



Kryptologie

Von Caesar bis RSA

gym | LERBERMATT
fms



Inhaltsverzeichnis

1. Geschichtliches zur Kryptologie	5
1.1. Skytale	5
1.2. Steganographie	5
1.3. Caesar Cipher	7
1.4. Polyalphabetische Verschlüsselung	7
2. Monoalphabetische Verfahren	9
2.1. CAESAR-Cipher	9
2.2. Tauschchiffre	11
2.2.1. Ein praktisches Beispiel	12
2.3. Das Entschlüsseln monoalphabetischer Texte	13
2.3.1. Ein weiteres Beispiel	15
2.4. Hilfsmittel zur Ver- & Entschlüsselung	16
2.4.1. Die Muster einer Sprache	16
2.4.2. Die Abhängigkeit der Häufigkeiten vom Klartext	17
2.4.3. Der Wortzwischenraum	17
2.4.4. Häufigkeiten von n-Grammen	17
2.5. Die Abhängigkeit von der Verfassersprache	17
2.5.1. Häufigkeitsgebirge	17
2.5.2. Der Koinzidenzindex kappa einer Sprache	19
3. Die Idee der polyalphabetischen Chiffrierung	22
3.1. Vigenère-Chiffrierung	22
3.2. Ein praktisches Beispiel	23
3.3. Die Theorie der Entschlüsselung	24
3.3.1. Der Kasiski-Test	24
3.3.2. Der Friedmann-Test	25
3.3.3. Die Bestimmung des Schlüsselwortes	26
3.4. Ein weiteres Beispiel	28
4. Die Enigma	32
4.1. Geschichte	32
4.2. Prinzip	32
4.3. Aufbau	33
4.4. Schlüsselraum	33
4.5. Entzifferung	34
4.6. Lernprogramm	35
5. RSA	36
5.1. Asymmetrie	36
5.2. Einwegfunktionen	36

5.3. Idee von RSA	37
5.4. Das Verfahren	38
5.5. RSA knacken	45
5.6. Übung zu RSA mit Mathematica	46
6. Pretty Good Privacy	48
A. Memorandum Bombes	49
B. Data Encryption Standard	51
B.1. Binärcodes	51
B.2. Erste DES Schicht	55
B.3. F-Modul und S-Module	59
B.4. DES Schlüssel	62
B.5. Sicherheit	63
B.5.1. Bedeutung der Blocklänge	63

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 $\kappa\rho\gamma\pi\tau\omega\sigma$ (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.



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 besitzen. 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, so wurde die Nachricht unlesbar, konnte aber mühelos von einer Person gelesen werden, die einen Zylinder genau desselben Umfangs hatte.

1.2. Steganographie

Auch die **Steganographie**, eine Art verschleierte Geheimschrift, war früh bekannt. Diese konnte entweder als unverfängliche, offen verständliche Nachricht oder in (winzigen) sichtbaren graphischen Details einer Schrift oder Zeichnung übermittelt werden.

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 6).



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

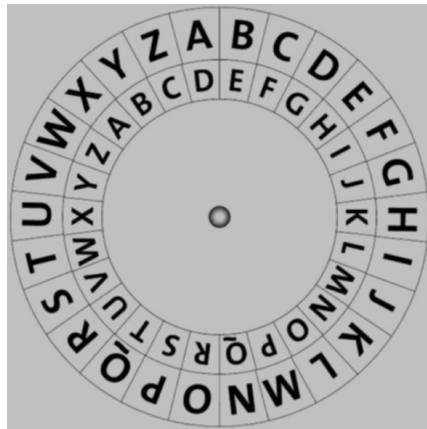


Abbildung 2: Caesar-Scheibe zum Ver- und Entschlüsseln

1.3. Caesar Cipher

JULIUS CAESAR verwendete darüber hinaus eine spezielle Methode der monoalphabetischen Chiffrierung. Er verschob die Buchstaben seines Klartextes um 3 Stellen bezüglich des Alphabets nach links, so dass aus einem Klartext a ein Geheimtext D wurde:



1.4. Polyalphabetische Verschlüsselung

Um 1400 gelang es den Arabern, Substitutionen zu brechen. 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. Eine Maschine, „ENIGMA“ genannt (ENIGMA, griechisch Rätsel, Geheimnis), in der sich Walzen mit eingravierten Buchstaben drehten, ver- und entschlüsselte Nachrichten. Der Empfänger besass eine ähnliche Maschine. Da er den jeweils verwendeten Code kannte, konnte er ihn in

1. Geschichtliches zur Kryptologie

diese Maschine eingeben und erhielt dann durch Tippen des verschlüsselten Textes unmittelbar die entschlüsselte Botschaft zurück. 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.
- 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.

Zu den Kryptoanalytikern der Engländer gehörte unter anderen auch ALAN 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.

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 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 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 Primfaktorzerlegung natürlicher Zahlen.

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

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

mögliche Buchstabenpermutationen.



2.1. CAESAR-Cipher

Zu den einfachsten Chiffren gehört die Verschiebechiffre, die schon von JULIUS 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** Sinn.

Übung 2.1.



Verschlüssle

veni vidi vici

mit Caesar.

Notizen zu Übung 2.1



YHQL YLGL YLFL



Übung 2.2.



Schreibe Code, der eine Message mit einem beliebigen Shift verschlüsselt.



Notizen zu Übung 2.2

Ein Beispiel ist:

```
cipher = ""  
message = "Testmessage-aBYz"  
shift = 1  
for k in range(len(message)):  
    if message[k].islower():  
        cipher = cipher + chr((ord(message[k]) - 97  
        + shift) % 26 + 97)  
    elif message[k] == "-":  
        cipher += "-"  
    else:  
        cipher += chr((ord(message[k]) - 65 + shift)  
        % 26 + 65)  
print(cipher)
```

Übung 2.3.



Entschlüssle den mit Caesar verschlüsselten Text

JDLXV MXOLXV FDHVDU.



Notizen zu Übung 2.3

Mit obigem Programm probiert man verschiedene Shifts durch. So lange, bis der Text sinnvoll scheint. Vielleicht startet man mit der Vermutung, dass vermutlich Shift $-3 \equiv 23 \pmod{26}$ verwendet werden muss. Dies ergibt dann **gaius julius caesar**.

Übung 2.4.



Zu monoalphabetischen Verschlüsselungsverfahren:

- Wie viele verschiedene Transpositionscipher gibt es?
- Wie viele Substitutionscipher gibt es?
- Wieso ist es, trotz der vielen Möglichkeiten, im Allgemeinen leicht eine Substitutionscipher zu knacken?

- d) Wie könnte man die Schwachstelle einer monoalphabetischen Cipher cachieren?

Notizen zu Übung 2.4



- a) Falls man die Identität inkludiert, so hat man 26 mögliche Shifts des Alphabets.
- b) Beginnend mit A hat man 26 mögliche Kandidaten als Substitution. Für den nächsten Buchstaben B sind es dann noch 25, weil einer an A vergeben wurde. So fährt man sinngemäss fort und erhält $26! \approx 4 \cdot 10^{26}$ Möglichkeiten. Das sind sehr sehr viele.
- c) Weil, angenommen dass die Verfassersprache bekannt ist, die für die Sprache charakteristischen Buchstabenhäufigkeiten erhalten bleiben. Sie werden bloss durch andere „Zeichen“ repräsentiert. Beispielsweise ist im Deutschen das E mit einer Häufigkeit von gut 17% dominant. Kennt man das E, so lassen sich häufige Konstellationen (EI, DIE, etc.) erschliessen und anschliessend zusammen mit weiteren, häufigen Buchstaben (z.B. T, S, N, ...) lässt sich der Text schliesslich knacken.
- d) Man versucht die Buchstabenhäufigkeiten zu verschleiern, indem beispielsweise mit wechselnden Shifts gearbeitet wird oder für häufig auftretende Buchstaben entsprechend mehrere Cipher-Zeichen eingesetzt werden.

2.2. Tauschchiffre

Eine weitere Möglichkeit bietet die **Substitutioncipher**. 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

Natürlich können die beiden Verfahren auch miteinander kombiniert werden.

2. Monoalphabetische Verfahren

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 Schlüsselbuchstaben

Häufig wird die Methode des Schlüsselwortes verwendet, d.h. Sender und Empfänger vereinbaren ein Schlüsselwort und einen Schlüsselbuchstaben. 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.

2.2.1. Ein praktisches Beispiel

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:

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», stoehtnt Gatti,
LSS... «WKRX PYTT, ARKGKMKQV», MEYKQXR PFEER,
«kannst du nicht einmal deine Socken anziehen»
«UFXXME IS XRHQE KRXWVF 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 14 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. Monoalphabetische Verfahren

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

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

- 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 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.

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

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

- 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

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

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.

- 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

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

2. Monoalphabetische Verfahren

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

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 KYIPMGHGJFNLGJ
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

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.:

OTTO → 1221
NGRGUUV → 1232445
PANAMAKANAL → 12324252326

Solche Muster bleiben bei der monoalphabetischen Chiffrierung erhalten, d.h. enthalt ein Geheimtext keine Muster, so ist er nicht durch monoalphabetischer Chiffrierung entstanden.

2.4.2. Die Abhangigkeit der Haufigkeiten vom Klartext

Die Angaben uber die Einzelbuchstaben schwanken und hangen ausserdem vom Genre des Textes ab. Haufigkeiten sind um so scharfer, je langer der Text ist.

2.4.3. Der Wortzwischenraum

Sicherlich kommen wir auf die Idee, den Wortzwischenraum (Space) mit zu verschlesseln. Dies fuhrt dann zu einem Alphabet mit 27 Buchstaben. Da der «Space» im Deutschen nach dem e das haufigste «Zeichen» und somit leicht zu enttarnen ist, lsst der professionelle Chiffrierer diesen Zwischenraum einfach weg.

2.4.4. Haufigkeiten von n-Grammen

n -Gramme sind Kolonnen von n Buchstaben. Die folgende Tabelle 6 auf Seite 18 zeigt die Haufigkeiten fur Bigramme im Deutschen und Englischen.

Im Deutschen kommt ch sehr haufig vor, aber nahezu niemals hc. Ferner tauchen ei und ie gleich haufig auf.

In Tabelle 7 finden Sie noch die Haufigkeit einiger Trigramme.

Abschliessend eine Aufzahlung haufiger Viergramme:

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

Aufschluss uber den Ursprung des Textes kann ubrigens auch die mittlere Wortlange geben, oder die 10 haufigsten Worter.

2.5. Die Abhangigkeit von der Verfassersprache

2.5.1. Haufigkeitsgebirge

Jede Sprache hat ihre eigenen Haufigkeiten. Die Diagrammdarstellungen werden **Haufigkeitsgebirge** genannt. Zwei Beispiele sind in Abbildung 3 auf Seite 19 illustriert.

2. Monoalphabetische Verfahren

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

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
franzosisch	4.4	russisch	6.3

Tabelle 8: Wortlangen von Sprachen

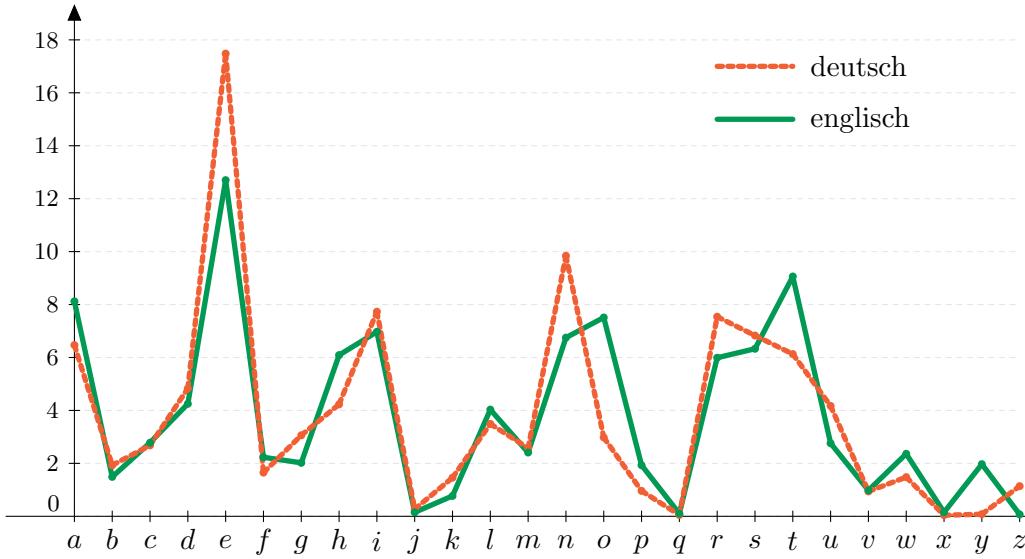


Abbildung 3: Haufigkeitsgebirge Deutsch & Englisch

Durch Haufigkeitsgebirge konnen wir bei geniigend langem monoalphabetisch chiffriertem Geheimtext feststellen, in welcher Sprache er verfasst wurde.

2.5.2. Der Koinzidenzindex κ einer Sprache

Um das κ einer Sprache zu bestimmen, schreiben wir zwei gleichlange unterschiedliche Texte untereinander und zahlen 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. Fur zwei Texte gleicher Lange,

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

definieren wir

2. Monoalphabetische Verfahren

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

Tabelle 9: Koinzidenzindex κ von Sprachen

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

Manchmal ist die Formel

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

praktischer, 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 9 auf Seite 20.

Bemerkung 2.5.1. Kommt jeder Buchstabe (bei einem Alphabet von 26 Buchstaben) mit der gleichen Wahrscheinlichkeit von $\frac{1}{26}$ vor, so ergibt sich

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

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 KIFGHY.
 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 2.5.2. Nach Bauer gilt:

Ein ideal fur die Chiffrierung vorbereiteter Klartext ist orthographisch falsch, sprachlich knapp und stilistisch grauenhaft.

3. Die Idee der polyalphabetischen Chiffrierung

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 10: 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 e n
K S Y S S G T U U T Z X V M U C

Tabelle 11: Vigenère-Chiffrierung

3. Die Idee der polyalphabetischen Chiffrierung

Eine polyalphabetische Chiffrierung kann z.B. für den Klartext **abba** so wie in Tabelle 10 auf Seite 22 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. Die bekannteste Methode der polyalphabetischen Chiffrierung ist die **Vigenère-Chiffrierung**. Zur Verschlüsselung hilft ein sogenanntes **Vigenère-Quadrat**.

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

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 Schlüsselwortbuchstabe ist. Also für den ersten Buchstaben des Beispiels von oben heisst das:

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 12: Vigenère-Verschlüsselung

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 der abgeschlossenen Roman. Das Schlüsselwort sei JAMES-BOND. Wir schreiben zunächst das Schlüsselwort über den Text. Am Schnittpunkt von Zeile und Spalte steht unser Geheimbuchstabe.

Wir erhalten somit den in Tabelle 13 dargestellten Geheimtext.

Übung 3.1.



Verschlüssle die Nachricht kryptologie mit dem Schlüssel waj mit Vigenère.

3. Die Idee der polyalphabetischen Chiffrierung

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

Tabelle 13: Vigenère-Verschlüsselung mit JAMESBOND

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 14: Idee Kasiski-Test

Notizen zu Übung 3.1



GRHLTXHOPEE

3.3. Die Theorie der Entschlüsselung

Die Kryptoanalyse von Vigenère-Chiffrierung ist umso leichter, 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 3.3.1. Wir finden sogar Zehngramme:

UEQPC VCKAH VNRZU RNLAO KIRVG **JTDVR** VRICV IDLMY
IYSBC COJQS ZNYMB VDLOK FSLMW EFRZA **VIQMF** **JTDIH**

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 15: Kasiski-Test

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 JIHЛИ IFWMB ZHUMP LEUWG RBHZO LCKCW THWDS
 ILDAG VNEMJ FRVQS VIQMU VSWMZ CTHII WGDJS XEOWS
 JTKIH KEQ

Man bildet nun den grössten gemeinsamen Teiler der Abstandszahlen (siehe Tabelle 15). In obigen Beispiel wäre dieser $\gcd = 1$, 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 \gcd der Abstände gleich 5. Das legt die Schlüsselwortlänge fünf nahe. Es könnte aber auch sein, dass sich VIQM und MWK zufällig wiederholen, dann wäre $\gcd(50, 90) = 10$. 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:

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

3. Die Idee der polyalphabetischen Chiffrierung

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, dass zwei beliebige Buchstaben in einer Spalte gleich sind, ist κ_K (Klartextkappa), denn die Buchstaben einer Spalte sind ja dann monoalphabetisch verschlüsselt. Die Wahrscheinlichkeit, dass zwei Buchstaben aus verschiedenen Spalten gleich sind, ist $3,85\%+ \geq$, da über verschiedene Alphabete verschlüsselt wurde. Die Wahrscheinlichkeit, dass zwei beliebige Buchstaben des Geheimtextes gleich sind, ist dann

$$\kappa_G = \frac{1}{n} \kappa_K + \left(1 - \frac{1}{n}\right) \cdot (3,85\%+ \geq) = \frac{1}{n} (\kappa_K - 3,85\%- \geq) + 3,85\%+ \geq$$

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

$$n = \frac{\kappa_K - 3,85\%- \geq}{\kappa_G - 3,85\%- \geq}$$

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

$$n = \frac{7.62 - 3,85\%- \geq}{\kappa_G - 3,85\%- \geq} = \frac{3.77\%- \geq}{\kappa_G - 3,85\%- \geq} \leq \frac{3.77}{\kappa_G - 3.85}$$

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:

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



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

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

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

Übung 3.2.



Erkläre, wie der Kasiski- und der Friedmann-Test funktionieren.

Notizen zu Übung 3.2



Beide Tests funktionieren umso besser, je mehr Ciphertext zur Verfügung steht. Beim Kasiski-Test sucht man sich wiederholende Ciphertext-Schnippsel, da in diesem Fall vermutlich ein identischer Wortteil mit demselben Schlüsselteil chiffriert wurde. Dadurch kann man die Schlüsselwortlänge abschätzen, da in solchen Fällen der Abstand ein Vielfaches der Schlüsselwortlänge betragen muss. Ein gemeinsamer Teiler dieser Abstände könnte somit mit der Schlüsselwortlänge korrelieren.

Beim Friedmann-Test schätzt man die Schlüsselwortlänge durch folgende Überlegungen nach oben ab. Zuerst misst man den Koinzidenzindex des Ciphertextes, κ_C . Setzt man für die vermutete Schlüsselwortlänge n an und bricht die Cipher nach jeweils n Zeichen um, so sind im Falle der korrekten Schlüsselwortlänge die einzelnen Spalten monoalphabetisch verschlüsselt. Somit ist die Wahrscheinlichkeit, dass zwei zufällig ausgewählte Buchstaben in der gleichen Spalte stehen $1/n$ und folglich $1 - 1/n$ die Wahrscheinlichkeit, dass zwei zufällig gewählte Buchstaben in verschiedenen Zeilen stehen. Der Koinzidenzindex des Ciphertextes setzt sich also aus einem eher zufälligen und einem durch das Schlüsselwort gegebenen Teil, der denselben Koinzidenzindex wie die Verfassersprache haben muss, zusammen:

$$\kappa_C = \frac{1}{n} \cdot \kappa_D + \left(1 - \frac{1}{n}\right) \cdot (\kappa_r + \varepsilon),$$

wobei $\varepsilon \in \mathbb{R}^+$, $n \in \mathbb{N}$ die Anzahl Spalten, κ_D den Koinzidenzindex der Verfassersprache und κ_r den Koinzidenzindex einer Randomsprache¹ bezeichnen. Es folgt nach Multiplikation mit n und umsortieren

$$n\kappa_C - n\kappa_r - n\varepsilon = \kappa_D - \kappa_r - \varepsilon$$

¹Eine Randomsprache definiere ich hier als eine Kunstsprache, bei der jedes Zeichen zufällig ausgewählt wird. Der Koinzidenzindex dieser Sprache ist offensichtlich $1/26$.

und weiter

$$n = \frac{\kappa_D - \kappa_r - \varepsilon}{\kappa_C - \kappa_r - \varepsilon} \leq \frac{\kappa_D - \kappa_r}{\kappa_C - \kappa_r}.$$

Damit ist die Schlüsselwortlänge n nach oben abgeschätzt.

Übung 3.3.



Schreibe in Python einen kurzen Codeschnippel, der eine Message (in Grossbuchstaben) mit einem Keyword (in Kleinbuchstaben) nach Vigenère verschlüsselt.

Notizen zu Übung 3.3



Ein Beispiel ist:

```
cipher = ""
message = "KASISKI-FRIEDMANN-TEST" # message in Grossbuchstaben, darf
key = "vigenere" # Schluessel in Kleinbuchstaben
keylength = len(key)
caps = 0

for k in range(len(message)):
    if message[k] == "-":
        cipher += "-"
        caps = caps + 1
    else:
        kshift = (k + caps) % keylength
        shift = ord(key[kshift]) - 97
        cipher += chr((ord(message[k]) - 65 + shift) % 26 + 65)

print(cipher)
```

3.4. Ein weiteres Beispiel

Gegeben sei folgender Text:

```
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 DMCKK 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 MOVMIK 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 gcd 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 die Formel

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

3. Die Idee der polyalphabetischen Chiffrierung

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									

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 möglich, dass 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.

Übung 3.4.



Entschlüsseln Sie folgendende Vigenère-Cipher:

PEK MDIXKYAGU GNW KQR DHEILRU TXZF KHLZNXU GNTITAXUSIZ CANXPZAGKQR
XPZGXZQTSA IEKKQN YBQR ULUDX AQSMZ USM LE VHU HOKAQIE DQNG KQR VPBHXYFEQA
XAGN USM TAEZSUCAZF WXUUGX MQHELD EGATAKSF UGK SRTTYAMPWAEPICA RMTTZFRHTAEC
PET SBPEF ZBIXSF DBL XAXUSE WLE GXDMEASFEG ZOHEBQSLLXWHYFEL BZD TBOH
WLESXU CUTSUTTLF EBUQ WBJTTBNQ RHSXE PPQ MTU EIVO XEBJTT WLZKXU WAGU
RAESE SBL PIXZQN KLXAMPH KNYLEG AQXM OMBXU WNTJWEG RAEGUQN LV SRTAGLBLDE
BJT HXYLLBJT

Notizen zu Übung 3.4



Der Schlüssel ist math. Damit lautet der Text:

DER FRIEDMANN UND DER KASISKI TEST KOENNEN UNABHAENGIG VONEINANDER
EINGESETZT WERDEN FUER BEIDE TESTS IST ES VON VORTEIL WENN DER CIPHERTEXT
LANG IST MOEGLICHST WENIGE FEHLER ENTHAELT UND GRAMMATIKALISCH KATASTROPHAL
IST ZUDEM SPIELT DIE LAENGE DES GEWAEHLTEN SCHLUESSELWORTES UND AUCH
DESSEN QUALITAET EINE WICHTIGE ROLLE WIE MAN SICH LEICHT DENKEN KANN
FALLS SIE DIESEN RELATIV KURZEN TEXT HABEN KNACKEN KOENNEN SO GRATULIERE
ICH HERZLICH

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 $\alpha\iota\nu\gamma\mu\alpha$ bedeutet «Rätsel».



4.1. Geschichte

Nach dem Ersten Weltkrieg suchten die deutschen Militärs nach einem Ersatz für die inzwischen veralteten, umständlichen und unsicheren manuellen Verschlüsselungsverfahren, die bis dahin verwendet wurden. Hierfür kamen maschinelle Verfahren in Betracht, weil sie eine einfachere Handhabung und eine verbesserte kryptographische Sicherheit versprachen.

Als Erfinder der Enigma gilt der promovierte deutsche Elektroingenieur ARTHUR SCHERBIUS (1878–1929) dessen erstes Patent hierzu vom 23. Februar 1918 stammt.

4.2. Prinzip

Die ENIGMA I inklusive Holzgehäuse wiegt rund 12 kg und die äusseren Abmessungen (Länge × Breite × Höhe) 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 Grossbuchstaben 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 fliesst elektrischer Strom 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.



Abbildung 4: Die ENIGMA

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. Der Strom verlässt den Walzensatz 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 schliesslich 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.

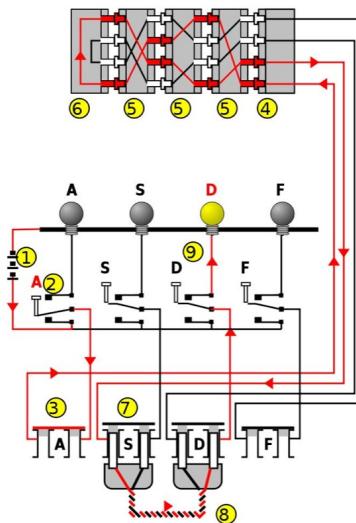


Abbildung 5: Prinzipieller Aufbau der ENIGMA: (1) Batterie, (2) Tastatur, (3,7) Steckbrett mit (8) Stecknabel, (5) Walzensatz mit (4) Eintrittswalze und (6) Umkehrwalze sowie dem (9) Lampenfeld

4.5. Entzifferung

Die Betreiber der Schlüsselmaschine ENIGMA waren der Meinung, dass die durch sie maschinell verschlüsselten Texte mit manuellen Methoden nicht zu knacken sind. Was übersehen wurde, ist, dass einer maschinellen Verschlüsselung durch maschinelle Entzifferung begegnet werden kann.

Dem 27-jährigen polnischen Mathematiker MARIAN REJEWSKI glückte bei seiner Arbeit in der polnischen Dechiffierstelle 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 einen schwerwiegenden verfahrenstechnischen Fehler aus, der den Deutschen unterlief. Um eine sichere Übertragung zu gewährleisten, wurde zu dieser Zeit der Spruchschlüssel noch zweimal hintereinander gestellt und verschlüsselt an den Anfang einer Nachricht geschrieben. Mit Hilfe zweier

speziell zu diesem Zweck gebauter Maschinen, genannt Zykrometer und Bomba, konnten die polnischen Codeknacker so den Suchraum gewaltig einengen.

Nachdem die Deutschen 1938 ihre Verfahrenstechnik änderten und kurze Zeit 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.

Damit 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 eine spezielle elektromechanische Maschine, genannt die Turing-Bombe, die auf der polnischen Bomba aufbaute und vom englischen Mathematiker ALAN TURING ersonnen wurde.

Mit Hilfe der Turing-Bombe brauchte man nur noch rund sechs Stunden, um sämtliche Möglichkeiten durchzutesten.

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. 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.

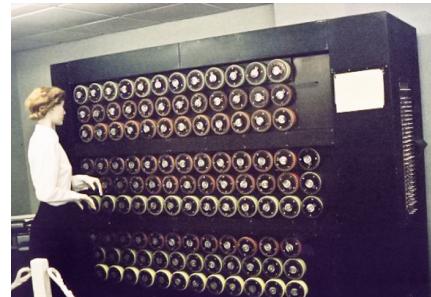


Abbildung 6: Die Turingbombe

4.6. Lernprogramm

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

[MathePrisma Enigma](#)

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

5. RSA

5.1. 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 abschliessen kann. Dabei wird zwar die Nachricht mehrfach hin und her geschickt, es wird aber kein Schlüssel ausgetauscht. Das geht so, wenn Alice Bob eine verschlüsselte Nachricht schicken möchte:



1. Alice verschliesst die Kiste mit ihrem Schloss und schickt sie 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 5.1.



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

Notizen zu Übung 5.1



Bob könnte Alice beispielsweise ein offenes Vorhängeschloss schicken, dessen Schlüssel er besitzt.

5.2. Einwegfunktionen

Eine Einwegfunktion ist eine Funktion, die einfach zu berechnen aber sehr schwer umkehrbar ist.

Definition 5.1: One-Way Function

Eine One-Way-Function ist eine injektive Funktion

$$f : X \rightarrow Y$$

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 5.2.1. Das heisst nicht, dass eine Einwegfunktion nicht umkehrbar ist, die Umkehrung ist nur so schwierig, dass sie sehr schwer umzusetzen ist. Man braucht also etwas anderes: Einwegfunktionen mit Falltür.

Definition 5.2: Trap-Door One-Way-Function

Eine trap-door one-way-function ist eine injektive Funktion $f : X \rightarrow Y$ 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.

5.3. Idee von RSA

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

Übung 5.2.

Bestimme die Primfaktorzerlegung von

- a) 21
- b) 65

c) 14803

d) 12863273

Notizen zu Übung 5.2 $3 \cdot 7, 5 \cdot 13, 113 \cdot 131$ und $3259 \cdot 3947$

Je grösser die Zahl, desto aufwändiger kann es werden, ihre Primfaktorzerlegung zu bestimmen.

Übung 5.3.

Bestimme die Primfaktorzerlegung von 2, 469, 135 und 782.

Notizen zu Übung 5.3 $2, 7 \cdot 67, 5 \cdot 3^3$ und $2 \cdot 17 \cdot 23$

Übung 5.3 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.

5.4. Das Verfahren

RSA ist ein asymmetrisches Verfahren, es gibt einen Public Key und einen Private Key. Beide Keys werden mit Hilfe von Primzahlen gebildet und in das Verfahren spielt die Modulo-Rechnung hinein. Den Namen hat das Verfahren von Ronald Rivest, Adi Shamir und Leonard Adleman; sie haben es 1977 kreiert.

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 ($e \mid n$). Eine dritte Zahl d (decipher = entschlüsseln), zu deren Berechnung wir später kommen, ist Grundlage des Private Keys. Verschlüsselt wird durch

$$C \equiv m^e \pmod{n}.$$

Die Zahlen n und e sind der Public Key. m ist der Klartext (Message), C der verschlüsselte Text (Cipher).



Abbildung 7: Shamir, Rivest und Adleman

Übung 5.4.

Nehmen wir an, du willst an Alice die Nachricht m schicken. Alices Public Key sei $(7 | 187)$, also $n = 187$ und $e = 7$. Das m lautet in ASCII 01001100, das entspricht der Dezimalzahl 76. Die Nachricht lautet also

Notizen zu Übung 5.4

$$76^7 \bmod 187 \equiv 32 = 100000_{(2)}$$

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

$$m \equiv C^d \bmod n$$

Übung 5.5.

Und umgekehrt? Angenommen, dein öffentlicher Schlüssel ist $(7 | 187)$. 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.

Notizen zu Übung 5.5

$$142^{23} \bmod 187 \equiv 65$$

Übung 5.6.

Probier es mit dem Taschenrechner aus obiger Aufgabe aus. Verschlüssele verschiedene



Zahlen und entschlüssle sie direkt wieder. Versuche Zahlen, die kleiner sind als $n = 187$ und Zahlen, die grösser sind!

Notizen zu Übung 5.6

...

Bemerkung 5.4.1. Das funktioniert allerdings nur, wenn die Nachricht m (als Zahl) kleiner ist als n . Dies 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 5.4.1. Um es einfach zu halten, codieren wir: Leerzeichen = 00, A = 01, B = 02, ..., Z = 26. Mit dem Schlüssel $(e | n) = (17 | 2773)$ 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 5.4.2. 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 5.1: 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 5.3: Modular Inverses

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

Übung 5.7.

Finde das modular Inverse Element c zu 3 modulo 5.

Notizen zu Übung 5.7

Mit Ausprobieren findet man $3^{-1} \equiv 2 \pmod{5}$.

In unserem Fall sind wir 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 das modular 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 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 5.8.

Schreibe allgemein das Verfahren des Euklidschen Algorithmus für zwei Zahlen a und b auf.

Notizen zu Übung 5.8

$$\begin{aligned}
 a &= q_1 b + r_1 \\
 b &= q_2 r_1 + r_2 \\
 r_1 &= q_3 r_2 + r_3 \\
 &\vdots = \vdots \\
 r_{k-2} &= q_k r_{k-1} + r_k \\
 r_{k-1} &= q_{k+1} r_k + 0
 \end{aligned}$$

Rechnet man nun rückwärts und ersetzt die Reste durch entsprechende Ausdrücke, so erhält man eine Vielfachendarstellung von a und b für ihren ggT 1.

Übung 5.9.



Zeige dies.

Notizen zu Übung 5.9



Aus der letzten Zeile folgt $0 = r_{k-1} - q_{k+1} r_k \Leftrightarrow r_{k-1} = q_{k+1} r_k$ und weiter

$$\begin{aligned}
 r_{k-2} &= q_k r_{k-1} + r_k \\
 &= q_k q_{k+1} r_k + r_k \\
 &= (q_k q_{k+1} + 1) r_k \\
 &= l_{k-2} \cdot r_k \\
 &\sim r_k
 \end{aligned} \tag{($l_{k-2} \in \mathbb{Z}$)}$$

Rückwärts eingesetzt folgt $b = l_b r_k$ und $a = l_a r_k$, also $r_k | a$ und $r_k | b$. Also gibt es $x, y \in \mathbb{Z}$ so, dass $r_k = xa + yb$

Dass der Euklid'sche Algorithmus stets den gcd liefert, wurde im Unterricht gezeigt.

Satz 5.2: Lemma von Bézout

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

$$\gcd(a, b) = x \cdot a + y \cdot b.$$

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 5.4.3. Dieses c lässt sich mit dem Euklid'schen Algorithmus berechnen.



Hat man also die Vielfachensummendarstellung

$$1 = \gcd((p-1)(q-1), e) = x \cdot (p-1)(q-1) + y \cdot e$$

bestimmt, so ist die kleinste positive Zahl der Form

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

gleich dem Private Key d .

Bemerkung 5.4.4. 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 5.4: Euler'sche phi-Funktion

Für eine natürliche Zahl n ist $\varphi(n)$ 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 5.3: Euler-phi einer Primzahl

Ist p prim, dann gilt

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

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

Satz 5.4: Satz von Euler

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

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

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

Satz 5.5: Multiplikativität der Euler'schen phi-Funktion

Sind p, q zwei verschiedene Primzahlen, so ist

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

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

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

Satz 5.6: 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:

Beweis. 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:

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

□

5.5. RSA knacken

Zum Schluss noch ein Beispiel, in dem es dem Codeknacker leicht gemacht wird. Also keine Angst, so schnell wie hier lässt sich RSA nicht knacken...

Beispiel 5.5.1. Man fängt eine RSA-verschlüsselte Nachricht ab. Sie lautet

10473054210223505497035330523101828168
2305497093070549713322042531164705144

Ausserdem weiss man, für wen die Nachricht bestimmt ist. Der öffentliche Schlüssel des Empfängers ist $n = 17947$, $e = 21$. Um die Nachricht zu entschlüsseln, muss man wissen, aus welchen beiden Primzahlen p und q sich n zusammensetzt. Eine erste Möglichkeit, p und q zu finden 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 die Zahlen p und q eruiert wurden, kann man den Private Key bestimmen.

5.6. Übung zu RSA mit Mathematica



Zum Schluss noch ein etwas grösseres Beispiel. Alice wähle 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 $\text{GCD}[C, r] = 1$, das heisst c und r sind teilerfremd. Der Mathematica-Befehl PowerMod[C,-1,r] liefert

$$d = 252883827627895253473.$$

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

$$m_1 = 82836532738384326973, m_2 = 78327169787369838482, m_3 = 69736772,$$

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 $\text{PowerMod}[m_1, d, n]$, $\text{PowerMod}[m_2, d, n]$, $\text{PowerMod}[m_3, d, n]$ 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!

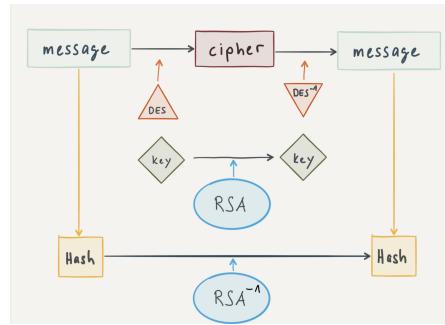


Abbildung 8: PGP Schema

6. 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 8 auf Seite 48 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 Bombe

~~TOP SECRET~~

Op-20-G/ev

TOP SECRET

Bombe History
X Bombe N. 5th 0
Lab Bldg
Dunn
Associate Bombe

24 March 1944.

MEMORANDUM FOR Op-20-G-1:

Subj: Bombe - 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 Bombe have three levels in each physical piece of equipment. Thus the concept of 336 Bombe led to the notion of 112 separate pieces of equipment slightly larger than the British Bombe. 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 Bombe 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 Bombe 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 Bombe required for the Naval problem reached a conclusion in March of 1943 that ninety-six Bombe 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.

A. Memorandum Bombes

~~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 (DUEENNA) 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 DUEENAS or other special devices.

Respectfully,

H. T. Engstrom

H. T. ENGSTRÖM,
Op-20-Gm.

- 2 -

B. Data Encryption Standard

B.1. Binärcodes

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 B.1.

Wie viele verschiedene Zeichen können mit ASCII codiert werden?

Notizen zu Übung B.1

$$2^8 = 256$$

Der Data Encryption Standard (DES) verschlüsselt solche Bit-Folgen. Abbildung 9 auf Seite 52 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 B.2.

Verwandle folgendes Binärwort zurück:

01100110 01100101 01101001 01110011
01110100 01100101 01101100

Notizen zu Übung B.2

feistel

Bei 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

	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
a	46 0 0 1 0 1 1 1 0	p	112 0 1 1 1 0 0 0 0
b	97 0 1 1 0 0 0 0 1	q	113 0 1 1 1 0 0 0 1
c	98 0 1 1 0 0 0 1 0	r	114 0 1 1 1 0 0 1 0
d	99 0 1 1 0 0 0 1 1	s	115 0 1 1 1 0 0 1 1
e	100 0 1 1 0 0 1 0 0	t	116 0 1 1 1 0 1 0 0
f	101 0 1 1 0 0 1 0 1	u	117 0 1 1 1 0 1 0 1
g	102 0 1 1 0 0 1 1 0	v	118 0 1 1 1 0 1 1 0
h	103 0 1 1 0 0 1 1 1	w	119 0 1 1 1 0 1 1 1
i	104 0 1 1 0 1 0 0 0	x	120 0 1 1 1 1 0 0 0
j	105 0 1 1 0 1 0 0 1	y	121 0 1 1 1 1 0 0 1
k	106 0 1 1 0 1 0 1 0	z	122 0 1 1 1 1 0 1 0
l	107 0 1 1 0 1 0 1 1	ü	123 1 0 0 0 0 0 0 1
m	108 0 1 1 0 1 1 0 0	ä	132 1 0 0 0 0 1 0 0
	109 0 1 1 0 1 1 0 1	ö	148 1 0 0 1 0 1 0 0

Abbildung 9: Auszug aus ASCII binär

Zahlen von 0 bis 15 dar:

$$0000 = 0$$

$$0001 = 1$$

$$0010 = 2$$

$$0011 = 3$$

...

$$1101 = 13$$

$$1110 = 14$$

$$1111 = 15$$

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 10 auf Seite 53 zeigt noch einmal den Ausschnitt aus ASCII. Für die hexadezimale Schreibweise benötigt man statt 8 nur zwei Ziffern.

Übung B.3.



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

6C 75 63 69 66 65 72

Notizen zu Übung B.3



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

Abbildung 10: Auszug aus ASCII hexadizimal

lucifer

Lucifer ist der Name eines Verschlüsselungsverfahrens, das HORST FEISTEL und DON COPPERSMITH 1973 im THOMAS J. WATSON Laboratory von IBM in New York entwickelten.

«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 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)

Eine 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	#

B. Data Encryption Standard

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	>>
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

B.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:

$$0 \text{XOR} 0 = 0, 0 \text{XOR} 1 = 1, 1 \text{XOR} 0 = 1, 1 \text{XOR} 1 = 0$$

oder kurz:

Bei zwei Bit-Folgen gleicher Länge wendet man XOR auf jede Position einzeln an. Das Ergebnis ist wieder eine Bit-Folge.

Übung B.4.



XOR	0	1
0	0	1
1	1	0

Tabelle 18: XOR

Bestimme das Ergebnis der XOR-Operation.

$$01000101 \oplus 00110011$$

Notizen zu Übung B.4



01110110

Eine andere wichtige Manipulation von Bit-Folgen ist, die Positionen der einzelnen Bits zu vertauschen. Eine Permutation wird einfach durch Aufzählen der neuen Positionen angegeben: Dies ist die Permutation

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

Beispiel B.2.1. Die 8-Bit-Folge

10011010

wird abgebildet auf

01001101

Übung B.5.



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

Notizen zu Übung B.5



10001100

Der 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 11 auf Seite 57).

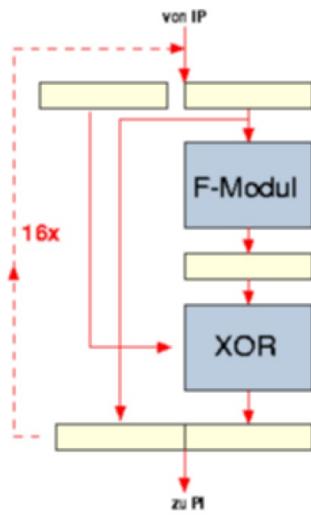


Abbildung 11: Feistel-Round

- 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.

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 2.1: 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 B.2.2. 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 B.6.



Gib die Inverse der Permutation

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

Notizen zu Übung B.6



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

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 b_1 aus b_2 XOR b_1 und b_2 ?

Übung B.7.



Angenommen, man kennt

$$b_2 = 11101001$$

und

$$b_2 \oplus b_1 = 10111010$$

Bestimme b_1 .

Notizen zu Übung B.7



$$b_1 = (b_2 \oplus b_2) \oplus b_1 = b_2 \oplus (b_2 \oplus b_1) = 01010011$$

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

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

Zum Verschlüsseln verwendet DES insgesamt 16 Schlüssel, je einen pro Runde.

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

- $r(b)$ bezeichnet die rechte Hälfte einer 64-Bit-Folge b ,
- $l(b)$ bezeichnet die linke Hälfte,
- b_1b_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-Verschlüsselung

```
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}
```

B.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

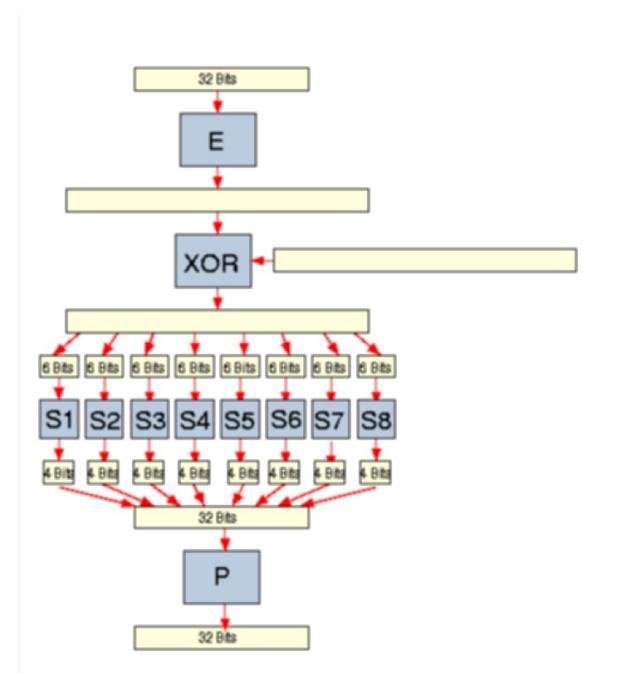


Abbildung 12: F-Modul

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 12 auf Seite 60. Das F-Modul führt folgende Operationen aus:

- Erweiternde Permutation der 32-Bit-Folge auf eine 48-Bit-Folge.
- XOR-Operation mit einem 8-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 B.3.1. Dies ist die erweiternde Permutation

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

Die 4-Bit-Folge

1001

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 13: S-Box

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.

Definition 2.2: 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 13 auf Seite 61 gezeigt.

Die Substitution ist fundamental verschieden von einer Permutation. Sie beruht nämlich nicht darauf, die Bits innerhalb der Bit-Folgen zu vertauschen. 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.

B.4. DES Schlüssel

DES lässt nur Schlüssel mit bestimmter Parität zu.

Definition 2.3: 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 B.8.



Gib die Parität der Bit-Folge 0010011 an.

Notizen zu Übung B.8



Parität 1

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 B.9.



Was erreicht man damit? Sind die folgenden Aussagen richtig oder falsch?

- 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.

Notizen zu Übung B.9



- a) ✓ und d) ✓

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

Definition 2.4: Linksverschiebung

Die Permutation $(n \ 1 \ 2 \ \dots \ n-1)$ heisst Linksverschiebung um 1. Die Permutation $(n-1 \ n \ 1 \ 2 \ \dots \ n-2)$ heisst Linksverschiebung um 2.

Definition 2.5: Vermindernde 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 B.4.1. 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.

B.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.

B.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 B.10.



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.

Notizen zu Übung B.10



nur c) ✗ ist falsch

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.

Eine Chiffre für Bit-Folgen der Länge n heisst linear, falls sich die verschlüsselte Bit-Folge b_2 aus b_1 über eine Vorschrift

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

mit einer festen Matrix A und einem festen Vektor \vec{c} ergibt. 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 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 B.11.



Wie viele verschiedene Schlüssel gibt es im DES?

Notizen zu Übung B.11



$$2^{56} \approx 10^{17}$$

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 B.12.



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

Notizen zu Übung B.12



$$\text{ungefähr } 10^{34}$$

Übung B.13.



Wieso nicht Double-DES?

Notizen zu Übung B.13



Double DES ist auf einen Meet in the Middle Angriff ähnlich anfällig wie DES.

Tatsächlich ist Triple DES eine heute häufig verwendete Chiffre.

Übung B.14.



Betrachtet wird die Bit-Folge

0101 1010 0110 0001 0110 0011 0110 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.

Notizen zu Übung ??



- a) 5A616369
- b) zaci
- c) 0010 1110 0000 0000 1000 1101 1111 1000

Übung B.15.



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.

Notizen zu Übung B.15



Für die vier Schlüssel 0101010101010101, FEFEEFEFEFEFEFE, 1F1F1F1F1F1F1F1F und E0E0E0E0E0E0E0 gilt $E_k(E_k(m)) = m$

Übung B.16.



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?

Notizen zu Übung B.16

Die Schlüssel sollen nicht zu lang werden. Für einen Angriff braucht man immerhin noch $\frac{10^{17}}{24 \cdot 365} \approx 11 \cdot 10^{18}$

Übung B.17.

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

- Gib für die vermindernde Permutation (– 2 4 1 3 –) zwei 6-Bit-Folgen an, die auf dasselbe Ergebnis abgebildet werden.
- Ergibt sich diese Problematik auch bei erweiternden Permutationen?
- 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.

Notizen zu Übung B.17

- 100010 zu 0010 und 100011 zu 0010
- X** nein
- Man könnte bei der Verschlüsselung die Änderung des Ciphertextes beobachten. Man muss nur zwei Klartexte mit fixer Differenz (im Sinne von XOR) wählen und dann entsprechend der Differenz der Chiffrentexte können den Schlüsseln verschiedene Wahrscheinlichkeiten zugeordnet werden. Bei genügend vielen Analysen wird sich ein Schlüssel als der wahrscheinlichste herauskristallisieren.

Die obigen Übungen stammen von der Site MathePrisma DES. Diese führt auch durch ein schönes Lernmodul.

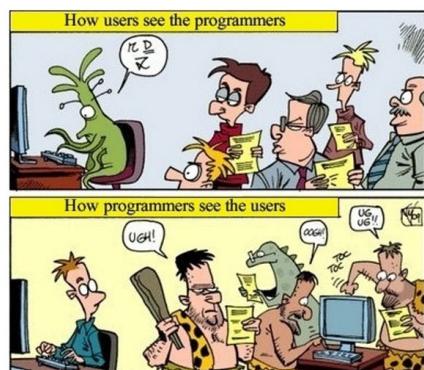


Abbildung 14: Programmers & Users

Abbildungsverzeichnis

1.	Semagramm mit Morsecode (aus Bauer)	6
2.	Caesar-Scheibe zum Ver- und Entschlüsseln	7
3.	Häufigkeitsgebirge Deutsch & Englisch	19
4.	Die ENIGMA	33
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	34
6.	Die Turingbombe	35
7.	Shamir, Rivest und Adleman	39
8.	PGP Schema	48
9.	Auszug aus ASCII binär	52
10.	Auszug aus ASCII hexadizimal	53
11.	Feistel-Round	57
12.	F-Modul	60
13.	S-Box	61
14.	Programmers & Users	68

Tabellenverzeichnis

1.	Caesar-Alphabet mit Translation 3	7
2.	Monoalphabetische Verschlüsselung mit Faktor 3	11
3.	Monoalphabetische Verschlüsselung mit Schlüsselwort und Schlüsselbuchstabe	12
4.	Beispiel einer monoalphabetische Verschlüsselung mit Schlüsselwort JAMES BOND und Schlüsselbuchstabe q	12
5.	Relative Deutsche Buchstabenhäufigkeiten von Mono- und Bigrammen	14
6.	Bigramm-Häufigkeiten deutsch und englisch	18
7.	Trigramm-Häufigkeiten deutsch und englisch	18
8.	Wortlängen von Sprachen	19
9.	Koizidenzindex κ von Sprachen	20
10.	Polyalphabetische Chiffrierung	22
11.	Vigenère-Chiffrierung	22
12.	Vigenère-Verschlüsselung	23
13.	Vigenère-Verschlüsselung mit JAMESBOND	24
14.	Idee Kasiski-Test	24
15.	Kasiski-Test	25
18.	XOR	56