GYMNASIUM LAURENTIANUM WARENDORF

Kryptographie

Beschreibung und Aufbau symmetrischer und asymmetrischer Kryptosysteme am Beispiel von RSA und dem diskreten Logarithmus-Problem unter Betrachtung der Umdefinierung auf elliptischen Kurven

Benedikt Buller

05.03.2015

Betreuer: Hannes Stoppel

Eine schriftliche Ausarbeitung bezüglich kryptographischer Sachverhalte, die im Rahmen einer besonderen Lernleistung verfasst wurde.

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Einleitung

Politisch-geschichtlicher Hintergrund

In unserer heutigen modernen vernetzten Gesellschaft ist ein zentraler Gesichtspunkt einer zuverlässigen elektronisch digitalen Infrastruktur, dass Informationen über genau definierte Verbindungen an einen ganz bestimmten Adressaten übermitteln werden. Zwecks der Realisierung dieses Gesichtspunktes ist der Einsatz kryptografischer Methoden unabdingbar, da es das Ziel Außenstehender und Unbefugter seit jeher war und ist, Informationen durch Datenaustausch zu bekommen, welche nicht für diese bestimmt sind. Zum Beispiel wickelten schon im alten Griechenland die Spartaner um einen Zylinder von genau definiertem Radius ein Pergamentband, auf das dann die Nachricht notiert wurde. Diese war ohne den Zylinder unlesbar, da alle Buchstaben zufällig vertauscht schienen. Nur diejenigen, die den Radius des Zylinders (manchmal auch anderer Rotationskörper) kannten, konnten die Wicklung erneut vornehmen und die Information verstehen. Alle anderen sollten eben dies nicht schaffen. Auch in der heutigen Zeit nach der digitalen Revolution der Computer und der schnellen Rechnernetze ist es für Staaten, Wirtschaft und Unternehmen existenziell wichtig dem Thema Datensicherheit höchste Priorität zu geben. Die Cyberkriminalität und deren erfolgreiche Bekämpfung wird heute und in der Zukunft eine fortlaufend hochaktuelle Herausforderung sein.
Viele neue Geschäftskonzepte sind im Zuge der allgemeinen Bewusstwerdung des unangenehmen Gedankens, man könne ausspioniert werden, entstanden. Den deutschen Gründerpreis 2014 konnte zum Beispiel ein Unternehmen namens “Secomba[1]“ gewinnen. Ihr ursprüngliches Konzept war es, Daten, die ein Nutzer in einer Cloud speichern will, vorab auf dem Nutzungsgerät zu verschlüsseln und dann erst hochzuladen. Der Nutzer kann genau steuern, welchen Geräten der Zugriff erlaubt wird und welchen nicht. So ist für den Nutzer, der die Cloud-Dienste nicht durchschauen kann, gewährleistet, dass niemand durch Hintertüren der Server Zugriff auf Informationen erhält. Die Aktualität der Kryptografie ist also mehr als gegeben.

Was zeichnet Kryptografie aus?

Das Besondere an Kryptografie, was kaum ein anderes mathematisches Gebiet so realisieren kann, ist die enorme mathematische Vielfalt von beliebiger Komplexität und zugleich die Eigenschaft, dass immer eine echte realistische Anwendung dieser Mathematik mal mehr und mal weniger im Hintergrund steht. So werden die Themen, die in der Schulmathematik angeschnitten werden (Analysis, lineare Algebra, Statistik) mehr als abgedeckt. Themen wie Zahlentheorie, denen man es eher nicht zutrauen würde, finden Anwendung in der Kryptografie (z.B.RSA). Dabei kann man an simplen Beispielen wie z.B. der Caesar-

[1] Mehr dazu unter http://www.deutscher-gruenderpreis.de/preistraeger/2014/secomba/

Verschlüsselung arbeiten bis hin zu kompliziertesten algebraischen Strukturen über elliptischen Kurven, mit deren Hilfe z.B. auch der Modularitätssatz und damit der Satz des Fermat bewiesen wurde, vordringen.

Vor dem Hintergrund der Komplexität des Themas Datensicherheit, welche durch Geschichte, Aktualität und thematische Vielfalt gegeben ist, sind besondere Anreize dafür geschaffen, sich mit der Kryptografie zu beschäftigen und interessantes Wissen zu erlangen, das im Rahmen des normalen und standardisierten Schulunterrichts nicht erreichbar bzw. erbringbar ist. Aus diesem Gedanken entstand diese Auseinandersetzung mit dem kryptografischen Bereich.

Persönliche Ziele

Mein primäres Ziel ist es ein solides, fundiertes Wissen über Methoden der gezielten Unkenntlichmachung von Informationen zu erlangen. Dabei will ich auch ein Verständnis für die Methodiken entwickeln und will nicht der Versuchung erliegen bei komplexen Inhalten Informationen auswendig lernen oder weglassen, wenn sie von Bedeutung für die Lösung eines Problems sind.
Wichtiges zweites persönliches Ziel bezüglich der Inhalte ist es, mich selbst zu verwirklichen, indem ich mir die Freiheit nehme, teilweise auch auf theoretische Konzepte abseits eines bestimmten kryptografischen Verfahrens einzugehen und so den Blick fürs ‘‘große Ganze‘‘ habe. Dazu gehört auch neben der vielen Theorie ein praktischer Teil in Form eines selbstgeschriebenen Programms, um sich auch der praktischen Implementierungsprobleme bewusst zu werden. Konkret handelt es sich dabei um eine Java Anwendung, bei der man Text mithilfe des RSA-Verfahrens verschlüsseln und entschlüsseln kann und die Informationen zum Kryptographie-System anzeigt.
Nach Abschluss dieses Projektes wird mir nicht eine Kenntnis des gesamten Gebiets gelingen, jedoch ist mein drittes Ziel, dass ich mir viele Unterthemen des Gebiets, deren Existenzen mir dann bewusst geworden sind, die ich aber im Speziellen nicht kenne, dennoch auf Grundlage des fundierten Wissens schnell aneignen kann.

Hinweis: Im Folgenden werden mathematische Operationen/Operatoren als bekannt vorausgesetzt. Hierzu zählen u.a. Modulo-Operation, Summenoperator, Produktoperator.

Prinzipien der Kryptographie

Zwecks des Schutzes von Informationen bedient sich die Kryptographie im Wesentlichen an vier Orientierungspunkten, an denen sich moderne kryptographische Verfahren orientieren. Zum einen soll das Verfahren die Information an sich nach dem Prinzip der Integrität möglichst nicht verändern. Des Weiteren soll eine Nachricht eindeutig zurück verfolgbar sein bzw. der Absender bekannt sein, und diesem soll es nicht möglich sein abzustreiten, dass er

die Nachricht gesendet hat. Es gelten also die Prinzipien der verbindlichen Authenzität. Als oberstes Ziel einer Geheim-Chiffrierung gilt jedoch die Sicherheit gegen Zugriffe von außen nach dem Prinzip der Vertraulichkeit zu gewährleisten. Nur dazu autorisierten Instanzen soll es möglich sein, Informationen aus der verschlüsselten Nachricht zu gewinnen.

1 Mathematische Grundbegriffe

Um ein systematisches Verständnis dafür aufzubauen, das den verschiedenen Kryptoverfahren allgemein zu Grunde liegt, muss zunächst auf einige zentrale Begriffe u.a. der Gruppentheorie eingegangen werden.

1.1 Menge

Am fundamentalsten ist der Begriff der Menge, der üblicherweise mit der Definition von Gregor Cantor definiert wird:

„Definition (Georg Cantor): Unter einer Menge verstehen wir jede Zusammenfassung M von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens zu einem Ganzen."

Nach dieser Definition muss eine Menge nicht unbedingt aus Zahlen bestehen, sondern kann auch aus anderen Objekten bestehen. Diese werden bezüglich einer Menge Elemente der Menge genannt.

Um zu signalisieren, dass ein Objekt aus einer bestimmten Menge stammt, verwendet man das Zeichen „$\in$" (gesprochen: „ist Element von") (vgl. [GB] S.8; [WR])

1.2 Gruppe

Definition 1.1:

Gegeben sei eine Menge M und eine Operation ʘ. Eine algebraische Struktur G(M; ʘ) heißt genau dann Gruppe, wenn folgende Voraussetzungen gelten:

1. $\forall\, a \in M, b \in M. \exists c = a \odot \mathrm{b} \in M$

   Die Menge ist bezüglich der jeweiligen Operation abgeschlossen.

2. $\forall\, a \in M, b \in M, c \in M.\ (a \odot \mathrm{b}) \odot \mathrm{c} = \mathrm{a} \odot (b \odot \mathrm{c})$

   Es gilt das Assoziativgesetz.

3. $\forall\, a \in M. \exists n \in M.\ a \odot n = a = n \odot a$

   Es existiert ein neutrales Element, das jedes Element der Menge M bezüglich der Operation auf sich selbst abbildet.

4. $\forall\, a \in M. \exists b \in M. a \odot \mathrm{b} = \mathrm{n}$

Es existiert zu jedem Element a der Menge M ein Element b der Menge M, das bezüglich der Operation a auf das neutrale Element abbildet. b ist das Inverse von a.

Eine Gruppe heißt abelsch, wenn zuzüglich der vier genannten Voraussetzungen auch folgende gilt:

5. $\forall\, a \in M, b \in M.\ \ a \odot \mathrm{b} = b \odot \mathrm{a}$

Es gilt das Kommutativgesetz.

Eine Gruppe G(M; Θ) heißt Halbgruppe genau dann, wenn sowohl dass Axiom der Abgeschlossenheit(1.) als auch das Axiom der Assoziativität (2.) gilt. (vgl. [ATM] S.9, vgl. [AF] S.4)

1.3 Ring

Definition 1.2:

Gegeben sei eine Menge M und zwei Operationen Θ und ◊. Eine algebraische Struktur R(M; Θ; ◊) heißt genau dann Ring, wenn folgende Voraussetzungen gelten:

1. Die Gruppe G(M; Θ) ist eine abelsche (kommutative) Gruppe.
2. Die Gruppe H(M; ◊) ist eine Halbgruppe

3. $\forall\, a \in M, b \in M, c \in M.\ a \lozenge (\mathrm{b} \odot \mathrm{c}) = a \lozenge \mathrm{b} \odot a \lozenge \mathrm{c},\ (\mathrm{b} \odot \mathrm{c}) \lozenge \mathrm{a} = b \lozenge \mathrm{a} \odot c \lozenge \mathrm{a}$

Es gilt das Distributivgesetz. (vgl. [ATM] S.85, vgl. [WR])

1.4 Körper

Definition 1.3:

Gegeben sei eine Menge M mit mindestens 2 Elementen und zwei Verknüpfungen +; * (Addition, Multiplikation). Eine algebraische Struktur K(M;+;*) heißt Körper genau dann, wenn folgende Axiome erfüllt sind:

1. Die Gruppe G(M; +) ist eine abelsche Gruppe.
2. Die Gruppe H(M{/0};*) ist eine abelsche Gruppe.
3. $\forall\, a \in M, b \in M, c \in M.\ a \lozenge (\mathrm{b} \odot \mathrm{c}) = a \lozenge \mathrm{b} \odot a \lozenge \mathrm{c},\ (\mathrm{b} \odot \mathrm{c}) \lozenge \mathrm{a} = b \lozenge \mathrm{a} \odot c \lozenge \mathrm{a}$

Es gilt das Distributivgesetz. (vgl. [ATM] S.85)

Diese Definitionen ist genau deshalb so elementar, weil in der Verschlüsselung immer Informationen mittels einer bestimmten, wohldefinierten Operation unkenntlich gemacht werden mit dem Ziel, zu einem späteren Zeitpunkt (oder nur von ganz bestimmten

Personen) durch die entsprechende Umkehroperation oder durch Verwenden von inversen Elementen die ursprünglichen Informationen wieder kenntlich zu machen. Gruppen sind für diesen Zweck meistens unabdingbar, da ihr Axiom der inversen Elemente genau dies garantiert. Viele Kryptosysteme bestehen im Kern aus einer Gruppe, die unter bestimmten Vorrausetzungen (z.B. über einer Kurve) definiert sind.

1.5 Nabla Operator

Der Nabla Operator ist ein Vektor, dessen Komponenten aus den partiellen Ableitungsoperatoren mit dem maximalen Index der individuellen Dimension und dem minimalen Index 1 besteht. Individuell kommt er als Zeilen- oder Spaltenvektor vor. Über ihn sind z.B. Gradient, Divergenz und Rotation definiert. Im Folgenden wird aber nur der Begriff des Gradienten als ‘‘Steigungsvektor‘‘ benötigt:

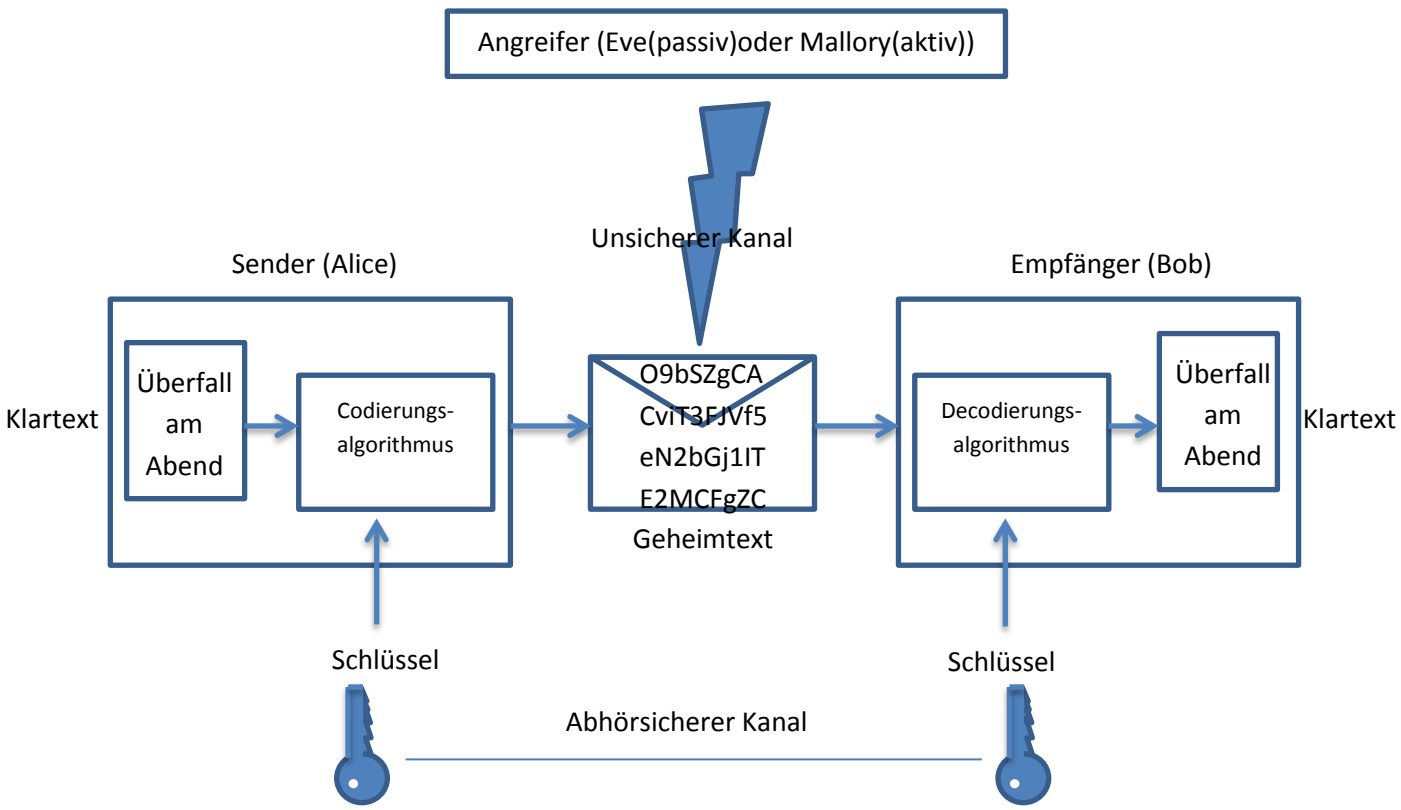
$$\vec{\nabla} =: \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \cdots \\ \cdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \qquad \vec{\nabla} f(x_1, x_2, \ldots, x_n) =: grad\left(f(x_1, x_2, \ldots, x_n)\right) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \cdots \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

(vgl. [GB] S.543)

2 Symmetrische Verschlüsselung

Ein symmetrisches Kryptoverfahren basiert darauf, dass die Teilnehmer zum Codieren und Decodieren den selben Schlüssel verwenden bzw., dass Sender und Empfänger identische, umfassende Informationen zu dem Kryptosystem kennen (bei asymmetrischen Verfahren ist dies z.B. nicht gegeben). Der Sender macht durch ein genau definiertes, vorher vereinbartes Verfahren eine Information unkenntlich. Die Nachricht wird abgesendet und der Empfänger ‘‘macht‘‘ genau die Schritte, auf die man sich geeinigt hat, umgekehrt ‘‘nach‘‘. Die Invertierbarkeit des Prozesses wird dabei als vergleichsweise einfach vorausgesetzt. Typische traditionelle Methoden, aus denen sich ein solcher Codierungsprozess zusammensetzt, sind monoalphabetische/polyalphabetische Substitution und Transposition (also die Ersetzung der Buchstaben auf Basis eines oder mehrerer Alphabete oder die systematische Vertauschung der Buchstaben innerhalb der Nachricht ohne diese jedoch zu verändern). Folgt man dem typischen Protokoll, muss dem Sender unter der Voraussetzung, dass man sich auf einen gemeinsamen geheimen Schlüssel über einen sicheren Kanal geeinigt hat, und der Codierungsalgorithmus und Decodierungsalgorithmus bekannt sind, zunächst ein Klartext vorliegen, der verschlüsselt werden soll. Diesen bildet er mit dem Codierungsalgorithmus (eine leicht invertierbare Funktion) auf den Geheimtext ab. Der Geheimtext wird nun an den Empfänger über einen unsicheren Kanal versendet. Ohne den Schlüssel soll es einem Angreifer unter Kenntnis des Verfahrens nicht möglich sein, an den Klartext zu gelangen. Der Empfänger wendet nun den Decodierungsalgorithmus an und erhält den Klartext. Visualisiert entspricht die symmetrische Verschlüsselung also folgendem Schema(vgl. [CM] S.99, [SV]):

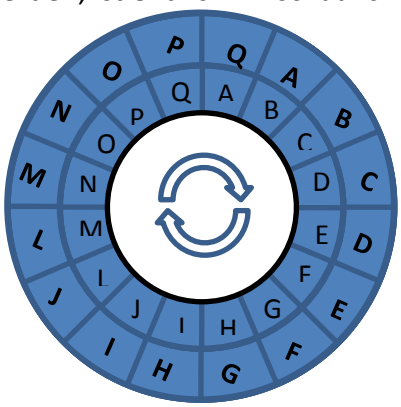
Abb. 1



2.1 Caesar Verschlüsselung

Die Caesar Verschlüsselung ist eine der ältesten symmetrischen Verschlüsselungsverfahren und wurde in abgewandelter Form schon von den Römern verwendet. Das Verfahren basiert auf linearer monoalphabetischer Substitution und bildet jede natürliche Zahl (oder Symbol) durch modulare Addition auf eine andere natürliche Zahl (oder Symbol)…ab. Die Anzahl der Elemente, die hierfür verwendet werden, ist endlich. Anschaulich funktioniert dies so:

Abb. 2



Im oberen Bild erkennt man zwei zu einander kreisförmig verschiebbare ‘‘Zeichenreihen“. Zum Verschlüsseln verschiebt man die innere Reihe und kann dann jedem Zeichen des Klartextes das neue Zeichen zuordnen, dass durch die Verschiebung bedingt nun ‘‘benachbart‘‘ sind. So erhält man dann den Geheimtext. Zum Entschlüsseln muss (sollte) man wissen, wie genau verschoben wurde, also um wie viele Stellen verschoben wurde, und kann dann den Vorgang umkehren, um aus dem Geheimtext wieder den Klartext zu erhalten.

Hält man den Schlüssel S (das Geheimnis) und die Zeichenanzahl x allgemein, so ergibt sich folglich folgende Funktion für den Geheimtext V in Abhängigkeit vom Klartext K:

$$V_S : \mathbb{Z}_x \to \mathbb{Z}, V_S(K) = K + S \; mod \; x$$

Um eine Funktion für den Klartext E in Abhängigkeit vom Geheimtext G zu erhalten, überlegt man sich, dass die inverse Operation zur Addition die Subtraktion ist bzw. das Addieren mit dem additiven modularen inversen Element von S ist:

$$E_S : \mathbb{Z}_x \to \mathbb{Z}, E_S(G) = G - S \; mod \; x$$

Ein Beispiel für die Verschlüsselung:

Verschlüsselt werden soll (Klartext T):

HALLO DIES IST VERSCHLÜSSELT

Man einigt sich auf eine Zuordnung von Symbolen auf Zahlen, also auf ein Alphabet. Dieses ist öffentlich einsehbar (public):

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
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	

Also:

7	0	11	11	14	26	3	8	4	18	26	8	18	19	26	21	4	17	18	2	7	11	20	18	18	4	11	19
H	A	L	L	O		D	I	E	S		I	S	T		V	E	R	S	C	H	L	U	S	S	E	L	T

Nun wendet man die modulare Addition für jeden Buchstaben einzeln an. Die Zeichenanzahl ist hier 27 und der Schlüssel exemplarisch 11 z.B. für H:

$$V_S(7) = 7 + 11 \; mod \; 27 = 18$$

Also ergibt sich für H codiert ein S. Führt man dies fort ergibt sich insgesamt:

18	11	22	22	25	10	14	19	15	2	10	19	2	3	10	5	15	1	2	13	18	22	4	2	2	15	22	3
S	L	W	W	Z	K	O	T	P	C	K	T	C	D	K	F	P	B	C	N	S	W	E	C	C	P	W	D

Zum Decodieren wendet man die Umkehrfunktion an z.B. für S:

$$E_S(18) = 18 - 11 \; mod \; 27 = 7$$

Also ergibt sich für den Buchstaben S decodiert wie erwartet ein H. Dies kann man für den gesamten Geheimtext durchführen und erhält den oben stehenden Klartext. (vgl. [IB] S.28)

Dieses Verfahren ist zwar gut dazu geeignet das Prinzip hinter symmetrischer Verschlüsselung klarzumachen, aber als effektiver Schutz gegen Abhörung und geheime Datenübertragung war es schon in der Antike völlig ungeeignet. Halbwegs gebildete Menschen konnten schon damals ohne Probleme die Nachricht herausfinden, indem sie alle Möglichkeiten (die Anzahl der Möglichkeiten ist durch die Zeichen Anzahl gegeben) ausprobierten oder eine Häufigkeitsanalyse durchführen.

Bei einer Häufigkeitsanalyse versucht man aus der Häufigkeit der auftretenden Buchstaben auf die tatsächlichen Buchstaben zu schließen. Zum Beispiel kommt das ``e`` in der deutschen Sprache statistisch gesehen erheblich häufiger vor als ein ‘‘y‘‘. Dadurch könnte man im oberen Beispiel z.B. fast ausschließen, dass sich hinter dem ‘‘W‘‘ welches 4mal vorkommt ein ‘‘y‘‘ verbirgt. Damit ist eine von 27 Kombinationen schon ausgeschlossen.

2.2 Vigenere Verschlüsselung

Der Unsicherheit der Caesarverschlüsselung verschaffte damals die Vigenere Verschlüsselung Abhilfe, die lange als relativ sicher galt und auf der in abgewandelter Form auch die Enigma-Maschinen des 2. Weltkrieges basieren. Der entscheidende Unterschied im Vergleich zur Caesar Verschlüsselung besteht darin, dass die Codierung nicht mehr monoalphabetisch sondern polyalphabetisch geschieht. Der Schlüssel ergibt sich also nicht

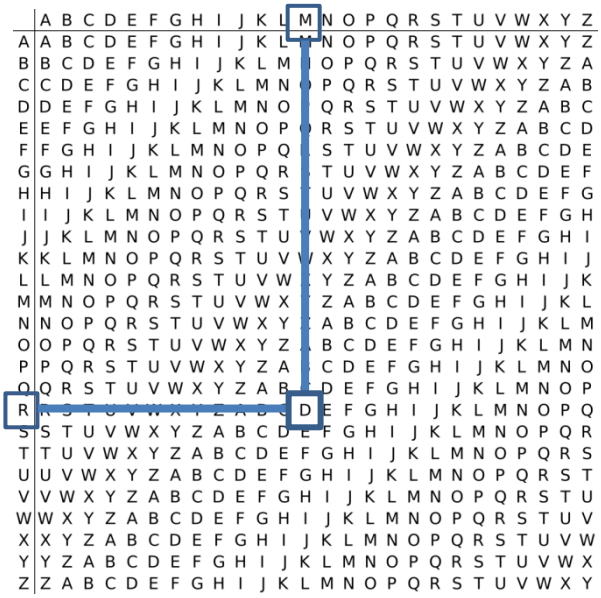
mehr nur aus einem Zeichen, sondern aus einer ganzen Zeichenkette. Die Verschlüsselung funktioniert folgendermaßen:

1. Man ‘‘schreibt‘‘ über das zu verschlüsselnde Wort den Schlüssel in folgender Art und Weise darüber(Geheimeswort, Schlüssel):

R	O	S	T	R	O	S	T	Schlüssel	Zeile
M	E	I	N	W	O	R	T	Wort	Spalte

2. Mithilfe der Tabula Recta (fälschlicher Weise oft Vigenere Quadrat) kann man nun die Codierung durchführen. Dabei bestimmt der Schlüssel, welche Zeile gewählt wird, und der zu codierende Text, welche Spalte gewählt wird. Zum Beispiel wird das M wie folgt verschlüsselt:

Abb. 3



3. Wer die verschlüsselte Botschaft und den Schlüssel hat, kann sehr ähnlich verfahren. Mit dem Schlüssel kann die Zeile herausgefunden werden und mit dem Geheimtext der Buchstabe in der entsprechenden Zeile. Daraus kann man auf die Spalte schließen und den Klartext herausfinden.

Der Vorteil dieser Verschlüsselung besteht darin, dass das selbe Zeichen auf verschiedenste Zeichen abgebildet wird. Ausprobieren führt nicht ohne weiteres zum Erfolg und die Möglichkeit zur Häufigkeitsanalyse wird erschwert. (vgl. [JV] S.17, [WP])

2.3 One-time-Pad

Bei dem symmetrischen Kryptoverfahren, das One-time-Pad genannt wird, handelt es sich um eine Variante der Vigenere Verschlüsselung, die ebenso auf polyalphabetischer Substitution basiert. Zum Verschlüsseln und Entschlüsseln wird ein Schlüssel verwendet, der mindestens so lang ist wie die zu verschlüsselnde Nachricht und zudem absolut zufällig ohne

Wortfragmente oder Wiederholungen konstruiert wurde. Hinzuzufügen ist außerdem, dass der Schlüssel ausschießlich einmal verwendet wird, dann ohne Vervielfältigung gelöscht wird und anschließend ein neuer generiert wird.

Ver- Entschlüsselung:

Gegeben sei der Klartext K die Länge des Alphabets n und der Schlüssel S. Den Geheimtext G erhält man durch:

$$G = K + S \; mod \; n$$

Den Klartext erhält man durch:

$$K = G - S \; mod \; n$$

Beispiel:

S = RZSLXWJFYUDM

r	z	s	l	x	w	j	f	y	u	d	m
18	26	19	12	24	23	10	6	25	21	4	13

K = STURMAMABEND

s	t	u	r	m	a	m	a	b	e	n	d
19	20	21	18	13	1	13	1	2	5	14	4

G = KTNDKXWGAZRQ

k	t	n	d	k	x	w	g	a	z	r	q
11	20	14	4	11	24	23	7	1	0	18	17

Dieses Verfahren wird sogar heute noch von Geheimdiensten verwendet, da es die Eigenschaft der perfekten Sicherheit besitzt. Das bedeutet, dass ein theoretisch möglicher Klartext auf ebenso viele Geheimtexte, wie es Klartexte geben könnte, abgebildet wird. Es liegt zudem eine absolute stochastische Unabhängigkeit vor, wie genau sich Abbildungen ereignen, so dass ein Angreifer aus dem Geheimtext keine Informationen, abgesehen von der Länge des Klartextes, schließen kann. Als Beispiel kann man eine Nachricht bestehend aus zwei Zeichen anführen. Für den Klartext gibt es 26*26=676 verschiedene Möglichkeiten. Auf Grund der zufälligen Schlüsselauswahl wird der konkrete Klartext zufällig auf eine von 26*26=676 Möglichkeiten abgebildet. Für einen Angreifer könnte die Nachricht alle Worte enthalten, die zwei Zeichen haben. Durch zusätzliche Verlängerung des Schlüssels lassen sich selbst solche Mutmaßungen und auch Mutmaßungen bezüglich der Länge der Nachricht verhindern. Nachteil der Verwendung dieses Verfahrens ist die große Schlüssellänge und der Aufwand ständig neue Schlüssel zu generieren. Dies macht das Verfahren oft für den normalen Alltag unbrauchbar.

2.4 Verschlüsselung mit Matrizen

Auch in der heutigen Zeit sind symmetrische Verfahren wie z.B. AES außerhalb von Geheimdiensten auf Grund des vergleichsweise geringen Rechenaufwandes von zentraler Bedeutung. Die meisten modernen Verfahren basieren aber nicht auf der Verschlüsselung einzelner Zeichen, sondern auf der Verschlüsselung ganzer Blöcke von Informationen, so wie es bei der Vigenere-Verschlüsselung in einfachster Form angedeutet wird. Typischerweise kann man, um dies umzusetzen, einen Block von Zeichen als Vektor ansehen und mit einer entsprechenden Abbildungsmatrix auf den Geheimtext abbilden:

2.4.1 Matrix als Schlüssel

Gegeben sei eine Matrix M mit:

$$M = \begin{pmatrix} 3 & 5 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix}$$

Der zu verschlüsselnde Klartext T sei:

$$T = 121314151$$

Also bestehen die Böcke B aus $B_1 = 121$, $B_2 = 314$, $B_3 = 151$

bzw:

$$\overrightarrow{B_1} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}; \overrightarrow{B_2} = \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix}; \overrightarrow{B_3} = \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix}$$

Verschlüsselt wird durch die Multiplikation des Vektors mit der Matrix. Die Vektoren des Klartextes werden so auf die Vektoren des Geheimtextes abgebildet:

$\begin{pmatrix} 3 & 5 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 14 \\ 15 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 3 & 5 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix} * \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 18 \\ 30 \\ 13 \end{pmatrix}$
$\begin{pmatrix} 3 & 5 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix} * \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 29 \\ 27 \\ 13 \end{pmatrix}$	$G = 14\ 15\ 07\ 18\ 30\ 13\ 29\ 27\ 13$

Um nun von dem Geheimtext an den Klartext zu gelangen muss der Geheimtext auf den Klartext abgebildet werden. Man benötigt folglich die Matrix, die die vorhergehende Verschlüsselung rückgängig macht. Diese Matrix heißt Inverse Matrix zu M. Um M^{-1} zu berechnen, bildet man eine Blockmatrix aus der zu invertierenden Matrix und dem neutralen Element der Matrizenmultiplikation, also der Einheitsmatrix E. Dann formt man die Matrix M mit entsprechenden Umformungsschritten zur Einheitsmatrix um und führt die selben

Umformungsschritte an der Einheitsmatrix durch. Die Einheitsmatrix wird dadurch zum multiplikativen Inversen der Matrix M. Denn man kann die Umformungsschritte U_n natürlich auch als Matrizen auffassen, mit denen die Matrix M multipliziert wird. Also:

$$1.\, M * \prod_{i=1}^{n} U_n = E$$

$$2.\quad M * M^{-1} = E$$

$$1. = 2.;\ M * M^{-1} = M * \prod_{i=1}^{n} U_n$$

$$\Rightarrow E * \prod_{i=1}^{n} U_n = M^{-1}$$

Das Inverse zu dem Beispielschlüssel M kann auf diese Weise errechnet werden. Dies gelingt z.B. durch das Anwenden des Schemas nach dem Gauß-Jordan-Verfahren(vgl. [GB] S.299):

$\left(\begin{array}{ccc\|ccc} 3 & 5 & 1 & 1 & 0 & 0 \\ 2 & 4 & 5 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 0 & 1 \end{array}\right) \begin{array}{l} \\ -2/3 * I \\ -1/3 * I \end{array}$	$\left(\begin{array}{ccc\|ccc} 3 & 5 & 1 & 1 & 0 & 0 \\ 0 & 2/3 & 13/3 & -2/3 & 1 & 0 \\ 0 & 1/3 & 5/3 & -1/3 & 0 & 1 \end{array}\right) -0{,}5 * II$
$\left(\begin{array}{ccc\|ccc} 3 & 5 & 1 & 1 & 0 & 0 \\ 0 & 2/3 & 13/3 & -2/3 & 1 & 0 \\ 0 & 0 & -1/2 & 0 & -0{,}5 & 1 \end{array}\right) \begin{array}{l} \\ \\ +2 * III \\ +26/3 * III \end{array}$	$\left(\begin{array}{ccc\|ccc} 3 & 5 & 0 & 1 & -1 & 2 \\ 0 & 2/3 & 0 & -2/3 & -10/3 & 26/3 \\ 0 & 0 & -1/2 & 0 & -0{,}5 & 1 \end{array}\right) -7{,}5 * II$
$\left(\begin{array}{ccc\|ccc} 3 & 0 & 0 & 6 & 24 & -63 \\ 0 & 2/3 & 0 & -2/3 & -10/3 & 26/3 \\ 0 & 0 & -1/2 & 0 & -0{,}5 & 1 \end{array}\right) \begin{array}{l} * 1/3 \\ * 3/2 \\ * (-2) \end{array}$	$\left(\begin{array}{ccc\|ccc} 1 & 0 & 0 & 2 & 8 & -21 \\ 0 & 1 & 0 & -1 & -5 & 13 \\ 0 & 0 & 1 & 0 & 1 & -2 \end{array}\right)$

$$\Rightarrow M^{-1} = \begin{pmatrix} 2 & 8 & -21 \\ -1 & -5 & 13 \\ 0 & 1 & -2 \end{pmatrix}$$

2.4.2 Matrizenschaar als Schlüssel

Durch die Linearität der Matrix und der Konstanz ihrer Elemente bestehen stochastische Abhängigkeiten, die ein Angreifer ausnutzen kann, um den Schlüssel herauszufinden. Mit zunehmender Länge der Nachricht werden die Möglichkeiten, Muster zu finden, mehr. Um die Sicherheit also zu verstärken, benutzt man in der modernen Kryptografie Schlüssel (in diesem Fall Verschlüsselungsmatrizen), die entweder von einem zusätzlichen Schlüssel abhängig sind (in der Regel eine Zahlenreihe) oder aber von der Nachricht selbst. Bei letzterer Methode besteht die Gefahr, dass die verschlüsselten Informationen auf Grund von

rekursiven Vorgängen gewaltige Werte annehmen, was die Auswahl der verwendbaren Matrizen erheblich einschränkt. Oft benutzt man z.B. den Trick eine Matrix mit alternierenden Elementen[2], die abhängig von der Nachricht selbst sind, mit Hilfe des Hadamard-Produktes (auch Schur-Produkt genannt) und der Matrixaddition mit unabhängigen Matrizen zu kombinieren und so eine brauchbare Verschlüsselungsmatrix zu entwerfen(vgl.[JV]S.30):

$$S = \left(\begin{pmatrix} z_{1,1} & \cdots & z_{1,m} \\ \vdots & \ddots & \vdots \\ z_{m,1} & \cdots & z_{m,m} \end{pmatrix} \circ \begin{pmatrix} x_{1,1}(\vec{a}) & \cdots & x_{1,m}(\vec{a}) \\ \vdots & \ddots & \vdots \\ x_{m,1}(\vec{a}) & \cdots & x_{m,m}(\vec{a}) \end{pmatrix} \right) + \begin{pmatrix} c_{1,1} & \cdots & c_{1,m} \\ \vdots & \ddots & \vdots \\ c_{m,1} & \cdots & c_{m,m} \end{pmatrix}$$

$$z \in \mathbb{Z}; x_{i,l}(\vec{a}) = (-1)^{f_{i,l}(\vec{a})} \in [1,-1]; c \in \mathbb{Z}$$

Wählt man $f_{i,l}(\vec{a})$ passend, so sind die auftauchenden Vorzeichen nahezu unvorhersehbar. Der Vektor $\vec{a}$ besteht aus dem verschlüsselten Text selbst bzw. bei Verschlüsselung des ersten Blockes aus dem letzten Block der Nachricht. Vorteil ist, dass durch die Abhängigkeit der Matrix von der Nachricht selbst ein Lawineneffekt entsteht, so dass unter der Voraussetzung, die Funktionen sind schlau gewählt, bei kleinen Änderungen komplett neue Geheimtexte entstehen. Eine Angriffsfläche könnte allerdings bei dieser Methode sein, dass die abhängige $m \times m$ Matrix höchstens nur 2^{m*m} mögliche Matrizen enthält. Daher ist die erste Methode (zumindest anschaulich) praktikabler.

Eine Umsetzung dieser ersteren Idee könnte am Beispiel wie folgt aussehen:

$$M = \begin{pmatrix} a_1 + 3 & 5 & 2 \\ 5 & {a_2}^2 & a_3 \\ a_2 + 2 & 3 & 3 \end{pmatrix}; S = 254137421\,; Klartext\ T = 121314151$$

Der Vektor $\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ setzt sich aus dem Zusatzschlüssel S zusammen, der ebenfalls in Blöcke geteilt wird. Verschlüsselt wird also wie folgt:

$\begin{pmatrix} 2+3 & 5 & 2 \\ 5 & 5^2 & 4 \\ 5+2 & 3 & 3 \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 16 \\ 59 \\ 16 \end{pmatrix}$	$\begin{pmatrix} 1+3 & 5 & 2 \\ 5 & 3^2 & 7 \\ 3+2 & 3 & 3 \end{pmatrix} * \begin{pmatrix} 3 \\ 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 25 \\ 52 \\ 30 \end{pmatrix}$
$\begin{pmatrix} 4+3 & 5 & 2 \\ 5 & 2^2 & 1 \\ 2+2 & 3 & 3 \end{pmatrix} * \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 34 \\ 26 \\ 22 \end{pmatrix}$	$G = 16\ 59\ 16\ 25\ 52\ 30\ 34\ 26\ 22$

Zum Entschlüsseln berechnet man entweder die allgemeine Inverse Matrix, wie oben beschrieben, oder löst einzeln die Gleichungssysteme unter Verwendung des Zusatzschlüssels S ($\vec{a}$) und des Geheimtextes G ($\vec{G}$) nach dem Klartext($\vec{T}$) auf:

[2] Ein ähnliches Kernkonzept verwenden auch AES und DES und benutzen dabei aber formal Xor-Funktionen

$$\begin{pmatrix} a_1+3 & 5 & 2 \\ 5 & {a_2}^2 & a_3 \\ a_2+2 & 3 & 3 \end{pmatrix} * \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \end{pmatrix}$$

2.4.3 Problematik bei dem Verfahren

Für alle Verfahren, die auf Matrizen als Schlüssel beruhen, muss sichergestellt werden, dass die Determinante der Schlüsselmatrix ungleich Null ist, damit die Invertierbarkeit garantiert ist. Denn die Determinante gibt darüber Auskunft, um welchen Faktor sich das n-dimensionale Volumen einer n- dimensionalen Vektor-Menge bei Abbildung mit der Matrix, die in der Determinante steht, ändert. Als Beispiel stelle man sich einen Würfel im 3-dimensionalen Raum vor. Bildet man diesen durch eine Matrix ab, entsteht ein neues geometrisches Objekt z.B. ein Parallelepiped. Der Faktor, um den sich das Volumen geändert hat, ist die Determinante der Abbildungsmatrix. Ist diese Determinante nun gleich Null, so kann kein 3- dimensionales Volumen der Abbildung existieren. Am Würfelbeispiel wird der Würfel in diesem Fall durch die Matrix in eine Ebene oder sogar in einen Punkt abgebildet. Dadurch ist es unausweichlich, dass mehrere Punkte in einen einzigen Punkt abgebildet werden, und es kann keine eindeutige Umkehrabbildung existieren. Die Invertierbarkeit ist aber Voraussetzung für die Decodierung und von elementarer Bedeutung in der Kryptographie.

Was heißt das für das Beispiel?

Aus der geometrischen Überlegung zur Determinante kann man auch auf den Berechnungsweg im 3-Dimensionalen schließen. Hat man den Einheitswürfel mit dem Volumen V= 1VE im Ursprung gegeben, so werden die Kantenvektoren auf die Vektoren bestehend aus den Komponenten der Matrix abgebildet:

$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ d \\ g \end{pmatrix} = \vec{a}$	$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} * \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} b \\ e \\ h \end{pmatrix} = \vec{b}$
$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} c \\ f \\ i \end{pmatrix} = \vec{c}$	Notiz: man kann die Matrix prinzipiell auch wie folgt schreiben: $\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = (\vec{a} \quad \vec{b} \quad \vec{c})$

Es entsteht also ein Parallelepiped, dessen Volumen gleich der Determinante von der Abbildungsmatrix sein muss, da die Determinante den Volumenfaktor berechnet. Das Volumen, das von 3 Vektoren aufgespannt wird, lässt sich auch über das Spatprodukt berechnen. Also gilt die Gleichung:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = \vec{a} * \left(\vec{b} \times \vec{c}\right)$$

Wie man $\left(\vec{b} \times \vec{c}\right)$ berechnet, lässt sich mit Hilfe des Laplaceschen Entwickelungssatzes herleiten[3]:

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = a_1 * \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - a_2 * \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} + a_3 * \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix}$$

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = \vec{a} * \begin{pmatrix} \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} \\ \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} \end{pmatrix} \qquad \left(\vec{b} \times \vec{c}\right) = \begin{pmatrix} \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} \\ \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} \end{pmatrix}$$

(vgl. [GB] S.269) Diese oberen Untersuchungen kann man nun auf das Beispiel anwenden:

$$|M| = \begin{vmatrix} a_1 + 3 & 5 & 2 \\ 5 & {a_2}^2 & a_3 \\ a_2 + 2 & 3 & 3 \end{vmatrix} = \begin{pmatrix} a_1 + 3 \\ 5 \\ a_2 + 2 \end{pmatrix} * \left(\begin{pmatrix} 5 \\ {a_2}^2 \\ 3 \end{pmatrix} \times \begin{pmatrix} 2 \\ a_3 \\ 3 \end{pmatrix} \right)$$

$$|M| = \begin{pmatrix} a_1 + 3 \\ 5 \\ a_2 + 2 \end{pmatrix} * \begin{pmatrix} 3({a_2}^2 - a_3) \\ -9 \\ -2{a_2}^2 + 5a_3 \end{pmatrix}$$

$$|M| = 3({a_2}^2 - a_3) * (a_1 + 3) - 45 + (a_2 + 2) * (-2{a_2}^2 + 5a_3)$$

$$|M| = 3{a_2}^2 a_1 + 9{a_2}^2 - 3a_3 a_1 - 9a_3 - 45 - 2{a_2}^3 + 5a_3 a_2 - 4{a_2}^2 + 10a_3$$

$$|M| = 3{a_2}^2 a_1 + 5{a_2}^2 - 3a_3 a_1 + a_3 - 45 - 2{a_2}^3 + 5a_3 a_2$$

Wenn folgende Beziehung gilt, funktioniert das Verfahren folglich nicht:

$$3{a_2}^2 a_1 + 5{a_2}^2 - 3a_3 a_1 + a_3 - 45 - 2{a_2}^3 + 5a_3 a_2 = 0$$

Dieses Problem kann man z.B. durch Testen des Zusatzschlüssels direkt bei der Konstruktion prüfen, was bei zunehmender Größe der Matrix aber mit enormen Rechenaufwand verbunden ist, oder aber man staffelt die Blockung anders auf und nimmt in Kauf, dass einige Informationen beim Verschlüsseln verloren gehen, auch wenn das Prinzip der Integrität dabei verletzt wird. Diese müssen dann nachträglich durch statistische Verfahren (z.B. Maximum Likelihood Methode) ‘‘gerettet‘‘ werden. Elegant könnte die Matrix auch so konstruiert werden, dass es keine Tupel ganzer Zahlen gibt, bei denen die Determinante der Matrix Null ist.

[3] theoretisch kann man auf die selbe Weise darüber auch ein 4d oder nd-Kreuzprodukt definieren (siehe Anhang), das in z.B der arT Anwendung finden kann.

Ähnliche ‘‘Stolpersteine‘‘ (dass in gewissen Fällen das Verfahren nicht funktioniert oder dass ein Sicherheitsproblem durch die Wahl von Schlüssel bzw. bestimmten Parametern besteht) findet man in der Kryptografie immer wieder.

3 Maximum Likelihood Decodierung

Übertragungsfehler treten beim Sendevorgang häufig auf. Meistens sind sie direkt durch Störungen in der Leitung bedingt, seltener (wie bei dem Beispiel der variablen Matrizen) durch die Verschlüsselung selbst. Der Information, die ursprünglich vorhanden war, wird also ein Rauschen hinzugefügt. Um dem entgegenzuwirken wird häufig die Maximum Likelihood Decodierung als die in der Kodierungstheorie relevanteste Störungsbehebungsmethode eingesetzt. Diese funktioniert derart, dass ein Wort, das fehlerhaft empfangen wurde (Existenz des Wortes ist nicht erkennbar), mit plausiblen Wörtern z.B. über den Hammingabstand verglichen wird. Das fehlerhafte Wort wird durch das Wort ersetzt, dessen Wahrscheinlichkeit, dass es passt, am höchsten ist. Häufig verwendetes Kriterium hierfür ist die Wahl des Wortes mit dem kleinsten Hammingabstand. Allgemein liegt diese Dekodierung aber der Maximum Likelihood Methode zu Grunde:

3.1 Maximum Likelihood Methode

Die Maximum-Likelihood-Methode ist eine Methode der Statistik um Parameter wie z.B. den Erwartungswert oder die Standartabweichung einer Messreihe (in diesem Fall der gesendete Code) zu schätzen, indem man eine dem Problem entsprechende Schätzfunktion aufstellt und die Extrema bestimmt. Sie kann also neben der Kryptografie in verschiedensten Bereichen bezüglich Auswertung von Informationen eingesetzt werden. Die Methode wurde von dem australischem Statistiker Ronald Aylmer Fisher (1890-1962) nach langer Zeit erstmalig wiederentdeckt, tauchte aber schon in Schriften des Carl Friederich Gauß (1777-1855) auf.

3.1.1 Beispiel mit diskreter Wahrscheinlichkeitsverteilung (Bernoulli-Experiment)

Die Seiten eines ungezinkten Würfels sind entweder schwarz oder weiß. Nachdem 42 mal gewürfelt wurde lag schwarz 5 mal oben und weiß 37 mal oben. Wie viele Seiten sind mit der größten Wahrscheinlichkeit schwarz gefärbt?

Bei diesem Zufallsexperiment ist die Zufallsvariable X „Auftreten von Schwarz“ Bernoulli-verteilt.

Angenommen die Wahrscheinlichkeit für Schwarz wäre 1/3, dann wäre die Wahrscheinlichkeit, dass sich diese Werte ergeben für eine bestimmte Permutation[4]:

$$\left(\frac{1}{3}\right)^{5} * \left(\frac{2}{3}\right)^{37} \approx 1{,}256 * 10^{-9}$$

Nimmt man 1/6 für die Wahrscheinlichkeit von schwarz an ergibt sich:

[4] Der Binomialkoeffizient muss an dieser Stelle nicht mit einbezogen werden, da er sich rauskürzt.

$$\left(\frac{1}{6}\right)^5 * \left(\frac{5}{6}\right)^{37} \approx 1{,}511 * 10^{-7}$$

Damit ist es wahrscheinlicher, dass die Wahrscheinlichkeit für schwarz 1/6 beträgt als 1/3. Also ist es wahrscheinlicher, dass genau eine Seite des Würfels schwarz ist, und nicht genau zwei Seiten schwarz sind. p = 1/6 ist somit mutmaßlicher als p=1/3.

Zur Bestimmung des mutmaßlichsten Wertes für p stellt man eine sogenannte Likelihoodfunktion auf und bestimmt anschließend den Hochpunkt z.B. durch Ableiten und Nullsetzen.

Man hat X1,X2,X3,..Xn voneinander unabhängige Stichproben einer Zufallsvariablen x (z.B. in Form eines empfangenen Codes). Um nun die gesamt Wahrscheinlichkeit zu erhalten, muss man die Wahrscheinlichkeit für den Ausgang des Ergebnisses der ersten Stichprobe mit der Wahrscheinlichkeit für den Ausgang des Ergebnisses der zweiten Stichprobe… mit der Wahrscheinlichkeit für den Ausgang des Ergebnisses der n-ten Stichprobe multiplizieren. Die Wahrscheinlichkeit für den Ausgang eines einzelnen Ereignisses sei nun $f(x, \theta)$. Also gilt für die gesamt Wahrscheinlichkeit:

$$f(x_1, \dots, x_n) = f(x_1, \theta) * f(x_2, \theta) * \dots * f(x_n, \theta) = \prod_{i=1}^{n} f(x_i, \theta)$$

x_i sind hierbei die zufälligen Ausgänge der Stichproben X1,X2,X3,… ,Xn. Für unsere Schätzung müssen wir sie aber als fix ansehen, um ein Maximum beim variablen Parameter θ finden zu können. (vgl. [PP] S.8)

Bei unserm Beispiel ist θ unsere Wahrscheinlichkeit für schwarz. In unserer Likelihoodfunktion kommt also als direkter Faktor θ^5 vor, da im gesamten Experiment fünfmal eine schwarze Seite oben lag. Um aber alle Ereignisse zu berücksichtigen muss man die Nicht-Treffer auch berücksichtigen. Die Wahrscheinlichkeit, dass weiß oben liegt, ist die Gegenwahrscheinlichkeit(1-θ). Also ist ein weiterer Faktor (1-θ)^37, da 37mal weiß oben liegt. Die gesamte Likelihoodfunktion lautet somit: $\mathcal{L}(\theta) = \theta^5 * (1-\theta)^{37}$

Nun könnte man ganz normal mit Hilfe der Produktregel nach θ ableiten und durch Nullsetzung der Ableitung das Maximum herausfinden. Oftmals gestaltet es sich aber als weitaus einfacher den Logarithmus der Funktion zu bilden und davon die Extrema zu bestimmen, da diese sogenannte Loglikelihoodfunktion auf Grund von Logarithmengesetzen zu einer leicht differenzierbaren („Summen“-)Funktion umgeformt werden kann. Dies funktioniert, da die ln-Funktion streng monoton ist und somit an den selben Stellen die Extrema besitzt. (bei komplexeren Beispielen ist dies noch wichtiger):

$$ln(\mathcal{L}(\theta)) = ln(\theta^5 * (1-\theta)^{37})$$

$$ln\big(\mathcal{L}(\theta)\big) = 5 * ln(\theta) + 37 * ln(1-\theta)$$

$$\frac{\partial \ln \mathcal{L}}{\partial \theta} = \frac{5}{\theta} + 37 * \frac{1}{1-\theta} * (-1)$$

$$\frac{\partial \ln \mathcal{L}}{\partial \theta} = \frac{5}{\theta} - \frac{37}{1-\theta}$$

$$\frac{5}{\theta} - \frac{37}{1-\theta} \stackrel{!}{=} 0$$

$$\theta = \frac{5}{42} = \frac{x}{n} \approx 0{,}119047 \approx \frac{1}{6}$$

Da 0,119047 am nächsten an 1/6 liegt ist es am plausibelsten, wenn die Wahrscheinlichkeit, dass schwarz oben liegt, 1/6 ist. Also ist es am wahrscheinlichsten, dass genau eine Seite schwarz ist.[5] (vgl. [PP] S.8)

3.1.2 Beispiel mit stetiger Wahrscheinlichkeitsverteilung

Ein komplexeres Beispiel hierfür, dem eine stetige Wahrscheinlichkeitsverteilung[6] zu Grunde liegt, ist folgendes:

Bsp.: Es gibt einen Bestand von n nicht alternden Tieren. Sie sind nicht unsterblich, können also nach einer Zeit ti überfahren werden. Die Exponentialverteilung ist definiert durch $f_\lambda(x) = \lambda e^{-\lambda x}$ Wie findet man den ML-Schätzer für λ, wenn eine Messreihe der Lebensdauer dieser Tiere gegeben ist?

$$\mathcal{L}(t_1, \dots, t_n; \theta) = \prod_{i=1}^{n} \theta e^{-\theta \times t_i}$$

$$\mathcal{L}(t_1, \dots, t_n; \theta) = \theta e^{-\theta \times t_1} * \theta e^{-\theta \times t_2} * \dots * \theta e^{-\theta \times t_n}$$

$$\mathcal{L}(t_1, \dots, t_n; \theta) = \theta^n * e^{-\theta * (t_1 + t_2 + \dots + t_n)}$$

$$\mathcal{L}(t_1, \dots, t_n; \theta) = \theta^n * e^{-\theta * \sum_{i=1}^{n} t_i}$$

Logarithmieren:

$$\ln\big(\mathcal{L}(t_1, \dots, t_n; \theta)\big) = \ln(\theta^n * e^{-\theta * \sum_{i=1}^{n} t_i})$$

$$\ln\big(\mathcal{L}(t_1, \dots, t_n; \theta)\big) = n * \ln(\theta) + \ln(e^{-\theta * \sum_{i=1}^{n} t_i})$$

$$\ln\big(\mathcal{L}(t_1, \dots, t_n; \theta)\big) = n * \ln(\theta) - \theta * \sum_{i=1}^{n} t_i$$

[5] Auf den Beweis das die entsprechenden Likelihood-Funktionen der Beispiele genau ein Extremum in Form eines Hochpunktes haben wird an dieser Stelle verzichtet.

[6] Nach dem vorgeführten Weg kann man beispielsweise auch den Erwartungswert μ oder die Varianz σ² bei normalverteilten Proben mit Hilfe der Gaußschen-Glockenkurve schätzen und die bekannten Formeln dafür herleiten. (siehe Anhang)

Nach θ Ableiten und Null setzen:

$$\frac{\partial \ \ln\left(\mathcal{L}(t_1, \dots, t_n; \theta)\right)}{\partial \ \theta} = \frac{n}{\theta} - \sum_{i=1}^{n} t_i$$

$$\frac{n}{\theta} - \sum_{i=1}^{n} t_i \quad \stackrel{!}{=} \quad 0$$

$$\boldsymbol{\theta = \frac{n}{\sum_{i=1}^{n} t_i}}$$

4 Asymmetrische Verschlüsselung

Bei verschiedenen Verschlüsselungsmethoden der Vergangenheit gab es immer ein großes Problem: der Empfänger und der Sender brauchen den selben Schlüssel, um eine Nachricht zu verschlüsseln und zu entschlüsseln. So gab es für Angreifer immer eine Chance, die Schlüssel beim Schlüsselaustausch abzufangen und somit den gesamten Datenverkehr abzufangen und zu entschlüsseln. Außerdem müssen viele verschiedene Schlüssel generiert werden, und die Verwaltung aller Schlüssel ist sehr aufwendig (mittlerweile nicht mehr). Als eine der ersten Möglichkeiten dies zu umgehen, wurde zu Beginn der 70er Jahre ein Verfahren entwickelt und in abgewandelter Form 1977 als RSA-Verfahren von Rivest, Shamir und Adleman praxistauglich gemacht. Zuvor ist es schon in England entwickelt worden (ca.1970). Es wurde aber vom Staat geheim gehalten. Das Prinzip ist bei den meisten asymmetrischen Verfahren bis heute das selbe. Statt einen Schlüssel für Sender und Empfänger zu haben, gibt es zwei verschiedene Schlüssel: einen öffentlichen, der jedem zugänglich ist (auch potenziellen Angreifern) und einen privaten (den man ganz allein für sich behält, und den niemand -außer einem selbst- kennt). Der Trick dabei ist, dass man mit dem öffentlichen Schlüssel zwar die Nachricht verschlüsseln kann, diese jedoch nicht entschlüsseln kann. Man kann die Nachricht ausschließlich nur mit dem privaten Schlüssel entschlüsseln. Veranschaulicht kann man sich das so vorstellen: Wenn Alice Bob eine Nachricht schicken will, stellt sie Bob eine Anfrage. Bob gibt ihr daraufhin eine Box und ein offenes Schnappschloss, mit dem man diese Box verriegeln kann. Identisches Material gibt Bob jedem, der ihm eine Nachricht schicken will. Den Schlüssel für all diese Schlösser behält Bob. Nun schreibt Alice ihre Nachricht auf einen Zettel, legt ihn in die Box und verriegelt sie mit dem Schloss, das Bob ihr gegeben hat. Nun ist es ihr nicht mehr möglich die Box zu öffnen. Wenn sie die Box nun an Bob sendet, kann er diese aber mit seinem privaten Schlüssel öffnen. Ein Angreifer kann nach diesem Prinzip sowohl die Kiste mit Schloss abfangen als auch die geschlossene Kiste mit Nachricht. Wenn er die Nachricht lesen will, ist er in jedem Fall dazu gezwungen das System zu knacken ohne zuvor die Chance gehabt zu haben einen Schlüssel abzufangen. Bei den meisten Verfahren hat der Angreifer ab einer gewissen Schlüssellänge so gut wie keine Chance in absehbarer Zeit den Code zu knacken ohne den Schlüssel zu kennen. (vgl. [CM] S.101, [DA] S.491)

4.1 RSA Verfahren

Um das wohl bekannteste asymmetrische Verschlüsselungsverfahren, das zu den Pionieren der modernen Geheimchiffrierung gezählt wird, zu verstehen, benötigt man einige theoretische Grundlagen:

4.1.1 Der Satz von Euler

Der Satz von Euler besagt, dass, wenn zwei Zahlen a und m teilerfremd sind, gilt, dass a hoch die Anzahl der zu m teilerfremden Zahlen kongruent zu 1 modulo m ist, also:

$$ggT(a,m) = 1 \rightarrow \qquad a^{\varphi(m)} \equiv 1 \; mod \; m$$

Diesen Satz kann man wie folgt beweisen:

Es gelte: $ggT(a,m) = 1$

Alle zu m teilerfremden Zahlen werden so bezeichnet: $\mathbb{Z}_m$: k_1, k_2, k_3,..., $k_{\varphi(m)}$

Man wählt nun eine ebenfalls teilerfremde Zahl a und multipliziert diese mit den k_i:

$\mathbb{Z}_a$: ak_1, ak_2, ak_3,..., $ak_{\varphi(m)}$

a ist teilerfremd zu m, alle k sind teilerfremd zu m, also sind auch alle ak teilerfremd.

$$ggT(ak_i, m) = 1$$

Bei Multiplikation mit a ergeben sich die selbe Menge wie die Menge $\mathbb{Z}_m$. Vergleicht man das entstandene Produkt ak_i mit dem vorher bestehenden k_i so weiß man also, dass es sich nicht um identische Zahlen handelt, sondern um verschiedene Permutationen.

Zu vermuten ist also, dass es sich nicht zweimal um die gleiche Zahl innerhalb aller ak_ihandeln kann. Dann müsste gelten:

$$ak_i \neq ak_j \; mod \; m$$

Diese Vermutung kann man wie folgt beweisen:

Man nimmt an, die Beziehung, die ausgeschlossen wurde, stimme. Also gilt:

$$ak_i \equiv ak_j \; mod \; m$$

Man kann auf beiden Seiten durch a teilen, denn a und m sind teilerfremd.

Also:

$$k_i \equiv k_j \; mod \; m$$

Das ist ein Widerspruch, da die beiden k nicht kongruent sein können.

Da man nun weiß, die Mengen $\mathbb{Z}_m$ und $\mathbb{Z}_a$ sind identische endliche Mengen, müssen die Produkte über alle Elemente der Mengen ebenfalls identisch sein.

Daher gilt:

$\prod_{i=1}^{\varphi(m)} k_i = \prod_{i=1}^{\varphi(m)} a * k_i$ bzw.

$k_1 * k_2 * k_3 * \ldots * k_{\varphi(m)} \equiv ak_1 * ak_2 * ak_3 * \ldots * ak_{\varphi(m)} \mod m$

Alle ki sind teilerfremd zu m, also kann man durch alle ki teilen und es ergibt sich:

$$1 \equiv a^{\varphi(m)} \mod m$$

Was zu beweisen war. (vgl. [MR]S.16)

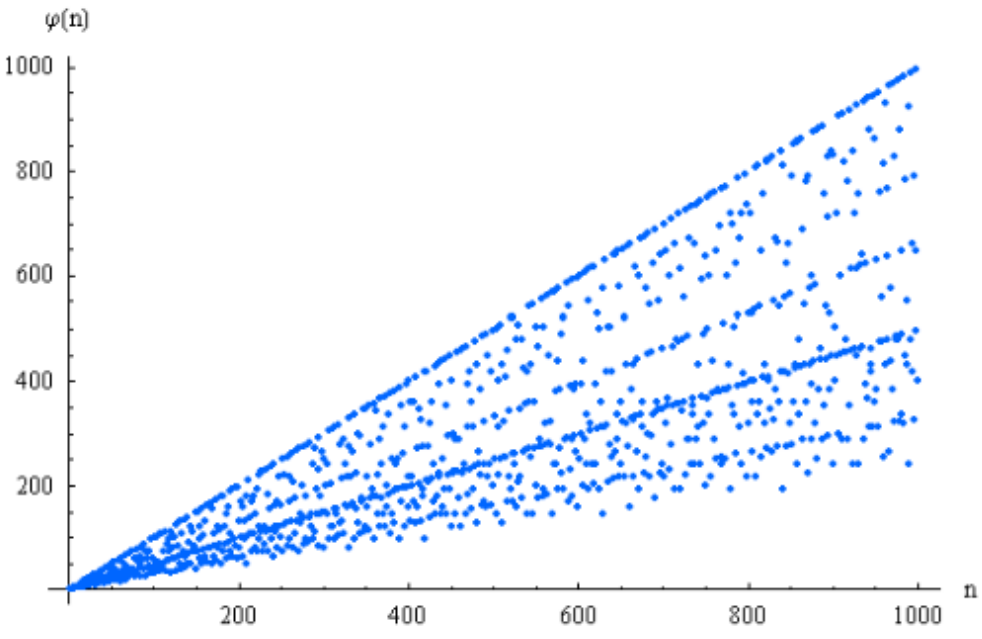
4.1.2 Eulersche phi-Funktion

Die Eulersche Phi-Funktion ist eine zahlentheoretische Funktion, die die Menge der teilerfremden Zahlen zu einer Zahl angibt (z.B. $\varphi(5) = 4$). In der Konsequenz folgt daraus, dass folgende Beziehung gilt: Sei p prim so gilt: $\varphi(p) = (p - 1)$. Dies ist direkt aus der Definition der Primzahlen (eine natürliche Zahl heißt prim, wenn sie genau zwei Teiler hat) zu schlussfolgern. Um die phi-Funktion nun mathematisch korrekt zu definieren, definiert man sie über die Mächtigkeit der Menge der natürlichen Zahlen, die sich zwischen 1 und dem variablen Wert befinden (oder gleich dieser sind) und für die gilt, dass sie teilerfremd zur Variablen sind:

$$\varphi(n) = |\{z \in \mathbb{N} | 1 \leq z \leq n \wedge ggt(z, n) = 1\}|$$

(vgl.[ZTM] S.36, [MR] S.15) Geplottet ergibt sich folgende Grafik[7]:

Abb. 4



[7] Abb 4 [1] Quelle: http://cs.uni-muenster.de/u/lammers/EDU/ss09/DiskreteStrukturen/Script/Kap5%20-%20Zahlentheorie%20+%20Arithmetik.mm.html

Die phi-Funktion ist eine multiplikative Funktion. Es gilt also : $\varphi(p * q) = \varphi(p) * \varphi(q)$. Sind p und q Primzahlen gilt also $\varphi(p * q) = (p - 1) * (q - 1)$.

4.1.3 Euklidischer Algorithmus

Eine weitere wichtige ‘‘Zutat‘‘ ist der einfache und der erweiterte euklidische Algorithmus. Der einfache euklidische Algorithmus liefert bei Anwendung auf ein gegebenes Zahlenpaar (a, b)den $ggT(a, b)$. Um dies zu erreichen dividiert man mit Rest die größere der beiden Zahlen durch die kleinere. Die kleinere Zahl dividiert man mit Rest anschließend durch den Rest aus dem ersten Schritt. Dies führt man fort bis der Rest gleich Null ist. Der $ggT(a, b)$ stimmt nun mit dem Rest der vorhergegangenen Rechnung überein. (vgl[IB] S.4) Am Beispiel:

$$a = 126;\ b = 48$$

$$126 = 2 * 48 + 30$$

$$48 = 1 * 30 + 18$$

$$30 = 1 * 18 + 12$$

$$18 = 1 * 12 + 6$$

$$12 = 6 * 2 + 0$$

$$\Rightarrow ggT(126, 48) = 6$$

Für den späteren Verwendungszweck benötigt man nun noch eine ‘‘Erweiterung‘‘, die der sog. erweiterte euklidische Algorithmus liefert. Dieser löst lineare diophantische Gleichungen der Form

$$ggT(a, b) = a * x + b * y \qquad a, b \in \mathbb{N}; x, y \in \mathbb{Z}$$

Nach oberen Beispiel also:

$$6 = 126 * x + 48 * y$$

Um x und y herauszufinden werden die in den Schritten des gewöhnlichen Euklidischen Algorithmus gemachten Gleichungen jeweils in die vorhergehende eingesetzt. Man “arbeitet“ sich also von der untersten Gleichung nach oben ‘‘rückwärts‘‘ vor:

$$6 = 18 - 1 * (30 - 1 * 18)$$

$$6 = 18 - 1 * 30 + 1 * 18$$

$$6 = -1 * 30 + 2 * 18$$

$$6 = -1 * 30 + 2 * (48 - 1 * 30)$$

$$6 = -1 * 30 + 2 * 48 - 2 * 30$$

$$6 = 2 * 48 - 3 * 30$$

$$6 = 2 * 48 - 3 * (126 - 2 * 48)$$

$$6 = 2 * 48 - 3 * 126 + 6 * 48$$

$$6 = -3 * 126 + 8 * 48$$

$$\Rightarrow x = -3, y = 8$$

Routiniert macht man diese vergleichsweise aufwändigen Schritte in tabellarischer Form(vgl. [ZTM] S.28, [MR] S.4):

		a/b	Rest		
a	b	q	r	x	Y
126	48	2	30	-3	8
48	30	1	18	2	-3
30	18	1	12	-1	2
18	12	1	6	1	-1
12	6	2	0	0	1

4.1.4 Aufbau des Kryptosystems

Wie ECC-Verfahren beispielsweise auf dem diskreten Logarithmusproblem basieren, baut RSA auf dem Problem auf, große Zahlen in Primfaktoren zu zerlegen. Daher beginnt der erste Schritt damit, eine Zahl zu konstruieren, bei der eine solche komplizierte Faktorzerlegung gegeben ist:

1. Man erstellt ein RSA-modul N aus zwei extrem großen Primzahlen p und q:

$$N = p * q$$

Hat man nur das N gegeben, ist es (vermutlich) extrem schwer p und q zu ermitteln.

2. Nun bildet man $\varphi(N) = \ \varphi(p * q) = \varphi(p) * \varphi(q) = (p - 1) * (q - 1)$

3. Anschließend wählt man ein e < $\varphi(N)$ und e > 1 , $e \in \mathbb{N}$ mit ggT(e, $\varphi(N)$)=1.

e und N bilden den öffentlichen Schlüssel. Geheim bleiben dagegen p,q und $\varphi(N)$.

Anschließend muss nur noch der private Schlüssel gebildet werden: Es gilt ggT(e, $\varphi(N)$)=1. Daher existiert ein multiplikatives modulares Inverses d von e mit $\varphi(N)$ als Modul. d wird also mit Hilfe der Beziehung $e * d \equiv 1 \ mod \ \varphi(N)$ als modulares Inverses zu e bezüglich $\varphi(N)$ berechnet. Diese Kongruenz kann man in eine Gleichung überführen und folgende Behauptung schlussfolgern:

$$\exists k \in \mathbb{Z}: \ \varphi(N) * k = e * d - 1$$

Also ergibt sich $e * d - k * \varphi(N) = 1$. Das Minus vor dem unbekannten Faktor kann als Bestandteil des Faktors aufgefasst werden, daher ergibt sich folgende lineare diophantische Gleichung:

$$e * d + v * \varphi(N) = 1$$

Auf diese Gleichung kann man umstandslos den erweiterten euklidischen Algorithmus anwenden, da sowohl e als auch $\varphi(N)$ bei der Aufstellung des Kryptosystems gegeben sind, und die Zahlen so gewählt wurden, dass der ggT als 1 festgelegt ist. v ist ein überflüssiges Ergebnis. d ist nun als multiplikatives modulares Inverses von e zu $\varphi(N)$ zusammen mit N der öffentliche Schlüssel.

Ver- und Entschlüsseln:

Um einen Text T zu einem Geheimtext G zu verschlüsseln rechnet man $G = T^e mod\ N$

Um ihn wieder zu entschlüsseln berechnet man $T = G^d mod\ N$

Die Begründung, warum diese Vorgehensweise funktioniert, wird offensichtlich, wenn man sich anschaut, welche Schritte getätigt wurden, und sich klar macht, was hinter jedem kryptographischen Vorgang steckt. Ziel ist es, die Veränderung (Unkenntlichmachung) einer Information zu erzeugen und rückgängig zu machen. Aus diesem Grund wurde das modulare Inverse gebildet. Wenn man sich nun klar macht, was ansonsten verwendet wurde, und den Satz von Euler ausnutzt, zeigt sich die Plausibilität des Kryptosystems. Es wurde so konstruiert dass:

$$e * d \equiv 1\ mod\ \varphi(N)$$

$$\rightarrow e * d = k * \varphi(N) + 1$$

Vor diesem Hintergrund lässt sich nun entscheidend die Richtigkeit zeigen:

$$T^{e^d} \equiv T^{e*d} \equiv T^{k*\varphi(N)+1} \equiv T^{k*\varphi(N)} * T \equiv T^{\varphi(N)^k} * T \equiv 1^k * T \equiv T\ mod\ N$$

In oberer Zeile wird der Satz von Euler so eingesetzt, dass folgender Fall betrachtet wird:

$$ggt(T, N) = 1 \rightarrow \qquad T^{\varphi(N)} \equiv 1\ mod\ m$$

Also muss T teilerfremd zu N sein, denn ansonsten dürfte man den Satz von Euler in dem obigen Konstrukt nicht ausnutzen, und das Kryptosystem würde nicht in jedem Fall funktionieren.

Um dem Problem zu begegnen, muss man sich in der Praxis auf eine maximale Größe der zu verschlüsselnden Zahlen einigen, die kleiner ist als beide Primzahlen, in dem man eine Blockung einführt. Dies ist dadurch zu begründen, dass N Produkt von genau zwei Primzahlen ist. Wenn man den zu verschlüsselnden Text T kleiner wählt als die beiden Primzahlen, dann ist T auch teilerfremd zu N. Alternativ kann man als maximale Größe auch N-1 zulassen, ist dann aber mit Störungen in der Nachricht, die der Empfänger erhält

konfrontiert. In diesem Fall müsste man ebenfalls durch geeignete Blockung der Nachricht und statistischen Verfahren die Nachricht ‘‘retten‘‘. Der Vorteil dabei ist, dass die ungefähre Größe der Primzahlen geheim gehalten wird, und Angreifer die Größe der ‘‘Nachrichtengrenzlänge‘‘ nicht ausnutzen können. (vgl. [MR] S.29, [IB] S.30, [HC]S. 286)

4.1.5 ein Beispiel

Am Beispiel kann man sich den Ablauf verdeutlichen und Hindernisse erkennen, die bei einer Implementation zusätzlich berücksichtigt werden müssen:

p=7; q=11, N=77, $\varphi(N) = 60$, e = 47

Bildung des Inversen ergibt:

$47 * d \equiv 1 \; mod \; 60$

→$47 * d + v * 60 = 1$

Wendet man den erweiterten Euklidischen Algorithmus an ergibt sich:

$$47 * x + 60 * y = 1$$

		a/b	Rest		
a	b	q	R	x	Y
47	60	0	47	23	-18
60	47	1	13	-18	23
47	13	3	8	5	-18
13	8	1	5	-3	5
8	5	1	3	2	-3
5	3	1	2	-1	2
3	2	1	1	1	-1
2	1	2	0	0	1

➔ Also ist das gesuchte modulare Inverse d=23.

Der private Schlüssel ist nun (23,60).

Nur Zahlen, die kleiner 7 sind, sind zur Verschlüsselung zugelassen. Als Beispieltext wird T=5 gewählt.

Verschlüsseln:

$$G = T^e mod \; N \Rightarrow G = 5^{47} mod \; 77 = 3$$

Entschlüsseln:

$$T = G^d mod \; N \Rightarrow T = 3^{23} mod \; 77 = 5$$

Ein Hindernis, das nicht durch konventionelles Rechnen lösbar scheint, ist das modulare Potenzieren, das beim Ver- und Entschlüsseln gebraucht wird. Den Weg die Potenz direkt auszurechnen und anschließend zu dividieren, kann man aber durch ‘‘geschickte‘‘ Umformungen des Ausdrucks umgehen.

4.2 Das „diskreter Logarithmus“ Problem

Jede gute Verschlüsselung basiert auf einem Problem. Um etwas sicher verschlüsseln zu können, braucht man ein besonders extrem schwer zu lösendes mathematisches Problem. Ein sogenanntes NP-vollständiges Problem. Um ein solches Problem zu konstruieren modelliert man sich oft eine sogenannte Falltürfunktion, also eine Funktion, die an sich einfach zu berechnen ist, aber nur schwer umkehrbar ist. Ein Beispiel für ein solches Problem basierend auf einer Falltürfunktion ist das Problem des diskreten Logarithmus.

Im Gegensatz zum normalen Logarithmus über den positiven reellen Zahlen wird der diskrete Logarithmus über eine zyklische Gruppe definiert. Die Ausgangsgleichung ist: $a^x \equiv m \mod p$. Der diskrete Logarithmus fragt nun nach dem x im Exponenten. „Wie hoch muss man a mindestens nehmen, um den Rest m bei Division durch p zu bekommen.“ Die Umkehrfunktion (diskrete Exponentialfunktion) stellt die umgekehrte Frage: “Was ist der Rest (das Ergebnis), den man bekommt, wenn man a^x durch (modulo) p rechnet“. Da sich die diskrete Exponentialfunktion offensichtlich leicht berechnen lässt, die Umkehrfunktion (diskreter Logarithmus) aber (vermutlich) nur sehr schwer, liegt hier eine der oben beschriebenen Falltürfunktionen vor, auf dessen Basis sich Kryptosysteme aufbauen lassen. Das Problem über zyklische Gruppen zu logarithmieren wird diskretes Logarithmusproblem genannt. Dieses Problem macht sich z.B. der Deffie Hellman Schlüsselaustausch zu Nutze. (vgl. [DL] S.3)

4.3 Deffie Hellman Schlüsselaustausch

Um zunächst eine zyklische Gruppe zu erzeugen, einigen sich Alice und Bob auf eine Primzahl p, die als Modul fungiert. Außerdem einigt man sich auf eine Primitivwurzel g von p oder auf eine Zahl kleiner p die keine Primitivwurzel ist. Diese würde die Sicherheit allerdings mindern.

Definition 4.3.1

Eine natürliche Zahl z heißt Primitivwurzel von p, wenn folgende Bedingung erfüllt ist:

$$g^i \mod p : \{0,1,2,3, \dots, p-2\} \leftrightarrow \{1,2,3, \dots, p-1\}$$

Die Abbildung ist sowohl linkseindeutig (injektiv) als auch rechtstotal (surjektiv) und damit bijektiv. z.B. ist 2 Primitivwurzel von 5 denn:

$$2^0 \ mod \ 5 = 1$$

$$2^1 \ mod \ 5 = 2$$

$$2^2 \ mod \ 5 = 4$$

$$2^3 \ mod \ 5 = 3$$

P und g sind nun öffentlich einsehbar. Nun denkt sich sowohl Alice als auch Bob eine natürliche Zahl (a,b), die kleiner als die Primzahl ist und idealerweise nicht der Primitivwurzel entspricht. Diese bleiben geheim. Alice kennt also Bobs Zahl(b) nicht, und Bob kennt Alice Zahl(a) nicht.

An ein und denselben Schlüssel, den ein Außenstehender, der den Austausch abhört, nicht gelangen kann, kommen sie durch ein bestimmtes Vorgehen:

1. Alice berechnet $X_A = g^a \ mod \ p$, Bob berechnet $X_B = g^b \ mod \ p$
2. Nun schickt Alice X_A an Bob und Bob schickt X_B an Alice
3. Schließlich berechnet Alice den gemeinsamen Schlüssel $S = {X_B}^a mod \ p$ ebenso wie Bob $S = {X_A}^b mod \ p$

Dass dieses Verfahren funktioniert, zeigt sich folgendermaßen:

$$S = {X_B}^a mod \ p = g^{b^a} mod \ p = g^{ba} mod \ p = g^{a^b} mod \ p = {X_A}^b mod \ p$$

Ein Abhörer würde nur g, p, $X_A = g^a \ mod \ p$, $X_B = g^b \ mod \ p$ kennen und steht nun vor dem Problem durch diskretes Logarithmieren entweder an a oder an b zu gelangen. Er muss also das diskrete Logarithmus Problem, zu dem es (vermutlich) kein effektives Lösungsverfahren gibt, lösen. (vgl. [IB]S.55, [MS] S.50, [WD], [DL]S.5,S.6)

4.4 Elgamal-Verschlüsselung

Die Idee des Deffie Hellmann Schlüsselaustausch kann man nun weiterführen, so dass man ein asymmetrisches Kryptosystem erhält, mit dem man auch ganze Nachrichten codiert versenden kann.

Zunächst einigen sich Alice und Bob erneut auf eine zyklische Gruppe durch Wahl einer Primzahl und auf eine Primitivwurzel g von p. Nun denken sich beide erneut Zahlen (a,b) unter denselben Bedingungen wie beim Deffie Hellman Schlüsselaustausch. Außerdem berechnet Bob $X_B = g^b \ mod \ p$

Den öffentlichen Schlüssel berechnet Alice durch $X_A = g^a \ mod \ p$. a ist ihr privater Schlüssel.

Bob verschlüsselt nun einen Klartext T zum Geheimtext G indem er $G = {X_A}^b * T \ mod \ p$ berechnet. G wird zusammen mit X_B an Alice geschickt, die durch $T = {X_B}^{-a} * G \ mod \ p$

entschlüsselt. Dass das Verfahren funktioniert, lässt sich durch folgende Umformungen zeigen:

$$T \equiv {X_B}^{-a} * G \equiv {X_B}^{-a} * {X_A}^{b} * T \equiv g^{b^{-a}} * g^{ab} * T \equiv g^{ab} * g^{-ab} * T \equiv 1 * T \equiv T \; mod \; p$$

Ein Abhörer steht nun vor demselben Problem wie beim Schlüsselaustausch.

Eine Möglichkeit für einen Angreifer ist bei beiden kryptographischen Methoden durch einen „Man in the Middle“ Angriff gegeben. Das bedeutet: zwischen den Teilnehmern Alice und Bob befindet sich ein Angreifer Mallory, so dass Alice denkt, Mallory sei Bob, und Bob denkt, Mallory sei Alice. Der Angreifer ist nun in der Lage sowohl mit Alice als auch mit Bob ein Kryptosystem aufzubauen. Jede Nachricht, die einer der beiden Teilnehmer zum jeweils anderen senden will, kommt in Folge dessen zuerst bei Mallory an. Dieser kann die Nachricht natürlich lesen und nun entscheiden, ob die Nachricht unverändert weitergesendet, verändert weitergesendet oder eine gefälschte Antwort zurück an den Absender gesendet werden soll. (vgl. [MS] S.51, [IB] S.56)

4.5 Elliptische Kurven

Methoden, wie der Deffie Hellman Schlüsselaustausch, hatten mit zunehmender Digitalisierung und weiterentwickelter Rechenleistung der Computer ein Problem. Denn die Verfahren auf Basis des diskreten Logarithmusproblems sind zwar weiterhin mit einfachen Berechnungen ungeknackt, aber Bruce Force Methoden erlauben durch gezieltes und systematisches Ausprobieren das Herausfinden des Schlüssels nach einer gewissen Zeit. Um ausreichend Sicherheit zu gewährleisten, ist es nötig, riesige Primzahlen zu wählen und die anderen Parameter ebenfalls entsprechend groß zu wählen. Die Berechnung des Schlüssels wird damit für die Teilnehmer rechenaufwendig und speicheraufwendig. Diesem Effizienzproblem schafft die Kryptographie auf elliptischen Kurven Abhilfe. Mit ihr konnten sowohl neue Verschlüsselungsverfahren geschaffen als auch alte verbessert werden.

4.5.1 Definition über den reellen Zahlen

Folgendes Polynom[8] ist gegeben:

$$F(x,y) \coloneqq x^3 - a_1xy + a_2x^2 - a_3y + a_4x + a_5 - y^2$$

Eine Punktemenge gilt dann als elliptische Kurve E(K) über einem Körper K, wenn die Menge aller Punkte(x,y), mit x,y ∈ K, die Gleichung F(x,y)=0 erfüllt.

Die elliptischen Kurven sind also über sogenannte Weierstraß-Gleichungen in Normalform definiert. Die Lösung einer solchen Weierstraß-Gleichung liefert in der Regel eine elliptische

[8] Streng genommen ist eine elliptische Kurve durch $f(x,y,z) = y^2z + a_1xyz + a_2yz^2 - x^3 + a_3x^2z + a_4xz^2 + a_5Z^3 \stackrel{m}{=} 0$ gegeben(vgl.[MS]S.14)

Kurve. Die Weierstraß-Gleichungen sind durch den Ausdruck $F(x,y) = 0$ gegeben. Am Beispiel ergibt sich für $F(x,y)$ ein 3d-Funktionsgraph[9]:

Abb. 5

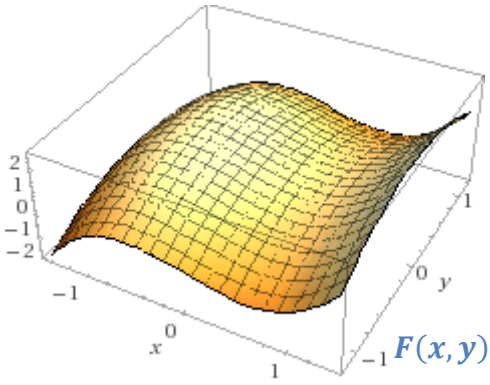
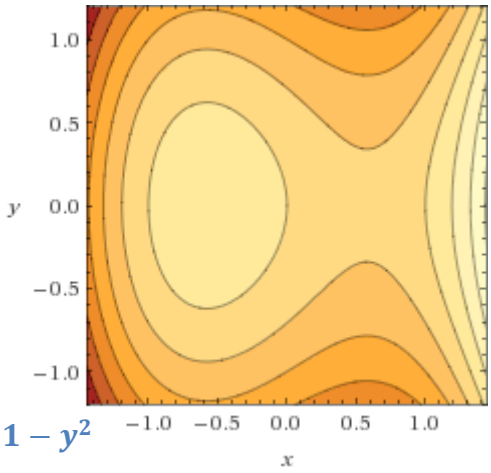


Abb. 6



$F(x,y) = x^3 - x + 1 - y^2$

Die Höhenlinien sind also verschiedene elliptische Kurven, die sich durch Variation von a_5 ergeben. Die Kryptographisch relevanten Kurven haben die vereinfachte Form von der im Folgenden ausgegangen wird:

$$F(x,y) \coloneqq x^3 + a_1 x + a_2 - y^2 = 0$$

Aus dieser ergibt sich die gebräuchliche oft anzutreffende Gleichung $y^2 = x^3 + a_1 x + a_2$.

Die verwendeten Kurven dürfen keine Singularitäten haben, damit man im Anschluss eine abelsche Gruppe über ihnen definieren kann. Also muss die Tangente in jedem Punkt exakt definiert sein. Um dies zu gewährleisten, muss mindestens eine partielle Ableitung von F ungleich null sein bzw. der Gradient von $F(x,y)$ darf nicht dem Nullvektor entsprechen:

$$\frac{\partial F}{\partial x}(a,b) \neq 0 \quad \vee \quad \frac{\partial F}{\partial y}(a,b) \neq 0$$

$$\nabla F(x,y) \neq \vec{0}$$

Aus dieser Forderung lässt sich eine Bedingung herleiten, die die Kurve auf Singularität prüft:

$$\nabla F(x,y) = \begin{pmatrix} \frac{\partial F}{\partial x}(x^3 + a_1 x + a_2 - y^2) \\ \frac{\partial F}{\partial y}(x^3 + a_1 x + a_2 - y^2) \end{pmatrix} = \begin{pmatrix} 3x^2 + a_1 \\ 2y \end{pmatrix} \neq \vec{0}$$

$$2y \neq 0 \Rightarrow y \neq 0$$

(vgl [EK]2.1)

[9] Abb. 5 und Abb. 6 [2] Quelle: http://www.wolframalpha.com/input/?i=F%28x%2Cy%29%3Dx^3-x%2B1-y^2

Also kann eine Singularität der gegeben Kurven nur ‘‘auf der x-Achse‘‘ existieren. Außerdem gilt:

$$3x^2 + a_1 \neq 0 \Rightarrow x \neq \pm\sqrt{-\frac{a_1}{3}}$$

Nach Einsetzen in die vorgegebene Gleichung ergibt sich:

$$\sqrt{-\frac{a_1}{3}}^3 + a_1\sqrt{-\frac{a_1}{3}} + a_2 = 0$$

Nach weiteren Vereinfachungen erhält man die übliche Formel der Bedingung für eine Singularität:

$$4{a_1}^3 + 27{a_2}^2 = 0$$

Äquivalent zu dieser Bedingung ist die Forderung der Quadratfreiheit des Polynoms. Was sich zeigen lässt, wenn man davon ausgeht, dass das Polynom nicht quadratfrei ist:
$x^3 + a_1x + a_2 = (x+b)^2 * (x+c) = x^3 + x^2(2b+c) + x(2bc+b^2) + b^2c$

Daraus folgen die Gleichungen:

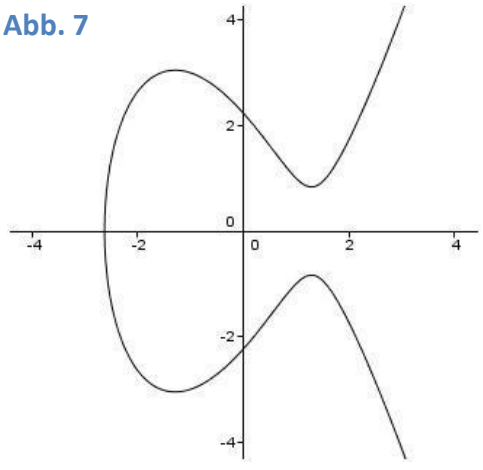
$$2b + c = 0$$

$$2bc + b^2 = a_1$$

$$b^2c = a_2$$

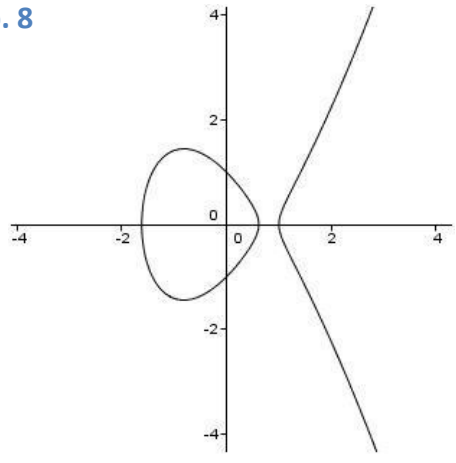
Aus denen sich dann ebenfalls die obere Bedingung herleiten lässt. Einige Beispiele für elliptische Kurven sind im Folgenden dargestellt:

Abb. 7



$y^2=x^3-5x+5$

Abb. 8



$y^2=x^3-2x+1$

Elliptische Kurven dieser Form haben Eigenschaften, die sich für die Kryptographie verwenden lassen, in dem man eine mathematische Operation über diesen Kurven definiert. Zum einen weisen sie stets eine Achsen-Symmetrie zur x-Achse auf. Denn löst man durch Wurzelziehen nach y auf, so entsteht ein Wurzelausdruck. Bei positiven reellen Lösungen für

das Polynom ergeben sich zwei Lösungen: die positive Wurzel des Polynoms und die negative. Des weiteren besteht eine wichtige Eigenschaft dadurch, dass, wenn man durch zwei beliebige Punkte auf der Kurve eine Gerade zieht, von dieser Geraden genau ein weiterer Punkt auf der Kurve geschnitten wird. Aus diesen beiden Eigenschaften wird die Addition auf elliptischen Kurven definiert.

4.5.2 Addition auf elliptischen Kurven

Man wählt zwei zu addierende Punkte A und B und ‘‘zieht‘‘ eine Gerade durch diese Punkte. Der weitere Schnittpunkt mit der Kurve sei -C. Man spiegelt den Punkt -C an der x-Achse und enthält das Ergebnis der Addition C.

Sonderfall: wenn die zu addierenden Punkte identisch sind, legt man eine Tangente an den Punkt und verfährt wie geschildert.

Abb. 9

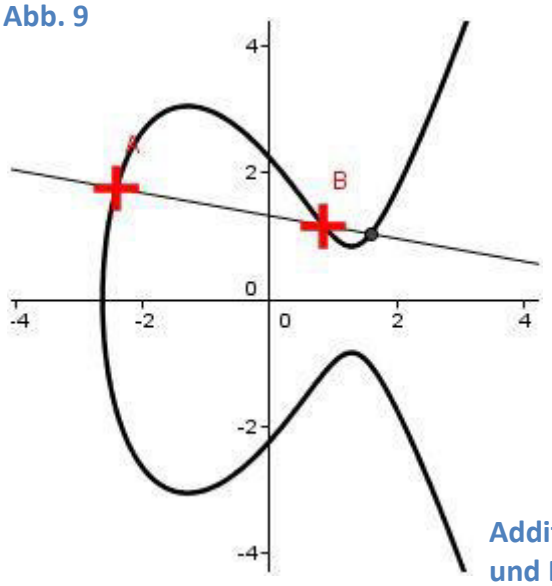
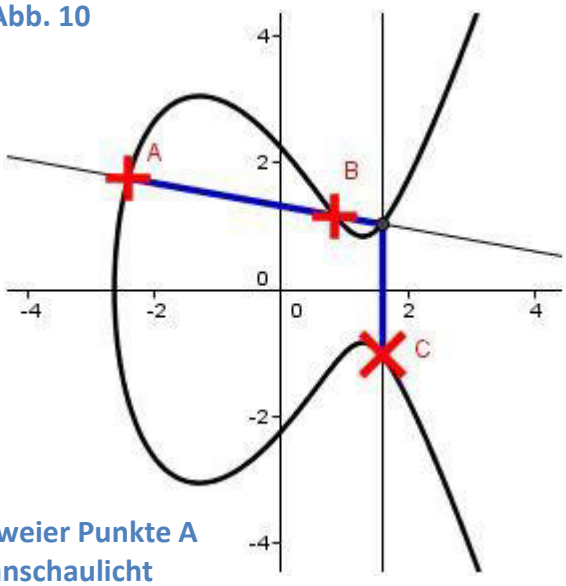


Abb. 10



Addition zweier Punkte A und B veranschaulicht

4.5.2.1 Das neutrale Element

Um eine vollständige abelsche Gruppe über einer elliptischen Kurve zu definieren, benötigt man ein neutrales Element. Dieses wird mit o bezeichnet. Dieser Punkt liegt im Unendlichen. Addiert man einen Punkt mit dem neutralen Element, erhält man den selben Punkt als Ergebnis. Weiterhin gilt: -P+P=o.

Konkret lässt sich die Addition also durch den Schnitt einer Geraden mit der elliptischen Kurve formalisieren. (vgl. [EK] 2.2)

4.5.3 Rechnen auf elliptischen Kurven

4.5.3.2 Addition zweier verschiedener Punkte P1 und P2

Um eine Addition zweier verschiedener Punkte vorzunehmen, definiert man sich zunächst einige Größen und Funktionen:

$$g(x) = mx + n \qquad P1(x_1; y_1) \qquad P2(x_2; y_2)$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$n = y_1 - \frac{y_2 - y_1}{x_2 - x_1} x_1$$

$$y^2 = x^3 + a_1 x + a_2$$

Schneidet man eine Gerade mit der elliptischen Kurve ergibt sich:

$$(mx + n)^2 = x^3 + a_1 x + a_2$$

$$m^2x^2 + 2mnx + n^2 = x^3 + a_1 x + a_2$$

$$0 = x^3 - m^2x^2 + (a_1 - 2mn)x + (a_2 - n^2)$$

Da bei der Addition drei Schnittstellen der Geraden mit der elliptischen Kurve vorhanden sind, hat das obenstehende Polynom genau 3 reelle Lösungen. Daher lässt es sich in Linearfaktoren zerlegen:

$$x^3 - m^2x^2 + (a_1 - 2mn)x + (a_2 - n^2) = (x - x_1)(x - x_2)(x - x_3)$$

$$(x - x_1)(x - x_2)(x - x_3) = (x^2 - x_1x - x_2x + x_1x_2)(x - x_3)$$

$$= x^3 - x_1x^2 - x_2x^2 + x_1x_2x - x^2x_3 + x_1x_3x + x_2x_3x - x_1x_2x_3$$

$$= x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3$$

Durch Koeffizientenvergleich ergibt sich:

$$x_1 + x_2 + x_3 = m^2$$

$$\Rightarrow x_3 = m^2 - x_1 - x_2$$

Einsetzen in die Geradengleichung und gespiegelt an der x-Achse ergibt:

$$y_3 = -(m\, x_3 + n)$$

Also berechnet sich der Punkt $P3(x_3; y_3)$ als Addition der Punkte $P1(x_1; y_1)$und $P2(x_2; y_2)$ auf der elliptischen Kurve $E: y^2 = x^3 + a_1 x + a_2$ mit $x, a_1, a_2 \in \mathbb{R}$ durch $P3(m^2 - x_1 - x_2; -(m\, x_3 + n))$.

Alternativ kann man die Einschränkung, dass immer 3 Schnittpunkte vorhanden sind, weglassen, und die bereits normierte Gleichung mittels Tschirnhaus-Transformation(vgl.[QG]) um das quadratische Glied des Polynoms reduzieren. Zu diesem Zweck substituiert man das Argument:

$$Substitution: x := \alpha t + \beta$$

$$0 = (\alpha t + \beta)^3 - m^2(\alpha t + \beta)^2 + (a_1 - 2mn)(\alpha t + \beta) + (a_2 - n^2)$$

Auflösen ergibt:

$$0 = t^3 + \frac{3\beta - m^2}{\alpha} * t^2 + \frac{3\beta^2 - 2m^2\beta + (a_1 - 2mn)}{\alpha^2} * t + \frac{\beta^3 - m^2\beta^2 + (a_1 - 2mn)\beta + (a_2 - n^2)}{\alpha^3}$$

Substituiert man nun ergibt sich:

$$Substitution: \beta := \frac{m^2}{3}$$

$$0 = t^3 + \frac{3\frac{m^2}{3} - m^2}{\alpha} * t^2 + \frac{3\left(\frac{m^2}{3}\right)^2 - 2m^2\frac{m^2}{3} + (a_1 - 2mn)}{\alpha^2} * t + \frac{\left(\frac{m^2}{3}\right)^3 - m^2\left(\frac{m^2}{3}\right)^2 + (a_1 - 2mn)\frac{m^2}{3} + (a_2 - n^2)}{\alpha^3}$$

$$t^3 + \frac{(a_1 - 2mn) - \frac{(-m^2)^2}{3}}{\alpha^2} * t + \frac{\frac{2(-m^2)^3}{27} - \frac{(-m^2)(a_1 - 2mn)}{3} + (a_2 - n^2)}{\alpha^3} = 0$$

Diese Gleichung kann man nun mit der Cardanischen Formel lösen und den Spezialfall betrachten, dass genau drei reelle Lösungen vorhanden sind. (Ironischerweise ist dies der Fall, wenn die Diskriminante ein negatives Vorzeichen hat)

4.5.3.2 Addition zweier identischer Punkte P1:

Genau wie zuvor werden einige wichtige Größen definiert:

$$g(x) = mx + n \qquad P1(x_1; y_1)$$

$$y^2 = x^3 + a_1 x + a_2$$

$$\frac{dy}{dx} = \frac{d}{dx}\sqrt{x^3 + a_1 x + a_2} = \frac{1}{2} * \frac{1}{\sqrt{x^3 + a_1 x + a_2}} * (3x^2 + a_1)$$

Mit $y = \sqrt{x^3 + a_1 x + a_2}$

$$\Rightarrow m = \frac{dy}{dx} = \frac{(3x^2 + a_1)}{2y}$$

In die Geradengleichung einsetzen ergibt:

$$y_1 = \frac{\left(3{x_1}^2 + a_1\right)}{2y_1} x_1 + n$$

$$n = y_1 - \frac{\left(3{x_1}^2 + a_1\right)}{2y_1} x_1$$

Da bei der Addition zweier identischer Punkte ein Schnittpunkt und ein ‘‘Berührpunkt‘‘ der Geraden mit der elliptischen Kurve vorhanden sind, hat die Gleichung

$$0 = x^3 - m^2 x^2 + (a_1 - 2mn)x + (a_2 - n^2)$$

genau 2 reelle Lösungen, von denen eine die ‘‘Berührstelle‘‘ eine doppelte Nullstelle der vorhandenen ganzrationalen Funktion dritten Gerades sein muss. Also gilt folgendes:

$$x^3 - m^2 x^2 + (a_1 - 2mn)x + (a_2 - n^2) = (x - x_1)^2 (x - x_2)$$

$$(x - x_1)^2 (x - x_2) = (x^2 - 2x x_1 + {x_1}^2)(x - x_2)$$

$$= x^3 - x^2 x_2 - 2x^2 x_1 + 2x x_1 x_2 + {x_1}^2 x - {x_1}^2 x_2$$

$$= x^3 - (x_2 + 2x_1)x^2 + (2x_1 x_2 + {x_1}^2)x - {x_1}^2 x_2$$

Durch Koeffizientenvergleich ergibt sich diesmal:

$$m^2 = x_2 + 2x_1$$

$$x_2 = m^2 - 2x_1$$

Einsetzen in die Geradengleichung, gespiegelt an der x-Achse ergibt:

$$y_2 = -(m\, x_2 + n)$$

Nun kann man eine Gruppe G(R;+) mit der besprochenen Punktaddition über eine elliptischen Kurve E definieren, was sich wie folgt zeigt:

1. Die Abgeschlossenheit gilt

2. die Addition ist kommutativ

3. o ist neutrales Element

4. Jedes Element P(x,y) hat ein Inverses P(x,-y) auf Grund der x-Achsensymmetrie

5. Es gilt das Assioziativgesetz

Der Beweis des Assoziativgesetzes kann mit Schulmathematik geführt werden, in dem man 3 Punkte allgemein hält und die Gleichungen ausnutzt. Durch endlose Umformungen kann man dann zeigen, dass $(A + B) + C = A + (B + C)$ gilt. Dies wird hier der Länge wegen nicht gezeigt, aber an einer Skizze veranschaulicht(vgl.[EK] 2.2):

Abb. 11

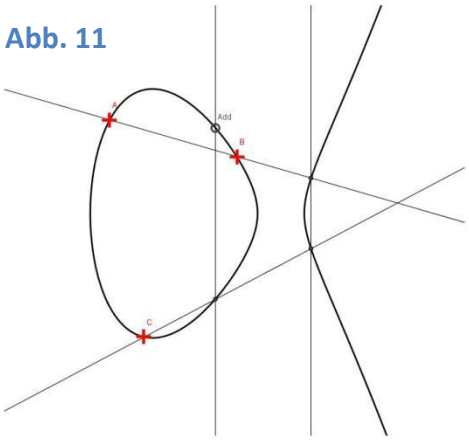
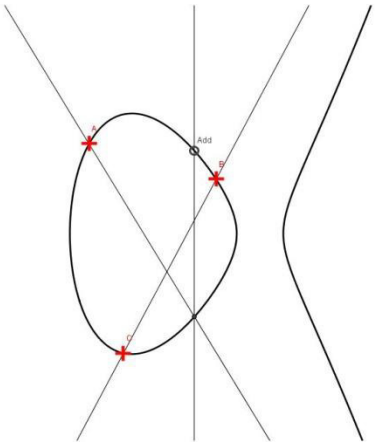


Abb. 12



4.5.4 Gruppe mit Restklassenkörper über einer elliptischen Kurve

Bisher wurden elliptische Kurven über ganz R betrachtet. In der Praxis ist diese Vorgehensweise gerade beim Verschlüsseln zu ungenau, da immer wieder gerundet werden muss. Die Kryptographie, die erzielt werden soll, verlangt aber eindeutige, exakte Lösungen. Um dies zu erreichen ist der Zahlenraum der elliptischen Kurve nicht mehr R, sondern Z_p. Die Kurve besteht nicht mehr aus unendlich vielen Punkten, sondern aus endlich vielen.

$Z_p$ist hierbei der Restklassenkörper und p eine Primzahl und das Modul. Die Wahl einer Primzahl ist daher notwendig, damit zu jeder Zahl ein eindeutiges Modulares inverses existiert. Denn mit einer Primzahl als Modul ist gewährleistet, dass zu jedem Element ein eindeutiges Inverses existiert. (Dies folgt aus dem Satz über die modularen Inverse)

Da nun durch die Definition über einem endlichen Körper stets ganzzahlige Werte gegeben sind, lässt sich die Kurve nicht mehr stetig zeichnen. Zeichnet man die möglichen Punkte dennoch ein, ergibt sich z.B. ein Bild wie in Abb. 13 [10]veranschaulicht. Auch hier kann man nach dem selben Schema, das vereinbart wurde, Punkte addieren so wie in Abb. 14 veranschaulicht.

[10] Abb. 13 und Abb. 14 [3] Quelle: http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/

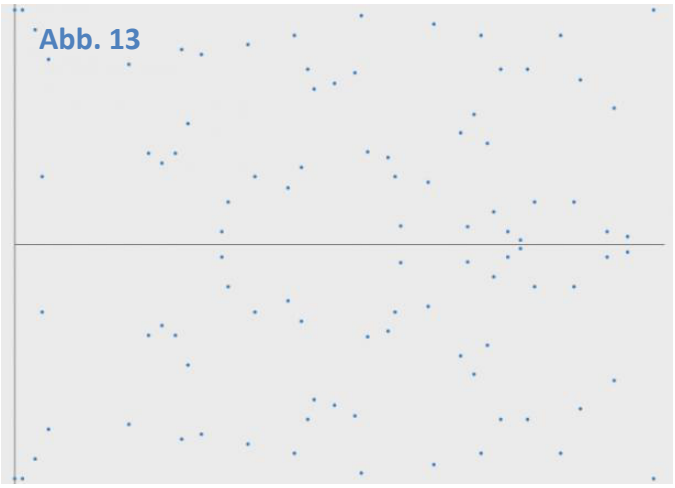


Abb. 13

$y^2=x^3-x+1$ mit Z_{97}

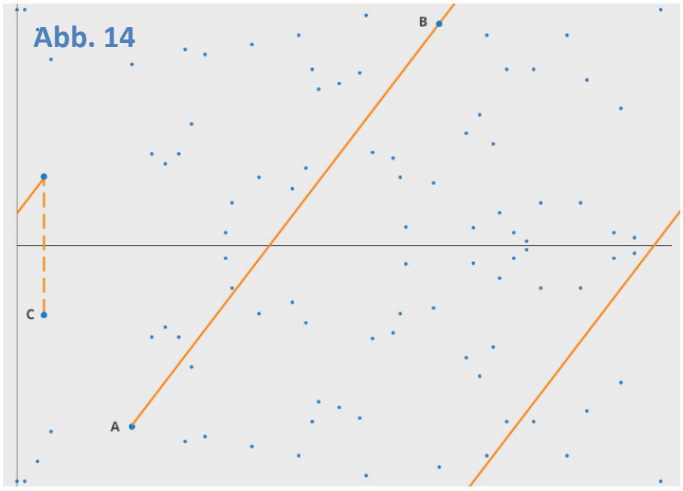


Abb. 14

Addition zweier Punkte auf der diskreten elliptischen Kurve

Wie kann man sich das anschaulich vorstellen?

Nun wird eine Modulo-Arithmetik verwendet, also gewisser weise eine „Uhren-Arithmetik“. Nur diesmal wird nicht der ‘‘Zahlenstrahl‘‘ zu einem Zyklus „gebogen“, sondern die x- und die y-Achse. Bei Krümmung einer Achse entsteht eine Röhre. Krümmt man zusätzlich die zweite Achse entsteht ein Torus. Zeichnet man nun Geraden ein, schlingen sie sich um den Torus.

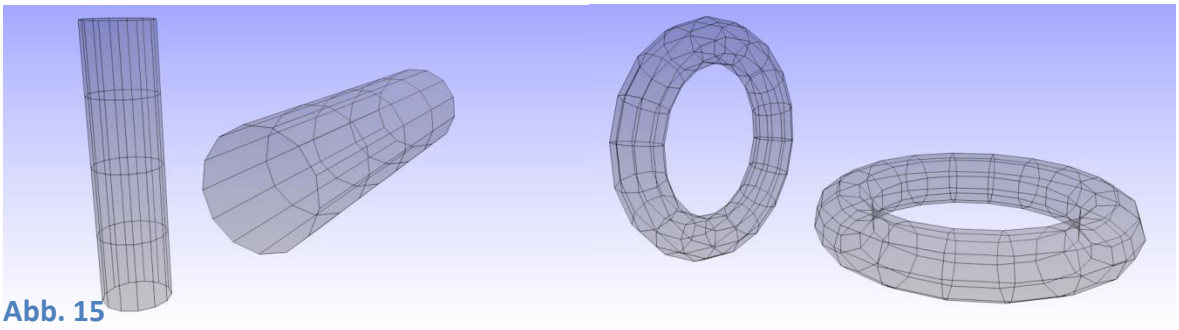


Abb. 15

4.5.5 Rechnen mit diskreten elliptischen Kurven

Das Rechnen auf diskreten elliptischen Kurven gestaltet sich sehr ähnlich zu den bereits gefundenen Rechenregeln. Auf eine genaue Herleitung soll aber an dieser Stelle verzichtet werden. Für die Addition zweier verschiedener Punkte gilt (vgl. [EC] S.18):

$$m = \frac{y_2 - y_1}{x_2 - x_1} \; mod \; p$$

$$x_3 = m^2 - x_1 - x_2 \; mod \; p$$

$$y_3 = -y_2 + m(x_2 - x_3) \; mod \; p$$

Für die Addition zweier identischer Punkte gilt entsprechend:

$$m = \frac{\left(3{x_1}^2 + a_1\right)}{2y_1} x_1 \; mod \; p$$

$$x_2 = m^2 - 2x_1 \; mod \; p$$

$$y_2 = -y_1 + m(x_1 - x_2) \; mod \; p$$

4.5.6 Wo ist die "Falltür"?

Wenn man einen Anfangspunkt und einen Endpunkt gegeben hat, der aus der n-fachen Multiplikation des Anfangspunktes resultiert, stellt es sich als schwer heraus, n festzustellen, wenn man nur den Anfangspunkt und den Endpunkt hat. Den Punkt mit sich selbst zu multiplizieren ist einfach, aber wenn man umgekehrt nur den Anfangs und Endpunkt gegeben hat, ist es schwierig n herauszufinden. Darin liegt die Grundlage der Falltürfunktion (Trapdoor-function). Denn auch hier gestaltet sich das Dividieren über elliptischen Kurven als noch schwerer zu lösendes Problem als das ''einfache'' diskrete Logarithmieren. Tatsächlich kann man prinzipiell jedes asymmetrische Kryptoverfahren, das auf dem diskreten Logarithmusproblem basiert, in ein Elliptic-Curve-Kryptosystem umwandeln.

4.6 Der elliptische Deffie Hellman Schlüsselaustausch

Mit den elliptischen Kurven lassen sich nun natürlich auch die oben vorgestellten Verfahren, die auf dem diskreten Logarithmusproblem aufbauen, verbessern und elegant praxistauglicher machen.

Zunächst einigen sich Alice und Bob auf eine öffentlich einsehbare Kurve und wählen also die Kurvenparameter a und b sowie eine Primzahl p als Modul. Diese Daten sind öffentlich einsehbar. Außerdem wählen die beiden einen öffentlichen Punkt P, der auf der Kurve liegt. Nun wählt jeder eine geheime natürliche Zahl, die kleiner als die Primzahl ist(A_k, B_k).

An ein und denselben Schlüssel, den ein Außenstehender, der den Austausch abhört, nicht gelangen kann, kommen sie erneut wie folgt:

1. Alice berechnet $X_A = A_k * P$, Bob berechnet $X_B = B_k * P$
2. Alice schickt X_A an Bob und Bob schickt X_B an Alice
3. Alice berechnet den gemeinsamen Schlüssel $S = X_B * A_k$ ebenso wie Bob $S = X_A * B_k$

Beide kennen jetzt den selben Schlüssel:

$$S = X_B * A_k = (B_k * P) * A_k = B_k * (P * A_k) = B_k * X_A$$

Will nun jemand den Schlüssel herausfinden, würde er nur die elliptische Kurve, den Punkt P, $X_A = A_k * P$ und $X_B = B_k * P$ kennen und steht nun vor dem Problem durch diskrete

Division auf einer elliptischen Kurve entweder an A_k oder an B_k zu gelangen. Da die Division auf elliptischen Kurven auf das diskrete Logarithmusproblem zurückzuführen ist, ist dieses Problem mindestens so schwierig wie das diskrete Logarithmusproblem (der genaue Beweis wird an dieser Stelle weggelassen). Vermutlich handelt es sich hierbei aber um ein noch schwerer zu lösendes Problem. Bis heute existiert kein Algorithmus, der dieses Problem in polynominaler Laufzeit lösen kann, also in einer Laufzeit, die mit zunehmender Schlüsselgröße sich nicht ‘‘stärker‘‘ als eine Polynomfunktion vergrößert.

4.7 Die elliptische Elgamal-Verschlüsselung

Alice und Bob schließen in der elliptische Elgamal-Verschlüsselung die selben Vorkehrungen ab wie beim elliptischen Deffie Hellman Schlüsselaustausch. Sie kennen also eine gemeinsame Kurve, einen Punkt auf der Kurve und haben X_A und X_B ausgetauscht. Ein Klartext T wird nun von Alice verschlüsselt, indem sie $G = X_B * A_k + T$ und an Bob schickt. Dieser entschlüsselt den Geheimtext G, indem er $T = X_A * (-B_k) + G$ berechnet. Denn:

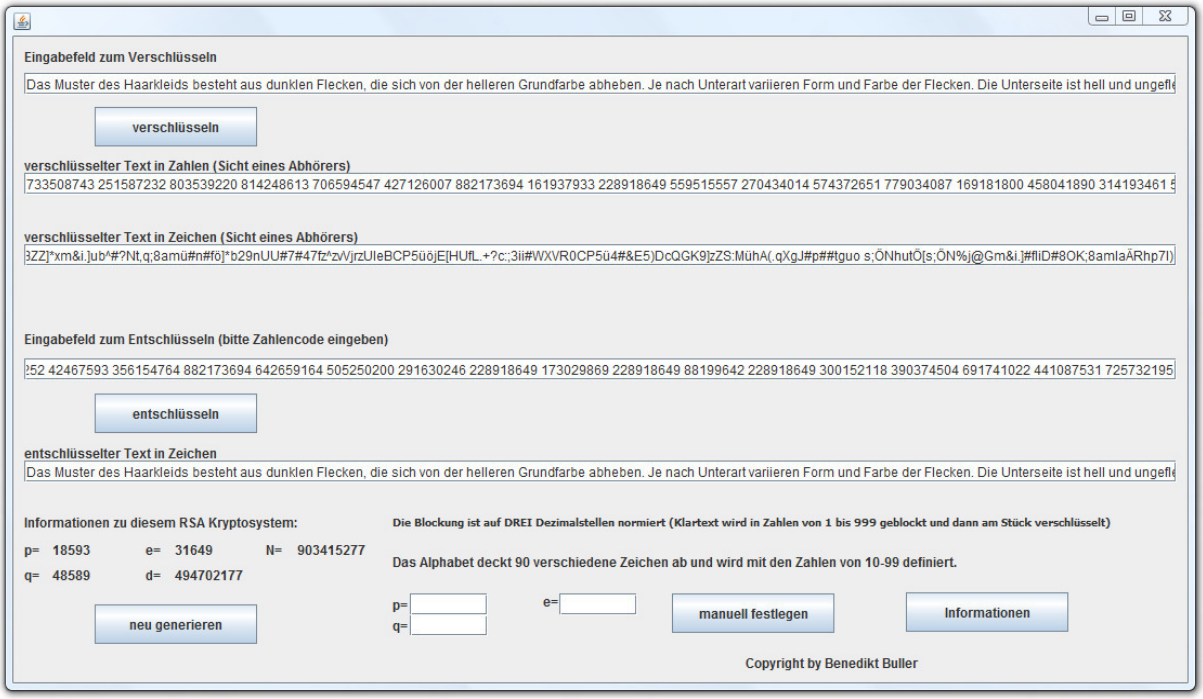
$$T = X_A * (-B_k) + G = X_A * (-B_k) + X_B * A_k + T = A_k * P * (-B_k) + B_k * P * A_k + T = T$$

Die Anfälligkeit für einen „Man-in-the Middle“ Angriff ist aber weiterhin gegeben. Ein Authentifizierungsverfahren ist also Notwendig.

5 Praktische Implementierung des RSA Verfahrens

Zu Beginn eines Kryptosystems steht in der Anwendung natürlich die theoretische Überlegung, die dahinter steckt. Im Kern wird ein implementiertes Verschlüsselungsprogramm immer die Methodik und die Idee als wichtigsten Bestandteil haben. Jedoch stellen sich in der Umsetzung Herausforderungen, mit denen man bei der theoretischen Planung evtl. nicht gerechnet hat. Es zeigt sich vermutlich auch, dass die Umsetzung selbst dann, wenn die Methodik einfach erscheint, immer noch kompliziert ist. Vor diesem Hintergrund habe ich selbst ein RSA Kryptoverfahren in Form einer Java Anwendung entwickelt:

Abb. 16

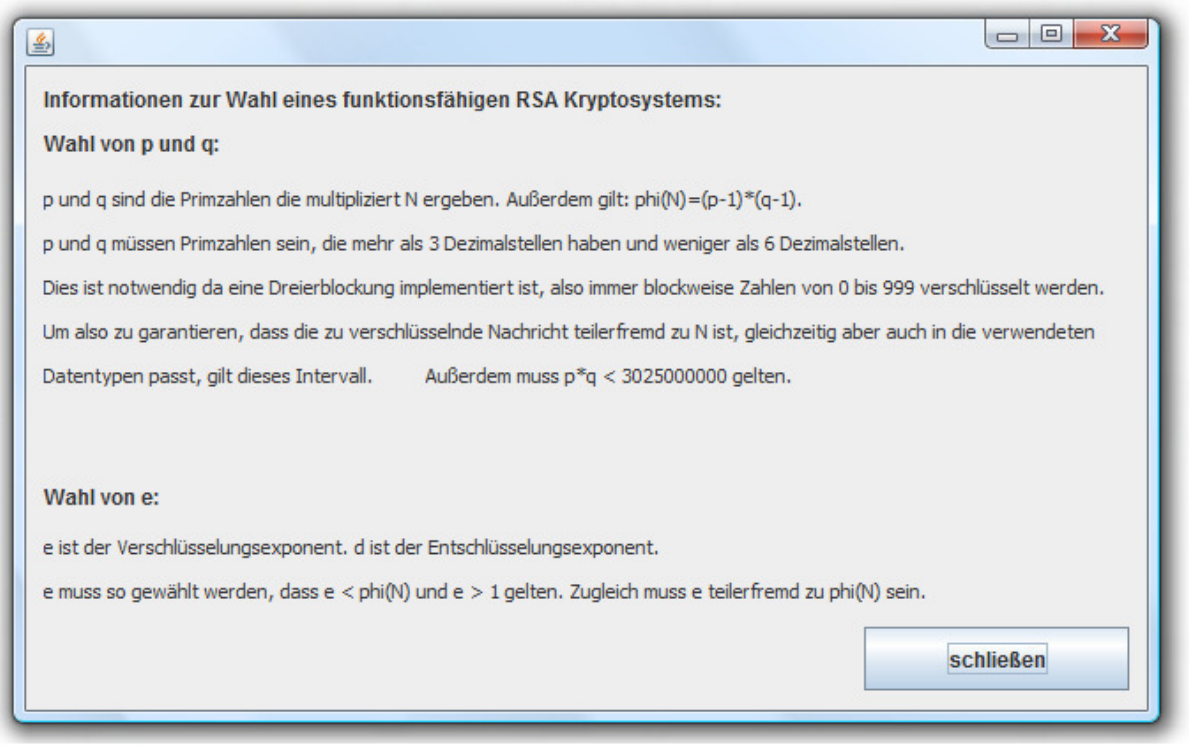


Die Benutzeroberfläche besteht aus einer Eingabezeile, in der der Benutzer einen Text bestehend aus sprachlichen Standartzeichen eingeben kann. Über den Button “verschlüsseln“ wird der Text verschlüsselt und zum einen als Zahlen-Blockcode und zum anderen als Zeichencode in den Zeilen darunter ausgegeben.

Unter dem Bereich zum Verschlüsseln befindet sich eine Zeile, in der man (den) Zahlenblockcode eingeben kann. Nach einem Klick auf den “entschlüsseln“-Button wird der Blockcode entschlüsselt, und es ergibt sich erneut der Klartext in der Zeile unter dem „entschlüsseln“-Button. Im unteren Viertel des Fensters befinden sich alle wichtigen Parameter des Kryptosystems. Man kann diese mit einem Klick auf ‘‘neu generieren“ neu erstellen lassen oder rechts daneben per Eingabe gewünschte Parameter festlegen. In der

manuellen Wahl von p, q und e sind bewusst keine ‘‘Schranken‘‘ im Hintergrund eingebaut worden. Daher gibt es auch den Button “Informationen“, der ein zusätzliches Fenster öffnet, in dem beschrieben wird, was für Eigenschaften p, q und e haben müssen, damit die Verschlüsselung ordnungsgemäß funktioniert:

Abb. 17



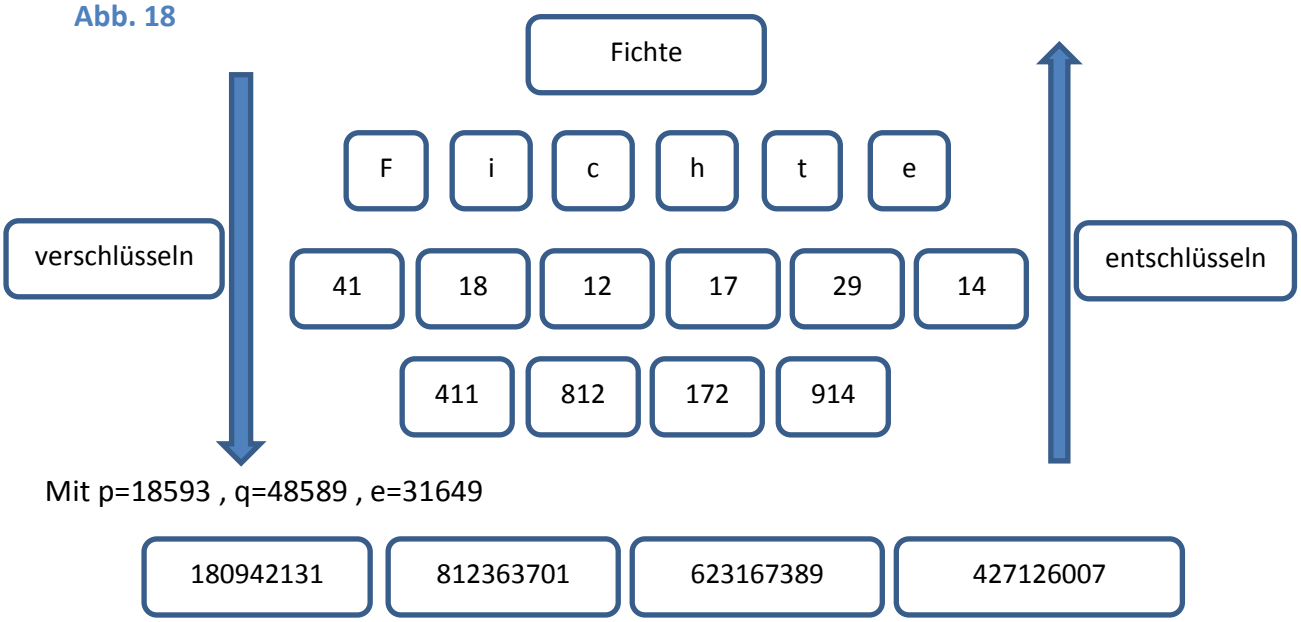
5.1 Funktion des Programms

Startet man das Programm, wird als erster wesentlicher Schritt das Kryptosystem nach den Regeln aus dem Kapitel RSA gebildet. Die Primzahlen werden zufällig aus einem Pool von ca. 8000 fünfstelligen Primzahlen gewählt, so dass es zumindest manuell schwer sein dürfte die Verschlüsselung zu knacken. e wird nebenbei ebenfalls aus diesem Pool entnommen um die Teilerfremdheit zu Phi(N) zu gewährleisten. Ergeben sich ungeschickte Konstellationen z.B. dass beide Primzahlen gleich sind wird neu ‘‘gezogen‘‘.

Gibt man einen Text in das Eingabefeld zum Verschlüsseln ein und drückt “verschlüsseln“, wird jedem einzelnen Zeichen eine zweistellige Zahl zugeordnet, die aus einem definiertem Alphabet entnommen wird. Aneinandergereiht werden die Zahlen als String gespeichert. Anschließend wird die Zeichenkette bestehend aus Ziffern in Dreier-Blöcke aufgeteilt und in Zahlen vom Typ long umdefiniert. Jeder dieser Blöcke wird durch Potenzieren verschlüsselt. Alle verschlüsselten Blöcke werden in einem String mit einem Leerzeichen getrennt aneinandergereiht und anschließend in dem Zeilenfeld unter dem Verschlüsselnbutton ausgegeben. “Zieht“ man über diesen Code das Alphabet “darüber“ ergibt sich die Zeile darunter, die dann offensichtlich mit sinnlosem Code gefüllt ist. Durch die direkte

Übersetzung geht nämlich die Blockung verloren und eine Zurückübersetzung wird unmöglich. Am Beispiel sieht das Verschlüsseln wie Folgt aus (Alphabet im Anhang):

Abb. 18



Mit p=18593 , q=48589 , e=31649

Gibt man in das Eingabefeld zum Entschlüsseln einen passenden Zahlenblockcode ein und drückt auf “entschlüsseln“, werden die einzelnen Blöcke voneinander getrennt und einzeln entschlüsselt. Die entstehenden Dreierblöcke werden in einem String ohne Trennung aneinander gereiht. Jedoch werden bei einem zweistelligen Ergebnis eine Null vorne und bei einem einstelligen Ergebnis 2 Nullen vorne ergänzt. Darauf wird nun das Alphabet ‘‘angewendet‘‘ und das Ergebnis ausgegeben.

5.2 Die Blockung

Da das verwendete Alphabet stets zweistellige Zahlen zuordnet, also eine ‘‘gerade Blockung“ vorhanden ist, ist es nicht sinnvoll beim Verschlüsseln ebenfalls eine zweier Blockung zu verwenden, da man dann prinzipiell nur eine monoalphabetische Substitution der Buchstaben vorliegen hätte. Eine Häufigkeitsanalyse wäre ohne Probleme möglich. Ähnlich verhält es sich beim Verwenden einer Vierer-Blockung. Anhand der Analyse von Buchstabenkombinationen wie ei, eu, äu etc. kann man theoretisch einiges über den Text erfahren. Daher liegt es nahe eine zur alphabetischen Blockung ‘‘antisymmetrische“ Aufteilung zu verwenden. Im Konkreten Fall also eine Dreier-Blockung. Für die sicherere Fünfer-Blockung müssten größere Primzahlen verwendet werden um zu garantieren, dass die Nachricht teilerfremd zu N ist. Praktisch könnte man es dennoch versuchen, da die Wahrscheinlichkeit dass die zu verschlüsselnde Nachricht tatsächlich einer der beiden Primzahlen entspricht minimal ist.

5.3 Probleme der Praxis

Beim Programmieren stellte sich schnell heraus, dass die Schwierigkeit weniger darin besteht, das RSA Verfahren zu implementieren, sondern vor allem darin, das RSA Verfahren auf jeden individuell unterschiedlichen Klartext anzuwenden. Größtes Hindernis dabei war, die ‘‘Pakete‘‘ zu definieren, die einzeln verschlüsselt werden sollten und diese geordnet zu verwalten. Außerdem durfte die Nutzerfreundlichkeit nicht vernachlässigt werden. Man konnte wohl kaum in der Ausgabe mit Arrays arbeiten, sondern musste stets den Array in einen String und den eingegebenen String in einen Array umwandeln. Auch um eine ungerade Dreier-Blockung gegenüber einer geraden Zweier-Alphabetblockung zu implementieren, musste man sich Tricks wie das Ergänzen von Nullen beim Entschlüsseln bestimmter Blöcke überlegen. Das modulare Potenzieren war im Vergleich dazu durch systematisches Vereinfachen schnell realisiert.

5.4 Verbesserungsmöglichkeiten

Als Verbesserung, die man bei dem Programm[11] noch vornehmen könnte, kann man die Erweiterung mit BigIntegern nennen. Dadurch wird es möglich beinahe beliebig große Primzahlen zu wählen. Dadurch könnte man ein größeres Alphabet und eine größere Blockung wählen und die Sicherheit dadurch verbessern. Man kann auch die Effizienz im allgemeinen betrachten und versuchen, den Quellcode weiter zu vereinfachen oder Erweiterungen (z.B. dass Files ausgelesen, verschlüsselt und abgespeichert werden) zu implementieren.

[11] Wichtige Teile des Quelltextes und das Alphabet sind im Anhang hinterlegt. Das gesamte Programm ist auf dem beigelegten Datenträger zu finden.

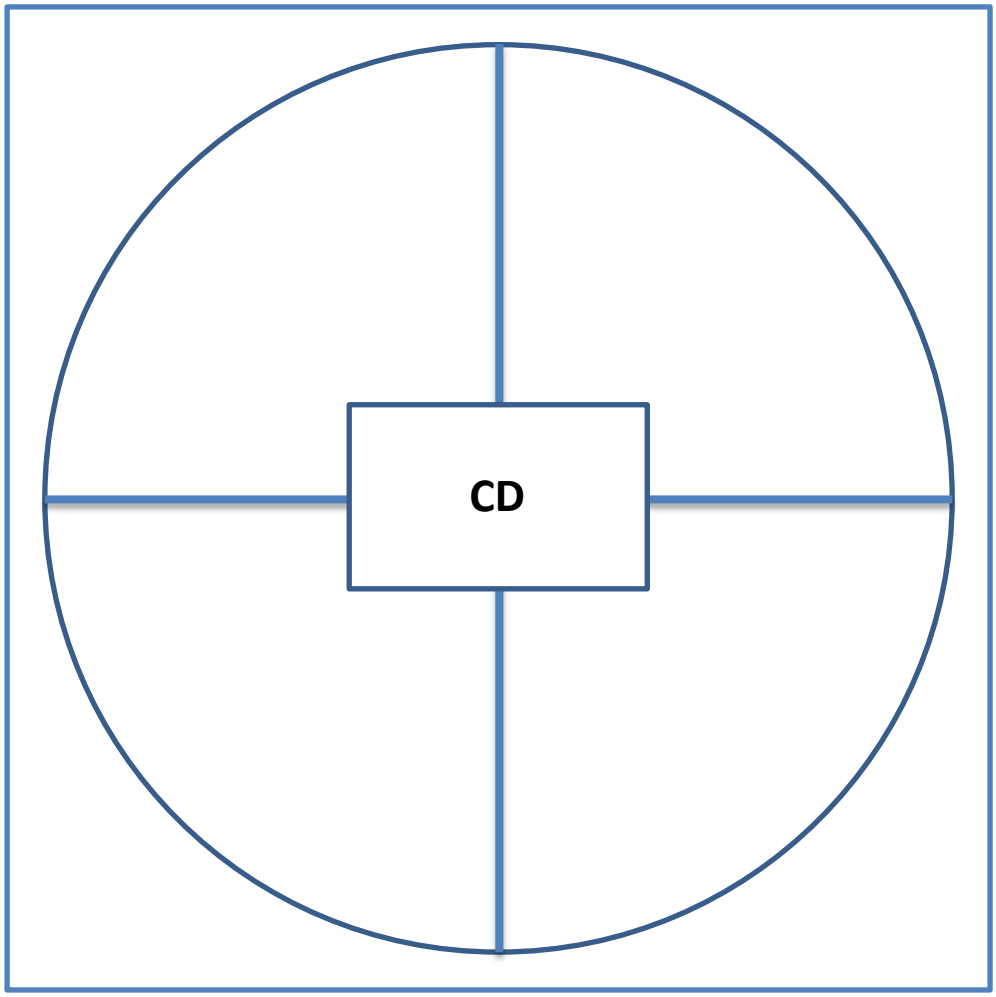
6 Ausblick

ECC und vor allem RSA ist heutzutage weitverbreitet. Aber selbst wenn diese Möglichkeiten der Verschlüsselung heute noch als sicher gelten, ist dies nicht für die Zukunft gewiss. Es scheint sich künftig eine vollkommen neue Art und Weise der Verschlüsselung und eine neue Generation der Computer zu entwickeln, die unter logischen Voraussetzungen der Quantenmechanik funktionieren. Schafft man es einen solchen Quantencomputer in einer gewissen Größe zu entwickeln, sind die vorgestellten asymmetrischen Verfahren vermutlich nicht mehr sicher. Es gibt z.B. einen Algorithmus der Quanteninformatik, der Shor-Algorithmus genannt wird, und der in einer Laufzeit von $\log(n)^3$ eine Primfaktorzerlegung durchführen kann. RSA wäre damit nicht mehr sicher. Die NSA soll an einem solchen Quantencomputer arbeiten. Um einen solchen herzustellen, scheitert es zur Zeit aber noch daran größere Strukturen aus quantenmechanischen Verschränkungen (Zusammenhänge zwischen den ‘‘Bauteilen‘‘) dauerhaft stabil und schnell aufzubauen. Aber wenn dies im geheimen gelingen sollte, so ist alles, was auf herkömmlichen Verfahren beruht, leicht zu knacken. Auch auf dieses Problem hat die Quantenmechanik eine Antwort. Es gibt z.B. Verfahren in der Quantenverschlüsselung, die heute schon realisierbar sind, aber da sie auf politischer Ebene nicht gefördert werden, kaum in Erscheinung treten. Z.B. werden beim Quantenschlüsselaustausch Polarisationseffekte ausgenutzt. Ein Angreifer, der das Signal abhören will, verändert es dadurch zwangsläufig. Alice und Bob fällt es also auf, wenn sie abgehört werden und einigen sich frühzeitig auf einen neuen Schlüssel. Wenn der Angreifer bei vergleichbaren Verfahren mithören will, verändert und zerstört er also womöglich sofort die Nachricht, so dass das Abhören unmöglich wird.

7 Resümee

Nachdem nun viele Aspekte vor dem Hintergrund der Ergebnisse und Zusammenhänge, die herausgestellt wurden, erläutert wurden, hoffe ich sehr, die Vielfalt des kryptographischen Themenbereiches, die schon zu Beginn angedeutet wurde, herausgestellt zu haben. Man kann vom Schwierigkeitsgrad gemäßigte Sachverhalte, wie die Cäsar Verschlüsselung, abhandeln. Man kann sich aber auf Grundlage des selben Prinzips über die Themen Vigenere-Verschlüsselung, one-time-pad bis hin zur Beschreibung von symmetrischen Verfahren auf Basis von polyalphabetischer Substitution durch Matrizen steigern. Weiter wurde auf asymmetrische Verschlüsselung eingegangen, die im Gegensatz zur symmetrischen Verschlüsselung vergleichsweise neu ist. Spätestens hier muss man sich intensiv mit verschiedensten mathematischen Bereichen, wie z.B. der Zahlentheorie auseinandersetzen. Das RSA Verfahren wurde hier als erstes beispielhaftes Verfahren aufgegriffen. Anschließend wurden Verfahren auf Basis des diskreten Logarithmusproblems erläutert und danach mit Hilfe elliptischer Kurven verstärkt. Um nach diesem sehr theorieorientiertem Abschnitt die Brücke zur Praxis zu schlagen, erfolgte eine Implementation des RSA Algorithmus, die mir nach meinem subjektiven Empfinden auch sehr viel Spaß gemacht hat. Vor dem Hintergrund meines anfänglichen Ziels bin ich rückblickend einmal komplett durch den kryptographischen Bereich ‘‘gereist‘‘ und habe sehr gute Eindrücke sammeln können. Ich hoffe, dass ich dies auch dem aufmerksamen Leser in ähnlicher Art und Weise ermöglicht habe und ihm einige der Hauptideen in der Kryptographie näher bringen konnte. Ich hoffe außerdem, dass ich an der ein oder anderen Stelle kreative Impulse für diejenigen schaffen konnte, die sich hiervon inspirieren lassen wollen.

Datenträger mit Programm



Auf dem vorliegenden Datenträger befindet sich ein Ordner “application“ in dem sich die Executable Jar File “RSA_Verschlüsselung“ befindet. Außerdem ist der unkompilierte Projektordner “projekts3“ enthalten.

Anhang

Äquivalent zum Kreuzprodukt im vier- und n-dimensionalen

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix} = a_1 * \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} - a_2 * \begin{vmatrix} b_1 & c_1 & d_1 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} + a_3 * \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_4 & c_4 & d_4 \end{vmatrix} - a_4 * \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{vmatrix}$$

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix} = \vec{a} * \begin{pmatrix} \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 & d_1 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_3 \end{vmatrix} \\ \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_4 & c_4 & d_4 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_3 \\ b_3 & c_3 & d_3 \end{vmatrix} \end{pmatrix} = \vec{a} * \vec{x}$$

Der Betrag von $\vec{x}$ ist also ein 3dimensionales Volumen im 4dimensionalen Raum, welches von den Vektoren $\vec{b}$, $\vec{c}$ und $\vec{d}$ aufgespannt wird. Zudem ist das Skalarprodukt von $\vec{x}$ mit jedem der drei Vektoren Null. Die Behauptungen lassen sich einfach prüfen:

Nach der Behauptung muss der Betrag des Vektors $\vec{x}$ sofern die 4te Vektorkomponente von $\vec{b}$, $\vec{c}$ und $\vec{d}$ Null ist gleich dem Spatprodukt der Vektoren $\vec{b}$, $\vec{c}$ und $\vec{d}$ sein. Diese Beziehung kann man aufstellen und durch Umformungen überprüfen:

Es wird behauptet:

$$\left| \begin{pmatrix} \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ 0 & 0 & 0 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 & d_1 \\ b_3 & c_3 & d_3 \\ 0 & 0 & 0 \end{vmatrix} \\ \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ 0 & 0 & 0 \end{vmatrix} \\ -\begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_3 \\ b_3 & c_3 & d_3 \end{vmatrix} \end{pmatrix} \right| = \left| \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_3 \\ b_3 & c_3 & d_3 \end{vmatrix} \right|$$

Nach der Regel von Sarrus erkennt man sofort, dass dies stimmen muss:

$$\left| \begin{pmatrix} 0 \\ -0 \\ 0 \\ -\begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_3 \\ b_3 & c_3 & d_3 \end{vmatrix} \end{pmatrix} \right| = \left| \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_3 \\ b_3 & c_3 & d_3 \end{vmatrix} \right|$$

Maximum Likelihood Methode zu Berechnung von Parametern einer Wahrscheinlichkeitsdichtefunktion

Schätzen des Erwartungswertes μ bzw. der Varianz σ² bei normalverteilten Proben.

Die Funktion der Wahrscheinlichkeitsdichte bei der Normalverteilung ist definiert durch:

$$f_{\mu;\sigma}(x) = \frac{1}{\sqrt{2\pi} \times \sigma} \times e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Schätzung des Erwartungswertes:

Um den Erwartungswert für beliebige σ und beliebige Proben schätzen zu können, ersetzen wir zunächst μ durch unser θ und sehen die Proben und σ als konstant an:

$$\mathcal{L}(x_1, \dots, x_n, \sigma; \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi} \times \sigma} \times e^{-\frac{(x_i-\theta)^2}{2\sigma^2}}$$

$$\mathcal{L}(x_1, \dots, x_n, \sigma; \theta) = \frac{1}{(\sqrt{2\pi} \times \sigma)^n} \times \prod_{i=1}^{n} e^{-\frac{(x_i-\theta)^2}{2\sigma^2}}$$

$$\mathcal{L}(x_1, \dots, x_n, \sigma; \theta) = \frac{1}{(\sqrt{2\pi} \times \sigma)^n} \times e^{\sum_{i=1}^{n} -\frac{(x_i-\theta)^2}{2\sigma^2}}$$

Nach Logarithmieren ergibt sich:

$$\ln\big(\mathcal{L}(x_1, \dots, x_n, \sigma; \theta)\big) = \ln\left(\left(\sqrt{2\pi} \times \sigma\right)^{-n}\right) + \ln(e^{\sum_{i=1}^{n} -\frac{(x_i-\theta)^2}{2\sigma^2}})$$

$$\ln\big(\mathcal{L}(x_1, \dots, x_n, \sigma; \theta)\big) = -n \times \ln\big(\sqrt{2\pi} \times \sigma\big) + \sum_{i=1}^{n} -\frac{(x_i-\theta)^2}{2\sigma^2}$$

$$\ln\big(\mathcal{L}(x_1, \dots, x_n, \sigma; \theta)\big) = -n\big(\ln\big(\sqrt{2\pi}\big) + \ln(\sigma)\big) - \frac{\sum_{i=1}^{n}(x_i-\theta)^2}{2\sigma^2}$$

Nach θ Ableiten und Null setzen:

$$\frac{\partial \ln\big(\mathcal{L}(x_1, \dots, x_n, \sigma; \theta)\big)}{\partial\theta} = -\frac{\sum_{i=1}^{n} 2(x_i-\theta) \times (-1)}{2\sigma^2}$$

$$\frac{\partial \ln\big(\mathcal{L}(x_1, \dots, x_n, \sigma; \theta)\big)}{\partial\theta} = \frac{2 \times \sum_{i=1}^{n}(x_i-\theta)}{2\sigma^2} = \frac{\sum_{i=1}^{n}(x_i-\theta)}{\sigma^2}$$

$$\frac{\sum_{i=1}^{n}(x_i-\theta)}{\sigma^2} = 0$$

$$\sum_{i=1}^{n}(x_i - \theta) = 0$$

$$\left(\sum_{i=1}^{n} x_i\right) - \theta \times n = 0$$

$$\sum_{i=1}^{n} x_i = \theta \times n$$

$$\theta = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\boldsymbol{\widehat{\theta} = \overline{x}}$$

Schätzen der Varianz σ²

Um die Varianz mit der Likelihood Methode abzuschätzen, muss nun μ als konstant angesehen werden. σ wird durch unseren Parameter θ ersetzt und nach ihm partiell abgeleitet. Anschließend wird die Ableitung erneut mit Null gleichgesetzt.

$$\ln(\mathcal{L}(x_1, \dots, x_n, \mu; \theta)) = \ln(\frac{1}{(\sqrt{2\pi} \times \theta)^n} \times e^{\sum_{i=1}^{n} -\frac{(x_i-\mu)^2}{2\theta^2}})$$

$$\ln(\mathcal{L}(x_1, \dots, x_n, \mu; \theta)) = \ln(\left(\sqrt{2\pi} \times \theta\right)^{-n} \times e^{-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\theta^2}})$$

$$\ln\left(\mathcal{L}(x_1, \dots, x_n, \mu; \theta)\right) = -n \times \ln\left(\sqrt{2\pi}\right) - n \times \ln(\theta) - \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{2\theta^2}$$

$$\frac{\partial \ln\left(\mathcal{L}(x_1, \dots, x_n, \mu; \theta)\right)}{\partial \theta} = -\frac{n}{\theta} - \frac{\sum_{i=1}^{n}(x_i - \mu)^2 \times (-2)}{2} \times \theta^{-3}$$

$$\frac{\partial \ln\left(\mathcal{L}(x_1, \dots, x_n, \mu; \theta)\right)}{\partial \theta} = -\frac{n}{\theta} + \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{\theta^3}$$

$$-\frac{n}{\theta} + \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{\theta^3} = 0$$

$$\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{\theta^3} = \frac{n}{\theta}$$

$$\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n} = \frac{\theta^3}{\theta} = \theta^2$$

$$\boldsymbol{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n} = \theta^2}$$

Wichtiger Quelltext des meines RSA Verschlüsselungsprogramms

```
public void makecrypt(){

        long p=np.primzahl();
        long q=np.primzahl();
        pr=p;
        qr=q;
        long phi;

        //legt Primzahlen fest un wählt bei gleichen Primzahlen erneut
        while (p==q){
             p=np.primzahl();
             q=np.primzahl();
             pr=p;
             qr=q;
         }

        // blidet modul und phi von n
         n=p*q;
         phi=(p-1)*(q-1);

         // legt e als teilerfremde primzahl zu phi von n die kleiner als phi un
         e=np.primzahl();
         while (e>=phi&&e==p&&e==q){
            e=np.primzahl();
         }

         // bildung des Entschlüsselungsexponenten
         d=eu.euklidischeralg(e,phi);
         if(d<0){
             d=d+phi;
         }
    }

public void setalphabet(){
```

alphabet[0]="#";	alphabet[34]="y";	alphabet[67]="1";
alphabet[1]="#";	alphabet[35]="z";	alphabet[68]="2";
alphabet[2]="#";	alphabet[36]="A";	alphabet[69]="3";
alphabet[3]="#";	alphabet[37]="B";	alphabet[70]="4";
alphabet[4]="#";	alphabet[38]="C";	alphabet[71]="5";
alphabet[5]="#";	alphabet[39]="D";	alphabet[72]="6";
alphabet[6]="#";	alphabet[40]="E";	alphabet[73]="7";
alphabet[7]="#";	alphabet[41]="F";	alphabet[74]="8";
alphabet[8]="#";	alphabet[42]="G";	alphabet[75]="9";
alphabet[9]="#";	alphabet[43]="H";	alphabet[76]=":";
alphabet[10]="a";	alphabet[44]="I";	alphabet[77]="+";
alphabet[11]="b";	alphabet[45]="J";	alphabet[78]="-";
alphabet[12]="c";	alphabet[46]="K";	alphabet[79]="*";
alphabet[13]="d";	alphabet[47]="L";	alphabet[80]="/";
alphabet[14]="e";	alphabet[48]="M";	alphabet[81]="ß";
alphabet[15]="f";	alphabet[49]="N";	alphabet[82]="ü";
alphabet[16]="g";	alphabet[50]="O";	alphabet[83]="ö";
alphabet[17]="h";	alphabet[51]="P";	alphabet[84]="ä";
alphabet[18]="i";	alphabet[52]="Q";	alphabet[85]="Ü";
alphabet[19]="j";	alphabet[53]="R";	alphabet[86]="Ö";
alphabet[20]="k";	alphabet[54]="S";	alphabet[87]="Ä";
alphabet[21]="l";	alphabet[55]="T";	alphabet[88]="%";
alphabet[22]="m";	alphabet[56]="U";	alphabet[89]="&";

```
        alphabet[23]="n";
        alphabet[24]="o";
        alphabet[25]="p";
        alphabet[26]="q";
        alphabet[27]="r";
        alphabet[28]="s";
        alphabet[29]="t";
        alphabet[30]="u";
        alphabet[31]="v";
        alphabet[32]="w";
        alphabet[33]="x";
```

```
        alphabet[57]="V";
        alphabet[58]="W";
        alphabet[59]="X";
        alphabet[60]="Y";
        alphabet[61]="Z";
        alphabet[62]=" ";
        alphabet[63]=",";
        alphabet[64]=".";
        alphabet[65]="?";
        alphabet[66]="!";
```

```
        alphabet[90]="§";
        alphabet[91]=";";
        alphabet[92]="[";
        alphabet[93]="]";
        alphabet[94]="(";
        alphabet[95]=")";
        alphabet[96]="@";
        alphabet[97]="0";
        alphabet[98]="^";
        alphabet[99]="°";
```

```
}
```

Primzahlenliste:

```
public class primzahlregister
{
   long[] primArray = {
                10007, 10009, 10037,10039, 10061, 10067, 10069, 10079, 10091,
10093, 10099, 10103, 10111, 10133, 10139, 10141, 10151,10159, 10163, 10169, 10177,
10181, 10193,…
};
    public primzahlregister()
    {

    }
    public long primzahl()
    {
                return primArray[(int) (Math.random()*(primArray.length-1)+1)];
    }
}}
```

exemplarisch Entschlüsselungsmethode:

```
public String entschlüsselnadvanced(String code1){
        int c=0;
        int i=0;

        // erstellen des Arrays aus dem String beim entschlüsseln
        ////////////////////////
        String output="";
        while(i<code1.length()){
            if((code1.substring(i,i+1)).equals(" ")){
                c++;
            }
            i++;
        }
        long codearr[]=new long[c+1];
        i=0;
        c=1;
        int untergrenze=-1;
        int obergrenze=1;
        while(i<codearr.length){

            while(!(code1.substring(c-1,c)).equals(" ")&&c<code1.length()){

                c++;
            }

            obergrenze=c-1;
```

```
            try{
                codearr[i]=
Long.parseLong(code1.substring(untergrenze+1,obergrenze));
            }
            catch(Exception e){

            }
            if((i+1)>=codearr.length){
                codearr[i]=
Long.parseLong(code1.substring(untergrenze+1,obergrenze+1));
            }

            untergrenze=obergrenze;

            c++;
            i++;
        }
        ////////////////////
        //jede Zelle des Arrays einzeln entschlüsseln
        ////////////////////
        i=0;
        while(i<codearr.length){
            output=output+entschlüsseln(codearr[i]);

            i++;
        }


        return alpha.Zahl_zu_Text_converter(output);
    }
//////////////////////////////
```

Literaturverzeichnis

[AF] A. Filler: Einführung in die Gruppentheorie(Skript zur Vorlesung “Algebra 2“); Heidelberg 2007

[ATM] Thomas Markwig: Algebraische Strukturen(Skript zur Vorlesung); Kaiserslautern 2008

[CM] Tim Cole und Michael Matzer: Managementaufgabe Sicherheit; Hanser Verlag München; Wien 2001

[DA] Daniel Amor: Die E-Buisness-(R)Evolution (das umfassende Executive-Briefing); Galileo Press, Bonn 2000 1. Auflage

[DF] Prof. Dr. Dirk Ferus: Lineare Algebra 1 (Skript zur Vorlesung); Berlin 2001

[DL] Anonymus: Kryptogafische Systeme auf Basis des diskreten Logarithmus; Dresden. http://www.inf.tu-dresden.de/content/institutes/sya/dud/lectures/sommersemester/KPDatensicherheit/v07_doku.pdf (04.03.2015)

[EC] Ingo Grebe: Elliptische Kurven in der Kryptografie, Potsdam 2005. http://www.cs.uni-potsdam.de/ti/lehre/05-Kryptographie/slides/Elliptische_Kurven.pdf (04.03.2015)

[EA]Prof.Dr. Sven Rahmann: Elegante Algorithmen; Dortmund 2009 .http://ls11-www.cs.uni-dortmund.de/people/rahmann/teaching/ss2008/EleganteAlgorithmen/skript.pdf (04.03.2015)

[EK] Thomas Laubrock: Krypto-Verfahren basierend auf elliptischen Kurven; Dortmund 1999. http://www.elliptische-kurven.de/ (04.03.2015)

[EKL] **Freiermuth**, K., **Hromkovič**, J., **Keller**, L., **Steffen**, B.: Einführung in die Kryptologie: Lehrbuch für Unterricht und Selbststudium; Wiesbaden 2010

[GB] Günter Bärwolff: Höhere Mathematik für Naturwissenschaftler und Ingenieure 2. Auflage, 1. Korrigierter Nachdruck; Berlin, Heidelberg 2009

[HC] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanston: Handbook of applied cryptogaphy; Boca Raton 1996

[IB] Prof. Dr. Irene I. Bouw: Elementare Zahlentheorie (Skript zur Vorlesung); Ulm 2008

[JV] Dr. Jörg Vogel: Kryptologie von einer Geheimwissenschaft zu einer Wissenschaft von den Geheimnissen (Skript zur Vorlesung); Jena 2006

[MR] Prof. Dr. Helmut Maier, Peter Reck: Angewandte diskrete Mathematik (Skript zur Vorlesung); Ulm 2009

[MS] Michael Stoll: Arithmetik elliptischer Kurven mit Anwendungen (Skript zur Vorlesung); Bayreuth 2009

[PP] Peter Pfaffelhuber: Statistik(Skript zur Vorlesung); Freiburg 2010

[PW] WDR/SWR/ARD-alpha Franziska Badenschier: Kyptogafie leicht gemacht-schreiben sie in Geheimschrift; 2013. http://www.planetwissen.de/natur_technik/forschungszweige/kryptologie/kryptologie_anleitungen.jsp (04.03.2015)

[QG]Alf Krause: Lösung Kubischer Gleichungen; Freiberg.http://www.mathe.tu-freiberg.de/~hebisch/seminar1/kubik.html (04.03.2015)

[REK] Stefan Linke, M. Serhat Cinar: Elliptische Kurven und ihre Anwendung in der Verschlüsselung; Köln 2003. http://www.graviton.de/ai/algoan/referate/elliptische%20kurven.pdf (04.03.2015)

[SS]Simon Singh: Fermats letzter Satz: Die abenteuerliche Geschichte eines mathematischen Rätsels

[SV] Siegfried Spolwig: Symmetrisches Verschlüsselungssystem Nachrichtenübermittelung; Berlin 2005. http://oszhdl.be.schule.de/gymnasium/faecher/informatik/krypto/symmetrisch.htm (04.03.2015)

[WD] Anonymus: Diffie-Hellman-Schlüsselaustausch; http://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch (04.03.2015)

[WM] Anonymus: Mengenlehre; http://de.wikipedia.org/wiki/Mengenlehre (04.03.2015)

[WP] Anonymus: Polyalphabetische Substitution; http://de.wikipedia.org/wiki/Polyalphabetische_Substitution (04.03.2015)

[WR] Anonymus: Ring (Algebra); http://de.wikipedia.org/wiki/Ring_%28Algebra%29 (04.03.2015)

[ZTM] Thomas Marwig: elementare Zahlentheorie(Skript zur Vorlesung); Kaiserslautern 2008

Abbildungsquellen

[1]Ditmar Lammers: Diskrete Strukturen Kap 5 Zahlentheorie; Münster 2009. http://cs.uni-muenster.de/u/lammers/EDU/ss09/DiskreteStrukturen/Script/Kap5%20-%20Zahlentheorie%20+%20Arithmetik.mm.html (05.03.2015)

[2]http://www.wolframalpha.com/input/?i=F%28x%2Cy%29%3Dx^3-x%2B1-y^2 (05.03.2015)

[3]Nick Sullivan: A (relatively easy to understand) primer on elliptic curve cryptography; 2013. http://arstechnica.com/security/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/ (05.03.2015)

Computerprogramme

Eclipse Luna (Java-Entwicklungsumgebung), Bluej (Java-Entwicklungsumgebung)

Geogebra (dynamische Mathematiksoftware(einfache Geometrie, Algebra und Analysis))

Blender 2.70a (professionelle 3d-Grafiksoftware)

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Benedikt Buller Warendorf 05.03.2015