MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Faktorisierungsalgorithmen natürlicher Zahlen“

verfasst von / submitted by

Dominik Ballwein, BEd

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Education (MEd)

Wien, 2023 / Vienna, 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 199 500 520 02
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Lehramt Sek (AB) UF Bewegung und Sport UF Mathematik
Betreut von / Supervisor:	ao. Univ.-Prof. Dr. Joachim Mahnkopf

Zusammenfassung

Bereits Carl Friedrich Gauß, einer der bedeutendsten Mathematiker seiner Zeit, bezeichnete die Zahlentheorie als die *Königin der Mathematik*. Lange jedoch galt die Zahlentheorie zwar als eine der interessantesten Disziplinen der Mathematik, jedoch auch als eine der anwendungsärmsten, bezogen auf deren praktische Relevanz im Alltag. Mit dem Aufkommen neuer Verschlüsselungsverfahren, der sogenannten Public-Key-Kryptografie, hat sich diesbezüglich einiges getan. Insbesondere das RSA-Verfahren, welches heutzutage für Bereiche wie Kartenzahlung, E-Mail-Verschlüsselung oder der Internet- und Telefon-Infrastruktur herangezogen wird, baut auf einem der Grundprobleme der algorithmischen Zahlentheorie auf, nämlich der Schwierigkeit der Faktorisierung natürlicher Zahlen. Bis heute ist kein Algorithmus effizient genug um dieses Problem für sehr große Zahlen in angemessener Zeit zu lösen. Nichtsdestotrotz forschen viele Personen an dieser Problematik und mit der Zeit wurden einige Faktorisierungsalgorithmen entwickelt, welche in Bezug auf den RSA-Algorithmus bei einer ungünstigen Wahl oder zu geringer Größe der Faktoren des RSA-Moduls erfolgreich sind. Die vorliegende Masterarbeit befasst sich im Hinblick auf das RSA-Verfahren mit den wichtigsten Faktorisierungsalgorithmen und deren Funktionsweisen.

Abstract

Carl Friedrich Gauß, one of the most important mathematicians of his time, already called number theory the *Queen of mathematics*. While being considered one of the most interesting disciplines of mathematics, it has also been regarded as one of the most irrelevant in terms of its usage in everyday life. With the advent of new encryption methods, the so-called Public-key cryptography, a lot has changed in this respect. In particular, the RSA method, which is used today for areas such as card payments, e-mail encryption, or the Internet and telephone infrastructure, builds on one of the basic problems of algorithmic number theory, namely the factorization of natural numbers. To this day, no algorithm is efficient enough to solve this problem for large numbers in a reasonable amount of time. Nevertheless, many people are doing research on this problem and, over time, some factorization algorithms have been developed which succeed in the course of the RSA method when the factors of the RSA-modulus are improperly or too small. With respect to the RSA method, this master thesis deals with the most important factorization algorithms and how they work.

Inhaltsverzeichnis

1 Einleitung

Der berühmte Mathematiker Carl Friedrich Gauß schrieb im Jahre 1889 in seiner Disquisitiones Arithmeticae:

> „Dass die Aufgabe, die Primzahlen von den zusammengesetzten zu unterscheiden und letztere in ihre Primfactoren zu zerlegen zu den wichtigsten und nützlichsten der gesamten Arithmetik gehört und die Bemühungen und den Scharfsinn sowohl der alten wie auch der neueren Geometer in Anspruch genommen hat, ist so bekannt, dass es überflüssig wäre, hierüber viele Worte zu verlieren. ...; ausserdem aber dürfte es die Würde der Wissenschaft erheischen, alle Hülfsmittel zur Lösung jenes so eleganten und berühmten Problems fleissig zu vervollkommnen.[sic] "([10], S. v)

Bereits bevor es die *Public-Key-Kryptografie* gab, wusste Gauss schon, wie wichtig es ist, zusammengesetzte Zahlen von Primzahlen zu unterscheiden und Erstere in ihre Primfaktoren zu zerlegen. Spätestens nach der Entwicklung des RSA-Verfahrens wusste man, welche Bedeutung das damalige Zitat hatte. Wäre man heutzutage in der Lage große Zahlen effizient in ihre Faktoren zu zerlegen, also zu faktorisieren, so wäre das RSA-Verfahren nicht mehr sicher. Dies würde große Auswirkungen mit sich bringen, da das RSA-Verfahren für diverse alltägliche Bereiche zur Verschlüsselung von Information herangezogen wird. Dessen ungeachtet gibt es trotzdem diverse Faktorisierungsalgorithmen, welche bei einer ungeeigneten Wahl der Faktoren p und q des RSA-Moduls N erfolgreich sein können. Auch wenn man vermutlich nie in der Lage sein wird das Faktorisierungsproblem zu lösen, sind kleine Fortschritte erreicht worden. Die vorliegende Arbeit beschäftigt sich mit diversen Faktorisierungsalgorithmen natürlicher Zahlen. Im zweiten Kapitel werden als Vorbereitung verschiedene mathematische Grundlagen angeführt, um die dahintersteckende Mathematik der jeweiligen Algorithmen nachvollziehen zu können. In Kapitel 3 wird die Funktionsweise des RSA-Verfahrens erläutert, da dieses eine der wichtigsten Anwendungsmöglichkeiten der Faktorisierungsproblematik darstellt. Das vierte Kapitel ist das Zentralste, da in diesem eine Auswahl an Faktorisierungsalgorithmen angegeben, erklärt und anhand von Beispielen untermauert wird. Nach einem Fazit in Kapitel 5 über die angeführten Algorithmen und deren Auswirkungen auf das RSA-Verfahren wird in Kapitel 6 noch ein Ausblick auf zukünftige Verfahren und Technologien gegeben.

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mir bei der Erstellung meiner Masterarbeit stets zur Seite gestanden sind. Zuerst gebührt mein Dank Herrn Prof. Dr. Joachim Mahnkopf, der mir von Beginn an durch konstruktive Kritik, diverse Hilfestellungen und nützliche Anregungen das Schreiben dieser Arbeit enorm erleichtert hat. Darüber hinaus möchte ich mich bei meiner Frau und meiner Schwester für das zahlreiche Korrekturlesen und die damit verbundene Geduld bedanken. Zu guter Letzt danke ich meinen Eltern, die mir

mein Studium durch jahrelange Unterstützung ermöglicht haben und die auch in schwierigen Zeiten immer für mich da waren.

2 Mathematische Grundlagen

Um die notwendige Mathematik für die einzelnen Faktorisierungsalgorithmen zu verstehen, werden in diesem Abschnitt diverse mathematische Grundlagen angeführt und mit Beispielen untermauert. Diese gliedern sich in zahlentheoretische, gruppentheoretische und informatische Grundlagen. Darüber hinaus werden auch diverse Grundlagen zu speziellen algebraischen Kurven, nämlich der elliptischen Kurven, angegeben. Dabei wurden lediglich kürzere und für diese Arbeit essentielle Beweise angeführt.

2.1 Zahlentheorie

Die zahlentheoretischen Grundlagen richten sich nach den Quellen ([3, 1, 8, 10, 5, 13, 7]). Für Sätze, welche im Rahmen dieses Abschnitts nicht bewiesen wurden, siehe ([3, 10]).

Definition. Die Menge der natürlichen Zahlen $\mathbb{N}$ wird wie folgt definiert:

$$\mathbb{N} := \{0, 1, 2, 3, 4, \ldots\}$$

Je nach ausgewählter Literatur ist die Zahl 0 ein Element dieser Menge oder nicht. Im Kontext dieser Arbeit, ist die 0 das erste Element der natürlichen Zahlen.

Definition. Die Menge der ganzen Zahlen $\mathbb{Z}$ beinhaltet alle natürlichen Zahlen und zusätzlich zu jeder natürlichen Zahl m außer der 0 eine negative Zahl $-m$, sodass $m + (-m) = 0$

$$\mathbb{Z} := \{\ldots, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 \ldots\}$$

Definition. (**Teilbarkeit**) Seien $a, b \in \mathbb{Z}$. Man sagt a teilt b, wenn es eine ganze Zahl d gibt mit $b = ad$. Dann heißt a Teiler von b und man schreibt $a \mid b$. Man kann auch sagen b ist durch a teilbar oder b ist Vielfaches von a. Wenn $a = bd$, so wird d als Komplementärteiler von a zu b bezeichnet. Falls a kein Teiler von b ist, dann schreibt man $a \nmid b$.

Beispiel. Es gilt $13 \mid 97$, weil $97 = 13 \cdot 7$. Genauso gilt $-4 \mid 28$, weil $28 = (-4) \cdot (-7)$. Die Menge aller Teiler der Zahl 12 besteht aus den Teilern $\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 12$.

Bemerkung. Jede natürliche Zahl a besitzt die folgenden Teiler: $-a, -1, 1, a$. Man nennt diese Teiler auch triviale Teiler. Falls $d \mid a$ und $d \notin \{-a, -1, 1, a, \}$, so nennt man d auch echten Teiler von a.

Definition. (**Gaussklammer**) Man nennt $\lfloor y \rfloor := \max\{n \in \mathbb{Z} \mid n \leq y\}$ die untere Gaußklammer für ein $y \in \mathbb{R}$. Dabei bezeichnet $\lfloor y \rfloor$ die größte ganze Zahl $\leq y$. Analog dazu wird

$\lceil y \rceil := \min\{n \in \mathbb{Z} \mid n \geq y\}$ als obere Gaussklammer definiert, wobei $\lceil y \rceil$ die kleinste ganze Zahl $\geq y$ bezeichnet.

Beispiel. Für $y = 5,6$ ist die unter Gaussklammer $\lfloor y \rfloor = 5$ und die obere Gaussklammer $\lceil y \rceil = 6$

Satz 2.1 (**Division mit Rest**). *Sind l, m ganze Zahlen mit $m > 0$, so existieren eindeutig bestimmte ganze Zahlen q und r, sodass $l = qm + r$ und $0 \leq r < m$. Dabei gilt $q = \lfloor \frac{l}{m} \rfloor$ und $r = l - qm$.*

Beweis. Existenz: Sei $q := \lfloor \frac{l}{m} \rfloor$. Somit ist $q \leq \frac{l}{m} < q+1 \Rightarrow qm \leq l < qm+m \Rightarrow 0 \leq l-qm < m$. Setze $r := l - qm$. Daher ist $l = qm + r$ und $0 \leq r < m$.
Eindeutigkeit: Angenommen es gäbe zwei verschiedene Darstellungen für l, nämlich $l = qm + r = q'm + r'$ mit $0 \leq r, r' < m$. Daher gilt $(q - q')m = r' - r \Rightarrow q - q' = \frac{r'-r}{m}$.
Da $-m < r' - r < m \Rightarrow -1 < \underbrace{\frac{r' - r}{m}}_{\epsilon \mathbb{Z}} < 1 \Rightarrow \frac{r'-r}{m} = 0 \Rightarrow r' = r \Rightarrow qm = q'm \Rightarrow q = q'$. □

Definition. (**Gemeinsames Vielfaches**) Seien $m_1, \ldots, m_k \in \mathbb{Z}$, so nennt man $n \in \mathbb{Z}$ gemeinsames Vielfaches von $m_1, \ldots, m_k$ falls $m_i \mid n$ (für $1 \leq i \leq k$).

Bemerkung. Existiert ein $i \in \{1, \ldots, k\} : m_i = 0$, dann ist die Menge der positiven gemeinsamen Vielfachen leer, weil nur die 0 das einzige gemeinsame Vielfache von $m_1, \ldots, m_k$ sein kann und nach unten beschränkt. Seien $m_1, \ldots, m_k \in \mathbb{Z}\backslash\{0\}$, dann ist die Menge der positiven gemeinsamen Vielfachen nicht leer, weil diese $|m_1 \cdots m_k| = |m_1| \cdots |m_k|$ enthält und mit der Zahl 1 nach unten beschränkt.

Definition. (**Kleinstes gemeinsames Vielfaches**) Seien $m_1, \ldots, m_k \in \mathbb{Z}\backslash\{0\}$, so wird

$$\text{kgV}\,(m_1, \ldots, m_k) = \min\{n \in \mathbb{N} : m_i \mid n \text{ für } 1 \leq i \leq k\}$$

als kleinstes gemeinsames Vielfaches bezeichnet, wobei als Minimum das kleinste Element der angeführten Menge verstanden wird.

Definition. (**Gemeinsamer Teiler**) Seien $m_1, \ldots, m_k \in \mathbb{Z}$, dann nennt man $d \in \mathbb{Z}$ gemeinsamer Teiler von $m_1, \ldots, m_k$ falls $d \mid m_i$ für $1 \leq i \leq k$.

Bemerkung. Seien $m_1 = \ldots = m_k = 0$, dann ergibt sich für die Menge der positiven gemeinsamen Teiler von $m_1, \ldots, m_k$ gleich $\mathbb{N}$. Wenn für $m_1 = \ldots = m_k$ nicht alle $= 0$ gilt, dann ist die Menge der positiven gemeinsamen Teiler von $m_1 = \ldots = m_k$ nicht leer, weil diese Menge stets die Zahl 1 enthält und auch nach oben beschränkt (z.B. durch $\min\{|m_i| : 1 \leq i \leq k, m_i \neq 0\}$). Daher muss auch ein größter gemeinsamer Teiler existieren, welcher im folgenden definiert wird.

Definition. (**Größter gemeinsamer Teiler**) Seien m, n zwei ganze Zahlen. Ein Element $d \in \mathbb{Z}$ wird größter gemeinsamer Teiler(ggT oder gcd für **g**reatest **c**ommon **d**ivisor) von m und n genannt, wenn folgende Bedingungen erfüllt sind:
i) $d \mid m$ und $d \mid n$
ii) Ist $d' \in \mathbb{Z}$ ein weiteres Element mit $d' \mid m$ und $d' \mid n$, so folgt $d' \mid d$.
Zwei Elemente $m, n \in \mathbb{Z}$ heißen teilerfremd oder relativ prim, falls die Zahl 1 größter gemeinsamer Teiler von m, n ist.

In der vorliegenden Arbeit wird für den größten gemeinsamen Teiler die Abkürzung *gcd* verwendet, da dieser Begriff in den Programmcodes von *ARIBAS* benutzt wird.

Definition. (**Primzahl**) Primzahl wird eine natürliche Zahl $p > 1$ genannt, wenn diese nur die trivialen Teiler $-p, -1, 1, p$ besitzt.

In den kommenden Abschnitten betrachten wir hauptsächlich die positiven Teiler einer Zahl, womit eine Primzahl p definitionsgemäß nur die trivialen Teiler p und 1 besitzt.

Satz 2.2 (**Primfaktorzerlegung**). *Sei $m \in \mathbb{N}, m > 1$, so kann man m als Produkt von (nicht notwendigerweise verschiedenen) Primzahlen darstellen. Also gibt es Primzahlen $p_1, \ldots, p_n$, sodass $m = p_1 \cdots p_n$. Bis auf die Reihenfolge der Primzahlen ist diese Zerlegung eindeutig.*

Beweis. Existenz: Angenommen die Aussage wäre falsch und es gäbe Zahlen die nicht in ihre Primfaktoren zerlegt werden können. Sei m die kleinste Zahl mit dieser Eigenschaft. Die Zahl m kann keine Primzahl sein, da sonst $m = p$ gelten würde, womit m ein triviales Produkt von Primzahlen wäre. Daher muss es eine geeignete Zahl a geben, welche m teilt. Das heißt für bestimmte Zahlen $a, b < m$ gilt $m = ab$. Da a, b kleiner als m sind, existiert nach Voraussetzung für diese beiden Zahlen eine Primfaktorzerlegung mit $a = p_1 \cdots p_k$ und $b = p_{k+1} \cdots p_n$. Infolgedessen gilt:

$$m = ab = p_1 \cdots p_k \cdot p_{k+1} \cdots p_n$$

wodurch ein Widerspruch zur Annahme entsteht.
Eindeutigkeit: Angenommen, es existieren natürliche Zahlen mit zwei unterschiedlichen Darstellungen als Produkt von Primzahlen. Sei $m = p_1 \cdots p_l = q_1 \cdots q_k$ die kleinste Zahl mit zwei verschiedenen Darstellungen. Daher gilt: $p_l \,|(q_1 \cdots q_k) \Rightarrow \exists i \in \{1, \ldots, k\} : p_l | \, q_i$. O.B.d.A. sei $i = k$, das heißt $p_l \mid q_k \Rightarrow p_l = q_k \Rightarrow \frac{m}{p_l} = p_1 \cdots p_{l-1} = q_1 \cdots q_{k-1} < m$. Dies bedeutet, dass $\frac{m}{p_l}$ ebenfalls zwei verschiedene Darstellungen als Produkt besitzen würde, was einen Widerspruch zur Minimalitat von m bedeuten würde. □

Dieser wichtige Satz wird in der Literatur als *Fundamentalsatz der Arithmetik* oder auch als *elementarer Satz der Zahlentheorie* bezeichnet. Bis jetzt ist bekannt, dass eine Zahl entweder

eine Primzahl ist oder eine aus mehreren Primzahlen, sogenannten Primfaktoren, zusammengesetzte Zahl ist. Bleibt noch zu klären wie größere Primzahlen bis zu einer gewissen Schranke N zu bestimmen sind. Dazu eignet sich ein Algorithmus, der auch das *Sieb des Eratosthenes* genannt wird.

Sieb des Eratosthenes
Zu Beginn werden alle Zahlen von 2 bis zur Schranke N aufgeschrieben. Streicht man alle Vielfachen mp (für $m > 1$) aller Primzahlen $p \leq \sqrt{N}$ so erhält man alle Primzahlen $\leq N$. Dieser Algorithmus funktioniert, weil durch das Streichen aller Primzahlen p und deren Vielfachen, welche kleiner oder gleich einer bestimmten Primzahl p_i für ein $i \in \mathbb{N}$ sind, die nächste darauffolgende und nicht gestrichene Zahl p_{i+1} eine Primzahl sein muss. Dies begründet sich darin, dass p_{i+1} kein Vielfaches von kleineren Zahlen als sie selbst ist und somit nur die trivialen Teiler 1 und p_{i+1} selbst in Frage kommen, was genau der Definition einer Primzahl entspricht. Es reicht dabei nur Zahlen kleiner als $\sqrt{N}$ zu betrachten, denn mindestens ein Primfaktor der Schranke N muss kleiner oder gleich $\sqrt{N}$ sein, womit diese Zahlen bereits gestrichen wären.
Als Beispiel werden alle Primzahlen bis $N = 50$ mit dem Sieb des Eratosthenes bestimmt. Da $\lfloor\sqrt{N}\rfloor = 7$ reicht es alle Vielfachen der Primzahlen ≤ 7 zu streichen. Zur Vereinfachung des Algorithmus werden die Zahlen in Tabellenform mit Zeilen zu je 6 Zahlen geschrieben. Dadurch ist es möglich alle Vielfachen der Zahlen $2, 3$ durch Streichen der jeweiligen Spalten und die Vielfachen der Zahlen $5, 7$ mittels diagonalem Streichen zu eliminieren.

	2	3	~~4~~	5	~~6~~
7	~~8~~	~~9~~	1~~0~~	11	1~~2~~
13	1~~4~~	1~~5~~	1~~6~~	17	1~~8~~
19	2~~0~~	2~~1~~	2~~2~~	23	2~~4~~
2~~5~~	2~~6~~	2~~7~~	2~~8~~	29	3~~0~~
31	3~~2~~	3~~3~~	3~~4~~	3~~5~~	3~~6~~
37	3~~8~~	3~~9~~	4~~0~~	41	4~~2~~
43	4~~4~~	4~~5~~	4~~6~~	47	4~~8~~
4~~9~~	5~~0~~				

Die ersten Primzahlen ≤ 50 lauten also: $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47$.
Das Sieb des Eratosthenes ist eine praktische Methode um relativ kleine Primzahlen zu bestimmen, jedoch ist es für sehr große Zahlen wenig geeignet. Es bleibt noch zu klären, ob es unendlich viele Primzahlen gibt.

Satz 2.3 (**Euklid**)**.** *Es gibt unendlich viele Primzahlen.*

Beweis. Es wird indirekt angenommen, dass es nur endlich viele Primzahlen gibt, wir bezeichnen diese als $p_1, p_2, \ldots, p_k$. Sei $N = \prod_{i=1}^{n} p_i + 1$. Somit ist N größer als jede Primzahl und muss

daher eine zusammengesetzte Zahl sein. Daher besitzt N einen Teiler d mit $1 < d < N$. Es wird das kleinste derartige d gewählt. Nun wird behauptet, dass d eine Primzahl sein muss. Falls d keine Primzahl wäre, hätte d einen echten Teiler $1 < a < d$, womit dieser dann auch ein Teiler von N wäre, was einen Widerspruch zur Minimalität von d darstellt. Deswegen muss der Teiler d von N gleich einem p_j für ein $j \in \{1, \ldots, k\}$ sein. Würde jedoch p_j das Produkt $\prod_{i=1}^{n} p_i$ und $N = \prod_{i=1}^{n} p_i + 1$ teilen, dann aber auch deren Differenz $\prod_{i=1}^{n} p_i + 1 - \prod_{i=1}^{n} p_i = 1$. Da $p_j \nmid 1$ gilt, führt dies zum Widerspruch, wodurch die Existenz von unendlich vielen Primzahlen geschlussfolgert wird. □

Für die Berechnung des größten gemeinsamen Teilers wird der rund 2000 Jahre alte *Euklidisiche Algorithmus* herangezogen.

Satz 2.4 (**Euklidischer Algorithmus**). *Sind $a, b \in \mathbb{N}$ und o.B.d.A. sei $a \geq b$. So existieren nach Satz 2.1 eindeutig bestimmte Zahlen $r_1 > r_2 > \ldots > r_n > r_{n+1} = 0$ und $q_0, q_1, \ldots, q_n \in \mathbb{N}$ mit*

$$\begin{aligned} a &= q_0 b + r_1 && 0 < r_1 < b \\ b &= q_1 r_1 + r_2 && 0 < r_2 < r_1 \\ &\vdots && \vdots \\ r_{n-2} &= q_{n-1} r_{n-1} + r_n && 0 < r_n < r_{n-1} \\ r_{n-1} &= q_n r_n + r_{n+1} && 0 = r_{n+1} \end{aligned}$$

Nach einer endlichen Anzahl an Schritten bricht der Algorithmus ab und man erhält den größten gemeinsamen Teiler durch $gcd\,(a, b) = r_n$.

Beweis. Durch die erste Gleichung folgt $gcd\,(a, b) = gcd\,(b, r_1)$, aus der zweiten $gcd\,(b, r_1) = gcd\,(r_1, r_2)$, usw. Schlussendlich folgt aus der letzten Gleichung

$$gcd\,(r_{n-1}, r_n) = gcd\,(r_n, r_{n+1}) = r_n$$

□

Beispiel. Als Beispiel soll der $gcd(142, 30)$ bestimmt werden.

$$\begin{aligned} 142 &= 4 \cdot 30 + 22 \\ 30 &= 1 \cdot 22 + 8 \\ 22 &= 2 \cdot 8 + 6 \\ 8 &= 1 \cdot 6 + \mathbf{2} \\ 6 &= 3 \cdot 2 + 0 \end{aligned}$$

Somit ist der $gcd(142, 30) = 2$.

Lemma 2.5 (**Lemma von Bézout**). *Zu $a, b \in \mathbb{N}$ existieren ganze Zahlen x und y mit $gcd\,(a, b) = x \cdot a + y \cdot b$*

Beweis. Folgt sofort durch den Euklidischen Algorithmus, wobei man mittels zurückrechnen die Koeffizienten x, y erhält. □

Beispiel. Mithilfe des Lemmas von Bézout wird der oben berechnete $gcd(142, 30) = 2$ als Linearkombination der Zahlen 142 und 30 dargestellt.

$$\begin{aligned}
&2 = 8 - 1 \cdot 6 \\
&2 = 8 - 1 \cdot (22 - 2 \cdot 8) = -22 + 3 \cdot 8 \\
&2 = -22 + 3 \cdot (30 - 1 \cdot 22) = 3 \cdot 30 - 4 \cdot 22 \\
&2 = 3 \cdot 30 - 4 \cdot (142 - 4 \cdot 30) = -4 \cdot 142 + 19 \cdot 30 \\
&\Rightarrow x = -4, \quad y = 19
\end{aligned}$$

Somit wird der größte gemeinsame Teiler von a und b als Linearkombination der Zahlen a und b dargestellt. Man nennt diese Darstellung auch *Vielfachsummendarstellung*. Der Euklidische Algorithmus und das Lemma von Bézout zusammen werden auch der *Erweiterte Euklidische Algorithmus* genannt. Bei den folgenden Sätzen werden nur die Wichtigsten bewiesen.

Definition. (**Kongruenzen**) Seien m, n ganze Zahlen und $m \in \mathbb{N}$. Gilt $m \mid (a - b)$, sagt man a und b sind kongruent modulo m oder man schreibt $a \equiv b$ (mod m) oder kürzer $a \equiv b$ (m), wobei mit m der Modul bezeichnet wird. $(a - b)$ ist dabei ein ganzzahliges Vielfaches von m. Gilt $m \nmid (a - b)$, so sind a und b inkongruent modulo m und man schreibt dafür $a \not\equiv b$ (mod m) oder wieder kurz $a \not\equiv b$ (m).

Bemerkung. Gilt $a \equiv b$ (m) so haben die Zahlen a und b bei Division durch den Modul m denselben Rest.

Beispiel. Es gilt: $5 \equiv 13$ (mod 4), da $4 \mid (5 - 13)$ und $-1 \equiv 9$ (mod 5), da $5 \mid ((-1) - 9)$

Lemma 2.6. *Die Relation kongruent modulo m ($\equiv$ mod m) ist eine Äquivalenzrelation.*

Definition. (**Restklassen modulo m**) Sei $m \in \mathbb{N}$. Man nennt jede Äquivalenzklasse $\overline{a}$ modulo m auch Resklasse modulo m. Jedes Element $x \in \overline{a}$ ist ein Repräsentant der Restklasse $\overline{a}$. Als Menge aller Restklassen bezeichnet man folgende Menge

$$\mathbb{Z}/\mathrm{m}\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{m-1}\}$$

wobei unter $\overline{a} \in \mathbb{Z}/\mathrm{m}\mathbb{Z}$ alle Zahlen verstanden werden, welche mod m denselben Rest haben.

Lemma 2.7. *Für alle $m \in \mathbb{N}$ besteht die Menge $\mathbb{Z}/\mathrm{m}\mathbb{Z}$ aus genau m verschiedenen Elementen, nämlich:*

$$\mathbb{Z}/\mathrm{m}\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{m-1}\}$$

Beweis. Sei a eine ganze Zahl und es gelte $a = qm + r$, $0 \leq r \leq m-1$. Daher ist $a \equiv r \pmod{\mathrm{m}} \Rightarrow \bar{a} = \bar{r}$. Dies bedeutet, dass es höchstens die Restklassen $\overline{0}, \overline{1}, \ldots, \overline{m-1}$ geben kann. Diese Restklassen sind alle verschieden, weil $r \not\equiv t \pmod{\mathrm{m}}$ $\forall r, t \in \{0, 1, \ldots, m-1\}$. □

Die Menge $\mathbb{Z}/\mathrm{m}\mathbb{Z}$ wird in der Literatur oftmals auch als $\mathbb{Z}_m$ bezeichnet. In der vorliegenden Arbeit wird ausschließlich $\mathbb{Z}/\mathrm{m}\mathbb{Z}$ verwendet, um die Menge aller Restklassen anzugeben.

Definition. Die Addition und Multplikation in $\mathbb{Z}/m\mathbb{Z}$ wird wie folgt definiert:

$$\overline{a} + \overline{b} := \overline{a+b}, \quad \overline{a} \cdot \overline{b} := \overline{ab}$$

Satz 2.8. *Die Addition und die Multiplikation in $\mathbb{Z}/m\mathbb{Z}$ sind wohldefiniert, das heißt unabhängig von der Wahl des Repräsentanten. Seien $a \equiv a' \pmod{m}$ und $b \equiv b' \pmod{m}$. Daraus folgt, dass $a + b \equiv a' + b' \pmod{m}$ und $ab \equiv a'b' \pmod{m}$*

Beweis. Da $a \equiv a' \pmod{\mathrm{m}}$ gilt $m \mid (a - a')$ und es existiert eine ganze Zahl x mit $a' = a + xm$. Aus $b \equiv b' \pmod{\mathrm{m}}$ folgt analog, dass es eine ganze Zahl y gibt, mit $b' = b + ym$. Es folgt:

$$a'b' = (a + xm)(b + ym) = ab + (ay + bx + xym)m$$

Somit ist $a'b' \equiv ab \pmod{\mathrm{m}}$, womit die Wohldefiniertheit der Multiplikation bewiesen wäre. Der Beweis für die Addition funktioniert analog. □

Beispiel. In $\mathbb{Z}/7\mathbb{Z}$ ist:

$$\overline{3} + \overline{4} = \overline{7} = \overline{0}$$

Ist beispielsweise $10 \equiv 3 \pmod{7}$ und $11 \equiv 4 \pmod{7}$, so ist $10 + 11 = 21 \equiv 0 \pmod{7}$.

Satz 2.9. *Seien a, b ganze Zahlen und s, m natürliche Zahlen mit $s, m > 0$, dann gilt:*

(i) $a \equiv b(\mathrm{mod} m)$ *und* $s \mid m \Rightarrow a \equiv b(\mathrm{mod}|s|)$

(ii) $a \equiv b(\mathrm{mod} m) \Rightarrow sa \equiv sb(\mathrm{mod}(|s|m))$

(iii) $sa \equiv sb(\mathrm{mod} m) \Leftrightarrow a \equiv b \left(\mathrm{mod} \frac{m}{\mathrm{gcd}(s,m)}\right)$

(vi) $a \equiv b(\mathrm{mod} m)$ *und* $a \equiv b(\mathrm{mod} n) \Rightarrow a \equiv b(\mathrm{mod}\, \mathrm{kgV}(m, n))$

(v) $\forall a_1, \dots a_n \in \mathbb{Z}; b_1, \dots, b_n \in \mathbb{Z}$: *Ist* $a_i \equiv b_i (\mathrm{mod} m)$ *für* $1 \leq i \leq n$, *so*

$$\sum_{i=1}^{n} a_i \equiv \sum_{i=1}^{n} b_i \pmod{m} \quad \textit{und} \quad \prod_{i=1}^{n} a_i \equiv \prod_{i=1}^{n} b_i (\mathrm{mod} m)$$

Im Besonderen gilt für alle ganzen Zahlen a, b *und alle natürlichen Zahlen* $k > 0$*:*
Wenn $a \equiv b (\mathrm{mod} m)$, *dann* $a^k \equiv b^k (\mathrm{mod} m)$

Korollar 2.10. *Sei* $m > 0$ *eine natürliche Zahl mit Primfaktorzerlegung* $m = p_1^{\alpha_1} \dots p_n^{\alpha_n}$, *wobei* $p_1, \dots, p_n$ *paarweise verschiedene Primzahlen sind mit Exponenten* $\alpha_1, \dots, \alpha_n \in \mathbb{N}$. *Für zwei ganze Zahlen* a *und* b *sind äquivalent:*

(i) $a \equiv b (\mathrm{mod} m)$

(ii) $a \equiv b \, (\mathrm{mod} p_i^{\alpha_i})$ *für* $1 \leq i \leq n$

Definition. (**Lineare Kongruenz)** Sei $m > 0$ eine natürliche Zahl und a, b sind ganze Zahlen. Man nennt $ax \equiv b (\mathrm{mod} m)$ lineare Kongruenz. Gesucht sind dabei alle $x \in \mathbb{Z}$, welche diese Kongruenz erfüllen.

Bemerkung. Wenn x_0 eine Lösung dieser linearen Kongruenz ist und $x_1 \equiv x_0 (\mathrm{mod} m)$, dann ist x_1 auch eine Lösung, da $ax_1 \equiv ax_0 \equiv b (\mathrm{mod} m)$. Daher versucht man nur die modulo m inkongruenten Lösungen zu finden.

Satz 2.11. *Ist* $m > 0$ *eine natürliche Zahl und* a, b *sind ganze Zahlen, dann ist die lineare Kongruenz* $ax \equiv b (\mathrm{mod} m)$ *genau dann lösbar, falls* $gcd(a, m) \mid b$. *Wenn* $gcd(a, m) \mid b$, *dann existieren genau* $gcd(a, m)$ *modulo* m *inkongruente Lösungen.*

Korollar 2.12. *Sei* a *eine ganze Zahl und* $m > 1$ *eine natürliche Zahl. Wenn* $gcd(a, m) = 1$, *so gibt es modulo* m *ein eindeutig bestimmtes* $x \in \mathbb{Z}$, *sodass* $ax \equiv 1 (\mathrm{mod} m)$.

Definition. (**Simultane lineare Kongruenz)** Seien $m_1, \dots, m_n$ natürliche Zahlen größer 0 und $c_1, \dots, c_n$ ganze Zahlen. Unter einer linearen simultanen Kongruenz versteht man folgendes System

$$x \equiv c_1 \pmod{m_1}, x \equiv c_2 \pmod{m_2}, \dots, x \equiv c_n \pmod{m_n}$$

Es werden dabei alle $x \in \mathbb{Z}$ gesucht, welche alle n Kongruenzen erfüllen.

Bemerkung. Beim Chinesischem Restsatz (s.u.) wird vorausgesetzt, dass die Module $m_1, \dots, m_n$ paarweise teilerfremd sind, da es sonst sein könnte, dass die simultane lineare Kongruenz nicht lösbar ist. Ist beispielsweise $x \equiv 1 (\mathrm{mod} 5)$ und $x \equiv 3 (\mathrm{mod} 10)$, so folgt aus den Rechenregeln für Kongruenzen aus $x \equiv 3 (\mathrm{mod} 10)$ auch $x \equiv 3 (\mathrm{mod} 5)$, womit man zwei sich gegenseitig widersprechende Kongruenzen erhalten würde. Dies liegt daran, dass die Module 5 und 10 nicht teilerfremd waren.

Satz 2.13 (**Chinesischer Restsatz**). *Es seien $m_1, \ldots, m_n$ natürliche Zahlen größer als 1 und $c_1, \ldots, c_n$ ganze Zahlen. Sind $m_1, \ldots, m_n$ paarweise teilerfremd zueinander, dann haben die simultanen Kongruenzen*

$$x \equiv c_1 \pmod{m_1} \ldots, x \equiv c_n \pmod{m_n}$$

genau eine Lösung modulo $m_1 \cdots m_n$.

Beweis. Für ein $1 \leq i \leq n$ sei $M_i := \frac{m_1 \cdots m_n}{m_i}$. Behauptung: $gcd\,(m_i, M_i) = 1$. Wäre nämlich $gcd\,(m_i, M_i) > 1$, so gäbe es eine Primzahl mit: $p \mid m_i$ und $p \mid M_i$ also $p \mid (m_1 \cdots m_{i-1} m_{i+1} \cdots m_n) \Rightarrow \exists j \in \{1, \ldots, n\}, i \neq j : p \mid m_j$. Daher würde $p \mid m_j$ und $p \mid m_i$ gelten, wodurch $gcd\,(m_i, m_j) \geq p$ wäre, was ein Widerspruch zur Teilerfremdheit der Module ist. Nach Satz 2.11. gilt: $\forall j \in \{1, \ldots n\} : \exists y_j \in \mathbb{Z} : M_j y_j \equiv 1 \pmod{m_j}$. Zusätzlich gilt: $M_i y_i \equiv 0 \pmod{m_j}$ für alle $i \neq j$, da $m_j \mid M_i$. Setze $x = \sum_{i=1}^{n} c_i M_i y_i$. Dann ist x eine Lösung der linearen simultanen Kongruenz. Denn für $1 \leq i \leq n, i \neq j$ gilt:

$$\Rightarrow x = \underbrace{c_1 M_1 y_1}_{\equiv 0 (\bmod m_j)} + \ldots + \underbrace{c_{j-1} M_{j-1} y_{j-1}}_{\equiv 0 (\bmod m_j)} + c_j M_j y_j + \underbrace{c_{j+1} M_{j+1} y_{j+1}}_{\equiv 0 (\bmod m_j)} + \ldots + \underbrace{c_n M_n y_n}_{\equiv 0 (\bmod m_j)}$$

$$\equiv c_j \underbrace{M_j y_j}_{\equiv 1 (\bmod m_j)} \equiv c_j \pmod{m_j} \text{ für } 1 \leq i \leq n$$

Es bleibt noch die Eindeutigkeit der Lösungen zu zeigen. Angenommen es gäbe zwei Lösungen $x \equiv c_j \pmod{m_j}$ und $y \equiv c_j \pmod{m_j}$, dann ist auch $x \equiv y \pmod{m_j}$. Nach Definition gilt $m_j \mid x - y$ und daher auch $m_1 \cdots m_n \mid x - y$. Insgesamt gilt also $x \equiv y \pmod{m_1 \cdots m_n}$, wodurch gezeigt wurde, dass es sich um mod $(m_1 \cdots m_n)$ kongruente Lösungen handelt. □

Beispiel. Zu lösen sei folgende simultane lineare Kongruenz:

$$x \equiv 2 \pmod{3}, x \equiv 3 \pmod{5}, x \equiv 2 \pmod{11}$$

Die Module $m_1 = 3$, $m_2 = 5$ und $m_3 = 11$ sind paarweise teilerfremd zueinander, da $gcd(3, 5) = 1$, $gcd(3, 11) = 1$ und $gcd(5, 11) = 1$. Daher gibt es genau eine $(\bmod 3 \cdot 5 \cdot 11)$ inkongruente Lösung.

$$M = 3 \cdot 5 \cdot 10 = 165,\ M_1 = \frac{M}{m_1} = \frac{165}{3} = 55,\ M_2 = \frac{M}{m_2} = \frac{165}{5} = 33,\ M_3 = \frac{M}{m_3} = \frac{165}{11} = 15$$

Nun müssen y_1, y_2 und y_3 berechnet werden:

$$\begin{aligned} 55y_1 &\equiv 1\,(\mathrm{mod}3) & 33y_2 &\equiv 1\,(\mathrm{mod}5) & 15y_3 &\equiv 1\,(\mathrm{mod}11) \\ y_1 &\equiv 1\,(\mathrm{mod}3) & 3y_2 &\equiv 1\,(\mathrm{mod}5) & 4y_3 &\equiv 1\,(\mathrm{mod}11) \\ & & y_2 &\equiv 2\,(\mathrm{mod}5) & y_3 &\equiv 3\,(\mathrm{mod}11) \end{aligned}$$

$$\Rightarrow x = c_1M_1y_1 + c_2M_2y_2 + c_3M_3y_3 = 2 \cdot 55 \cdot 1 + 3 \cdot 33 \cdot 2 + 2 \cdot 15 \cdot 3 = 398 \equiv 68\,(\mathrm{mod}165)$$

Diese Lösungsmethode benutzt den Beweis des Chinesischen Restsatzes. Es gibt noch eine andere Lösungsmethode, welche zumeist schneller und einfacher ist. Die einzelnen Kongruenzen werden dabei durch sukzessives Einsetzen gelöst:
Aus der ersten Kongruenz folgt: $x \equiv 2(\mathrm{mod}3) \Rightarrow \exists t \in \mathbb{Z} : x = 2 + 3t$. Setzt man dieses x nun in die zweite Kongruenz ein, erhält man folgendes:

$$x \equiv 3(\mathrm{mod}5) \Rightarrow 2 + 3t \equiv 3(\mathrm{mod}5) \Rightarrow 3t \equiv 1(\mathrm{mod}5) \Rightarrow 6t \equiv 2(\mathrm{mod}5) \Rightarrow t \equiv 2(\mathrm{mod}5)$$
$$\Rightarrow \exists u \in \mathbb{Z} : t = 2 + 5u \Rightarrow x = 2 + 3t = 2 + 3(2 + 5u) = 8 + 15u$$

Wiederum setzt man das neu berechnete x in die nächste Kongruenz ein und erhält:

$$x \equiv 2(\mathrm{mod}11) \Rightarrow 8 + 15u \equiv 2(\mathrm{mod}11) \Rightarrow 15u \equiv -6(\mathrm{mod}11) \Rightarrow 4u \equiv 5(\mathrm{mod}11) \Rightarrow u \equiv 15 \equiv 4(\mathrm{mod}11)$$
$$\Rightarrow \exists v \in \mathbb{Z} : u = 4 + 11v \Rightarrow x = 8 + 15u = 8 + 15(4 + 11v) = 68 + 165v$$

Somit gilt: $x \equiv 68\,(\mathrm{mod}165)$

Definition. (**Prime Restklasse)** Sei $m > 0$ eine natürliche Zahl und a eine ganze Zahl. Eine Restklasse $\overline{a} \in \mathbb{Z}/m\mathbb{Z}$ wird prim genannt, wenn $gcd(a, m) = 1$. Die Menge aller primen Restklassen wird wie folgt definiert:

$$\mathbb{Z}/m\mathbb{Z}^\times = \{a \in \mathbb{Z}/m\mathbb{Z} \mid gcd(a, m) = 1\}$$

Oftmals werden prime Restklassen auch als $\mathbb{Z}_m^*$ bezeichnet.

Definition. (**Eulersche φ-Funktion)** Für $m \in \mathbb{N}$ bezeichnet man die Anzahl der primen Restklassen, also der Anzahl der zu m teilerfremden Restklassen als Eulersche φ-Funktion:

$$\varphi(m) := |\mathbb{Z}/m\mathbb{Z}^\times| = |\{a \in \mathbb{N} \mid 1 \leq a \leq m \wedge \gcd(a, m) = 1\}|$$

Die Eulersche φ-Funktion ist also die Ordnung oder Mächtigkeit von $\mathbb{Z}/m\mathbb{Z}^\times$.

Satz 2.14. *Für jede Primzahl p besteht $\mathbb{Z}/p\mathbb{Z}$ nur aus primen Restklassen und es gilt daher:*

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

Beweis. Folgt sofort aus der Definition der primen Restklasse, da bei einer Primzahl p als Modul, stets $gcd(a, p) = 1$ für alle Elemente $\overline{a} \setminus \overline{0}$ von $\mathbb{Z}/p\mathbb{Z}$. Daher gibt es in $\mathbb{Z}/p\mathbb{Z}$ genau $p - 1$ prime Restklassen. □

Lemma 2.15. *Sei p eine Primzahl und $\alpha > 1$ eine natürliche Zahl. Dann gilt:*

$$\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$$

Beweis. Für $m = p^\alpha$ besteht die Menge $\mathbb{Z}/m\mathbb{Z}$ aus p^α Restklassen. Falls $d = gcd(k, p^\alpha) > 1$, dann ist diese Restklasse $\overline{k}$ nicht prim. Daraus folgt, dass $d \mid k$ und $d \mid p^\alpha$ und daher muss auch $p \mid d$ gelten. Wenn $p \mid d$ gilt auch $p \mid k$ und deswegen $k = pl < p^\alpha$ für $l = 0, \ldots, p^{\alpha-1} - 1$. Also gilt $gcd(k, p^\alpha) > 1$ für genau $p^{\alpha-1}$ Elemente, weswegen $gcd(k, p^\alpha) = 1$ für die übrigen $p^\alpha - p^{\alpha-1}$ Restklassen ist. Insgesamt gilt

$$\varphi(p^\alpha) = |\mathbb{Z}/p^\alpha\mathbb{Z}^\times| = p^\alpha - p^{\alpha-1}$$

wodurch die obige Behauptung bewiesen wäre. □

Satz 2.16. *Seien m und n natürliche Zahlen mit $gcd(m, n) = 1$, dann gilt*

$$\varphi(mn) = \varphi(m) \cdot \varphi(n)$$

Beweis. Folgt sofort aus dem Chinesischen Restsatz (Satz 2.13) □

Korollar 2.17. *Sei $m > 1$ eine natürliche Zahl mit der Primfaktorzerlegung $m = p_1^{\alpha_1} \cdots p_n^{\alpha_n}$, wobei $p_1, \ldots, p_n$ paarweise verschiedene Primzahlen sind und $\alpha_1, \ldots, \alpha_n \in \mathbb{N}$. Somit gilt:*

$$\begin{aligned}
\varphi(m) &= \varphi\left(p_1^{\alpha_1} \cdots p_n^{\alpha_n}\right) \\
&= \left(p_1^{\alpha_1} - p_1^{\alpha_1-1}\right) \cdots \left(p_n^{\alpha_n} - p_n^{\alpha_n-1}\right) \\
&= p_1^{\alpha_1-1} \cdots p_n^{\alpha_n-1} (p_1 - 1) \cdots (p_n - 1)
\end{aligned}$$

Bemerkung. Korollar 2.17. kann auch umgeschrieben werden:

$$\begin{aligned}
\varphi(m) &= p_1^{\alpha_1-1} \cdots p_n^{\alpha_n-1} (p_1 - 1) \cdots (p_n - 1) \\
&= \prod_{i=1}^{n} p_i^{\alpha_i-1} (p_i - 1) \\
&= \prod_{i=1}^{n} p_i^{\alpha_i} \prod_{i=1}^{n} \left(1 - \frac{1}{p_i}\right) \\
&= m \prod_{p|m} \left(1 - \frac{1}{p}\right)
\end{aligned}$$

Dabei durchläuft p alle Primzahlen, welche m teilen.

Beispiel. Es soll die Eulersche φ-Funktion für $m = 3528 = 2^3 \cdot 3^2 \cdot 7^2$ berechnet werden:

$$\varphi(3528) = (2^3 - 2^2) \cdot (3^2 - 3^1) \cdot (7^2 - 7^1) = (8-4) \cdot (9-3) \cdot (49-7) = 4 \cdot 6 \cdot 42 = 1008$$

oder

$$\varphi(3528) = 3528 \cdot \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{7}\right) = 3528 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{6}{7} = \frac{42336}{42} = 1008$$

Satz 2.18 (**Satz von Euler**). *Es sei a eine ganze Zahl und $m > 1$ eine natürliche Zahl. Falls $gcd(a, m) = 1$, also a und m teilerfremd sind, so gilt:*

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

Beweis. Es sei $\mathbb{Z}/m\mathbb{Z}^\times = \left\{k_1, k_2, \ldots, k_{\phi(m)}\right\}$ die Menge aller primen Restklassen modulo m, wobei $k_1, k_2, \ldots, k_{\phi(m)}$ die Repräsentanten der jeweiligen Restklassen sind. Weil a und m teilerfremd sind, ist auch $a \cdot k_i \in \mathbb{Z}/m\mathbb{Z}^\times$ teilerfremd zu n für alle $i \in \{1, \ldots, \phi(m)\}$. Darüber hinaus gilt $a \cdot k_i \not\equiv a \cdot k_j \bmod m$ für $i \neq j$, denn sonst wäre m ein Teiler von $a \cdot (k_i - k_j)$. Weil jedoch a und m teilerfremd sind, muss m ein Teiler von $(k_i - k_j)$ sein. Dies ist nur möglich, wenn $k_i - k_j = 0$ gilt, denn die Differenz zwischen k_i und k_j ist betragsmäßig stets kleiner als m. Daraus folgt:

$$\prod_{i=1}^{\phi(m)} a \cdot k_i \equiv \prod_{i=1}^{\phi(m)} k_i (\mathrm{mod} m) \Leftrightarrow a^{\phi(m)} \prod_{i=1}^{\phi(m)} k_i \equiv \prod_{i=1}^{\phi(m)} k_i (\mathrm{mod} m) \Leftrightarrow a^{\phi(m)} \equiv 1 (\mathrm{mod} m)$$

□

Der Satz von Euler verallgemeinert folgende wichtige Sätze:

Korollar 2.19 (**Der kleine Satz von Fermat**). *Ist p eine Primzahl und a eine ganze Zahl, welche teilerfremd zu p ist, so gilt:*

$$a^{p-1} \equiv 1 (\mathrm{mod} p)$$

Beweis. Die Aussage kann sofort aus dem Satz von Euler gefolgert werden, wenn man den Modul m durch die Primzahl p ersetzt, wodurch man für die Eulersche Phi-Funktion $\varphi(p) = p-1$ einsetzt. □

Korollar 2.20 (**Modifizierter Satz von Euler**). *Sind p und q zwei verschiedene Primzahlen mit m=pq, dann gilt für alle ganzen Zahlen a und k:*

$$a^{k\phi(m)+1} \equiv a (\mathrm{mod} m)$$

Dabei gilt es zu beachten, dass im Gegensatz zu den Sätzen von Euler und Fermat hier keine Teilerfremdheit von a und m vorausgesetzt wird.

Beweis. Der Beweis erfolgt mittels Fallunterscheidung:
1. Fall: Falls a teilerfremd zu m ist, so folgt die Behauptung direkt aus dem Satz von Euler.

$$a^{k\phi(m)+1} \equiv a^{k\phi(m)} \cdot a \equiv \left(a^{\phi(m)}\right)^k \cdot a \equiv 1^k \cdot a \equiv a(\mathrm{mod} m)$$

2. Fall: Falls a und m nicht teilerfremd sind, so existiert mindestens ein echter gemeinsamer Teiler. Wegen $m = pq$, können folgende zwei Fälle auftreten: Entweder ist a ein Vielfaches von m, oder a besitzt lediglich einen gemeinsamen Primteiler mit m.
(a) $m \mid a$. Daher ist $a \equiv 0(\mathrm{mod} m)$ und

$$a^{k\phi(m)+1} \equiv 0^{k\phi(m)+1} \equiv 0 \equiv a(\mathrm{mod} m)$$

(b) $p \mid a, q \nmid a$. Dann ist $a \equiv 0(\mathrm{mod} p)$ und damit

$$a^{k\phi(m)+1} \equiv 0^{k\phi(m)+1} \equiv 0 \equiv a(\mathrm{mod} p)$$

und nach Anwendung von Korollar 2.19.

$$a^{k\phi(m)+1} \equiv a^{k(q-1)(p-1)+1} \equiv \left(a^{(q-1)}\right)^{k(p-1)} \cdot a \equiv 1^{k(p-1)} \cdot a \equiv a(\mathrm{mod} q).$$

Wenn eine Zahl modulo p und modulo q denselben Rest hat, dann hat diese auch modulo pq denselben Rest. Deswegen gilt:

$$a^{k\phi m+1} \equiv a(\mathrm{mod} m)$$

□

Der Satz von Fermat kann beispielsweise verwendet werden um große Potenzen modulo m einfach zu berechnen.

Beispiel. Berechne $2^{10000} \equiv (\mathrm{mod} 67)$. Da der Modul 67 eine Primzahl ist, kann der Satz von Fermat wie folgt verwendet werden:

$$2^{66} \equiv 1(\mathrm{mod} 67)$$

Als nächstes wird eine Division mit Rest durchgeführt und man erhält $10000 = 151 \cdot 66 + 34$. Die zu lösende Kongruenz kann wie folgt dargestellt werden:

$$\begin{aligned} 2^{10000} = (2^{66})^{151} \cdot 2^{34} \equiv 1^{151} \cdot (2^6)^5 \cdot 2^4 \equiv 64^5 \cdot 16 \equiv (-3)^5 \cdot 16 \equiv (-3) \cdot (-3)^4 \cdot 16 \equiv (-3) \cdot 81 \cdot 16 \\ \equiv (-3) \cdot 14 \cdot 16 \equiv (-42) \cdot 16 \equiv 25 \cdot 16 \equiv 400 \equiv 65(\mathrm{mod} 67) \end{aligned}$$

Durch Anwendung des Satzes von Fermat und der Rechenregeln für Kongruenzen konnte hierbei viel Zeit und Aufwand gespart werden.

Definition. (Quadratische Reste und Nichtreste) Sei $m > 1$ eine natürliche Zahl und a eine ganze Zahl mit $gcd(m, a) = 1$. Dann nennt man a quadratischen Rest modulo m, falls die Kongruenz $x^2 \equiv a(\mathrm{mod}m)$ lösbar ist. Wenn diese Kongruenz nicht lösbar ist, so bezeichnet man a als quadratischen Nichtrest modulo m.

Definition. (Legendre-Symbol) Sei p eine ungerade Primzahl und $a \in \mathbb{Z}$. Das Legendre-Symbol $\left(\frac{a}{p}\right)$ wird definiert durch:

$$\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{wenn a quadratischer Rest modulo } p \text{ ist, also } x^2 \equiv a(\mathrm{mod}p) \text{ lösbar ist} \\ -1 & \text{wenn a quadratischer Nichtrest modulo } p \text{ ist, also } x^2 \equiv a(\mathrm{mod}p) \text{ nicht lösbar ist} \\ 0 & \text{wenn } p \mid a \end{cases}$$

Bemerkung. Man sagt „a nach p“ für $\left(\frac{a}{p}\right)$. Gilt $a \equiv b(\mathrm{mod}p)$, dann ist offenbar $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$.
Beispiel. Es sollen die quadratischen Reste und Nichtreste für $p = 7$ bestimmt werden:

x	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
x^2	0	1	4	9	16	25	36
$x^2(\mathrm{mod}7)$	0	1	4	2	2	4	1

Somit ergeben sich als quadratische Reste die Zahlen $1, 2$ und 4 und als quadratische Nichtreste die Zahlen $3, 5$ und 6. Aufgrund dessen würde man zum Beispiel $\left(\frac{2}{7}\right) = 1$ schreiben.

Satz 2.21 (**Euler-Kriterium**). *Ist p eine ungerade Primzahl, so gilt für jede ganze Zahl a:*

$$a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \bmod p.$$

Korollar 2.22. *Sei p eine ungerade Primzahl und $a_1 \ldots, a_n \in \mathbb{Z}$. Dann gilt:*

$$\left(\frac{a_1 \cdots a_n}{p}\right) = \left(\frac{a_1}{p}\right) \cdots \left(\frac{a_n}{p}\right)$$

Demnach ist das Legendre-Symbol multiplikativ im ersten Argument.

Korollar 2.23 (**Erster Ergänzungssatz**). *Wenn p eine ungerade Primzahl ist, so gilt:*

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} +1, & \textit{falls} \quad p \equiv 1 \bmod 4 \\ -1, & \textit{falls} \quad p \equiv 3 \bmod 4 \end{cases}$$

Das heißt die Zahl -1 ist genau dann quadratischer Rest modulo p, wenn der Exponent $\frac{p-1}{2}$ gerade ist. Dies ist genau dann der Fall, falls $p \equiv 1(\mathrm{mod}4)$ gilt. Umgekehrt ist -1 ein

quadratischer Nichtrest modulo p, wenn $p \equiv 3(\text{mod}4)$*, denn dann wäre der Exponent* $\frac{p-1}{2}$ *ungerade.*

Beispiel. Zu berechnen ist das Legendre-Symbol $\left(\frac{-1}{7}\right)$:

$$\left(\frac{-1}{7}\right) = (-1)^{\frac{7-1}{2}} = (-1)^3 = -1 \Rightarrow -1 \text{ ist quadratischer Nichtrest modulo } 7$$

oder

$$7 \equiv 3(\text{mod}4) \Rightarrow \left(\frac{-1}{7}\right) = -1$$

Korollar 2.24 (**Zweiter Ergänzungssatz**)**.** *Wenn p eine ungerade Primzahl ist, so gilt:*

$$\left(\frac{2}{p}\right) = (-1)^{\frac{\left(p^2-1\right)}{8}} = \begin{cases} +1, & \textit{falls} \quad p \equiv \pm 1 \bmod 8 \\ -1, & \textit{falls} \quad p \equiv \pm 3 \bmod 8 \end{cases}$$

Beispiel. Zu berechnen ist das Legendre-Symbol $\left(\frac{2}{7}\right)$:

$$\left(\frac{2}{7}\right) = (-1)^{\frac{7^2-1}{8}} = (-1)^6 = 1 \Rightarrow 2 \text{ ist quadratischer Rest modulo } 7$$

oder

$$7 \equiv -1(\text{mod}8) \Rightarrow \left(\frac{2}{7}\right) = 1$$

Satz 2.25 (**Quadratisches Reziprozitätsgesetz**)**.** *Seien* $p \neq q$ *zwei unterschiedliche ungerade Primzahlen. Es gilt*

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}$$

Durch Umformulieren wird das quadratische Reziprozitätsgesetz meist in folgender Form dargestellt:

$$\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}$$

Es gilt also:

$$\left(\frac{p}{q}\right) = \begin{cases} +\left(\frac{q}{p}\right), & \textit{falls} \quad p \equiv 1 \bmod 4 \ \textit{oder}\ q \equiv 1 \bmod 4 \\ -\left(\frac{q}{p}\right), & \textit{falls} \quad p \equiv q \equiv 3 \bmod 4 \end{cases}$$

Für den Beweis siehe ([10], Satz 11.4., S. 85)

Beispiel. Zu lösen sei $x^2 \equiv -24(\bmod 61)$, wobei die Zahl 61 eine Primzahl ist. Da das Legendre-Symbol multiplikativ im ersten Argument ist, gilt:

$$\left(\frac{-24}{61}\right) = \left(\frac{-1}{61}\right)\cdot\left(\frac{3}{61}\right)\cdot\left(\frac{2}{61}\right)\cdot\left(\frac{2}{61}\right)\cdot\left(\frac{2}{61}\right)$$

Unter Anwendung des ersten Ergänzungssatzes ergibt sich für $\left(\frac{-1}{61}\right) = 1$, da $61 \equiv 1(\mathrm{mod}4)$. Aus dem zweiten Ergänzungssatz erhält man $\left(\frac{2}{61}\right) = -1$, da $61 \equiv -3(\mathrm{mod}8)$. Schließlich folgt aus dem Quadratischem Reziprozitätsgesetz:

$$\left(\frac{3}{61}\right) = \left(\frac{61}{3}\right) = \left(\frac{1}{3}\right) = 1$$

Der erste Schritt begründet sich darin, dass $61 \equiv 1(\mathrm{mod}\,4)$ ist und beim zweiten Schritt wurde verwendet, dass $61 \equiv 1(\mathrm{mod}3)$, was bedeutet, dass $\left(\frac{61}{3}\right) = \left(\frac{1}{3}\right)$. Insgesamt folgt also:

$$\left(\frac{-24}{61}\right) = 1 \cdot 1 \cdot (-1) \cdot (-1) \cdot (-1) = -1$$

$\Rightarrow -24$ ist quadratischer Nichtrest modulo 61 und daher ist $x^2 \equiv -24(\mathrm{mod}61)$ nicht lösbar.

2.2 Algebraische Strukturen

Sämtliche in diesem Abschnitt angeführten Erkenntnisse zu den algebraischen Strukturen richten sich nach den Quellen ([3, 10, 8, 13, 7, 5]).

Definition (**Gruppe**)**.** Eine Menge zusammen mit einer Verknüpfung $(G, \circ)$ wird als Gruppe bezeichnet, wenn folgende Bedingungen erfüllt sind:
1) Assoziativität:

$$\forall a, b, c \in G : (a \circ b) \circ c = a \circ (b \circ c)$$

2) Neutrales Element:

$$\exists e \in G : \forall a \in G : e \circ a = a \circ e = a$$

3) Inverses Element:

$$\forall a \in G : \exists a^{-1} \in G : a^{-1} \circ a = a \circ a^{-1} = e$$

Gilt zusätzlich
4) Kommutativität:

$$\forall a, b \in G : a \circ b = b \circ a$$

so nennt man G eine kommutative oder abelsche Gruppe.

Bemerkung. Die folgenden Mengen $(\mathbb{Z}, +)$, $(\mathbb{Q}, \cdot)$ und $(\mathbb{R}, +)$ mit deren Verknüpfungen sind abelsche Gruppen. Insbesondere ist $(\mathbb{Z}/m\mathbb{Z}, +)$ eine abelsche Gruppe. Gilt lediglich die Eigenschaft der Assoziativität so nennt man $(G, \circ)$ auch Halbgruppe.

Definition (**Untergruppe**)**.** Es sei $(G, \circ, e)$ eine Gruppe, wobei e das neutrale Element bezeichnet. Eine Teilmenge $H \subseteq G$ wird Untergruppe von G genannt, wenn $(H, \circ, e)$ selbst eine Gruppe ist.

Bemerkung. Alle Gruppen G besitzen die beiden trivialen Untergruppen $\{e\}$, welche lediglich aus dem neutralem Element der Gruppe besteht, und die Gruppe G selbst. Beispielsweise ist $(\mathbb{Z}, +)$ eine Untergruppe von $(\mathbb{R}, +)$.

Definition. Sei $(G, \circ)$ eine Gruppe. Für ein $n \in \mathbb{N}$ wird die $n - te$ Potenz eines Elements $a \in G$ definiert als:

$$a^n = a \circ \ldots \circ a \quad (n - \text{mal}) \quad , a^{-n} = (a^{-1})^n \quad \text{sowie} \quad a^0 = e$$

wobei a^{-1} das Inverse von a ist.

Definition. Ist $(G, \circ)$ eine endliche Gruppe und a ein Element von G. Dann nennt man

$$\langle a \rangle = \{a^n \mid n \in \mathbb{N}\}$$

die von a erzeugte Untergruppe, wobei a als erzeugendes Element bezeichnet wird.

Beispiel. Man betrachte die Menge $\mathbb{Z}/8\mathbb{Z} = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}, \overline{7}\}$. Versehen mit der Addition als Verknüpfung ist $(\mathbb{Z}/8\mathbb{Z}, +)$ eine Gruppe und die Elemente 0, 1, 2 und 4 erzeugen alle möglichen Untergruppen:

$$\begin{aligned}
\langle 0 \rangle &= \{0\} \\
\langle 1 \rangle &= \{1, 2, 3, 4, 5, 6, 7, 0\} \\
\langle 2 \rangle &= \{2, 4, 6, 0\} \\
\langle 4 \rangle &= \{4, 0\}
\end{aligned}$$

Die übrigen Elemente erzeugen dieselben Untergruppen, wie oben angegeben. Es wurden dabei die Elemente solange miteinander verknüpft, oder in diesem Fall addiert, bis sich kein neues Element mehr ergibt.

Definition (**Zyklische Gruppe**)**.** Man nennt eine Gruppe G zyklisch, falls sie aus den Potenzen eines einzigen Elements $a \in G$ erzeugbar ist, das heißt:

$$G \text{ zyklisch} \iff \text{es gibt ein } a \in G \text{ mit } G = \langle a \rangle = \{a^n \mid n \in \mathbb{Z}\}$$

Bemerkung. Die im vorigen Beispiel beschriebene Gruppe $(\mathbb{Z}/8\mathbb{Z}, +)$ ist zyklisch, da sie vom Element 1 erzeugt wird. Darüber hinaus ist auch die Gruppe $(\mathbb{Z}/7\mathbb{Z}^\times, \cdot)$ zyklisch, da alle Elemente von $\mathbb{Z}/7\mathbb{Z}^\times = \{1, 2, 3, 4, 5, 6\}$ durch wiederholtes Potenzieren der Zahl 5 erzeugt werden können, was man durch nachrechnen leicht überprüfen kann. Für eine Primzahl p die Gruppe $(\mathbb{Z}/p\mathbb{Z}^\times, \cdot)$ zyklisch ([5], S. 81).

Definition. Sei $(G, \circ)$ eine endliche Gruppe, das heißt die Menge G besteht aus einer endlichen Anzahl an Elementen. Diese Anzahl der Elemente von G nennt man Ordnung von G und man schreibt dafür $|G|$ oder $\text{ord}(G)$.

Für ein $a \in G$ bezeichnet $ord(a)$ die Ordnung der von a erzeugten Untergruppe. Man schreibt dafür $\text{ord}(a) = |\langle a \rangle|$.

Bemerkung. Gilt $\text{ord}(a) = \text{ord}(G)$ für eine endliche Gruppe G und für ein $a \in G$, so ist G trivialerweise zyklisch.

Satz 2.26 (**Satz von Lagrange**). *Ist G eine endliche Gruppe und $H \subset G$ eine Untergruppe, so gilt* $\text{ord}(H)$ *ist ein Teiler von* $\text{ord}(G)$. *Dabei nennt man den folgenden ganzzahligen Quotienten*

$$[G : H] := \frac{\text{ord}(G)}{\text{ord}(H)}$$

auch Index von H in G.

Beweis. Ist $a \in G$ ein beliebiges Element, dann nennt man folgende Menge

$$aH := \{ah : h \in H\}$$

auch Linksnebenklasse von H. Trivialerweise hat aH gleich viele Elemente wie die Untergruppe H selbst. Daher gilt, dass, $|(aH)| = \text{ord}(H)$ für alle $a \in G$. Seien a_1H und a_2H zwei Nebenklassen, so muss genau einer der folgenden beiden Fälle eintreten:

$$\text{(i) } a_1H = a_2H \qquad \text{(ii) } a_1H \cap a_2H = \emptyset$$

Wenn Fall (ii) nicht gilt, dann existiert ein $b \in a_1H \cap a_2H$. Daher muss es also Elemente $h_1, h_2 \in H$ geben mit $b = a_1h_1 = a_2h_2$. Durch Umformen erhält man daraus, dass $a_1^{-1}a_2 = h_1h_2^{-1} \in H$. Wenn x ein beliebiges Element von a_1H ist, so gilt $x = a_1y$ für ein $y \in H$. Insgesamt gilt also:

$$x = a_1 \left(a_1^{-1}a_2\right) \left(h_1h_2^{-1}\right)^{-1} y = a_2 \left(h_2h_1^{-1}y\right) \in a_2H$$

Aufgrund dessen wäre gezeigt, dass $a_1H \subset a_2H$. Aus Symmetriegründen gilt auch $a_2H \subset a_1H$, woraus (i) folgt. Nach dem bisher Gezeigtem lässt sich die Gruppe G als disjunkte Vereinigung endlich vieler Linksnebenklassen $x_1H, \ldots, x_sH$ schreiben, wodurch schlussendlich folgt, dass $\text{ord}(G) = s \cdot \text{ord}(H)$. ([10], Satz 7.1., S. 53) □

Bemerkung. Aus dem Satz von Lagrange folgt direkt, dass eine echte Untergruppe $H \subset G$ maximal die Hälfte der Elementanzahl von G haben kann.

Korollar 2.27. *Ist G eine endliche Gruppe und sei a ein Element dieser Gruppe, so ist $ord(a)$ ein Teiler von $ord(G)$.*

Beweis. Unter $\text{ord}(a)$ wird definitionsgemäß die Ordnung der von a erzeugten Untergruppe verstanden und diese teilt nach dem Satz von Lagrange die Ordnung der Gruppe G selbst. □

Die Sätze von Euler und Fermat wurden bereits im vorigen Abschnitt angeführt und bewiesen. Diese elementaren Sätze der Zahlentheorie können jedoch auf Spezialfälle von Aussagen über endliche Gruppen zurückgeführt werden. ([10], S. 53)

Satz 2.28. *Ist G eine endliche Gruppe, so gilt für jedes Element x dieser Gruppe*

$$x^{ord(G)} = e$$

wobei mit e das neutrale Element der Gruppe bezeichnet wird.

Beweis. Wegen Korollar 2.27. existierte eine ganze Zahl s mit $\text{ord}(G) = \text{ord}(x) \cdot s$. Daher folgt $x^{\text{ord}(G)} = \left(x^{\text{ord}(x)}\right)^s = e^s = e$ □

Bemerkung. Betrachtet man dabei ein beliebiges Element a der primen Restklassengruppe $(\mathbb{Z}/m\mathbb{Z}^\times, \cdot)$ mit deren Ordnung $\varphi(m)$, so ergibt sich daraus der Satz von Euler wie folgt:

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

Falls m eine Primzahl p ist, so ergibt sich für die Restklassengruppe $(\mathbb{Z}/p\mathbb{Z}, \cdot)$ die Ordnung $\varphi(p) = p - 1$, woraus sofort der Kleine Satz von Fermat folgt:

$$a^{p-1} \equiv 1 \bmod p$$

Das multiplikative neutrale Element der Gruppe $(\mathbb{Z}/m\mathbb{Z}^\times, \cdot)$ ist die Zahl 1, wodurch es auch Einselement genannt wird.

Definition. Als Ring bezeichnet man ein Tripel $(R, +, \cdot)$ mit zwei Verknüpfungen, falls folgende Eigenschaften erfüllt sind:

1) $(R, +, 0)$ ist abelsche Gruppe
2) $(R, \cdot)$ ist assoziativ, also eine Halbgruppe
3) $(R, +, \cdot)$ erfüllt die Distributivgesetze von $+$ und $\cdot$, das heißt $\forall a, b, c \in R$ gilt:

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad \text{und} \quad (a + b) \cdot c = (a \cdot c) + (b \cdot c)$$

In der Gruppe $(R, +, 0)$ bezeichnet die 0 das additiv neutrale Element, welches man auch Nullelement nennt. Falls zusätzlich die multiplikative Halbgruppe $(R, \cdot, 1)$ ein neutrales Element 1 besitzt, so nennt man den entsprechenden Ring $(R, +, \cdot)$ auch *Ring mit Eins(element).* Ein Ring $(R, +, \cdot)$ ist kommutativ, falls die Halbgruppe $(R, \cdot)$ kommutativ ist. Besitzt der Ring $(R, +, \cdot)$ ein Einselement und ist darüber hinaus kommutativ, so nennt man diesen *kommutativen Ring mit Einselement.*

Bemerkung. Sowohl die Menge der ganzen Zahlen $(\mathbb{Z}, +, \cdot)$, als auch die Menge der reellen Zahlen $(\mathbb{R}, +, \cdot)$ bilden mit den Verknüpfungen $+$ und $\cdot$ einen kommutativen Ring mit Einselement.

Proposition 2.29. *Die Menge der Restklassen $\mathbb{Z}/m\mathbb{Z}$ bildet mit der herkömmlichen Addition und Multiplikation einen kommutativen Ring mit Einselement.*

Beweis. Die Gültigkeit von Assoziativgesetz, Kommutativgesetz und der Distributivgesetze folgt sofort aus der Gültigkeit für die Menge der ganzen Zahlen $\mathbb{Z}$, da die Repräsentanten der Restklassen ganze Zahlen sind. Als Nullelement wird die Restklasse $\overline{0}$ bezeichnet und das Einselement ist die Restklasse $\overline{1}$. Das inverse Element bezüglich der Addition einer Restklasse $\overline{x}$ ist $\overline{-x}$. □

Definition (Einheit). Existiert ein multiplikatives Inverses für ein Element a des Rings $(R, +, \cdot)$ mit Einselement, so nennt man dieses *Einheit*.

Bemerkung. Die Menge aller Einheiten eines Rings mit Einselement wird auch als Einheitengruppe $R^\times$ bezeichnet, da diese stets eine Gruppe bildet. Bei den ganzen Zahlen sind die Elemente 1 und -1 Einheiten, da diese als einzige ein multiplikatives Inverses besitzen. Die Einheitengruppe der rationalen Zahlen ist $\mathbb{Q}^\times = \mathbb{Q}\backslash\{0\}$, da bis auf 0 alle Zahlen invertierbar sind. Gleiches gilt für die reellen Zahlen, also $\mathbb{R}^\times = \mathbb{R}\backslash\{0\}$.

Definition (Nullteiler, Integritätsbereich). Ist $(R, +, \cdot)$ ein kommutativer Ring, so nennt man ein Elemente $a \neq 0 \in R$ *Nullteiler*, falls ein $b \neq 0 \in R$ existiert, sodass $a \cdot b = 0$. Wenn $(R, +, \cdot)$ ein Einselement besitzt, so heißt $(R, +, \cdot)$ auch Integritätsbereich, falls folgendes gilt:

$$\forall a, b \in R : a \cdot b = 0 \Rightarrow a = 0 \vee b = 0$$

Bemerkung. Nach der vorhergehenden Definition besitzt ein Integritätsbereich keine Nullteiler, wodurch dieser ein *nullteilerfreier kommutativer Ring mit Einselement* ist. Die Menge der ganzen Zahlen $(\mathbb{Z}, +, \cdot)$ und die Menge der reellen Zahlen $(\mathbb{R}, +, \cdot)$ sind Integritätsbereiche.

Satz 2.30. *Im Restklassenring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ sind die Nullteiler die Restklassen $\overline{a} = a + m\mathbb{Z}$, für welche $1 < gcd(a, m) < m$ gilt.*

Beweis. Angenommen $\overline{a} = a + m\mathbb{Z}$ wäre ein Nullteiler im Restklassenring $\mathbb{Z}/m\mathbb{Z}$. Daher muss es ein Element $\overline{b} = b + m\mathbb{Z}$ geben mit $a \cdot b \equiv 0 \bmod m$, aber dennoch darf weder $a \equiv 0 \bmod m$ noch $b \equiv 0 \bmod m$ gelten. Aufgrund dessen ist m ein Teiler von ab, aber $m \nmid a$ und $m \nmid b$, woraus $1 < gcd(a, m) < m$ folgt. Wenn umgekehrt $1 < gcd(a, m) < m$ gilt und $b = \frac{m}{gcd(a,m)}$ gesetzt wird, dann folgt, dass $a \not\equiv 0 \bmod m$, $ab \equiv 0 \bmod m$ und $b \not\equiv 0 \bmod m$. Somit wäre $\overline{a} = a + m\mathbb{Z}$ ein Nullteiler von $\mathbb{Z}/m\mathbb{Z}$. □

Bemerkung. Nach dem obigen Satz folgt, dass für eine Primzahl p der Restklassenring $\mathbb{Z}/p\mathbb{Z}$ ein Integritätsbereich ist, da für alle Elemente a des Rings $gcd(a, p) = 1$ gilt.

Satz 2.31. *Im Ring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ besitzt eine Restklasse $\overline{a}$ modulo m genau dann ein multiplikatives Inverses, falls $gcd(a, m) = 1$.*

Beweis. Gilt $gcd(a, m) = 1$, so gibt es nach Lemma 2.5 Zahlen $x, y \in \mathbb{Z}$, sodass

$$gcd(a, m) = 1 = x \cdot a + y \cdot m \Rightarrow x \cdot a \equiv 1 \bmod m$$

Somit wäre x das inverse Element von a mod m . Setzt man umgekehrt voraus, dass a mod m im Ring $\mathbb{Z}/m\mathbb{Z}$ ein Inverses x mod m besitzt, dann folgt für eine ganze Zahl l:

$$x \cdot a \equiv 1 \bmod m \Rightarrow x \cdot a = 1 + l \cdot m$$

Mittels Umformung dieser Gleichung und durch erneute Anwendung von Lemma 2.5 erhält man:

$$x \cdot a + (-l) \cdot m = 1 \Rightarrow gcd(a, m) = 1$$

□

Bemerkung. Insbesondere besitzt jedes Element $\overline{a} \neq 0$ des Restklassenrings $\mathbb{Z}/p\mathbb{Z}$ ein multiplikatives inverses Element, da für jede ganze Zahl $a \neq 0$ gilt, dass $gcd(a, p) = 1$.

Definition (Körper). Als Körper $(K, +, \cdot)$ bezeichnet man einen Ring mit Einselement, wenn zusätzlich $(K\backslash\{0\}, \cdot)$ eine abelsche Gruppe ist.

Bemerkung. Das heißt ein Ring mit Einselement ist genau dann ein Körper, wenn dieser kommutativ ist und jedes von 0 verschiedene Element ein multiplikatives Inverses besitzt. Die Menge der rationalen und reellen Zahlen bildet mit den Verknüpfungen + und · die Körper $(\mathbb{Q}, +, \cdot)$ und $(\mathbb{R}, +, \cdot)$. Insbesondere folgt aus Satz 2.31., dass der kommutative Ring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ genau dann ein Körper ist, wenn m eine Primzahl ist. Für eine Primzahl p wird der Körper $(\mathbb{Z}/p\mathbb{Z}, +, \cdot)$ in der Literatur üblicherweise mit $\mathbb{F}_p$ bezeichnet. Darüber hinaus folgt aus Satz 2.30. und Satz 2.31., dass Körper keine Nullteiler besitzen und somit Integritätsbereiche sind.

2.3 Elliptische Kurven

Die elliptischen Kurven sind ein Paradebeispiel dafür, wie nützlich ein mathematisches Thema bezogen auf alltägliche Anwendungen sein kann, auch wenn man dies im ersten Moment nicht vermuten würde. Man beschäftigt sich schon lange und umfassend mit diesen speziellen geometrischen Objekten, aber erst im Jahre 1985 erkannte man deren praktische Relevanz im Zuge kryptografischer Anwendungen. ([2]) Das ElGamal-Verschlüsselungsverfahren, welches auf dem Diffie-Hellmann-Schlüsselaustausch beruht, benutzt elliptische Kurven, da unter Verwendung dieser das Problem des diskreten Logarithmus ausgesprochen schwierig zu lösen ist ([1], S. 146). Außerdem spielen elliptische Kurven eine wesentliche Rolle bei der Faktorisierung großer Zahlen

([9], S. 317). Daher sollen in diesem Abschnitt sämtliche Grundlagen zu diesen algebraischen Kurven angeführt werden, um den dazugehörigen Faktorisierungsalgorithmus nachvollziehen zu können. Grob gesagt handelt es sich bei elliptischen Kurven um die Lösungsmenge der folgenden Gleichung

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

wobei $a_i \in K$ für einen beliebigen Körper K sind. Man bezeichnet eine derartige Gleichung auch als *allgemeine Weierstraß-Gleichung.* ([9], S. 284) Die etwas verwirrende Bezeichnung der Koeffizienten a_i und das Fehlen von a_5 hat einen historischen Hintergrund, auf welchen nicht näher eingegangen werden soll ([2], S. 23). Im Grunde benötigt man einen Körper K und ein Polynom F der Gestalt

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 \in K[x, y]$$

Die Menge aller Nullstellen (z.B. in K oder in einem algebraischen Abschluss $\overline{K}$ von K) dieses Polynoms bezeichnet man als elliptische Kurve, wobei noch zwei Modifikationen vorgenommen werden müssen. Auf der einen Seite müssen von dieser Nullstellenmenge die singulären Punkte ausgeschlossen werden. Andererseits benötigt man noch einen zusätzlichen Punkt im „Unendlichen“, um schlussendlich bezüglich der Addition eine Gruppenstruktur auf den Punkten einer elliptischen Kurven zu erhalten. Diese beiden Modifikationen werden nun genauer unter die Lupe genommen, um eine vollständige Definition der elliptischen Kurven zu erhalten. Im Folgenden wird mit K ein spezieller Körper bezeichnet. Besonders interessant und praxisrelevant sind elliptische Kurven über dem endlichen Körper $\mathbb{F}_p$. Betrachtet man elliptische Kurven über dem kommutativen Ring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$, führt dies mit der hier definierten Gruppenstruktur zu Problemen, weil für bestimmte Elemente eine Division in $\mathbb{Z}/m\mathbb{Z}$ nicht durchführbar ist. Gerade bei Faktorisierungsalgorithmen spielt dies eine äußerst wichtige Rolle, da man durch die eben erwähnte Problematik nicht nur unterscheiden kann, ob es sich beim Modul m um eine Primzahl handelt oder nicht, sondern man m auch gegebenenfalls faktorisieren kann. Es sei von nun an $F(x, y)$ ein Polynom aus $K[x, y]$ mit der oben angeführten Form (Weierstraß-Gleichung). Unter $E(K)$ versteht man die Nullstellenmenge von $F(x, y)$ über dem Körper K.

Definition (Singuläre Punkte). Man nennt einen Punkt (v, w) der durch $F(x, y) = 0$ definierten algebraischen Kurve singulär, falls für den Gradienten von F gilt

$$\nabla F(v, w) = (0, 0) \iff \frac{\partial F}{\partial x}(v, w) = \frac{\partial F}{\partial y}(v, w) = 0$$

Mit ∇ bezeichnet man den sogenannten Nabla-Operator, mit welchem in diesem Kontext der Gradient beschrieben wird. Unter dem Gradienten versteht man hier den Zeilenvektor, bestehend aus allen partiellen Ableitungen von F. Grafisch gesehen hätte eine elliptische Kurve

in einem singulären Punkt eine nicht eindeutig bestimmte Tangente. Die Neilsche Parabel oder der Newtonsche Knoten haben beide im Punkt $(0, 0)$ einen singulären Punkt, wie man in den folgenden Abblidungen 1 und 2 sehen kann. Nachdem (eindeutige) Tangenten aber

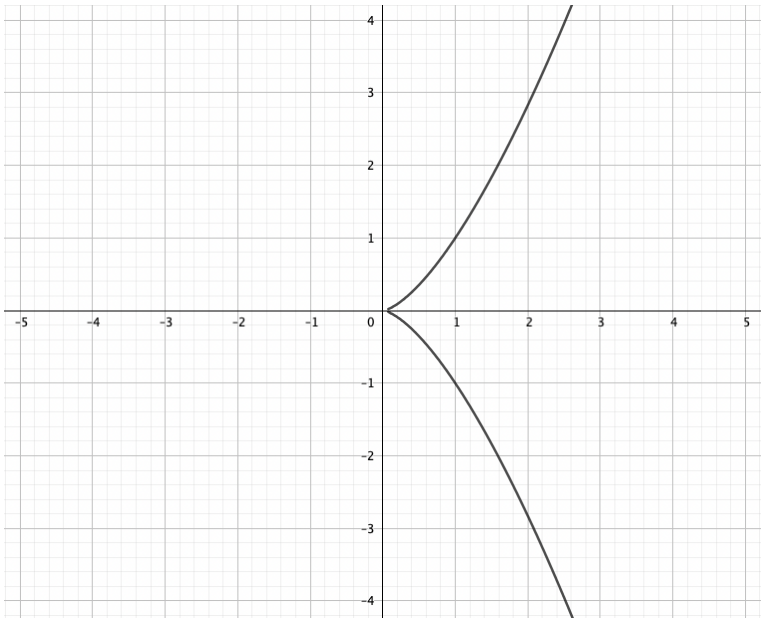

Abbildung 1: Neilsche Parabel ($y^2 = x^3$) in GeoGebra

äußerst wichtig für die noch zu definierende Verknüfung der Gruppe elliptischer Kurven sind, werden die singulären Punkte ausgeschlossen. Dabei bezeichnet man S als die Menge aller singulären Punkte. ([9], S. 284-286) Des Weiteren gilt für singuläre Punkte einer algebraischen Kurve folgender Satz.

Satz 2.32. *Sei K ein Körper und $(v, w) \in K^2$ ein Punkt der algebraischen Kurve, welche definiert wird durch $F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0$ über K. Der Punkt (v, w) ist genau dann singulär, falls sämtliche Geraden, welche durch diesen Punkt verlaufen, die algebraische Kurve mehrmals in diesem Punkt (v, w) schneiden. (d.h. sei eine Gerade durch (v, w) gegeben durch die Parameterdarstellung $g : (x, y) = (v, w) + t(r_x, r_y)$ mit Richtungsvektoren (r_x, r_y) und setzt man diese in $F(x, y)$ ein, so erhält man ein Polynom $\tilde{F}(t)$ in einer Variablen t und der Punkt (v, w) ist singulär, wenn $\tilde{F}(t)$ die 0 als mehrfache Nullstelle besitzt, d.h. g ist Tangente an die Kurve F im Punkt (v, w))*

Beweis. Man kann jede Gerade in K^2, welche durch den Punkt (v, w) verläuft in der Paramterdarstellung auch schreiben als $g : (x, y) = (v, w) + t(r_x, r_y)$. Für jeden Wert t bekommt man einen Punkt, sodass, wenn der Parameter t alle Werte von K durchlaufen hat, man die vollständige Gerade erhält. Setzt man die angeführte Parameterdarstellung in $F(x, y)$ ein, dann bekommt man ein Polynom $\tilde{F}(t)$ mit einer Variablen t. Da der Punkt (v, w) ein Punkt der Kurve ist, erhält man mit dem Parameterwert $t = 0$ eine Nullstelle von $\tilde{F}(t)$. Es

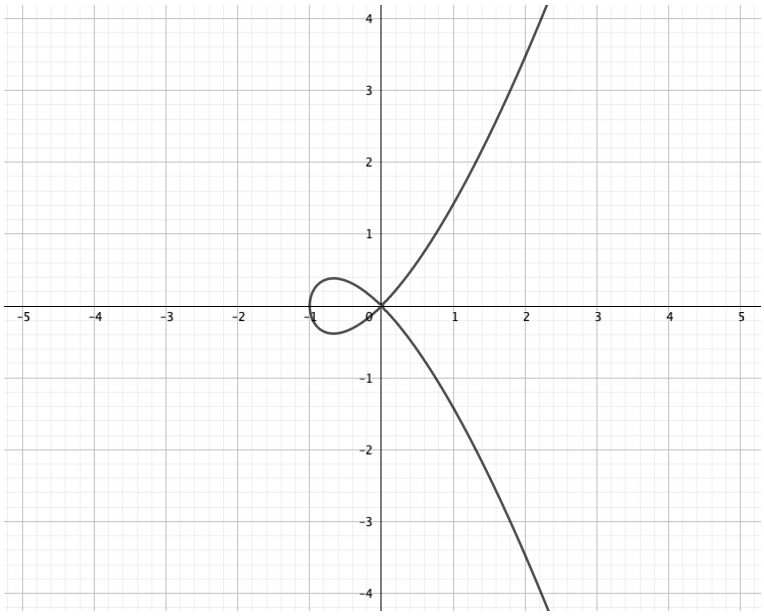

Abbildung 2: Newtonscher Knoten ($y^2 = x^3 + x^2$) in GeoGebra

gilt nun zu zeigen, dass es sich mit $t = 0$ um eine mehrfache Nullstelle handelt. Dazu setzt man (x, y) von der Paramterdarstellung der Geraden in F und in die partiellen Ableitungen $\frac{\partial F}{\partial x} = a_1 y - 3x^2 - 2a_2 x - a_4$ und $\frac{\partial F}{\partial y} = 2y + a_1 x + a_3$ ein. Man bezeichnet die neuen Polynome in t mit $\tilde{F}$, F'_x und $F'_y \in K[t]$

$$
\begin{aligned}
\tilde{F}(t) =& (w + tr_y)^2 + a_1 (v + tr_x)(w + tr_y) + a_3 (w + tr_y) - (v + tr_x)^3 - a_2 (v + tr_x)^2 \\
& - a_4 (v + tr_x) - a_6 \\
=& w^2 + 2wtr_y + t^2 r_y^2 + a_1 vw + a_1 vtr_y + a_1 wtr_x + a_1 t^2 r_x r_y + a_3 w + a_3 tr_y - v^3 \\
& - 3v^2 tr_x - 3vt^2 r_x^2 - t^3 r_x^3 - a_2 v^2 - 2a_2 vtr_x - a_2 t^2 r_x^2 - a_4 v - a_4 tr_x - a_6 \\
=& - r_x^3 t^3 + \left(-3r_x^2 v - a_2 r_x^2 + r_y^2 + a_1 r_x r_y\right) t^2 \\
& + \left(r_y (2w + a_1 v + a_3) + r_x \left(a_1 w - 3v^2 - 2a_2 v - a_4\right)\right) t \\
& + \left(w^2 + a_1 vw + a_3 w - v^3 - a_2 v^2 - a_4 v - a_6\right) \\
F'_x(t) =& a_1 (w + tr_y) - 3 (v + tr_x)^2 - 2a_2 (v + tr_x) - a_4 \\
=& a_1 w + a_1 tr_y - 3v^2 - 6vtr_x - 3t^2 r_x^2 - 2a_2 v - 2a_2 tr_x - a_4 \\
=& - 3r_x^2 t^2 + (a_1 r_y - 6vr_x - 2a_2 r_x) t + \left(a_1 w - 3v^2 - 2a_2 v - a_4\right) \\
F'_y(t) =& 2 (w + tr_y) + a_1 (v + tr_x) + a_3 \\
=& 2w + 2tr_y + a_1 v + a_1 tr_x + a_3 \\
=& (2r_y + a_1 r_x) t + (2w + a_1 v + a_3)
\end{aligned}
$$

Wenn (v, w) ein singulärer Punkt der algebraischen Kurve ist, so gilt für $t = 0$ definitionsgemäß $\tilde{F}(0) = F'_x(0) = F'_y(0) = 0$. Aufgrund dessen müssen die Absolutglieder von $\tilde{F}(0)$, $F'_x(0)$ und $F'_y(0)$ gleich 0 sein. Betrachtet man nun den Koeffizienten von t in $\tilde{F}$, so ist ersichtlich, dass dieser eine Linearkombination der absoluten Glieder von $F'_x(0)$ und $F'_y(0)$ ist, welche beide 0 sein müssen. Daher ist auch der Koeffizient des linearen Glieds 0. Umgekehrt sei $t = 0$ eine mehrfache Nullstelle von $\tilde{F}$ für alle möglichen Geraden. Der Koeffizient von t in $\tilde{F}$ muss daher unabhängig von der Wahl der Geraden, also unabhängig von den Richtungsvektoren r_x und r_y, 0 ergeben. Wählt man daher beispielsweise $(r_x, r_y) = (1, 0)$ und $(r_x, r_y) = (0, 1)$, so ergibt sich auch für die absoluten Glieder von F'_x und F'_y, dass beide 0 sein müssen, wodurch (v, w) nach Definition ein singulärer Punkt sein muss. ([9], Lemma 23.1, S. 286-287) □

Im Gegensatz zu beliebigen algebraischen Kurven sind elliptische Kurven definitionsgemäß singularitätenfrei. Zusammenfassend ergibt sich also folgende Definition für eine elliptische Kurve $E(K)$ über dem Körper K:

Definition (**Elliptische Kurve**)**.** Eine elliptische Kurve $\overline{E}(K)$ über einem Körper K wird definiert als

$$\overline{E}(K) := O \cup \Big\{(x, y) \in K^2 \mid F(x, y) = 0\Big\}$$

wobei $\overline{E}(K)$ keine singulären Punkte besitzen darf. Mit O wird der unendlich ferne Punkt bezeichnet. Eine elliptische Kurve ist also eine singularitätenfreie algebraische Kurve definiert durch ein Polynom der Gestalt

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 \in K[x, y]$$

Es bleibt noch zu klären, was es mit dem unendlich fernen Punkt O auf sich hat und wofür man diesen im Zusammenhang mit elliptischen Kurven benötigt. In der Literatur verwendet man auch oft ∞ anstelle von O. Im folgenden Abschnitt soll gezeigt werden, dass es sich bei elliptischen Kurven mit einer noch zu beschreibenden Verknüpfung tatsächlich um eine Gruppe handelt. Im Zuge dessen wird auch der unendlich ferne Punkt entsprechend erläutert. ([9], S. 284-285)

2.3.1 Gruppenstruktur elliptischer Kurven

Dass elliptische Kurven versehen mit einer Verknüpfung eine Gruppe ergeben, ist für die Kryptografie von großer Bedeutung. Nun gilt es diese Verknüpfung zu definieren und zu erklären. Doch bevor diese näher beschrieben werden, gilt es die Definition der elliptischen Kurve ein wenig zu vereinfachen, da dies für die Elliptische Kurven Methode ausreicht. Man kann die in den Grundlagen angeführte allgemeine Weierstraß-Gleichung unter gewissen Bedingungen in eine „kurze Weierstraß-Gleichung“ umwandeln. ([9], S. 320, 345-346).

Definition. Sei $x^3 + ax + b$ ein Polynom mit $a, b \in \mathbb{K}$ und ohne mehrfache Nullstelle $(4a^3 + 27b^2 \neq 0)$. Als elliptische Kurve über dem Körper K mit $char(K) \notin \{2, 3\}$ bezeichnet man die Menge $\overline{E}(K)$ der Punkte (x, y), welche die kubische Gleichung $y^2 = x^3 + ax + b$ erfüllen. Zusätzlich ist auch ein weiteres Element O, welches der unendlich ferne Punkt ist, ein Teil dieser Menge $\overline{E}(K)$. Man nennt eine derartige Polynomgleichung auch kurze Weierstraß-Gleichung. ([9], Definition 27.8, S. 346)

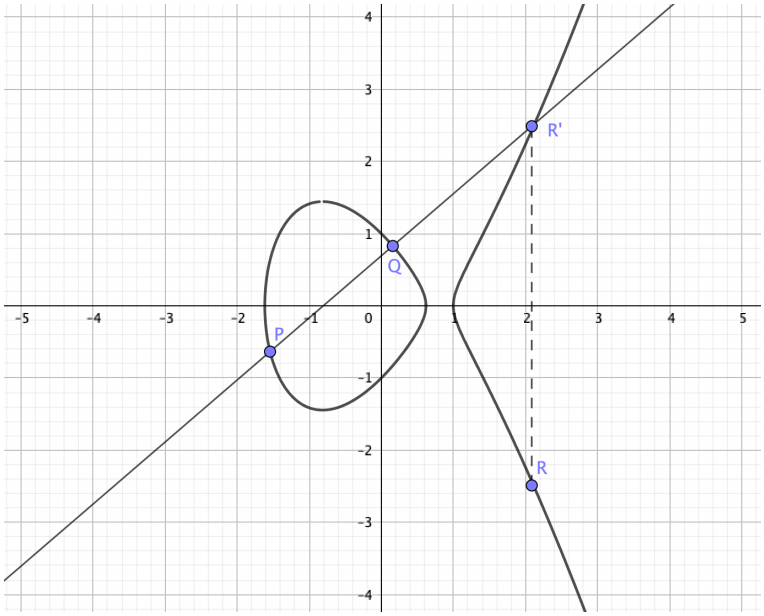

Abbildung 3: Punktaddition auf der elliptischen Kurve $y^2 = x^3 - 2x + 1$ in GeoGebra

Bemerkung. 1) Ist die Bedingung $4a^3 + 27b^2 \neq 0$ erfüllt, so ist die zugehörige elliptische Kurve singularitätenfrei und besitzt daher keine mehrfachen Nullstellen. Ein Beweis dieser Bedingung ist in ([10], Lemma 22.2., S. 200) zu finden.
2) Die Kurze Weierstraß-Gleichung existiert nur, falls $char(K) \notin \{2, 3\}$ gilt ([2], S. 24-25).
3) Ist die elliptische Kurve $\overline{E}(K)$ durch eine kurze Weierstraß-Gleichung gegeben, so ist die horizontale Symmetrieachse die $x-$Achse.

Definition (**Punktaddition**)**.** Um zwei Punkte P und Q einer elliptischen Kurve $\overline{E}(K)$ miteinander zu verknüpfen verbindet man diese mittels einer Geraden(Tangente). Dabei schneidet die daraus entstehende Gerade(Tangente) die Kurve in genau einem weiteren Punkt R'. Spiegelt man nun diesen Punkt an der horizontalen Symmetrieachse von $\overline{E}(K)$ erhält man einen weiteren Punkt R. Die Summe der Punkte P und Q und wird nun wie folgt definiert: $P \oplus Q := R$. Man spricht hier auch von einer Punktaddition. Im Zuge dieser Punktaddition ist der unendlich ferne Punkt O dadurch definiert, dass jede Gerade, welche durch O führt, senkrecht ist und dass jede senkrechte Gerade durch O geht.

Abbildung 3 zeigt eine derartige Verknüpfung zweier Punkte der elliptischen Kurve $y^2 = x^3 - 2x + 1$, wobei die horizontale Symmetrieachse hier die $x-$Achse ist. Die Existenz eines dritten Schnittpunkts mit der Geraden muss noch geklärt werden. Folgende Sätze konkretisieren diese Idee. Der nächste Satz zeigt die Existenz und Eindeutigkeit der Inversen und rechtfertigt obige Definition. ([9], S. 288-289, [1], S. 147, [2], S. 42)

Satz 2.33. *Sei (v, w) ein Punkt einer elliptischen Kurve $\overline{E}(K)$. Dann existiert maximal ein weiterer Punkt (v, w') auf $\overline{E}(K)$ mit derselben $x-$Koordinate. Für diesen Punkt gilt $(v, w') = (v, -a_1 v - a_3 - w) \in \overline{E}(K)$. Sollte es keinen von (v, w) verschiedenen Punkt mit der gleichen $x-$Koordinate geben, so ist $w' = w$ und für die partielle Ableitung gilt $\frac{\partial F}{\partial y}(v, w) = 0$.*

Beweis. Die elliptische Kurve $\overline{E}(K)$ ist durch die Gleichung $F(x, y) = 0$ bestimmt. Wählt man nun ein fixes v für die $x-$Koordinate, also gilt $x = v$, so erhält man eine neue Gleichung in y der Gestalt

$$\begin{aligned} F(y) &= y^2 + a_1 v y + a_3 y - v^3 - a_2 v^2 - a_4 v - a_6 = 0 \\ &= y^2 + \underbrace{(a_1 v + a_3)}_{=p} y + \underbrace{\left(-v^3 - a_2 v^2 - a_4 v - a_6\right)}_{=q} = 0 \end{aligned}$$

wobei man den Koeffizienten des linearen Glieds als p und das absolute Glied als q bezeichnet. Nach dem Fundamentalsatz der Algebra hat diese Gleichung maximal zwei Lösungen. Dies bedeutet, dass es maximal zwei verschiedene $y-$Koordinaten für ein fixes v geben kann. Betrachtet man einen Punkt (v, w) der Kurve, so ist w eine Lösung der Gleichung $F(y)$. Nach dem Satz von Vieta gilt für eine quadratische Gleichung $y^2 + py + q = 0$ mit Lösungen y_1 und y_2 die Bedingung $y_1 + y_2 = -p$. Bezogen auf $F(y)$ gilt für die zweite Lösung w' daher

$$w + w' = -a_1 v - a_3 \iff w' = -a_1 v - a_3 - w$$

Es bleibt noch zu zeigen, ob der Punkt (v, w') nicht singulär ist, damit dieser ein Element der elliptischen Kurve ist. Nach Satz 2.32. müsste (v, w') ein mehrfacher Schnittpunkt der durch $F(x, y) = 0$ festgelegten algebraischen Kurve und aller Geraden durch diesen Punkt sein. Wählt man eine Gerade $x = v$, so müsste man zumindest einen zweifachen Schnittpunkt mit der Kurve in (v, w) erhalten. Jedoch hat man oben bereits gezeigt, dass es maximal zwei Punkte mit der gleichen $x-$Koordinate geben kann. Dies bedeutet, dass $(v, w) = (v, w')$ gelten müsste, womit diese zwei Punkte ident wären. Daher wäre (v, w) ein singulärer Punkt, womit man einen Widerspruch zur Voraussetzung hätte, dass (v, w) ein Punkt der elliptischen Kurve $\overline{E}(K)$ sei. Sollten (v, w) und $(v, w') = (v, -a_1 v + a_3 - w)$ ident sein, so gilt klarerweise $w = -a_1 v - a_3 - w$, wodurch mittels Umformung folgt $2w + a_1 v + a_3 = 0$. Da die partielle Ableitung $\frac{\partial F}{\partial y}(v, w) = 2w + a_1 v + a_3$ ergibt, ist diese somit ebenfalls 0, wodurch der Satz bewiesen wäre. ([9], Lemma 23.3, S. 289-290) □

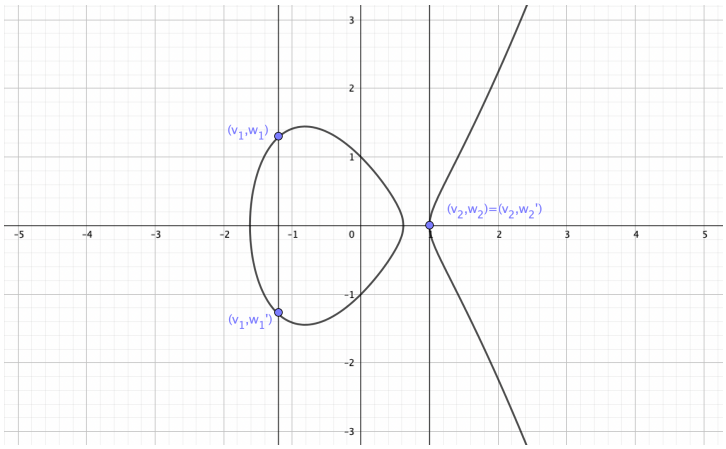

Abbildung 4: Inverse Elemente auf der elliptischen Kurve $y^2 = x^3 - 2x + 1$ in GeoGebra

Abbildung 4 zeigt die zueinander inversen Elemente (v_1, w_1) und (v_1, w_1') und das Element (v_2, w_2), welches zu sich selbst invers ist, weswegen auch eine Tangente durch diesen Punkt geht. Bis jetzt wurde gezeigt, dass zwei Elemente mit der gleichen $x-$Koordinate zueinander invers sind, auch wenn diese Elemente ident wären. Die senkrechte Gerade durch diese Elemente geht nach Definition ebenfalls durch einen dritten Punkt, nämlich dem unendlich fernen Punkt O. Es bleiben noch zwei Fälle zu überprüfen, wobei der erste Fall untersucht, ob Elemente, welche nicht direkt übereinander liegen, also keine gleiche $x-$Koordinate haben, genau einen dritten Schnittpunkt besitzen, damit man die Definition für die Verknüpfung zu Ende führen kann.

Satz 2.34. *Seien (v_1, w_1) und (v_2, w_2) zwei Punkte einer elliptischen Kurve $\overline{E}(K)$, wobei $v_1 \neq v_2$ gilt. Legt man eine Gerade durch diese zwei Punkte, so erhält man genau einen dritten Schnittpunkt (v_3, w_3) mit $\overline{E}(K)$. Es gilt nun für die Verknüpfung von $(v_1, w_1) \oplus (v_2, w_2) := -(v_3, w_3)$ mit*

$$\begin{aligned} k &:= \frac{w_1 - w_2}{v_1 - v_2} \\ d &:= w_1 - kv_1 \\ v_3 &:= k^2 + a_1 k - a_2 - v_1 - v_2 \\ w_3 &:= kv_3 + d \end{aligned}$$

Unter $-(v_3, w_3)$ wird das Inverse von (v_3, w_3) bezüglich der Punktaddition auf der elliptischen Kurve verstanden.

Beweis. Betrachtet man nun die Punkte (v_1, w_1) und (v_2, w_2), so ergibt sich für die Gerade durch diese Punkte die Funktionsgleichung $y = kx + d$, wobei sich mit $k = \frac{w_1 - w_2}{v_1 - v_2}$ die Steigung der Geraden und durch Einsetzen $d = w_1 - kv_1$ ergibt. Da sich die $x-$Koordinaten unterscheiden handelt es sich hierbei nicht um eine senkrechte Gerade. Setzt man die Geradengleichung in

$F(x,y)=0$ ein, so erhält man eine Polynomgleichung in x der Gestalt:

$$\begin{aligned}
F(x) =&(kx+d)^2 + a_1x(kx+d) + a_3(kx+d) - x^3 - a_2x^2 - a_4x - a_6 = 0\\
=&k^2x^2 + 2kxd + a_1x^2k + a_1xd + a_3kx + a_3d - x^3 - a_2x^2 - a_4x = 0\\
=&-x^3 + \left(k^2 + a_1k - a_2\right)x^2 + (2kd + a_1d + a_3k - a_4)\,x + a_3d = 0\\
\Longrightarrow&x^3 + \underbrace{\left(-k^2 - a_1k + a_2\right)}_{=a}x^2 + \underbrace{(-2kd - a_1d - a_3k + a_4)}_{=b}x - \underbrace{a_3d}_{=c} = 0
\end{aligned}$$

Klarerweise muss diese kubische Gleichung drei Lösungen haben, wobei man bereits die Lösungen v_1 und v_2 kennt. Für den dritten Schnittpunkt ergibt sich abermals durch Anwendung des verallgemeinerten Satzes von Vieta für kubische Gleichungen

$$-(v_1+v_2+v_3) = a = -k^2 - a_1k + a_2 \Longrightarrow v_3 = k^2 + a_1k - a_2 - v_1 - v_2$$

Unter Verwendung der Geradengleichung ist man nun in der Lage die $y-$Koordinate des Punktes (v_3, w_3) zu berechnen. Man erhält dabei $w_3 = kv_3 + d$. Somit wurde gezeigt, dass dieser dritte Schnittpunkt stets existiert und darüber hinaus eindeutig bestimmt ist. □

In Abbildung 3 ist die definierte Verknüpfung, bei welcher es sich um eine Punktaddition handelt, ersichtlich. Der letzte Fall, der behandelt werden muss, ist, falls die Punkte $(v_1, w_1) = (v_2, w_2)$ ident sind, aber für die partielle Ableitung in diesem Punkt $\frac{\partial F}{\partial y}(v_1, w_1) \neq 0$ gilt. Bezogen auf die Verknüpfung würde man hier zwei gleiche Punkte addieren, während man bildlich gesprochen, eine Gerade durch die identen Punkte $(v_1, w_1) = (v_2, w_2)$ legen würde, das heißt man erhält eine Tangente durch diesen Punkt. Mittels umformen und implizitem differenzieren von $F(x,y)$ erhält man die Steigung der Tangente:

$$\begin{aligned}
&y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\\
&\Longrightarrow 2yy' + a_1y + a_1xy' + a_3y' = 3x^2 + 2a_2x + a_4\\
&\Longleftrightarrow y' = \frac{3x^2 + 2a_2x + a_4 - a_1y}{2y + a_1x + a_3}
\end{aligned}$$

Setzt man nun den Punkt (v_1, w_1) ein, erhält man unter Berücksichtigung der in Satz 2.32. angeführten partiellen Ableitungen

$$k = y' = \frac{3v_1^2 + 2a_2v_1 + a_4 - a_1w_1}{2w_1 + a_1v_1 + a_3} = -\frac{\frac{\partial F}{\partial x}(v_1, w_1)}{\frac{\partial F}{\partial y}(v_1, w_1)}$$

Man bekommt damit eine Tangentengleichung der Gestalt $y = kx + d$, wobei neben den gerade berechnetem k sich mittels einsetzen von (v_1, w_1) in die Tangentengleichung $d = w_1 - kv_1$ ergibt. Würde man nun abermals diese Gleichung in $F(x,y) = 0$ einsetzen, so würde man eine kubische Polynomgleichung in x erhalten. Der einzige Unterschied besteht darin, dass man anstelle einer

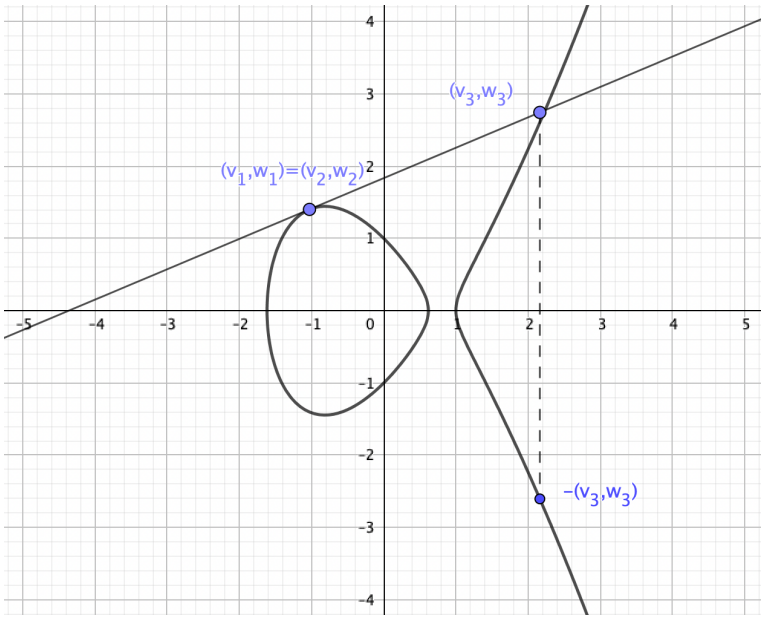

Abbildung 5: Verknüpfung identischer Punkte auf der elliptischen Kurve $y^2 = x^3 - 2x + 1$ in GeoGebra

Geraden durch zwei verschiedene Punkte, anschaulich eine „Gerade durch zwei idente Punkte“ legt, womit man eine Tangente erhält. Da man hier zwei idente Nullstellen des Polynoms $F(x)$ bereits kennt, bekommt man die Koordinaten des dritten Schnittpunktes (v_3, w_3) mit denselben Formeln wie in der vorangehenden Definition, mit Ausnahme von k, das wie oben angeführt mit den partiellen Ableitungen berechnet wird. ([9], S. 292-293) In Abbildung 5 ist der letzte Fall angeführt, wobei wiederum die gleiche elliptische Kurve $\overline{E}(K)$ gewählt wurde. Insgesamt ist gezeigt, dass die Punktaddition auf elliptischen Kurven wohldefiniert ist und dabei haben wir auch eine Darstellung der Punktaddition mittels Koordinaten erhalten.

Satz 2.35 (**Punktaddition in elliptischen Kurven**). *Es sei $E(K)$ eine elliptische Kurve und $(v_1, w_1), (v_2, w_2) \in E(K)$ seien beliebige Punkte auf dieser Kurve. Man definiert:*

1) $O \oplus O := O$

2) $(v_1, w_1) \oplus O := O \oplus (v_1, w_1) := (v_1, w_1)$

3) $(v_1, w_1) \oplus (v_2, w_2) := (v_2, w_2) \oplus (v_1, w_1) := O$ *wenn* $v_1 = v_2$ *und* $w_1 \neq w_2$
oder wenn $(v_1, w_1) = (v_2, w_2)$ *und* $\frac{\partial F}{\partial y}(v_1, w_1) = 0$

4) Alle anderen Fälle betreffend gilt: $(v_1, w_1) \oplus (v_2, w_2) := (v_3, a_1 v_3 - a_3 - w_3)$, *mit*

$$k := \begin{cases} \frac{w_1 - w_2}{v_1 - v_2} & \text{wenn } v_1 \neq v_2 \\ -\frac{\frac{\partial F}{\partial x}(v_1, w_1)}{\frac{\partial F}{\partial y}(v_1, w_1)} & \text{wenn } (v_1, w_1) = (v_2, w_2) \text{ und } \frac{\partial F}{\partial y}(v_1, w_1) \neq 0 \end{cases}$$

$$v_3 := k^2 + a_1 k - a_2 - v_1 - v_2$$

$$w_3 := kv_3 + (w_1 - kv_1) = w_1 + k(v_3 - v_1)$$

(vgl. [9], Definition 23.5, S. 294)

Es bleibt zu zeigen, dass $\overline{E}(K)$ bezüglich der Punktaddition eine kommutative Gruppe bildet. Der unendlich ferne Punkt O ist das neutrale Element, was direkt aus der Definition von O folgt. Der nächste Satz 2.36. zeigt, dass jeder Punkt von $\overline{E}(K)$ ein eindeutiges Inverses hat.

Satz 2.36. *Seien $P = (v, w), Q = (v, w') \in \overline{E}(K)$ zwei Punkte einer elliptischen Kurve, wobei $w \neq w'$. Es gilt $(v, w) \oplus (v, w') := (v, w') \oplus (v, w) := O$. Falls $w = w'$ und die partielle Ableitung $\frac{\partial F}{\partial y}(v, w) = 0$ gilt, dann ist $(v, w) \oplus (v, w') := O$. ([9], Definition 23.2, S. 289)*

Beweis. Haben zwei Punkte P und Q dieselbe $x-$Koordinate, womit sie genau übereinander liegen, so ist die Verbindungsgerade senkrecht. Nach obiger Definition ergibt die Addition dieser Punkte das neutrale Element O, wodurch klarerweise P und Q invers zueinander sind. Ist darüber hinaus auch die $y-$Koordinate gleich, so verstehen wir unter der Gerade, welche einen Punkt P mit sich selbst verbindet, eine Tangente. Diese ist genau dann senkrecht, wenn die partielle Ableitung nach x in diesem Punkt 0 ergibt. Demzufolge gilt $P \oplus P = O$ und daher ist P definitionsgemäß zu sich selbst invers. □

Zusätzlich ist die Verknüpfung kommutativ, wie man mittels einiger Rechnungen nachprüfen könnte. Die Kommutativität folgt direkt aus der Definition der Punktaddition, da es egal ist ob man die Gerade zuerst durch den einen Punkt und dann durch den anderen Punkt legt oder umgekehrt. Wie man aus der Geometrie weiß, ist eine Gerade durch zwei Punkte eindeutig bestimmt. Bis dato konnte somit gezeigt werden, dass die Verknüpfung ein neutrales und inverses Element besitzt und darüber hinaus auch kommutativ ist. Es fehlt lediglich die Assoziativität, also dass für drei Punkte $P, R, Q \in \overline{E}(K)$ stets die Eigenschaft $P \oplus (Q \oplus R) = (P \oplus Q) \oplus R$ gilt. Der Beweis für die Assoziativität ist jedoch sehr aufwändig und mühsam, weswegen hier darauf verzichtet und auf die Referenz ([9], S. 294-303) verwiesen wird. Demnach gilt auch die Assoziativität, wodurch mit der gewählten Verknüpfung eine elliptische Kurve $\overline{E}(K)$ eine abelsche Gruppe ist. ([9], S. 294)

2.4 Grundlagen Informatik

Die in dieser Arbeit vorgestellten Faktorisierungsalgorithmen werden auf leistungsstarken Computern verwirklicht. Um diverse Anwendungen im Zusammenhang mit diesen besser verstehen zu können, werden in diesem Abschnitt einige informatische Grundlagen angeführt.

2.4.1 Bit und Byte

Im Gegensatz zum Computer wird im Alltag das Zehnersystem, bestehend aus 10 Ziffern verwendet. So wird beispielsweise die Zahl $243 = 2 \cdot 10^2 + 4 \cdot 10^1 + 3 \cdot 10^0$ mithilfe von Zehnerpo-

tenzen angeschrieben. Das Zehnersystem, auch dekadisches System genannt, hat als Basis b die Zahl 10, wodurch die Ziffern von $0, \ldots, b-1$, also $0, \ldots, 9$ gebraucht werden. Der Computer hingegen arbeitet mit der Basis $b = 2$, woduch lediglich mit zwei unterschiedlichen Zahlen, nämlich mit Nullen und Einsen, gearbeitet wird. Die Zahl 23 wird in diesem Zweiersystem wie folgt dargestellt:

$$23 = \mathbf{1} \cdot 2^4 + \mathbf{0} \cdot 2^3 + \mathbf{1} \cdot 2^2 + \mathbf{1} \cdot 2^1 + \mathbf{1} \cdot 2^0 = 16 + 4 + 2 + 1 \Rightarrow 23 = \mathbf{10111}$$

Dabei wird jede Zahl als Summe von Zweierpotenzen geschrieben, wobei man für jede vorkommende Zweierpotenz eine 1 und für jede nicht darin Auftretende eine 0 schreibt. Im Beispiel $23 = 10111$ steht die 0 für die nicht vorkommende Zweierpotenz 2^3. Sowohl im Zehnersystem, als auch im Zweiersystem ist der Stellenwert der einzelnen Ziffern ausschlaggebend. Durch das Zweiersystem dargestellte Zahlen nennt man Binärzahlen. Das Zweiersystem wird daher auch Binärsystem genannt. Als *Bits* bezeichnet man nun die Ziffern 0 und 1. Eine Folge von 8 Bit, wie beispielsweise 10101010, nennt man *Byte*. Die Größe von Computerspeichern wird in Byte angegeben, weil dieser mindestens 8 Bit gleichzeitig speichert. Zum Beispiel ist ein Megabyte ein Speicher von einer Millionen Bytes oder 8 Millionen Bits. ([5], S. 45)

2.4.2 Algorithmus-Laufzeit

Als Algorithmus bezeichnet man einen deterministischen Lösungsweg mit einer variablen Eingabe, welcher nach einer endlich langen Zeit eine bestimmte Ausgabe liefert ([1], S. 45). Bei der Erstellung von Algorithmen ist es notwendig den Berechnungsaufwand und den benötigten Speicherplatz bereits im vorhinein adäquat abzuschätzen ([8], S. 7). Um Algorithmen zu beschreiben bedient man sich unterschiedlicher Modelle, wobei das gängigste das der Turing-Maschine ist. Unter einer Turing-Maschine kann man sich einen sehr vereinfachten Computer vorstellen, welcher eine bestimmte Anzahl an Speicherzellen hat und die entsprechenden Inhalte dieser lesen oder schreiben kann. Die Turing-Maschine führt nun eine Zustandskontrolle eines zuvor festgelegten Programms aus, in welchem gelesen oder geschrieben wird. Diese Kontrolle kann man auch als Algorithmus verstehen, in dem dieser die Reihenfolge des Schreibens und Lesens vorgibt. Jeder einzelne Berechnungsschritt einer Speicherzelle wird dabei als Operation bezeichnet. Sämtliche in der heutigen Zeit verwendete Rechner lassen sich aus diesem Modell ableiten. Im Wesentlichen interessiert man sich jetzt für die Laufzeit L eines Algorithmus. Unter dieser versteht man die Anzahl der Operationen $O(n)$, welche absolviert werden müssen, um bei einer bestimmten Eingabe n eine Ausgabe zu erhalten. Dabei wird selten die genaue Laufzeit angegeben, sondern vielmehr wird diese mittels einer oberen Schranke nach oben abgeschätzt. Diese obere Schranke oder *worst case Laufzeit* wird auch als Komplexität bezeichnet. Man sagt ein Algorithmus ist polynomiell, wenn dessen Laufzeit mittels eines Polynoms in der Eingabelänge nach oben abschätzbar ist. Sind die einzelnen Paramter nicht allzu groß, so lässt sich ein Algorithmus mit polynomieller Laufzeit vergleichsweise schnell ausführen und

gilt daher als effizient zu lösen. ([1], S. 45-46). Die nachfolgende Tabelle zeigt die Laufzeiten verschiedener Algorithmen, wobei der hierbei verwendete Computer für eine Operation 10^{-6} Sekunden benötigt hat:

Laufzeitenvergleich (Quelle: [1], S. 46)

	10	20	30	40	50	60
n	10^{-5} Sek.	$2 \cdot 10^{-5}$ Sek.	$3 \cdot 10^{-5}$ Sek.	$4 \cdot 10^{-5}$ Sek.	$5 \cdot 10^{-5}$ Sek.	$6 \cdot 10^{-5}$ Sek.
n^2	10^{-4} Sek.	$4 \cdot 10^{-4}$ Sek.	$9 \cdot 10^{-4}$ Sek.	0,0016 Sek.	0,0025 Sek.	0,0036 Sek.
n^3	0,001 Sek.	0,008 Sek.	0,027 Sek.	0,064 Sek.	0,125 Sek.	0,216 Sek.
2^n	0,001 Sek.	1 Sek.	17,9 Min.	12,7 Tage	35,7 Jahre	366 Jhrt.
e^n	0,02 Sek.	8,1 Min.	123,7 Tage	74,6 Jhrt.	$1,6 \cdot 10^6$ Jhrt.	$3,6 \cdot 10^{10}$ Jhrt.

In der vorliegenden Tabelle werden unterschiedliche Laufzeiten $(n^2, \ldots, e^n)$ unter Einbeziehung unterschiedlicher Eingabelängen $(10, \ldots, 60)$ angeführt. Es ist sofort ersichtlich, dass sich die polynomiellem Laufzeiten n, n^2 und n^3 auch bei sehr großen Eingabelängen noch sehr schnell durchführen lassen. Bei den exponentiellen Laufzeiten 2^n und e^n hingegen führt beispielsweise eine Änderung der Eingabenlänge von 10 auf 40 bereits zu einer enormen Erhöhung der Laufzeit. Die Multiplikation natürlicher Zahlen lässt sich beispielsweise mit polynomiellen Zeitaufwand lösen, bei deren Umkehrung jedoch, also der Faktorisierung natürlicher Zahlen, ist bis dato noch kein Algorithmus mit polynomieller Laufzeit bekannt. Zwar gibt es einige Algorithmen dafür, welche dieses Problem in subexponentieller Laufzeit lösen können, also schneller sind als jene mit exponentieller Laufzeit, jedoch sind diese immer noch nicht polynomiell und somit nicht effizient durchführbar. Es existieren jedoch auch Algorithmen welche nur mit einer bestimmten Wahrscheinlichkeit eine richtige Ausgabe liefern. Diese Algorithmen nennt man probabilistisch. Einige Primzahltests, also Algorithmen welche bei großen Zahlen prüfen, ob diese eine Primzahl ist, sind probabilistisch und lassen sich effizient durchführen. Es gibt auch probabilistische Faktorisierungsalgorithmen, wie beispielsweise die Pollard'sche Rho-Methode, welche in Abschnitt 4 näher beschrieben wird. Zusammengefasst lässt sich jedoch sagen, dass die Durchführbarkeit eines Algorithmus von seiner Laufzeit und der Größe des benötigten Speicherplatzes abhängt. ([1], S. 46-47)

2.4.3 Schnelle Exponentation

Zahlreiche Anwendungen der Kryptologie, unter anderem diverse Faktorisierungsalgorithmen, benötigen das Berechnen großer Potenzen. Um diese Berechnungen zu erleichtern wird in diesem

Abschnitt ein Verfahren dargestellt, dass auch als *Square-and-Multiply-Algorithmus* bekannt ist. Mit der naiven Methode würde man für eine Potenz a^m mittels wiederholtem multiplizieren mit a schrittweise $a^2, a^3, \ldots$ solange berechnen, bis man bei a^m angelangt ist. Offensichtlich sind hierbei $m-1$ Multiplikationen von Nöten. Um beispielsweise eine verhältnismäßig kleine Potenz wie a^{32} zu berechnen benötigt man 31 Multiplikationen. Stattdessen kann mittels wiederholter Quadrierungen zuerst die Potenz a^2, dann a^4, a^8, a^{16} und schließlich a^{32} berechnet werden. Hierbei werden nun statt 31 Multiplikationen lediglich 5 benötigt. Das Verfahren funktioniert auch wenn der Exponent nicht aus einer Zweierpotenz besteht. ([10], S. 13) Sei dazu

$$m = \sum_{i=0}^{\ell} b_i 2^i, \quad b_i \in \{0,1\}, b_\ell = 1$$

die Binär-Darstellung des Exponenten m. Als ℓ bezeichnet man hier die Bit-Länge, also die Anzahl der Binärstellen von m. Nun kann die Potenz a^m wie folgt geschrieben werden:

$$a^m = a^{\sum_{i=0}^{\ell} b_i 2^i} = \prod_{i=0}^{\ell} \left(a^{2^i}\right)^{b_i} = \prod_{0 \le i \le \ell, b_i = 1} a^{2^i}$$

Durch diese Umformung ergibt sich folgender Algorithmus:
1. Sukzessives Quadrieren: $a^{2^i}, 0 \le i \le \ell$.
2. Berechne das Produkt a^m aus a^{2^i}, für die gilt: $b_i = 1$
Dabei gilt es zu bedenken, dass die Potenz $a^{2^{i+1}}$ wie folgt berechnet werden kann:

$$a^{2^{i+1}} = \left(a^{2^i}\right)^2$$

Beispiel. Zu berechnen ist die Potenz 7^{71} mod 100. Der Exponent 71 wird in Binärschreibsweise als 100111 angegeben, das heißt $71 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$. Als nächstes werden sukzessiv die dafür notwendigen Quadrate berechnet: $7^1 (\text{mod} 100) \equiv 7, 7^2 \equiv 49 (\text{mod } 100), 7^{2^2} \equiv 49^2 \equiv 2401 \equiv 1 (\text{mod} 100), 7^{2^3} \equiv 1^2 \equiv 1 (\text{mod} 100), 7^{2^4} \equiv 1^2 \equiv 1 (\text{mod} 100), 7^{2^5} \equiv 1^2 \equiv 1 (\text{mod} 100)$. Somit ist:

$$7^{71} = 7 \cdot 7^2 \cdot 7^{2^2} \cdot 7^{2^5} \equiv 7 \cdot 49 \cdot 1 \cdot 1 \equiv 343 \equiv 43 \bmod 100$$

Anstelle von 70 Multiplikationen modulo 100 wurden hier lediglich 5 Quadrierungen und 3 Multiplikationen verwendet. ([8], S. 38-39) Generell entspricht die Zahl der benötigten Quadrierungen der Anzahl der Bits der entsprechenden Zahl [9], S. 342). Das Beispiel zeigt, dass durch die Anwendung des Square-and-Multiply-Algorithmus die Laufzeit enorm verringert werden kann. Das im folgenden Abschnitt dargestellte RSA-Verschlüsselungsverfahren nutzt diesen Algorithmus um eine Nachricht m mit dem entsprechenden öffentlichen Schlüssel (n, e) zu verschlüsseln, wobei e der Exponent und n der Modul ist. Die Verschlüsselungsvorschrift lautet $m^e (\text{mod} n)$. In der heutigen Zeit haben sowohl die Nachricht m als auch der Exponent e eine Größenordnung von 2048 Bit, entsprechend einer Zahl mit ungefähr 600 Stellen, um

genügend Sicherheit zu gewährleisten. Hierfür müsste man also mit der naiven Methode für eine Zahl mit ungefähr 600 Dezimalstellen $10^{600}(\approx 2^{2048}$)Multiplikationen mit der Zahl selbst durchführen, wohingegen durch Anwendung des Square-and-Multiply-Algorithmus nur rund 10^{10} Multplikationen notwendig wären. Die besten Resultate liefert der Square-and-Multiply-Algorithmus, falls die Basis a relativ klein und der Exponent m der Potenz a^m in seiner Binärschreibweise nur wenige Einsen enthält. Dies sollte bei der Wahl des öffentlichen Schlüssels berücksichtigt werden. ([1], S. 119-120)

3 RSA

3.1 Einführung-Asymmetrische Verschlüsselung

Lange Zeit wurden in der Kryptografie hauptsächlich symmetrische Verschlüsselungsverfahren angewendet. Diese Verfahren benötigen zur Verschlüsselung einer Nachricht einen geheimen Schlüssel, welcher aber gleichzeitg auch zur Entschlüsselung verwendet wird. Mit der Kenntnis des geheimen Schlüssels ist man also in der Lage sowohl die Ver- als auch die Entschlüsselung auszuführen. Die große Problematik bei symmetrischen Verschlüsselungsverfahren ist der Schlüsselaustausch. Bevor eine Kommunikation stattfinden kann, muss ein geheimer Schlüssel zwischen den kommunizierenden Personen vereinbart werden. Dies ist mit erheblichem Aufwand und Sicherheitsrisiken verbunden, da der geheime Schlüssel jederzeit von einem Angreifer abgefangen werden kann. Die Cäsar-Verschlüsselung oder die im zweiten Weltkrieg verwendete Enigma sind Vertreter der symmetrischen Verschlüsselung. ([12], S. 325-327) Um eine vertrauliche Kommunikation ohne der oben erwähnten Problematiken gewährleisten zu können, entwickelten Withfield Diffie und Martin E. Hellman das Modell der asymmetrischen Verschlüsselung, sogenannter Public-Key-Verfahren. Bei der Anwendung derartiger Verfahren verwenden Sender und Empfänger zwei unterschiedliche Schlüssel, nämlich einen privaten und einen öffentlichen Schlüssel. Möchte beispielsweise Alice eine geheime Nachricht an Bob schicken, benötigt sie dazu den öffentlichen Schlüssel von Bob, welcher für alle Personen frei zugänglich ist. Mit diesem öffentlichen Schlüssel chiffriert Alice ihre Nachricht und sendet sie an Bob, wobei diese Nachricht nicht unbedingt geheim gehalten werden muss. Ausschließlich Bob, welcher im Besitz des privaten Schlüssels ist, kann diese Geheimnachricht nun wieder entschlüsseln. Falls Bob eine Anwort an Alice senden möchte, benutzt er ihren öffentlichen Schlüssel und geht dabei gleichermaßen vor. Der Vorteil der asymmetrischen Kryptografie liegt auf der Hand, denn die Problematik des Schlüsselaustausches wäre somit eliminiert, denn es müssen keine Schlüssel mehr ausgetauscht werden. ([4], S. 10-11)

3.2 RSA-Algorithmus

Das geläufigste asymmetrische Verschlüsselungsverfahren ist der nach seinen Erfindern (Ron **R**ivest, Adi **S**hamir und Leonard **A**dleman) benannte *RSA-Algorithmus*. Die Sicherheit dieses

Algorithmus baut genau auf jenem Grundproblem der Zahlentheorie auf, welchem sich diese Arbeit widmet, nämlich der Faktorisierung natürlicher Zahlen. Während es relativ leicht ist, zwei große Zahlen miteinander zu multiplizieren, gibt es nach wie vor keinen Algorithmus, der effizient genug wäre um das Produkt dieser Zahlen zu faktorisieren. ([1], S. 115) Bereits seit 2000 Jahren versuchen die Mathematiker eine effiziente Lösung für das Faktorisierungsproblem zu finden, wobei mittlerweile davon ausgegangen wird, dass dieses Problem naturgemäß nicht zu lösen sei ([12], S. 335-336). Man geht also bei der Multiplikation natürlicher Zahlen davon aus, dass es sich um eine Einwegfunktion handelt, also das Bild $y = f(x)$ simpel berechnet werden kann, im Gegensatz zum Urbild $x = f^{-1}(y)$ ([4], S. 11).

3.2.1 Ablauf

Zu Beginn werden zwei große Primzahlen p und q ausgewählt und man bildet das Produkt $n = p \cdot q$. Üblicherweise wählt man heutzutage Primzahlen in der Größenordnung von 1024 Bit, wodurch der entstehende Modul n eine Größenordnung von 2048 Bit aufweist. Mittels Primzahltests, wie beispielsweise dem Miller-Rabin-Test, kann überprüft werden, ob es sich bei den ausgewählten Zahlen auch wirklich um Primzahlen handelt. Anschließend berechnet man die Eulersche ϕ-Funktion $\phi(n) = (p-1) \cdot (q-1)$. Als nächstes wird eine Zahl e so ausgewählt, sodass $gcd(e, \phi(n)) = 1$. Zu guter Letzt wird eine Zahl d berechnet mit $e \cdot d \equiv 1 (\mathrm{mod} \phi(n))$. Für die Berechnung von d wird der Erweiterte Euklidische Algorithmus herangezogen. Nun bildet das Zahlenpaar (n, e) den öffentlichen Schlüssel und die Zahl d den privaten Schlüssel. Es ist sehr wichtig die Parameter $p, q, \phi(n)$ und d geheim zu halten. Um nun eine Nachricht m zu verschlüsseln geht der Sender der Nachricht wie folgt vor: In einem öffentlichen Verzeichnis sucht dieser das öffentliche Schlüsselpaar (n, e) des Empfängers und verschlüsselt damit seine Nachricht:

$$c := m^e (\mathrm{mod} n)$$

Notwendigerweise muss $m < n$ sein, andernfalls kann man die Nachricht m in mehrere kleinere Blöcke zerlegen. Nun kann der Geheimtext c an den jeweiligen Empfänger gesendet werden, der mit seinem privaten Schlüssel dazu in der Lage ist den Geheimtext wieder zu entschlüsseln:

$$c^d (\mathrm{mod} n) = m$$

Um zu antworten wiederholt der Empfänger den geschilderten Prozess mit dem öffentlichem Schlüsselpaar des Senders. ([1], S. 115-119, [4], S. 11-12)

3.2.2 Verifikation des RSA-Algorithmus

Der folgende Beweis orientiert sich an ([5], S. 90). Nachdem der Ablauf der Verschlüsselung geschildert wurde, bleibt noch zu zeigen, warum der RSA-Algorithmus überhaupt funktioniert und ob mittels des privaten Schlüssels der Geheimtext eindeutig dechiffriert werden kann. Zu

zeigen ist:

$$c^d \equiv (m^e)^d \equiv m(\mathrm{mod} n)$$

Beweis. Der öffentliche Teil des Schlüssepaares e und der private Schlüssel wurden wie folgt bestimmt:

$$ed \equiv 1(\mathrm{mod}\phi(n))$$

Also sind die Zahlen e und d bezüglich der Multiplikation modulo $\phi(n)$ invers zueinander. Für eine natürliche Zahl k kann dies auch umgeschrieben werden zu:

$$e \cdot d = k \cdot \phi(n) + 1$$

Verwendet man nun den Modifizierten Satz von Euler bekommt man

$$c^d \equiv (m^e)^d \equiv m^{e \cdot d} \equiv m^{k \cdot \phi(n)+1} \equiv m(\mathrm{mod} n)$$

womit die Korrektheit des RSA-Algorithmus nachgewiesen wäre. □

3.2.3 Beispiel

Beispielsweise soll die Nachricht „geheim“ verschlüsselt werden. Dazu muss das Wort zunächst in Form einer Zahl angegeben werden, in dem jeder Buchstabe eindeutig einer Zahl zugeordnet wird. Die Buchstaben werden folgenderweise codiert: $A = 01, B = 02, C = 03, \ldots, Z = 26$ Dabei ist es wichtig bei den einstelligen Zahlen die vorangehenden Nullen ebenfalls zu schreiben und in der Berechnung zu berücksichtigen. Somit lautet die Nachricht:

$$\text{geheim} \rightarrow 0705 \quad 0805 \quad 0913$$

Wie bereits erwähnt wird je nach Größe des Moduls die codierte Nachricht in Blöcke geteilt. Dazu wird diese in Blöcke zu je 4 Zahlen zerlegt. Im Anschluss werden die Primzahlen $p = 23$ und $q = 97$ festgelegt, wodurch man als Modul $n = 23 \cdot 97 = 2231$ erhält. Als Eulersche ϕ-Funktion von n erhält man nun $\phi(n) = (p-1) \cdot (q-1) = 22 \cdot 96 = 2112$. Nachdem eine Zahl e für den öffentlichen Schlüssel mit $e = 13$ und $gcd(e, \phi(n)) = gcd(13, 2112) = 1$ bestimmt wurde, berechnet man den geheimen Schlüssel d mit $e \cdot d \equiv 1 \bmod \phi(n)$. Dazu kann der Erweiterte Euklidische Algorithmus herangezogen werden:

$$2112 = 162 \cdot 13 + 6$$
$$13 = 2 \cdot 6 + 1$$
$$6 = 1 \cdot 6 + 0$$

Mit dem Lemma von Bézout erhält man:

$$\begin{aligned} 1 &= 13 - 2 \cdot 6 \\ 1 &= 13 - 2 \cdot (2112 - 162 \cdot 13) = -2 \cdot 2112 + 325 \cdot 13 \\ \Rightarrow d &= 325 \end{aligned}$$

Für das öffentliche Schlüsselpaar ergibt sich $(n, e) = (2231, 13)$ und für den privaten Schlüssel $d = 325$. Nun kann mit der Verschlüsselung begonnen werden, wobei zur Berechnung der Potenzen der bereits erläuterte Square-and-Multiply-Algorithmus angewendet werden kann:

$$\begin{aligned} c_1 &= 0705^{13} \bmod 2231 = 0718 \\ c_2 &= 0805^{13} \bmod 2231 = 2024 \\ c_3 &= 0913^{13} \bmod 2231 = 0716 \end{aligned}$$

Die Geheimnachricht in Ziffernfolge lautet daher 071820240716. Um diese Nachricht zu entschlüsseln benötigt man den privaten Schlüssel $d = 325$ und berechnet abermals in Blöcken zu je 4 Zahlen $m_i = {c_i}^d \bmod n$, $1 \leqslant i \leqslant 3$:

$$\begin{aligned} m_1 &= 0718^{325} \bmod 2231 = 0705 \\ m_2 &= 2024^{325} \bmod 2231 = 0805 \\ m_3 &= 0716^{325} \bmod 2231 = 0913 \end{aligned}$$

Wird die entschlüsselte Ziffernfolge abermals in Zweier-Blöcke geteilt und in Buchstaben codiert, so erhält man wiederum den Klartext „geheim“ zurück.

3.2.4 Sicherheit

Die Sicherheit des RSA-Algorithmus ist von einigen Kriterien abhängig, vor allem aber durch die Wahl der Primzahlen p und q. Denn im wesentlichen basiert die Sicherheit darauf, dass es sehr schwierig ist große Zahlen zu faktorisieren. Der folgende Satz zeigt, dass die Berechnung des geheimen Schlüssels aus dem Wissen des öffentlichen Schlüsselpaars äquivalent zur Faktorisierung natürlicher Zahlen ist.

Satz 3.1. *Sei eine Zahl $n = p \cdot q$ das Produkt von zwei verschiedenen Primzahlen. Dann sind folgende zwei Aussagen äquivalent:*
(i) Aus der Zahl n kann man die Faktoren p und q bestimmen. (das heißt man kann die Zahl n faktorisieren)
(ii) Aus der Zahl n und einer Zahl e, welche teilerfremd zu $\phi(n)$ ist, kann man eine Zahl d berechnen mit $e \cdot d \equiv 1 (\mathrm{mod} \phi(n))$. (das heißt man kann das multiplikative Inverse von e zu $\phi(n)$ bestimmen)

Beweis. (i) $\Rightarrow$ (ii): Wenn (i) gilt, so ist man sofort in der Lage $\phi(n)$ zu berechnen und mittels

des Erweiterten Euklidischen Algorithmus den geheimen Schlüssel d zu berechnen.
(ii) $\Rightarrow$ (i): Gilt (ii), so kann man mit einem Algorithmus bei Kenntnis der Zahlen n und e, sofort die Zahl d mit $e \cdot d \equiv 1(\mathrm{mod}\,\phi(n))$ bestimmen. Die Kongruenz $e \cdot d \equiv 1(\mathrm{mod}\,\phi(n))$ kann umgeschrieben werden zu $e \cdot d - 1 = \ell \cdot \phi(n)$ für eine ganze Zahl ℓ. Angenommen, man findet eine Zahl $x \in \mathbb{Z}/m\mathbb{Z}^{\times}$ und eine Zahl $k \in \mathbb{N}$ mit folgenden Eigenschaften:

$$x^{2k} \equiv 1(\mathrm{mod}n) \Longleftrightarrow n \mid \left(x^{2k} - 1\right) \Longleftrightarrow n \mid \left(x^k - 1\right)\left(x^k + 1\right).$$
$$x^k \not\equiv \pm 1(\mathrm{mod}n)$$

Aus der zweiten Kongruenz folgt somit, dass $n \nmid \left(x^k - 1\right)$ und $n \nmid \left(x^k + 1\right)$, aber n nach der ersten Kongruenz deren Produkt teilt und daher muss in jedem der beiden Faktoren jeweils genau einer der beiden Primfaktoren p und q enthalten sein. Wird nun $gcd((x^k - 1), n)$ und $gcd((x^k + 1), n)$ berechnet, erhält man die beiden Primfaktoren von n. ([1], Satz 10.4, S. 126-127) $\square$

Zusammenfassend lässt sich sagen, dass man aus dem öffentlichen Schlüsselpaar (n, e) den geheimen Schlüssel d nur berechnen kann, wenn man bereits faktorisieren kann. Jedoch muss man bei der Auswahl der Primzahlen p und q einige Aspekte beachten, da sonst diverse Faktorisierungsalgorithmen erfolgreich sein könnten. Wie bereits erwähnt sollten diese eine Größe von mindestens 1024 Bit aufweisen. Darüber hinaus darf der Unterschied zwischen diesen beiden Primfaktoren nicht zu groß sein, da andernfalls das Zahlkörpersieb, eine spezielle Form des Quadratischen Siebes, zu wirkungsvoll wäre. Ist der Größenunterschied der beiden Primzahlen zu gering, so könnte man mit dem Fermat-Algorithmus Erfolg haben. Falls die Zahlen $(p-1)$ oder $(q-1)$ relativ kleine Primteiler besitzen, ist die $(p-1)$-Faktorisierungsmethode von Pollard sehr effizient. ([1], S. 118-127) Um die Sicherheit des RSA-Algorithmus zu demonstrieren, veröffentlichten die Autoren des Verfahrens im Jahre 1977 eine zusammengesetzte Zahl mit 129 Stellen, welche auch als RSA-129 bezeichnet wird. Mit dieser Herausforderung wollten die Autoren zeigen, dass diese Zahl nicht zu faktorisieren wäre. Erst im Jahre 1994 konnte ein Team von Mathematikern unter Verwendung von 1600 verbundenen Computern die Zahl RSA-129 mit dem Quadratischen Sieb in ihre Faktoren zerlegen. 17 Jahre hat es also gedauert, bis man diese Herausforderung lösen konnte, womit gezeigt wurde, dass das RSA-Verfahren eine ausreichende Sicherheit gewährleistet. Mittlerweile existieren jedoch weitaus schnellere Faktorisierungsalgorithmen und wesenlich leistungsstärkere Computer, wodurch es notwendig wurde, dass die Primzahlen p und q ungefähr 300 Stellen (1024 Bit, das heißt $p, q \approx 2^{1024}$) haben. ([10], S. 117)

4 Faktorisierungsalgorithmen

In der heutigen Zeit ist es mittels Primzahltests und leistungsfähiger Computer effizient möglich, bei einer großen Zahl herauszufinden, ob diese eine Primzahl oder eine zusammengesetzte Zahl ist. Bereits Euklid hatte ungefähr 300 Jahre v. Chr. bewiesen, dass jede natürliche Zahl eine eindeutige Primfaktorzerlegung hat (siehe Satz 2.2.). Bis ins 20. Jahrhundert reichte den Mathematikern diese Erkenntnis und man verwendete vergleichweise wenig Zeit dafür, um die konkreten Primfaktoren einer zusammengesetzten Zahl zu bestimmen. Mit dem Aufkommen des im vorigen Abschnitt beschriebenen RSA-Verfahrens änderte sich dies jedoch. Da die RSA-Verschlüsselung nur dann geknackt werden kann, wenn man dazu in der Lage ist effizient zusammengesetzte Zahlen zu faktorisieren, wurde ab diesem Zeitpunkt viel Arbeit in diverse Faktorisierungsalgorithmen investiert. Auch wenn einige, sich in Grenzen haltende Erfolge erzielt werden konnten, gibt es bis dato noch kein Faktorisierungsverfahren, welches effizient genug wäre, um Zahlen in der Größenordnung wie sie heute verwendet werden, zu faktorisieren. ([9], S. 331-332) Die folgenden Kapitel dieser Arbeit widmen sich den klassischen, aber auch aktuellen Faktorisierungsalgorithmen. Dabei werden die dahintersteckenden Ideen aller Algorithmen entsprechend angeführt und anhand von Beispielen erläutert. Zusätzlich dazu werden die Algorithmen mittels des von Otto Forster bereitgestelltem Multipräzisions-Interpreter *ARIBAS* beschrieben, wodurch auch Beispiele mit größeren Zahlen berechnet werden können. Zu guter Letzt werden die verschiedenen Vor- und Nachteile der jeweiligen Verfahren erklärt und auf deren unterschiedlichen Laufzeiten verwiesen.

Bevor mit der Faktorisierung einer Zahl N begonnen wird, gilt es jedoch einige Vorüberlegungen zu berücksichtigen:
1) Zu Beginn werden sämtliche „kleine“ Teiler von N herausgeteilt, da die Laufzeit für künftige Verfahren sonst sehr hoch sein könnte.
2) Als nächstes wird in Form eines Primzahltests ermittelt, ob die Zahl N zusammengesetzt ist oder nicht. Dazu eignen sich beispielsweise der *Solovay-Strassen-Primzahltest* oder der *Miller-Rabin-Test*. Beide Primzahltests sind relativ schnell, jedoch gehören sie zu den probabilistischen Primzahltests. Dies bedeutet, dass sie nur mit einer großen Wahrscheinlichkeit eruieren können, ob es sich um eine Primzahl handelt. Falls eine Zahl bei einem derartigen Test jedoch versagt, so ist sie sicher zusammengesetzt.
3) Falls N eine zusammengesetzte Zahl ist, wird nun versucht einen nichttrivialen Teiler d von N mithilfe eines Faktorisierungsalgorithmus zu finden. Hat man einen Teiler d gefunden, so muss man zunächst prüfen, ob d eine Primzahl ist oder ebenfalls eine zusammengesetzte Zahl. Sollte d zusammengesetzt sein, muss auch diese Zahl faktorisiert werden. Ansonsten beginnt man abermals bei Schritt 2) mit der Zahl $\frac{N}{d}$, bis die Zahl N vollständig faktorisiert wurde. ([15], S. 1, [10], S. 92)

4.1 Probedivision

Die vermutlich einfachste und gleichzeitig langsamste Methode eine Zahl N zu faktorisieren ist die Probedivision. Dabei dividiert man solange durch sämtliche Zahlen $2, 3, \ldots$ bis ein Teiler von N gefunden wurde. Nach Satz 2.2. ist jede Zahl aus Primfaktoren zusammengesetzt, deswegen genügt es lediglich durch Primzahlen zu dividieren, da die Primfaktoren eines echten Teilers von N wesentlich früher getestet werden als der Teiler selbst. ([9], S. 332) Es reicht die Probedivision bis zu $\sqrt{N}$ durchzuführen, wie folgender Satz zeigt:

Satz 4.1. *Ist $N \in \mathbb{N}$ eine zusammengesetzte Zahl, so muss N einen Primteiler p haben, mit $p \leq \sqrt{N}$.*

Beweis. Weil N eine zusammengesetzte Zahl ist, besitzt diese nichttriviale Teiler x und y, sodass $N = x \cdot y$. Dabei muss entweder $x \leq \sqrt{N}$ oder $y \leq \sqrt{N}$ sein, denn sonst wäre $N = x \cdot y > \sqrt{N} \cdot \sqrt{N} = N$, was offensichtlich zu einem Widerspruch führt. Die Zahlen x und y haben nach Satz 2.2. Primteiler, welche auch das Produkt N teilen. ([8], Theorem 8.1.1., S. 123) □

Zur Bestimmung der Primzahlen, durch welche geteilt werden soll, kann beispielsweise das Sieb des Eratosthenes herangezogen werden. Diese Primzahlen werden anschließend in einer Tabelle gespeichert. Falls ein Primfaktor gefunden worden ist, sucht man solange nach weiteren Primfaktoren, bis die Zahl vollständig faktorisiert wurde. ([8], S. 123) In der nachstehenden Box wird der Algorithmus zusammenfassend dargestellt, wobei davon ausgegangen wird, dass es sich bei N um keine Primzahl handelt.

Algorithmus (Probedivision)
Zu faktorisieren sei eine zusammengesetzte Zahl N:

1) Berechne $\frac{N}{p_i}$ für alle Primzahlen von $2 \leq p_i \leq \sqrt{N}$ solange, bis $p_1 \mid N$.
2) Ersetze N durch $\frac{N}{p_1}$ und wiederhole 1) für alle Primzahlen von $p_1 \leq p_i \leq \sqrt{N}$ solange bis erneut ein Faktor p_2 gefunden wurde.
3) Wiederhole 2) solange bis alle Primfaktoren gefunden wurden.

Beispiel. Es soll die Zahl $N = 1173$ mittels Probedivision faktorisiert werden. Da $\lfloor \sqrt{n} \rfloor = 34$, reicht es die Primzahlen $p \leq 34$ zu betrachten: $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$
Der erste Primteiler ist die Zahl 3, womit $\frac{1173}{3} = 391$. Der zweite Primfaktor ist die Zahl 17 und weil das Ergebnis von $\frac{391}{17} = 23$ eine weitere Primzahl ist, wurde die Zahl $1173 = 3 \cdot 17 \cdot 23$ vollständig faktorisiert.

In *ARIBAS* ist eine Liste aller Primzahlen $< 2^{16}$ gespeichert. Die folgende Funktion ist somit in der Lage Zahlen $< 2^{32} = 4294967296$ zu faktorisieren.

Beispiel. Es soll die Zahl $N = 2^{29} + 1 = 536870913$ mithilfe der ARIBAS-Funktion „factors“ faktorisiert werden.

```
==> factors(2**29+1).
3
59
-: 3033169
```

Abbildung 6: Probedivision in ARIBAS

4.1.1 Laufzeit

Prinzipiell ist die Laufzeit um einen beliebigen Faktor von N zu finden $O\left(\sqrt{N}\right)$. Man kann relativ leicht zeigen, dass die Probedivision eine exponentielle Laufzeit hat, in dem man die zu faktorisierende Zahl N in Binärschreibweise angibt, also $N = \sum_{i=0}^{n} b_i 2^i$ mit $b_i \in \{0, 1\}, b_n = 1$. Man erhält daher mit

$$O\left(\sqrt{N}\right) = O\left(\sqrt{2^n}\right) = O\left(2^{\frac{n}{2}}\right)$$

eine exponentielle Laufzeit. Um konkret den Primfaktor p einer Zahl N zu finden ergibt sich eine Laufzeit von $O(p)$, das heißt, man muss solange durch sämtliche Primzahlen dividieren, bis man bei p angelangt ist ([9], S. 333). Eine genauere Abschätzung ist möglich, wenn man ungefähr weiß, wie viele Primzahlen es bis zu einer gewissen Schranke gibt.

Satz 4.2 (**Primzahlsatz**). *Sei* $\pi(x) :=\mid \{p \text{ Primzahl}, p \leq x\} \mid$. *So gilt:*

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln(x)} = 1$$

([9], Satz 27.5, S. 341)

Der vorangehende Satz besagt, dass man die Anzahl der Primzahlen kleiner oder gleich einer festgelegten Schranke x durch $x/\ln(x)$ abschätzen kann ([9], S. 341). Mit dieser Erkenntnis ergibt sich nun eine Laufzeit von $O\left(\frac{\sqrt{N}}{\ln\sqrt{N}}\right)$. Für Zahlen von der Größenordnung wie sie im RSA-Algorithmus verwendet werden ($N \approx 10^{600}$), müsste man daher $\frac{\sqrt{10^{600}}}{\ln\sqrt{10^{600}}} \approx 1,45 \cdot 10^{297}$ Probedivisionen durchführen, um Erfolg zu haben. Dies ist jedoch mit den heutigen Rechenleistungen in angemessener Zeit unmöglich. Der schlechteste Fall, welcher eintreten kann ist aber, wenn N bereits selbst eine Primzahl ist. Dennoch eignet sich die Probedivison sehr gut um verhältnismäßig kleine Primfaktoren $p < 10^6$ zu finden. Zumeist wird bevor ein effizienterer Faktorisierungsalgorithmus angewendet wird, eine unvollständige Probedivison durchgeführt, um eben solche kleinen Primteiler zu finden. ([8], S. 124) Dies begründet sich darin, dass Faktorisierungsalgorithmen für sehr große Zahlen, wie beispielsweise das Zahlkörpersieb, für bis zu sechsstellige Zahlen länger brauchen als die Probedivision ([16], S. 109).

4.2 Faktorisierung nach Idee von Fermat

In diesem Abschnitt werden der Fermat Algorithmus und dessen Weiterentwicklung, der Lehman Algorithmus, angeführt. Darüber hinaus wird auch der Algorithmus des Quadratischen Siebes beschrieben, da dieser aus der Idee des Fermat Algorithmus hervorgeht.

4.2.1 Fermat Algorithmus

Der Fermat Algorithmus gehört zu den klassischen Faktorisierungsalgorithmen und wurde bereits im Jahre 1643 von Pierre de Fermat in einem Brief beschrieben. Fermat kam auf die Idee eine zusammengesetzte Zahl N als Differenz zweier Quadratzahlen darzustellen, womit diese mithilfe der dritten binomischen Formel wie folgt umgeschrieben werden kann:

$$N = x^2 - y^2 = (x + y) \cdot (x - y)$$

Somit hätte man bereits zwei Faktoren von N gefunden. Fermat nutzte diese Erkenntnis und formte die Gleichung um auf $x^2 - N = y^2$. Nun überprüfte er, ob die Differenz $x^2 - N$ für verschiedene Zahlen x eine Quadratzahl ergibt. Ist man dabei erfolgreich, so hat man die Zahl N bereits in kleinere Faktoren zerlegt, wobei diese nicht prim sein müssen. Die Wahrscheinlichkeit eine solche Quadratzahl y zu finden ist umso höher, je näher x^2 bei N liegt. ([9], S. 333) Demnach wird eine Folge x_s definiert:

$$x_0 := \lceil \sqrt{N} \rceil \quad \text{und} \quad x_s := x_0 + s \quad \text{mit} \quad s = 1, 2, 3, ...$$

Nun werden diese Folgenglieder x_s solange erzeugt bis $z_s := {x_s}^2 - N$ eine Quadratzahl ergibt. Zusammenfassend entsteht der folgende Algorithmus, wobei hier N eine ungerade zusammengesetzte Zahl ist.

Algorithmus (Fermat)
Zu faktorisieren sei eine zusammengesetzte ungerade Zahl N:

1) Erstelle eine Folge mit $x_0 := \lceil \sqrt{N} \rceil$ und $x_s := x_0 + s$.
2) Berechne $z_s := {x_s}^2 - N$ solange bis $z_s = y^2$.
3) $(x_s + y)$ und $(x_s - y)$ sind die Faktoren von N. Sind die Faktoren prim ist man fertig und wenn nicht wiederhole 1) und 2) solange bis alle Primfaktoren gefunden wurden.

Als obere Schranke für x kann hier $x = \frac{N+9}{6}$ verwendet werden, womit sich für y der Wert $y = \frac{N-9}{6}$ ergeben würde. Würde der Faktor $(x - y)$ existieren bei jener oberen Grenze, so erhält man folgenden Wert für diesen:

$$x - y = \frac{N + 9}{6} - \frac{N - 9}{6} = \frac{18}{6} = 3$$

Demnach wäre $(x - y)$ minimal und es macht klarerweise wenig Sinn den Fermat Algorithmus bis zu jener Grenze durchzuführen, weil der investierte Aufwand bereits den der Probedivision überschritten hätte. Ob dieser Algorithmus immer funktioniert, soll nun untersucht werden. Angenommen N sei eine ungerade Zahl mit $N = s \cdot t$ und $t \leq s$, wobei s, t nicht unbedingt prim sein müssen. Damit N ungerade ist, müssen auch die beiden Faktoren s, t ungerade sein. Definiert man Zahlen x und y wie folgt

$$x := \frac{s+t}{2} \quad \text{und} \quad y := \frac{s-t}{2}$$

dann sind $s, t \in \mathbb{Z}$ und es gilt:

$$x^2 - y^2 = \left(\frac{s+t}{2}\right)^2 - \left(\frac{s-t}{2}\right)^2 = \frac{s^2 + 2st + t^2 - s^2 + 2st - t^2}{4} = \frac{4st}{4} = s \cdot t = N$$

Unter der Voraussetzung das N ungerade ist, erhält man daher mit dem Fermat-Algorithmus immer eine entsprechende Faktorisierung. ([9], S. 333-335)

Beispiel. Faktorisiert werden soll die Zahl $N = 10147$ mittels Fermat-Algorithmus. Die Folge x_s beginnt bei $x_0 = \lceil \sqrt{10147} \rceil = 101$.

s	x_s	$z_s = x_s^2 - N$
0	101	$54 = 3 \cdot 7 \cdot 53$
1	102	$257 = 257$
2	103	$462 = 2 \cdot 3 \cdot 7 \cdot 11$
3	104	$669 = 3 \cdot 223$
4	105	$878 = 2 \cdot 439$
5	106	$1089 = 33^2$

Für die Spalte der Werte z_s wurde zusätzlich die Primfaktorzerlegung angeführt, damit sofort ersichtlich ist, ob es sich dabei um eine Quadratzahl handelt oder nicht. Nach insgesamt fünf Wiederholungen wurde eine Quadratzahl gefunden, wodurch sich folgende Zerlegung ergibt:

$$N = {x_5}^2 - z_5 = 106^2 - 33^2 = (106 + 33) \cdot (106 - 33) = 139 \cdot 73$$

Bei den gefundenen Faktoren 139 und 73 handelt es sich um Primzahlen, also wurde die Zahl N vollständig faktorisiert.

Mittels einiger simpler Überlegungen kann der Algorithmus deutlich optimiert werden. Für jede Berechnung von z_s muss die Zahl x_s quadriert werden. Diese Quadrierungen kann man sich durch folgenden Trick ersparen:

$$x_{s+1}^2 = (x_s + 1)^2 = x_s^2 + 2x_s + 1$$

Durch Umformung von $z_s = x_s^2 - N$ auf $N = x_s^2 - z_s$ und Einsetzen des Ergebnisses von x_{s+1}

erhält man nun:

$$z_{s+1} = x_{s+1}^2 - N = x_s^2 + 2x_s + 1 - (x_s^2 - z_s) = z_s + 2x_s + 1$$

Somit lässt sich jede Zahl z_s als Addition der bereits berechneten Zahl z_{s-1}, $2x_{s-1}$ und 1 darstellen, was wesentlich einfacher zu berechnen ist, als das Quadrat von x_s.
Um zu entscheiden, ob es sich bei z_s um eine Quadratzahl handelt, muss man die Wurzel berechnen. Da auch diese Berechnungen bei größeren Zahlen sehr aufwendig sind, verwendet man quadratische Reste um zu entscheiden, ob es sich auch wirklich um eine Quadratzahl handelt. Ist nämlich ein Rest $r \equiv z_s \bmod m$ ein quadratischer Nichtrest modulo einer bestimmten Zahl m, so ist z_s auch keine Quadratzahl. Wird $m = 10$ als Modul gewählt, so ergeben sich die quadratischen Reste $\{0, 1, 4, 5, 6, 9\}$. Dies bedeutet, dass man alle Zahlen, welche auf $2, 3, 7$ oder 8 enden im vorhinein ausschließen kann, weil sie keine Quadratzahlen sein können. Mit dem Modul $m = 16$ ergeben sich nur noch $\{0, 1, 4, 9\}$ als quadratische Reste, womit man von insgesamt 16 möglichen Resten bereits 75% nicht weiter beachten muss. Betrachtet man das vorige Beispiel hätte man mit dem Modul $m = 10$ sofort z_1, z_2, z_4 und mit $m = 16$ die Zahlen z_0, z_2, z_3, z_4 ausschließen können. Im Gegensatz zu $m = 10$ muss man bei $m = 16$ zwar die entsprechenden Reste erst berechnen, dennoch liefert dieser Trick eine enorme Erleichterung. ([9], S. 334)

4.2.1.1 Laufzeit Um die ungefähre Laufzeit des Fermat Algorithmus angeben zu können benötigt man folgenden Satz:

Satz 4.3. *Sei $N = s \cdot t$ eine ungerade zusammengesetzte Zahl. Gilt $|s - t| \leq c \cdot \sqrt[4]{N}$ mit $c > 0$, so findet man die Faktoren s und t nach $O(c^2)$ Operationen.*

Beweis. O.B.d.A sei $t \leq s$ und $N = (x + y) \cdot (x - y)$ mit $x = \frac{s+t}{2}$, $y = \frac{s-t}{2}$ die entsprechende Faktorisierung, die nach k Operationen gefunden wurde. Nun werden folgende Ungleichungen benutzt:
1) $x_0 := \lceil \sqrt{N} \rceil \leq \sqrt{N} + 1$ und daher ist $x_0^2 \leq N + 2\sqrt{N} + 1$
2) Mittels einiger Umformungen aus der Voraussetzung $|s - t| \leq c\sqrt[4]{N}$ gilt:

$$\begin{aligned} s - t &\leq c \cdot \sqrt[4]{N} \\ (s-t)^2 &\leq c^2\sqrt{N} \\ \frac{(s-t)^2}{4} &\leq \frac{c^2}{4}\sqrt{N} \\ z_k = y^2 = \left(\frac{s-t}{2}\right)^2 &\leq \frac{c^2}{4}\sqrt{N} \end{aligned}$$

Mittels dieser Ungleichungen und durch Auflösen der expliziten Darstellung $z_k = z_0 + 2kx_0 + k^2$ nach k ergibt sich eine Abschätzung für die Anzahl der benötigten Schritte k. Der Wert

z_k lässt sich leicht herleiten aus der im vorigen Abschnitt berechneten rekursiven Folge $z_{s+1} = z_s + 2x_s + 1$. Auf diese Weise erhalten wir

$$\begin{aligned}
k &= \sqrt{x_0^2 + z_k - z_0} - x_0 \\
&\leqslant \sqrt{N + 2\sqrt{N} + 1 + \frac{c^2}{4}\sqrt{N} - 1} - \sqrt{N} \\
&= \sqrt{N + \left(2 + \frac{c^2}{4}\right)\sqrt{N}} - \sqrt{N} \\
&= \sqrt{N + \left(\frac{8 + c^2}{4}\right)\sqrt{N}} - \sqrt{N} \\
&= \sqrt{N\left(1 + \frac{8 + c^2}{4\sqrt{N}}\right)} - \sqrt{N} \\
&= \sqrt{N}\left(\sqrt{1 + \frac{8 + c^2}{4\sqrt{N}}} - 1\right) \\
&\leqslant \sqrt{N}\left(1 + \frac{8 + c^2}{8\sqrt{N}} - 1\right) \\
&= 1 + \frac{c^2}{8}
\end{aligned}$$

Bei großen Zahlen sind die Zahl 1 und der Faktor $\frac{1}{8}$ von c^2 der letzten Zeile vernachlässigbar, wodurch die Aussage bewiesen wäre. ([9], Satz 27.1, S. 335) □

Ist die Differenz der Faktoren s und t beispielsweise genau $\sqrt[4]{N}$, wäre $c = 1$, womit man nach obiger Abschätzung die Faktoren unmittelbar gefunden hätte. Je größer die Differenz der beiden Faktoren ist, desto größer wird auch die Laufzeit des Verfahrens. Demnach eignet sich der Fermat Algorithmus nur für Zahlen, bei welchen die Faktoren nahe beieinander liegen. Ab einer gewissen Größe der Differenz zwischen den Faktoren ist die Probedivision schneller. Wie bereits erwähnt sind im schlechtesten Fall $\frac{N+9}{6}$ Schritte notwendig, um den kleinstmöglichen Faktor, das heißt den Faktor 3, zu finden. ([9], S. 335)

4.2.2 Lehman Algorithmus

Im Jahre 1974 entwickelte Russell Sherman Lehman eine Methode zur Faktorisierung, welche eine Kombination aus Probedivision und Fermat-Algorithmus darstellt ([9], S. 335). Es soll hier jedoch nur eine kurze Erklärung des Verfahrens stattfinden. Grundsätzlich basiert diese Methode auf folgendem, ebenfalls von Lehman formuliertem Satz:

Satz 4.4. *Sei $N = p \cdot q$ eine ungerade Zahl, wobei mit p und q prim sind. Weiters gelte $1 \leq r < \sqrt{N}$ mit $\sqrt{\frac{N}{r+1}} \leq p \leq \sqrt{N}$. Dann existieren natürliche Zahlen x, y und k mit den*

folgenden Eigenschaften:
1. $x^2 - y^2 = 4kN$
2. $x \equiv 1(\mathrm{mod}2)$*, wenn* k *gerade und* $x \equiv k + N(\mathrm{mod}4)$*, wenn* k *ungerade*
3. $\sqrt{4kN} \leq x \leq \sqrt{4kN} + \frac{1}{4(r+1)}\sqrt{\frac{N}{k}}$
Falls N *prim ist, existieren solche Zahlen nicht. ([9], Satz 27.2, S. 336)*

Für den Beweis dieses Satzes siehe ([11], Satz 5.1.3., S. 227-228).
Um die Laufzeit so gering wie möglich zu halten und damit alle Voraussetzungen des vorangehenden Satzes erfüllt sind, wird zuerst $r = \sqrt[3]{N}$ bestimmt. Um eine ungerade zusammengesetzte Zahl mittels Lehman Algorithmus zu faktorisieren werden zu Beginn Probedivisionen bis zur Schranke $r = \sqrt[3]{N}$ durchgeführt. Wenn kein Teiler gefunden wird, werden alle in Satz 4.4. genannten Voraussetzungen eingehalten. Nun wird für alle möglichen Zahlenpaare (k, x) mit $1 \leq k \leq r$, welche die dritte Eigenschaft von Satz 4.4. erfüllen, überprüft ob die Differenz $x^2 - 4kN$ eine Quadratzahl y^2 ist. Falls dies zutreffen sollte, erhält man durch Berechnung von $gcd(x-y, N)$ und $gcd(x+y, N)$ die zu ermittelnden Primfaktoren p und q. ([9], S. 335-336)

Algorithmus (Lehman)
Zu faktorisieren sei eine zusammengesetzte ungerade Zahl N:

1) Führe Probedivisionen durch für $2 \leq p \leq \sqrt[3]{N}$. Falls $p \mid N$ hat man einen Faktor gefunden. Ansonsten gehe zu 2).
2) Suche alle Paare (k, x) mit $\sqrt{4kN} \leq x \leq \sqrt{4kN} + \frac{1}{4(r+1)}\sqrt{\frac{N}{k}}$.
3) Überprüfe für alle Paare, ob $z := x^2 - 4kN$ eine Quadratzahl ist, also $z = y^2$.
4) Berechne $gcd(x - y, N)$ und $gcd(x + y, N)$ um die Faktoren von N zu erhalten. Sind die Faktoren prim ist man fertig und wenn nicht wiederhole 1) bis 4) solange bis alle Primfaktoren gefunden wurden.

Beispiel. Faktorisiert werden soll die Zahl $N = 957$ mittels Lehman Algorithmus. Zu Beginn wird eine Probedivision mit $2 \leq p \leq \lfloor \sqrt[3]{N} \rfloor = 9$ durchgeführt. Diese bringt keinen Teiler von N. Für $k = 1$ ist nur $x = 62$ möglich und daher ist $z = x^2 - 4kN = 62^2 - 4 \cdot 1 \cdot 957 = 3844 - 3828 = 16$. Da $z = 16 = 4^2 = y^2$ eine Quadratzahl ist, bekommt man folgende Faktoren durch Berechnung von $gcd(x - y, N) = gcd(58, 1073) = 29$ und $gcd(x + y, N) = gcd(66, 1073) = 33$. Die vollständige Faktorisierung ergibt also $N = 957 = 29 \cdot 33$.

4.2.2.1 Laufzeit

Die unvollständige Probedivision zu Beginn des Verfahrens benötigt $O\left(\sqrt[3]{N}\right)$ Operationen. Für die kommenden Schritte ergibt sich folgende Laufzeit:

$$O\left(\sum_{k=1}^{\sqrt[3]{N}} \frac{\sqrt[6]{N}}{4\sqrt{k}}\right) = O\left(\int_{k=1}^{\sqrt[3]{N}} \frac{\sqrt[3]{N}}{4\sqrt{k}} dk\right) = O\left(\sqrt[3]{N}\right)$$

Sowohl die Laufzeit für die unvollständige Probedivision, als auch für die kommenden Schritte ergibt $O\left(\sqrt[3]{N}\right)$, womit man damit bereits die Gesamtlaufzeit hätte. Wird N wieder als Binärzahl angegeben, wie es bereits bei der Probedivision gezeigt wurde, ergibt sich dadurch die ebenfalls exponentielle Laufzeit von $O\left(2^{\frac{n}{3}}\right)$. Verglichen mit der Probedivision ist der Lehman Algorithmus asymptotisch schneller. ([9], S. 336-337)

4.2.3 Quadratisches Sieb

Aufbauend auf der Idee von Fermat eine Zahl als Differenz zweier Quadratzahlen anzuschreiben um deren Faktoren zu bestimmen, entwickelte sich ein wesentlich effizienterer Algorithmus, nämlich das von Carl Pomerance im Jahre 1981 entwickelte Quadratische Sieb. Anstelle von $N = x^2 - y^2 = (x + y) \cdot (x - y)$ verwendet man hier folgende Kongruenz:

$$x^2 \equiv y^2 \bmod N \Longleftrightarrow (x + y) \cdot (x - y) \equiv 0 \bmod N \Rightarrow N \mid (x + y) \cdot (x - y)$$

Gilt darüber hinaus $x - y \not\equiv 0 \bmod N$ und $x + y \not\equiv 0 \bmod N$, also $x \not\equiv \pm y \bmod N$, dann erhält man durch Berechnung von $gcd(x + y, N)$ und $gcd(x - y, N)$ zwei nichttriviale Faktoren von N. Hier setzt nun die Idee des Quadratischen Siebes an, in dem verschiedene betragskleine quadratische Reste kombiniert werden, anstatt die Quadratzahlen der oben angeführten Kongruenzen direkt zu ermitteln. Bereits im Jahre 1931 benutzte die Kettenbruchmethode ebenfalls diese Kombination von quadratischen Resten zur Faktorisierung. Um diese Idee zu erläutern wird hier ein kleines Beispiel angeführt:

Beispiel. Sei $N = 14129$ mit $\lceil \sqrt{N} \rceil = 119$ die zu faktorisierende Zahl. Durch Kombination der quadratischen Reste

$$119^2 \equiv 2^5 (\mathrm{mod} 14129)$$
$$121^2 \equiv 2^9 (\mathrm{mod} 14129)$$

erhält man bereits nach 3 Berechnungen ein gewünschtes Resultat. Multipliziert man dabei neben den linken Seiten auch die rechten Seiten miteinander, so ist leicht ersichtlich, dass sich auf beiden Seiten Quadrate ergeben:

$$(119 \cdot 121)^2 \equiv \left(2^7\right)^2 (\mathrm{mod} 14129)$$
$$\Rightarrow (14399)^2 \equiv 270^2 \equiv 128^2 (\mathrm{mod} 14129)$$

Durch Berechnung von $gcd(270 + 128, 14129) = 199$ und $gcd(270 - 128, 14129) = 71$ erhält man die gewünschte Faktorisierung $N = 199 \cdot 71$, da die gefundenen Faktoren beide prim sind. Mit dem Fermat Algorithmus müsste man stattdessen bis $x = 135$ rechnen, damit $z = y^2$ eine Quadratzahl ergibt ($135^2 - 14129 = 64^2$).

Bereits dieses Beispiel mit sehr kleinen Zahlen zeigt die Vorteile gegenüber dem Fermat Algorithmus. Im Wesentlichen lässt sich nun der Algorithmus des Quadratischen Siebes auf

zwei Schritte aufteilen. Während im vorangehenden *Siebschritt* hinlänglich viele quadratische Kongruenzen gebildet werden, erfolgt im *Auswahlschritt* eine Kombination dieser Kongruenzen derart, sodass $x^2 \equiv y^2 \bmod N$. Gilt darüber hinaus $x \not\equiv \pm y \bmod N$, so hat man zwei Faktoren von N gefunden. Die Wahrscheinlichkeit, dass eine derartige Kongruenz nichttrivial ist (d.h. $N \nmid (x-y)$ und $N \nmid (x+y)$), beträgt $1 - \left(\frac{1}{2}\right)^k$, wobei man unter k die Anzahl der Primteiler von N versteht. Dies begründet sich darin, dass ein beliebiger Primfaktor p von N die selbe Wahrscheinlichkeit hat ein Teiler von $x-y$ oder von $x+y$ zu sein.

Siebschritt

Im Siebschritt wird versucht möglichst betragskleine quadratische Reste zu erzeugen, die wiederum nur kleine Primfaktoren aufweisen sollten. Je kleiner die Primfaktoren, desto höher ist die Wahrscheinlichkeit, dass diese in vielen Kongruenzen auftreten, wodurch sie sich eher kombinieren lassen als mit großen Primfaktoren. Um zu entscheiden, bis zu welcher Größe diese Primteiler bei den kommenden Berechnungen berücksichtigt werden, benötigt man folgende zwei Definitionen. ([9], S. 352-354)

Definition (**glatt, potenzglatt**)**.** Seien N und B natürliche Zahlen. Man sagt, N sei $B-$glatt, wenn alle Primfaktoren von N kleiner oder gleich B sind. Gilt dies auch für alle Primzahlpotenzen, so nennt man N $B-$potenzglatt. ([9], Definition 27.6, S. 341)

Beispiel. Die Zahl $637 = 7^2 \cdot 13$ ist $13-$glatt, $49-$potenzglatt, jedoch nicht $13-$potenzglatt.

Definition (**Faktorbasis**)**.** Sei B eine natürliche Zahl. Man nennt eine Faktorbasis zu B eine Menge $\mathcal{F} = \{-1, p_2, \ldots, p_k\}$, wobei für die Primzahlen $p_2, \ldots, p_k \leq B$ gilt. ([9], Definition 27.11, S. 354)

Beispiel. Sei $B = 5$, dann ist die dazugehörige Faktorbasis die Menge $\mathcal{F} = \{-1, 2, 3, 5\}$. Somit sind nur jene Zahlen erlaubt, die sich vollständig mit der Faktorbasis zerlegen lassen, also nur aus Primzahlen der Faktorbasis bestehen. Die Zahlen $30 = 2 \cdot 3 \cdot 5$ und $200 = 2^3 \cdot 5^2$ lassen sich mit der Faktorbasis zerlegen, während dies bei $35 = 5 \cdot 7$ nicht funktioniert, da $7 \notin \mathcal{F}$.

Aus der Definition der Faktorbasis ist sofort ersichtlich, dass mit $p_1 := -1$ auch negative Reste möglich oder sogar erwünscht sind. Mittels des folgenden Siebpolynoms $Q(i)$ werden nun die entsprechenden quadratischen Reste erzeugt.

$$Q(i) = \left(i + \lfloor \sqrt{N} \rfloor\right)^2 - N$$

Dieses Siebpolynom erinnert sehr an die im Fermat Algorithmus verwendete Folge $z_s := x_s^2 - N$ mit $x_0 := \lceil \sqrt{N} \rceil$. Durch Verwendung des Legendre-Symbols können bereits im vorhinein bestimmte Primzahlen p_i der Faktorbasis vernachlässigt werden, da diese keine Teiler der mittels $Q(i)$ erzeugten quadratischen Reste r_i sein können. Ist eine Primzahl p Teiler von $Q(i)$

so muss gelten:

$$\begin{aligned} p \mid Q(i) &\Leftrightarrow p \mid \left(\left(i + \lfloor\sqrt{N}\rfloor\right)^2 - N\right) \\ &\Leftrightarrow \left(i + \lfloor\sqrt{N}\rfloor\right)^2 \equiv N \bmod p \\ &\Leftrightarrow \left(\frac{N}{p}\right) = 1 \end{aligned}$$

Das bedeutet, dass N ein quadratischer Rest modulo p sein muss, um ein Teiler von $Q(i)$ zu sein. Daher wird zu Beginn für alle Primzahlen der Faktorbasis das Legendre-Symbol berechnet und im nächsten Schritt werden nur jene p berücksichtigt, bei welchen N ein quadratischer Rest modulo p ist. Da es modulo einer ungeraden Primzahl p genauso viele quadratische Reste wie Nichtreste gibt, kann man dadurch bereits ungefähr die Hälfte aller Primzahlen der Basis vernachlässigen. Zur Berechnung des Legendre-Symbols können die Ergänzungssätze und das Quadratische Reziprozitätsgesetz, welche in den Grundlagen angeführt wurden, verwendet werden. Um festzulegen, welche $i \in \mathbb{Z}$ für die Berechnung der $Q(i)$ herangezogen werden, wird ein Siebintervall folgendermaßen definiert:

$$I_\varepsilon := \left\{i \in \mathbb{Z} : -N^\varepsilon \leq i \leq N^\varepsilon\right\}, \text{mit} \quad 0 \leq \varepsilon \leq \tfrac{1}{2}$$

Nach Festlegung des Siebintervalls werden nun fortwährend $Q(i)$ berechnet und man überprüft gleichzeitig ob diese über der gewählten Faktorbasis zerfallen. Über der Faktorbasis zerfallen bedeutet, dass die Werte von $Q(i)$ lediglich aus Primfaktoren zusammengesetzt sind, welche in der Faktorbasis enthalten sind. Dabei würde der Siebvorgang viel zu lange dauern, wenn hier die Probedivision verwendet werden würde, das heißt man dividiert nach der Reihe durch sämtliche Primzahlen der Faktorbasis. Hier macht man sich folgende Äquivalenz zu Nutze, um die Effzienz des Siebens zu erhöhen, wobei $r \in \mathbb{N}$ und $k \in \mathbb{Z}$:

$$p^r \mid Q(i) \Leftrightarrow p^r \mid Q\left(i + k \cdot p^r\right)$$

Man kann sich relativ leicht von der Richtigkeit dieser Äquivalenz überzeugen, in dem man dafür $x_i := i + \lfloor\sqrt{N}\rfloor$ setzt und wie folgt vorgeht:

$$(x_i + k \cdot p^r)^2 - N \bmod p^r \equiv x_i^2 + 2x_i k p^r + k^2 p^{2r} - N \bmod p^r \equiv x_i^2 - N$$

Dies bedeutet, dass die Werte $Q(i)$ stets denselben Rest modulo p^r haben wie alle in p^r Schritten folgenden Zahlen $Q(i + k \cdot p^r)$. Darüber hinaus gilt daher die oben beschriebene Äquivalenz. Anschließend versucht man sämtliche Kongruenzen

$$Q(i) \equiv 0 \bmod p_s^r$$

für alle Primzahlen p_s der Faktorbasis und deren möglichen Exponenten r zu berechnen. In

einer Siebtabelle werden diese Resultate nun entsprechend dargestellt. Die Zeilen werden dabei nach dem Index $i \in I_\varepsilon$ und die Spalten nach den Primzahlen p_s strukturiert. Dabei wurde darauf geachtet, dass nur jene Primzahlen der Faktorbasis mit $\left(\frac{N}{p_s}\right) = 1$ angeführt werden. Die Tabelleneinträge stehen für den Exponenten r_{ij} der jeweiligen Primzahl, das heißt wie oft diese Primzahl in $Q(i)$ enthalten ist. Die Exponenten r_{ij} sind daher definiert durch $Q(i) \equiv 0(\mathrm{mod} {p_j}^{r_{ij}})$. Die folgende Siebtabelle bezieht sich auf das bereits angeführte Beispiel $N = 14129$, wobei als Siebintervall $I = [-5, 5]$ festgelegt wurde.

i	x_i	$Q(i)$	-1	2	5	11	17	29	39	41	
-5	113	-1360	1	4	1	-	1	-	-	-	1
-4	114	-1133	1	-	-	1	-	-	-	-	103
-3	115	-904	1	3	-	-	-	-	-	-	113
-2	116	-673	1	-	-	-	-	-	-	-	673
-1	117	-440	1	3	1	1	-	-	-	-	1
0	118	-205	1	-	1	-	-	-	-	1	1
1	119	32	-	5	-	-	-	-	-	-	1
2	120	271	-	-	-	-	-	-	-	-	271
3	121	512	-	9	-	-	-	-	-	-	1
4	122	755	-	-	1	-	-	-	-	-	151
5	123	1000	-	3	3	-	-	-	-	-	1

Üblicherweise werden in der Siebtabelle nur jene $Q(i)$ angeführt, welche sich vollständig mit der Faktorbasis zerlegen lassen. Dies ist genau dann der Fall wenn bei fortwährender Division von $Q(i)$ durch alle Primzahlen p_s und deren Potenzen p_s^r nur noch der Trivialfaktor 1 bleibt. Der Vollständigkeit halber wurden in der vorliegenden Siebtabelle alle $Q(i)$ angeführt, auch jene, welche nicht über der Faktorbasis zerfallen und somit noch größere Faktoren übrigbleiben. Beispielsweise bleibt für $Q(-4) = -1133$ die Zahl 103 übrig, wobei es sich hierbei um eine Primzahl handelt. Wie bereits erwähnt gilt, dass wenn ein $Q(i)$ durch p teilbar ist, dann sind auch alle darauffolgenden $Q(i + k \cdot p)$ mit $k \in \mathbb{Z}$ teilbar durch p. Im Beispiel ist $Q(-5)$ durch 2 teilbar und daher auch $Q(-3), Q(-1), Q(1), Q(3)$ und $Q(5)$. Daneben ist $Q(-5)$ und $Q(-1)$ teilbar durch 5 und deswegen sind es gleichermaßen $Q(0)$ und $Q(5)$ sowie $Q(4)$.

Auswahlschritt

Nach der Erstellung der Siebtabelle folgt der Auswahlschritt, bei welchem versucht wird, sämtliche gefundene Kongruenzen

$$x_i^2 \equiv Q(i)(\mathrm{mod} N)$$

zu einer entsprechenden Lösung zu kombinieren. Hat man k quadratische Reste $Q(i_1), \ldots, Q(i_k)$

gefunden, so versucht man nun eine derartige Auswahl $\{i_{l_1}, \ldots, i_{l_n}\} \subseteq I_\varepsilon$ zu finden

$$\left(x_{i_{l_1}} \cdot \ldots \cdot x_{i_{l_n}}\right)^2 \equiv Q\left(i_{l_1}\right) \cdot \ldots \cdot Q\left(i_{l_n}\right) (\mathrm{mod} N)$$

bei welcher nicht nur auf der linken, sondern auch auf der rechten Seite der Kongruenz ein Quadrat ist. Um dies zu erreichen dürfen die Exponenten der jeweiligen Primfaktoren p_s im Produkt der rechten Seite nur gerade sein. Andernfalls lässt sich das gesamte Produkt der jeweiligen $Q(i)$ nicht als Quadrat schreiben. Da r_{ij} die ausgewählten Exponenten der Tabelle sind, muss die folgende Gleichung erfüllt sein, um zu gewährleisten, dass alle Exponenten der rechten Seite gerade sind, d.h. $z_1, \ldots, z_k$ werden so gewählt, dass gilt

$$z_1 r_{1j} + z_2 r_{2j} + \ldots + z_k r_{kj} \equiv 0 (\mathrm{mod} 2)$$

Die Werte $z_1, \ldots, z_k \in \{0, 1\}$ ergeben zusammen den Lösungsverktor z. Dieser Lösungsvektor gibt an, welche $Q(i)$ kombiniert werden müssen, damit für alle j die Exponenten in obiger Gleichung gerade sind. Die vorher genannte Kongruenz muss für jedes der m Elemente der gewählten Faktorbasis stimmen. Insgesamt ergibt sich also folgendes Gleichungssystem $Rz \equiv 0 \pmod 2$

$$\begin{pmatrix} r_{11} & \cdots & r_{k1} \\ \vdots & \ddots & \vdots \\ r_{1m} & \cdots & r_{km} \end{pmatrix} \cdot \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} \equiv 0 (\mathrm{mod} 2)$$

Mit 0 wird hier der Nullvektor mit m Einträgen bezeichnet. Um den Lösungsvektor z zu erhalten ist ein lineares Gleichungssystem über dem Restklassenkörper $\mathbb{F}_2$ zu lösen. Sollte $k > m$ sein, so muss eine Lösung existieren. Wenn als Lösung die triviale Kongruenz herauskommen sollte, muss nach weiteren Kongruenzen gesucht werden. Eine weitere Überlegung zeigt, dass bei Betrachtung der Spaltenvektoren der Matrix R, welche die Exponenten der $Q(i)$ modulo 2 angeben, höchstens m davon linear unabhängig sein können. Im schlechtesten Fall braucht man $m + 1$ Spaltenvektoren, damit diese Menge linear abhängig ist und man durch Kombination spezieller Spaltenvektoren den Nullvektor erhält. Betrachtet man abermals das Beispiel $N = 14129$ und die bereits erstellte dazugehörige Siebtabelle, so ergibt sich das folgende Gleichungssystem.

$$R = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \text{ und } \quad z = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Hier ist anzumerken, dass sämtliche $Q(i)$ der Siebtabelle, welche nicht über der Faktorbasis zerfallen sind, weggelassen wurden. Aus dem mittels Gaußschem Eliminationsverfahren erhaltenem Lösungsvektor z ist ersichtlich, dass die Kongruenzen 4 und 5, also $Q(1)$ und $Q(3)$ kombiniert werden müssen um auf beiden Seiten Quadratzahlen zu haben. Man erhält:

$$\begin{aligned}(x_1 \cdot x_3)^2 &\equiv Q_1 \cdot Q_3 && (\mathrm{mod} 14129)\\ (119 \cdot 121)^2 &\equiv 2^5 \cdot 2^9 && (\mathrm{mod} 14129)\\ 270^2 &\equiv \left(2^7\right)^2 && (\mathrm{mod} 14129)\end{aligned}$$

Mittels der letzten Kongruenz konnte man wie im vorigen Beispiel die Faktoren 199 und 71 von $N = 14129$ finden.

Algorithmus (Quadratisches Sieb)
Zu faktorisieren sei eine zusammengesetzte ungerade Zahl N:

1) Wähle eine Faktorbasis $\mathcal{F} = \{-1, p_2, \ldots, p_k\}$ und überprüfe für alle Primzahlen p von $\mathcal{F}$, ob $\left(\frac{N}{p}\right) = 1$. Nur diese Primzahlen werden in der Siebtabelle verwendet.
2) Erzeuge quadratische Reste $Q(i) = (i + \lfloor\sqrt{N}\rfloor)^2 - N$ und sei r_{ij} der Exponent von p_j in $Q(i)$ für alle p_j aus $\mathcal{F}$. Gilt $Q(i) \equiv 0 \left(\mathrm{mod} p_j^{r_{ij}}\right)$, dann auch $Q(i + k \cdot p_j^{r_{ij}}) \equiv 0 \left(\mathrm{mod} p_j^{r_{ij}}\right)$. Ist keine vollständige Zerlegung bezüglich $\mathcal{F}$ möglich, gehe zu $Q(i+1)$. Halte die Resultate in einer Siebtabelle fest.
3) Erstelle anhand der Einträge (Exponenten der Primfaktoren) der Siebtabelle eine Matrix R.
4) Löse das Gleichungssystem $Rz \equiv 0 \bmod 2$ mit dem Gaußschem Eliminationsverfahren.
5) Die Einträge mit 1 des Lösungsvektors z geben die entsprechenden $Q(i)$ an, welche wie folgt kombiniert werden müssen, wobei $x_i := (i + \lfloor\sqrt{N}\rfloor)$.
   Wir erhalten: $\prod Q(i) \equiv (\prod x_i)^2 \bmod N$, wobei die Produkte über alle i mit $z_i = 1$ laufen.
6) Sei $x := \prod Q(i) \bmod N$ und $y := \prod x_i \bmod N$. Gilt $x \not\equiv \pm y \bmod N$, so sind $gcd(x + y, N)$ und $gcd(x - y, N)$ zwei nichttriviale Faktoren von N. Ansonsten gehe zu 2) und berechne weitere Kongruenzen. Sind die Faktoren prim ist man fertig und wenn nicht wiederhole 2) bis 6) solange bis alle Primfaktoren gefunden wurden.

4.2.3.1 Optimierungen

Sämtliche Berechnungen im Zuge des oben erwähnten Siebschritts können parallel auf verschiedenen Rechnern durchgeführt werden, in dem man unterschiedliche $Q(i)$ berechnet und diese mit unterschiedlichen Primzahlen der Faktorbasis zu zerlegen versucht. Darüber hinaus vernachlässigt man das Sieben durch große Primzahlpotenzen p^r für große p, da es sich vorwiegend für kleine p rentiert. Natürlich können dadurch einige Kongruenzen nicht über

der Faktorbasis zerfallen, dennoch bringt dieser Schritt eine Verbesserung der Laufzeit. Um sehr große Zahlen zu faktorisieren bedient man sich noch einer weiteren Idee um das Sieben effizienter zu gestalten. Anstelle vielfacher Divisionen der $Q(i)$ durch sämtliche Elemente der Faktorbasis p und deren Potenzen p^r benutzt man hier deren Logarithmen. Durch Verwendung der Logarithmen von p^r und $Q(i)$ werden anstatt von Divisionen die wesentlich schneller durchführbaren Subtraktionen erledigt. Wenn sich nun ein Element $Q(i)$ vollständig mit der gewählten Faktorbasis zerlegen lässt, so ergibt die Differenz von $Q(i)$ und der jeweiligen p^r einen Wert, welcher optimalerweise der 0 sehr nahe ist. Die entsprechenden Logarithmen sollten vorab mit größtmöglicher Genauigkeit abgespeichert werden, um etwaige Rundungsfehler zu vermeiden. ([9], S. 352-358)

Eine wesentliche Verbesserung des Verfahrens stellt das erstmals von Robert D. Silverman entwickelte **M**ulti**p**olynomiale **Q**uadratische **S**ieb (MPQS) dar ([10], S. 166). Im Gegensatz zum allgemeinen Verfahren, welches nur das Siebpolynom $Q(x_i) = x_i^2 - N$ benutzt, werden hier mehrere verschiedene Siebpolynome verwendet. Dies sollte dazu führen, dass sich wesentlich mehr $Q(i)$ erzeugen lassen, welche mit der Faktorbasis vollständig zerlegbar sind. Das MPQS verwendet Polynome mit folgenden Eigenschaften:

$$Q(x) = (ax + b)^2 - N$$

Die Zahlen a und b werden derart gewählt, sodass $a \mid b^2 - N$. Definiert man $c := \frac{b^2-N}{a}$, so gilt:

$$Q(x) = (ax + b)^2 - N = a^2x^2 + 2abx + b^2 - N = a \cdot \Big(ax^2 + 2bx + c\Big)$$

Das letzte Resultat zeigt, dass $a \mid Q(x)$ und daher werden nur die ganzen Zahlen $\frac{Q(x)}{a}$ in den weiteren Berechnungen berücksichtigt. Durch Auswahl von verschiedenen a erhält man somit gleich eine Reihe verschiedener Siebpolynome, welche zur Erzeugung adäquater Kongruenzen herangezogen werden können. Da $a \mid b^2 - N$ muss man zunächst die Kongruenz $b^2 \equiv N \bmod a$ lösen, um den entsprechenden Wert für b zu erhalten. Um diese Berechnung zu erleichtern, sollte a einerseits nicht zu groß sein und andererseits nur möglichst wenige Primfaktoren besitzen. ([9], S. 358)

Eine weitere Erhöhung der Effizienz liefert die *Big Prime Variation*. Bei der Zerlegung der $Q(i)$ mittels der Faktorbasis kommt es häufig vor, dass lediglich ein großer Primfaktor, ein sogenannter *Big Prime*, übrig bleibt. Betrachtet man die oben angeführte Siebtabelle, so sieht man relativ schnell, dass bei allen $Q(i)$, welche nicht vollständig zerlegt werden konnten, der übrig gebliebene Faktor eine Primzahl ist. Wenn bei der Faktorzerlegung von $Q(i) \equiv x_i^2 \bmod N$

nur noch ein Primfaktor fehlt kann man $Q(i)$ wie folgt anschreiben:

$$Q(i) = v \cdot p$$

Mit der Zahl v wird das Produkt diverser Elemente der Faktorbasis bezeichnet und $p > p_{last}$ sei eine Primzahl, welche größer als p_{last}, die letzte und damit größte Primzahl der Faktorbasis, ist. Nun legt man eine zusätzliche Schranke für p fest, wie beispielsweise $p < p_{last}^2$ und speichert für diese p sämtliche quadratische Reste

$$Q(i) \equiv x_i^2 \bmod N \quad \text{mit} \quad Q(i) = v \cdot p$$

Unter der Zahl t wird die Anzahl sämtlicher Primzahlen zwischen p_{last} und der zuvor festgelegten Schranke p_{last}^2 verstanden. Durch Anwendung des Geburtstagsparadoxon (siehe Satz 4.5.) ist nach $O(\sqrt{t})$ quadratischen Resten $Q(i)$ zu erwarten, dass zwei davon dieselbe Big Prime haben. Man spricht hier von Big Prime Kollisionen:

$$Q(i) \equiv x_i^2 \bmod N, \quad Q(i) = v_i \cdot p, \quad i = 1, 2$$

Mit diesen Erkenntnissen ist es möglich einen neuen quadratischen Rest zu kombinieren, welcher vollständig über der Faktorbasis zerfällt. Sei dazu u das inverse Element von p modulo N, also $p \cdot u \equiv 1 \bmod N$, dann ergibt sich durch Verwendung der Big Prime Kollisionen

$$\begin{aligned} v_1 \cdot v_2 = \frac{Q(1)}{p} \cdot \frac{Q(2)}{p} &\equiv \left(\frac{x_1 \cdot x_2}{p}\right)^2 \bmod N \\ &\equiv (x_1 \cdot x_2 \cdot u)^2 \bmod N \end{aligned}$$

Hiermit wurde gezeigt, dass das Produkt $v_1 \cdot v_2$ vollständig über der Faktorbasis zerfällt und daher ein neuer quadratischer Rest generiert werden konnte. Im folgenden Beispiel wird eine Faktorisierung mit dem MPQS in *ARIBAS* dargestellt.

Beispiel. Faktorisiert werden soll die Mersenne Zahl $M_{137} = 2^{137} - 1$ mittels der ARIBAS-Funktion „QSfactorize“. Bei genauerem Hinblick auf die Abbildung sind einige wichtige Informationen sofort ablesbar. Einerseits ist die Anzahl der Elemente der verwendeten Faktorbasis 585. Andererseits hat das hier verwendete MPQS 1577 verschiedene Polynome benötigt, um nach insgesamt 567 über der Faktorbasis zerfallenen quadratischen Resten $Q(i)$ eine nicht triviale Faktorisierung zu finden. Die angeführten Punkte in der Abbildung stehen für vollständig faktorisierte quadratische Reste. Da die Faktorisierung etwas länger gedauert hat, wurden hier nur der Anfang und das Ende der entsprechendenen Ausgabe angeführt. Der fehlende Teil wird durch die senkrecht verlaufenden Punkte symbolisiert. Mit dem Rufzeichen zum Schluss der Ausgabe wird angegeben, dass nur eine Relation benötigt wurde, um die gewünschte Faktorisierung zu erhalten. Beim ersten gefundenen Faktor handelt es sich um

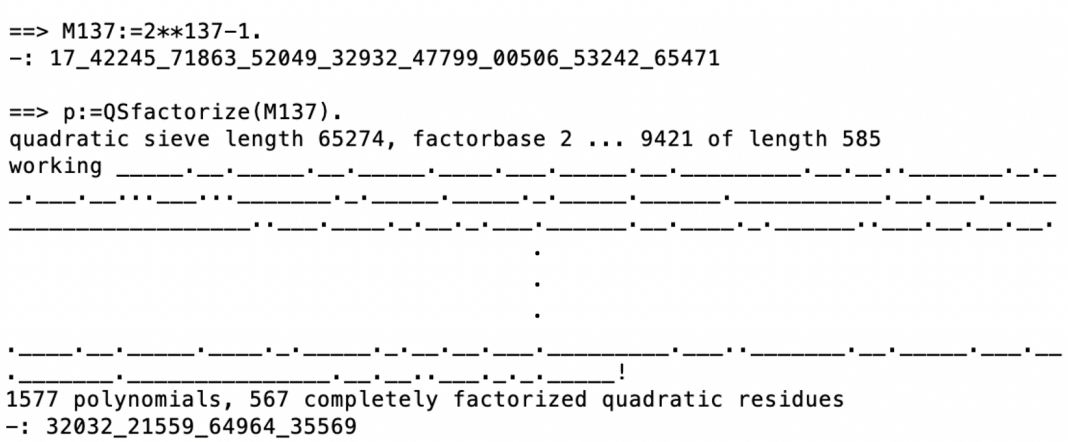

Abbildung 7: Quadratisches Sieb in ARIBAS

eine Primzahl. Durch Division von M_{137} durch diesen Primfaktor erhält man den zweiten Primfaktor, wie sich leicht mittels Primzahltests überprüfen lässt. ARIBAS besitzt noch

```
==> q:=M137 div p.
-: 54_39042_18360_02042_90159
```

Abbildung 8: Divison in ARIBAS

die Fuktion „qs_factorize“, welche die *Big Prime Variation* verwendet. Dadurch konnte die Faktorisierung wesentlich beschleunigt werden, wie Abbildung 9 zeigt. Neben den bereits

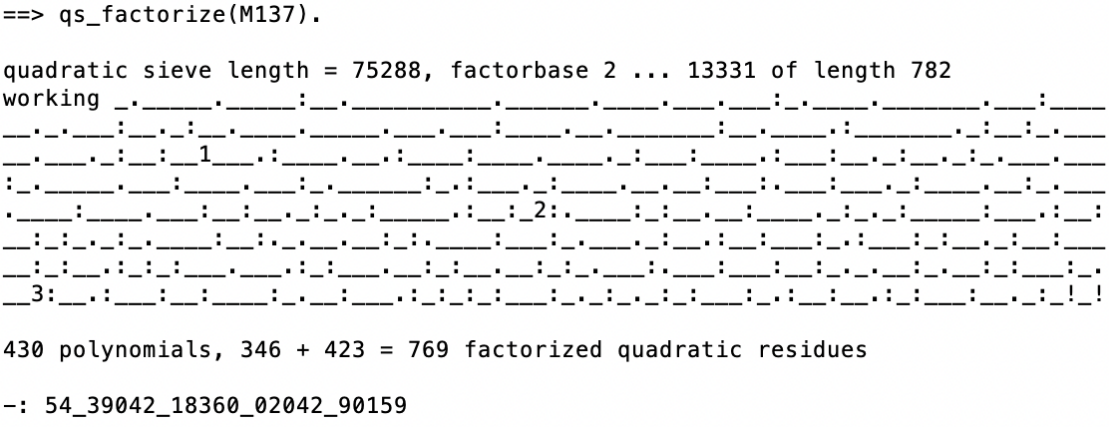

Abbildung 9: Quadratisches Sieb mit Big Prime Variation in ARIBAS

zuvor erklärten Punkten sind hier auch Doppelpunkte zu sehen. Diese stehen für quadratische Reste, welche durch kombinieren von zwei Resten mit der gleichen Big Prime entstanden sind. Die Addition $346 + 423 = 769$ bedeutet, dass von den insgesamt 769 vollständig zerlegten quadratischen Resten 423 mittels dieser Kombinationen erzeugt worden sind. Der restlichen

346 quadratischen Reste wurden wie üblich mit der Faktorbasis zerlegt. In Abbildung 8 konnte sogar der vollständige Ausschnitt angeführt werden, da die Faktorisierung bedeutend schneller war, im Vergleich zur Funktion ohne *Big Prime Variation*. ([10], S. 179-181)

4.2.3.2 Laufzeit

Im Wesentlichen ist die Laufzeit vom Siebschritt abhängig. Das Gaußsche Elliminitaionsverfahren, welches im Auswahlschritt verwendet wird, lässt sich in der Regel für diese dünn besetzten Matrizen vergleichsweise rasch lösen. Im Siebschritt hingegen spielt die gewählte Faktorbasis eine entscheidende Rolle. Wird diese zu groß gewählt, so findet man zwar sehr viele zerlegbare quadratische Reste $Q(i)$, jedoch steigt hierbei der Rechenaufwand enorm an. Falls die Faktorbasis zu klein gewählt werden sollte, kann es womöglich sehr lange dauern, bis genügend quadratische Reste gefunden werden, die sich mit dieser zerlegen lassen. ([9], S. 358-360) Folgendes Resultat von Canfield, Erdös und Pomerance hilft bei der Größenwahl der Faktorbasis, da es die Wahrscheinlichkeit für eine vollständige Zerlegung mit einer zuvor ausgwählten Faktorbasis abschätzt. $\psi(X, B)$ bezeichnet dabei die Anzahl aller B-glatten Zahlen $n \in \mathbb{N}$ mit $n \leqslant X$. Sei $u > 1$ eine reelle Zahl. Dann ist $\psi\left(X, X^{1/u}\right)$ für die nachfolgenden Berechnungen von großer Wichtigkeit. Es gilt nun:

$$\frac{\psi\left(X, X^{1/u}\right)}{X} = u^{-(1+o(1))u}$$

für $X \to \infty, u \to \infty$. Als Nebenbedingung sollte $X^{1/u} > (\log X)^{1+\varepsilon}$ für jedes feste $\varepsilon > 0$ gelten. Unter $o(1)$ versteht man eine gegen 0 gehende asymptotische Funktion. Die rechte Seite dieser Gleichung kann grob abgeschätzt werden auf $u^{-u}\left(= \frac{1}{u^u}\right)$. Dies bedeutet, dass mit einer Wahrscheinlichkeit von $\frac{1}{u^u}$ eine Zahl $x \in \mathbb{N}$ lediglich Primfaktoren $\leqslant x^{1/u}$ besitzt. Betrachtet man Zahlen von der Größe 10^{12}, so kann man durch obige Abschätzung davon ausgehen, dass rund jede 27−ste Zahl nur Primfaktoren kleiner als 10^4 hat. Dies begründet sich darin, weil $u^u = 3^3 = 27$. Wesentlich seltener sind derartige Zahlen mit Primfaktoren, welche kleiner als 10^3 sind, da aufgrund von $u^u = 4^4 = 256$ nur ungefähr jede 256-ste Zahl in Frage kommen würde. Alles in allem gilt es diese Erkenntnisse bei der Erstellung der Faktorbasis entsprechend zu berücksichtigen und eine möglichst effiziente Wahl zu treffen. ([10], S. 165) Insgesamt ergibt sich für das Quadratische Sieb folgende Laufzeit:

$$O(N) = \mathrm{e}^{\sqrt{(1+o(1))\ln(N)\ln(\ln(N))}}$$

Dies ist eine subexponentielle Laufzeit, womit sie zwischen einer polynomiellen und exponentiellen Laufzeit liegt. Je nach Eigenschaften der zu faktorisierenden Zahl hat das Quadratische Sieb in der Regel eine bessere Laufzeit als alle bisher angeführten Verfahren. Darüber hinaus ist sie vergleichbar mit der Laufzeit der Elliptischen Kurven Methode, welche in Abschnitt 4.3.3 beschrieben wird. Die Laufzeit beider Faktorisierungsverfahren ist grundsätzlich von

glatten Zahlen abhängig, wodurch die ähnlichen Laufzeiten begründet wären. Ein weiterer Vorteil des Quadratischen Siebes ist, dass die Laufzeit ausschließlich von der Größe der zu faktorisierenden Zahl abhängt. Die Primfaktoren selbst spielen dabei nur sekundär eine Rolle. Im Alltag werden das Quadratische Sieb und die Elliptische Kurven Methode oftmals kombiniert. Kleinere Faktoren bis $\sqrt[3]{N}$ werden zumeist von der Elliptischen Kurven Methode schneller gefunden, während das Quadratische Sieb effizienter arbeitet, falls genau zwei Faktoren, welche größer als $\sqrt[3]{N}$ sind, übrig bleiben. Generell gilt das Quadratische Sieb, auch wenn es sehr rechenintensiv ist, für Zahlen bis zur Größenordnung 10^{10} als eines der schnellsten Verfahren falls die Zahl, die zerlegt werden soll keine spezifischen Merkmale hat, bei welchen andere Faktorisierungsverfahren äußerst effizient arbeiten.

Es existiert ein Faktorisierungsalgorithmus, der für Zahlen ab einer Größe von ungefähr 120 Dezimalstellen asymptotisch schneller ist als das Quadratische Sieb. Dieses, in den 80er-Jahren entwickelte Verfahren nennt man Zahlkörpersieb, da es algebraische Zahlkörper verwendet. Im Gegensatz zum allgemeineren Quadratischen Sieb lassen sich mittels Zahlkörpersieb glatte Relationen schneller und auch zahlreicher finden. Dabei gibt es zwei unterschiedliche Arten, nämlich das spezielle und das allgemeine Zahlkörpersieb. Verglichen mit dem allgemeinen eignet sich das spezielle Zahlkörpersieb bespielsweise für Mersenne- und Fermat-Zahlen besonders gut. Bislang konnten mit diesem äußerst komplizierten Faktorisierungsverfahren große Erfolge erzielt werden. So konnte etwa die 9-te Fermat-Zahl $F_9 = 2^{512} + 1$ und die Mersenne-Zahl $M_{1039} = 2^{1039} - 1$ mit dem speziellen Zahlkörpersieb in ihre Primfaktoren zerlegt werden. Darüber hinaus konnte durch Verwendung des allgemeinen Zahlkörpersiebes eine Zahl mit 768 Bit ($\approx 10^{230}$) der RSA-Challenge faktorisiert werden. Trotz allem besitzt auch das Zahlkörpersieb noch immer eine subexponentielle Laufzeit. ([9], S. 358-360, [10], S. 181-182)

4.3 Faktorisierung nach Ideen von Pollard

In diesem Abschnitt werden Faktorisierungsalgorithmen beschrieben, welche auf verschiedenen Ideen des britischen Mathematikers John M. Pollard beruhen. Anfangs werden die Pollard'sche Rho-Methode und die $(p-1)$-Faktorisierungsmethode - beides von ihm erfundene und entwickelte Verfahren - behandelt. Zu guter Letzt wird die Elliptische Kurven Methode näher betrachtet, welche im Wesentlichen auf der $(p-1)$-Faktorisierungsmethode basiert. ([16], S. 112)

4.3.1 Pollard'sche Rho-Methode

Im Jahre 1975 stellte John M. Pollard ein nicht deterministisches Faktorisierungsverfahren vor. Im Vergleich zur Probedivision liefert die Pollard'sche Rho-Methode eine bessere Laufzeit, jedoch unter Verlust der Erfolgsgarantie. Da dieses Verfahren nur zu einer bestimmten Wahrscheinlichkeit erfolgreich ist, gehört es zu den Monte-Carlo-Methoden. ([9], S. 337) Um

die dahintersteckende Idee dieses Verfahrens besser zu verstehen, wird diese anhand von mehreren kleinen Beispielen erläutert. Dazu sei N eine ungerade zusammengesetzte Zahl und p ein dazugehöriger Primfaktor von N. Weiters sei a_k eine Zahlenfolge modulo N, womit $a_k \in \{0, 1, \ldots, N-1\}$ ist. Gilt nun $a_i \equiv a_j \bmod p$ für zwei Indizes mit $i < j$ und darüber hinaus $a_i \not\equiv a_j \bmod N$, so hat man mit $gcd(a_i - a_j, N)$ einen nichttrivialen Faktor von N gefunden.

Beispiel. Sei $N = 35263 = 179 \cdot 197$ und a_k modulo N eine mittels Microsoft Excel erstellte „Zufallsfolge".

k	1	2	3	4	5	6	7	8	9	10
a_k	23301	16016	34012	25809	6568	14847	2740	6720	30405	25693
$a_k(\text{mod}179)$	31	85	2	33	124	169	55	97	154	96
$a_k(\text{mod}197)$	55	59	128	2	**67**	72	179	22	**67**	83

k	11	12	13	14	15	16	17	18	19	20
a_k	10492	21587	33303	13200	11465	3004	11510	21012	30521	6222
$a_k(\text{mod}179)$	110	107	**9**	133	**9**	140	54	69	91	136
$a_k(\text{mod}197)$	51	114	10	1	39	49	84	130	183	115

Aus der ersten Tabelle folgt

$$a_5 \equiv a_9(\text{mod}197) \equiv 67, \quad gcd(a_9 - a_5, N) = gcd(30405 - 6568, 35263) = 197$$

und nach der zweiten Tabelle gilt

$$a_{13} \equiv a_{15}(\text{mod}179) \equiv 9, \quad gcd(a_{13} - a_{15}, N) = gcd(33303 - 11465, 35263) = 179$$

womit man N vollständig faktorisiert hätte.

Klarerweise wusste man bei diesem Beispiel bereits aus welchen Faktoren N zusammengesetzt war. Nun gilt es die beschriebene Idee in einen brauchbaren Algorithmus zu verwandeln. Dazu benötigt man folgenden Satz, auf welchem der Pollard'sche Rho Algorithmus aufbaut.

Satz 4.5. *Es sei M eine Menge mit m paarweise verschiedenen Elementen und $a_1, a_2 \ldots, a_\ell \in M$ eine Zufallsfolge eben dieser Elemente. Die Wahrscheinlichkeit dafür, dass sämtliche Glieder der Zufallsfolge paarweise voneinander verschieden sind, beträgt*

$$p_{m,\ell} = \left(1 - \frac{1}{m}\right) \cdot \left(1 - \frac{2}{m}\right) \ldots \left(1 - \frac{\ell - 1}{m}\right) = \prod_{i=1}^{\ell-1} \left(1 - \frac{i}{m}\right)$$

Beweis. Für jedes Element a_i gibt es m verschiedene Möglichkeiten und daher gibt es für die Zufallsfolge $(a_1, a_2 \ldots, a_\ell)$ insgesamt m^ℓ Möglichkeiten. Da nach obigem Satz alle Glieder

verschieden sein sollen, hat das erste Element a_1 noch m günstige Möglichkeiten, für a_2 bleiben nur noch $m-1$ Möglichkeiten usw. Somit existieren $m \cdot (m-1) \cdots (m-(\ell-1))$ günstige Möglichkeiten für eine Zufallsfolge bestehend aus unterschiedlichen Elementen. Durch Division erhält man

$$p_{m,\ell} = \frac{m \cdot (m-1) \cdots (m-(\ell-1))}{m^\ell} = \frac{m}{m} \cdot \frac{m-1}{m} \cdots \frac{m-(\ell-1)}{m} = 1 \cdot \left(1 - \frac{1}{m}\right) \cdots \left(1 - \frac{\ell-1}{m}\right)$$
$$= \prod_{i=1}^{\ell-1} \left(1 - \frac{i}{m}\right)$$

□

In der Stochastik würde man hier vom „Ziehen mit Zurücklegen“ sprechen. Doch Satz 4.5. ist auch als *Geburtstagsparadoxon* bekannt. Betrachtet man nämlich einen Raum mit 23 Personen, so ist die Wahrscheinlichkeit, dass zwei Personen am selben Tag Geburtstag haben 50%. Bei 60 Personen liegt die Wahrscheinlichkeit schon bei 99%. Man kann dies leicht überprüfen, in dem man in die obige Formel für $m = 365$ die Anzahl der Tage im Jahr und mit ℓ die Anzahl der Personen angibt. Um ein Paradoxon handelt es sich deswegen, weil die meisten Personen bei dieser Fragestellung dazu neigen die Wahrscheinlichkeit weitaus geringer einzuschätzen. Bezogen auf die Pollard'sche Rho Methode bedeutet dies, dass die Wahrscheinlichkeit, dass zwei zufällige Zahlen $a_i \equiv a_j \bmod N$ bei weitem höher ist als man normalerweise vermuten würde. Um diese Wahrscheinlichkeit $p_{m,\ell}$ ungefähr abschätzen zu können, geht man folgendermaßen vor:

$$\ln p_{m,\ell} = \ln \prod_{i=1}^{\ell-1} \left(1 - \frac{i}{m}\right) = \sum_{i=1}^{\ell-1} \ln \left(1 - \frac{i}{m}\right)$$

Mit Bezug auf Faktorisierungsalgorithmen ist die Zahl m verhältnismäßig groß im Vergleich zu ℓ, weswegen man die Abschätzung $\ln(1-x) \approx -x$ für sehr kleine x-Werte verwenden kann. Bezogen auf diese Berechnung bedeutet dies, dass wenn m sehr groß im Vergleich zu ℓ ist, dann ist $\ln(1 - \frac{i}{m}) \approx -\frac{i}{m}$. Daher gilt:

$$\ln p_{m,\ell} \approx -\sum_{i=1}^{\ell-1} \frac{i}{m} = -\frac{\ell(\ell-1)}{2m}$$

Im letzten Schritt wurde die Gaußsche Summenformel verwendet. Letztendlich erhält man eine leicht berechenbare Abschätzung für $p_{m,\ell}$.

$$p_{m,\ell} \approx e^{-\frac{\ell(\ell-1)}{2m}}$$

Ist m groß und gilt $\ell = c\sqrt{m}$ mit $c \approx 3,1$ dann ergibt sich $p_{m,\ell} = 0,0082$, wie man durch Nachrechnen leicht überprüfen kann. Das heißt, die Wahrscheinlichkeit, dass aus einer ℓ-elementigen

Folge der Menge M zwei Elemente gleich sind, beträgt mehr als 99%. Je nach Wahl von c ergeben sich unterschiedliche Abschätzungen der Wahrscheinlichkeit $p_{m,\ell}$. Beispielsweise beträgt die Wahrscheinlichkeit $p_{m,\ell} = 0,5$, also 50%, wenn $c \approx 1,2\sqrt{m}$.

Mit Blick auf die Pollard'sche Rho-Methode ergeben sich daher folgende Erkenntnisse für eine zusammengesetzte Zahl N mit Primfaktor p. Ist a_k mit $1 \leq k \leq \ell$ eine zufällige Folge von Zahlen modulo N, so ist es wesentlich wahrscheinlicher zwei gleiche Folgenglieder $a_i \equiv a_j \bmod p$ zu finden, da $\mid \mathbb{Z}/p\mathbb{Z} \mid < \mid \mathbb{Z}/N\mathbb{Z} \mid$. Mittels der zuvor hergeleiteten Abschätzungen für die Wahrscheinlichkeit $p_{m,\ell}$ findet man, wenn $\ell \geq 3,1\sqrt{p}$, zu 99% Indizes i, j mit $a_i \equiv a_j \bmod p$. Gilt zusätzlich $a_i \not\equiv a_j \bmod N$ hat man durch Berechnung von $gcd(a_i - a_j, N)$ einen nichttrivialen Teiler von N gefunden. Der Algorithmus kann nun in zwei Schritte aufgeteilt werden. Zuerst muss eine Zufallsfolge erstellt werden und im Anschluss werden Folgenglieder derart ausgewählt, sodass man die gerade erwähnten Kongruenzen erhält. ([15], S. 1-3)

Erstellung der Zufallsfolge

Pollard nutzte zur Erstellung dieser zufälligen Elemente eine rekursive Pseudo-Zufallsfolge. Die Elemente werden durch das Anfangselement a_0 und die Vorschrift $a_{k+1} = f(a_k)$ festgelegt, wobei $f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}$ eine Funktion ist, auf welche in den kommenden Abschnitten noch detaillierter eingegangen wird. Folgendes Beispiel soll eine derartige Pseudo-Zufallsfolge zeigen. ([10], S. 100)

Beispiel. Mit der Funktion $f(a) = a^2 + 2 (\text{mod} 69)$, $a_0 = 2$ und der Vorschrift $a_{k+1} = f(a_k)$ soll eine Zufallsfolge erstellt werden.

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
2	6	38	66	11	54	**20**	57	8	66	11	54	**20**	57	8	66	11

Man sieht sofort, dass die Folge ab a_6 periodisch ist.

Dies gilt für jede derartig konstruierte Folge, wie der nächste Satz beweisen wird.

Satz 4.6. *Für eine Abbildung $f : M \to M$ einer endlichen Menge M in sich und einen beliebigen Anfangswert $a_0 \in M$ wird eine Folge (a_k) rekursiv definiert mit $a_{k+1} := f(a_k)$. Daher existieren Zahlen $s, t \in \mathbb{Z}$ mit $t > 0$ und $s \geqslant 0$, sodass*

$$a_{k+t} = a_k \quad \text{für alle } k \geqslant s.$$

(Nimmt man dabei s und t minimal, so bezeichnet man s als Vorperiode und t als Periode oder Periodenlänge der Folge (a_k).)

Beweis. Aufgrund der Endlichkeit der Menge M ist es nicht möglich, dass alle a_k unterschiedlich sind. Daher existieren ein $s \geqslant 0$ und ein $t > 0$ damit $a_{s+t} = a_s$ gilt. Aufgrund dessen gilt wiederum

$$a_{s+1} = f(a_s) = f(a_{s+t}) = a_{(s+1)+t}$$

Wird weiterhin die Funktion f wiederholt angewendet, erhält man $a_{k+t} = a_k$ für alle $k \geqslant s$, womit der Satz bewiesen wäre. ([10], Satz 13.2., S. 100-101) □

Versucht man nun eine derartige Pseudozufallsfolge mit Vorperiode s und Periode t grafisch darzustellen, erhält man folgendes Bild:

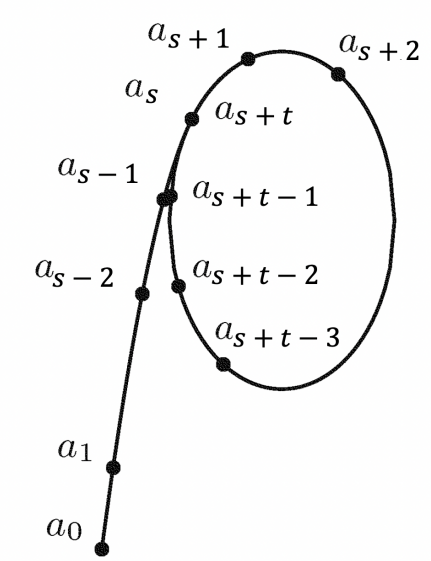

Abbildung 10: (vgl. Quelle: [9], Abb. 27.1, S. 338) Bahn der Pseudozufallsfolge

Die Bahn der Folge a_k ähnelt dem griechischen Buchstaben ρ, weswegen der Algorithmus auch den Namen Rho-Methode bekommen hat. Nun wird ein genauerer Blick auf die Funktion $f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}$ geworfen. Diese sollte möglichst leicht berechenbar sein. Es hat sich gezeigt, dass Funktionen der Gestalt

$$a_{k+1} = f(a_k) = a_k^2 + c(\mathrm{mod} N), \quad c \notin \{-2, 0\}$$

gute Resultate liefern, weshalb diese auch in der Praxis häufig verwendet werden. ([9], S. 338)

Auswahl der Folgenglieder

Nun stellt sich die Frage, wie man mit der erstellten Folge (a_k) Paare findet, für welche $a_i \equiv a_j \bmod p$ gilt. Würde man alle möglichen Paarungen (a_i, a_j) mit $0 \leq i, j \leq \ell$ testen, so benötigt man insgesamt $\binom{\ell+1}{2} = \frac{\ell(\ell+1)}{2}$ Vergleiche, was einen enormen Aufwand bedeutet, falls ℓ sehr groß wird. Darüber hinaus benötigt man erhebliche Speicherkapazitäten, da man für derartige Vergleiche sämtliche Folgenglieder speichern müsste. Der Mathematiker Robert W. Floyd hatte eine Idee um dieser Problematik entgegenzuwirken. ([15], S. 5)

Satz 4.7. *Es sei $f : M \to M$ eine Abbildung, wobei die Menge M endlich ist. Die Folge (a_k)*

sei rekursiv definiert durch $a_{k+1} := f(a_k)$ *mit* $a_0 \in M$*. Die Folge* (a_k) *besitzt eine Vorperiode* s *und eine Periode* t*. Führt man Vergleiche mit* a_k *und* a_{2k} *der Reihe nach durch für* $k = 1, 2, \ldots$*, dann trifft man auf ein minimales* $k_0 > 0$ *mit* $a_{k_0} = a_{2k_0}$*. Die Zahl* k_0 *ist dabei das kleinste positive ganzzahlige Vielfache der Periode* t *mit* $k_0 \geq s$*. Insbesondere ist* $k_0 = t$*, falls* $s \leq t$*.*

Beweis. Sei die Periode $t > 0$ minimal gewählt. Dann gilt für jedes $t' > 0$ mit $a_{k+t'} = a_k$, dass t' ein ganzzahliges Vielfaches von t sein muss. Andererseits wäre $t' = qt + t''$ für q, t'', t' mit $0 < t'' < t'$ und deswegen würde $a_{i+t''} = a_i$ für ein i von ausreichender Größe folgen. Dies würde der vorausgesetzten Minimalität von t widersprechen. Aus diesem Grund resultiert aus $a_{k_0} = a_{2k_0} = a_{k_0+k_0}$, dass k_0 ein ganzzahliges Vielfaches der Periodenlänge t sein muss. Ist nun $k_0 = qt$ mit $q \in \mathbb{Z}$ und sei $v \in \mathbb{Z}$ derart groß, sodass $k_0 + vqt \geq s$, wobei s die Vorperiode bezeichnet, so gilt:

$$a_{k_0+t} = a_{k_0+vqt+t} = a_{k_0+vqt} = a_{k_0}$$

Aus der Gleichheit von $a_{k_0+t} = a_{k_0}$ und Satz 4.6. folgt, dass $k_0 \geq s$ für die minimal gewählte Vorperiode s gelten muss. Um die Umkehrung zu zeigen, wählt man $k_0 := qt \geq s$ mit $q \in \mathbb{Z}$, wodurch $a_{k_0} = a_{2k_0}$ gelten muss. ([10], Satz 13.2., S. 100-101) □

Um nun das gesuchte Paar $a_i \equiv a_j \bmod p$ zu finden, vergleicht man also schrittweise die Paare (a_k, a_{2k}) bis man erfolgreich ist. Problematisch wird es aber, wenn die Periode sehr groß ist, da man sämtliche Folgenglieder $(a_1, a_2, \ldots, a_k, \ldots, a_{2k})$ für derartige Vergleiche speichern müsste. Dies ist jedoch ab einer gewissen Periodenlänge nicht mehr möglich. Um dieses Speicherproblem zu lösen benötigt man nach Robert W. Floyd zwei Folgen, welche parallel berechnet werden. Man definiert dabei für $a_0 = b_0$ die Folge $a_{k+1} = f(a_k)$ und $b_{k+1} = f(f(b_k))$. Aufgrund dessen wäre dann $a_{2k} = b_k$, womit man die gewünschten Vergleiche hätte. Nun muss nur noch das Paar (a_k, b_k) gespeichert werden, aus welchem im Anschluss das nächste Paar berechnet werden kann. Folgender Satz zeigt die Gültigkeit dieser Annahme. ([10], S. 101)

Satz 4.8. *Es sei* $f : M \to M$ *eine Abbildung, wobei die Menge* M *endlich ist. Die Folge* (a_k) *sei rekursiv definiert durch* $a_{k+1} := f(a_k)$ *mit* $a_0 \in M$*. Sei* (b_k) *eine weitere Folge definiert durch* $b_0 = a_0$ *und* $b_{k+1} = f(f(b_k))$*. Dann ist* $b_k = a_{2k}$ *und insbesondere ist* $a_{k_0} = a_{2k_0}$ *gleichwertig mit* $a_{k_0} = b_{k_0}$*.*

Beweis. Mit Induktion soll gezeigt werden, dass $b_k = a_{2k}$. Nach Definition der Folge (b_k) gilt $b_0 = a_0$, womit der Fall für $k = 0$ gezeigt wäre. Zusätzlich ist unter Annahme der Induktionsvoraussetzung

$$b_{k+1} = f(f(b_k)) = f(f(a_{2k})) = f(a_{2k+1}) = a_{2k+2} = a_{2(k+1)}$$

womit der Satz vollständig bewiesen wäre. ([15], Lemma 4, S. 5) □

Verwendet man diese Erkenntnisse beim Algorithmus kommt es einerseits zu einer Einsparung von Speicherplatz, andererseits jedoch zu einer Erhöhung der Rechenzeit von ungefähr

50%. Mittlerweile gibt es jedoch schon Optimierungen des Floyd'schen Zyklenalgorithmus, welche nur zu minimalen Erhöhungen der Rechenzeit führen. ([10], S. 101)

Beispiel. Man betrachte abermals die Folge (a_k) mit der Funktion $f(a) = a^2 + 2(\text{mod}69)$, $a_0 = 2$ und der Vorschrift $a_{k+1} = f(a_k)$ und die eben erwähnte Folge (b_k) mit $b_0 = a_0$ und $b_{k+1} = f(f(b_k))$.

k	0	1	2	3	4	5	**6**	7	8	9	10	11	**12**	13	14	15
a_k	2	6	38	66	11	54	**20**	57	8	66	11	54	**20**	57	8	66
b_k	2	38	11	20	8	11	**20**	8	11	20	8	11	**20**	8	11	20

Für die Folge (a_k) ergibt sich eine Vorperiode von $s = 6$ und eine Periode $t = 6$. Weiters erhält man $k_0 = 6$ für $a_{k_0} = b_{k_0} = a_{2k_0}$. Nach unseren Vorüberlegungen gilt sogar, dass $k_0 = t$, da $s = 6 \leq t = 6$ gilt.

Berücksichtigt man die Abschätzung der in Satz 4.5. angeführten Wahrscheinlichkeit, so ist für eine derartige Zufallsfolge (a_k) einer m-elementigen Menge mit $0 \leq k \leq \lceil (3,1\sqrt{m}) \rceil$ die Wahrscheinlichkeit, dass ein k_0 existiert mit $1 \leq k_0 \leq \lceil (3,1\sqrt{m}) \rceil$ und $a_{k_0} = a_{2k_0}$, höher als 99%. ([15], S. 6)
Zusammenfassend erhält man also folgenden Ablauf für die Pollard'sche Rho-Methode:

Algorithmus (Pollard-Rho)
Zu faktorisieren sei eine ungerade zusammengesetzte Zahl N:

1) Wähle eine natürliche Zahl a_0 mit $0 \leq a_0 < N$, eine ganze Zahl $c \notin \{0, 2\}$ und setze $b_0 := a_0$.
2) Berechne mit $f(a) := a^2 + c \bmod N$ die Folgen $a_{k+1} := f(a_k)$ und $b_{k+1} := f(f(b_k))$
3) Berechne nun für $k = 1, 2, \ldots$ die Zahl $d = gcd(b_k - a_k, N)$ solange bis $1 < d < N$ gilt. Dann ist d ein nichttrivialer Teiler von N. Wiederhole das Verfahren solange bis alle Faktoren gefunden wurden.

Wie bereits erwähnt ist die Pollard'sche Rho-Methode probabilistisch und kann daher aus zwei Gründen fehlschlagen. Entweder ist für sämtliche Paare a_k, b_k ihr größter gemeinsamer Teiler $d = gcd(b_k - a_k, N) = 1$. Dies ist genau dann der Fall, wenn die Primfaktoren p der zu faktorisierenden Zahl N zu groß sind und daher noch kein Zyklus modulo p erzeugt werden konnte. Sollte der größte gemeinsame Teiler $d = N$ ergeben, so haben sich für sämtliche Primfaktoren von N simultan Zyklen gebildet, was bedeutet, es ist modulo sämtlicher Primfaktoren von N zur gleichen Zeit eine Periode entstanden. Dies ist genau dann der Fall, wenn alle Primfaktoren p von N dieselbe k-Folge besitzen. Mit der k-Folge einer Zahl p werden hier jene Indizes k verstanden, bei welchen $a_k \equiv a_{2k} \bmod p$ ist. Sollten die Primfaktoren von N nahe beieinander liegen, so tritt dieser Fall häufiger ein als sonst. Als Abhilfe könnte man die Zahl a_0 neu wählen. In der Praxis wird üblicherweise nicht für jede Differenz $(b_k - a_k)$

der größte gemeinsame Teiler berechnet, sondern es werden mehrere Differenzen als Produkt zusammengefasst:

$$gcd\left(\prod_{k=k_0}^{k_0+n}(b_k - a_k), N\right)$$

Für dieses Produkt aus n Differenzen wird nun der größte gemeinsame Teiler berechnet. Natürlich erhöht sich damit das Risiko einen zweiten Primfaktor in diesem Produkt zu erhalten, oder schlimmstenfalls sogar alle Primfaktoren, wodurch der größte gemeinsame Teiler gleich N wäre. ([10], S. 101-102, [15], S. 8) Im Folgenden wird ein vollständiges Beispiel zum genannten Faktorisierungsverfahren durchgeführt, wobei die Zahlen aus dem Einführungsbeispiel verwendet werden.

Beispiel. Faktorisiert werden soll die zusammengesetzte Zahl $N = 35263$ mit der Pollard'schen Rho-Methode. Als Funktion wurde $f(a) := a^2 + 2$ gewählt und für die Folgen (a_k) und (b_k) gilt $a_0 = b_0 = 2$ mit $a_{k+1} := f(a_k)$ und $b_{k+1} := f(f(b_k)$.

k	a_k	$b_k = a_{2k}$	$a_{2k} - a_k (\mathrm{mod} N)$	$gcd(a_{2k} - a_k, N)$
1	6	38	32	1
2	38	10401	10363	1
3	1446	22939	21493	1
4	10401	5060	29922	1
5	29182	9035	15116	1
6	22939	12904	25228	1
7	3237	11076	7839	1
8	5060	4572	34775	1
9	2664	14012	11348	1
10	9035	18634	9599	1
11	32645	20000	22618	1
12	12904	32842	19938	1
13	1332	18274	16942	197
14	11076	19668	8592	179

Nach 13 Wiederholungen erhält man den ersten Faktor 197 und bereits nach einem weiteren Schritt den zweiten Faktor 179, und da beide Faktoren prim sind, ist die Zahl $N = 35263 = 179 \cdot 197$ vollständig faktorisiert.

Auch in *ARIBAS* gibt es eine Funktion, welche die Pollard'sche Rho-Methode ausführt. Dabei muss man neben der zu faktorisierenden Zahl N, als Schranke die Zahl der Iterationsschritte festlegen. Wie bereits erwähnt sind $gcd-$Berechnungen sehr aufwändig, weswegen meist mehrere Differenzen in einem Produkt zusammengefasst werden. In *ARIBAS* werden jeweils 256 Differenzen zusammengefasst.

Beispiel. Faktorisiert werden soll die Zahl N mittels der ARIBAS-Funktion „poll_rho“. N ist dabei das Produkt aller Primzahlen kleiner als 50 addiert mit 1. Nachdem die Zahl N

```
==> P:=(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47).
-: (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47)

==>  N:=product(P)+1
.
-: 614_88978_25884_91411

==> P:=(2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47).
-: (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47)

==>  N:=product(P)+1.
-: 614_88978_25884_91411

==> p:=poll_rho(N,50000).
working .
factor found after
256 iterations
-: 953

==> q:=N div p.
-: 64521_48820_44587

==> q1:=poll_rho(q,50000).
working ..
factor found after
512 iterations
-: 46727

==> q2:=q div q1.
-: 1_38081_81181
```

Abbildung 11: Pollard’sche Rho-Methode in ARIBAS

entsprechend berechnet wurde, konnte mit der Faktorisierung begonnen werden. Bereits nach 256 Iterationen konnte der erste Primfaktor p gefunden werden. Der Komplementärfaktor q wird klarerweise mittels Division von N durch p berechnet. Da dieser nicht prim ist, wendet man erneut den Algorithmus an. Nach weiteren 512 Iterationen bekommt man den Primfaktor q_1 und durch erneute Divsion den Primfaktor q_2. Somit wurde $N = p \cdot q_1 \cdot q_2$ vollständig faktorisiert.

Es bleibt noch zu erwähnen, dass man bei mehrmaligem Faktorisieren mit dieser Funktion stets ein anderes Resultat erhält, da *ARIBAS* im Zuge der Funktion „poll_rho“ Zufallselemente verwendet. Daher kann es bei wiederholter Faktorisierung der gleichen Zahl zu einer unterschiedlichen Anzahl an Iterationsschritten kommen, oder es werden die Faktoren in einer anderen Reihenfolge gefunden. Es kann auch vorkommen, dass die Funktion manchmal erfolglos ist und manchmal nicht. Zusätzlich wurde mehrmals versucht die Mersenne-Zahl M_{137}, die bereits im Rahmen des Quadratischen Siebs erfolgreich faktorisiert werden konnte,

mit der Pollard'schen Rho-Methode in ihre Faktoren zu zerlegen. Diese Versuche blieben erfolglos, da die Zahl selbst einerseits sehr groß ist und andererseits aus lediglich zwei großen Faktoren besteht. ([10], S. 102-104)

4.3.1.1 Laufzeit

Nach den wahrscheinlichkeitstheoretischen Überlegungen zu Beginn dieses Abschnitts erhält man eine heuristische Laufzeit von $O\left(\sqrt{p}\right)$ um einen Primfaktor einer zusammengesetzten Zahl N zu finden. Natürlich hängt die Anzahl der benötigten Schritte auch von der Wahl der Zahlen a_0 und c in der verwendeten Funktion ab. Der schlechteste Fall wäre, wenn $N = p \cdot q$ ein Produkt aus zwei fast gleich großen Primfaktoren ist. Dann wären $p, q \approx \sqrt{N}$. Als Laufzeit würde sich nun $O\left(N^{\frac{1}{4}}\right)$ ergeben. Schreibt man N abermals als Binärzahl an, mit $N = \sum_{i=0}^{n} b_i 2^i$ mit $b_i \in \{0, 1\}, b_n = 1$, so ergibt sich durch

$$O\left(N^{\frac{1}{4}}\right) = O\left((2^n)^{\frac{1}{4}}\right) = O\left(2^{\frac{n}{4}}\right)$$

eine exponentielle Laufzeit. Im Vergleich zur Probedivision können daher also auch Zahlen mit einer doppelt so großen Anzahl an Stellen faktorisiert werden, wobei zu beachten ist, dass ein Erfolg nicht sicher ist. Zusammenfassend lässt sich sagen, dass sich die Pollard'sche Rho-Methode für Zahlen N, bestehend aus mehreren verhältnismäßig kleinen Primfaktoren gut eignet. ([9], S. 340-342, [15], S. 9, [10], S. 99)

4.3.2 (p-1)-Faktorisierungsmethode

Im Jahre 1974 entwickelte John M. Pollard ein weiteres Faktorisierungsverfahren, nämlich die $(p-1)$-Faktorisierungsmethode. Dieses Verfahren macht sich die speziellen Eigenschaften der Gruppe $(\mathbb{Z}/p\mathbb{Z}^{\times}, \cdot)$ und den Kleinen Satz von Fermat zu Nutze. Im Wesentlichen wird versucht, die Elemente dieser Gruppe durch wiederholte Verknüpfungen mit sich selbst in Untergruppen mit geringerer Ordnung zu bringen. Da es sich um eine multiplikative Gruppe handelt, spricht man beim Verknüpfen klarerweise vom Potenzieren. Folgender Satz bezieht sich auf die Grundidee dieses Algorithmus. ([10], S. 107)

Satz 4.9. *Es sei G eine multiplikative zyklische Gruppe mit der Ordnung s. Für ein $r \in \mathbb{N}$ sei*

$$f_r : G \longrightarrow G, \quad a \mapsto a^r$$

eine Abbildung, welche jedem Gruppenelement die r-te Potenz zuordnet. Für das Bild der Abbildung f_r gilt daher

$$\mathrm{Card}\left(\mathrm{Im}\left(f_r\right)\right) = \frac{s}{gcd(r, s)}$$

(Das heißt, die Anzahl der Elemente der Bildmenge ist durch den Quotient von s und $gcd(r, s)$ berechenbar.)

Beweis. Man wähle ein erzeugendes Element $x \in G$ und definiere $d := \gcd(r, s)$. Nun gilt es zu beweisen, dass mit x^d das Bild von f_r erzeugt wird. Unter Verwendung des Erweiterten Euklidischen Algorithmus lässt sich $d = vr + ws$ mit $v, w \in \mathbb{Z}$ als Linearkombination von r und m anschreiben. Daher ist

$$x^d = x^{vr+ws} = x^{vr} \cdot x^{ws} = (x^v)^r \in \mathrm{Im}\,(f_r)$$

ein Element der Bildmenge, wobei nach Satz 2.28. $x^{ws} = 1$ gelten muss. Auf der anderen Seite ist jedes Element $b \in \mathrm{Im}\,(f_r)$ darstellbar als

$$b = a^r = \left(x^i\right)^r = \left(x^i\right)^{jd} = \left(x^d\right)^{ij}$$

wobei $i, j \in \mathbb{Z}$. Die Zahl r wurde hierbei als $r = jd$ für ein bestimmtes j gewählt, da d klarerweise ein Teiler von r sein muss. Somit wäre die Korrektheit der Behauptung gezeigt. Für die Elementanzahl der Bildmenge $\mathrm{Im}\,(f_r)$ gilt daher

$$\left(x^d\right)^n \qquad \text{für } 0 \leqslant n < \frac{s}{d}$$

wie man leicht durch Umformung von $nd < r$ erkennen kann. Setzt man für $d = gcd(r, s)$ ein, ist der Satz bewiesen. ([10], Satz 14.1., S. 107) □

Bezogen auf die zyklische Gruppe $(\mathbb{Z}/p\mathbb{Z}^\times, \cdot)$ und deren Ordnung $p-1$ sieht man sofort, dass die Anzahl der Elemente der Untergruppe aller r-Potenzen umso geringer ist, je größer $gcd(r, s)$ ist. Falls der Exponent r bereits ein Vielfaches der Gruppenordnung $p-1$ ist, so gilt nach dem Kleinen Satz von Fermat, dass die daraus entstehende Untergruppe nur aus einem Element, nämlich der 1 besteht. ([10], S. 107) Darüber hinaus gilt es zu erwähnen, dass in der Kryptografie mit Gruppen gearbeitet wird, welche bestenfalls keine kleinen Untergruppen besitzen. Wie bereits erwähnt, kann man durch wiederholte Verknüpfungen mit sich selbst ein Element in eine kleinere Untergruppe befördern. Wenn diese Untergruppe nur aus wenigen Elementen besteht, können durch weitere Verknüpfungen nur diese geringe Anzahl an Elementen in Frage kommen. Es stellt sich nun die Frage, wann eine Gruppe viele Untergruppen hat. Nach dem Satz von Lagrange teilt die Ordnung einer Untergruppe stets die Ordnung der Gruppe selbst. Bezogen auf die Gruppe $(\mathbb{Z}/p\mathbb{Z}^\times, \cdot)$ mit deren Ordnung $p-1$ ergibt sich, dass wenn $p-1$ viele Teiler hat, es auch dementprechend viele Untergruppen gibt. Um die Anzahl an Untergruppen möglichst gering zu halten sollte die Primzahl p spezielle Eigenschaften aufweisen. Da p prim ist, muss $p-1$ eine gerade Zahl sein, wodurch man $p-1$ auch schreiben kann als

$$p - 1 = 2 \cdot q$$

Aufgrund dessen ergeben sich für $p-1$ neben den trivialen Teilern 1 und $2q$, die Teiler 2

und q. Falls q ebenfalls eine Primzahl sein sollte, existieren nur diese vier Teiler. Ist q also eine Primzahl, so entstehen nach dem Satz von Lagrange vier unterschiedliche Untergruppen. Anhand eines Beispiels soll diese Erkenntnis erläutert werden.

Beispiel. Mittels Microsoft Excel sollen sämtliche Untergruppen der multiplikativen Gruppe $(\mathbb{Z}/11\mathbb{Z}^\times, \cdot)$ dargestellt werden. Da für die Primzahl 11 die oben genannte Eigenschaft gilt,

	A	B	C	D	E	F	G	H	I	J
1	p=11	p-1=10=2*5								
2										
3	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^10
4	**1**									
5	2	4	8	5	10	9	7	3	6	1
6	**3**	**9**	**5**	**4**	**1**					
7	**4**	**5**	**9**	**3**	**1**					
8	**5**	**3**	**4**	**9**	**1**					
9	6	3	7	9	10	5	8	4	2	1
10	7	5	2	3	10	4	6	9	8	1
11	8	9	6	4	10	3	2	5	7	1
12	**9**	**4**	**3**	**5**	**1**					
13	**10**	**1**								

Abbildung 12: Untergruppen von $(\mathbb{Z}/11\mathbb{Z}^\times, \cdot)$

existieren vier unterschiedliche Untergruppen, wobei die echten Untergruppen farblich markiert sind im Gegensatz zu den nicht markierten trivialen Untergruppen. Unter einer trivialen Untergruppe versteht man eine Untergruppe, welche mit der Gruppe selbst übereinstimmt oder nur aus dem Element 1 besteht. Außerdem erzeugen bis auf die Elemente 1 und $(11-1)$ die Hälfte der übrigen Elemente die gesamte Gruppe, während die andere Hälfte von Elemente eine Untergruppe mit Ordnung 5 erzeugen.

Das Beispiel zeigt, dass beinahe die Hälfte aller Elemente von $(\mathbb{Z}/p\mathbb{Z}^\times, \cdot)$ die gesamte Gruppe erzeugt. Dies ist jedoch nur dann der Fall, wenn $p - 1 = 2 \cdot q$ für eine Primzahl q gilt. Man nennt eine Primzahl p mit dieser Eigenschaft auch starke Primzahl, wie folgende Definition zeigt.

Definition (Starke Primzahl). Hat eine Primzahl p die Form

$$p = 2 \cdot q + 1$$

wobei q eine Primzahl ist, so nennt man p starke Primzahl.

Bemerkung. Beispielsweise sind die Primzahlen $5, 7, 11, 23, 39$ starke Primzahlen. In der englischen Sprache spricht man hier von sogenannten *safe primes*, da derartige Zahlen in der Kryptografie für mehr Sicherheit sorgen. ([1], S. 83)

Mittels dieser Erkenntnisse hat man gesehen, dass je nach Wahl der Primfaktoren p einer zusammengesetzten Zahl, die dazugehörigen Zahlen $p - 1$ verschieden viele und große

Untergruppen besitzen. ([5], S. 81-83) Der $(p-1)$-Algorithmus nutzt diese Eigenschaften der multiplikativen Gruppe $(\mathbb{Z}/p\mathbb{Z}^{\times}, \cdot)$ um Zahlen wie folgt zu faktorisieren. Sei N eine zusammengesetzte Zahl mit einem Primfaktor p und $p-1 = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ mit $p_1 \geq p_2 \geq \ldots \geq p_k$ die Primfaktorzerlegung der Zahl $p-1$. Ist eine Zahl m durch alle Faktoren $p_i^{\alpha_i}$ mit $1 \leq i \leq k$ von $p-1$ teilbar, also durch $p-1$ selbst teilbar, dann gilt nach dem Kleinen Satz von Fermat

$$a^m \equiv 1 \bmod p$$

falls p und a teilerfremd sind. Findet man nun eine derartige Kongruenz $a^m \equiv 1 \bmod p$ und gilt zusätzlich dazu $a^m \not\equiv 1 \bmod N$, so hat man mit $gcd(a^m - 1, N)$ einen echten Teiler von N gefunden. Gilt die Kongruenz $a^m \equiv 1 \bmod p$ genau für einen Primfaktor p, so ergibt sich sogar $p = gcd(a^m - 1, N)$, wodurch man sogar einen Primfaktor von N anstelle eines womöglich zusammengesetzten Teilers erhält. Da man die Primfaktoren von N klarerweise im vorhinein nicht kennt, muss man den Exponent m derart wählen, sodass $(p-1) \mid m$ gilt. Eine mögliche Vorgehensweise wäre eine Schranke $B > 0$ festzulegen und $m = B!$ zu rechnen. Im Normalfall reicht es aber für eine Schranke $B > 0$ alle Primzahlen p und deren Primzahlpotenzen p^{β} zu betrachten, wobei man unter $\beta := \beta(p, B)$ den größtmöglichen Exponenten der Primzahlen p versteht, damit $p^{\beta} \leq B$. B sollte daher eine Zahl sein, für welche $p-1$ potenzglatt ist, also alle Primteiler (und ggf. deren Potenzen) von $p-1$ kleiner oder gleich B sind. Berechnet man nun das Produkt

$$m := \prod_{p \leqslant B} p^{\beta(p,B)}$$

und sollten alle Primteiler (und deren Potenzen) von $p-1$ auch $B-$potenzglatt sein, so gilt $(p-1) \mid m$. Andernfalls muss man die Schranke B solange erhöhen, bis man erfolgreich ist. Je nach Beschaffenheit der Zahlen kann es auch reichen, wenn man nur die kleinen Primfaktoren mehrfach in die Berechnungen einfließen lässt und bei den größeren nur die Primzahl selbst berücksichtigt und nicht deren Primzahlpotenzen. Je nach Größe der Primfaktoren von $p-1$ stellt sich nun die Frage, ob man ab einer gewissen Größe $a^m - 1 \bmod N$ überhaupt noch effizient berechnen kann. Für diese Berechnung großer Potenzen kann der in Abschnit 2.4.3 vorgestellte Square-and-Multiply-Algorithmus herangezogen werden, welcher auch für große Potenzen sehr effizient arbeitet. Die $(p-1)$-Faktorisierungsmethode berechnet die Kongruenz $a^m - 1 \bmod N$ nicht auf einmal, sondern Schritt für Schritt. Dazu wird für alle Primzahlpotenzen $q_i := p_i^{\alpha_i}$ von m nacheinander $a_{i+1} := a_i^{q_i} \bmod N$ mit $a_0 := a$ berechnet. Währenddessen wird immer wieder geprüft, ob $1 < gcd(a_i - 1, N) < N$ gilt, bis man einen Primfaktor p der Zahl N erhält. Sollte dies nicht der Fall sein, erhöht man die Schranke B um alle B-potenzglatten Primfaktoren von $p-1$ zu bekommen. ([10], S. 107-108, [9], S. 342) Insgesamt ergibt sich also folgender Algorithmus:

Algorithmus (Pollard's (p-1)-Verfahren)
Zu faktorisieren sei eine ungerade zusammengesetzte Zahl N:

1) Wähle eine natürliche Zahl a_0 mit $1 \leq a_0 < N$, $gcd(a_0, N) = 1$ und eine natürliche Zahl B als Schranke.
2) Berechne nun $a_{i+1} := a_i^{q_i} (\mathrm{mod} N)$ mit obigem a_0 für alle B-glatten Primzahlpotenzen q_i.
3) Berechne zwischendurch immer wieder $d := gcd(a_i - 1, N)$ solange bis $1 < d < N$ gilt. Dann ist d ein echter Teiler von N.
4) Sollten keine echten Teiler gefunden werden, erhöhe die Schranke B und gehe zu 2). Wiederhole das Verfahren bis die Zahl N vollständig faktorisiert wurde.

Anhand eines Beispiels soll nun die Vorgehensweise des Algorithmus erläutert werden.

Beispiel. Zu faktorisieren ist die zusammengesetzte Zahl $N = 164501$ mittels $(p-1)$-Algorithmus. Dabei soll $a_0 = 2$ und als Schranke $B = 30$ gewählt werden. Für die Exponenten q_i wurden die Primzahlen $2, 3$ dreifach und die Primzahl 5 zweifach genommen, wodurch diese 30-potenzglatt sind. Bei den übrigen Primzahlen wurden keine Potenzen berücksichtigt.

i	q_i	$a_i \equiv a_{i-1}^{q_i} (\mathrm{mod} N)$	$gcd(a_i - 1, N)$
1	2^3	256	1
2	3^3	152338	1
3	5^2	145490	1
4	7	122739	1
5	11	38936	1
6	13	44815	1
7	17	12867	919

Bereits nach sieben Iterationen erhält man den Primfaktor 919 von N. Nach einer Division von N durch 919 erhält man den zweiten Primfaktor, wie man leicht mittels Primzahltest überprüfen kann. Damit wurde $N = 164501 = 919 \cdot 179$ vollständig faktorisiert. Warum man hier relativ rasch erfolgreich war, erkennt man bei näherer Betrachtung von $p = 919$. Die Primfaktorzerlegung von $p - 1 = 918 = 2 \cdot 3 \cdot 3 \cdot 3 \cdot 17$. Da $p - 1$ aus verhältnismäßig kleinen Primfaktoren besteht und daher ein kleines B als Schranke ausreicht, hat man nach dem oben angegebenen Algorithmus in Schritt 2 und 3 sehr schnell Erfolg bei der Faktorisierung. Darüber hinaus ist ersichtlich, dass man Primzahlpotenzen berücksichtigen muss, da die 3 beispielsweise dreifach in $p - 1$ enthalten ist. Im Gegensatz dazu, hat der Komplementärteiler $q = 179$ mit $q - 1 = 178 = 2 \cdot 89$ einen weitaus größeren Teiler, wodurch dieser erst bei einer Schranke von $B \geq 89$ gefunden wird. Das heißt, die Zahl $q - 1$ ist weniger glatt als $p - 1$.

4.3.2.1 Optimierungen

Sollte man mit dem $(p-1)$-Faktorisierungsverfahren keinen Erfolg haben, dann war die gewählte Schranke B zu klein gewählt. Doch nach Satz 4.9. befindet sich das zuletzt berechnete Element $a_k = a_{k-1}^{q_k}$ bereits in einer kleineren Untergruppe, wobei die Ordnung dieser Untergruppe $r := \frac{(p-1)}{gcd(k,p-1)}$ beträgt. Wie bereits erwähnt werden für den Exponenten k sämtliche Primzahlpotenzen kleiner oder gleich einer gewählten Schranke B miteinander multipliziert. Ergibt sich für r eine Primzahl, bedeutet dies, dass bei der Berechnung des Exponenten lediglich eine Primzahl fehlt um bei der Faktorisierung erfolgreich zu sein. Diese Primzahl ist daher größer als die Schranke B. Sollte $r := p_1$ nun eine Primzahl sein, so kann man eine Optimierung namens *Big Prime Variation* durchführen. Diese Form der Optimierung spielte bereits beim Quadratischen Sieb eine wesentliche Rolle. Sei dazu $b := a^{p_2 \cdots p_k}$. Findet man die fehlende Primzahl p_1 und gilt zusätzlich $b^{p_1} \not\equiv 1 \bmod N$, so hat man mit $gcd(b^{p_1} - 1, N)$ einen echten Teiler von N gefunden. Da man die Primzahl p_1 aber nicht kennt, versucht man es mit allen Primzahlen $p_j < C$, wobei $C > B$ eine neu gewählte größere Schranke ist. Um diese Berechnungen zu erleichtern verwendet man die Differenzen zwischen diesen Primzahlen. Es gilt nämlich $b^{p_{j+1}} = b^{p_j + d_j} = b^{p_j} \cdot b^{d_j}$, wobei mit d_j die jeweiligen Primzahldifferenzen gemeint sind. Somit muss man anstelle der weitaus größeren Potenzen b^{p_j} nur die Potenz b^{p_1} und die kleineren Potenzen b^{d_j} berechnen und miteinander multiplizieren. Die jeweiligen Primzahldifferenzen sind relativ klein und viele Primzahlen haben sogar die gleiche Differenz. Aufgrund dessen können die Zahlen b^{d_j} bereits im voraus berechnet und gespeichert werden. Der Aufwand für die Speicherung dieser Werte ist verhältnismäßig gering, da die entsprechenden Primzahldifferenzen d_j unter Verwendung des Primzahlsatzes eine ungefähre Größe von $\ln p_j$ aufweisen. Insgesamt sind dadurch die Berechnungen deutlich schneller, weswegen die zweite Schranke C erheblich größer sein darf als B. Somit ist es möglich bei Zahlen $p-1$ mit einem sehr großen Primfaktor erfolgreich zu sein.

Ähnlich wie bei der Pollard'schen Rho-Methode können auch beim $(p-1)$-Verfahren mehrere *gcd*-Berechnungen zusammengefasst werden, da diese ansonsten zu rechenintensiv sind. Es werden dabei also n Werte $a_i - 1$ miteinander multipliziert und man berechnet anschließend

$$gcd\left(\prod_{i=i_0}^{i_0+n} (a_i - 1), N\right)$$

um einen Teiler von N zu erhalten. ([9], S. 343-345, [10], S. 111-112) In *ARIBAS* gibt es sowohl eine Funktion für das „normale" $(p-1)$-Verfahren, als auch eine Funktion, welche die *Big Prime Variation* berücksichtigt. Im Gegensatz zur Funktion „p1_factorize" führt die Funktion „p1_factbpv" die *Big Prime Variation* aus. Anhand eines Beispiels sollen beide Funktionen erläutert werden.

Beispiel. Faktorisiert werden soll die Mersenne Zahl $M_{79} = 2^{79} - 1$ mittels der ARIBAS-Funktionen „p1_factorize" und „p1_factbpv", wobei sämtliche Arbeitsschritte in Abbildung

13 ersichtlich sind.

```
==> M79:=2**79-1.
-: 6044_62909_80731_45873_53087

==> N1:=p1_factorize(M79,10000).
working .
factor found with bound 128
-: 2687

==> N2:=M79 div N1.
-: 2_24958_28426_02584_99201

==> N3:=p1_factorize(N2,10000).
working ..................................................................
.........
-: 0

==> N3:=p1_factbpv(N2,10000,10**8).
working ....................................
entering big prime variation .......
-: 202029703

==> N4:= N2 div N3.
-: 111_34911_39767
```

Abbildung 13: (p-1)- Faktorisierungsmethode in ARIBAS

Die Funktion „p1_factorize“ hat als erste Zwischenschranke die Zahl 128. Sollte man mit dieser keinen echter Teiler finden, so werden solange Zwischenschranken mit einem Abstand von 128 gebildet, bis man die selbst gewählte globale Schranke B erreicht hat. Der erste Teiler N_1 wurde beispielsweise bereits mit der Zwischenschranke 128 gefunden, wobei man als globale Schranke 10000 wählte. In der folgenden Abbildung 14 ist ersichtlich, warum dieser Faktor so schnell gefunden wurde. Die Zahl $N_1 - 1$ besteht aus relativ kleinen Primfaktoren, wobei der größte 79 ist und somit kleiner ist als die erste Zwischenschranke 128. Mittels Primzahltests wurde im Anschluss festgestellt, dass es sich bei der Zahl N_1 um eine Primzahl handelt, während der Komplementärteiler N_2 eine zusammengesetzte Zahl ist. Eine weitere Faktorisierung von N_2 mit „p1_factorize“ und 10000 als globaler Schranke liefert keinen Erfolg, da die Schranke zu klein gewählt war. Mit der Funktion „p1_factbpv“ wurde nun bis zur ersten Schranke 10000 mit der allgemeinen Version des Verfahrens gerechnet. Da kein echter Teiler gefunden werden konnte, benutzt man die *Big Prime Variation* mit der zweiten Schranke 10^8. Mit dieser höheren Schranke konnte ein weiterer Primfaktor N_3 von N gefunden werden. Betrachtet man abermals die Primfaktorzerlung von $N_3 - 1$, sieht man sofort, warum man mit der *Big Prime Variation* einen echten Teiler gefunden hat. Neben mehreren kleinen Primfaktoren besitzt $N_3 - 1$ auch den weitaus größeren Teiler 60889. Dieser konnte klarerweise mit der allgemeinen Version des Verfahrens mit der Schranke 10000 nicht gefunden werden. Eine erneute Division von N_2 durch N_3 liefert den Faktor N_4, welcher ebenfalls prim ist.

Infolgedessen wurde die Zahl $N = N_1 \cdot N_3 \cdot N_4$ vollständig faktorisiert.

```
==> factorlist(N1-1).
-: (2, 17, 79)

==> factorlist(N3-1).
-: (2, 3, 7, 79, 60889)
```

Abbildung 14: Primfaktorzerlegung von $N_1 - 1$ und $N_3 - 1$ in ARIBAS

4.3.2.2 Laufzeit

Um einen Faktor p mittels $(p-1)$-Verfahren zu finden benötigt man $O(p_1)$ Operationen, wobei man unter p_1 den größten Primfaktor von $p-1$ versteht. Klarerweise kennt man die Größe des Primfaktors p_1 im vorhinein nicht, wodurch man nicht sagen kann, ob das Verfahren für bestimmte Zahlen effizient arbeitet oder nicht. Mit dem folgenden Satz kann man wenigstens eine ungefähre Abschätzung der Laufzeit geben. ([9], S. 345)

Satz 4.10 (von Hardy/Ramanujan). *Wenn $\Omega(N)$ die Anzahl aller Primteiler einer Zahl N ist, so kann man diese durchschnittlich mit $\Omega(N) \approx \ln\ln(N)$ abschätzen. ([9], Satz 27.7, S. 345)*

Besitzt $p - 1 = p_1 \cdots p_k$ eine derartige Primfaktorzerlegung, so gilt nach obigem Satz $k \approx \ln\ln p$. Darüber hinaus ist

$$\begin{aligned} k - 1 \approx \ln\ln\left(\frac{p}{p_1}\right) &= \ln\left(\ln(p) - \ln(p_1)\right) = \ln\left(\ln(p) \cdot \left(1 - \frac{\ln(p_1)}{\ln(p)}\right)\right) \\ &= \ln\ln(p) + \ln\left(1 - \frac{\ln(p_1)}{\ln(p)}\right) \approx k + \ln\left(1 - \frac{\ln(p_1)}{\ln(p)}\right) \end{aligned}$$

Löst man die letzte Abschätzung mittels Technologieeinsatz nach p_1 auf, so erhält man

$$\ln(p_1) \approx \left(1 - \frac{1}{e}\right)\ln(p) \approx 0,632 \cdot \ln(p)$$

Man bekommt daher einen ungefähren „Erwartungswert“ von $O\left(p^{0,632}\right)$ für die Laufzeit, wobei man beachten muss, dass die Varianz sehr hoch ist. Sollte $N = p \cdot q$ beispielsweise aus zwei nahezu gleich großen Primfaktoren bestehen, so wären $p, q \approx \sqrt{N}$. Mit den vorigen Überlegungen würde sich nun eine Laufzeit von $O\left(\left(N^{\frac{1}{2}}\right)^{0,632}\right) = O\left(N^{0,316}\right)$. Schreibt man N nun als Binärzahl an, mit $N = \sum_{i=0}^{n} b_i 2^i$ mit $b_i \in \{0,1\}, b_n = 1$, so erhält man durch

$$O\left(N^{0,316}\right) = O\left((2^n)^{0,316}\right) = O(2^{n \cdot 0,316}) \approx O\left(2^{\frac{n}{3}}\right)$$

eine exponentielle Laufzeit für das $(p-1)$-Verfahren. Unter Berücksichtigung der hohen Varianz dieser Abschätzung wäre das $(p-1)$-Verfahren somit langsamer als die Rho-Methode.

Je nach Beschaffenheit der Primfaktoren können jedoch auch sehr große Zahlen effzient faktorisiert werden. Gilt beispielsweise für einen Primfaktor p von N, dass $p-1$ nur aus kleinen Primfaktoren besteht, so ist das $(p-1)$-Faktorisierungsverfahren sehr effizient. ([9], S. 345)

4.3.3 Elliptische Kurven Methode

Die 1985 von Hendrik Lenstra erfundene *Elliptische Kurven Methode*, oder auch kurz *ECM* für ***E**lliptic **C**urve **M**ethod*, baut auf der vorhin beschriebenen $(p-1)$-Faktorisierungsmethode auf, mit dem Unterschied, dass mit elliptischen Kurven über dem Restklassenkörper $\mathbb{F}_p$ gearbeitet wird anstelle von $(\mathbb{Z}/m\mathbb{Z})$. ([6]) Das $(p-1)$-Verfahren ist im Wesentlichen von der Glattheit der Zahl $p-1$ eines Primfaktors p abhängig. Sollte die Schranke B dabei zu klein gewählt worden sein und $p-1$ nicht B-potenzglatt sein, kann man als nächsten Schritt lediglich die Schranke erhöhen und hoffen, dass alle Primzahlpotenzen von $p-1$ miteinbezogen werden. Mit der Elliptischen Kurven Methode kann jedoch die Chance einen Primfaktor p zu finden deutlich erhöht werden, indem man einfach mehrere verschiedene elliptische Kurven für die Berechnungen heranzieht. Damit ist die hier vorgestellte Methode für eine erheblich größere Menge an Zahlen zur Faktorisierung geeignet als das $(p-1)$-Verfahren. In den mathematischen Grundlagen wurden die elliptischen Kurven allgemein über einem noch nicht näher definierten Körper K angeführt. Für die Kryptografie oder den hier vorgestellten Faktorisierungsalgorithmus spielen hauptsächlich elliptische Kurven über dem Restklassenkörper $\mathbb{F}_p$ eine Rolle. ([9], S. 320, 345-346). Ab jetzt werden elliptische Kurven ausschließlich in Form der kurzen Weierstraß-Gleichung angeführt, das heißt es gilt $char(K) \notin \{2,3\}$. Aufgrund dieser leicht veränderten Definition ergeben sich auch leichte Abänderungen für die Verknüpfung von Punkten auf der elliptischen Kurve. Wegen der Beschaffenheit der nun verwendeteten Polynomgleichung ist die horizontale Symmetrieachse der elliptischen Kurve $\overline{E}(K)$ die x-Achse selbst. Dies bedeutet, dass für die Invertierung eines Punktes (x,y) lediglich die y-Koordinate mit -1 multipliziert werden muss. Insgesamt ergibt sich für zwei Punkte $P=(x_1,y_1)$, $Q=(x_2,y_2)$ und die Summe von $P \oplus Q = (x_3,y_3)$ folgendes:

1) Das Inverse von P ist $-P=(x_1,-y_1)$. Daher gilt: ist $P \oplus Q = O$, so ist $Q=-P$.
2) Ist $P \oplus Q = (x_3,y_3)$, so gilt für die Koordinaten $x_3 = k^2 - x_1 - x_2$ und $y_3 = -y_1 + k(x_1 - x_3)$

$$k = \begin{cases} \frac{y_1-y_2}{x_1-x_2} & \text{wenn } P \neq Q \\ \\ \frac{3x_1^2+a}{2y_1} & \text{wenn } P = Q \end{cases}$$

Nach der Definition der Verknüpfung in den Grundlagen müsste für $x_3 = k^2 + a_1k - a_2 - x_1 - x_2$ gelten. Für die kurze Weierstraß-Gleichung gilt jedoch, dass die Koeffizienten $a_1 = a_2 = a_3 = 0$. Unter Berücksichtigung dieser Erkenntnis ergibt sich obige Darstellung für x_3. Wie bereits erwähnt muss zur Invertierung die y-Koordinate des Schnittpunkts y_3' mit der Geraden oder

der Tangente mit -1 multipliziert werden. Dies bedeutet, dass $y_3 = (y_3') \cdot (-1)$ und daher ergibt sich für y_3 die angeführte Darstellung. Weiters ist für $P \neq Q$ die Steigung klarerweise $k = \frac{y_1 - y_2}{x_1 - x_2}$, wie bereits in den Grundlagen angeführt wurde. Die Tangentensteigung für den Fall, dass $P = Q$, erhält man mittels implizitem differenzieren:

$$\begin{aligned} & y^2 - x^3 - ax - b = 0 \\ \Longrightarrow & 2yy' - 3x^2 - a = 0 \Longleftrightarrow y' = \frac{3x^2 + a}{2y} \end{aligned}$$

Nach dieser Einschränkung des Begriffs der elliptischen Kurven und den damit verbundenen Auswirkungen auf die Verknüpfung, kann nun mit der Beschreibung der elliptischen Kurven über dem Restklassenkörper $\mathbb{F}_p$ begonnen werden. ([9], S. 346) Es ist bereits bekannt, dass $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ genau dann ein Körper ist, wenn m eine Primzahl p ist. Aufgrund dessen ist eine elliptische Kurve $\overline{E}(\mathbb{F}_p)$ mit der genannten Verknüpfung auch eine abelsche Gruppe. Eine elliptische Kurve, definiert durch die Gleichung $y^2 = x^3 - 2x + 1$ modulo einer Primzahl $p = 7$, besteht nun aus jenen Punkten $(x, y) \in (\mathbb{Z}/7\mathbb{Z})$, welche die folgende Kongruenz erfüllen:

$$y^2 \equiv x^3 - 2x + 1 \bmod 7$$

Für x kommen die Zahlen $0, 1, 2, 3, 4, 5, 6$ in Frage, für welche nach dem Einsetzen in die obige Gleichung die entsprechenden Werte für y berechnet werden müssen. Man bekommt mit

$$\begin{aligned} x \equiv 0 \quad & \Rightarrow y^2 \equiv 1 && \Rightarrow y \equiv 1, 6 \\ x \equiv 1 \quad & \Rightarrow y^2 \equiv 0 && \Rightarrow y \equiv 0 \\ x \equiv 2 \quad & \Rightarrow y^2 \equiv 5 && \Rightarrow \text{ keine Lösungen} \\ x \equiv 3 \quad & \Rightarrow y^2 \equiv 22 \equiv 1 && \Rightarrow y \equiv 1, 6 \\ x \equiv 4 \quad & \Rightarrow y^2 \equiv 57 \equiv 1 && \Rightarrow y \equiv 1, 6 \\ x \equiv 5 \quad & \Rightarrow y^2 \equiv 116 \equiv 4 && \Rightarrow y \equiv 2, 5 \\ x \equiv 6 \quad & \Rightarrow y^2 \equiv 205 \equiv 2 && \Rightarrow y \equiv 3, 4 \end{aligned}$$

die Punkte der elliptischen Kurve $(0, 1)$, $(0, 6)$, $(1, 0)$, $(3, 1)$, $(3, 6)$, $(4, 1)$, $(4, 6)$, $(5, 2)$, $(5, 5)$, $(6, 3)$, $(6, 4)$, O. Für das Lösen derartiger quadratischer Kongruenzen können die Erkenntnisse zum Thema quadratische Reste in den Grundlagen verwendet werden. Stellt man nun diese Punkte wie in Abbildung 15 grafisch dar, so erhält man keine durchgängigen Linien wie bei den Beispielen der Grundlagen, sondern lediglich die Punkte selbst. Die Verknüpfung in Form der Punktaddition bleibt dabei für elliptische Kurven über dem Körper $\mathbb{F}_p$ mit den bisherigen Formeln gleich. Da in den hergeleiteten Formeln rationale Zahlen vorkommen, werden diese

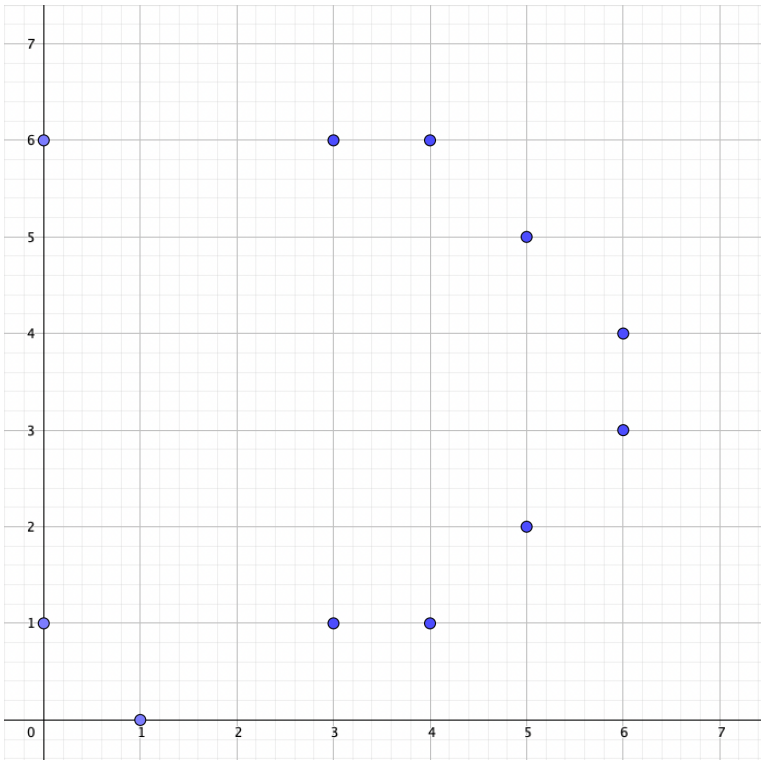

Abbildung 15: Elliptische Kurve über $\mathbb{F}_7$ in GeoGebra

im Zuge von elliptischen Kurven über $\mathbb{F}_p$ anders behandelt. Beispielsweise ist eine rationale Zahl $\frac{x}{y}$ nichts anderes als xy^{-1}, wobei $yy^{-1} \equiv 1 \bmod p$ gelten muss. Anstelle einer Divison wird hier also mit dem Inversen des Divisors multipliziert, vorausgesetzt dieses existiert. Nach Satz 2.31 existiert dieses Inverse Element genau dann, wenn $gcd(y, p) = 1$. Bis jetzt wurden nur elliptische Kurven über einem Köper K betrachtet. In der Praxis wird jedoch auch mit elliptischen Kurven über dem Restklassenring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ mit einem zusammengesetzten m als Modul gearbeitet. In Anbetracht der bisherigen Verknüpfung erhält man mit $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$, falls m nicht prim ist aber keine Gruppe. Denn nach Satz 2.30 gilt für ein $a \in \mathbb{Z}/m\mathbb{Z}$ mit $gcd(a, m) \neq 1$, dass a ein Nullteiler in $\mathbb{Z}/m\mathbb{Z}$ ist. Nach Satz 2.31 gilt wiederum, dass Nullteiler kein Inverses besitzen. Für die Divisionen in den Berechnungen im Zuge der Punktaddition ist es jedoch notwendig, dass die verwendeten Elemente ein Inverses besitzen, da ansonsten die Berechnung der Punktaddition über $\mathbb{Z}/m\mathbb{Z}$ nicht möglich ist (siehe Formel für k auf S. 76 unten). Hier knüpft nun die Idee der Elliptischen Kurven Methode zur Faktorisierung an. Sollte eine Punktaddition $P \oplus Q$ mit den Punkten $P = (v_1, w_1)$ und $Q = (v_2, w_2)$ fehlschlagen, so weiß man, dass es sich beim Modul m von $\mathbb{Z}/m\mathbb{Z}$ um eine zusammengesetzte Zahl handeln muss. Dies liegt daran, dass man bei den dafür notwendigen Divisionen im Rahmen der Punktaddition durch einen Nullteiler $(v_1 - v_2)$ dividieren muss, was ja bekanntermaßen nicht möglich ist. Berechnet man nun den größten gemeinsamen Teiler des Nullteilers $(v_1 - v_2)$

mit m, so erhält man im Allgemeinen einen echten Teiler von m. Doch bevor die Idee dieser Faktorisierungsmethode konkretisiert wird, soll noch ein Beispiel der Punktaddition mit der in Abbildung 15 dargestellten elliptischen Kurve $\overline{E}(\mathbb{F}_7)$ durchgeführt werden. ([14], S. 352-353)

Beispiel. Für die durch $y^2 \equiv x^3 - 2x + 1 (\mathrm{mod}\, 7)$ berechnete elliptische Kurve soll $(3,1) \oplus (1,0)$ berechnet werden.

$$k \equiv \frac{1-0}{3-1} \equiv \frac{1}{2} \equiv 1 \cdot 2^{-1} \equiv 4 (\mathrm{mod} 7)$$

Daher ergibt sich für die Koordinaten (x_3, y_3):

$$x_3 \equiv k^2 - x_1 - x_2 \equiv 4^2 - 3 - 1 \equiv 5 (\mathrm{mod} 7)$$
$$y_3 \equiv -y_1 + k\,(x_1 - x_3) \equiv -1 + 4(3-5) \equiv -9 \equiv 5 (\mathrm{mod} 7)$$

Insgesamt erhält man als Ergebnis

$$(3,1) \oplus (1,0) = (5,5)$$

für die Verknüpfung der angegebenen Punkte. ([14], S. 353)

Für die Berechnung der inversen Elemente kann der Erweiterte Euklidische Algorithmus herangezogen werden. Mit diesem wurde das Inverse $2^{-1} \equiv 4 (\mathrm{mod} 7)$ des obigen Beispiels bestimmt. Das eben durchgeführte Beispiel zeigt auch sehr gut, dass die bildliche Vorstellung der Verknüpfung nicht verloren geht. Legt man eine Gerade durch die Punkte $(3,1)$ und $(1,0)$, so ergibt sich für den dritten Schnittpunkt $(5,2)$, wie man in Abbildung 15 überprüfen kann. Im Beispiel war das Ergebnis der Punktaddition der Punkt $(5,5)$. Dieser ist jedoch das Inverse des Schnittpunkts $(5,2)$, wie man nach obigen Rechenregeln durch Multiplikation der y-Koordinate mit -1 leicht einsehen kann. Denn es gilt

$$5 \cdot (-1) \equiv -5 \equiv 2 \bmod 7$$

womit gezeigt wurde, dass die Punkt $(5,2)$ und $(5,5)$ invers zueinander sind. Dies ist jedoch auch grafisch ersichtlich, da eine Gerade durch diese beiden Punkte senkrecht wäre, womit sich als dritter Schnittpunkt der unendlich ferne Punkt ergeben würde. Nach Definition wären diese Punkte somit invers zueinander. Manchmal ist es aufgrund der Modulo-Rechnung notwendig, eine Gerade oder Tangente, die einen der Ränder verlässt um einen dritten Schnittpunkt zu erhalten, am gegenüberliegenden Rand weiterzuführen ([5], S. 132). Der kommende Satz spielt ebenfalls eine entscheidende Rolle für die Elliptische Kurven Methode. Nach Satz 2.28 gilt für eine endliche Gruppe, dass man nach n-maliger Verknüpfung eines Elements dieser Gruppe mit sich selbst, das neutrale Element erhält. Ein Spezialfall dieses Satzes ist der Kleine Satz von Fermat, welcher im $(p-1)$-Verfahren von großer Bedeutung ist. Da die elliptschen Kurven

$\overline{E}(\mathbb{F}_p)$ eine additiv geschriebene Gruppe sind, gilt nach Satz 2.28

$$n \cdot P = \underbrace{P \oplus P \oplus \ldots \oplus P}_{n- \text{ mal}} = O$$

wobei man unter $n = ord(\overline{E}(\mathbb{F}_p))$ die Ordnung der Gruppe, also die Anzahl der Punkte der elliptischen Kurve $\overline{E}(\mathbb{F}_p)$ bezeichnet. Um diese Ordnung abschätzen zu können, verwendet man den Satz von Hasse. ([9], S. 347)

Satz 4.11 (**Hasse**). *Ist $E(\mathbb{F}p)$ eine elliptische Kurve über dem Restklassenkörper $\mathbb{F}p$, dann gilt für die Ordnung $ord(E(\mathbb{F}p))$ der elliptischen Kurve*

$$p + 1 - 2\sqrt{p} \leq ord(E(\mathbb{F}p)) \leq p + 1 + 2\sqrt{p}$$

([9], Satz 27.9, S. 347)

Der Beweis dieses Satzes ist relativ komplex und würde den Rahmen dieser Arbeit sprengen, weswegen dieser hier nicht angeführt wird. Der obige Satz besagt, dass je nach Auswahl der Koeffizienten a, b der Polynomgleichung der Kurve, sich eine andere Ordnung der Gruppe beziehungsweise andere Anzahl an Punkten der elliptischen Kurve ergibt. Die Punkteanzahl schwankt innerhalb des angeführten Intervalls um p. ([9], S. 347) Darüber hinaus existiert auch ein spezieller Algorithmus zur Berechnung der Ordnung einer elliptischen Kurve, nämlich der Punktezähl-Algorithmus von Schoof. Mit diesem Algorithmus ist es möglich, die Ordnung mit polynomialer Laufzeit festzustellen. Darüber hinaus ist es möglich mit der Funktion „ecp_ord0“ in *ARIBAS* diese Ordnung zu berechnen. Als Argumente für diese Funktion sind nacheinander die Werte p, a, b einer elliptischen Kurve, welche gegeben ist durch $y^2 \equiv x^3 + ax + b \bmod p$, einzugeben.

Beispiel. Es soll die Ordnung (Punkteanzahl) der durch $y^2 \equiv x^3 - 2x + b \bmod 7$ mit $0 \leq b \leq 10$ gegebenen elliptischen Kurve mit der Funktion „ecp_ord0“ in ARIBAS berechnet werden. Zusätzlich werden über die Funktion „factorlist“ die jeweiligen Faktoren der Ordnungen angführt. In Abbildung 16 sieht man sofort, dass sich je nach Größe des Koeffizienten b verschiedene Gruppenordnungen ergeben, die sich aus kleineren oder auch größeren Primzahlen zusammensetzen. Manche Ordnungen sind daher weniger glatt (benötigen ein größeres B als Schranke) als andere. Darüber hinaus erhält man mit der elliptischen Kurve mit $b = 1$ genau jene Kurve, welche bereits in Abblidung 15 ersichtlich ist. Durch Zählen der Punkte in dieser Abbildung ergibt sich die Zahl 11, wobei der unendlich ferne Punkte O noch dazu gezählt werden muss. Daher ist die Ordnung der elliptischen Kurve 12 und somit ident zur Berechnung. ([10], S. 209)

Mit diesen Erkenntnissen über die Ordnung der elliptischen Kurven gilt es nun die Elliptische Kurven Methode zu konkretisieren. Dazu sei N eine zusammengesetzte Zahl mit einem

```
==> p:=7;
for b:=0 to 10 do N:=ecp_ord0(p,-2,b);
writeln(b:3,": ",N:4," ",factorlist(N));end.
  0:    8 (2, 2, 2)
  1:   12 (2, 2, 3)
  2:    9 (3, 3)
  3:    6 (2, 3)
  4:   10 (2, 5)
  5:    7 {7}
  6:    4 (2, 2)
  7:    8 (2, 2, 2)
  8:   12 (2, 2, 3)
  9:    9 (3, 3)
 10:    6 (2, 3)
```

Abbildung 16: Ordnungen der elliptischen Kurve in ARIBAS

Primfaktor p. Da man p im vorhinein klarerweise nicht kennt, muss man die Berechnungen in Form von Punktadditionen auf einer elliptischen Kurve über $(\mathbb{Z}/N\mathbb{Z}, +, \cdot)$ durchführen, anstatt über $\mathbb{F}_p$. Nachdem p definitionsgemäß ein Teiler von N ist, rechnet man einfach mit größeren Repräsentanten für die entsprechenden Werte modulo p. Für die Punktaddition $P \oplus Q$ mit den Punkten $P = (x_1, y_1)$ und $Q = (x_2, y_2)$ ist die Berechnung der Steigung k der Gerade oder Tangente von Nöten. Da diese Berechnungen über $(\mathbb{Z}/N\mathbb{Z}, +, \cdot)$ stattfinden, bedeutet eine Division nichts anderes als eine Multiplikation mit dem Inversen von $(x_2 - x_1)$. Im Zuge dessen müssen die Inversen $(x_2 - x_1)^{-1} \bmod N$ für den Fall $P \neq Q$ oder $(2y_1)^{-1} = (y_1 + y_2)^{-1} \bmod N$ für $P = Q$ berechnet werden. Diese Inversen existieren jedoch nicht. Berechnet man nun den größten gemeinsamen Teiler $gcd(x_2 - x_1, N)$ falls $P \neq Q$ oder $gcd(2y_1, N)$ wenn $P = Q$, so erhält man in der Regel den einen Faktor d von N. Um die gewünschte Addition zu generieren geht man ähnlich wie beim $(p-1)$-Verfahren vor. Beim $(p-1)$-Verfahren wird versucht ein spezielles Element mit einem Exponenten m zu potenzieren, wobei m ein Vielfaches der Ordnung eines Primfaktors sein muss. Unter Verwendung des Kleinen Satzes von Fermat erhält man schlussendlich die gewünschte Faktorisierung. Zur Berechnung der sich damit ergebenden großen Potenzen wird hierbei der Square-and-Multiply-Algorithmus verwendet. Anstelle von Multiplikationen addiert man bei der Elliptischen Kurven Methode jedoch einen Punkt P der elliptischen Kurve solange mit sich selbst bis die Berechnungen fehlschlagen. Man verwendet zur Berechnung von $m \cdot P$ abermals den Square-and-Multiply-Algorithmus. Da es sich aber bei elliptischen Kurven um eine additive Gruppenverknüpfung handelt, wäre der Name *Double-and-Add-Algorithmus* wohl passender. Ein wesentlicher Vorteil der Elliptischen Kurven Methode gegenüber dem $(p-1)$-Verfahren ist, dass man durch Änderung der Koeffizienten a, b die Gruppenordnung und damit verbunden auch die Glattheit beeinflussen kann, wodurch sich die Erfolgsaussichten deutlich erhöhen. Beim $(p-1)$-Verfahren ist die Ordnung durch den Primfaktor p fixiert und nur durch eine sukzessive Erhöhung der Schranke ist es möglich die Wahrscheinlichkeit für einen Erfolg zu erhöhen. Zusammenfassend ergibt sich also folgender

Algorithmus:

Algorithmus (Elliptische Kurven Methode)
Zu faktorisieren sei eine ungerade zusammengesetzte Zahl N:

1) Wähle eine beliebige elliptische Kurve E über $(\mathbb{Z}/N\mathbb{Z}, +, \cdot)$, einen Punkt $P \in E$ und eine natürliche Zahl B als Schranke.
2) Berechne nun $m \cdot P \bmod N$, wobei m das Produkt B-glatter Primzahlpotenzen ist.
3) Ist eine Berechnung der Punktadditionen nicht möglich, weil das Inverse Element von $x_2 - x_1$ - wobei x_1, x_2 die jeweiligen x-Koordinaten der zu addierenden Punkte sind - nicht existiert, so erhält man in der Regel mit $gcd(x_2 - x_1, N)$ einen echten Teiler von N.
4) Sollte kein echter Teiler gefunden werden, ändere die Koeffizienten der elliptischen Kurve und beginne erneut mit 1). Wiederhole das Verfahren solange bis die Zahl N vollständig faktorisiert wurde.

Nun soll der Algorithmus anhand eines Beispiels umgesetzt werden. Dabei wurden sämtliche Berechnungen mit dem Programm Microsoft Excel durchgeführt.

Beispiel. Faktorisiert werden soll die zusammengesetzte Zahl $N = 43229$ mit der Elliptischen Kurven Methode. Um eine elliptische Kurve und einen auf der Kurve liegenden Startpunkt P zu wählen geht man folgendermaßen vor. Man wählt zunächst einen beliebigen Punkt P und Koeffizienten a. Setzt man nun die jeweiligen Werte des Punktes P als auch a in die Gleichung $y^2 \equiv x^3 + ax + b \bmod N$ ein, so ergibt sich b von selbst und man erhält schlussendlich eine elliptische Kurve mit einem daraufliegenden Punkt. Man wählt $P = (1, 1)$ und $a = 1$. Daher gilt:

$$1^2 \equiv 1^3 + 1 \cdot 1 + b (\mathrm{mod} 43229) \Longleftrightarrow 1 \equiv 2 + b (\mathrm{mod} 43229) \Rightarrow b = -1$$

Es wird also die elliptische Kurve gegeben durch $y^2 \equiv x^3 + x - 1$ und der Punkt $P = (1, 1)$ zur Faktorisierung verwendet. Als Schranke wird $B = 12$ gewählt. Aufgrund dieser Schranke ergibt sich der Faktor $m = 2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 = 27720$, da jene Zahl 12-potenzglatt ist. Wie bereits erwähnt verwendet man hier den Double-and-Add-Algorithmus, da man ansonsten 27719 Addition durchführen müsste. Dazu muss m als Summe von Zweierpotenzen geschrieben werden. Hat man m bereits in Binär-Schreibweise ist man so gut wie fertig. In diesem Fall wäre die Zahl $m = 110110001001000$, wodurch man folgende Darstellung erhält:

$$m = 2^3 + 2^6 + 2^{10} + 2^{11} + 2^{13} + 2^{14}$$

Mittels wiederholter Punktverdopplung erhält man:

$$
\begin{aligned}
2^1P &= (2, 43226)\\
2^2P &= (22816, 20614)\\
2^3P &= (9669, 38522)\\
2^4P &= (28102, 4572)\\
2^5P &= (27086, 5242)\\
2^6P &= (31490, 22768)\\
2^7P &= (35452, 3793)\\
2^8P &= (9521, 9200)\\
2^9P &= (4367, 23589)\\
2^{10}P &= (33129, 32798)\\
2^{11}P &= (6020, 1457)\\
2^{12}P &= (359, 32201)\\
2^{13}P &= (32788, 39212)\\
2^{14}P &= (3492, 3344)
\end{aligned}
$$

Addiert man nun die benötigten Zweierpotenzen so ergibt sich:

$$
\begin{aligned}
2^3P \oplus 2^6P &= (17947, 25068)\\
2^3P \oplus 2^6P \oplus 2^{10}P &= (33054, 9148)\\
2^3P \oplus 2^6P \oplus 2^{10}P \oplus 2^{11}P &= (41359, 11092)\\
2^3P \oplus 2^6P \oplus 2^{10}P \oplus 2^{11}P \oplus 2^{13}P &= (1407, 12919)
\end{aligned}
$$

Zu guter Letzt muss zur bisherigen Summe noch $2^{14}P$ addiert werden. Die Punktaddition $(1407, 12919) \oplus (3492, 3344)$ ist jedoch nicht möglich, da $3492 - 1407 = 2085$ kein Inverses modulo 43229 besitzt. Berechnet man $gcd(2085, 43229) = 139$ erhält man den Primfaktor 139. Nach einer Division von 43229 durch 139 erhält man den zweiten Primfaktor 311. Die Zahl $N = 43229 = 139 \cdot 311$ wurde somit vollständig faktorisiert.

Betrachtet man die Ordnung dieser elliptischen Kurve über dem Körper $\mathbb{F}_{139}$ in Abbildung 17, so erkennt man, dass alle Faktoren dieser Ordnung kleiner als die gewählte Schranke $B = 12$ sind weshalb die Faktorisierung auch erfolgreich war. Zusätzlich liefert der chinesische Restsatz eine Begründung, warum das Faktorisierungsverfahren erfolgreich war. Denn laut diesem gilt

$$
\left.
\begin{aligned}
y^2 &\equiv x^3 + x - 1 \pmod{139}\\
y^2 &\equiv x^3 + x - 1 \pmod{311}
\end{aligned}
\right\} y^2 \equiv x^3 + x - 1 (\mathrm{mod} 139 \cdot 311)
$$

```
==> p:=139; n:=ecp_ord0(p,1,-1).
-: 154

==> factorlist(n).
-: (2, 7, 11)
```

Abbildung 17: Faktoren der Ordnung der elliptischen Kurve in ARIBAS

Da es sich bei 139 und 311 um Primzahlen handelt, sind die dazugehörigen elliptischen Kurven $\overline{E}(\mathbb{F}_{139})$ und $\overline{E}(\mathbb{F}_{311})$ versehen mit der Punktaddition auch abelsche Gruppen. Es ergibt sich nun nach Satz 2.28. bezüglich $\overline{E}(\mathbb{F}_{139})$ $154 \cdot P \equiv O \bmod 139$, aber auch $154 \cdot P \not\equiv O \bmod 311$ für $\overline{E}(\mathbb{F}_{311})$. Da $154 \cdot P$ nur modulo dem Primfaktor 139 das neutrale Element O ergibt und darüber hinaus die Berechnung $154 \cdot P$ modulo $139 \cdot 311 = 43229$ nicht möglich ist, war die Faktorisierung erfolgreich. ([14], S. 357)

4.3.3.1 Optimierungen

Verglichen mit dem $(p-1)$-Verfahren ergeben sich einige Vorteile für die Elliptische Kurven Methode. Das $(p-1)$-Verfahren ist von der Glattheit der Zahl $p-1$ eines Primfaktors p abhängig. Die Elliptische Kurven Methode ist hingegen davon abhängig, ob für eine gewählte Schranke B ausreichend glatte Zahlen in einem bestimmten Intervall um p existieren. Denn je nach Wahl der Koeffizienten ergibt sich eine andere Ordnung für die elliptische Kurve. ([14], S. 358) Des Weiteren kann man die Elliptische Kurven Methode sehr gut auf mehreren Rechnern parallelisieren, indem jeder Rechner mit einer anderen elliptischen Kurve gleichzeitig versucht dieselbe Zahl zu faktorisieren. Ähnlich wie bei der Pollard'schen Rho-Methode oder dem $(p-1)$-Verfahren können mehrere *gcd*-Berechnungen durch multiplizieren der jeweiligen Werte $(x_2 - x_1)$ zusammengefasst werden, um Rechenzeit zu sparen. Außerdem kann es auch bei der Elliptischen Kurven Methode der Fall sein, dass die Ordnung einer elliptischen Kurve modulo eines Primfaktors p, aus relativ kleinen Primfaktoren besteht, mit Ausnahme eines größeren Faktors. Hier leistet abermals die *Big Prime Variation* Abhilfe. Durch Angabe einer zweiten und weitaus größeren Schranke B_2 kann der noch fehlende Primfaktor der Ordnung wesentlich schneller ermittelt werden. Der Ablauf gleicht dabei dem des $(p-1)$-Verfahrens, wobei klarerweise mit elliptischen Kurven gerechnet wird. Das kommende Beispiel zeigt eine Faktorisierung in *ARIBAS*, wobei es auch eine Funktion für die *Big Prime Variation* gibt.

Beispiel. Faktorisiert werden sollen die Mersenne Zahl $M_{137} = 2^{137} - 1$ mit der Funktion „ECfactorize“ und die Mersenne Zahl $M_{149} = 2^{149} - 1$ mit der Funktion „ec_factorize“, welche die *Big Prime Variation* anwendet. Die Funktion „ECfactorize“ hat als Argumente die zu faktorisierende Zahl, die Schranke B und die Maximalanzahl an Versuchen. Die Funktion rechnet immer mit einer speziellen elliptischen Kurve. Sollte mit der angegebenen Schranke kein Faktor gefunden werden, so werden die Koeffizienten der elliptischen Kurve neu gewählt

und es wird ein neuer Versuch gestartet. Dies wird so oft wiederholt, bis ein Primfaktor gefunden wurde, oder die angegebene Maximalanzahl an Versuchen erreicht wurde. Unter

```
==> M137:=2**137-1.
-: 17_42245_71863_52049_32932_47799_00506_53242_65471

==> p:=ECfactorize(M137,10000,400).
working ..............................................................
........................................................................
.............................................................
factor found after 214 curves with prime bound 7680
-: 32032_21559_64964_35569

==> q:=M137 div p.
-: 54_39042_18360_02042_90159
```

Abbildung 18: Elliptische Kurven Methode in ARIBAS

Verwendung von 214 verschiedenen elliptischen Kurven und einer Schranke von 7680 konnte die gewünschte Faktorisierung gefunden werden. Beide Faktoren stellen sich als prim heraus. Die Funktion „ec_factorize“ benötigt zusätzlich als Eingabe die zweite Schranke B_2 für die *Big Prime Variation*. Auch hier konnten die beiden Primfaktoren von M_{149} gefunden werden, wie in Abbildung 19 ersichtlich ist. Neben dem Primfaktor erhält man bei der Ausgabe auch den Koeffizient $a = 15796645$ und wiederum eine Schranke. Dieses mal handelt es sich bei besagter Schranke um jene, welche im Zuge der *Big Prime Variation* verwendet wurde. Es soll erwähnt werden, dass beide Funktionen bei der Erstellung der elliptischen Kurven Zufallselemente verwenden, wodurch es bei erneuter Faktorisierung zu unterschiedlichen Ergebnissen kommen kann. Es wäre auch möglich, dass die Funktion keinen Faktor findet und deshalb den Wert 0 ausgibt. ([9], S. 350-351, [10], S. 215-217)

4.3.3.2 Laufzeit

Ähnlich wie beim $(p-1)$-Verfahren hängt die Laufzeit von der Glattheit der entsprechenden Gruppenordnungen ab. Im Zuge der elliptischen Kurven varieren diese jedoch um einen Primfaktor p, wodurch sich die Wahrscheinlichkeit deutlich erhöht auf einen Primfaktor zu stoßen. Die Elliptische Kurven Methode eignet sich also besonders gut für Zahlen mit verhältnismäßig kleinen Primfaktoren und elliptische Kurven mit Ordnungen, welche ebenfalls kleine Primteiler besitzen. Betrachtet man abermals das Resultat von Canfield, Erdös und Pomerance, welches bereits in Abschnitt 4.2.3.2 angeführt wurde, ist man in der Lage die Laufzeit des Verfahrens abzuschätzen. Bezogen auf die Elliptische Kurven Methode bedeutet dies, dass mit einer Wahrscheinlichkeit von $\frac{1}{u^u}$ eine Kurvenordnung $m \in \mathbb{N}$ lediglich Primfaktoren $\leqslant m^{1/u}$ aufweist. Berücksichtigt man ebenfalls die Laufzeit, welche der Double-and-Add-Algorithmus benötigt erhält man eine Gesamtlaufzeit von

$$O(N) = \mathrm{e}^{\sqrt{\ln(p)\ln(\ln(p))(2+o(1))}}$$

```
==> M149:=2**149-1.
-: 71362_38463_52979_94052_91429_84724_74756_81913_73311

==> p:=ec_factorize(M149,(10000,50000),400).

EC factorization, prime bound 10000, bigprime bound 50000
working .:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.
:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:
.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:1:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.
:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:
.:.:.:.:.:.:.:.:
factor found with curve parameter 15796645 and bigprime 25127

-: 86656_26856_62821_83151

==> q:=M149 div p.
-: 82351_09336_69084_67239_86161
```

Abbildung 19: Elliptische Kurven Methode mit *Big Prime Variation* in ARIBAS

für die Elliptische Kurven Methode, wobei unter $o(1)$ eine gegen 0 gehende asymptotische Funktion zu verstehen ist. Es handelt sich um eine subexponentielle Laufzeit, welche sehr stark an die Laufzeit des Quadratischen Siebs erinnert, mit dem Unterschied, dass die Laufzeit hier vom kleinsten Primfaktor p abhängig ist und nicht von N selbst. Generell eignet sich die Elliptische Kurven Methode sehr gut um unter Verwendung einer oberen Schranke von $B = 10^8$, Faktoren mit ungefähr 40 Stellen zu erhalten. Um größere Faktoren zu finden eignet sich das Quadratische Sieb oder dessen Weiterentwicklung in Form des Zahlkörpersiebs besser. Zusätzlich lässt sich das Verfahren auf mehreren Rechnern gleichzeitig ausführen, was einen weiteren Vorteil dieser Methode darstellt.([9], S. 321, 351-352, [10], S. 216)

5 Fazit

Nachdem nun einige traditionelle, aber auch aktuell verwendete Faktorisierungverfahren entsprechend erläutert wurden, soll hier noch ein kurzes Fazit angeführt werden. Das älteste Verfahren ist die Probedivision, welche sich jedoch nur für vergleichweise kleine Zahlen eignet. Dennoch wird sie heutzutage oftmals am Beginn anderer Faktorisierungsalgorithmen verwendet, um kleine Teiler einer zusammengesetzten Zahl herauszudividieren. Das Faktorisierungsverfahren nach Fermat und dessen Optimierung der Lehman-Algorithmus arbeiten nur dann effizient, falls die jeweiligen Faktoren verhältnismäßig nahe beieinander liegen. Das auf dem Fermat-Verfahren aufbauende Quadratische Sieb und das dazugehörige Multipolynomiale Quadratische Sieb gehören aktuell zu den effizientesten Faktorisierungsverfahren. Ein großer Vorteil dieses Algorithmus ist, dass er nur von der Größe der zu faktorisierenden Zahl abhängt und nur sekundär von der Beschaffenheit der Primfaktoren ([9], S. 360). Dennoch ist es ein sehr rechenaufwändiges Verfahren, welches sich erst für Zahlen ab einer gewissen Größe rentiert.

Für Zahlen bis zu einer Größenordnung von 10^{10} gehört es zu den schnellsten und besten Faktorisierungsverfahren. Für größere Zahlen eignet sich eine Weiterentwicklung dieses Verfahrens, nämlich das Zahlkörpersieb. Dieses äußerst komplexe Verfahren ist der derzeit schnellste Algorithmus zur Faktorisierung großer Zahlen. Die Pollard'sche Rho-Methode gehört zu den probabilistischen Algorithmen, da diese Methode nur mit einer gewissen Wahrscheinlichkeit erfolgreich ist. Am besten eignet sich diese Methode zum Auffinden relativ kleiner Primfaktoren. Sollten die Faktoren zu groß werden oder beinahe gleich groß sein, ist diese Methode zu wenig effizient, da sie sehr rechenintensiv wird und nur mit geringer Wahrscheinlichkeit zu einer erfolgreichen Faktorisierung führt, wobei auch hier Ausnahmen möglich sind, da die Methode immerhin auf Zufallselementen aufbaut. Das ebenfalls von Pollard stammende $(p-1)$-Verfahren ist nur dann effizient, falls die Zahl $p-1$ eines Primfaktors p der zusammengesetzten Zahl, aus kleinen Primfaktoren besteht. Eine Verbesserung der $(p-1)$-Methode bedient sich der elliptischen Kurven. Die Elliptische Kurven Methode nach Lenstra funktioniert nach demselben Prinzip wie das $(p-1)$-Verfahren, mit dem Unterschied, dass man durch eine Variation der Koeffizienten die Glattheit der Gruppenordnung beeinflussen kann. Dadurch ist das Verfahren deutlich effizienter als das $(p-1)$-Verfahren. Die Effizienz der Elliptischen Kurven Methode hängt von der Größe des kleinsten Primfaktors p und der Glattheit der Gruppenordnung der verwendeten elliptischen Kurve ab. Um bis zu 40-stellige Faktoren zu finden eignet sich das Verfahren in der Regel sehr gut.

Nichtsdestotrotz besitzen bis auf das Quadratische Sieb und die Elliptische Kurven Methode, welche eine subexponentielle Laufzeit aufweisen, alle übrigen Verfahren eine exponentielle Laufzeit. Dies bedeutet, dass diese Verfahren ab einer gewissen Größe der zu faktorisierenden Zahl nur noch in bestimmten Fällen brauchbar sind. Wählt man die zu faktorisierende Zahl groß genug und berücksichtigt bestimmte Eigenschaften, so ist kein Verfahren effizient genug, um diese Zahl vollständig zu faktorisieren. Je nachdem, ob man im vorhinein weiß welche speziellen Eigenschaften die zu faktorisierende Zahl oder deren Faktoren aufweisen, eignen sich manche Verfahren besser als andere. Bezogen auf den RSA-Algorithmus eignet sich am besten das Quadratische Sieb und dessen Optimierungen, da die beiden Faktoren der zusammengesetzten Zahl ungefähr die gleiche Größenordnung haben sollen, vorausgesetzt die Faktoren haben keine der speziellen Eigenschaften der oben genannten Algorithmen. Die heutzutage verwendeteten Faktoren haben jedoch eine Größenordnung, bei welche sich nur noch das Zahlkörpersieb verwenden lässt, um den Hauch einer Chance auf eine erfolgreiche Faktorisierung zu haben. Alles in allem gibt es bis heute noch keinen Algorithmus, welcher effizient genug wäre, um die im RSA-Algorithmus verwendeten Zahlen in angemessener Zeit zu faktorisieren.

6 Ausblick in die Zukunft

Sollte sich nicht herausstellen, dass es sich bei der Multiplikation natürlicher Zahlen doch nicht um eine Einwegfunktion handelt, oder kein effizienter Faktorisierungsalgorithmus gefunden wird, kann nur eine höhere Rechenleistung von Computern zum Erfolg bei der Lösung des Faktorisierungsproblems führen. Seit im Jahre 1985 das bis dato erste theoretische Modell eines Quantencomputers veröffentlicht wurde, wird intensiv am Bau dieser Rechner geforscht und gearbeitet. Ein Quantencomputer rechnet, wie der Name des Computers bereits verrät mit Quantenbits, welche es ermöglichen, sogar anspruchsvolle mathematischen Probleme effektiv zu lösen. Im Jahre 1994 konnte Peter Shor einen Faktorisierungsalgorithmus speziell für Quantencomputer entwickeln. ([1], S. 296, 299) Der nach ihm benannte Shor-Algorithmus wäre dazu in der Lage mit polynomialer Laufzeit auch sehr große Zahlen zu faktorisieren. Im Zuge der bereits erwähnten RSA-Challenge benötigten 600 Rechner mehrere Monate um eine zusammengesetzte Zahl mit 129 Stellen zu faktorisieren. Im Vergleich dazu würde ein Quantencomputer mit dem Shor-Algorithmus eine um ein Vielfaches größere Zahl in einem winzigen Bruchteil dieser Zeit faktorisieren. Da es bis heute jedoch keine Quantencomputer gibt, welche auch nur annäherend weit genug entwickelt sind, um Probleme wie die Faktorisierung natürlicher Zahlen zu lösen, handelt es sich hierbei noch um eine Wunschvorstellung. Darüber hinaus gibt es noch zahlreiche Hürden, welche beim Bau eines Quantencomputers überwunden werden müssen. Solange man diese Probleme nicht meistert, gilt das RSA-Verfahren als sehr sicher. Auch wenn man erst am Beginn der Entwicklung eines vollkommen funktionstüchtigen Quantencomputers ist, hätten diese Rechner jedoch enorme Auswirkungen auf die Sicherheit unserer Daten. Denn sollten diese Quantencomputer jemals zum Einsatz kommen, so wären sämtliche Informationen nicht mehr sicher und könnten stets abgefangen und abgehört werden. Dies hätte nicht nur verheerende Auswirkungen auf unsere Privatsphäre, sondern auch auf die nationale Sicherheit selbst. ([12], S. 384-387, 398-400) Dennoch gibt es auch andere Verfahren, welche auf der Grundlage der Quantentheorie beruhen, nämlich die Quantenkryptografie. Die im Jahre 1984 entwickelte Quantenkryptografie nutzt für einen Austausch an Informationen polarisiertes Licht. Diese Art der Verschlüsselung wäre im Gegensatz zu allen bisher verwendeten Verfahren nicht zu brechen, wie man auch beweisen kann. Erste Prototypen funktionieren bereits über eine geringe Reichweite von mehreren Kilometern. ([1], S.293, [12], S. 400, 419) Für eine detaillierte Beschreibung der Quantencomputer, des Shor-Algorithmus und der Quantenkryptografie siehe ([1], 293-303,[12], S. 383-421).

Nach momentanem Stand geht man davon aus, dass der Grund warum es keine effiziente und einfache Methode zur Faktorisierung natürlicher Zahlen gibt vermutlich in der Natur der Mathematik selbst liegt. Sollten eines Tages Quantencomputer die dafür brauchbare Rechenleistung besitzen, um dieses Problem zu lösen, benötigt man neue Verschlüsselungssysteme, da der RSA-Algorithmus dadurch unbrauchbar werden würde. Bis jetzt ist allerdings die

Quantenkryptografie weiter entwickelt und fortschrittlicher als der Bau von Quantencomputern. Sobald die Quantenkryptografie alltagstauglich geworden ist, hat man ein beweisbar sicheres Verschlüsselungsverfahren gefunden. Auch wenn dies im Laufe der Geschichte bereits öfter behauptet wurde, ist man sich hier vollkommen sicher. Ansonsten hätte dies zur Folge, dass die für die Physik überaus wichtige Quantentheorie falsch wäre, womit sich das bisherige Bild des Universums auf den Kopf stellen würde. Aus Sicherheitsgründen bleibt zu hoffen, dass die Quantenkryptografie zuerst praktikabel wird. Es sollte jedoch nicht vergessen werden, dass eine absolut sichere Verschlüsselungstechnik ebenso große Schwierigkeiten mit sich bringen kann wie eine unsichere. ([12], S. 419-421)

Literatur

[1] Beutelspacher Albrecht, Neuman Heike B. und Schwarzpaul Thomas. *Kryptografie in Theorie und Praxis.* [Wiesbaden]: GWV Fachverlage Gmbh, 2010.

[2] Werner Annette. *Elliptische Kurven in der Kryptographie.* [Berlin Heidelberg]: Springer-Verlag, 2002.

[3] Baxa Christoph. „Zahlentheorie Skriptum zur Vorlesung von Christoph Baxa". Skriptum. Universität Wien, 2017.

[4] Ballwein Dominik. „Die Mathematik der Kryptologie von der Antike bis heute". Bachelorarbeit. Universität Wien, 2020.

[5] Lang Hans Werner. *Kryptografie für dummies.* [Weinheim]: WILEY-VCH Verlag GmbH & Co. KGaA, 2018.

[6] Lenstra Hendrik W. „Factoring integers with elliptic curves". In: *Annals of Mathematics, Vol.126 (3), p.649-673* (1987).

[7] Schichl Hermann und Steinbauer Roland. *Einführung in das mathematische Arbeiten (2., überarbeitete Auflage).* [Berlin Heidelberg]: Springer-Verlag, 2012.

[8] Buchmann Johannes. *Einführung in die Kryptographie (4. erweiterte Auflage).* [Berlin Heidelberg]: Springer-Verlag, 2008.

[9] Wohlgemut (Hrsg.) Martin. *Mathematisch für fortgeschrittene Anfänger.* [Heidelberg]: Spektrum Akademischer Verlag, 2010.

[10] Forster Otto. *Algorithmische Zahlentheorie (2., überarbeitete und erweiterte Auflage).* [Wiesbaden]: Springer Fachmedien, 2015.

[11] Crandall Richard und Pomerance Carl. *Prime Numbers, A Computational Perspective(Second Edition).* [New York]: Springer Science+Business Media, Inc, 2005.

[12] Singh Simon. *Geheime Botschaften-Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet.* [München]: dtv Verlagsgesellschaft mbH & Co.KG, 2019.

[13] Müller-Stach Stefan und Pionkowski Jens. *Elementare und algebraische Zahlentheorie-Ein moderner Zugang zu klassischen Themen (2., erweiterte Auflage)).* [Wiesbaden]: Springer Fachmedien, 2011.

[14] Trappe Wade und Washington Lawrence C. *Introduction to Cryptography with Coding Theory(Second Edition).* [Upper Saddle River,NJ]: Pearson Prentice Hall, 2006.

[15] Ruppert Wolfgang. „Kryptographie I - Kryptographie für Lehramt (WS 2022/2023)". Skriptum. Department Mathemathik, 2023.

[16] Seiler Wolfgang K. „Zahlentheorie". Skriptum. Universität Mannheim, 2018.