographie — der Wissenschaft von der Entwicklung von Kryptosystemen — und Kryptoanalyse — der Kunst des Brechens dieser Systeme — unterschieden. Die Begriffe Kryptologie und Kryptographie sind aus dem griechischen Wörtern $\kappa\varrho\psi\pi\tau\omega\sigma$ (geheim) und $\lambda\omega\gamma\omega\sigma$ (das Wort, die Lehre, der Sinn) gebildet. Die **Kryptologie** beschäftigt sich mit der Ver- und Entschlüsselung von Informationen. Dieses Thema mutet zwar recht Antik an, wird aber in unserer zeit wieder verstärkt benötigt, denkt man nur an die Sicherheit im Internet, an Chipkarten und Passwörter.

1.1. Skytale

Vor ungefähr 2500 Jahren verwendete die Regierung von Sparta eine trickreiche Methode zur übermittlung geheimer Nachrichten. Sender und Empfänger mussten beide einen sogenannten Skytale haben. Ein **Skytale** ist ein Zylinder mit einem bestimmten Durchmesser und vorgegebener Flächenzahl. Der Sender wickelte ein schmales Band aus Pergament spiralförmig um seinen Zylinder und schrieb dann der Länge nach seine Nachricht auf das Band. War nun das Band abgewickelt, konnte die Nachricht mühelos von einer Person gelesen werden, die einen Zylinder genau desselben Umfangs hatte.

1.2. Steganographie

Auch die **Steganographie**, eine Art von gedeckten Geheimschriften war früh bekannt. Diese Geheimschriften konnten entweder als unverfängliche, offen verständliche Nachricht oder in (winzigen) sichtbaren graphischen Details einer Schrift oder Zeichnung erscheinen. Letzteres bezeichnete man auch als **Semagramm**.

Die Nachricht steht im Morsecode, der aus kurzen und langen Grashalmen links von der Brücke entlang des Flusses und auf der kleinen Mauer gebildet wird (siehe Abbildung 1 auf Seite 2).

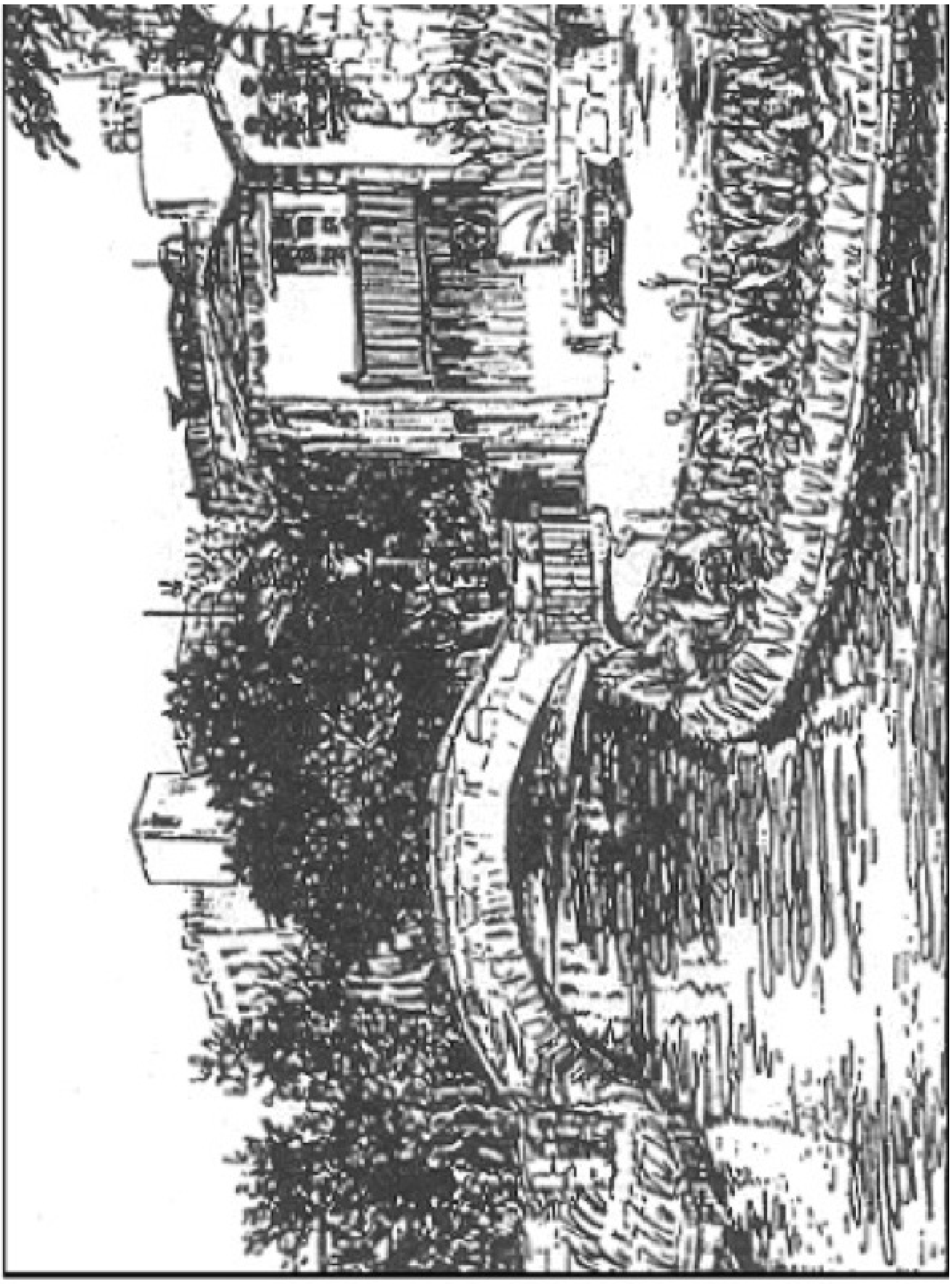


Abbildung 1: Semagramm mit Morsecode (aus Bauer)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
x	y	z																				
A	B	C																				

Tabelle 1: Caesar-Alphabet mit Translation 3

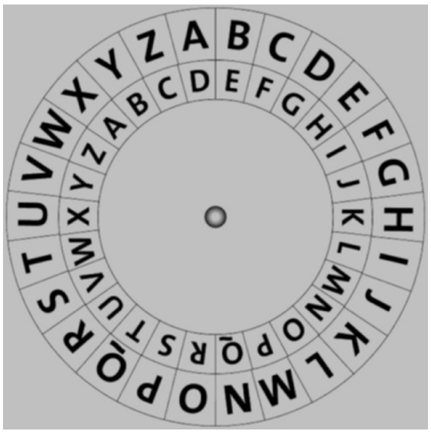


Abbildung 2: Caesar-Scheibe zum Ver- und Entschlüsseln

1.3. Caesar Cipher

Julius Caesar verwendete darüber hinaus eine spezielle Methode der monoalphabetischen Chiffrierung, den Verschiebechiffre. Er verschob die Buchstaben seines Klartextes um 3 Stellen bezüglich des Alphabetes nach links, so dass aus einem `a` ein `D` wurde:

1.4. Polyalphabetische Verschlüsselung

Als eigentlicher Begründer der Kryptologie gilt L.B. Alberti, der 1466 erstmals den polyalphabetischen Schlüssel beschrieb. Parallel zur Weiterentwicklung der Kryptologie gab es auch Fortschritte in der Berechnung von Schlüsseln. Um 1400 gelang es den Arabern, Substitutionen zu brechen. G.B. Della Porta löste erstmals einen polyalphabetischen Schlüssel. Wichtige Beiträge zur Kryptologie lieferten im 19.Jh. u.a. C. Wheatstone, F. Beaufort und Friedrich W. Kasiski.

Die polyalphabetische Chiffrierung hat jahrtausendelang seine Bedeutung erhalten und wurde von den Deutschen noch im Zweiten Weltkrieg benutzt. Allerdings wurden dabei nicht Papierstreifen gegeneinander verschoben, sondern eine Maschine verwendet, in der sich Walzen mit eingravierten Buchstaben drehten. Durch Veränderung von Schaltungen, das heisst durch Neustecken von elektrischen Kontakten, konnte man den Code, also die Grösse der einzelnen Verschiebungen, bei der Wahl eines jeden Buchstabens automatisch verändern. Der Empfänger besass eine ähnliche Maschine, „ENIGMA" genannt (ENIGMA, griechisch Rätsel, Geheimnis). Da er den jeweils verwendeten Code kannte, konnte er ihn in diese Maschine eingeben und erhielt dann durch Tippen des verschlüsselten Textes unmittelbar die entschlüsselte Botschaft. Auf deutscher Seite war man überzeugt, dass die kriegswichtigen Nachrichten vom Gegner nicht entziffert werden könnten. Man hatte sich aber getäuscht. Zwei Umstände machten es den Polen und Engländern möglich, den Code zu knacken:

- Weil die Verschlüsselung jeweils nur durch Herstellung von verschiedenen Schaltungen erfolgte, gab es eine endliche Zahl von verschiedenen Codes. Es wurden somit nicht immer wieder neue Codes verwendet, sondern nach einiger Zeit alte nochmals eingesetzt. Durch diese Wiederholung wurden Ansatzpunkte geschaffen, die Verschlüsselung zu knacken.

- Die Engländer verfügten über die ersten leistungsfähigen elektronischen Rechenmaschinen (Bomben). Dadurch konnten sie in Sekundenbruchteilen zahlreiche Zuordnungsmöglichkeiten ausprobieren, bis sie durch Zufall auf die richtige stiessen.

Den Deutschen blieb verborgen, dass die Engländer ihre Nachrichten verstehen konnten, was nicht unwesentlich zur Entscheidung des Krieges beigetragen haben soll. Zu der Kryptoanalytikergruppe der Engländer gehörte unter anderen auch A. Turing, der nicht unwesentlichen Einfluss auf die Weiterentwicklung der noch jungen Computertechnik hatte. Wesentliche Veränderung für die Kryptologie brachte das Aufkommen des Computers mit sich. Unter dem Aspekt des Datenschutzes hat das Interesse an der Kryptologie ganz erheblich zugenommen. Andererseits bietet der Computer selber die Möglichkeit, grosse Datenmengen schnell analysieren zu können, was neue Ansätze zum Brechen von Schlüsseln schafft.

2. Monoalphabetische Verfahren

Eine sehr einfache Verschlüsselung erhalten wir, indem wir jedem Buchstaben ein festes Symbol zuordnen. Diese Verfahren heissen **monoalphabetisch**. Sie sind in der Regel, steht genügend Material zur Verfügung, leicht durch Häufigkeitsbetrachtungen zu berechnen. Wesentlich schwieriger ist es, polyalphabetische Geheimtexte, das sind solche, bei denen einem Buchstaben mehrere Symbole entsprechen können, zu brechen, weil hier statistische Erwägungen nicht ohne weiteres angewendet werden können. Die klassischen Verfahren haben den Nachteil, dass sich Sender und Empfänger über den zu verwendenden Schlüssel verständigen müssen, was eine zusätzliche Unsicherheit bedeutet. Dies entfällt bei den Public-Key-Systemen, die seit 1976 entwickelt werden. Das bekannteste unter ihnen, das RSA-Verfahren (nach R. Rivest, A. Shamir und L. Adleman), verwendet die Primfaktorenzerlegung natürlicher Zahlen. Es ist nur so lange sicher, wie es keine wesentlich schnelleren Algorithmen zur Primfaktorzerlegung gibt als die heute bekannten. Daneben setzt es die Kenntnis genügend vieler grosser Primzahlen voraus.

Monoalphabetische Chiffrierung besteht darin, das Klartextalphabet zu permutieren, d.h. die Buchstabenanordnung wird vertauscht. Unter der Annahme, dass das verwendete Alphabet 26 Buchstaben besitzt (deutsch — mit ä = ae, ö = oe, ü = ue), erhalten wir also

$$26! = 403291461126605635584000000 \approx 4 \cdot 10^{26}$$

Mäglichkeiten der Anordnung der Buchstaben.

2.1. CAESAR-Cipher

Zu den einfachsten Chiffren gehört die Verschiebechiffre, die schon von Caesar verwendet wurde. Hierbei werden nur die Buchstaben in ihrer Reihenfolge verschoben. Einen solchen Geheimtext können wir einfach brechen, da für ein beliebiges Wort nur alle möglichen 26 Verschiebungen betrachtet werden müssen, um ein sinnvolles zu finden. Betrachten wir zum Beispiel `RBC`, so ergibt nur das Wort `ist` einen Sinn. Versuchen Sie nun den folgenden berühmten Satz zu dechiffrieren:

```
LFK NDP VDK XQG VLH JWH
```

Übung 1 (Caesar)**.** Werden in `Mathematica` oder `Python` erledigt.

2.2. Tauschchiffre

Eine weitere Möglichkeit bietet die **Tauschchiffre**. Hierbei wird nicht einfach das gesamte Alphabet verschoben, sondern die Buchstaben untereinander vertauscht. Mathematisch ausgedrückt heisst das: jedem Buchstabe des Klartextalphabetes wird gemäss der Reihenfolge die entsprechende natürliche Zahl zugeordnet. Multiplizieren wir den Wert eines jeden Klartextbuchstaben mit einer frei wählbaren Zahl, erhalten wir ein neues (im Allgemeinen nicht eindeutiges) Geheimtextalphabet. Soll diese Abbildung eindeutig sein, müssen wir beachten, dass die Geheimzahl und die Anzahl der Klartextbuchstaben zueinander teilerfremd sind. Für ein Alphabet mit 26 Buchstaben sind also nur die Faktoren: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 und 25 möglich. Wählen wir zum Beispiel 3 als Faktor, so entsteht das Alphabet in Tabelle 2

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
C	F	I	L	O	R	U	X	A	D	G	J	M	P	S	V	Y	B	E	H
u	v	w	x	y	z														
K	N	Q	T	W	Z														

Tabelle 2: Monoalphabetische Verschlüsselung mit Faktor 3

Übung 2 (Tauschchiffre)**.** Beschreibe die Chiffrierung mathematisch.

Natürlich können die beiden Verfahren auch miteinander kombiniert werden.

Häufig wird die Methode des Schlüsselwortes verwendet, d.h. Sender und Empfänger vereinbaren ein Schlüsselwort und einen Schlüsselbuchstaben. Dies kann z.B. das fünfte Wort in der Bibel und der zweite Buchstabe des dritten Wortes sein. Somit kann die Chiffrierung jeden Tag mit anderen Voraussetzungen begonnen werden. Zur Vereinfachung wird folgendes angenommen:

Schlüsselwort: `GEHEIMSCHRIFT` Schlüsselbuchstabe: `e`

Zur Chiffrierung werden nun die im Schlüsselwort mehrfach auftretenden Buchstaben bei Wiederholung gestrichen, wir erhalten also

`GEHIMSCRFT`.

Dann wird der Rest des Schlüsselwortes unter das Klartextalphabet geschrieben, beginnend beim Schlüsselbuchstaben. Es folgt das Auffüllen der restlichen Alphabetbuchstaben.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
W	X	Y	Z	G	E	H	I	M	S	C	R	F	T	A	B	D	J	K	L
u	v	w	x	y	z														
N	O	P	Q	U	V														

Tabelle 3: Monoalphabetische Verschlüsselung mit Schlüsselwort und Schlüsselbuchstabe

2.2.1. Ein praktisches Beispiel

Wir wollen unserem Verleger die aktuelle Version des neuen Romans schicken. Da der benutzte Weg sehr unsicher ist, soll das Stück verschlüsselt werden. Als Schlüsselwort wurde `JAMES BOND` vereinbart, der Schlüsselbuchstabe soll das `q` sein. Wir entfernen zunächst das Leerzeichen aus dem Schlüsselwort und schreiben das Schlüsselwort beginnend beim Buchstaben `q` auf. Anschliessend ergänzen wir die fehlenden Geheimtextbuchstaben, so dass keiner doppelt vorkommt:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
F	G	H	I	K	L	P	Q	R	T	U	V	W	X	Y	Z	J	A	M	E
u	v	w	x	y	z														
S	B	O	N	D	C														

Tabelle 4: Beispiel einer monoalphabetische Verschlüsselung mit Schlüsselwort `JAMES BOND` und Schlüsselbuchstabe `q`

Nun wandeln wir schrittweise den Klartext in den Geheimtext um, indem wir für den jeweiligen Klartextbuchstaben den darunter stehenden Geheimtextbuchstaben verwenden.

> Der abgeschlossene Roman
> IKA FGPKMHQVYMMKXK AYWFX
>
> Los, hierher ihr beiden!! Wollt ihr wohl hoeren?!
> VYM, QRKAQKA RQA GKRIKX!! OYVVE RQA OYQV QYKAKX?!
> Hierher sag' ich! Ja, so ist es brav!
> QRKAQKA MFP' RHQ! TF, MY RME KM GAFB!
> So - und jetzt macht ihr schoen Platz!
> MY - SXI TKECE WFHQE RQA MHQYKX ZVFEC!
> Na bitte! Und jetzt bei Fuss! Na los!
> XF GREEK! SXI TKECE GKR LSMM! XF VYM!
> Bei Fuss hab' ich gesagt!

GKR LSMM QFG' RHQ PKMFPT!
Fuu. . . „Mein Gott, Riebesehl„, stoehnt Gatti,
LSS. . . „WKRX PYTT, ARKGKMKQV„, MEYKQXR PFEER,
„kannst du nicht einmal deine Socken anziehen
„UFXXME IS XRHQE KRXWFV IKRXK MYHUKX FXCRKQKX
wie jeder andere auch??„
ORK TKIKA FXIKAK FSHJ??„

[aus STERN Hamburg; Heft 29/94 S. 74]

2.3. Das Entschlüsseln monoalphabetischer Texte

Um einen solchen Geheimtext zu entschlüsseln, müssen zwei Bedingungen erfüllt sein. Zum einen muss der Klartext in einer natürlichen Sprache verfasst worden sein, und zum zweiten ein längeres Stück des Geheimtextes vorliegen. Die Analyse des Textes beruht auf der Häufigkeitsverteilung von Buchstaben und Bigrammen in der Sprache. Für Deutsch sieht die Verteilung wie in Tabelle 5 auf Seite 9 aus.

Die Vorgehensweise zum Entschlüsseln ist folgende:
Man zählt die Häufigkeiten der Buchstaben im Geheimtext und findet so `e` und `n` und die Menge $\{r, i, t, s, a\}$. Durch Auszählen der Bigramme kann man dann `r,i,t,s,a` isolieren und schliesslich über `ch` noch c und h bestimmen, da das Bigramm `hc` fast nie vorkommt. Die Buchstaben `e,n,i,s,r,a,t,c` und `h` machen bereits rund 65% des Textes aus. Der Rest ergibt sich durch Probieren.

2.3.1. Ein weiteres Beispiel

Gegeben ist folgender monoalphabetisch verfasster Geheimtext:

```
 FWJNKYICW CAFFL NGXJMHGTK IWLLG FGMTG KYIPMGHGJFNLLGJ PMGZGJ FWR
FMHJWGTG FGMTG CWLVGT IWXG MYI HGHGT VRALU GMTHGLWNKYIL ZGTGT HMTH
GK KAPMGKA TMYIL XGKATZGJK PWK FWYIL ZGMT INTZ JWNYIL GJ MFFGJ TAYI
KA OMGR
```

1. Zählen der einzelnen Buchstaben.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
7	0	3	0	0	11	30	8	11	10	11	12	14	6	1	4	0
R	S	T	U	V	W	X	Y	Z								
3	0	15	1	2	11	3	8	5								

Der Buchstabe `G` tritt am häufigsten auf, deshalb vermuteten wir: `G`=`e`. Da `n` der

Buchstabe	Häufigkeit	Buchstabe	Häufigkeit	Bigramm	Häufigkeit
a	6.47%	n	9.84%	en	3.88%
b	1.93%	o	2.98%	er	3.75%
c	2.68%	p	0.96%	ch	2.75%
d	4.83%	q	0.02%	te	2.26%
e	17.48%	r	7.54%	de	2.00%
f	1.65%	s	6.83%	nd	1.99%
g	3.06%	t	6.13%	ei	1.88%
h	4.23%	u	4.17%	ie	1.79%
i	7.73%	v	0.94%	in	1.67%
j	0.27%	w	1.48%	es	1.52%
k	1.46%	x	0.04%		
l	3.49%	y	0.08%		
m	2.58%	z	1.14%		

Tabelle 5: Relative Deutsche Buchstabenhäufigkeiten von Mono- und Bigrammen

zweithäufigste Buchstabe ist, sehen wir, das `n` entweder `T` oder `M` sein muss. Aus der Gleichverteilung der Buchstaben `s,i,r,a,n,t` folgt, das sie `T,M,L,F,I,K,J` oder `W` sind.

$$\texttt{n} \in \{\texttt{T,M}\}$$

$$\{\texttt{s,i,r,a,t,n}\} \subset \{\texttt{T,M,L,F,I,K,J,W}\}$$

2. Zählen der Bigramme, die mit `e` beginnen, also `e`? = `G`?.

GX	GT	GM	GH	GJ	GZ	GL	GK
1	6	4	1	6	1	1	3

   Aus der Häufigkeitsverteilung der Bigramme folgt

$$\{\texttt{en,er}\} = \{\texttt{GT,GJ}\}$$

   und damit `n` = `T` und `r` = `J`. Wir suchen nun nach `ei` und `ie`, da diese mit gleicher Häufigkeit vorkommen. So finden wir `i` = `M`, damit muss aber `s` = `K` sein.

3. Zählen der Bigramme, die mit `e` enden, also ?`e`=?`G`.

NG	HG	LG	FG	TG	MG	ZG	WG	VG	XG
1	5	2	3	4	4	4	1	1	2

   Da `ie` = `MG` und `ne` = `TG` bereits feststehen, gilt

$$\{\texttt{te,de}\} \subset \{\texttt{HG,LG,FG,ZG,XG}\}$$

   Durch Vergleich mit obigen Mengen erhalten wir: $\texttt{t} \in \{\texttt{L,F}\}, \texttt{a} \in \{\texttt{L,F,I,W}\}, \texttt{d} \in \{\texttt{H,L,F,Z,X}\}$

4. Zählen der am häufigsten auftretenden Bigramme.

YI	GJ	GT	HG
8	6	6	5

   Da `IY` nicht im Text vorkommt, liegt der Schluss zu `ch` = `YI` nah.

5. Aufschreiben der gefundenen Buchstaben.

```
FWJNKYICW CAFFL NGXJMHGTK IWLLG FGMTG KYIPMGHGJFNLLGJ
m r sch     mmt  e ri ens h tte meine sch ie erm tter
PMGZGJ FWR FMHJWGTG FGMTG CWLVGT IWXG MYI HGHGT VRALU
 ie er m l m  r ene meine   t en h  e ich gegen     t
GMTHGLWNKYIL ZGTGT HMTH GK KAPMGKA TMYIL XGKATZGJK PWK
einget  scht  enen ging es s  ies   icht  es n ers  as
FWYIL ZGMT INTZ JWNYIL GJ MFFGJ TAYI KA OMGR
m cht  ein  un  r  cht er immer   ch s   ie
```

Durch einfache Tests findet man schnell die Buchstaben für `t` und `a` sowie: `t`=`L`, `a`=`W`, `u`=`N`, `w`=`P`, `g`=`H`, `m`=`F`, `b`=`X`, `d`=`Z`, `o`=`A`, `k`=`Z`, `l`=`R`, `v`=`O`, `z`=`V`, `y`=`U`.

So heisst der nun geknackte Geheimtext:

> Maruschka kommt — †brigens hatte meine Schwiegermutter wieder mal Migräne — Meine Katzen habe ich gegen Zloty eingetauscht — denen ging es sowieso nicht besonders — Was macht Dein Hund? — Raucht er immer noch so viel?

2.4. Hilfsmittel zur Ver- & Entschlüsselung

Um Kryptoanalyse betreiben zu können, müssen wir uns mit den Gesetzmässigkeiten der Sprache vertraut machen. Solche Normen hat jede Sprache und erstere können auch durch geschickte Chiffrierung nicht vollständig beseitigt werden.

2.4.1. Die Muster einer Sprache

Muster sind die Art und Weise, wie sich Buchstaben in einem Wort wiederholen. Sie werden über Ziffern ausgedrückt, wobei jeder neue Buchstabe auch eine neue Ziffer erhält, also z.B.:

$$\begin{aligned} \texttt{OTTO} &\longrightarrow 1221 \\ \texttt{NGRGUUV} &\longrightarrow 1232445 \\ \texttt{PANAMAKANAL} &\longrightarrow 12324252326 \end{aligned}$$

Solche Muster bleiben bei der monoalphabetischen Chiffrierung erhalten, d.h. enthält ein Geheimtext keine Muster, so ist er nicht durch monoalphabetischer Chiffrierung entstanden. Muster sind sehr hilfreich bei kurzen Texten.

2.4.2. Die Abhängigkeit der Häufigkeiten vom Klartext

Die Angaben über die Einzelbuchstaben schwanken und hängen ausserdem vom Genre des Textes ab. So schreibt Beutelsbacher:

> „Ein von Zitaten strotzender zoologischer Text über den Einfluss von Ozon auf die Zebras im Zentrum von Zaire wird eine andere Häufigkeitsverteilung aufweisen, als ein Traktat über die amourösen Adventüren des Balthasar Matzbach am Rande des Panamakanals.“

Bigramm engl.	Häufigkeit in %	Bigramm dt.	Häufigkeit in %
`th`	3.15	`en`	3.88
`he`	2.51	`er`	3.75
`an`	1.72	`ch`	2.75
`in`	1.69	`te`	2.26
`er`	1.54	`de`	2.00
`re`	1.48	`nd`	1.99
`on`	1.45	`ei`	1.88
`es`	1.45	`ie`	1.79
`ti`	1.28	`in`	1.67
`at`	1.24	`es`	1.52

Tabelle 6: Bigramm-Häufigkeiten deutsch und englisch

Häufigkeiten sind um so schärfer, je länger der Text ist.

2.4.3. Der Wortzwischenraum

Sicherlich kommen wir auf die Idee, den Wortzwischenraum (Space) mit zu verschlüsseln. Dies führt dann zu einem Alphabet mit 27 Buchstaben. Da der „Space“ im Deutschen nach dem `e` das häufigste „Zeichen“ und somit leicht zu enttarnen ist, lässt der professionelle Chiffrierer diesen Zwischenraum einfach weg, denn dies erschwert die Kryptoanalyse nur unwesentlich.

2.4.4. Häufigkeiten von n-Grammen

n-Gramme sind Kolonnen von n Buchstaben. Die folgende Tabelle 6 auf Seite 12 zeigt die Häufigkeiten für Bigramme im Deutschen und Englischen.

Im Deutschen kommt `ch` sehr häufig vor, aber nahezu niemals `hc`. Ferner tauchen `ei` und `ie` gleichhäufig auf.

deutsch	Häufigkeit in %	english	Häufigkeit in %
`ein`	1.22	`the`	3.53
`ich`	1.11	`ing`	1.11
`nde`	0.89	`and`	1.02
`die`	0.87	`ion`	0.75
`und`	0.87	`tio`	0.75
`der`	0.86	`ent`	0.73
`che`	0.75	`ere`	0.69

Tabelle 7: Trigramm-Häufigkeiten deutsch und englisch

deutsch	5.9	italienisch	4.5
englisch	4.5	spanisch	4.4
französisch	4.4	russisch	6.3

Tabelle 8: Wortlängen von Sprachen

In Tabelle 7 finden Sie noch die Häufigkeit einiger Trigramme.

Abschliessend eine Aufzählung häufiger Viergramme:

deutsch: `icht, keit, heit, chon, chen, cher, urch, eich, ...`

Aufschluss über den Ursprung des Textes kann übrigens auch die mittlere Wortlänge geben, oder die 10 häufigsten Wörter. Betrachten Sie dazu die Tabellen 8 und 9 auf Seite 13.

2.5. Die Abhängigkeit von der Verfassersprache

2.5.1. Häufigkeitsgebirge

Jede Sprache hat ihre eigenen Häufigkeiten. Die Diagrammdarstellungen werden **Häufigkeitsgebirge** genannt. Zwei Beispiele sind in Abbildung 3 auf Seite 15 illustriert.

deutsch	`die, der, und, den, am , in, zu, ist, dass, es`
englisch	`the, of, and, to, a, in, that, it, is, I`
französisch	`de, il, le, et, que, je, la, ne, on, les`
italienisch	`la, die, che, il, non, si, le, una, lo, in`
spanisch	`de, la, el, que, en, no, con, un, se, sa`

Tabelle 9: Zehn häufigste Wörter von Sprachen

Durch Häufigkeitsgebirge können wir bei genügend langem monoalphabetisch chiffriertem Geheimtext feststellen, in welcher Sprache er verfasst wurde.

2.5.2. Der Koinzidenzindex kappa einer Sprache

Um das κ einer Sprache zu bestimmen, schreiben wir zwei gleichlange unterschiedliche Texte untereinander und zählen alle „Spalten“ mit gleichen Buchstaben. Anschliessend teilen wir diese Zahl durch die Anzahl der Buchstaben in einer Zeile. Die mathematische Beschreibung von κ sieht wie folgt aus. Für zwei Texte gleicher Länge

$$T_x = x_1 x_2 \ldots x_n \text{ und } T_y = y_1 y_2 \ldots y_n$$

definieren wir

$$\kappa(T_x, T_y) = \sum_{i=1}^{n} \frac{\delta(x_i, y_i)}{n} \qquad \text{wobei} \qquad \delta(x_i, y_i) = \begin{cases} 1 & \text{für } x_i = y_i \\ 0 & \text{für } x_i \neq y_i \end{cases}$$

ist. Praktischer ist jedoch die Formel

$$\kappa = \frac{\sum_{i=1}^{26} n_i^2}{n(n-1)},$$

wobei n die Gesamtanzahl der Buchstaben ist und n_i die absolute Häufigkeit der einzelnen Buchstaben. Es zeigt sich, dass jede Sprache ihr eigenes κ hat; siehe dazu Tabelle 10 auf Seite 15.

Bemerkung. Kommt jeder Buchstabe (bei einem Alphabet von 26 Buchstaben) mit der gleichen Wahrscheinlichkeit von $\frac{1}{26}$ vor, so ergibt sich

$$\kappa = \frac{1}{26} \approx 3.85\%$$

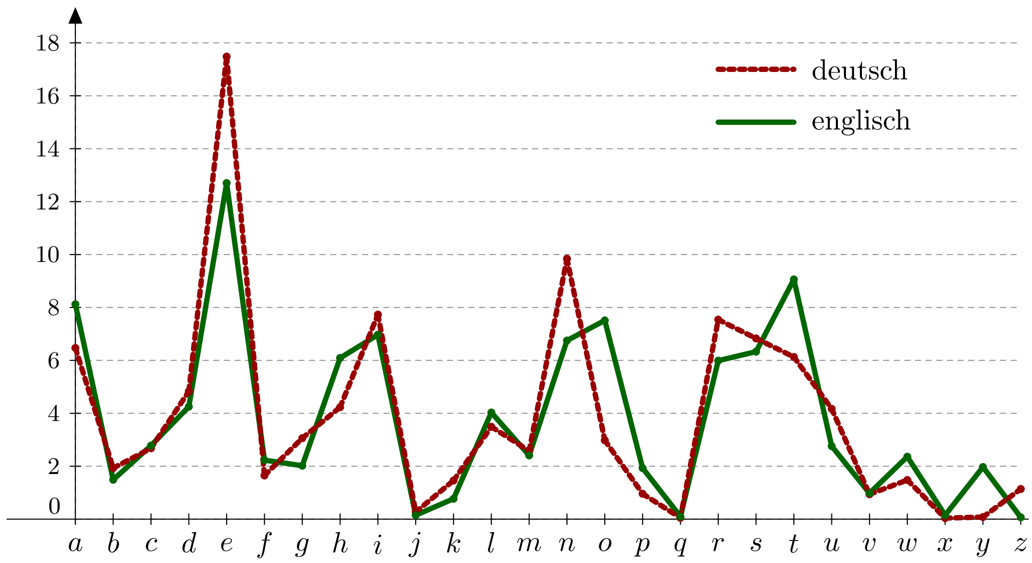


Abbildung 3: Häufigkeitsgebirge Deutsch & Englisch

deutsch	7.62%	italienisch	7.38%
englisch	6.61%	spanisch	7.75%
französisch	7.78%	russisch	5.29%
japanisch	8.19%		

Tabelle 10: Koinzidenzindex κ von Sprachen

Durch Zerschneiden des Geheimtextes in zwei Hälften und der Ermittlung des κ der beiden so entstandenen Texte kann festgestellt werden, mit welcher Sprache der Klartext verfasst wurde; oder ob es sich um eine Kunstsprache handelt.

Gegeben ist folgender monoalphabetischer Text. Versuche diese Nachricht zu knacken!

```
 IU MGF IWH UVHHWKIF DVFKIH. RWI SGOHIWRITAUI AGYYI IU UWTA GOC IWHID
UYIWH KISOIYXWTA KIDGTAY. EXVIYSXWTA NGD RIF COTAU GOU RID MGXR KIFGHHY.
IF UYOIFSI GOC RWI ITAUI SO. RG NGD RWI NFIOSVYYIF ZVD KGOD OHR CWIX
GOC RIH COTAU UWI UGA WAH GH OHR VICCHIYI WAFIH DOHR. GXU RIF COTAU
RWI UEWYSIH KWCYSGIAHI UGA, FGHHYI IF UV UTAHIXX IF NVHHYI RGZVH.
OHR MIHH IF HWTAY KIUYVFKIH UV FIHHY IF HVTA AIOYI.
```

Bemerkung. Nach Bauer gilt:

> Ein ideal für die Chiffrierung vorbereiteter Klartext ist orthographisch falsch, sprachlich knapp und stilistisch grauenhaft.

Polyalphabetische Chiffrierung bedeutet, dass das Prinzip der monoalphabetischen Chiffrierung nach gewissen Regeln ständig verändert wird. Es wird nicht der gesamte Klartext monoalphabetisch, sondern jede Buchstabengruppe mit einem anderen monoalphabetischen Schlüssel chiffriert.

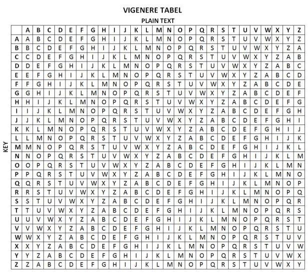
Klartextalphabet	a	b	c	d	e	...
erstes Geheimtextalphabet	H	L	W	X	D	...
zweites Geheimtextalphabet	U	L	V	W	A	...
drittes Geheimtextalphabet	N	A	R	T	I	...
viertes Geheimtextalphabet	D	Y	Z	L	M	...

Tabelle 11: Polyalphabetische Chiffrierung

V	E	N	U	S	V	E	N	U	S	V	E	N	U	S	V
p	o	l	y	a	l	p	h	a	b	e	t	i	s	c	h
K	S	Y	S	S	G	T	U	U	T	Z	X	V	M	U	C

Tabelle 12: Vigenère-Chiffrierung

3. Die Idee der polyalphabetischen Chiffrierung



VIGENERE TABEL

PLAIN TEXT

KEY

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

Eine polyalphabetische Chiffrierung kann z.B. für den Klartext `abba` so wie in Tabelle 11 auf Seite 17 erfolgen. Damit wird aus `abba` ganz einfach `HLAD`. Als Konsequenz ergibt sich, dass die Häufigkeiten und Muster verschwunden sind! Allerdings müssen Empfänger und Sender die Regeln wissen, nachdem sich die Zuordnung Klartextalphabet zu Geheimtextalphabet ändert, im allgemeinen besitzen beide 26 Alphabete und müssen dazu das Anfangswort übermitteln. Die bekannteste Methode der polyalphabetischen Chiffrierung ist die **Vigenère-Chiffrierung**. Dazu benötigt man ein sogenanntes **Vigenère-Quadrat**.

3.1. Vigenère-Chiffrierung

Neben dem Vigenère-Quadrat benötigen wir noch ein Schlüsselwort. Um Klartext zu chiffrieren, schreibt man das Schlüsselwort, hier `VENUS`, periodisch über den Klartext.

Jeder Klartextbuchstabe wird dann mit dem Geheimtextalphabet verschlüsselt, dessen erster Buchstabe im Vigenère-Quadrat über dem Klartextbuchstabe stehende Schlüssel-

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	...
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	...
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	...
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	...
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	...
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	...
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	...
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	...
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	...
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	...
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	...
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	...
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	...
								...	...	...								

Tabelle 13: Vigenère-Verschlüsselung

wortbuchstabe ist. Also für den ersten Buchstaben des Beispiels von oben heisst das: Wenn man das erste `p` aus polyalphabetisch verschlüsselt will, muss man die Geheimtextalphabetzeile nehmen, die mit `V` beginnt. Dann sucht man sich im Klartextalphabet über den Vigenère-Quadrat das `p`, geht die Spalte nach unten bis auf die Höhe von `V`. Der Geheimtextbuchstaben ist `K`. Die Dechiffrierung erfolgt analog.

3.2. Ein praktisches Beispiel

Dazu verwenden wir den Text, den abgeschlossenen Roman; das Schlüsselwort sei `JAMESBOND`. Wir schreiben zunächst das Schlüsselwort über den Text. Wir gehen in die Zeile, die mit dem Schlüsselwortbuchstaben beginnt und suchen die Spalte, die mit dem Klartextbuchstaben beginnt. Am Schnittpunkt von Zeile und Spalte steht unser Geheimbuchstabe. Wir erhalten somit den in Tabelle 14 dargestellten Geheimtext.

Übung 3 (Vigenère)**.** Die Verschlüsselung des restlichen Textes ist eine mögliche Übungsaufgabe.

JAMES	BONDJ	AMESB	ONDJA	ME
Derab	gesch	losse	neRom	an
MEDET	HSFFQ	LAWKF	BRUXM	MR

Tabelle 14: Vigenère-Verschlüsselung mit JAMESBOND

V	E	N	U	S	V	E	N	U	S	V	E	N	U	S	V	E	N	U	S
				e	i	n				e	i	n		e	i	n			
				W	D	R				Z	M	A		W	D	R			

Tabelle 15: Idee Kasiski-Test

3.3. Die Theorie der Entschlüsselung

Die Kryptoanalyse von Vigenère-Chiffrierung ist leicht, wenn das Schlüsselwort relativ kurz ist und der Geheimtext lang. Das Knacken erfolgt in zwei Schritten:

1. Bestimmen der Länge des Schlüsseltextes (Kasiski- & Friedmann-Test).
2. Bestimmen des Schlüsselwortes selbst.

3.3.1. Der Kasiski-Test

Der Test basiert auf folgender Idee: Treten im Klartext gewisse Buchstabenfolgen häufig auf (Trigramme u.ä.), so werden sie stets gleich übersetzt, wenn ein Vielfaches des Schlüsselwortes dazwischen passt. Man sucht im Geheimtext sich wiederholende Zeichenfolgen und vermutet, dass deren Abstand ein Vielfaches der Schlüsselwortlänge ist.

Beispiel 1. Wir finden sogar Zehngramme:

```
UEQPC VCKAH VNRZU RNLAO KIRVG JTDVR VRICV IDLMY
IYSBC COJQS ZNYMB VDLOK FSLMW EFRZA VIQMF JTDIH
CIFPS EBXMF FTDMH ZGNMW KAXAU VUHJH NUULS VSJIP
JCKTI VSVMZ JENZS KAHZS UIHQV IBXMF FIPLC XEQXO
CAVBV RTWMB LNGNI VRLPF VTDMH ZGNMW KRXVR QEKVR
LKDBS EIPUC EAWJS BAPMB VSZCF UEGIT LEUOS JOUOH
UAVAG ZEZIS YRHVR ZHUMF RREMW KULKV KGHAH FEUBK
LRGMB JIHLI IFWMB ZHUMP LEUWG RBHZO LCKCW THWDS
```

Folge	Abstand	Primfaktorzerlegung
KAH	128	2^7
JTD	50	$2 \cdot 5^2$
VIQM	265	$3 \cdot 5^2$
TDMHZGNMWK	90	$2 \cdot 3^2 \cdot 5$
MWK	75	$3 \cdot 5^2$

Tabelle 16: Kasiski-Test

```
ILDAG VNEMJ FRVQS VIQMU VSWMZ CTHII WGDJS XEOWS
JTKIH KEQ
```

Man bildet nun den grössten gemeinsamen Teiler der Abstandszahlen (siehe Tabelle 16). In obigen Beispiel wäre dieser ggT $= 1$, d.h. die Schlüsselwortlänge eins. Dies scheidet aber aus, weil der Klartext dann monoalphabetisch verschlüsselt sein müsste. Nimmt man aber an, dass `KAH` nur zufällig mehrmals auftritt, dann ist der ggT der Abstände gleich 5. Das legt die Schlüsselwortlänge fünf nahe. Es könnte aber auch sein, dass sich `VIQM` und `MWK` zufällig wiederholen, dann wäre $\text{ggT}(50, 90) = 10$. Also: die Schlüsselwortlänge ist höchstwahrscheinlich fünf, könnte aber auch zehn sein.

3.3.2. Der Friedmann-Test

Die Idee des Friedmanntests ist folgende: Je länger das Schlüsselwort ist, desto regelmässiger sind die Häufigkeiten verteilt, desto kleiner bzw. näher liegt κ_G an $3,85\%$. Man nimmt nun an, die Schlüsselwortlänge sei n und schreibt den Geheimtext in n Spalten. Das sieht dann wie folgt aus:

$$\begin{array}{llll} S_1 & S_2 & \dots & S_n \\ B_{n+1} & B_{n+2} & \dots & B_{2n} \\ B_{2n+1} & B_{2n+2} & \dots & B_{3n} \\ B_{3n+1} & B_{3n+2} & \dots & B_{4n} \\ \vdots & \vdots & \vdots & \vdots \end{array}$$

Die Wahrscheinlichkeit, dass zwei beliebige Buchstaben in der gleichen Spalte stehen ist

$$\frac{1}{n} \cdot \frac{1}{n} + \frac{1}{n} \cdot \frac{1}{n} + \dots + \frac{1}{n} \cdot \frac{1}{n} = n \cdot \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n}$$

Die Wahrscheinlichkeit, dass zwei beliebige Buchstaben in unterschiedlichen Spalten stehen, ist dann $1 - \frac{1}{n}$. Die Wahrscheinlichkeit, das zwei beliebige Buchstaben in einer Spalte gleich sind, ist κ_K (Klartextkappa); die Buchstaben einer Spalte sind ja dann monoalphabetisch verschlüsselt. Die Wahrscheinlichkeit, dass zwei Buchstaben aus verschiedenen Spalten gleich sind, ist $3,85\% + \varepsilon$, da über verschiedene Alphabete verschlüsselt wurde. Die Wahrscheinlichkeit, dass zwei beliebige Buchstaben des Geheimtextes gleich sind, ist dann

$$\kappa_G = \frac{1}{n}\kappa_K + (1 - \frac{1}{n}) \cdot (3.85\% + \varepsilon) = \frac{1}{n}(\kappa_K - 3.85\% - \varepsilon) + 3.85\% + \varepsilon$$

Da κ_K und κ_G leicht bestimmt werden können, lässt sich die Schlüsselwortlänge bestimmen, indem man nach n auflöst.

$$n = \frac{\kappa_K - 3.85\% - \varepsilon}{\kappa_G - 3.85\% - \varepsilon}$$

Für deutsch ist $\kappa_K = 7.62$ und daher in Prozenten

$$n = \frac{7.62 - 3.85 - \varepsilon}{\kappa_G - 3.85 - \varepsilon} = \frac{3.77 - \varepsilon}{\kappa_G - 3.85 - \varepsilon} \leq \frac{3.77}{\kappa_G - 3.85}$$

Im oben aufgeführten Beispiel ist $\kappa_G = 4.39$, woraus $n \leq 6.98$ folgt. Kasiski- und Friedmann-Test legen also die Schlüsselwortlänge $n = 5$ nahe.

3.3.3. Die Bestimmung des Schlüsselwortes

Da die Schlüsselwortlänge nun bekannt ist, kann das Schlüsselwort einfach gefunden werden. Ist die Länge des Schlüsselwortes gleich n, schreibt man den Text wie folgt:

$$\begin{array}{llll}
S_1 & S_2 & \dots & S_n \\
B_{n+1} & B_{n+2} & \dots & B_{2n} \\
B_{2n+1} & B_{2n+2} & \dots & B_{3n} \\
B_{3n+1} & B_{3n+2} & \dots & B_{4n} \\
\vdots & \vdots & \vdots & \vdots
\end{array}$$

Jede Spalte wurde monoalphabetisch verschlüsselt. Es genügt also, das `e` zu finden, um herauszukriegen mit welchem Alphabet die entsprechende Spalte verschlüsselt wurde. Man entnimmt diese `e`s der Tabelle.

Das Schlüsselwort lautet also `RADIO`. Nun kann der gesamte Geheimtext dechiffriert werden.

3.4. Ein weiteres Beispiel

Gegeben sei folgender Text:

Spalte	häufigster Buchstabe	Schlüsselwortbuchstabe
1	V⟶e	R
2	E⟶e	A
3	H,D,U⟶e	D,Z,Q
4	N⟶e	I
5	S⟶e	O

```
KWCSS GXYUT ZBZMU CMRFY JZZNZ HMEBS WMEXA ZMZAW
IATBS ABUUK NMZHT ZAKCE HBVLY ZPVCE OMONT PKYML
VJVGW CZRFK ZQEYF FTRLL ZFKVM XPJNS WMEXS MAKYD
GMEES IVRVW MURHV VZWHA XPKPW MOVMK ZVUUK NLVLY
ZPVCE OMONV ZVBFS MBVRL ZQEXW PBZAT ZAKCE HMEGM
NAQOE WMZMH DMCCK OMJHA XPKGG ZOICU CLRMK DMVCF
ZURFY JZZNZ HCJXW MOVBW DUKYP OJLWZ NBRVW IQDED
VZKYP OMZHE VTVOF YMZHS ILVLW NURFK ZVKMH MQTBL
JPEYV VAJYK YIWOW MMZHW MMXYD BQSNV DMUYE ZUGZS
ZVXYJ BMEUM NIXNO VVEYJ ZCEXO VVEYJ NMENK KZZWZ
OMJCK OMENK XPVCV ZVUXS NARHB ZLVLK OMCFW YMJEJ
TXKIY MIDGK YMIMU CTLYK NMCYA ILVOL DOUYF FTRLL
ZFKVM XPJNS WMETM EMUYE BMYYA HBVRL WCTBK OISYF
AMJND ZOK
```

Wir führen zunächst den Kasiski-Test durch. Dazu suchen wir im Text nach gleichen Textfolgen:

```
KWCSS GXYUT ZBZMU CMRFY JZZNZ HMEBS WMEXA ZMZAW
IATBS ABUUK NMZHT ZAKCE HBVLY ZPVCE OMONT PKYML
VJVGW CZRFK ZQEYF FTRLL ZFKVM XPJNS WMEXS MAKYD
GMEES IVRVW MURHV VZWHA XPKPW MOVMK ZVUUK NLVLY
ZPVCE OMONV ZVBFS MBVRL ZQEXW PBZAT ZAKCE HMEGM
NAQOE WMZMH DMCCK OMJHA XPKGG ZOICU CLRMK DMVCF
ZURFY JZZNZ HCJXW MOVBW DUKYP OJLWZ NBRVW IQDED
VZKYP OMZHE VTVOF YMZHS ILVLW NURFK ZVKMH MQTBL
JPEYV VAJYK YIWOW MMZHW MMXYD BQSNV DMUYE ZUGZS
ZVXYJ BMEUM NIXNO VVEYJ ZCEXO VVEYJ NMENK KZZWZ
OMJCK OMENK XPVCV ZVUXS NARHB ZLVLK OMCFW YMJEJ
TXKIY MIDGK YMIMU CTLYK NMCYA ILVOL DOUYF FTRLL
ZFKVM XPJNS WMETM EMUYE BMYYA HBVRL WCTBK OISYF
AMJND ZOK
```

Folge	Abstand	Primfaktorzerlegung
SWMEX	80	$2^2 \cdot 5$
UUK	105	$3 \cdot 5 \cdot 7$
OMO	95	$5 \cdot 19$
YFFTRLLZFKVMXPJNSWME	380	$2^2 \cdot 5 \cdot 19$
ZVU	265	$5 \cdot 53$

Wir bilden nun den grössten gemeinsamen Teiler der Abstandszahlen. Dieser ggT ist 5, d.h. die Schlüsselwortlänge hat vermutlich die Länge fünf.

Wir führen jetzt den Friedmann-Test durch. Dazu bestimmen wir die Anzahl der Buchstaben und ihre absolute Häufigkeit. In unserem Fall ist $n = 528$ und Damit können wir

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
16	17	20	11	27	15	8	15	12	17	30	21	51	20	22	13	6
R	S	T	U	V	W	X	Y	Z								
13	14	18	39	22	16	29	43									

die Formel

$$\kappa = \frac{\sum_{i=1}^{26} n_i^2}{n(n-1)}$$

anwenden und erhalten

$$\kappa = \frac{13644}{528 \cdot 527} = 4.9\%.$$

Nun ist es möglich, die Schlüsselwortlänge zu bestimmen:

$$l \approx \frac{3.77}{4.9 - 3.85} \approx 3.6.$$

Wir vermuten also eine Schlüsselwortlänge von drei. Der Vergleich der beiden Methoden zeigt, dass kein eindeutiges Resultat entsteht. Wir probieren zunächst die Vermutung der Schlüsselwortlänge fünf, da der Kasiski-Test eindeutig funktionierte. Dazu schreiben wir den Text in Spalten zu je fünf Zeichen.

```
KWCSS ...
GXYUT ...
ZBZMU ...
CMRFY ...
JZZNZ ...
```

Wir suchen nun in jeder Spalte den häufigsten Buchstaben und nehmen an, dass dieser für `e` steht. Wie unschwer zu sehen ist, lautet das Schlüsselwort vermutlich `VIRUS`. Wir

Spalte	häufigster Buchstabe	Schlüsselwortbuchstabe
1	Z⟶e	V
2	M⟶e	I
3	V⟶e	R
4	Y⟶e	U
5	W,K⟶e	S,G

können den restlichen Text dechiffrieren.

Wir entschlüsseln und erhalten als Resultat:

> Polyalphabetische Algorithmen haben die Eigenschaft, dass ein bestimmter Geheimtextbuchstabe mehr als einen Klartextbuchstaben darstellen kann. Aber man darf nicht vergessen, dass der Geheimtext den Klartext eindeutig bestimmen muss. Zum Beispiel ist es nicht moeglich, das sie einem Algorithmus der Geheimtextbuchstaben im Klartext einmal `e` und ein anderes mal `s` entspricht, ohne dass es dafuer eine Regel gibt, die dem Empfaenger genau sagt, wann er `e` und wann er `s` entspricht. Es ist entscheidend, dass an jeder Stelle des Kryptogramms der Schluessel eindeutig den Klartextbuchstaben zu jedem Geheimtextbuchstaben festlegt.

3.5. Vigenère knacken

Entschlüssle folgende, mit Vigenère verschlüsselte, Nachricht.

```
JHGTAEKDHMMEYGCOGKEA ZPBFJHADOWBUAYWMOVUE
ECPPYLIOHIIAKSRLISAI IFVJAZLRYDPTRUDRQQSE
ARNJQCEGEVWDUOPSMXQS ESNTHMBNTLVHWGGLNFFC
CAPMZUUVVAZSHGRVZTXH DBKIEYLZPVNEEWMOVUEE
HMWFANVFCHABRNQBSFAE YOOSEKEFIPGFIAYOQSEI
AAGZGRYHNWHSUYAHWJFV ATNONXRLIAVXVJLIMHMB
NAIBQVZPVAPKQCEPHZGZ GUHLOVXVRBTFLXVQPEIH
MPNUDFVKWGGEVKIVVUZH KVZGLNHQYGBHLYHITNSL
FVZWALNNENQUPEQCPDEV VBCDSELNNEZFOLRUUPBT
ZNTVOSFPNQVXVYLCUWZO ENUZHIHRMRUHMCQLRFSO
SETUFVYWRCEEEVBQZFUU PBTARNQNDNYEAWWSTYNQ
HIKNYUZVDSZPTULONSLL QZZWGLRNUWSVAEAPXVGL
UAGYOFBNNECBTPGIRIGR PNRPNHNAUFXIRFLIAHHD
NSMNUN
```

4. Die Enigma

Die Enigma ist eine Rotor-Schlüsselmaschine, die das deutsche Militär während des Zweiten Weltkriegs zur Verschlüsselung des Nachrichtenverkehrs verwendete. Das griechische Wort $\alpha\iota\nu\iota\gamma\mu\alpha$ bedeutet „Rätsel“. Obwohl die Verschlüsselungsqualität der Maschine während des Krieges mehrfach weiterentwickelt wurde, konnten die Alliierten durch enormen Aufwand zur Entschlüsselung während der meisten Zeit die deutschen Funksprüche mitlesen.

4.1. Geschichte

Nach dem Ersten Weltkrieg suchten die deutschen Militärs nach einem Ersatz für die inzwischen veralteten, umständlichen und unsicheren manuellen Verschlüsselungsverfahren (beispielsweise ADFGX oder Codebücher), die bis dahin verwendet wurden. Hierfür kamen maschinelle Verfahren in Betracht, weil sie eine einfachere Handhabung und eine verbesserte kryptographische Sicherheit versprachen. Basierend auf zu Beginn des 20. Jahrhunderts neu aufgekommenen Techniken, wie der elektrischen Schreibmaschine und dem Fernschreiber, kamen unabhängig voneinander und nahezu zeitgleich vier Erfinder auf die Idee des Rotor-Prinzips zur Verschlüsselung von Texten. Dabei handelt es sich um den Amerikaner Edward Hugh Hebern im Jahr 1917 (Patentanmeldung 1921), den Deutschen Arthur Scherbius im Jahr 1918 sowie den Niederländer Hugo Koch und den Schweden Arvid Gerhard Damm im Jahr 1919, die alle ihre Ideen zu Rotor-Chiffriermaschinen zum Patent anmeldeten.

Als Erfinder der Enigma gilt der promovierte deutsche Elektroingenieur Arthur Scherbius (1878–1929) dessen erstes Patent hierzu vom 23. Februar 1918 stammt. Die Enigma war zunächst als ziviles Chiffriersystem konzipiert und wurde kommerziell auf Messen — wie 1923 auf dem internationalen Postkongress des Weltpostvereins in Bern — zum Kauf angeboten. Gegen Ende der 1920er Jahre zeigten militärische Stellen verstärkt Interesse, so dass die Maschine bald darauf vom zivilen Markt verschwand. Gerade im Aufschwung des bis dahin eher schleppend verlaufenden Vertriebs verunglückte Scherbius tödlich.

Die nationalsozialistische Herrschaft hatte bereits begonnen. Da im Zuge der Aufrüstung der Wehrmacht ein zuverlässiges Verschlüsselungssystem benötigt wurde, stand dem Erfolg der Enigma nun nichts mehr im Wege. Man schätzt, dass während des Zweiten Weltkriegs mehr als 30 000 Maschinen produziert wurden, einige Schätzungen reichen bis 200 000 Stück, vermutlich liegt die tatsächliche Zahl der eingesetzten Maschinen bei etwa 100 000 Stück. Im Laufe der Zeit — bis zum Kriegsende 1945 und noch darüber hinaus — kamen viele verschiedene Modelle und Varianten der Enigma zum Einsatz.

Abbildung 4: Die ENIGMA

Die meistgebrauchte war die Enigma 1, die ab 1930 von der Reichswehr und später von der Wehrmacht eingesetzt wurde und das während des Zweiten Weltkriegs wohl am häufigsten benutzte Verschlüsselungsverfahren verkörpert.

4.2. Prinzip

Die ENIGMA I inklusive Holzgehäuse wiegt rund 12 kg und die äusseren Abmessungen (Länge × Breite × Hähe) betragen etwa 340 mm × 280 mm × 150 mm. Sie sieht auf den ersten Blick wie eine Schreibmaschine aus und besteht im Wesentlichen aus der Tastatur, einem Walzensatz von drei austauschbaren Walzen (Rotoren mit einem Durchmesser von etwa 100 mm) und einem Lampenfeld zur Anzeige.

Der Walzensatz ist das Herzstück zur Verschlüsselung. Die drei Walzen sind drehbar angeordnet und weisen auf beiden Seiten für die 26 Gro§buchstaben des lateinischen Alphabets 26 elektrische Kontakte auf, die durch 26 isolierte Drähte im Inneren der Walze paarweise und unregelmässig miteinander verbunden sind, beispielsweise (Walze III) Kontakt A mit B, B mit D, und so weiter. Drückt man eine Buchstabentaste, so flie§t elektrischer Strom von einer in der ENIGMA befindlichen 4.5-Volt-Batterie über die gedrückte Taste durch den Walzensatz und lässt eine Anzeigelampe aufleuchten. Der aufleuchtende Buchstabe entspricht der Verschlüsselung des gedrückten Buchstabens. Da sich bei jedem Tastendruck die Walzen ähnlich wie bei einem mechanischen Kilometerzähler weiterdrehen, ändert sich das geheime Schlüsselalphabet nach jedem Buchstaben.

Gibt man `otto` ein, so leuchten nacheinander beispielsweise die Lampen `PQWS` auf. Wichtig und kryptographisch stark ist, dass aufgrund der Rotation der Walzen jeder Buchstabe auf eine andere Weise verschlüsselt wird, im Beispiel das erste `o` von `otto` zu `P`,

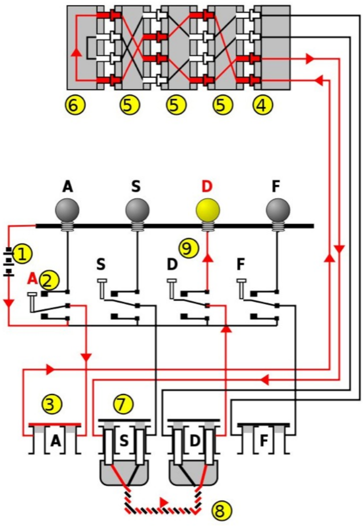


Abbildung 5: Prinzipieller Aufbau der ENIGMA: (1) Batterie, (2) Tastatur, (3,7) Steckbrett mit (8) Stecknabel, (5) Walzensatz mit (4) Eintrittswalze und (6) Umkehrwalze sowie dem (9) Lampenfeld

das zweite aber zu S. Man spricht von vielen unterschiedlichen (Geheim-) Alphabeten, die zur Verschlüsselung benutzt werden und bezeichnet dies als polyalphabetische Substitution. Würden sich die Walzen der ENIGMA nicht drehen, so bekäme man auch bei ihr nur eine einfache monoalphabetische Verschlüsselung.

4.3. Aufbau

In Abbildung 5 sehen Sie den schematischen Aufbau der ENIGMA. Rechts der drei drehbaren Walzen (5) des Walzensatzes (siehe gelb hinterlegte Zahlen in der Prinzipskizze) befindet sich die Eintrittswalze (4) (Stator), die sich nicht dreht und deren Kontakte über 26 Drähte (hier sind nur vier davon gezeichnet) mit den Buchstabentasten (2) verbunden sind. Links des Walzensatzes befindet sich die Umkehrwalze (6) (UKW), die ebenfalls feststeht. Bei ihr handelt es sich um eine Erfindung (patentiert am 21. März 1926) von Willi Korn, einem Mitarbeiter von Scherbius. Sie weist nur auf ihrer rechten Seite 26 Kontakte auf (in der Skizze sind wieder nur vier davon eingezeichnet), die paarweise miteinander verbunden sind. Die Umkehrwalze bewirkt, dass der Strom, der den Walzensatz zunächst von rechts nach links durchläuft, umgelenkt wird und ihn noch einmal durchfliesst, nun von links nach rechts. Der Strom verlässt den Walzensatz, wie er gekommen ist, wieder über die Eintrittswalze.

An der Gerätefront ist ein Steckerbrett mit doppelpoligen Steckbuchsen für jeden der

26 Buchstaben angebracht. Der Strom von der Buchstabentaste (2) wird, bevor er die Eintrittswalze (4) erreicht, über dieses Steckerbrett (3) geführt. Nach Durchlaufen des Walzensatzes fliesst er ein zweites Mal über das Steckerbrett (7,8) und bringt schlie§lich eine der 26 Buchstabenlampen (9) zum Aufleuchten.

4.4. Schlüsselraum

Der gesamte Schlüsselraum einer ENIGMA I mit drei aus einem Vorrat von fünf ausgewählten Walzen und einer von zwei Umkehrwalzen sowie bei Verwendung von zehn Steckern lässt sich aus dem Produkt der 120 Walzenlagen, 676 Ringstellungen, 16 900 Walzenstellungen und 150 738 274 937 250 Steckermöglichkeiten berechnen. Er beträgt:

$$120 \cdot 676 \cdot 16\,900 \cdot 150\,738\,274\,937\,250 = 206\,651\,321\,783\,174\,268\,000\,000.$$

Das sind etwa $2 \cdot 10^{23}$ Möglichkeiten und entspricht einer Schlüssellänge von ungefähr 77 bit.

Der Schlüsselraum war für diese Zeit enorm gross, ein vollständiges Durchsuchen war mit der damaligen Technologie aussichtslos. Wie wir gesehen haben, ist aber die Grösse des Schlüsselraums jedoch nur eine notwendige, aber keine hinreichende Bedingung für die Sicherheit eines kryptographischen Verfahrens. Selbst eine so simple Methode wie die einfache monoalphabetische Substitution verfügt über 26! mögliche Schlüssel. Das sind grob $4000 \cdot 10^{23}$ Schlüssel und entspricht ungefähr 88 bit und ist folglich sogar noch um etwa den Faktor 2000 grösser als bei der ENIGMA I. Dennoch ist eine monoalphabetische Substitution leicht zu brechen.

4.5. Entzifferung

Die Betreiber der Schlüsselmaschine ENIGMA waren der Meinung, dass die durch sie maschinell verschlüsselten Texte (im Gegensatz zu fast allem, was bis 1918 gebräuchlich war) mit manuellen Methoden nicht zu knacken sind. Was übersehen wurde, ist, dass einer maschinellen Verschlüsselung durch maschinelle Entzifferung begegnet werden kann.

Nachdem es weder Franzosen noch Briten gelang, diese Informationen zu nutzen, und sie die ENIGMA nach wie vor als unknackbar einstuften, glückte dem 27-jährigen polnischen Mathematiker Marian Rejewski bei seiner Arbeit in der polnischen Dechiffrierstelle bereits im Jahre 1932 der erste Einbruch in die ENIGMA. Rejewski erriet die von den Deutschen für die militärische Variante gewählte Verdrahtungsreihenfolge. Anschliessend schaffte er es mit Hilfe seiner exzellenten Kenntnisse der Permutationstheorie, die Verdrahtung der drei Walzen (I bis III) sowie der Umkehrwalze zu erschliessen — eine kryptoanalytische Meisterleistung, die ihn mit den Worten des amerikanischen Historikers David Kahn in das Pantheon der grössten Kryptoanalytiker aller Zeiten erhebt.

Die nächste Aufgabe, die gelöst werden musste, war, jeweils die richtige Walzenlage und Walzenstellung zu erschliessen. Dazu nutzte Rejewski zusammen mit seinen 1932 hinzugekommenen Kollegen Jerzy R—życki und Henryk Zygalski einen schwerwiegenden verfahrenstechnischen Fehler aus, der den Deutschen unterlief. Um eine sichere †bertragung zu gewährleisten, wurde zu dieser Zeit der Spruchschlüssel noch zweimal hintereinander gestellt und verschlüsselt an den Anfang einer Nachricht geschrieben. Somit war der erste und vierte, der zweite und fünfte sowie der dritte und sechste Geheimtextbuchstabe jeweils demselben Klartextbuchstaben zuzuordnen. Mit Hilfe zweier speziell zu diesem Zweck gebauter Maschinen, genannt Zyklometer und Bomba, die zwei beziehungsweise sechs hintereinander geschaltete und um drei beziehungsweise eine bis fünf Drehpositionen versetzte ENIGMA-Maschinen verkörperten, konnten die polnischen Codeknacker für jede der sechs möglichen Walzenlagen feststellen, bei welchen Walzenstellungen die beobachtete Zuordnung der Buchstabenpaare möglich war und so den Suchraum gewaltig einengen. Nach Analyse mehrerer Funksprüche war der korrekte Spruchschlüssel gefunden.

Nachdem die Deutschen am 15. September 1938 ihre Verfahrenstechnik änderten und drei Monate später mit Einführung der Walzen IV und V die Anzahl der möglichen Walzenlagen von sechs auf sechzig erhöhten, konnten die Polen nicht mehr mithalten, und die ENIGMA war wieder sicher. Die nächste Hürde musste überwunden werden.

Mit diesem Anschub, konnten die britischen Kryptoanalytiker mit Ausbruch des Krieges im etwa 70 km nordwestlich von London gelegenen Bletchley Park einen erneuten Angriff auf die ENIGMA starten. Das wichtigste Hilfsmittel dabei war — neben ihrer intellektuellen Leistungsfähigkeit und dem hohen Personaleinsatz von später zehn- bis vierzehntausend Frauen und Männern — vor allem eine spezielle elektromechanische Maschine, genannt die Turing-Bombe, die auf der polnischen Bomba aufbaute und vom englischen Mathematiker Alan Turing ersonnen wurde. Turings Idee zur Schlüsselsuche bestand darin, durch ringförmige Verkettung von mehreren (meist zwölf) ENIGMA-Walzensätzen die Wirkung des Steckerbretts komplett abzustreifen. Dadurch gelang es ihm, die praktisch unüberschaubare Anzahl von Verschlüsselungsmöglichkeiten, auf die die deutschen Kryptographen ihre Hoffnungen setzten, drastisch zu reduzieren.

Mit Hilfe der Turing-Bombe brauchte man nur noch rund sechs Stunden, um sämtliche Möglichkeiten durchzutesten. Leistet man sich den Aufwand, sechzig Bomben einzusetzen, jeweils eine für jede Walzenlage, dann schrumpft die Zeit von sechs Stunden auf sechs Minuten — eine durchaus erträgliche Zeit. Tatsächlich waren bis zum Kriegsende mehr als 210 Bomben allein in England in Betrieb.

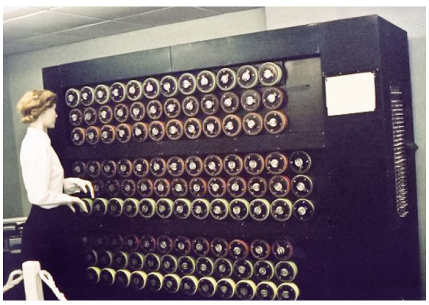


Abbildung 6: Die Turingbombe

Entscheidend wichtig für die Funktion der Bombe sind "wahrscheinliche Wörter", deren Auftreten man im Text erwarten kann; fehlen diese, dann scheitert die Entzifferung. Man profitierte von der deutschen Gründlichkeit bei der Abfassung von Routinemeldungen, wie Wetterberichte, die jeden Morgen pünktlich zur selben Zeit und vom selben Ort gesendet wurden. Aus britischer Sicht war eine täglich frisch verschlüsselte ENIGMA-Meldung, die stets mit den Worten `WETTERVORHERSAGEBEREICHSIEBEN` begann, ähnlich wertvoll wie es eine direkte öffentliche Bekanntgabe des jeweils gültigen Tagesschlüssels gewesen wäre. So wurde beispielsweise der ENIGMA-Schlüssel vom „D-Day" durch das Wort `WETTERVORHERSAGEBISKAYA`, den die britischen Kryptoanalytiker leicht erraten konnten und korrekt vermuteten, in weniger als zwei Stunden nach Mitternacht gebrochen. Nun kamen auch die Amerikaner zu Hilfe, die ab April 1943 mehr als 120 Stück Hochgeschwindigkeitsvarianten der Turing-Bombe produzierten. Danach waren die deutschen U-Boote nie mehr sicher. Unmittelbare Folge der amerikanischen Entzifferungen war die Versenkung von neun der zwölf deutschen U-Tanker („Milchkühe") innerhalb weniger Wochen im Sommer 1943. Dies führte zu einer Schwächung aller Atlantik-U-Boote, die nun nicht mehr auf See versorgt werden konnten, sondern dazu die lange und gefährliche Heimreise durch die Biskaya zu den U-Boot-Stützpunkten an der französischen Westküste antreten mussten.

4.6. Lernprogramm

Das Innenleben der Enigma erleben und mehr zur Entschlüsselungstechnik kennen lernen können bzw. sollen Sie auf MathePrisma unter

```
MathePrisma Enigma
```

Dort kann auch während des Moduls eine Java-Enigma benutzt werden.

		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0			2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	32	0	0	1	0	0	0	0	0	n	110	0	1	1	0	1	1	1	0
,	44	0	0	1	0	1	1	0	0	o	111	0	1	1	0	1	1	1	1
.	46	0	0	1	0	1	1	1	0	p	112	0	1	1	1	0	0	0	0
a	97	0	1	1	0	0	0	0	1	q	113	0	1	1	1	0	0	0	1
b	98	0	1	1	0	0	0	1	0	r	114	0	1	1	1	0	0	1	0
c	99	0	1	1	0	0	0	1	1	s	115	0	1	1	1	0	0	1	1
d	100	0	1	1	0	0	1	0	0	t	116	0	1	1	1	0	1	0	0
e	101	0	1	1	0	0	1	0	1	u	117	0	1	1	1	0	1	0	1
f	102	0	1	1	0	0	1	1	0	v	118	0	1	1	1	0	1	1	0
g	103	0	1	1	0	0	1	1	1	w	119	0	1	1	1	0	1	1	1
h	104	0	1	1	0	1	0	0	0	x	120	0	1	1	1	1	0	0	0
i	105	0	1	1	0	1	0	0	1	y	121	0	1	1	1	1	0	0	1
j	106	0	1	1	0	1	0	1	0	z	122	0	1	1	1	1	0	1	0
k	107	0	1	1	0	1	0	1	1	ü	129	1	0	0	0	0	0	0	1
l	108	0	1	1	0	1	1	0	0	ä	132	1	0	0	0	0	1	0	0
m	109	0	1	1	0	1	1	0	1	ö	148	1	0	0	1	0	1	0	0

Abbildung 7: Auszug aus ASCII binär

5. Data Encryption Standard

5.1. Binärcodes

Auf einem Rechner werden alle Daten als Bit-Folgen dargestellt (binär codiert). Der Data Encryption Standard (DES) verschlüsselt solche Bit-Folgen.

Eine der gebräuchlichsten Binärcodierungen ist ASCII (American Standard Code for Information Interchange). Im ASCII wird je ein Zeichen durch ein Byte (= 8 Bits) codiert.

Übung 4. Wie viele verschiedene Zeichen können mit ASCII codiert werden?

Abbildung 7 auf Seite 31 zeigt einen kleinen Auszug aus ASCII (in blau ist jeweils als Dezimalzahl angegeben, welchen Wert die Bit-Folge hat, wenn man sie als Zahl im Dualsystem interpretiert).

Übung 5. Verwandle folgendes Binärwort zurück:

```
01100110 01100101 01101001 01110011
01110100 01100101 01101100
```

Der Deutsche Horst Feistel wurde am 30. Januar 1915 in Berlin geboren. 1934 emigrierte er nach Amerika. Wegen seiner kryptographischen Forschungen legte ihm die NSA (National Security Agency) immer wieder Steine in den Weg. Bei IBM in New York entwickelte er schliesslich in den siebziger Jahren das Prinzip der Feistel-Netzwerke, Grundlage einiger bedeutender Verschlüsselungsverfahren. Feistel starb 1990.

Zurück zu ASCII stellen wir fest, dass ein so codierter Text sich nun allerdings 8-mal so lang darstellt wie der ursprüngliche (uncodierte) Text. Man kann ihn kürzer darstellen, wenn man für je vier Bits eine Hexadezimalziffer schreibt. Denn vier Bits stellen genau die Zahlen von 0 bis 15 dar:

$$\begin{aligned} 0000 &= 0 \\ 0001 &= 1 \\ 0010 &= 2 \\ 0011 &= 3 \\ &\dots \\ 1101 &= 13 \\ 1110 &= 14 \\ 1111 &= 15 \end{aligned}$$

Man verwendet dafür die Ziffern

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.$$

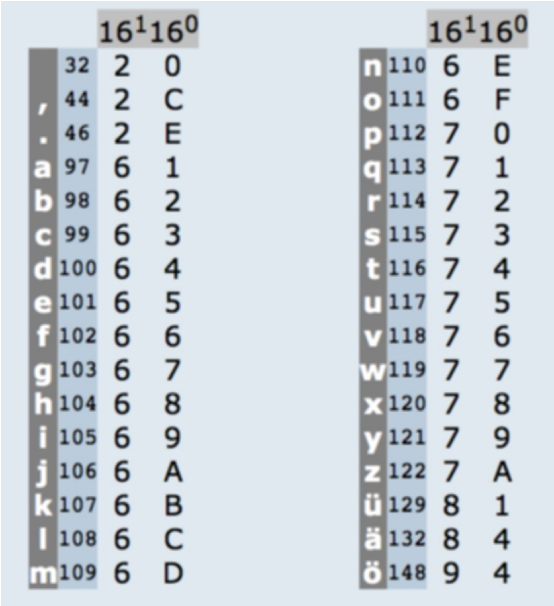
Um die unübersichtlichen Bit-Folgen eines ASCII-codierten Textes kürzer zu schreiben, stellt man jedes Byte durch zwei Hex-Ziffern dar, eine für die ersten und eine für die letzten vier Bits. Abbildung 8 auf Seite 33 zeigt noch einmal den Ausschnitt aus ASCII. Für die hexadezimale Schreibweise benätigt man statt 8 nur zwei Ziffern.

Übung 6. Kannst du folgendes binärcodiertes, mit Hex-Ziffern dargestelltes Wort zurückwandeln?

```
6C 75 63 69 66 65 72
```

Lucifer ist der Name eines Verschlüsselungsverfahrens, das Horst Feistel und Don Coppersmith 1973 im Thomas J. Watson Laboratory von IBM in New York entwickelten. Die NSA (National Security Agency) wurde gebeten, die Sicherheit der Lucifer-Verschlüsselung zu beurteilen. Die NSA scheint von der Sicherheit Lucifers beeindruckt gewesen zu sein: Sie bestand darauf, die Zahl der Schlüssel zu begrenzen (das Verfahren also künstlich unsicherer zu machen).

> „Offenbar glaubte die NSA, eine solche Obergrenze würde im zivilen Gebrauch die Sicherheit gewährleisten, weil keine nichtmilitärische Organisation einen Computer besass, der leistungsfähig genug war, um jeden möglichen



		16^1	16^0			16^1	16^0
	32	2	0	n	110	6	E
,	44	2	C	o	111	6	F
.	46	2	E	p	112	7	0
a	97	6	1	q	113	7	1
b	98	6	2	r	114	7	2
c	99	6	3	s	115	7	3
d	100	6	4	t	116	7	4
e	101	6	5	u	117	7	5
f	102	6	6	v	118	7	6
g	103	6	7	w	119	7	7
h	104	6	8	x	120	7	8
i	105	6	9	y	121	7	9
j	106	6	A	z	122	7	A
k	107	6	B	ü	129	8	1
l	108	6	C	ä	132	8	4
m	109	6	D	ö	148	9	4

Abbildung 8: Auszug aus ASCII hexadizimal

Schlüssel in einem vernünftigen Zeitraum zu prüfen. Die NSA selbst jedoch, die über die besten Computer der Welt verfügt, würde gerade noch in der Lage sein, in den verschlüsselten Nachrichtenverkehr einzubrechen.“

(aus: Simon Singh, Geheime Botschaften)

Diese Version von Lucifer mit beschränkter Schlüsselzahl wurde 1976 unter dem Namen Data Encryption Standard (DES) der offizielle amerikanische Verschlüsselungsstandard.

Dez	Hex	Okt	Zeichen	Dez	Hex	Okt	Zeichen
0	0x00	000	NUL	32	0x20	040	SP
1	0x01	001	SOH	33	0x21	041	!
2	0x02	002	STX	34	0x22	042	“
3	0x03	003	ETX	35	0x23	043	#
4	0x04	004	EOT	36	0x24	044	$
5	0x05	005	ENQ	37	0x25	045	%
6	0x06	006	ACK	38	0x26	046	&
7	0x07	007	BEL	39	0x27	047	’
8	0x08	010	BS	40	0x28	050	(
9	0x09	011	TAB	41	0x29	051	)
10	0x0A	012	LF	42	0x2A	052	*
11	0x0B	013	VT	43	0x2B	053	+
12	0x0C	014	FF	44	0x2C	054	,

13	0x0D	015	CR	45	0x2D	055	-
14	0x0E	016	SO	46	0x2E	056	.
15	0x0F	017	SI	47	0x2F	057	/
16	0x10	020	DLE	48	0x30	060	0
17	0x11	021	DC1	49	0x31	061	1
18	0x12	022	DC2	50	0x32	062	2
19	0x13	023	DC3	51	0x33	063	3

Dez	Hex	Okt	Zeichen	Dez	Hex	Okt	Zeichen
20	0x14	024	DC4	52	0x34	064	4
21	0x15	025	NAK	53	0x35	065	5
22	0x16	026	SYN	54	0x36	066	6
23	0x17	027	ETB	55	0x37	067	7
24	0x18	030	CAN	56	0x38	070	8
25	0x19	031	EM	57	0x39	071	9
26	0x1A	032	SUB	58	0x3A	072	:
27	0x1B	033	ESC	59	0x3B	073	;
28	0x1C	034	FS	60	0x3C	074	≪
29	0x1D	035	GS	61	0x3D	075	=
30	0x1E	036	RS	62	0x3E	076	γ
31	0x1F	037	US	63	0x3F	077	?
64	0x40	100	@	96	0x60	140	‘
65	0x41	101	A	97	0x61	141	a
66	0x42	102	B	98	0x62	142	b
67	0x43	103	C	99	0x63	143	c
68	0x44	104	D	100	0x64	144	d
69	0x45	105	E	101	0x65	145	e
70	0x46	106	F	102	0x66	146	f
71	0x47	107	G	103	0x67	147	g
72	0x48	110	H	104	0x68	150	h
73	0x49	111	I	105	0x69	151	i
74	0x4A	112	J	106	0x6A	152	j
75	0x4B	113	K	107	0x6B	153	k
76	0x4C	114	L	108	0x6C	154	l
77	0x4D	115	M	109	0x6D	155	m
78	0x4E	116	N	110	0x6E	156	n
79	0x4F	117	O	111	0x6F	157	o
80	0x50	120	P	112	0x70	160	p
81	0x51	121	Q	113	0x71	161	q
82	0x52	122	R	114	0x72	162	r
83	0x53	123	S	115	0x73	163	s
84	0x54	124	T	116	0x74	164	t
85	0x55	125	U	117	0x75	165	u

86	0x56	126	V	118	0x76	166	v
87	0x57	127	W	119	0x77	167	w
88	0x58	130	X	120	0x78	170	x
89	0x59	131	Y	121	0x79	171	y
90	0x5A	132	Z	122	0x7A	172	z
91	0x5B	133	[	123	0x7B	173	{
92	0x5C	134	\	124	0x7C	174	|
93	0x5D	135	]	125	0x7D	175	}
94	0x5E	136	^	126	0x7E	176	-
95	0x5F	137	_	127	0x7F	177	DEL

5.2. Erste DES Schicht

DES ist eine Block-Chiffre. Bei einer Block-Chiffre wird zum Verschlüsseln einer Nachricht diese in mehrere Blöcke gleicher Länge aufgeteilt. Jeder Block wird nach dem gleichen Prinzip verschlüsselt. Beim DES ist die Länge der Blöcke 64 Bits = 8 Bytes. Für die Sicherheit einer Block-Chiffre ist eine relativ grosse Blocklänge notwendig. Enthält der letzte Block der zu verschlüsselnden Nachricht weniger als 64 Bit, so füllt man ihn mit irgendwelchen Dummy-Bits auf. Beim Verschlüsseln von Bit-Folgen im DES verwendet man verschiedene Standard-Operationen, die wir jetzt behandeln. DES arbeitet mit Blöcken von je 64 Bits = 8 Bytes. Um den Überblick nicht zu verlieren, nehmen wir hier aber immer nur 1 Byte.

Die XOR-Operation verknüpft zwei Bits nach folgenden Regeln:

$$0XOR0 = 0, 0XOR1 = 1, 1XOR0 = 1, 1XOR1 = 0$$

oder kurz: Bei zwei Bit-Folgen gleicher Länge wendet man XOR auf jede Position einzeln

XOR	0	1
0	0	1
1	1	0

Tabelle 19: XOR

an. Das Ergebnis ist wieder eine Bit-Folge.

Übung 7. Bestimme das Ergebnis der XOR-Operation.

$$\texttt{01000101} \oplus \texttt{00110011}$$

Eine andere wichtige Manipulation von Bit-Folgen ist, die Positionen der einzelnen Bits zu vertauschen. Man sagt: die Bit-Folge wird durch eine Permutation auf eine neue Bit-Folge abgebildet. Eine Permutation wird einfach durch Aufzählen der neuen Positionen angegeben: Dies ist die Permutation

$$(2\ 3\ 1\ 8\ 5\ 4\ 6\ 7)$$

Beispiel 2. Die 8-Bit-Folge

$$10011010$$

wird abgebildet auf

$$01001101$$

Übung 8. Gib die durch (4 5 2 3 1 6 8 7) permutierte Folge von 10010100 an.

Das DES verwendet insgesamt drei verschiedene Permutationen, welche mit IP, PI und P bezeichnet werden. IP und PI permutieren 64-Bit-Folgen, P permutiert 32-Bit-Folgen. Ausserdem verwendet DES noch Varianten von Permutationen (E, PC1, PC2), die wir später besprechen werden.

Wie eine Zwiebel ist der DES aus mehreren Schichten aufgebaut. Wir schauen uns die erste Schicht jetzt an (siehe Abbildung 9 auf Seite 37).

- DES besteht aus einer Anfangsphase, einer mittleren Phase mit 16 Runden (Nummer 0 bis 15) und einer Endphase.
- Die Anfangsphase besteht aus einer Permutation der gesamten Folge.
- In den Runden werden linke und rechte Hälfte der Bit-Folge getrennt behandelt.

  Jede Runde besteht aus einer Vertauschung von linker und rechter Hälfte, einer Manipulation der rechten Hälfte im F-Modul und einer XOR-Operation der manipulierten rechten mit der linken Hälfte.

  Jede Runde verwendet ihren eigenen Schlüssel.
- Die mittlere Phase wird mit einer Vertauschung von linker und rechter Hälfte abgeschlossen.
- Die Endphase besteht aus einer abschliessenden Permutation der gesamten Folge.

Wie entschlüsselt man eine DES-verschlüsselte Nachricht? Das kann man herausbekommen, ohne das F-Modul näher zu kennen. Man muss dazu aber wissen, wie man Permutationen und XOR-Operationen rückgängig macht.

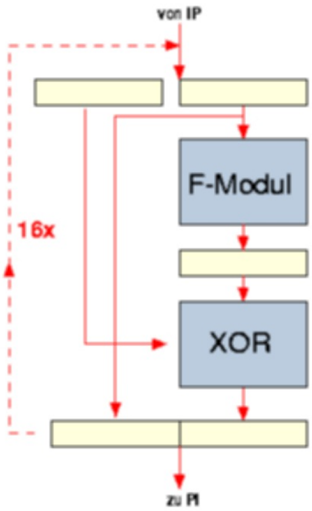


Abbildung 9: Feistel-Round

Zuerst kümmern wir uns um Permutationen. Ist b eine Bit-Folge und P eine Permutation, so bezeichnet P(b) die Bit-Folge, die aus b durch Anwendung der Permutation P entsteht.

Definition 5.1: Inverse Permutation

Die Permutation P_2 heisst Inverse zur Permutation P_1, wenn für jede Bit-Folge b gilt:

$$P_2(P_1(b)) = b.$$

Wendet man also nach einer Permutation ihre Inverse an, so bleiben alle Positionen unverändert.

Beispiel 3. Es sei

$$P_1 = (\ 2\ 3\ 1\ 8\ 5\ 4\ 6\ 7\ ).$$

Dann ist

$$P_2 = (\ 3\ 1\ 2\ 6\ 5\ 7\ 8\ 4\ )$$

die Inverse von P_1, denn P_2 nach P_1 angewendet lässt alle Positionen unverändert.

Übung 9. Gib die Inverse der Permutation

$$(\ 1\ 2\ 6\ 7\ 4\ 5\ 8\ 3\ )$$

Im DES ist die End-Permutation PI die Inverse zur Anfangspermutation IP.

Und jetzt kümmern wir uns darum, wie man die XOR-Operation rückgängig macht.

Wie rekonstruiert man b1 aus b2 XOR b1 und b2?

Übung 10. Angenommen, man kennt

$$b_2 = 11101001$$

und

$$b_2 \oplus b_1 = 10111010$$

Bestimme b_1.

Aus der Übung sollten Sie erkennen: b_1 ergibt sich aus b_2 und $b_2 \oplus b_1$ durch eine weitere XOR-Operation:

$$b_1 = b_2 \oplus (b_2 \oplus b_1)$$

Sie haben gesehen: Zum Verschlüsseln verwendet DES insgesamt 16 Schlüssel, je einen pro Runde.

Nach dieser Vorbereitung sollte Ihnen das Entschlüsseln auch gelingen, wenn mehrere Runden durchgeführt werden. Eventuell hilft Ihnen auch die formale algorithmische Beschreibung weiter.

- $r(b)$ bezeichnet die rechte Hälfte einer 64-Bit-Folge b,
- $l(b)$ bezeichnet die linke Hälfte,
- $b_1 b_2$ bezeichnet das Aneinanderhängen zweier Bit-Folgen b_1 und b_2.

  Die Zuordnung

  $$b := r(b)l(b)$$

  beschreibt also das Vertauschen von linker und rechter Hälfte.
- Mit $F(b, i)$ bezeichnen wir das Ergebnis des F-Moduls bei Anwendung auf die 32-Bit-Folge b in Runde i.

Algorithmus:

```
input: 64-Bit-Folge b
output: DES-Verschluesselung

c := IP(b)    {Anfangspermutation}
```

```
fuer i = 0,1,...,15      {Runden}
     c1 := r(c)
     c2 := l(c) XOR F(c1,i)
     c := c1c2
c := r(c)l(c)    {abschliessende Vertauschung}
c := PI(c)   {Endpermutation}
```

5.3. F-Modul und S-Module

Eine Verschlüsselung ist dann besonders gut, wenn jedes Bit des Schlüssels und jedes Bit des Originaltextes „gleich starken“ Einfluss auf jedes Bit des Ergebnisses haben. Dafür sorgt im DES der 16-fache Einsatz des F-Moduls, die zweite Schicht der DES-Zwiebel.

Wir untersuchen jetzt die Funktionsweise des F-Moduls (siehe Abbildung 10 auf Seite 40. Das F-Modul führt folgende Operationen aus:

- Erweiternde Permutation der 32-Bit-Folge auf eine 48-Bit-Folge.
- XOR-Operation mit einem 48-Bit-Schlüssel
- Reduktion von je 6 Bit auf 4 Bit mit Hilfe der S-Module S1 bis S8.
- Abschliessende Permutation.

Eine erweiternde Permutation bildet Bit-Folgen b_1 auf Bit-Folgen b_2 grösserer Länge ab. Im Gegensatz zu gewöhnlichen Permutationen werden nun einige Positionen in b_1 auf mehrere Positionen in b_2 abgebildet. Notation:

Beispiel 4. Dies ist die erweiternde Permutation

$$(\ 1, 3 \ 2 \ 4, 6 \ 5 \ ).$$

Die 4-Bit-Folge

$$1001$$

wird damit abgebildet auf

$$101010.$$

Die im F-Modul verwendete erweiternde Permutation E ist in der Tabelle ebenso angegeben wie die abschliessende Permutation P. Die S-Module bilden die dritte Schicht im DES. Sie verarbeiten 6-Bit-Folgen unter Verwendung von Substitutionen.

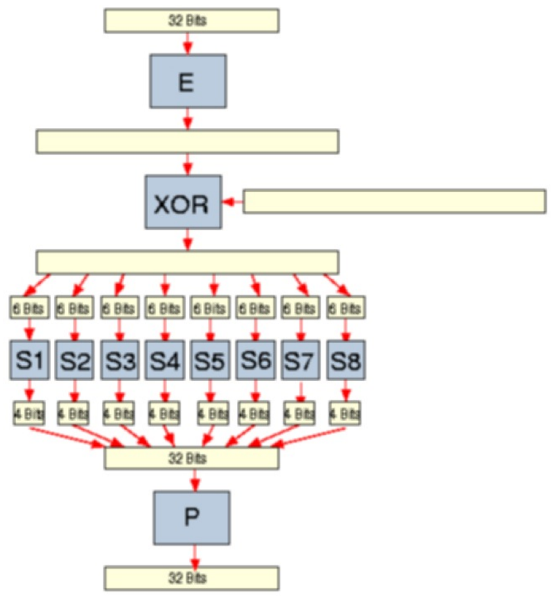


Abbildung 10: F-Modul

Definition 5.2: Substitution

Eine Substitution bildet jede Bit-Folge einer bestimmten Länge wieder auf eine Bit-Folge derselben Länge ab.

Substitutionen gibt man am Besten in Form einer Tabelle an, wie zum Beispiel in Abbildung 11 auf Seite 40 gezeigt. Die Substitution ist fundamental verschieden von einer Permutation. Sie beruht nämlich nicht darauf, die Bits innerhalb der Bit-Folgen zu ver-

S0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
01	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
10	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
11	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

Abbildung 11: S-Box

tauschen. Und so arbeiten die S-Module im DES:

- Jedes S-Modul verwendet 4 verschiedene Substitutionen auf Bit-Folgen der Länge 4. Jede Zeile der Tabelle repräsentiert eine solche Substitution.
- Verarbeitet werden 6-Bit-Folgen
- Anfangs- und Endbit der 6-Bit-Folge bestimmen, welche Substitution angewendet wird.
- Die mittleren 4 Bit werden mittels dieser Substitution auf eine 4-Bit-Folge abgebildet.

5.4. DES Schlüssel

DES lässt nur Schlüssel mit bestimmter Parität zu.

Definition 5.3: Parität

Ist die Zahl der Einsen in einer Bit-Folge ungerade, so ist die Parität der Bit-Folge 1, andernfalls ist die Parität 0.

Übung 11. Gib die Parität der Bit-Folge 0010011 an.

Im DES verwendet der Benutzer eine von ihm gewählte (geheime) 64-Bit-Folge als Schlüssel. Diese muss er dem Empfänger, der ja entschlüsseln soll, auch mitteilen. Aus Sicherheitsgründen sollte der Schlüssel selbst wieder verschlüsselt weitergegeben werden. Eine fehlerbehaftete Übertragung des Schlüssels sollte vom Empfänger erkannt werden können. Deshalb gibt es im DES die folgende Konvention:

> Es sind nur Schlüssel zugelassen, bei welchen jedes Byte Parität 1 hat.

Übung 12. Was erreicht man damit?

(a) Fehler von einem Bit pro Byte werden immer erkannt.

(b) Fehler von einem Bit pro Byte werden nur manchmal erkannt.

(c) Fehler von mehreren Bit pro Byte werden immer erkannt.

(d) Fehler von mehreren Bit pro Byte werden nur manchmal erkannt.

(e) Fehler von einem Bit pro Byte können korrigiert werden.

DES erzeugt aus dem 64-Bit-Schlüssel sechzehn 48-Bit-Schlüssel unter Verwendung von Linksverschiebungen und verringernden Permutationen.

Definition 5.4: Linksverschiebung

Die Permutation $(\; n \; 1 \; 2 \; \ldots \; n{-}1 \;)$ heisst Linksverschiebung um 1. Die Permutation $(\; n-1 \; n \; 1 \; 2 \; \ldots \; n-2 \;)$ heisst Linksverschiebung um 2.

Definition 5.5: Verminderte Permutation

Eine vermindernde Permutation bildet Bit-Folgen b_1 auf Bit-Folgen b_2 kleinerer Länge ab. Im Gegensatz zu gewöhnlichen Permutationen werden nun einige Positionen in b_1 ignoriert.

Beispiel 5. Eine vermindernde Permutation von 6 Bit auf 4 Bit ist

$$(\; - \; 2 \; 1 \; - \; 4 \; 3 \;)$$

Dadurch wird 101001 auf 1010 abgebildet.

Die im DES zur Schlüsselgenerierung verwendeten vermindernden Permutationen PC1 und PC2 sind in der Tabelle angegeben. PC1 bildet 64-Bit-Folgen auf 56-Bit-Folgen ab, PC2 bildet 56-Bit-Folgen auf 48-Bit-Folgen ab.

5.5. Sicherheit

Welche Bausteine des DES tragen wie zu dessen Sicherheit bei? Diese Frage wollen wir hier in ganz groben Zügen beantworten.

Bei einem statistischen Angriff versucht man, aus der Häufigkeit von Mustern in der verschlüsselten Nachricht auf den entschlüsselten Text zu schliessen. DES macht solche Angriffe aufgrund der grossen Blockgrösse von 64 Bit praktisch unmöglich.

5.5.1. Bedeutung der Blocklänge

Stelle dir vor, eine Nachricht wird in ASCII codiert. Auf die resultierende Bit-Folge wird eine Block-Chiffre mit nur der Länge 8 angewendet. Diese Nachricht kannst du dann so knacken:

1. Du teilst die verschlüsselte Nachricht in Blöcke zu je 8 Bit ein.
2. Du erstellst eine Tabelle, in welcher gezählt wird, wie häufig jede der 256 möglichen 8-Bit-Folgen vorkommt.
3. Die Tabelle vergleichst Du mit den durchschnittlichen Häufigkeiten der Buchstaben und Zeichen in deutschen Texten.
4. Die häufigste 8-Bit-Folge entspricht dann wohl dem 'e', die zweithäufigste dem 'n' usw.
5. Nach etwas Ausprobieren bist du am Ziel.

Übung 13. Warum kann eine Blocklänge von 64 als sicher gegen statistische Angriffe angesehen werden? Welche Aussagen sind richtig?

(a) 64 Bit entsprechen 8 ASCII-codierten Zeichen

(b) Es gibt 256^8 ($\approx 2 \cdot 10^{19}$) verschiedene Bit-Folgen der Länge 64.

(c) Kommt im unverschlüsselten Text ein Wort aus 8 Buchstaben (oder mehr) öfter vor, so wiederholt sich im verschlüsselten Text eine 64-Bit-Folge genauso oft.

(d) Kommt im unverschlüsselten Text ein Wort aus 8 Buchstaben (oder mehr) öfter vor, so wiederholt sich auch im verschlüsselten Text eine 64-Bit-Folge, aber im Schnitt nur rund ein Achtel so oft.

(e) Selbst wenn im verschlüsselten Text eine 64-Bit-Folge häufiger vorkommt, ist es sehr schwer, ihr das richtige Wort aus dem unverschlüsselten Text zuzuordnen.

(f) Wenn ich eine 64-Bit-Folge im verschlüsselten Text geknackt habe, so kenne ich die 8 Zeichen des Wortes. Damit finde ich dann alle Stellen im verschlüsselten Text, die diesen 8 Zeichen entsprechen.

Manche Chiffren sind leicht zu knacken, wenn man die Verschlüsselungen einiger weniger Blöcke kennt ('lineare' Chiffren). Dass dies im DES nicht möglich ist, wird durch die S-Module und die 16 Runden erreicht. Sie machen DES zu einer hochgradig nichtlinearen Chiffre.

Um das Konzept der linearen Chiffren zu verstehen, braucht man etwas mathematische Vorbildung.

Eine Chiffre für Bit-Folgen der Länge n heisst linear (eigentlich besser: affin), falls sich die verschlüsselte Bit-Folge b_2 aus b_1 ergibt über eine Vorschrift

$$b_2 = \boldsymbol{A} \cdot b_1 + \vec{c}$$

mit einer festen Matrix $\boldsymbol{A}$ und einem festen Vektor $\vec{c}$. Die arithmetischen Operationen sind dabei modulo 2 zu verstehen.

Bei einer linearen Chiffre reicht es, wenn man für einen Satz von $n+1$ Bit-Folgen, welche eine Basis enthalten, die jeweiligen verschlüsselten Folgen kennt. Daraus kann man nämlich $\boldsymbol{A}$ vollständig rekonstruieren.

Zum Abschluss besprechen wir noch den wohl naheliegendsten Angriff.

Man spricht von einem Brute-Force-Angriff auf eine Chiffre, wenn man einfach versucht, alle möglichen Schlüssel durchzuprobieren. Brute Force ist bis jetzt die erfolgreichste Angriffstaktik gegen DES.

Übung 14. Wie viele verschiedene Schlüssel gibt es im DES?

Nachdem schon PCs heutzutage um die 10^{10} Rechenoperationen in der Sekunde leisten können, ist es möglich, mit vielen PCs (und einige Tagen Rechenzeit) einen erfolgreichen Brute-Force-Angriff auf den DES zu fahren.

Man behilft sich deshalb mit dem Triple DES.

Beim Triple DES wird zur Verschlüsselung der DES 3-fach hintereinander angewendet, und zwar zuerst mit einem ersten Schlüssel, dann mit einem zweiten und schliesslich nochmals mit dem ersten.

Übung 15. Wieviele Möglichkeiten muss man bei einem Brute-Force-Angriff jetzt in Betracht ziehen? Gib die richtige Grössenordnung an! (Exponent e in 10^e)

Übung 16. Wieso nicht Double-DES?

Tatsächlich ist Triple DES eine heute häufig verwendete Chiffre.

Übung 17. Betrachtet wird die Bit-Folge

$$0101 \quad 1010 \quad 0110 \quad 0001 \quad 0110 \quad 0011 \quad 0110 \quad 1001$$

(a) Stelle diese Bit-Folge hexadezimal dar.

(b) Welchem Wort entspricht die Bit-Folge, wenn man je 1 Byte als ASCII-Code auffasst?

(c) Wende auf die Bit-Folge die Permutation P des DES an.

Übung 18. Das DES besitzt vier sog. „schwache Schlüssel“. Dies sind 64-Bit-Schlüssel, aus denen DES bei der Erzeugung der sechzehn 48-Bit-Schlüssel sechzehn Mal denselben Schlüssel generiert. Finde die vier schwachen Schlüssel. Hinweis: Alle Bytes der schwachen Schlüssel haben Parität 0. Es sind also eigentlich gar keine zugelassenen Schlüssel.

Übung 19. Bei Triple DES werden der erste und der dritte Schlüssel identisch gewählt.

(a) Welchen Grund mag diese Einschränkung haben?

(b) In nicht zu ferner Zukunft schafft man es vielleicht, das für einen Brute-Force Angriff auf das gewöhnliche DES notwendige Ausprobieren aller Schlüssel in einer Stunde durchzuführen. Wie lange braucht man dann (in Jahren) für einen Brute-Force-Angriff auf Triple DES?

Übung 20. Eine vermindernde Permutation führt zu einem Informationsverlust, da zwei unterschiedliche Bit-Folgen auf das selbe Ergebnis abgebildet werden können.

(a) Gib für die vermindernde Permutation $(- \ 2\ 4\ 1\ 3 \ -)$ zwei 6-Bit-Folgen an, die auf dasselbe Ergebnis abgebildet werden.

(b) Ergibt sich diese Problematik auch bei erweiternden Permutationen?

(c) Das DES ist nur sinnvoll, wenn verschiedene Bit-Folgen auch verschieden verschlüsselt werden (warum eigentlich?). Erläutere, warum dies tatsächlich so ist, obwohl im DES vermindernde Permutationen eingesetzt werden.

Die obigen Übungen stammen von der Site

`MathePrisma DES`

Diese führt auch durch ein schönes Lernmodul.

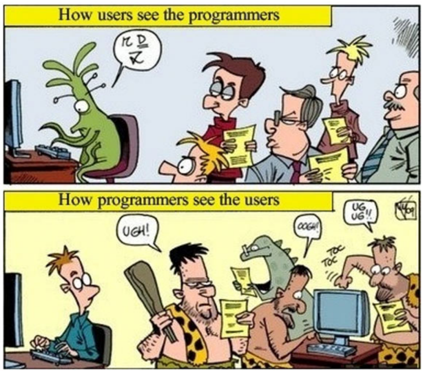


Abbildung 12: Programmers & Users

6. RSA

6.1. Asymmetrie

Wird der Schlüssel beim Austausch abgefangen, so ist die gesamte Verschlüsselung wertlos. Lange Zeit galt diese Sicherheitslücke als unvermeidbar. Es gibt aber auch Verschlüsselungen, die ganz ohne Schlüsselaustausch auskommen. Wie geht das?

Wir denken uns für eine erste Idee, dass eine Kiste geschickt wird, die man mit einem Vorhängeschloss verschliessen kann. Dabei wird zwar die Nachricht mehrfach hin und her geschickt, es wird aber kein Schlüssel ausgetauscht; das geht so:

1. Alice verschliesst die Kiste mit ihrem Schloss und schickt sie an Bob.
2. Bob verschliesst die Kiste ein zweites mal mit seinem Schloss und schickt sie Alice zurück.
3. Alice entfernt ihr Schloss mit ihrem Schlüssel und schickt die Kiste wieder an Bob.
4. Bob kann die Kiste mit seinem Schlüssel öffnen.

Übung 21. Das Verfahren geht natürlich einfacher, ohne die Kiste hin und her schicken zu müssen. Nämlich ...

Bob muss also nur dafür sorgen, dass Alice ein Schloss von ihm hat. Das ist kein Problem, denn verschliessen kann man ja ohne Schlüssel. Aber nur Bob kann mit dem Schlüssel die Kiste wieder öffnen.

6.2. Einwegfunktionen

Eine Einwegfunktion ist eine Funktion, die einfach zu berechnen aber sehr schwer umkehrbar ist. Im täglichen Leben findet man viele „Einwegfunktionen“:

- sich ein Bein brechen
- Zahnpasta aus der Tube drücken

Definition 6.1: one-way function

Eine one-way-function ist eine injektive Funktion

$$f : X \to Y$$

so dass gilt:

- Es gibt ein effizientes Verfahren zur Bestimmung von $y = f(x) \quad \forall x \in X$.
- Die Umkehrung ist praktisch unmöglich, d.h. es gibt kein effizientes Verfahren zur Bestimmung von $x = f^{-1}(y) \quad \forall y \in f(X)$.

Bemerkung. Das heisst nicht, dass eine Einwegfunktion nicht umkehrbar ist, die Umkehrung ist nur so schwierig, dass sie praktisch nicht umzusetzen ist. Das bedeutet dann aber, dass eine Verschlüsselung durch eine Einwegfunktion weder von Unbefugten geknackt, noch vom rechtmässigen Empfänger der Nachricht entschlüsselt werden kann. Man braucht also etwas anderes: Einwegfunktionen mit Falltür.

Definition 6.2: one-way-trap-door function

Eine trap-door one-way-function ist eine injektive Funktion $f : X \to Y$ für die gilt:

- Es gibt effiziente Verfahren zur Berechnung von $y = f(x) \quad \forall x \in X$ und $x = f^{-1}(y) \quad \forall y \in f(X)$.
- Das Verfahren zur Berechnung von f^{-1} kann nicht aus dem Verfahren zur Berechnung von f hergeleitet werden. Man benötigt eine (geheime) Zusatzinformation.

Beispiele. Einwegfunktionen mit Falltür aus dem Leben gegriffen:

- Vorhängeschloss zuschnappen lassen
- Brief in einen Briefkasten werfen.

Die entscheidende Frage für uns ist also, wie man nun eine konkrete Einwegfunktion mit Falltür findet oder erfindet, um eine Nachricht zu verschlüsseln.

Abbildung 13: Shamir, Rivest und Adleman

6.3. Idee von RSA

Wir wissen, dass das Produkt zweier Primzahlen einfach zu berechnen ist. Wie sieht es mit der Umkehrung aus?

Übung 22. Bestimme die Primfaktorzerlegung von

(a) 21

(b) 65

(c) 14803

(d) 12863273

Je grösser die Zahl, desto aufwändiger kann es werden, ihre Primfaktorzerlegung zu bestimmen. Kann es werden, denn dies muss nicht uneingeschränkt sein.

Übung 23. Bestimme die Primfaktorzerlegung von 2 469 135 782.

†bung 23 ist einfach, weil einer der beiden Primfaktoren sehr klein ist. Im Allgemeinen gilt aber, dass das Multiplizieren zweier grossen Primzahlen eine Einwegfunktion mit Falltür ist. Darauf beruht das bekannteste asymmetrische Verschlüsselungsverfahren RSA. Wählt nämlich Bob zwei grosse Primzahlen p und q und hält diese geheim, so kann er das Produkt $N = p \cdot q$ veröffentlichen ohne in Gefahr zu laufen, dass jemand die Primfaktoren p und q bestimmen kann. Den Namen hat das Verfahren von Ronald Rivest, Adi Shamir und Leonard Adleman, den drei Männern, die es 1977 erfunden haben. RSA ist ein asymmetrisches Verfahren, es gibt also einen public key und einen private key.

Beide werden über Primzahlen gebildet. Ausserdem geht in das Verfahren die Modulo-Rechnung ein.

Für die Erzeugung der beiden Schlüssel wählt man zunächst zwei grosse Primzahlen p und q. Diese werden multipliziert und man erhält $N = p \cdot q$. Diese Zahl N sowie eine weitere (fast) beliebig wählbare Zahl e (encipher = verschlüsseln) bilden den public key. Eine dritte Zahl d (decipher = entschlüsseln), zu deren Berechnung wir später kommen, ist der private key. Verschlüsselt wird durch

$$C \equiv M^e \mod N.$$

Die Zahlen N und e sind der public key. M ist der Klartext (Message), C der verschlüsselte Text (Cipher).

Übung 24. Nehmen wir an, du willst an Alice die Nachricht L schicken. Alices public key sei $(187, 7)$, also $N = 187$ und $e = 7$. Das L lautet in ASCII 01001100, das entspricht der Dezimalzahl 76. Die Nachricht lautet also ....

Zum Entschlüsseln benötigt man den geheimen Schlüssel d. Entschlüsselt wird damit durch

$$M \equiv C^d \mod N$$

Übung 25. Und umgekehrt? Angenommen, dein öffentlicher Schlüssel ist $(187, 7)$. Du erhältst die damit verschlüsselte Nachricht $C = 142$. Um diese zu entschlüsseln, benötigst du deinen private key. Der ist in diesem Fall $d = 23$. Wie d berechnet wird, kommt später.

Übung 26. Probier es mit dem Taschenrechner aus obiger Aufgabe aus. Verschlüssele verschiedene Zahlen und entschlüssele sie direkt wieder. Versuche Zahlen, die kleiner sind als $N = 187$ und Zahlen, die grösser sind!

Bemerkung. Das funktioniert allerdings nur, wenn die Nachricht M (als Zahl) kleiner ist als N. Das ist allerdings keine Einschränkung. Will man eine Nachricht verschlüsseln, deren Zahlenwert zu gross ist, so teilt man sie in kleinere Blöcke und verschlüsselt jeden Block einzeln.

Beispiel 6. Um es einfach zu halten, codieren wir: Leerzeichen = 00, A = 01, B = 02, ..., Z = 26. Mit dem Schlüssel (N, e) = (2773, 17) wird die Nachricht:

05 18 18 01 18 05 00 08 21 13 01 14 21 13 00 05 19 20

in 3er-Blöcke geteilt (die sind sicher kleiner als N):

051 818 011 805 000 821 130 114 211 300 051 920

und verschlüsselt zu (mit führenden Nullen auf vier Stellen aufgefüllt):

2236 2000 1725 0542 0000 0436 1988 1684 1064 0567 2236 0948

Nun zur Frage, wie man den private key d berechnet.

Bemerkung. Der private key d errechnet sich aus der Gleichung

$$e \cdot d \mod (p-1)(q-1) = 1$$

Grundlage zur Berechnung von d ist folgender Satz, der garantiert, dass es eine solche Zahl überhaupt gibt — vorausgesetzt die Zahl e hat eine bestimmte Eigenschaft...

Satz 6.1: Existenz eines modular Inversen

Sind $a, b \in \mathbb{Z}$ teilerfremd, so gibt es eine ganze Zahl c, so dass

$$b \cdot c \mod a \equiv 1$$

Definition 6.3: Modular Inverses

Mit den Bezeichnungen aus obigem Satz sagt man, b ist modulo a invertierbar und nennt c die modular Inverse von b.

Übung 27. Finde das modular Inverse Element c zu 3 modulo 5.

In unserem Fall sind wir gerade daran interessiert e modulo $(p-1)(q-1)$ zu invertieren. Das heisst insbesondere, dass e und $(p-1)(q-1)$ teilerfremd sein müssen.

Und wie berechnet man die modulare Inverse nun? Ein Verfahren für diese Berechnung ist in dem Beweis zu obigem Satz versteckt. Deswegen folgt er hier ausführlich. Er erfolgt in drei Schritten:

1. Mit dem euklidschen Algorithmus zur Bestimmung des ggT

2. Summendarstellung des ggT von von entsprechenden Vielfachen der beiden Zahlen (erweiterter Euklidscher Algorithmus)

3. Betrachtung des Falls, dass a und b teilerfremd sind, d.h. dass ihr ggT 1 ist.

Wir wissen bereits, wie man den ggT von zwei Zahlen mit dem Euklidschen Algorithmus bestimmt. Der ggT ist der letzte, nichttriviale Rest.

Übung 28. Schreibe allgemein das Verfahren des Euklidschen Algorithmus für zwei Zahlen a und b, sagen wir bis zu Schritt fünf, auf.

Rechnet man nun rückwärts und ersetzt die Reste durch entsprechende Ausdrücke, so erhält man eine Vielfachendarstellung von a und b für ihren ggT.

Übung 29. Zeige dies.

Satz 6.2: Satz von Bézout

Zu zwei Zahlen $a, b \in \mathbb{Z}$ gibt es immer Zahlen $x, y \in \mathbb{Z}$, so dass

$$\mathrm{ggT}(a, b) = x \cdot a + y \cdot b.$$

Beweis. Damit ist Schritt drei nur noch Formsache. Für teilerfremde a, b gilt jetzt

$$1 = x \cdot a + y \cdot b$$

Dies gilt natürlich auch noch modulo a bzw. b, da $x \cdot a \mod a = 0$ ist.

$$x \cdot a + y \cdot b \mod a = y \cdot b \mod a = 1.$$

Also ist $c = y$ modulo a invers zu b. □

Bemerkung. Dieses c lässt sich mit dem Euklidschen Algorithmus berechnen.

Hat man also die Vielfachensummendarstellung

$$1 = \mathrm{ggT}((p-1)(q-1), e) = x \cdot (p-1)(q-1) + y \cdot e$$

bestimmt, so ist die kleinste positive Zahl der Form

$$y + k \cdot (p-1)(q-1)$$

gleich dem private key d.

Bemerkung. Mit Hilfe des euklidischen Algorithmus' kann man also den private key d bestimmen. Kennt man die beiden Primzahlen p und q, so ist es also einfach, d zu berechnen und damit dann die mit N und e verschlüsselten Nachrichten zu entschlüsseln. Allerdings ist es nahezu unmöglich, d zu berechnen, ohne p und q zu kennen.

Aber warum funktioniert das Verfahren denn nun überhaupt? Warum kommt wirklich wieder der Klartext heraus, mit anderen Worten: warum gilt

$$C^d \mod N = M?$$

Wir erinnern uns, wie der Cipher-Text entstanden ist,

$$C^d \mod N = (M^e)^d \mod N = M^{e \cdot d} \mod N.$$

und können daher die Frage umformulieren: gilt tatsächlich

$$M^{e \cdot d} \mod N = M?$$

Die Antwort auf diese Frage lautet natürlich ja. Für den Beweis brauchen wir den kleinen Satz von Fermat, der ein Spezialfall des Satzes von Euler ist. Zur einfachen Formulierung des Satzes brauchen wir die sogenannte Euler'sche Φ-Funktion.

Definition 6.4: Euler'sche phi-Funktion

Für eine natürliche Zahl n ist $\Phi(n)$ die Anzahl der natürlichen Zahlen $\leq n$, deren grösster gemeinsamer Teiler mit n gleich 1 ist. Man nennt Φ die Euler'sche Φ-Funktion.

Satz 6.3: Euler-phi einer Primzahl

Ist p prim, dann gilt

$$\Phi(p) = p - 1.$$

Beweis. Da p teilerfremd zu $1, 2, 3, \ldots, p - 1$ ist folgt der Satz. □

Satz 6.4: Satz von Euler

Sind $m, n \in \mathbb{N}$ teilerfremd, so gilt

$$m^{\Phi(n)} \mod n = 1.$$

Ein interessanter Spezialfall, den man auch zum Beweis der Korrektheit der Entschlüsselung verwenden kann, ist

Satz 6.5: Multiplikativität der Euler'schen phi-Funktion

ind p, q zwei verschiedene Primzahlen, so ist

$$\Phi(p \cdot q) = (p-1)(q-1).$$

Ist m teilerfremd zu $p \cdot q$, so gilt nach dem Satz von Euler

$$m^{(p-1)(q-1)} \mod pq = 1.$$

Satz 6.6: Kleiner Satz von Fermat

Sei m eine natürliche Zahl und teilerfremd zur Primzahl p. Dann gilt

$$m^{p-1} \mod p = 1.$$

Der Beweis, dass die Entschlüsselung korrekt ist, geht damit so: Wir wissen, dass

$$e \cdot d \mod (p-1)(q-1) = 1.$$

Das heisst, es gibt ein $k \in \mathbb{Z}$ mit

$$e \cdot d = k \cdot (p-1)(q-1) + 1.$$

Damit

$$\begin{aligned} M^{e \cdot d} - M \mod p = M^{k \cdot (p-1)(q-1)+1} - M \mod p = (M^{(p-1)})^{k(q-1)} \cdot M - M \mod p = \\ = 1^{k(q-1)} \cdot M - M \mod p = 0. \end{aligned}$$

Dies gilt insbesondere auch, wenn p ein Teiler von M ist, denn dann ist $M^{e \cdot d} - M \mod p$ sowieso gleich 0. Ferner sieht man analog, dass $M^{e \cdot d} - M \mod q = 0$. Die Primzahlen p und q teilen also dieselbe Zahl $M^{e \cdot d} - M$, also muss auch ihr Produkt diese Zahl teilen. Somit gilt:

$$M^{ed} \mod pq = M.$$

6.4. RSA knacken

Zum Schluss noch ein Beispiel, wie RSA aus Sicht des Codeknackers aussieht. Um es schon mal vorwegzunehmen: es ist ein Beispiel, in dem es dem Codeknacker leicht gemacht wird. Also keine Angst, so schnell wie hier lässt sich RSA nicht knacken...

Beispiel 7. Sie haben eine RSA-verschlüsselte Nachricht abgefangen. Sie lautet

10473054210223505497035330523101828168
2305497093070549713322042531164705144

Ausserdem wissen Sie für wen die Nachricht bestimmt ist. Der öffentliche Schlüssel des Empfängers ist

$$N = 17947, \quad e = 21$$

Um die Nachricht zu entschlüsseln, müssen Sie herausfinden, aus welchen beiden Primzahlen p und q sich N zusammensetzt. Eine erste Möglichkeit, p und q herauszufinden ist, bei 2 anzufangen und jede Zahl zu testen, ob sie ein Teiler von N ist. Dabei kann man sich die Arbeit einfacher machen, wenn man bedenkt, dass

- nicht p und q beide grösser als $\sqrt{N}$ sein können (man braucht also „nur“ die Zahlen bis $\sqrt{N}$ zu testen)
- die beiden gesuchten Teiler von N Primzahlen sind (man braucht also z.B. die geraden Zahlen gar nicht testen).

Allerdings dauert diese Suche um so länger je grösser p und q sind. Es scheint also geschickter zu sein, die Suche bei möglichst grossen Zahlen anzufangen, d.h. bei $\sqrt{N}$. Wenn Sie die Zahlen p und q herausgefunden haben, können Sie nun ganz normal den privaten Schlüssel bestimmen.

6.5. übung zu RSA mit Mathematica

Zum Schluss noch ein etwas grösseres Beispiel. Alice wöhle als p Primzahl Nummer 1000000000 und als q Primzahl Nummer 1000005000. Der Befehl Prime[.] des Programms Mathematica liefert ihr

$$p = 22801763489, \quad q = 22801881559.$$

Daraus folgt

$$n = pq = 519923110412508599351.$$

Nun muss sie die Zahlen c und d so bestimmen, dass d zu c invers Modulo $(p-1)(q-1)$ ist. Es ist $r = (p-1)(q-1) = 519923110366904954304$. Sie wählt

$$c = 4699873.$$

Mithilfe des Mathematica-Befehls GCD[c, r], der den grössten gemeinsamen Teiler der beiden Zahlen c und r bestimmt, erhält man GCD[c, r] = 1, das heisst c und r sind teilerfremd. Der Mathematica-Befehl PowerMod[c,-1,r] liefert für d

$$d = 252883827627895253473.$$

Tatsächlich ergibt die Division

$$\frac{cd-1}{r} = k = 2285957.$$

Alice veröffentlicht nun in der Zeitung die beiden Zahlen

$$n = 519923110412508599351, \quad c = 4699873.$$

Ihr Freund Bob hat nur darauf gewartet und beschliesst, ihr die folgende Botschaft zu schicken

`RSA IST EIN GENIESTREICH`

Diesen Text übersetzt er zuerst nach ASCII in einen Block von Zahlen. Für die Grossbuchstaben A, B, C, ... verwendet ASCII der Reihe nach die Zahlen 65, 66, 67, ... Ein Leerschlag erhält die Zahl 32. Das ergibt folgenden Zahlenstring

$$828365327383843269737832716978736983848269736772$$

Bob bildet daraus drei Zahlen m_1, m_2, m_3

$$m_1 = 828365327383843269 73, m_2 = 78327169787369838482, m_3 = 69736772,$$

die alle kleiner als n sind. Weil

$$\text{PowerMod}[m, c, n] = m^c \mod n$$

gilt, kann Bob mit Mathematica leicht

$$\overline{m_1} = m_1^c \mod n = 479529267290958755149$$

berechnen, und ebenso

$$\overline{m_2} = m_2^c \mod n = 428935216287316816132,$$

$$\overline{m_3} = m_3^c \mod n = 135766055009022893446.$$

Diese drei Zahlen veröffentlicht er in der Zeitung, in der sie Alice alsbald entdeckt. Sie berechnet nun, ebenfalls mit dem Mathematica-Befehl PowerMod sukzessive PowerMod$[m_1, d, n]$, PowerMod$[m_2, d, n]$, PowerMod$[m_3, d, n]$ und findet die drei Zahlen

$$82836532738384326973,$$

$$78327169787369838482,$$

$$69736772.$$

Indem sie den ASCII Code rückwärts übersetzt, findet sie Bobs Mitteilung

`RSA IST EIN GENIESTREICH`

und ist enttäuscht: So eine Banalität hätte er nun wirklich nicht zu verschlüsseln brauchen!

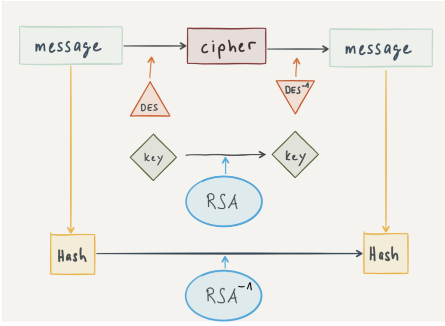


Abbildung 14: PGP Schema

7. Pretty Good Privacy

PGP benutzt ein sogenanntes Public-Key-Verfahren, in dem es ein eindeutig zugeordnetes Schlüsselpaar gibt:

Genutzt wird ein öffentlicher Schlüssel, mit dem jeder Daten für den Empfänger verschlüsseln und dessen Signaturen prüfen kann, und ein privater geheimer Schlüssel, den nur der Empfänger besitzt und der normalerweise durch ein Passwort geschützt ist. Nachrichten an einen Empfänger werden mit dessen öffentlichem Schlüssel verschlüsselt und können dann ausschliesslich mittels seines privaten Schlüssels entschlüsselt werden. Diese Verfahren werden auch asymmetrische Verfahren genannt, da Sender und Empfänger zwei unterschiedliche Schlüssel verwenden.

Die erste Version wurde im Jahr 1991 geschrieben und verwendete einen RSA-Algorithmus zur Verschlüsselung der Daten.

Bei PGP wird aber nicht die ganze Nachricht asymmetrisch verschlüsselt, denn dies wäre viel zu rechenintensiv und es wäre nicht praktikabel, dieselbe Nachricht an mehrere Empfänger zu schicken. Stattdessen wird die eigentliche Nachricht symmetrisch und nur der verwendete Schlüssel asymmetrisch verschlüsselt (Hybride Verschlüsselung). Dazu wird jedes Mal ein symmetrischer Schlüssel (session key) zufällig erzeugt.

Dieser symmetrische Schlüssel wird dann z.B. per RSA- oder Elgamal-Kryptosystem mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und der Nachricht hinzugefügt. Dadurch ist es möglich, eine Nachricht für mehrere Empfänger gleichzeitig zu verschlüsseln.

(Quelle: Wikipedia, 2020)

Schematisch habe ich dies so wie in Abbildung 14 auf Seite 57 illustriert. Bei mir hier im Unterricht simulieren wir das Verfahren mit RSA und verwenden als Hash/Fingerprint die check-digit aus der ISBN-10 Nummer aus dem Abschnitt Barcode.

A. Memorandum Bombes

TOP SECRET

Op-20-G/ev

TOP SECRET

Bombe History
X Bombe N-530
Lab Bldg
Dunning
50 added Bombes

24 March 1944.

MEMORANDUM FOR Op-20-G-1:

Subj: Bombes - History of.

1. The original project for Bombe construction was approved by Rear Admiral J. R. Redman and Vice Admiral F. J. Horne on 4 September 1942. The original directive did not specify the number of units to be provided.

2. Efforts during the latter months of 1942 were directed principally at effective design of the proposed equipment. In the original concept of the problem, it appeared that 336 units were desirable since there are 336 possible wheel orders which could then be run simultaneously. In forming preliminary notions of the size and power requirements of the equipment, the British three-wheel Bombe was strongly in mind. The British three-wheel Bombes have three levels in each physical piece of equipment. Thus the concept of 336 Bombes led to the notion of 112 separate pieces of equipment slightly larger than the British Bombes. The Laboratory Building was thus designed to take 112 such units. Preliminary designs for the building were approved in January of 1943.

3. Dr. Turing on his visit to Op-20-G in December of 1942 entered discussions concerning the number of Bombes to be built. In view of technical conditions limiting wheel order choices, the original notion of running one Bombe for each wheel order was modified. At about the same time designs on the two experimental American Bombes were nearing completion. Due to other requirements of automatic switching and the high speeds necessary to do the four-wheel job, following the British pattern of three banks per unit proved impracticable and our machines were designed with one Bombe in each unit.

4. Discussions as to the number of Bombes required for the Naval problem reached a conclusion in March of 1943 that ninety-six Bombes was the optimum number for the Naval problem. These conclusions have been borne out by experience. Furthermore, more than 100 production units would have overtaxed the manufacturing facilities.

TOP SECRET

24 March 1944.

Subj: Bombes - History of.

5. Production models of the Bombes were put in operation in August of 1943. Early in September of 1943 considerations again centered around the total number of Bombes to be produced. British views were invited on this subject on 7 September. They replied as follows:

> "In our view, present favorable SHARK position may not continue. Also your assistance in non-SHARK jobs may be very valuable. We should therefore be sorry if your production figure were reduced."

The German Naval position (SHARK) has remained favorable for the past six months. At present all units are used for Naval jobs until the keys are out. The machines are then put on non-Naval research. During this period about 45% of the Bombe time has been devoted to these non-Naval problems.

6. Since 1 February 1944 there has been some deterioration in the Naval cribbing situation. However, present equipment has handled the traffic adequately.

7. While the fifty additional Bombes will result in a 33-1/3% decrease in time required for pulling SHARK keys, their major contribution will be in the direction of other problems.

8. Introduction by the Germans on certain air circuits of a pluggable reflector has caused some concern. A machine for breaking both the external and reflector plugging has been designed and construction started. While this new machine (DUENNA) has similarity to a Bombe, it must be considered a separate device. In view of the possibility of introduction of this device on more keys and in view of the general German situation, it is considered that 150 Bombes will be completely adequate. Additional equipment will probably be necessary in the form of DUENNAS or other special devices.

Respectfully,

H. T. Engstrom

H. T. ENGSTROM,
Op-20-GM.

Abbildungsverzeichnis

1. Semagramm mit Morsecode (aus Bauer) 2
2. Caesar-Scheibe zum Ver- und Entschlüsseln 3
3. Häufigkeitsgebirge Deutsch & Englisch 15
4. Die ENIGMA 26
5. Prinzipieller Aufbau der ENIGMA: (1) Batterie, (2) Tastatur, (3,7) Steckbrett mit (8) Stecknabel, (5) Walzensatz mit (4) Eintrittswalze und (6) Umkehrwalze sowie dem (9) Lampenfeld 27
6. Die Turingbombe 29
7. Auszug aus ASCII binär 31
8. Auszug aus ASCII hexadizimal 33
9. Feistel-Round 37
10. F-Modul 40
11. S-Box 40
12. Programmers & Users 46
13. Shamir, Rivest und Adleman 49
14. PGP Schema 57

Tabellenverzeichnis

1. Caesar-Alphabet mit Translation 3 3
2. Monoalphabetische Verschlüsselung mit Faktor 3 6
3. Monoalphabetische Verschlüsselung mit Schlüsselwort und Schlüsselbuchstabe 7
4. Beispiel einer monoalphabetische Verschlüsselung mit Schlüsselwort `JAMES BOND` und Schlüsselbuchstabe `q` 7
5. Relative Deutsche Buchstabenhäufigkeiten von Mono- und Bigrammen . . 9
6. Bigramm-Häufigkeiten deutsch und englisch 12
7. Trigramm-Häufigkeiten deutsch und englisch 13
8. Wortlängen von Sprachen 13
9. Zehn häufigste Wörter von Sprachen 14
10. Koinzidenzindex κ von Sprachen 15
11. Polyalphabetische Chiffrierung 17
12. Vigenère-Chiffrierung 17
13. Vigenère-Verschlüsselung 18
14. Vigenère-Verschlüsselung mit `JAMESBOND` 19
15. Idee Kasiski-Test 19
16. Kasiski-Test 20
19. XOR 35