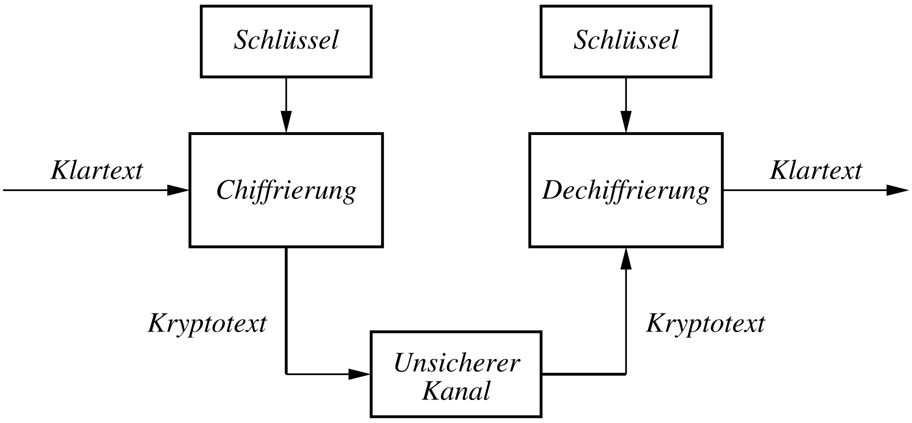
Elemente der Kryptologie

Gerhard Berendt

1 Bezeichnungen

Die *Kryptologie* befaßt sich mit der Verschlüsselung und Entschlüsselung von Nachrichten, die über einen nicht abhörsicheren Kanal versandt werden sollen. Unter dem Namen *Kryptographie* faßt man die Methoden der Verschlüsselung und Entschlüsselung zusammen, während der Begriff *Kryptoanalyse* die Verfahren bezeichnet, die verwandt werden, um einen verschlüsselten Text ohne Kenntnis des benutzten Schlüssels zu entziffern. Ein aus einem Klartext, dem zugehörigen verschlüsselten Text und den Schlüsseln zum Chiffrieren und Dechiffrieren bestehendes System wird auch als *Kryptosystem* bezeichnet. Das Prinzip eines Kryptosystems ist in der Figur 1 dargestellt.
Kryptosysteme wurden bis weit in die Mitte unseres Jahrhunderts fast ausschließlich im militärischen und diplomatischen Bereich benutzt; entsprechend spärlich waren daher die der Öffentlichkeit zugänglichen Informationen über solche Systeme. Erst die dramatische Entwicklung der modernen Kommunikationstechnik führte dazu, daß heutzutage aus Gründen der Datensicherheit die Kryptologie auch im öffentlichen Leben eine immer größere Rolle spielt. Inzwischen werden kryptologische Verfahren in so großer Zahl in den verschiedensten Bereichen der Gesellschaft verwandt, daß wir uns im vorliegenden Rahmen auf die Schilderung der Grundlagen beschränken und darauf verzichten müssen, auch nur annähernd das gesamte Feld kryptologischer Anwendungen abzudecken. Aus dem gleichen Grunde können wir im folgenden auch nicht auf die – sehr reizvolle – Geschichte der Kryptologie eingehen. Inzwischen existiert auch eine reichhaltige Literatur sowohl zu den mathematischen Aspekten dieses Themas als auch zu den mannigfaltigen Anwendungen der Kryptologie auf dem Feld der modernen Datenkommunikation und Datensicherheit. Der interessierte Leser kann daher auf eine große Reihe von Darstellungen des Gebietes auf allen Ebenen zurückgreifen (vgl. z.B.[1],[2],[3]).



Figur 1: Prinzip eines Kryptosystems

Wir wollen nun darangehen, den der Kryptologie zugrundeliegenden Sachverhalt mathematisch zu beschreiben. Die zu verschlüsselnde Nachricht besteht in der Regel aus einer Abfolge von alphanumerischen Zeichen, also Buchstaben eines Alphabetes oder Ziffern; allgemein seien solche Nachrichten Elemente einer Menge K. Die Verschlüsselung eines bestimmten Elementes $k \in K$ geschieht durch eine Abbildung $e : K \to V$ von K auf eine Menge V mit $v = e(k)$, wobei also $v \in V$ das Bild von $k \in K$ sein soll. Diese Abbildungsvorschrift muß umkehrbar eindeutig sein, damit der Empfänger der Nachricht durch Anwenden der Umkehrabbildung $d = e^{-1} : V \to K$ von e das ursprüngliche Element k in der Form $k = d(v)$ zurück erhält; es ist also $d(e(k)) = k$ für alle $k \in K$ (ëßteht hier für “encrypting”, “d” für “decrypting”). Ein Kryptosystem im Sinne der vorangegangenen Definition ist also ein Tripel (K, V, S); hierbei ist S eine Menge von *Schlüsseln*, wobei jedem Schlüssel $s \in S$ eine bijektive Funktion $e_s : K \to V$ (und $d_s := e_s^{-1}$) zugeordnet ist. Die Menge S definiert den sogenannten *Chiffrieralgorithmus*, der dann im konkreten Fall durch den jeweiligen Schlüssel s realisiert wird.

Beispiel (CAESAR-Chiffre):
Sei $K = V = \{A, B, C, \ldots, Z\}$ und $S = \{e\}$ mit

$$\begin{array}{rccccc} k = & A & B & C & \ldots & Z \\ v = e(k) = & D & E & F & \ldots & C \end{array}$$

Die einer erfolgreichen Verschlüsselung als nächstliegend zugrundeliegende Idee besteht nun darin, bei gegebenen Mengen K und V aus der Menge S einen Schlüssel s auszuwählen, welcher nur dem Absender und dem Empfänger der zu versendenden Nachricht bekanntgemacht wird. Auf dieser Basis beruhen alle Verschlüsselungsverfahren der *klassischen Kryptologie*. Daneben ist in den letzten Jahrzehnten ein Zweig der Kryptologie entstanden, der als sogenannte *“Public*

Key Kryptologie" von der Idee ausgeht, den Schlüssel s öffentlich bekanntzumachen. Dabei wird allerdings die Funktion e_s so ausgesucht, daß die Auswertung ihrer Umkehrfunktion $d_s = e_s^{-1}$ mit einem derart hohen Rechenaufwand verbunden ist, daß – selbst bei Einsatz der schnellsten Computer – die Berechnung ihrer Werte nur in Jahrtausenden möglich ist (Funktionen e_s, die diese Eigenschaften haben, werden auch als *FalltürFunktionen* bezeichnet). Während es bei den klassischen Chiffrierverfahren erforderlich ist, daß zwei Gesprächspartner auf geheimem Wege ihre Schlüssel austauschen, entfällt dies bei den Public-Key-Verfahren: Hier können die Verschlüsselungsfunktionen öffentlich bekanntgemacht und lediglich die Entschlüsselungsfunktionen jedes Teilnehmers am Datenverkehr geheimgehalten werden. Wenn ein Teilnehmer X einem Teilnehmer Y eine Nachricht schicken will, beschafft er sich aus dem Katalog die Verschlüsselungsfunktion von Y und chiffriert seinen Klartext mit diesem Schlüssel. Nur Y kann dann den erhaltenen Kryptotext mittelds seiner Entschlüsselungsfunktion dechiffrieren.
Wir wollen im folgenden typische Verfahren aus beiden Gebieten kennenlernen, wobei wir uns im vorliegenden Rahmen wieder auf eine exemplarische Behandlung beschränken müssen.

2 Klassische Methoden der Kryptographie

Die im Beispiel des vorangegangenen Abschnitts vorgestellte CAESAR-Chiffre, die bereits von dem römischen Imperator benutzt worden sein soll, gehört zu den ältesten und einfachsten Verschlüsselungen, zu den sogenannten *Transpositionschiffren.* Der Chiffrieralgorithmus dieser Transpositions- oder *Verschiebe*-Chiffren läßt sich am einfachsten formulieren, wenn man die Buchstaben des Alphabets durch die Zahlen 1 bis 26 ersetzt, also $K = \{1, 2, 3, \dots, 26\}$ setzt und die Menge S als $S = \{0, 1, 2, \dots, 25\}$ mit

$$v = e_s(k) := (k + s) \bmod 26$$

wählt[1]. Entsprechend geschieht dann die Entschlüsselung des Kryptotextes durch

$$k = d_s(v) = (v - s) \bmod 26 \ .$$

Für die CAESAR-Chiffre gilt natürlich $s = 3$ in dieser Bezeichnungsweise.
Es liegt auf der Hand, daß Verschiebechiffren von Unbefugten leicht zu entschlüsseln sind; etwas kompliziertere Verschlüsselungen werden wir im folgenden Abschnitt behandeln.

---

[1] Für natürliche Zahlen a und b bezeichnet bekanntlich $a \bmod b$ den ganzzahligen Rest bei der Division von a durch b.

2.1 Monoalphabetische Chiffrierungen

Verschiebealgorithmen sind ein einfacher Spezialfall von *monoalphabetischen* Chiffrierungen. Diese sind dadurch gekennzeichnet, daß bei der Chiffrierung eines Textes jedes Element aus K zu stets dem gleichen Element aus V verschlüsselt wird (dabei braucht V nicht die gleichen Elemente (also etwa Buchstaben) zu enthalten, die auch in K vorkommen). Während man jedoch nur insgesamt 25 nicht-triviale Verschiebechiffren angeben kann – die zudem noch extrem leicht zu analysieren sind –, lassen sich andere monoalphabetische Chiffren in fast beliebiger Anzahl konstruieren. Das vielleicht allgemeinste Konstruktionsprinzip hierbei benutzt als Schlüssel ein *Schlüsselwort* (beispielsweise HUBERTUS) und einen *Schlüsselbuchstaben* (etwa J). Zunächst werden eventuell mehrfach vorkommende Buchstaben aus dem Schlüsselwort entfernt (aus HUBERTUS wird damit HUBERTS); danach wird die Funktion $e : K \to V = K$ für das Alphabet wie folgt definiert:
Beginnend mit dem Schlüsselbuchstaben werden dieser und die folgenden Buchstaben mit dem Schlüsselwort chiffriert. Die weiteren Buchstaben des Alphabets und danach die ersten Buchstaben bis zum Vorgänger des Schlüsselbuchstabens werden mit den noch übrigen, nicht im Schlüsselwort vorkommenden Buchstaben des Alphabets in ihrer natürlichen Reihenfolge chiffriert. Für das angenommene Beispiel sieht das dann also wie folgt aus:

$$\begin{array}{lccccccccccccc}
k = & A & B & C & D & E & F & G & H & I & \mathbf{J} & K & L & M \\
v = e(k) = & N & O & P & Q & V & W & X & Y & Z & \mathbf{H} & \mathbf{U} & \mathbf{B} & \mathbf{E}
\end{array}$$

$$\begin{array}{lccccccccccccc}
k = & N & O & P & Q & R & S & T & U & V & W & X & Y & Z \\
v = e(k) = & \mathbf{R} & \mathbf{T} & \mathbf{S} & A & C & D & F & G & I & J & K & L & M
\end{array}$$

In gewissem Sinne kann man sagen, daß alle monoalphabetischen Chiffrierungen über das Konstruktionsprinzip von – u.U. allerdings recht merkwürdigen – Schlüsselwörtern herstellbar sind. Man könnte daher leicht zu dem Schluß gelangen, daß bei genügend phantasievoller Wahl von solchen Schlüsselwörtern die entstandenen Kryptotexte bei Unkenntnis des verwendeten Schlüssels hinreichend schwer entzifferbar seien. Dies ist allerdings bei hinreichender Länge des zu chiffrierenden Textes ein Irrtum, wie wir im folgenden sehen werden.

2.2 Kryptoanalyse monoalphabetischer Chiffren

Wie kompliziert auch die Verschlüsselung mit einer monoalphabetischen Chiffrierung angelegt wird, sie besitzt stets die Eigenschaft, daß für die gesamte Textlänge jeder Buchstabe des Alphabets K durch *genau ein* Zeichen aus dem Alphabet V dargestellt wird. Liegt der Klartext nun in einer natürlichen Sprache vor, dann kann man sich dies zunutze machen, um den chiffrierten Text auch ohne Kenntnis des Schlüssels zu entziffern.

Hierzu benutzt man die Tatsache, daß in jeder natürlichen Sprache in einem längeren zusammenhängenden Text die einzelnen Buchstaben des Alphabets nicht alle ungefähr gleich häufig vorkommen, sondern daß vielmehr die Häufigkeitsverteilung der Buchstaben ausgesprochen ungleichmäßig ist, daß aber – und dies ist das Entscheidende – diese Häufigkeitsverteilung weitgehend *unabhängig* von dem Inhalt des Textes ist (sofern es sich nicht um ganz fachspezifische Texte, etwa die Beschreibung eines chemischen Prozesses oder eines formelmäßigen Sachverhalts handelt). In der folgenden Tabelle sind die mittleren Häufigkeiten der Buchstaben in deutschsprachigen Texten notiert; entsprechende Häufigkeitsanalysen liegen natürlich auch für andere Sprachen vor.

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

Relative Häufigkeit der Buchstaben in deutschsprachlichen Texten
(nach Beutelspacher [1])

Man erkennt, daß hier der Buchstabe "**e**" signifikant am häufigsten auftritt (dies trifft übrigens auch für andere lebende Sprachen, etwa das Englische zu). Es liegt daher beispielsweise für einen Kryptoanalytiker nahe zu vermuten, daß das weitaus häufigst vorkommende Zeichen in einem vorliegenden, monoalphabetisch chiffrierten Text das Äquivalent zu einem "**e**" im zugehörigen Klartext ist. Entsprechend sind die nächsthäufig vorkommenden Buchstaben im Deutschen das "**n**", das "**i**" und das "**s**", so daß entsprechend oft vorkommende Zeichen im Kryptotext gute Kandidaten für diese drei Klartextbuchstaben sind.
Die auf der relativen Häufigkeit der Buchstaben beruhende statistische Analyse von Texten kann natürlich noch verfeinert werden. So kommen neben den einzelnen Buchstaben des Alphabets auch Buchstabenkombinationen – insbesondere Doppelbuchstaben, wie "**nn**", "**pp**" oder oder nebeneinanderstehende Buchstaben, wie "**ie**", "**er**" oder "**ei**" – mit textunabhängigen relativen Häufigkeiten vor, die statistisch signifikant sind. Aus der Kenntnis solcher Kombinationen kann ein geübter Kryptoanalytiker weitere Informationen aus einem vorliegenden Kryptotext entnehmen, die die Entschlüsselung des Textes ohne Kenntnis des benutzten Schlüssels im Endeffekt ermöglichen.
Die Kryptoanalyse monoalphabetischer Chiffrierungen mit Hilfe der Häufigkeitsanalyse der Buchstaben in Texten natürlicher Sprachen war bereits im vorigen Jahrhundert weitgehend bekannt; ein schönes Beispiel dafür ist etwa die Entzifferung einer Anweisung zur Schatzsuche in einer Kurzgeschichte von E.A. Poe [4].

Unter diesem Gesichtspunkt ist es erstaunlich, daß das einzige, derzeit noch – hauptsächlich im Bankgeschäft benutzte – klassische Chiffrierverfahren ein monoalphabetischer Algorithmus ist – und es bisher noch niemandem gelungen ist, diesen Algorithmus ohne Kenntnis des benutzten Schlüssels zu "knacken". Der Grund für die Sicherheit des Verfahrens liegt darin, daß als Grundmenge K kein Alphabet einer natürlichen, sondern das wesentlich größere einer künstlichen Sprache benutzt wird und daß die Schlüsselmenge ebenfalls außerordentlich groß ist. Bevor wir uns mit diesem Verfahren, dem 1975 vom National Bureau of Standards der USA veröffentlichten *"Data Encryption Standard"* – kurz *DES* genannt – genauer befassen, wollen wir einige grundlegende Bemerkungen zur Definition der "Sicherheit" eines Kryptosystems machen.

2.3 Die Sicherheit von Kryptosystemen

Intuitiv ist es klar, wann ein Kryptosystem als "sicher" betrachtet werden kann: Ein (im Idealfall beliebig guter) Kryptoanalytiker darf ohne Kenntnis des Schlüssels auch bei Einsatz aller vorhandenen technischen Mittel nicht in der Lage sein, seine Kenntnis über das System zu verbessern, nachdem er seine Kunst daran geübt hat. Diese Idee führt dazu, ein Kryptosystem (K, V, S) als "gut" (oder "sicher") zu bezeichnen, wenn es die folgenden Bedingungen erfüllt:

1. Für jeden Schlüssel $s \in S$ und für jeden Klartext $k \in K$ sind $e_s(k) = v$ und $d_s(v) = e_s^{-1}(v) = k$ leicht zu berechnen.

2. Für jeden Klartext $k \in K$ hat der chiffrierte Text $v = e_s(k)$ keine wesentlich grössere Länge als k selbst (damit die Übertragung der chiffrierten Nachricht nicht erheblich länger dauert als die des Klartextes).

3. Aus einem chiffrierten Text v kann ohne Kenntnis des Schlüssels s (und damit der Funktion d_s) nicht oder nur sehr schwer der zugehörige Klartext k gewonnen werden.

Wie wir gesehen haben, ist die Bedingung 3 beispielsweise bei monoalphabetischen Chiffrierverfahren über Alphabeten natürlicher Sprachen ganz offenbar nicht erfüllt, da bereits mit einfachen statistischen Überlegungen die Entschlüsselung eines Kryptotextes ermöglicht wird. Es liegt daher nahe zu untersuchen, ob sich Chiffrierverfahren, bei denen die Zuordnung der Buchstaben des Klartextes zu denen des Kryptotextes nicht stets die gleiche ist, einer unbefugten Entzifferung besser widersetzen. Solche Verfahren werden *polyalphabetisch* genannt, und diese sollen im nächsten Abschnitt betrachtet werden. Danach werden wir die Frage erörtern, ob es überhaupt ein hinreichend sicheres klassisches Kryptosystem gibt.

2.4 Polyalphabetische Chiffrierungen

Die statistische Analyse eines Chiffretextes vermittels der relativen Häufigkeiten von Einzelbuchstaben und Buchstabenkombinationen versagt, wenn die Zuordnung von Klartextbuchstaben zu Kryptotextbuchstaben entlang eines Textes in rascher Folge variiert wird. Verfahren, die dies leisten, sind vom *polyalphabetischen* Typ. Im einfachsten Fall wird dabei ein Verschiebealgorithmus verwendet, der für jeden Buchstaben eines Schlüsselwortes die Verschlüsselung mit der zu diesem Buchstaben gehörigen Transposition vornimmt. Diese Chiffrierung wird nach dem französichen Diplomaten **Blaise de Vigenère**, der sie bereits im 16. Jahrhundert veröffentlichte, als *Vigenère-Chiffre* bezeichnet (ähnliche Chiffrierungen wurden übrigens im gleichen Jahrhundert auch von anderen Autoren beschrieben).
Zur Vigenère-Chiffre gehören also zum einen eine Übersicht über alle Transpositionsalphabete und zum anderen ein Schlüsselwort. Die möglichen Transpositionen werden am einfachsten durch eine Tabelle dargestellt, aus der man jedes Verschiebealphabet unmittelbar entnehmen kann – beispielsweise in der in Figur 2 angegebenen Form (in der ersten Zeile dieser Tabelle stehen die Buchstaben des zu chiffrierenden Klartextes).

Das Schlüsselwort besteht aus einer beliebigen Buchstabenfolge, es sei beispielsweise als "HERZOG" gewählt. Nun wird der n-te Buchstabe des Klartextes mit derjenigen Verschiebechiffre codiert, die sich in der ständig bis zur Länge des Klartextes wiederholten Buchstabenfolge des Schlüsselwortes, also aus "HERZOGHERZOGHERZOGHERZ..." an der Stelle n ergibt. Lautet also der Klartext etwa

Schau mir tiefer in die Augen,

dann wird gemäß

HERZOGHERZOGHERZOGHERZOG
SCHAUMIRTIEFERINDIEAUGEN

das **S** durch **Z** - nämlich das Zeichen aus Zeile "S" und Spalte "H"-, das **C** durch **G**, das **H** durch **Y** usw., schließlich das **N** durch **T** chiffriert; der gesamte Kryptotext ergibt sich damit in diesem Beispiel zu

ZGYZISPVKHSLVZMROLELFST.

Wie intendiert, wird hier an verschiedenen Stellen des Klartextes derselbe Buchstabe durch *verschiedene* Buchstaben verschlüsselt; so wird beispielsweise das "a" aus dem ersten Wort des Klartextes zu "z" im Kryptotext, während das "a" aus dem letzten Wort zu "e" codiert wird. Nichtdestoweniger ist es für den Empfänger bei Kenntnis des Schlüsselwortes völlig unproblematisch, den Kryptotext unter Anwendung der in Figur 2 angegebenen Tafel wieder zu entziffern.

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

Figur 2: Das Vigenère-Quadrat

2.5 Kryptoanalyse polyalphabetischer Chiffren

Da die einfache Häufigkeitsanalyse des Vorkommens der einzelnen Buchstaben des Alphabets, wie wir gesehen haben, nicht auf einen polyalphabetisch chiffrierten Text anwendbar ist, könnte man vermuten, daß polyalphabetische Chiffren sehr viel sicherer seien als monoalphabetische. Tatsächlich jedoch existieren auch für diesen Typ von Verschlüsselungen schon seit langem systematische Verfahren der Kryptoanalyse. Bereits im Jahre 1863 wurde von dem preussischen Infanteriemajor **F.W. Kasiski** eine Methode zur Entzifferung polyalphabetischer Chiffren bei Unkenntnis des Schlüsselwortes publiziert; ein weiteres viel benutztes Verfahren zur Analyse polyalphabetischer Chiffren mit Schlüsselwort stammt beispielsweise von **W. Friedman**. Wir wollen den Kasiski-Test im folgenden kurz diskutieren; eine ausführliche Beschreibung der genannten beiden Verfahren findet sich in [1],[3] und [8].

Kasiski's Idee besteht darin, die Länge des Schlüsselwortes zu bestimmen. Ist diese bekannt, dann kann auch auf einen mittels der Vigenère-Chiffre verschlüsselten, genügend langen Text die Häufigkeitsanalyse angewandt werden. Denn gleiche Klartextbuchstaben, die in verschiedenen jeweils durch das Schlüsselwort definierten Blöcken an der gleichen Stelle stehen, werden ja in jedem dieser Blöcke zu dem gleichen Buchstaben des Kryptotextes chiffriert (so wird in unserem Beispiel das zweite "**e**" aus dem Wort "**tiefer**" auf die gleiche Weise chiffriert wie das "**e**" aus dem Wort "**die**", nämlich zu dem Buchstaben "**l**"). Anders ausgedrückt: Hat das Schlüsselwort die Länge n und der Klartext die Länge l, dann werden die Buchstaben mit gleichem k mod n für $k = 1, 2, 3, \ldots, l$ monoalphabetisch chiffriert, und diese Teilmenge des Kryptotextes ist daher z.B. der einfachen Häufigkeitsanalyse zugänglich.

Der Kryptoanalytiker, der eine Vigenère-Chiffre ohne Kenntnis des Schlüsselwortes entziffern will, sucht daher im Kryptotext nach *gleichen* Buchstabenfolgen der Länge 3 oder 4 (da die Wahrscheinlichkeit, daß zwei identische Buchstabenfolgen des chiffrierten Textes von solchen des Klartextes herrühren und sich dann an der gleichen Stelle des Schlüsselwortes befinden, mit der Anzahl der aufeinanderfolgenden Buchstaben der Folge wächst) und prüft, welchen Abstand diese Folgen voneinander haben. Der kleinste gemeinsame Teiler dieser Abstände – oder aber ein Vielfaches davon – ist dann ein guter Kandidat für die Länge des Schlüsselwortes. Mit ein wenig Raten und Gefühl läßt sich so durch eine Häufigkeitsanalyse in der Regel relativ leicht prüfen, ob die gemachte Vermutung korrekt ist oder nicht und somit die Entzifferung bewerkstelligen. Wir wollen das Verfahren an einem Beispiel erläutern:

Sei der in der Figur 3 dargestellte Kryptotext gegeben, von dem angenommen werden kann, er sei nach der Vigenère-Methode chiffriert worden (und der lediglich der besseren Übersicht halber in Blöcken zu je 5 Buchstaben aufgeteilt ist).

Als erstes suchen wir nun mehrfach vorkommende Dreierkombinationen von Buchstaben in diesem Text und bestimmen deren Abstand voneinander. Insgesamt gibt es 74 verschiedene solcher Kombinationen; die ersten davon sind

Folge **LGE**, bei Position 0, mit Abstand 103 zum nächsten Vorkommen,
Folge **EMR**, bei Position 2, mit Abstand 126 zum nächsten Vorkommen,
Folge **YAY**, bei Position 13, mit Abstand 44 zum nächsten Vorkommen,
Folge **IWI**, bei Position 16, mit Abstand 120 zum nächsten Vorkommen,
Folge **QWE**, bei Position 22, mit Abstand 360 zum nächsten Vorkommen,
Folge **XJI**, bei Position 25, mit Abstand 384 zum nächsten Vorkommen,
Folge **TBF**, bei Position 30, mit Abstand 444 zum nächsten Vorkommen,
Folge **RFE**, bei Position 40, mit Abstand 30 zum nächsten Vorkommen,
Folge **QSK**, bei Position 46, mit Abstand 186 zum nächsten Vorkommen,
Folge **SKY**, bei Position 47, mit Abstand 186 zum nächsten Vorkommen,
...
...

Bis auf wenige "Ausreißer" sind alle gefundenen Abstände, wie man sofort sieht, Vielfache von 6, so daß als erste Näherung die Vermutung angebracht ist, das Schlüsselwort bestehe aus 6 Buchstaben. Wir ordnen daher den Kryptotext in Zeilen von je 6 Spalten an und suchen in jeder Spalte nach dem Buchstaben mit der größten relativen Häufigkeit (von dem wir vermuten dürfen, daß er das jeweilige Äquivalent des in deutschsprachlichen Texten am häufigsten vorkommenden "**e**" ist). Eine simple Analyse zeigt, daß dies die Buchstaben **V**, **Y**, **F**, **I**, **R** und **W** sind. Suchen wir nun in dem Vigenère-Quadrat diejenigen Zeilen, in denen das Klartext-"**e**" durch die genannten Buchstaben eretzt wird, dann folgt als vermutetes Schlüsselwort unmittelbar das Wort *"RUBENS"*. Mit diesem Wort entschlüsselt, ergibt sich tatsächlich der Klartext (der hier bereits durch additive Satzzeichen und Zwischenräume "lesbar" gemacht wurde):

> **Um die Nachteile zu vermeiden, die sich ergeben, wenn zur Datensicherung lediglich Kennwoerter verwendet werden, wurden Verfahren entwickelt, bei denen die Daten verschluesselt werden. Zu diesen Daten haben dann zwar auch Unbefugte Zugriff, da sie jedoch nicht wissen, wie diese Daten zu entschluesseln sind, sind derart chiffrierte Nachrichten fuer sie wertlos. Unter der Annahme, dass die benutzten Codierungsverfahren schwer oder gar nicht geknackt werden koennen, sind auf diese Weise die Informationen vor unbe-**

```
LGEMR FRWIX RACYA YIWIG FMQWE XJIFA
TBFVT WSYOA RFETV VQSKY OWVUY YSYAY
CYEMT DZWIO RFEQP IELVL WIEOV HEIGO
VLEIA OLLEI ANVLG EUJVH FRGOZ WLIYL
SYJHR FVHEM RVRNF RIWIM DLYMV MTIYL
NYSHR FQOEM RKVHE EGWEB BFRFU UORMO
RLBYP ZLHCI SMXNF DHYIC GJQSJ CFNRV
FWIRV UYNXM FKVHX MRVZY TIQSK YODHW
ENTGU DLYTW RDEMJ RQKZH EHRJR LUGUA
WZSMR JKYOE PZICD LGWEZ VIEKZ YXIEL
CITYA LVLEI ESEHB LZWUU TWQAV VFRHL
QNFRP GUCFV HFXMW IEXRB SIAKT BXIEG
UYSKN JECDL GYVEO EPCKQ FVQWE EPIAF
VHTMA VROGH VWJYX IVKVX JIVFW ISQNL
ZIOIA NFLVR OWWOH XREQO HVVXW AFWPZ
LYUDG VVLOE PZKYJ PRAEY TWBDT BFRIW
IZBLE WEMCI FLVBU RNLLY SPVUY XBVVF
UUTWQ AVGFX UGUYE IENVL TGUDL YTWRD
LHHFR AUYOT NJKHF VATVE BRALJ YJRHF
UXFVF UYFVI FKVFE EUWIC OHRJI YHIYR
LPPVM OZMDL RFJYO HRJLH EIZHW UFRTW
IUVWT WKUVW PZKQF VQWEG VWF
```

Figur 3: Ein Kryptotext

> **fugtem Zugriff geschuetzt. Der Nachteil eines solchen Verfahrens besteht natuerlich darin, dass die Methode der Verschluesselung beiden Partnern bekannt sein und der Schluessel daher in der Regel zuvor zwischen Sender und Empfaenger ausgetauscht werden muss.**

Das Beispiel zeigt, daß die Dechiffrierung eines nach dem Verfahren von Vigenère verschlüsselten Textes in der Regel umso leichter ist, je kürzer das Schlüsselwort (das natürlich auch aus einer Wortfolge bestehen kann) im Vergleich zum gesamten Text ist. Die Idee mit dem Abstand der Buchstabenkombinationen wird nur dann sinnvolle Resultate liefern, wenn das Schlüsselwort genügend oft zur Chiffrierung verwandt wird. Andererseits wird der notwendige Austausch des Schlüsselwortes zwischen Sender und Empfänger umso riskanter sein, je länger dieses Wort ist, so daß davon auszugehen ist, daß in der Regel die Länge des Schlüsselwortes in der Tat klein ist gegenüber der Länge des Textes[2].
Insgesamt kann also festgehalten werden, daß auch eine – praktikable – Chiffrierung auf polyalphabetischer Grundlage keinen durchschlagenden Erfolg im Sinne der Forderung 3. an die sicheren Kryptosysteme gewährleistet. Im mono- wie im polyalphabetischen Fall scheitert die Forderung nach absoluter Sicherheit im Endergebnis daran, daß es nicht sinnvoll ist, die im Schlüssel enthaltene Information in die gleiche Größenordnung wie die im Gesamt-Klartext vorhandene zu setzen[3]. Damit wird in der Regel eine Dechiffrierung von Kryptotexten nach "Trial and Error" und mit Hilfe von mehr oder weniger raffinierten statistischen Überlegungen ermöglicht. Im nächsten Abschnitt werden wir einen Ausweg aus diesem Dilemma skizzieren.

2.6 Der Data Encryption Standard

Wie läßt sich einerseits die Anzahl der Zeichen in einem Alphabet signifikant vergrößern und andererseits der Schlüssel so oft wechseln, daß statistische Methoden zur Dechiffrierung des Kryptotextes versagen? Im Jahre 1975 veröffentlichte das National Bureau of Standards der USA ein von der Firma IBM entwickeltes neues Chiffrierverfahren, den sogenannten *Data Encryption Standard*, kurz *DES* genannt, mit dem binär codierte Nachrichten außerordentlich sicher verschlüsselt werden können (tatsächlich hat bis heute noch niemand den Code ohne

[2] Tatsächlich kann man zeigen, daß ein polyalphabetischer Vigenère-Algorithmus mit einem Schlüsselwort, das die gleiche Länge wie der Klartext besitzt, im Sinne der Definition aus 2.3 als absolut sicher angesehen werden kann (vgl. etwa [1]). Aus dem hier genannten Grunde ist ein solcher Algorithmus jedoch unpraktikabel.

[3] Eine Möglichkeit, den Umfang des *auszutauschenden* Teils der Schlüsselinformation klein zu halten, könnte z.B. darin bestehen, Schlüsselwörter auf bestimmten Seiten eines allgemein zugänglichen Buches zu verwenden, was allerdings die Präsenz eben dieses Buches bei beiden Partnern voraussetzt.

Kenntnis des Schlüssels zu "knacken" vermocht)[4]. Um den DES-Algorithmus zu verstehen, müssen wir etwas weiter ausholen, wobei wir uns allerdings auf die Erläuterung des Prinzips beschränken werden. Eine ausführliche Beschreibung des DES-Verfahrens ist beispielsweise in [3] nachzulesen.
Die Tatsache, daß der DES-Algorithmus für die digitale Datenkommunikation entwickelt wurde, zieht in naheliegender Weise sowohl eine implizite Vergrößerung des Alphabets als auch eine solche der Schlüsselmenge nach sich. Informationen werden in diesem Algorithmus ausschließlich binär (d.h. mit den *Bits* "0" und "1" als Einheiten) verarbeitet. Die zu verschlüsselnden Zeichen besteht aus 256 *Bytes* – Blöcken zu je 8 Bits –, wobei die normalen Buchstaben, Ziffern und Satzzeichen nach dem sogenannten *ASCII-Code*[5] codiert sind.
DES ist eine *Produkt-Block-Chiffre*[6] . Da bei Blockchiffren ein bestimmter Klartext-Block stets in den gleichen Kryptotext-Block überführt wird, handelt es sich um eine monoalphabetische Verschlüsselung. Um einer Häufigkeitsanalyse zu widerstehen, muß

a) das benutzte Alphabet möglichst groß sein und nicht nur eine kleine Teilmenge daraus verwendet werden sowie

b) die Schlüsselmenge genügend groß sein, um eine vollständige Suche durch Austesten aller möglichen Schlüssel zu verhindern.

DES arbeitet mit Ein- und Ausgabeblöcken von jeweils 64 Bit Länge; dh. das DES-Alphabet ist $2^{64} \simeq 10^{19}$ Zeichen groß; die DES-Schlüssel-Menge enthält $2^{56} \simeq 7,2 \cdot 10^{16}$ verschiedene Elemente. Damit sind beide Forderungen in hinreichendem Umfang erfüllt.
Darüber hinaus bedingt die Forderung nach Sicherheit des Verfahrens weitere Eigenschaften des Algorithmus:

1. *Jedes* Bit des Ausgabeblocks muß von *allen* Bits des Eingabeblocks und von *allen* Bits des Schlüssels abhängen.

2. Die Änderung eines einzelnen Bits im Klartext oder im Schlüssel muß eine Änderung um ca. 50% des Chiffretextes nach sich ziehen. Damit wird verhindert, daß ähnliche Klartexte auf ähnliche Chiffretexte führen.

3. Der Algorithmus darf nicht linear sein; damit ein triviales Auflösen der Chiffrierfunktion durch Lösen linearer Gleichungssysteme verhindert wird.

[4] Der DES-Algorithmus kann bereits mittels spezieller Chips hardwaremäßig implementiert werden, so daß es vergleichsweise einfach wird, damit Texte zu chiffrieren und dechiffrieren.

[5] **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange

[6] Chiffre!Produkt-Ch. sind Chiffren, die durch Verknüpfung elementarer Kryptofunktionen, wie beispielsweise Permutationen oder Substitutionen entstehen. In Blocksystemen werden Eingabeblöcke bestimmter Länge *schlüsselgesteuert* in Ausgabeblöcke bestimmter Länge transformiert.

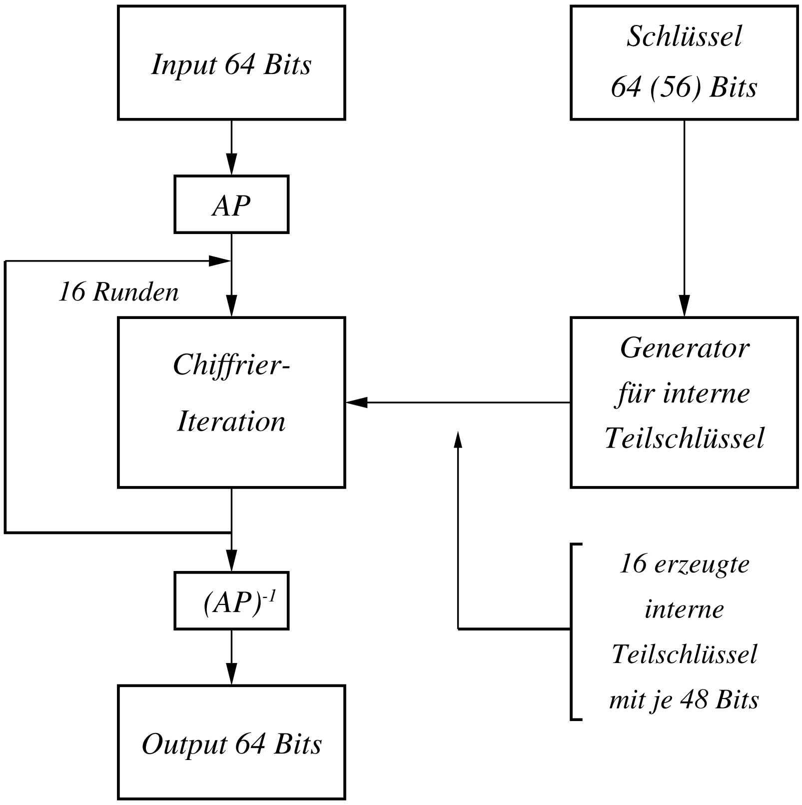
4. Die Einzelchiffren einer Produktchiffre müssen sich in ihren kryptographischen Eigenschaften so ergänzen, daß die statistischen Merkmale des Klartextes im Endergebnis zerstört werden.

5. Verschiedene Schlüssel dürfen einander nicht äquivalent sein. Dies verhindert eine Reduktion der Mächtigkeit der Schlüsselmenge durch Bildung von Äquivalenzklassen (da diese Eigenschaft bei nichtlinearen Chiffren in der Regel nicht beweisbar ist, wird stattdessen oft gefordert, daß aus Schlüsseln viele Teilschlüssel erzeugt werden können, so daß die Anzahl Z aller möglichen Abbildungen $Klartextblock \rightarrow Chiffreblock$ wesentlich größer wird als die Anzahl S möglicher Schlüssel – beim DES ist $Z = 2^{64}$ und $S = 2^{56}$).

6. Nicht zwingend, aber zweckmäßig ist die Forderung, daß die Teilchiffren T einer Produkt-Chiffre involutorisch sind; d.h., daß $T \circ T = id$ bzw. $T^{-1} = T$ gilt. Das reduziert den Aufwand bei der Implementierung des Algorithmus, da dann keine inversen Abbildungen implementiert werden müssen.

Der DES-Algorithmus erfüllt alle diese Bedingungen und kann daher als außerordentlich sicher angesehen werden. Bisher sind keine Erfolge einer Kryptoanalyse eines mittels DES chiffrierten Textes ohne Kenntnis des benutzten Schlüssels bekannt geworden, obgleich der Algorithmus als solcher, wie oben angegeben, bereits vor fast 20 Jahren publiziert wurde.
Wie sieht nun der DES-Algorithmus im Detail aus? In der Figur 4 ist seine Grobstruktur dargestellt. Der (in binärer Form vorliegende) Klartext wird zunächst in Blöcke von jeweils 64 Bits zerlegt, wobei ggf. der letzte Block durch Hinzufügen von Nullen aufgefüllt wird. Ferner wird ein 56 Bits langer Schlüssel (beispielsweise mit einem Zufallsgenerator) erzeugt und mit 8 Prüfbits erweitert, so daß auch der Schlüssel nunmehr 64 Bits lang wird. Jeder Klartext-Block wird dann mit dem gewählten Schlüssel chiffriert und versandt. Der Empfänger kann nun mit dem gleichen Schlüssel den wiederum in 64-Bit-Blocks zerlegten Kryptotext dechiffrieren und erhält unmittelbar den Klartext zurück.
Im einzelnen läuft die Verschlüsselung wie folgt ab:
Aus dem 64-Bit-Schlüssel werden zunächst 16 verschiedene, jeweils 48 Bits lange *Teilschlüssel* hergestellt; ferner wird der Eingabeblock einer Eingangspermutation AP unterworfen. Danach findet – jeweils mit einem der 16 generierten Teilschlüssel – eine Verschlüsselung des Input-Blocks statt, wobei jedesmal das Resultat dieser Verschlüsselung als Input für die nächste Verschlüsselungs-Iteration verwandt wird[7]. Auf das Ergebnis der letzten Verschlüsselung wird schließlich die inverse Eingangspermutation $AP^{-1} = AP$ angewandt und damit der Chiffretext erzeugt.
Bei der Dechiffrierung wird genau so vorgegangen; lediglich die Reihenfolge der

[7]Tatsächlich wird jeder Block vor den einzelnen Iterationen noch in zwei Teile gespalten, unterschiedlich verschlüsselt und nach der betreffenden Iteration wieder zusammengefügt.



Figur 4: Prinzipieller Aufbau des DES-Algorithmus

Teilschlüssel wird umgekehrt, so daß Chiffrierung und Dechiffrierung mit ein und demselben Gesamtschlüssel vorgenommen werden können.

Mit dieser Beschreibung des dem DES-Algorithmus zugrundeliegenden Prinzips müssen wir uns hier begnügen, um nicht den Rahmen der vorliegenden Darstellung zu sprengen. Die Details des Verfahrens können, wie erwähnt, beispielsweise in [3], aber auch an vielen anderen Stellen nachgelesen werden, da der Algorithmus ja weltweit veröffentlicht ist. Das Verfahren wird auch heute noch in bestimmten Branchen des Bankverkehrs, im Scheckkartenbereich, aber auch bei der Einwegverschlüsselung von Passworten in EDV-Rechenanlagen und an vielen anderen Stellen benutzt; es hat allerdings nach Einführung der Public-Key-Kryptosysteme, die im folgenden zweiten Teil dieses Artikels besprochen werden, leicht an Bedeutung verloren.

3 Public-Key-Kryptosysteme

Alle bisher besprochenen Kryptosysteme teilen – unabhängig vom Grad ihrer Sicherheit – die Eigenschaft, daß ihr Schlüssel, also sowohl die Chiffrier- als auch die Dechiffrierfunktion geheimgehalten und zwischen Sender und Empfänger ausgetauscht werden müssen, da ein unbefugter Kryptoanalytiker bei Kenntnis auch der Chiffrierfunktion allein den Kryptotext entschlüsseln kann. Diese Bedingung wird bei den sogenannten *Public-Key-Systemen* bewußt fallen gelassen: Hier werden alle Chiffrierfunktionen veröffentlicht und lediglich die Dechiffrierfunktionen geheim gehalten. Da andererseits die Dechiffrierfunktion $d : V \to K$ die Inverse der Chiffrierfunktion $e : K \to V$ ist, bedeutet dies, daß die Chiffrierung mit solchen Funktionen erfolgen muß, deren – prinzipiell bekannte – Inverse nur mit einem Aufwand *berechnet* werden kann, der so groß ist, daß sich eine Entschlüsselung des Kryptotextes nur in unakzeptabler Zeit bewerkstelligen läßt. Funktionen, die dies leisten, werden daher, wie schon erwähnt, auch oft als *Einweg-* oder *Falltür-* Funktionen bezeichnet. Als tionFalltür-F. bekanntestes Beispiel für eine Chiffrierung mittels einer solchen Einweg-Funktion wollen wir hier kurz das sogenannte *RSA-Verfahren* vorstellen. Daneben existieren noch weitere Public-Key-Chiffrierverfahren, die jedoch nicht eine so allgemeine Verbreitung gefunden haben wie der RSA-Algorithmus (vgl. hierzu etwa [5]).
Der RSA-Algorithmus beruht auf einigen einfachen zahlentheoretischen Grundlagen, die zunächst kurz notiert werden sollen, ehe wir im Detail auf das Verfahren selbst eingehen.

3.1 Einige zahlentheoretische Grundlagen

Zum Verständnis des RSA-Algorithmus sind zwei Sätze aus der elementaren Zahlentheorie erforderlich, der Satz von *Euler* und – als ein Spezialfall davon – der sogenannte kleine Satz von *Fermat* (bezüglich einer ausführlicheren Darstellung des Gebietes und der Beweise der zitierten Sätze vgl. etwa [6],[7],[10]):

Satz 3.1 (EULER) *Seien a und n natürliche teilerfremde Zahlen (d.h., $ggT(a, n) = 1$) und sei $\varphi(n)$ die Anzahl der zu n teilerfremden natürlichen Zahlen kleiner als n. Dann gilt*

$$a^{\varphi(n)} \bmod n \ = 1.$$

Ist $n = p$ eine Primzahl, dann folgt mit $\varphi(p) = p - 1$:

Satz 3.2 (FERMAT) *Ist p Primzahl und $a \in \{0, 1, \ldots, p-1\}$ eine natürliche Zahl, dann ist*

$$a^p \bmod p \ = a.$$

Ist n – wie beim RSA-Algorithmus – das Produkt zweier Primzahlen, also $n = pq$, $\varphi(n) = (p-1)(q-1)$ (Beweis?), dann wird für eine natürliche Zahl $m \in \{0, 1, \ldots, pq-1\}$ mit $ggT(m,p) = 1$ und $ggT(m,q) = 1$ und $k \in I\!N$

$$m^{k(p-1)(q-1)+1} \bmod pq \; = m. \tag{3.1.1}$$

(Diese Aussage gilt übrigens auch, wenn m nicht teilerfremd zu p und q ist.)
Nach diesen Vorbemerkungen können wir nun den RSA-Algorithmus erläutern.

3.2 Das RSA-Verfahren

Das zu besprechende Verfahren, das erstmals im Jahre 1978 von den Autoren R. **R**ivest, A .**S**hamir und L. **A**dleman veröffentlicht und nach den Anfangsbuchstaben der Autorennamen benannt wurde [9], beruht auf der Tatsache, daß es praktisch fast unmöglich ist, zwei große (etwa mehr als hundertstellige) Primzahlen p und q aus der Kenntnis ihres Produktes $n = pq$ zurückzugewinnen. Zur Chiffrierung benötigt man im wesentlichen nur die Zahl n – die also ohne Gefahr allgemein bekanntgegeben werden kann –, während man zur Dechiffrierung die Zahlen p und q selbst kennen muß (die demnach der geheimzuhaltende Teil des Schlüssels sind). Die Vergabe des Schlüssels erfolgt entsprechend der in Figur 5 skizzierten Methode.
In der Schlüsselvergabestelle werden für jeden Teilnehmer am Datenaustausch zunächst aus zwei großen Primzahlen p und q deren Produkt n sowie die Funktion $\varphi(n) = (p-1)(q-1)$ berechnet. Danach werden zwei weitere natürliche Zahlen e mit $ggT(e, \varphi(n)) = 1$ und d gewählt, für die

$$e \cdot d = 1 \bmod \varphi(n)$$

gilt (man zeigt mit Hilfe des EUKLIDischen Divisionsalgorithmus leicht, daß dies immer möglich ist[8]). Schließlich werden dem Teilnehmer die Zahlen e und n als öffentlicher und die Zahl d als geheimer Schlüssel mitgeteilt (der öffentliche Teil des Schlüssels kann natürlich auch direkt – wie die Telefonnummer eines Teilnehmers in einem Telefonbuch – publiziert werden).
Chiffrierung und Dechiffrierung einer Nachricht geschehen nun folgendermaßen:

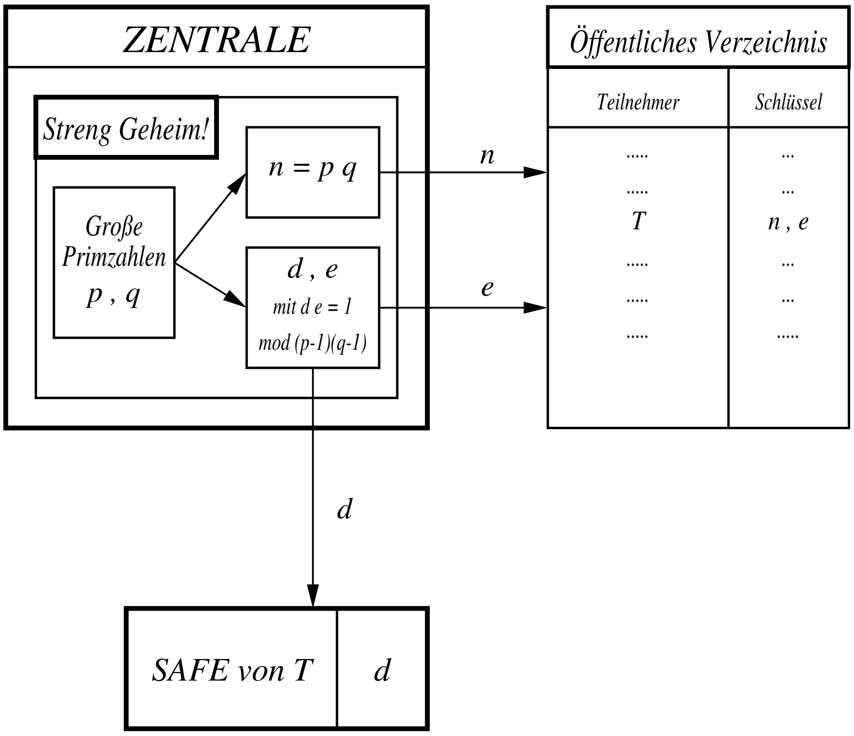
- Als erstes stellt der Sender seine (wie schon beim DES binär codierte) Nachricht in Form einer oder mehrerer natürlicher Zahlen $m \leq n$ dar (wie das gemacht werden kann, zeigt das anschließende Beispiel).

- Anschließend verschlüsselt er m in der Form

  $$c = m^e \bmod n \tag{3.2.2}$$

  und übermittelt c dem Empfänger.

[8] Zweckmäßigerweise wählt man zunächst ein e mit $ggT(e, \varphi(n)) = 1$ und bestimmt danach ein d – das nicht zu klein sein sollte – so, daß die angegebene Bedingung erfüllt wird.



Figur 5: Chiffrierung unter RSA

- Der Empfänger unterwirft die empfangene chiffrierte Zahl c der Operation

  (3.2.3) $$m' = c^d \bmod n$$

  mit dem Ergebnis m', und erhält damit wegen $m' = m$ die ursprüngliche Klartextzahl zurück.

Der Beweis für die Aussage $m' = m$ folgt aus Gleichung (3.1.1):
Mit
$$m' = c^d \bmod n = m^{ed} \bmod n = m \bmod n$$
wird, da $m \leq n$ vorausgesetzt war, $m' = m$, wie behauptet.
Die Anwendbarkeit des Verfahrens beruht also ganz entscheidend darauf, daß es einem unbefugten Kryptoanalytiker nicht gelingt, aus der ihm bekannten – weil öffentlich gemachten – Zahl n ihre Primfaktoren p und q zu ermitteln. Denn gelänge ihm dies, dann könnte er zunächst sofort die Zahl $\varphi(n)$ und dann mit Hilfe der ihm ebenfalls bekannten Zahl e den geheimen Schlüssel d bestimmen. Tatsächlich kann man mit den derzeit verfügbaren Computern selbst unter Zuhilfenahme raffinierter Berechnungsverfahren in vertretbarer Rechenzeit höchstens

Zahlen mit wenigen hundert Stellen faktorisieren. Für den RSA-Formalismus sind daher noch größere Primzahlen ein kostbarer Rohstoff (der natürlich – falls eine solche Zahl wieder einmal gefunden wird – streng gehütet wird).

Um das Verfahren zu demonstrieren, wollen wir jedoch im folgenden ein Beispiel betrachten, das noch mit einfachen Rechenhilfen überblickt werden kann (und demzufolge natürlich auch keineswegs genügend sicher für eine echte praktische Anwendung ist).

Sei also etwa der Klartext

GEHEIM!

vorgelegt. Verwendet man zur binären Codierung den ASCII-Code, dann geht dieser Klartext in

$$01000111010001010100100001000101010010010100110100100001$$

über.
Nun wählen wir z.B. $p = 7$ und $q = 13$, und erhalten also $n = 7 \cdot 13 = 91$ und $\varphi(91) = 72$. Ferner sei etwa $e = 5$, woraus $d = 29$ als der zweite Schlüssel folgt $(5 \cdot 29 = 145 = 1 \bmod 72)$[9].
Damit die Zahlen m, in die der Klartext zerlegt wird, allesamt kleiner als $n = 91$ bleiben, wählen wir eine Zerlegung in 8-Bit-Blöcke (was in diesem einfachen Fall darauf hinausläuft, als Zahlen m einfach die ASCII-Codes der Klartextbuchstaben zu benutzen, da keiner der im Text vorkommenden Buchstaben einen ASCII-Code größer als 90 besitzt[10]). Das bedeutet, daß wir nacheinander die Zahlen

$$71,\ 69,\ 72,\ 69,\ 73,\ 77,\ 33$$

chiffrieren müssen. Dies führt entsprechend Gl.(3.2.2) zu der Kryptofolge

$$15,\ 62,\ 11,\ 62,\ 47,\ ,77,\ ,24\ ,$$

die über Gl.(3.2.3) – wovon Sie sich überzeugen sollten (s.u.) –, wieder in die ursprüngliche Klartextfolge übersetzt wird. Da hierbei jedoch bereits sehr große Zahlen als Ergebnis der Potenzierung entstehen, verwendet man zweckmäßigerweise die leicht zu beweisende Relation

$$(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n\ . \tag{3.2.4}$$

Damit ergibt sich beispielsweise für die erste Dechiffrierung

$$15^{29} \bmod 91 \ = \ ((15^{15} \bmod 91) \cdot (15^{14} \bmod 91)) \bmod 91\ .$$

[9] d ist die *einzige* Inverse von e mod (72), die kleiner als 72 ist.

[10] Bei größerem n könnte man auch m entsprechend größer wählen und dann längere Ziffernfolgen des Klartextes zu der Zahl m zusammenfassen, was zur Folge hätte, daß weniger Zahlen zu chiffrieren und zu dechiffrieren wären.

Weiter folgt

$$15^{15} \bmod 91 \quad = \quad ((15^8 \bmod 91) \cdot 15^7 \bmod 91)) \bmod 91$$

und

$$15^{14} \bmod 91 \quad = \quad ((15^7 \bmod 91) \cdot 15^7 \bmod 91)) \bmod 91$$

Da $15^8 \bmod 91 = 22$ und $15^7 \bmod 91 = 50$ ist wird schließlich

$$15^{29} \bmod 91 = 71$$

und somit bekommen wir die erste Klartextzahl zurück. Entsprechend erhalten wir die übrigen Zahlen der Klartextfolge über die jeweils Modulo 91 reduzierten Potenzen

$$62^{29}, \ 11^{29}, \ 62^{29}, \ 47^{29}, \ 77^{29} \text{ und } 24^{29} \ .$$

Als letzter Schritt werden diese Zahlen vom Empfänger mit dem ASCII-Code decodiert und liefern das Ergebnis

GEHEIM!

4 Zusammenfassung

Wir haben in dieser sehr knappen Darstellung nur einige der wichtigsten klassischen und modernern kryptologischen Verfahren angesprochen. Der enge Rahmen des vorliegenden Artikels verbietet es, wie bereits eingangs erwähnt, darüber hinaus auf die zahlreichen weiteren Verfahren zur Chiffrierung und Dechiffrierung von Texten sowie auf die vielfältigen Anwendungen solcher Verfahren in der heutigen Telekommunikation einzugehen, deren Anzahl und Anwendungsbereich in den letzten Jahren rapide gewachsen sind. Der interessierte Leser sei auf die angeschlossene Bibliographie verwiesen, die ihm hier von Nutzen sein kann.

Literatur

[1] A. Beutelspacher, *Kryptologie*, Vieweg, 1993

[2] R.H. Schulz, *Codierungstheorie*, Vieweg, 1992

[3] P. Horster, *Kryptologie*, BI-Wissenschaftsverlag, 1985

[4] Edgar A. Poe, *Der Goldkäfer*, Winkler-Verlag, München, 1959

[5] A. Salomaa, *Public Key Cryptography*, Springer- Verlag, 1990

[6] F. Padberg, *Elementare Zahlentheorie*, BI-Wissenschaftsverlag, 1989

[7] G. Frey, *Elementare Zahlentheorie*, Vieweg, 1984

[8] O.I. Franksen *Mr. Babbage's Secret. The Tale of a Cypher – and APL*, Prentice Hall, Englewood Cliffs, 1984

[9] R. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public key cryptosystems*, Comm.ACM**21** (1978), 120-126

[10] C.Niederdrenk-Felgner, *Computer im Mathematikunterricht*, Deutsches Institut für Fernstudien, Universität Tübingen, 1988