Primzahlen

und ihre Anwendung in der Kryptographie

Facharbeit aus dem Fach Mathematik

Karlsgymnasium München

Kollegstufenjahrgang 1998 / 2000

Thema:	Primzahlen und ihre Anwendung in der Kryptographie
Verfasser:	Nico Kaiser
Kursleiterin:	Frau OStRin Elfriede Herberg
Abgabetermin:	1. Februar 2000

Erzielte Note: _____ in Worten: _____

Erzielte Punkte: _____ in Worten: _____

_____

Unterschrift der Kursleiterin

Inhalt

1. Einleitung

In einer Welt der elektronischen Post, des Online-Shoppings und des bargeldlosen Zahlungsverkehrs, in einer Welt, in der „Daten und Informationen das wirtschaftlich wichtigste Gut“[1] sind, bekommt die Datensicherheit eine immer größere Rolle. Elektronische Post darf nicht abgehört oder verfälscht werden können, beim Online-Shopping übertragene Konten- oder Kreditkarten-Informationen dürfen nicht frei zugänglich übertragen werden und nur der Besitzer der EC-Karte und der zugehörigen Geheimzahl darf über sein Konto verfügen. Gerade Computernetzwerke, allen voran das weltweite *Internet*, bieten jedoch durch eine ungeschützte Datenübertragung eine Fülle von Möglichkeiten für Spionage und Manipulation.

Schon im alten Rom war das Problem bekannt: Wenn *Julius Caesar* Nachrichten an seine Generäle versandte, traute er seinen Boten nicht und vertauschte die Buchstaben des Alphabets in der Nachricht, so dass nur die Generäle die Botschaft verstanden.

Aufgrund bestimmter Gegebenheiten sind besonders *Primzahlen* als Ausgangspunkt für viele moderne Verschlüsselungs-Verfahren sehr gut geeignet.

In der vorliegenden Facharbeit im Fach Mathematik werde ich mich zunächst mit der Ermittlung von Primzahlen und mit ihren Eigenschaften auseinandersetzen und dann auf deren Verwendung in der Kryptographie eingehen. Hierbei werde ich den RSA-Algorithmus genauer betrachten, da dieser der „Urvater“ aller asymmetrischen Verschlüsselungstechniken ist.

[1] aus: PGP – Pretty Good Privacy, Deutsches Handbuch

2. Primzahlen

2.1 Definition

„*Primzahlen* sind natürliche Zahlen, die nur durch 1 und durch sich selbst teilbar sind; alle Primzahlen außer 2 sind also *ungerade* Zahlen. Es gibt unendlich viele Primzahlen; das Gesetz ihrer Aufeinanderfolge ist jedoch noch nicht bekannt. Sieht man von der Reihenfolge der Faktoren ab, so kann jede natürliche Zahl als Produkt von Primzahlen, den *Primfaktoren*, dargestellt werden“[2]

Eine Primzahl hat also nur *unechte Teiler*, d.h. keine Zahlen die die Zahl „teilen“, sondern nur 1 und sich selbst.

Laut Definition ist die 1 selbst keine Primzahl, die kleinste Primzahl ist somit die 2. 2 ist auch die einzige gerade Primzahl, schließlich enthält jede größere gerade Zahl den Faktor 2.

Die ersten 100 Primzahlen:

2	3	5	7	11	13	17	19	23	29	31	37
41	43	47	53	59	61	67	71	73	79	83	89
97	101	103	107	109	113	127	131	137	139	149	151
157	163	167	173	179	181	191	193	197	199	211	223
227	229	233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349	353	359
367	373	379	383	389	397	401	409	419	421	431	433
439	443	449	457	461	463	467	479	487	491	499	503
509	521	523	541								

[2] aus: Natur – kleine Enzyklopädie

2.2 Die Ermittlung von Primzahlen

In diesem Kapitel werden die wichtigsten und bekanntesten Verfahren zur Ermittlung der Primeigenschaft einer natürlichen Zahl vorgestellt.

2.2.1 Das Sieb des Eratosthenes

Eratosthenes (276–194 v. Chr.), Bibliothekar der großen Bibliothek von Alexandria, hat im Jahr 240 v. Chr. ein Verfahren angegeben, mit dem man alle Primzahlen bis zu einer bestimmten Zahl berechnet.

Funktionsweise:

- Man notiert alle natürlichen Zahlen, bis zu der Zahl, bis zu der man die Primzahlen berechnen will.

  1 2 3 4 5 6 7 8 9 10 11 12 13

- Die 1 ist per Definition keine Primzahl und wird deshalb gestrichen.

  ~~1~~ 2 3 4 5 6 7 8 9 10 11 12 13

- Nun nimmt man die erste ungestrichene Zahl, also die 2 und streicht alle Vielfachen der 2, aber nicht die 2 selbst.

  ~~1~~ **2** 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13

- So verfährt man nun mit der nächsten ungestrichenen Zahl, also der 3.

  ~~1~~ **2** **3** ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13

- Die größte Zahl, deren Vielfache man streichen muss, ist maximal so groß wie die Wurzel der Zahl, bis zu der man die Reihe aufgestellt hat. In diesem Fall ist man nun fertig, da 5 schon größer ist als die Wurzel von 13. Alle ungestrichenen Zahlen sind Primzahlen.

  ~~1~~ **2** **3** ~~4~~ **5** ~~6~~ **7** ~~8~~ ~~9~~ ~~10~~ **11** ~~12~~ **13**

Da hier nach und nach alle Nicht-Primzahlen wie bei einem Sieb herausfallen, wird es das „Sieb des Eratosthenes“ genannt. Dies ist das in der Praxis am meisten verwendete Verfahren.

2.2.2 Mersennsche Primzahlen

Definition: Wenn eine Zahl der Form $2^n - 1$ Primzahl ist, wird sie Mersennsche Primzahl genannt.

Früher war man der Meinung, dass *alle* Zahlen dieser Form Primzahlen sein müssen, aber im Jahre 1536 zeigte *Hudalricus Regius*, dass $2^{11} - 1 = 2047$ keine Primzahl ist, sondern aus den beiden Faktoren 23 und 89 besteht. *Marin Mersenne* (1588–1648) bemerkte im Vorwort zu seiner 1644 veröffentlichten *Cogitata Physica-Mathematica* dass für $n \in \{2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257\}$ die Zahlen $2^n - 1$ Primzahlen sind. Mit den damals zur Verfügung stehenden Mitteln konnten nicht alle Zahlen überprüft werden, erst 100 Jahre später, im Jahr 1750, bewies *Euler*, dass die nächste Zahl auf Mersenne und Regius' Liste, $2^{31} - 1$, Primzahl ist. Ein weiteres Jahrhundert später, 1876, bewies *Francois Edouard Anatole Lucas* (1842–1891), dass $2^{127} - 1$ auch Primzahl ist. Sieben Jahre später zeigte *Pervouchine*, dass $2^{61} - 1$ Primzahl ist, Mersenne diese also nicht berücksichtigt hatte. Anfang 1900 zeigte *Powers*, dass Mersenne auch die Primzahlen $2^{89} - 1$ und $2^{107} - 1$ außer Acht gelassen hatte. 1947 schließlich waren sämtliche Mersennschen Zahlen mit $n \leq 258$ getestet und folgende – korrekte – Liste wurde aufgestellt:

$$n \in \{2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127\}$$

Inzwischen sind 38 solcher Mersennschen Zahlen bekannt, eine Auflistung der Zahlen und deren Entdecker ist im Anhang A zu finden.

2.2.3 Der kleine Satz von Fermat

Der französische Mathematiker *Pierre de Fermat* (1601–1665) stellte einen entscheidenden Satz auf. Er ist ein Spezialfall des Satzes von Euler, auf den hier nicht weiter eingegangen werden soll. Der kleine Satz von Fermat sagt aus:

Ist p eine Primzahl und a eine beliebige ganze Zahl, dann ist

$$a^p \equiv a \pmod{p}$$

Wenn in p kein Teiler von a ist, ergibt

$$a^{p-1} \equiv 1 \pmod{p}.$$

Wenn die Bedingung also nicht zutrifft ist die Zahl p keine Primzahl.

Dieser Satz konnte noch nicht in seiner Umkehrung bewiesen werden, deshalb ist es nicht möglich vom Zutreffen der oben genannten Bedingungen auf eine Primzahl zu schließen.

➔ Der kleine Satz von Fermat ist eine *notwendige* Bedingung für die Primeigenschaft, jedoch nicht *hinreichend*.

2.2.4 Der Satz von Wilson

Eine sowohl notwendige als auch hinreichende Bedingung dafür, dass eine Zahl eine Primzahl ist, ist der *Satz von Wilson*. Schon *Gottfried Wilhelm Leibnitz* (1646–1716) hat in seinen Schriften auf diesen Satz hingewiesen, und etwa 100 Jahre später hat ihn auch der englische Richter und Mathematiker *Sir John Wilson* nachentdeckt. Genau wie Leibnitz konnte auch er ihn nicht begründen, der vollständige Beweis erfolgte im Jahre 1771 durch den französischen Mathematiker *Joseph Louis Lagrange*.

Der Satz von Wilson wendet die Kongruenzrechnung an und kommt zu einem erstaunlichen Ergebnis:

Die natürliche Zahl p teilt $(p-1)! + 1$ genau dann, wenn p Primzahl ist.

Eine andere Schreibweise:

p ist genau dann eine Primzahl, wenn gilt: $(p-1)! \equiv -1 \pmod{p}$

Beispiele:

- $p = 5$: $(p-1)! + 1 = 4! + 1 = 25$; 25 ist durch 5 teilbar → Primzahl
- $p = 6$: $(p-1)! + 1 = 5! + 1 = 121$; 121 ist nicht durch 6 teilbar → keine Primzahl

Im Gegensatz zum kleinen Satz von Fermat ist der Satz von Wilson ein *echter Primzahltest*, d.h. wenn diese Kongruenz für eine Zahl erfüllt ist, dann ist sie eine Primzahl.

In der Praxis ist dieser Satz allerdings kaum zu gebrauchen, da die Fakultätsfunktion so stark ansteigt, dass man auch mit den heute zur Verfügung stehenden Mitteln ihn nur für kleinere Zahlen anwenden kann.

2.3 Die Verteilung der Primzahlen unter den natürlichen Zahlen

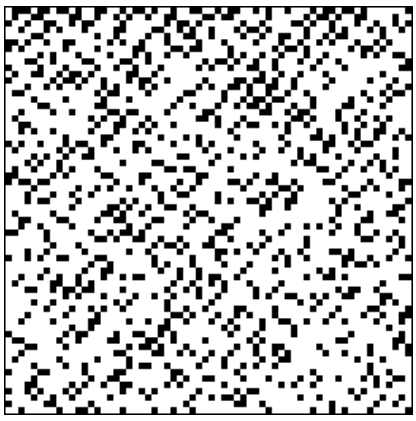
Ein Problem, das viele Mathematiker lange beschäftigt hat, ist die Verteilung der Primzahlen unter den natürlichen Zahlen. Bis heute wurde kein System gefunden, das es ermöglicht, alle Primzahlen nacheinander zu finden, das heißt das System der Verteilung der Primzahlen ist ebenfalls unbekannt. Es können lediglich Aussagen über die Anzahl der Primzahlen in einem bestimmten Zahlenbereich getroffen werden:

Die Mathematiker *Carl Friedrich Gauss* (damals erst 19jährig) und *Legendre* stellten jeweils Funktionen auf, mit denen sich die Größenordnung der Anzahl der Primzahlen unter den ersten *n* natürlichen Zahlen bestimmen lassen. Die Funktion, die die Anzahl der Primzahlen unter den ersten *n* Zahlen angibt nennt man $\pi(x)$:

$$\text{Legendre: } \pi(x) = \frac{x}{\log(x) - 1{,}0836} \qquad \text{Gauss: } \pi(x) = \int \frac{dx}{\log(x)}$$

Erst 1896 konnten *Hadamard* und *de la Valleé-Poussin* beweisen, dass Gauss mit seiner Gleichung recht hatte.

Prof. Dr. Otto Forster vom mathematischen Institut der Ludwig-Maximilian-Universität München beschäftigt sich mit der grafischen Darstellung der Primzahlverteilung:



Dieses Bild zeigt die grafische Darstellung der ungeraden Primzahlen kleiner als 8192. Es ist in 64 Zeilen und 64 Spalten mit insgesamt 4096 schwarzen oder weißen Quadraten, die von 0 bis 4095 durchnummeriert sind, eingeteilt.

Das *n*-te Quadrat ist schwarz, wenn die Zahl $2n + 1$ eine Primzahl ist.

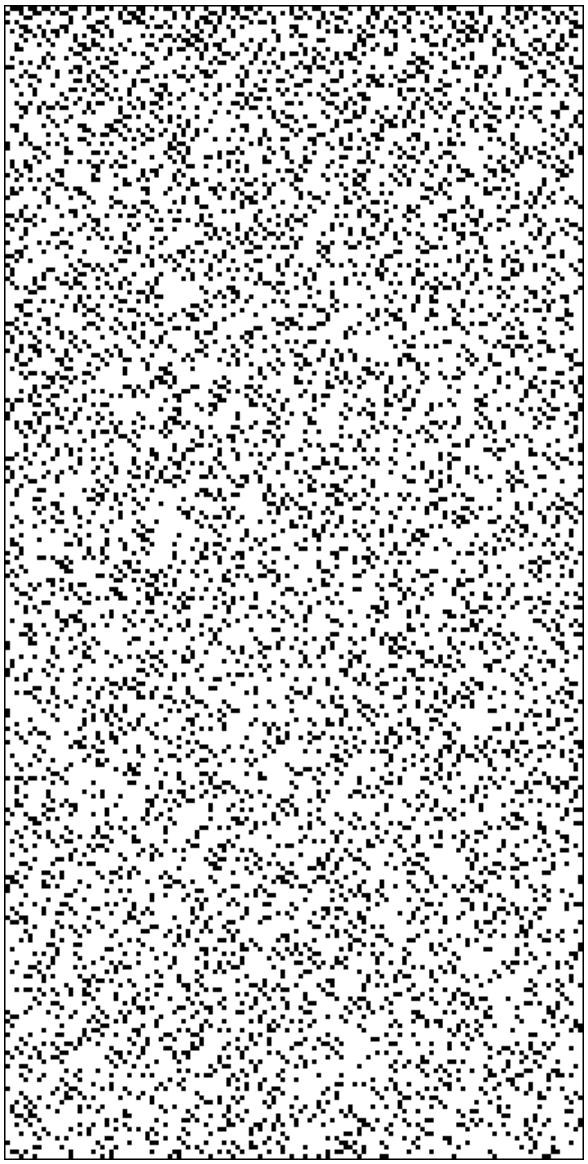
So steht beispielsweise das schwarze Quadrat in der rechten oberen Ecke für die Primzahl 127, das schwarze Quadrat in der rechten unteren Ecke für die Primzahl 8191.

Da jede zweite natürliche Zahl eine Primzahl ist, ist der kleinste Abstand zwischen zwei Primzahlen 2. Auf dem Bild sind diese sogenannten *Primzahlzwillinge* als zwei direkt nebeneinanderliegende Punkte zu erkennen.

Folgen, die auf dem Bild als ■■_■ bzw. ■_■■ erscheinen, heißen *Primzahldrillinge*, die noch seltenere Folge ■■_■■ wird *Primzahlvierling* genannt.

Obwohl die Anzahl dieser Folgen nach oben hin abnimmt, gibt es unendlich viele dieser Folgen.

Das folgende Bild besteht aus 32768 Quadraten (256 Zeilen und 128 Spalten) und stellt die Verteilung der Primzahlen, die kleiner als $2^{16} = 65536$ sind, dar:



2.4 Gibt es unendlich viele Primzahlen?

Die Anzahl der entdeckten Primzahlen nimmt zwar ständig zu, die Zunahme verlangsamt sich jedoch. Findet man im Intervall zwischen 0 und 100 noch 25 Primzahlen, so sind es in einem Intervall gleicher Länge bei größeren Zahlen deutlich weniger. Man findet zwischen 1 000 000 und 1 000 100 nur noch 6 Primzahlen, zwischen 10^{12} und $10^{12} + 100$ noch 4, zwischen 10^{24} und $10^{24} + 100$ noch 2 und im Intervall zwischen 10^{48} und $10^{48} + 100$ schließlich keine Primzahl mehr. Es stellt sich also die Frage, ob nicht alle Zahlen durch eine endliche Anzahl von Primzahlen dargestellt werden können.

Euklid (365–300 v. Chr.) hat die Frage, ob es unendlich viele Primzahlen gibt, durch einen indirekten Beweis, also einen Beweis durch Widerspruch, beantwortet:

1. Euklid nimmt an, dass es nur endlich viele Primzahlen gibt.
2. Wenn es nur endlich viele gibt, dann kann man diese durchnumerieren und nach Größe sortieren. Die Menge der Primzahlen ist dann $P = \{2, 3, 5, 7, 11, 13, ..., p_n\}$
3. Euklid bildet das Produkt aller dieser Primzahlen und addiert 1 hinzu: $q = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot ... \cdot p_n + 1$
4. q besitzt nun Teiler aus P, da P alle Primzahlen enthält.
5. Durch welche Primzahl aus P man q auch teilt, es bleibt immer der Rest 1.
6. Keine Primzahl aus P kann ein Teiler von q sein, q hat also einen anderen Primteiler oder ist selbst eine Primzahl.
7. (6) ist ein Widerspruch zur Annahme, dass in P schon alle Primzahlen enthalten sind. Die Annahme, dass es nur endlich viele Primzahlen gibt, ist somit falsch.

➔ Es muss unendlich viele Primzahlen geben.

2.5 Die größten bekannten Primzahlen

Nachdem nun bekannt ist, dass es unendlich viele Primzahlen gibt, erübrigt es sich zu sagen, dass mit der Zeit ein regelrechter Wettbewerb entstand, neue Primzahlen zu finden.

Im Jahr 1903 wurde die Primeigenschaft der Zahl 2 305 843 009 213 693 951 nachgewiesen, und nach längerer Beschäftigung brachte 1914 der amerikanische Mathematiker *Derrick Norman Lehmer* eine vollständige Liste aller Primzahlen bis 10 006 721 heraus. 1960 war $2^{3217} - 1$ die größte ermittelte Primzahl; sie umfasst 969 Ziffern. Nachdem 1963 an der University of Illinois die 23-ste Mersennsche Primzahl, $2^{11213} - 1$, gefunden wurde, war man dort so stolz, dass ein eigener Poststempel gefertigt wurde:

Ende 1995 stellte *George Woltman* eine Datenbank über alle Primzahlen auf und veröffentlichte sie zusammen mit einem kostenlosen Programm zur Ermittlung von Primzahlen im Internet. Dies war der Anfang des Projekts „The Great Internet Mersenne Prime Search" (GIMPS), dessen Ziel es ist, immer größere Primzahlen zu finden und die Zahlenräume zwischen den schon bekannten Primzahlen nach evtl. nicht berücksichtigten Primzahlen zu durchsuchen. 1997 wurde das *PrimeNet* gegründet, das diese Vorgänge automatisiert und überwacht. Damit ist es nun jedem Computer-Besitzer möglich, am Projekt teilzunehmen und den häuslichen PC mit der Berechnung von Primzahlen zu beschäftigen, wenn er gerade nicht gebraucht wird. Zur Zeit sind etwa 4200 Computer-Benutzer in das Projekt eingebunden. Die seit 1996 vom GIMPS erbrachte Rechenleistung würde auf einem einzigen Computer mehrere Jahrhunderte in Anspruch nehmen.

Am 1. Juni 1999 wurde im Rahmen des GIMPS von *Nayan Hajratwala* die momentan größte bekannte (Mersennsche) Primzahl, $2^{6972593} - 1$, gefunden. Sie hat über 2 Millionen Stellen. Für den Finder einer Primzahl, die mehr als 10 Millionen Stellen hat ist sogar ein Geldpreis von $ 100 000 ausgesetzt.

Auch hier sei wieder auf den Anhang A verwiesen, in dem alle bisher bekannten Mersennschen Primzahlen aufgelistet sind.

2.6 Das Faktorisieren von sehr großen Zahlen

Jede Zahl, die keine Primzahl ist, ist Produkt aus mehreren Primzahlen und lässt sich somit in ihre Bestandteile, die *Primfaktoren* zerlegen. Dies ist beispielsweise bei der Suche eines gemeinsamen Nenners im Mathematikunterricht der Mittelstufe notwendig.

In der Mittelstufenmathematik kommt man durch einfaches Ausprobieren relativ schnell zu einem Ergebnis, da die verwendeten Primzahlen klein sind.

Bei größeren Zahlen steigt der Aufwand, der zur Primfaktorbestimmung notwendig ist, stark an: versucht man, die zu überprüfende Zahl *n* durch alle Primzahlen, die kleiner als die Wurzel von *n* sind, zu teilen, gerät man schnell an die Grenzen der Technik. Schon bei einer 20-stelligen Zahl *n* wären etwa 300 Millionen Divisionen nötig, bei einer 40-stelligen Zahl *n* schon $2{,}2 \cdot 10^{21}$. Mit der 1978 zur Verfügung stehenden Technik hätte man für die Zerlegung eines 200-stelligen *n* eine Zeit benötigt, die dem Alter des Weltalls entspräche. Zwar steigt die Leistungsfähigkeit der Computer ständig an, doch für ein 500-stelliges *n* wäre eine 40-stellige Zahl von Rechenoperationen nötig, damit sind selbst die heutigen Hochleistungsrechner überfordert.

Die Geschwindigkeit einer Faktorisierung hängt jedoch nicht nur von der Größe der zu faktorisierenden Zahl ab, sondern auch von Größe und Anzahl der Primfaktoren. Z.B. wird die Zahl 5^{1000} sehr viel schneller faktorisiert als eine Zahl, die aus zwei großen Primzahlen besteht, deren Differenz ebenfalls nicht zu klein ist. Grund hierfür ist, dass 5^{1000} nur die Primzahl 5 beinhaltet. Ein System, das eine Division durch sämtliche Primzahlen ausprobiert, hat nach erfolglosen Divisionen durch 2 und durch 3 schon beim dritten Rechenschritt den einzigen Primfaktor der Zahl, nämlich 5, gefunden. Eine aus zwei großen Primzahlen bestehende Zahl müsste zunächst durch sämtliche kleinere Primzahlen geteilt werden, bevor ein Primfaktor gefunden wird.

3. Die Anwendung der Primzahlen in der Kryptographie

3.1 Einleitung

Seit über 2000 Jahren, wahrscheinlich sogar schon seit Erfindung der Schrift, versuchten die Menschen, Geheimschriften zu erfinden, um geheime Botschaften zu übermitteln, die nur vom gewünschten Empfänger gelesen werden konnten. Besonders im – noch längst nicht abgeschlossenen – Aufschwung der Informations- und Kommunikationstechnologie gewinnt die Datensicherheit an Wichtigkeit. Gerade große, international arbeitende, Firmen müssen auf die Sicherheit ihrer Daten vertrauen und Spionage und Manipulation durch die Konkurrenz oder die eigenen Mitarbeiter ausschließen können.

Beim Einsatz der Kryptographie gibt es zwei Ebenen:

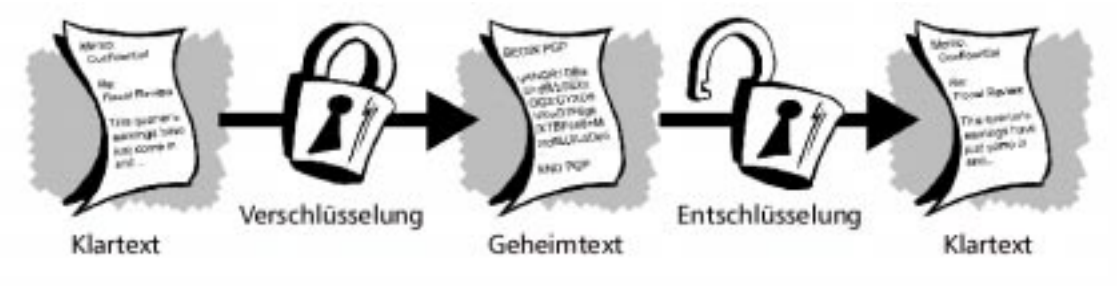
- Verschlüsselung von Botschaften
  Eine Nachricht wird in einen Geheimtext verwandelt, nur ein bestimmter Empfänger kann sie zurückverwandeln. Sie kann so von niemandem Unbefugtem gelesen werden.

- Authentifikation
  Eine Fälschung oder Manipulation einer Nachricht und deren Absender wird verhindert, die Herkunft kann so eindeutig bestimmt werden.

In diesem Kapitel werden zunächst die Grundlagen der Kryptographie behandelt. Später wird genauer auf die asymmetrische Kryptographie und den darauf basierenden RSA-Algorithmus eingegangen, da hier Primzahlen als mathematische Grundlage verwendet werden.

3.2 Definition der Kryptographie

Daten, die ohne besondere Mittel gelesen werden können, werden *Klartext* genannt. Der Vorgang durch Veränderung der Daten den Inhalt zu verstecken wird *Verschlüsselung* oder *Chiffrierung* genannt. Das Resultat ist der *Geheimtext*, auch *Chiffre* genannt. Verschlüsselung wird benutzt, um Daten vor Unbefugten zu schützen, auch wenn sie die chiffrierten Daten sehen können. Der Vorgang, verschlüsselte Daten wieder in Klartext umzusetzen, wird *Entschlüsselung* genannt.

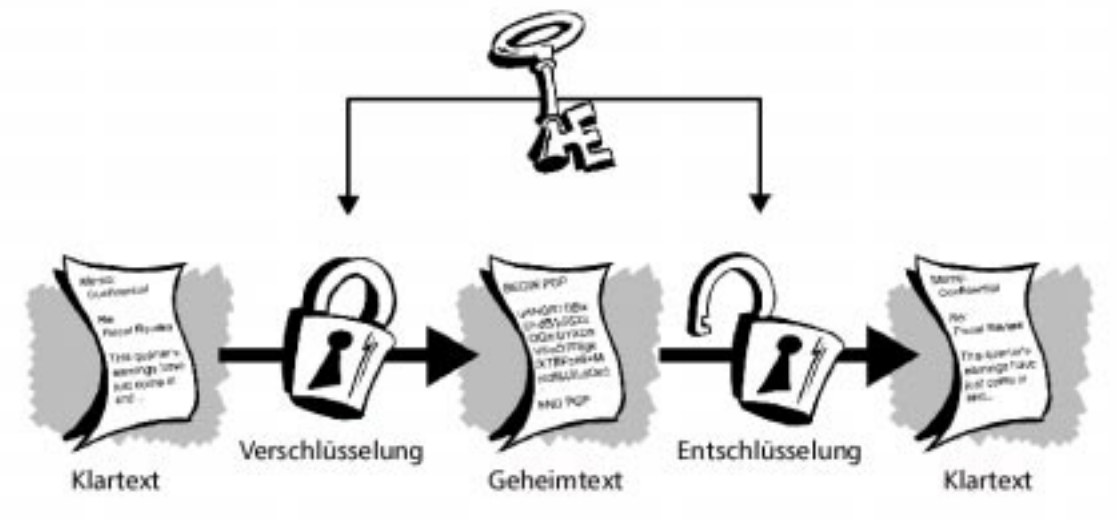


Die *Kryptographie* befasst sich mit dem Entwerfen von Chiffren. Mit deren Aufbrechen beschäftigt sich die *Kryptoanalyse*. Der Oberbegriff ist die *Kryptologie*.

Es wird zwischen zwei Arten der Kryptographie unterschieden: der symmetrischen Kryptographie und der asymmetrischen Kryptographie.

3.2.1 symmetrische Kryptographie

Bei der symmetrischen Kryptographie wird ein einziger Schlüssel zur Ver- und Entschlüsselung verwendet:



Ein sehr einfaches Beispiel für die symmetrische Kryptographie ist ein sogenannter Ersetzungs-Schlüssel. Dieser kommt meist bei der Ersetzung von Buchstaben des Alphabets zum Einsatz. Bei der nach der Idee *Julius Caesars* benannten *Caesar-Verschlüsselung* etwa wird das gesamte Alphabet um eine bestimmte Anzahl von Buchstaben verschoben, im Fall Caesars um drei Buchstaben. Augustus wendete das gleiche Verfahren an, verschob jedoch nur um einen Buchstaben:

Klartextalphabet	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Geheimtextalphabet (Caesar):	DEFGHIJKLMNOPQRSTUVWXYZABC
Geheimtextalphabet (Augustus):	BCDEFGHIJKLMNOPQRSTUVWXYZA

Um einen Text zu verschlüsseln musste man also nur den Buchstaben aus dem Klartextalphabet durch den direkt darrunterstehenden Buchstaben aus dem Geheimtextalphabet ersetzen. „VENI VIDI VICI“ (ich kam, sah und siegte) wurde so bei Caesar zu „YHGL YLGL YLFL“.

Wenn man diese Verschlüsselungsmethode etwas variiert, hat man 25 verschiedene Schlüssel zur Verfügung. Durch Mischen der Buchstaben (die Reihenfolge des Alphabets geht verloren) kann man maximal 26!, also ungefähr 10^{26} Schlüssel herstellen.

Allerdings ist es ziemlich einfach, einen so verschlüsselten Text zu entschlüsseln. In der deutschen Sprache – sowie auch in anderen Sprachen – kommen verschiedene Buchstaben mit einer bestimmten Häufigkeit vor. Man muss also nur die Häufigkeit der Buchstaben im verschlüsselten Text mit der statistischen Häufigkeit vergleichen. Im Falle der Caesar-Verschlüsselung muss man sogar nur den häufigsten Buchstaben im verschlüsselten Text finden und hat somit fast sicher das „E“ gefunden (17% Häufigkeit in einem deutschen Text). Nun kann man die Verschiebung sofort ablesen und erhält damit das Geheimtextalphabet nach Caesar.

Diese Verschlüsselungstechnik wurde noch im ersten Weltkrieg von der russischen Armee praktiziert.

Nachteile der symmetrischen Kryptographie:

- Jeder der verschlüsseln kann, kann auch entschlüsseln.
- Beide Partner müssen einen gemeinsamen geheimen Schlüssel tauschen.

3.2.2 asymmetrische Kryptographie

Angesichts der Nachteile der symmetrischen Kryptographie wurde ein neues Verfahren entwickelt, das deren Schwachstelle, den Schlüsseltausch, durch die Einführung von zwei unabhängigen Schlüsseln umgeht.

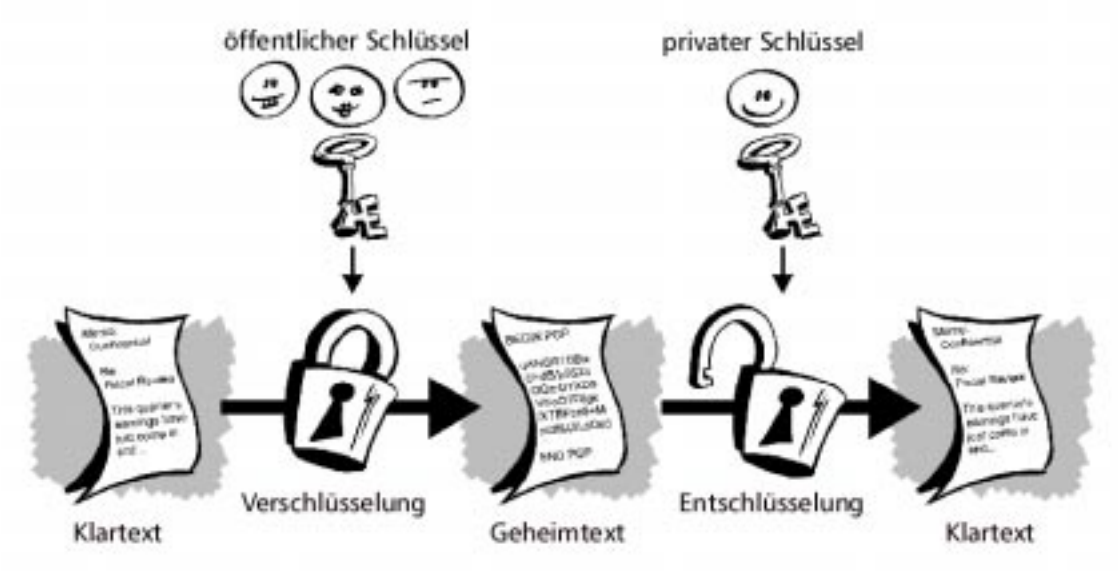
Das Konzept dieser asymmetrischen Kryptographie, auch „public-key"–Kryptographie genannt, wurde 1975 von *Whitfield Diffie* und *Martin Hellmann* vorgestellt (es stellte sich heraus, dass der Geheimdienst Großbritanniens dieses Verfahren schon ein paar Jahre vor Diffie und Hellmann erfand, es aber geheimhielt und nicht verwendete).

Zur Verschlüsselung wird ein *Schlüsselpaar* benötigt:

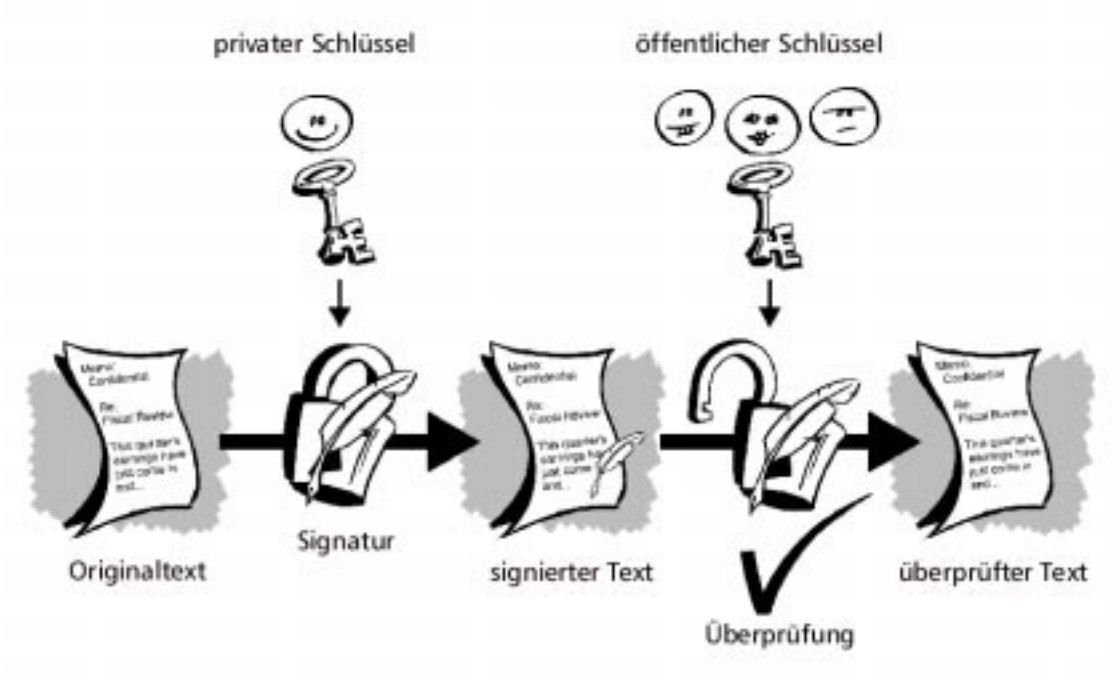
1. ein *öffentlicher Schlüssel* E, der an den Empfänger weitergegeben wird, oder der sogar veröffentlicht werden kann

2. ein *privater Schlüssel* D, der geheimgehalten wird.

Die Schlüssel haben die Eigenschaften, dass man Botschaften entweder mit D verschlüsseln und mit E entschlüsseln kann oder mit E verschlüsseln und mit D entschlüsseln. Zudem kann man aus dem öffentlichen Schlüssel nicht den privaten Schlüssel herleiten. Diese Eigenschaft macht die Sicherheit dieses Verfahrens aus.



Ein weitere Anwendung liegt in der *digitalen Unterschrift*: hierbei wird die Nachricht mit dem eigenen privaten Schlüssel verschlüsselt. Da der öffentliche Schlüssel frei zugänglich ist, kann damit die Nachricht von jedem entschlüsselt werden; dadurch wird sichergestellt, dass die Nachricht wirklich vom angegebenen Absender stammt und auf dem Weg zum Empfänger nicht verändert werden kann, da sie ansonsten nicht mehr mit dem öffentlichen Schlüssel entschlüsselt werden kann.



Anwendungsbeispiele:

- **Verschlüsselung**: Person A, in diesem Beispiel *Agathe* genannt, will eine geheime Botschaft an B wie *Benedikt* schicken. Sie sieht nach, wie der öffentliche Schlüssel von Benedikt lautet, verschlüsselt die Botschaft damit und schickt sie an Benedikt. Benedikt benutzt seinen privaten Schlüssel und entschlüsselt die Botschaft. Nur Benedikt kann Botschaften lesen, die mit seinem öffentlichen Schlüssel verschlüsselt wurden.

- **Digitale Unterschrift**: Agathe möchte eine Botschaft an Benedikt schicken, so dass Benedikt weiß, dass die Botschaft auch wirklich von Agathe stammt. Hierzu verschlüsselt sie ihre Botschaft mit ihrem geheimen Schlüssel. Benedikt muss nun ihren öffentlichen Schlüssel verwenden, um die Botschaft zu entschlüsseln. Wenn sie korrekt entschlüsselt wird, kann sie nur von Agathe stammen.

Vor- und Nachteile

Verschlüsselungsverfahren mit einem öffentlichen Schlüssel bieten folgende **Vorteile**:

- Es ist kein Schlüsseltausch notwendig. Dies bedeutet weniger Aufwand und mehr Sicherheit.
- Durch die Trennung in öffentliche und private Schlüssel kann jeder verschlüsseln, jedoch nur der Besitzer des privaten Schlüssels entschlüsseln.
- Die Echtheit von Dokument und Absender kann durch eine *digitale Unterschrift* gewährleistet werden.

Sie besitzen jedoch auch **Schwachstellen**:

- Asymmetrische Verfahren sind relativ rechenintensiv und dadurch langsam.
- Die Geheimhaltung des privaten Schlüssels ist wichtig.
- Öffentliche Schlüssel können evtl. gefälscht werden.

3.3 Das RSA-Verfahren

1977 entwickelten *Ronald L. Rivest*, *Adi Shamir* und *Leonard Adelman* ein asymmetrisches Verschlüsselungsverfahren, das man heute nach den Anfangsbuchstaben ihrer Nachnamen *RSA* nennt. Dieses Verschlüsselungsverfahren erfüllte die folgenden Forderungen:

- Chiffrier- und Dechiffrierschlüssel sind verschieden
- Der Chiffrierschlüssel kann öffentlich bekannt sein, während der Dechiffrierschlüssel geheim bleibt.

3.3.1 Grundlagen

Grundlage des RSA-Algorithmus ist der **Satz von Euler**:

Sind *m* und *n* zwei natürliche teilerfremde Zahlen, dann gilt:

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

$\varphi(n)$ ist die Anzahl der zu *n* teilerfremden natürlichen Zahlen.

die Anzahl aller Zahlen kleiner gleich *n*, deren größter gemeinsamer Teiler mit *n* gleich 1 ist.

Beispiele:

$\varphi(10) = 4$: Teilerfremde Zahlen sind $\{1,3,7,9\}$

$\varphi(11) = 10$: alle Zahlen von 1 bis 10 sind teilerfremd, da es sich um eine Primzahl handelt

Zu einer Primzahl *p* sind alle Zahlen von 1 bis (*p*-1) teilerfremd: $\varphi(p) = p - 1$

Für das Produkt zweier Primzahlen *p* und *q* ergibt sich also die Beziehung:

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

3.3.2 Schlüsselerzeugung

Grundlage für die Erzeugung des öffentlichen Schlüssels (des Codierschlüssels) und des privaten Schlüssels (des Decodierschlüssels) sind zwei Primzahlen p und q. Mit deren Produkt $n=pq$ und dem Satz von Euler wird zunächst $\varphi(n) = (p-1)\cdot(q-1)$ bestimmt. Für eine sichere Verschlüsselung sollten die beiden Primzahlen p und q in einer Größenordnung von mindestens 10^{200} liegen, dadurch wird ein Errechnen der beiden Primfaktoren von n in absehbarer Zeit unmöglich.

Codierschlüssel c und Decodierschlüssel d werden durch die Formel

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

berechnet. c muss hierzu teilerfremd zu $\varphi(n)$ sein; dies kann entweder dadurch erreicht werden, indem für c eine Primzahl gewählt wird, oder indem man eine Zahl c bestimmt, die kleiner als $\varphi(n)$ ist und deren ggT mit $\varphi(n)$ eins ist. Außerdem sollten die größten gemeinsamen Teiler von $c-1$ und $p-1$ bzw. von $c-1$ und $q-1$ möglichst klein sein.

Da bei der Ganzzahlendivision von $c \cdot d$ durch $\varphi(n)$ der Rest 1 bleibt, kann das Produkt von c und d auch folgendermaßen geschrieben werden:

$$c \cdot d = 1 + k \cdot \varphi(n)$$
$$c \cdot d = 1 + k \cdot (p-1)\cdot(q-1) \qquad k \in \mathbb{N}$$
$$d = \frac{1 + k \cdot (p-1)\cdot(q-1)}{c}$$

Nun werden n und Codierschlüssel c veröffentlicht, p und q werden vernichtet. Es ist fast unmöglich aus n ohne Kenntnis von p und q $\varphi(n)$ zu ermitteln, mit der Kenntnis von n und c kann also nicht der Decodierschlüssel d errechnet werden.

3.3.3 Ver- und Entschlüsselung

Da nur Zahlen verschlüsselt werden können, muss der Klartext t zunächst in eine Zahlenfolge umgewandelt werden. In der Praxis geschieht dies über den ASCII-Code (American Standard Code for Information Interchange).

Der Geheimtext *g* ist der Rest der Ganzzahlendivision aus dem Klartext *t* potenziert mit dem Codierschlüssel *c* und *n*:

$$g = t^c \bmod n$$

Analog dazu erfolgt die Entschlüsselung:

$$t = g^d \bmod n$$

Mit dem RSA-Algorithmus kann weiterhin eine *digitale Unterschrift* realisiert werden, indem das Prinzip der Ver- und Entschlüsselung umgekehrt wird. Es wird zwischen *universeller* und nicht *universeller Unterschrift* unterschieden:

- **universelle Unterschrift**:
  Die Unterschrift wird mit dem eigenen Decodierschlüssel chiffriert und an den Empfänger verschickt. Mit Hilfe des öffentlichen Schlüssels kann dieser die Chiffrierung rückgängig machen und erhält so die unverschlüsselte Unterschrift. Da nur der Besitzer des Decodierschlüssels die Unterschrift chiffriert haben kann, wird so die Authentizität des Absenders gewährleistet.

- **nicht universelle Unterschrift**:
  Bei der nicht universellen Unterschrift wird die mit dem eigenen Decodierschlüssel chiffrierte Unterschrift zusätzlich mit dem öffentlichen Schlüssel des Empfängers verschlüsselt, damit nur der Empfänger die Unterschrift entschlüsseln und identifizieren kann.

Am 19. Januar 2000 wurde von der EU die Richtlinie über elektronische Signaturen veröffentlicht und in Kraft gesetzt, die Grundlage dafür hatte die Bundesregierung schon 1997 mit dem *Signaturgesetz* (Art. 3 des *Informations- und Kommunikationsdienste-Gesetzes*) geschaffen. Damit erhält ein per e-mail übermitteltes Dokument, das mit einer digitalen Unterschrift signiert ist, einen rechtlich verbindlichen Status. So könnten in Zukunft Besuche in Ämtern, wie z.B. dem Einwohnermeldeamt, dem Finanzamt, oder einer Kfz-Zulassungsstelle vermieden werden.

Ein einfaches Rechenbeispiel zum RSA-Verfahren ist im Anhang B zu finden.

3.3.4 Der Einsatz von RSA in Computerprogrammen

Aufgrund des immer weiter ansteigenden Einsatzes von Computern in allen Lebensbereichen und der daraus folgenden steigenden Verantwortung dieser Systeme, gewinnt die Sicherheit stark an Bedeutung.

In **Unternehmensnetzwerken**, beispielsweise beim Netzwerksystem Novell Netware, kommt das RSA-Verfahren bei der Benutzerauthentifizierung zum Einsatz. Passwörter werden hier verschlüsselt übertragen, womit die Möglichkeit ausgeschlossen wird, dass ein Passwort in irgendeiner Weise abgehört und Missbraucht werden kann. So wird unbefugter Zugriff auf sensible Daten verhindert.

Auch bei der elektronischen Post, der **e-mail**, kommt es zunehmend auf Abhörsicherheit an. So kann eine unverschlüsselt gesendete e-mail an jedem Knotenpunkt im Internet, den sie auf dem Weg zum Empfänger passiert, gelesen und sogar unbemerkt verändert werden. Um dies zu vermeiden, kann die Nachricht verschlüsselt, oder – wenn es „nur" um die Sicherstellung der Echtheit des Absenders geht – digital signiert werden. Standards für sichere Internet-Transfers, wie *SSL* und *S/MIME*, ebenfalls zum größten Teil auf dem RSA-Algorithmus.

Das wohl bekannteste Verschlüsselungsprogramm für PC ist **PGP (Pretty Good Privacy)**. Es wurde von *Philip Zimmermann* ab 1991 entwickelt und 1997 von der Firma Network Associated übernommen.

Neben symmetrischen Verfahren wie dem *IDEA*-Algorithmus wird RSA hier als eine von mehreren Möglichkeiten für die asymmetrische Verschlüsselung unterstützt. PGP verschlüsselt nicht nur e-mail, sondern jegliche Art von Daten, und es ist so sicher, dass es unter das militärische Gesetz zur Spionageabwehr fiel und zeitweise nicht mehr exportiert werden durfte. Philip Zimmermann, äußerte sich dazu folgendermaßen:

> *„Regierungen haben eine Menge Geheimnisse vor ihrem Volk. ... Warum darf das Volk im Gegenzug keine Geheimnisse vor der Regierung haben?"*

Die Möglichkeiten des **Online-Banking**, **Online-Shopping** und der Zugriff auf **vertrauenswürdige Dokumente** über das Internet fordern ebenfalls Abhörsicherheit und eine eindeutige Absender-Authentifizierung.

3.3.5 Mögliche Schwachstellen von RSA

Neben den in Kapitel 3.2.2 schon aufgeführten Nachteilen der asymmetrischen Kryptographie, nämlich

- der hohen Anforderungen an die Rechenleistung
- der unbedingt notwendigen Geheimhaltung des privaten Schlüssels und
- der – wenn auch eher unrealistischen – Möglichkeit des Fälschens von öffentlichen Schlüsseln

hat RSA noch weitere Schwachstellen:

- Für eine hohe Sicherheit müssen relativ große Schlüssel verwendet werden. Das Verfahren kann deshalb z.B. in vielen Chipkarten mit niedriger Kapazität nicht realisiert werden.
- Die Verwendung von bestimmten Zahlen als Ausgangspunkt für die Schlüsselerzeugung muss unbedingt vermieden werden, da manche Primzahlprodukte sehr schnell faktorisiert werden können (siehe Kap. 2.6)
- Die Erfindung einer Möglichkeit zur schnellen und einfachen Erzeugung von großen Primzahlen oder zur schnelleren Primfaktorzerlegung wäre das *Aus* für den RSA-Algorithmus, da dessen Sicherheit ausschließlich auf der komplizierten Faktorisierung von großen Primzahlprodukten beruht.

3.4 Schlussbemerkung

Durch die immer weiter fortschreitende globale Vernetzung werden Verschlüsselungstechniken immer wichtiger, die Anforderungen an die Sicherheit wird immer höher. Da die Entwicklung der Computertechnologie immer schneller voranschreitet, können Computer immer komplexere mathematische Probleme (wie die Faktorisierung von großen Zahlen) in immer kürzerer Zeit lösen. Für das RSA-Verfahren heißt das, man muss, um diesem Phänomen entgegenzuwirken, die verwendete Schlüssellänge immer weiter erhöhen und dadurch eine langsamere Verarbeitungsgeschwindigkeit und höhere Kapazitätsanforderungen in Kauf nehmen.

Beispiele für eine sinnvolle Koexistenz von symmetrischer und asymmetrischer Kryptographie sind Verfahren, die Daten mit einem symmetrischen Algorithmus und einem gemeinsamen Schlüssel verschlüsseln, bei denen der Schlüsseltausch aber durch asymmetrische Verfahren geschützt ist. Sie haben den Vorteil, dass die Verschlüsselung schneller geht, das Sicherheitsrisiko beim Schlüsseltausch aber dennoch ausgeschaltet wird.

Es werden also neue Verschlüsselungsstrategien gefordert, nicht als Ersatz, sondern als Ergänzung.

Einen weiteren Ansatz bietet das *ElGamal*-Verfahren. Hier wird die Sicherheit durch die Schwierigkeit bei der Bestimmung von diskreten Logarithmen erreicht.

Steganoraphische Verfahren machen es möglich, Informationen jeder Art in Bildern oder in Audiodateien zu verstecken. Ohne genaues Wissen z.B. über die Position der – für den Bildbetrachter unsichtbaren – Nachricht in einem Bild, ist es so gut wie unmöglich, sie zu finden.

Eine zuverlässige Prognose zur weiteren Entwicklung der Kryptographie lässt sich nicht stellen, da auf diesem Gebiet nach wie vor sehr viel geforscht wird und die Geschwindigkeit der gesamten technologischen und naturwissenschaftlichen Entwicklung nur schwer voraussagbar ist.

Anhang A:

Tabelle der bekannten Mersenne-Primzahlen

Alle bisher bekannten Primzahlen p, für die gilt $p = 2^n - 1$

Nr.	Exponent n	Stellen	Entdeckungsjahr	Entdecker
1	2	1	—	—
2	3	1	—	—
3	5	2	—	—
4	7	3	—	—
5	13	4	1456	anonym
6	17	6	1588	Cataldi
7	19	6	1588	Cataldi
8	31	10	1772	Euler
9	61	19	1883	Pervushin
10	89	27	1911	Powers
11	107	33	1914	Powers
12	127	39	1876	Lucas
13	521	157	1952	Robinson
14	607	183	1952	Robinson
15	1279	386	1952	Robinson
16	2203	664	1952	Robinson
17	2281	687	1952	Robinson
18	3217	969	1957	Riesel
19	4253	1281	1961	Hurwitz
20	4423	1332	1961	Hurwitz
21	9689	2917	1963	Gillies
22	9941	2993	1963	Gillies
23	11213	3376	1963	Gillies
24	19937	6002	1971	Tuckerman
25	21701	6533	1978	Noll & Nickel
26	23209	6987	1979	Noll
27	44497	13395	1979	Nelson & Slowinski
28	86243	25962	1982	Slowinski
29	110503	33265	1988	Colquitt & Welsh
30	132049	39751	1983	Slowinski
31	216091	65050	1985	Slowinski
32	756839	227832	1992	Slowinski & Gage
33	859433	258716	1994	Slowinski & Gage
34	1257787	378632	1996	Slowinski & Gage
35	1398269	420921	1996	Armengaud, Woltman (GIMPS)
36	2976221	895932	1997	Spence, Woltman (GIMPS)
37	3021377	909526	1998	Clarkson, Woltman, Kurowski (GIMPS)
38	6972593	2098960	1999	Hajratwala, Woltman, Kurowski (GIMPS)

Anhang B

Rechenbeispiel für das RSA-Verfahren

An dieser Stelle ein kurzes Rechenbeispiel für die Anwendung des RSA-Verfahrens zur Chiffrierung von Zahlen.

Generierung der Schlüssel:

- Zwei Primzahlen: $p = 7$; $q = 17$
- $n = p \cdot q = 7 \cdot 17 = 119$
- $c = 5$, da 5 teilerfremd zu $\varphi(n) = (p - 1) \cdot (q - 1) = 96$ und $5 < n = 119$
- $d = 77$, da $\frac{d \cdot c - 1}{\varphi(n)} = \frac{77 \cdot 5 - 1}{96} = 4$ (die Division geht auf)

➔ öffentlicher Schlüssel $\boldsymbol{c = 5}$ privater Schlüssel $\boldsymbol{d = 77}$; $\boldsymbol{n = 119}$

Verschlüsselung:

- Klartext: $t = 12$
- Geheimtext: $g = t^c \bmod n = 12^5 \bmod 119 = 3$

Entschlüsselung:

- Geheimtext: $g = 3$
- Klartext: $t = g^d \bmod n = 3^{77} \bmod 119 = 12$

Digitale Signatur

- Klartext: $t = 12$
- Signierter Text: $g = t^d \bmod n = 12^{77} \bmod 119 = 31$
- Überprüfung: $t = g^c \bmod n = 31^5 \bmod 119 = 12 = t$

Auf die Verwendung von großen Zahlen für p und q wird hier der Einfachheit halber verzichtet.

Literaturverzeichnis

Creutzig, Christopher und **Buhl, Andreas**: PGP – Pretty Good Privacy, Deutsches Handbuch; 4. Auflage, Oktober 1999, FoeBuD e.V., Bielefeld

Glatfeld, Martin: Mathematik Lehren – Primzahlen I, Friedrich Verlag, Heft 57, April 1993b

Kippenhahn, Rudolf: Verschlüsselte Botschaften, Rowohlt Taschenbuch Verlag GmbH, Dezember 1999; S.287ff

Network Associates, Inc.: An Introduction to Cryptography, USA, Juni 1999

Schönleber, Claus: Verschlüsselungsverfahren für PC-Daten, Poing, 1995, S. 181ff

Dr. Spiegel, Gerald: Gesicherter Umschlagplatz, in: c't 26/1999, Verlag Heinz Heise, S. 160ff

VEB Bibliographisches Institut: Natur – kleine Enzyklopädie, 16. überarbeitete Auflage, Leipzig 1966

Haselböck, Barbara und Rickert, Andrea: GMX-Info 4/2000, 30. Januar 2000

„Der RSA-Algorithmus" 8. April 1999
http://www.uni-mainz.de/Schulen/Nieder-Olm/projekte/infschul/kr_rsa.html

„Erastothenes – Founder of mathematical geography", Dezember 1999
http://maps.unomaha.edu/Peterson/carta/Notes/eratosthenes.html

„Great Internet Mersenne Prime Search (GIMPS)", 2. Januar 2000
http://www.mersenne.org/

„Informations- und Kommunikationsdienste-Gesetz – IuKDG", 13. Juni 1997
http://www.iid.de/rahmen/iukdgk.html

„Pierre de Fermat", Dezember 1996
http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fermat.html

„Otto Forster's home page", 5. November 1999
http://www.mathematik.uni-muenchen.de/~forster/

„Sieve of Erastothenes", 3. September 1998
http://www2.andrews.edu/~calkins/math/webtexts/sieveOfErastothenes.htm

„The Prime Pages", Dezember 1999
http://www.utm.edu/research/primes/

„Verschlüsselungsverfahren - Darstellung und Aufwandsabschätzung", Jan. 2000
http://www.wirtschaft.tu-ilmenau.de/~punisher/

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Facharbeit

- ohne fremde Hilfe angefertigt und
- nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt

habe.

München, den 1. Februar 2000, _____

http://www.nicokaiser.de/facharbeit/