Primzahltests

Gábor SAS

2002 - 2008

Inhaltsverzeichnis

Kapitel 1

Einleitung und Geschichte[1]

1.1 Der Primzahlbegriff

Definition (Primzahl)**.** *Eine natürliche Zahl $p > 1$ ist eine Primzahl, falls sie nur durch sich und 1 teilbar ist.*

Soweit es bekannt ist, beschäftigten sich die Mathematiker der pythagoräischen Schule (ab 500 bis 300 v. Chr.) als erste mit den Primzahlen. Sie untersuchten perfekte und befreundete Zahlen und beschäftigten sich folglich mit Primzahlen und zusammengesetzten Zahlen. Sie machten zwar zahlreiche bedeutende Entdeckungen, es gelang ihnen allerdings nicht, ihre Theorien zu beweisen. Um 300 v. Chr. veröffentlichte Euklid die Bücher der „Elemente“, die viele wichtige Erkenntnisse der Primzahlforschung mit korrekt geführten Beweisen beinhalteten. (Diese Schriften sind z.B. in [3] nachzulesen.) Erstmals wurde bewiesen, daß es unendlich viele Primzahlen gibt, weiters bewies er den Fundamentalsatz der Zahlentheorie - die (bis auf die Reihenfolge) eindeutige Zerlegung von Zahlen in Primfaktoren.

Etwa um diese Zeit (um die Lebzeit von Konfuzius) stellten die Chinesen die Vermutung auf, daß eine Zahl n dann und nur dann eine Primzahl ist, falls diese $2^n - 2$ teilt - was sich aber als falsch herausstellte. Es wird aber oft bezweifelt, daß die Chinesen tatsächlich diese Vermutung aufstellten, es scheint sogar so zu sein, daß sie das Konzept der Primzahlen nie formuliert hätten. Trotzdem ging diese Vermutung als Chinesische Hypothese (Chinese Hypothesis) in die Geschichte ein.

Die Zeit der großen griechischen Mathematiker endete mit Eratosthenes um 200 v. Chr., der einen Algorithmus zur Erstellung einer Tabelle mit allen Prim-

[1] Abseits im Text angeführten Quellen werden [1] und [2] herangezogen.

zahlen bis zu einer gegebenen Zahl entdeckte. Dieser wird heute nach ihm „Sieb des Eratosthenes“ genannt. Dieser Algorithmus ist der erste, der systematisch an die Primzahlen abseits einer Faktorisierung angeht.

1.2 Sieb von Eratosthenes

Das Sieb von Eratosthenes wurde - wie schon erwähnt - nach Eratosthenes von Kyrene benannt und stellt eine Vorgangsweise dar, wie man alle Primzahlen bis zu einer fest vorgegebenen Zahl findet, diese Methode wird z.B. in [4] behandelt.

Ausgangspunkt des Algorithmus ist eine Liste aller natürlichen Zahlen von 2 bis zu der gewünschten Zahl. Man markiert die erste Zahl, konkret die 2, z.B. durch einkreisen als Primzahl und streicht alle Vielfachen von dieser Zahl durch. Die erste Zahl in dieser bearbeiten Liste, welche weder durchgestrichen noch als prim markiert ist, ist eine Primzahl und wird ebenfalls als solche gekennzeichnet. Alle Vielfachen von dieser Zahl werden danach ebenfalls durchgestrichen, da diese Zahlen durch die soeben markierte teilbar sind. Nun ist wieder die erste nicht durchgestrichene oder als prim markierte Zahl die nächste Primzahl, da wenn eine andere Zahl diese teilen würde, wäre diese Zahl nicht nicht durchgestrichen. Deren Vielfache werden nun wieder durchgestrichen und so weiter und so fort.

Daß dadurch alle Primzahlen bis zu der gewünschten Zahl erhalten werden, ist offensichtlich, da alle als prim gekennzeichnete Zahlen keine Teiler (abgesehen von 1 und sich selbst) besitzen, die durchgestrichenen Zahlen haben alle einen Teiler.

Die Methode läßt sich ein wenig beschleunigen. Man muß mit dem Durchstreichen erst beim Quadrat der gerade ermittelten Primzahl beginnen, da für kleinere Zahlen schon zumindest einen anderen Teiler geben muß, falls die Zahl zusammengesetzt ist. (Für jede zusammengesetzte Zahl n existiert ein echter Teiler t mit $t \leq \sqrt{n}$.) Aus dem gleichen Grund kann man auch den Algorithmus schon bei der Wurzel der zu erreichenden Zahl beenden, alle danach nicht durchgestrichenen Zahlen sind Primzahlen.

Diese Methode ist zwar geeignet, um eine Tabelle aller Primzahlen bis zu einer kleinen Schranke aufzustellen, jedoch ist das Sieb von Eratosthenes als Primzahltest nicht geeignet. Um die Primalität einer Zahl n mit dem Sieb von Eratosthenes festzustellen, muß die Primalität aller Zahlen kleiner als n festgestellt werden.

1.3 Feststellung der Primalität durch Division

Nach der Zeit von Eratosthenes wurde lange Zeit keine mathematische Forschung betrieben. Mehr sogar, fast sämtliche wissenschaftliche Errungenschaften der Griechen gerieten während der Römerzeit und des Mittelalters in Vergessenheit, so auch die Erkenntnisse über Primzahlen. Erst während der Renaissance begann man, sich wieder mit der Mathematik und damit auch mit den Primzahlen zu beschäftigen. Dabei mussten viele Erkenntnisse aus der Zeit der Griechen erst wieder neu entdeckt werden. Die ersten Erforschungen der Neuzeit behandelten Zahlen der Form $2^n - 1$. Diese Zahlen wurden nach dem schweizer Mönch Mersenne Mersennezahlen genannt, in Zeichen M_n. Es ist leicht zu zeigen, daß n prim sein muß, damit M_n prim sein kann, daß nicht alle Zahlen dieser Form mit einer Primzahl n wieder eine Primzahl ist, wurde 1536 entdeckt ($2^{11} - 1 = 2047 = 23 \cdot 89$). 1588 bewies Cataldi, dass $2^{19} - 1$ eine Primzahl ist, welche ca. 200 Jahre lang die größte bekannte Primzahl blieb. Er war auch der erste (bekannte), der eine Primzahlsammlungen erstellte. Er hat eine Tafel mit den Primzahlen bis 750 aufgestellt, welche auch reichte, die Zahl M_{19} als prim zu entlarven. Von einigen wird es jedoch bezweifelt und vermutet, daß er den Beweis für diese Zahl nicht korrekt geführt hat, da er auch noch ein paar größere nichtprime Mersennezahlen als prim angab. Vor ihm hat es auch ein paar „größte bekannte Primzahlen gegeben“, jedoch fehlt bei denen jeglicher Beweis dafür, es wird allgemein als „gutes“ Raten interpretiert. Auch heutzutage sind die größten bekannten Primzahlen durchgehend Mersennezahlen, da es für diese schnelle Tests auf Primalität gibt (siehe Kapitel 5).

Die erste wirklich bedeutende Entdeckung seit Eratosthenes gelang Fermat zu Beginn des 17. Jahrhunderts. Er erfand eine neue Technik zur Zerlegung von „größeren“ Zahlen. Die wohl bedeutendste Entdeckung in Bezug auf Primzahlen war der nach ihm benannte Kleine Fermatsche Satz: Falls p eine Primzahl ist gilt für jede ganze Zahl a, daß $a^p \equiv a \mod p$ ist. Diesen Satz verwendete er für den Beweis der einen Hälfte der chinesischen Hypothese, die andere Hälfte ist ja bekanntlich falsch, da z.B. $2^{341} - 2$ durch 341 teilbar ist, jedoch gilt $341 = 31 \cdot 11$. Zusammengesetzte Zahlen, für die die chinesische Hypothese gilt, werden auch Poulet-Zahlen genannt. Fermats Satz ist die Grundidee für viele andere Erkenntnisse in der Zahlentheorie und viele von Computern genützte Verfahren zum Prüfen von Primzahlen beruhen auf diesem Satz. Fermat schrieb in einem Brief an Mersenne die Vermutung, daß die Zahlen $2^n + 1$ für alle n prim sind, falls n ein Potenz von 2 ist. So heißen die Zahlen $F_n = 2^{2^n} + 1$ Fermatsche Zahlen, jedoch sind diese Zahlen nur für $n = 0, 1, 2, 3, 4$ prim, sonst hat man bis heute keine weiteren prime Fermatzahlen gefunden. (Oft werden auch die Zahlen $2^n + 1$ als Fermatzahlen genannt.) Bis heute ist von 235 Fermatzahlen bekannt, daß diese nicht prim sind, jedoch nicht die ersten 235 beginnend mit $n = 5$. Die Fermatzahlen 5 - 11 sind vollständig faktorisiert, F_{14} ist die kleinste Fermatzahl,

von der nur bekannt ist, daß diese zusammengesetzt ist, F_{33} ist die erste, deren Charakter noch unerforscht ist. Der aktueller Stand der Erforschung dieser Zahlen ist unter [5] einsehbar. Fermat hat auch als erster erkannt, daß eine nichtprime Mersennezahl M_n Teiler der Form $2kn + 1$ haben muß, womit er auch etliche Mersennezahlen als nicht prim erkannt hat.

Mehr als 100 Jahre nach der Vermutung, Fermatzahlen seien Primzahlen, bewies Leonard Euler, daß 641 F_5 teilt. Euler löste auch mit der Zahl M_{31} den Rekord für die größte bekannte Primzahl ab und hielt diesen einen Jahrhundert lang. Für den Beweis benützte er prinzipiell auch nur einfaches Durchprobieren durch eine verkleinerte Liste (ähnlich wie Fermat) von möglichen Teilern. Euler leistete weiters grundlegende Arbeit in der Zahlentheorie, er kann als Begründer der analytischen Zahlentheorie betrachtet werden und verfeinerte den kleinen Satz von Fermat mit der Einführung der Eulerschen φ-Funktion.

Im Jahre 1867 stellte Landry nocheinmal einen neuen Rekord für die größte bekannte Primzahl mit $(2^{59} - 1)/179951$ auf, er verwendete die gleiche Methode wie Euler. Es sollte das letzte Mal sein, daß die größte bekannte Primzahl durch einfache Division bestimmt wurde, ab dann wurden die ersten ernstzunehmenden Tests auf Primalität entwickelt.

1.4 Erkennung der Primalität ohne Teiler zu kennen

Bis ins späte 19. Jahrhundert wurden zusammengesetzte Zahlen entlarvt, indem man einen echten Teiler fand bzw. unter den möglichen Zahlen alle als Teiler ausschließen konnte. 1876 entwickelte Lucas einen Test für die Mersennezahlen M_n, falls $n \equiv 3 \mod 4$ ist. Diesen verwendete er auch zum Beweis, daß M_{127} prim ist. Später gab Lucas auch einen Test an, welcher für $n \equiv 1 \mod 4$ funktionierte. Auch Lucas könnte es schon gewußt haben, aber erst Lehmer (1930) bewies, daß dieser Test auch für $n \equiv 3 \mod 4$ funktioniert - deswegen heißt dieser Test Lucas-Lehmer Test. Damit wurde das erste Mal ein Test entwickelt, welcher bei zusammengesetzten Zahlen nur die Antwort nichtprim hat, jedoch nichts über den Teiler der Zahl verrät. Lucas hat auch noch für andere Zahlenformate Tests entwickelt, die bei weitem nicht die Berühmtheit des Tests für Mersenneprimzahlen erreichten. So zum Beispiel gab er einen Test für $2 \cdot 3^n - 1$, falls $4 \nmid n$, an.

Mitte des 20. Jahrhunderts begann das Zeitalter der Computer. Dieses brachte zwar kaum neue Erkenntnisse auf dem Gebiet der Zahlentheorie, jedoch einen Primzahlrekord nach dem anderen. Der erste, der den Computer zum Finden von Primzahlen nutzte, war der Amerikaner Robinson. Die größte Primzahl, die

er fand, war M_{2281} im Jahre 1952. In der Folgezeit wurde alle paar Jahre ein neuer Rekord aufgestellt. Der neueste Rekord $M_{43112609}$ wurde am 23. August 2008 im Rahmen von GIMPS (Great Internet Mersenne Prime Search, siehe [6]) gefunden. GIMPS ist eine Internet-Organisation, bei der jedes Mitglied einen bestimmten Zahlenraum zugewiesen bekommt und in diesem nach Mersenneschen Primzahlen sucht. Weiterführende Informationen zu GIMPS siehe Kapitel 5.6. Seit 1996 hält dieses Projekt den Rekord für die größte bekannte Primzahl, insgesamt wurden in diesen 12 Jahren 12 Mersennesche Primzahlen gefunden.

1.5 Weitere Primzahltests

Abseits der Suche nach immer größeren Primzahlen beschäftigten sich einige Mathematiker mit Primzahltests für nicht Mersennesche Primzahlen, so z.B. Caldwell, der Zahlen der Form $n! \pm 1$ auf ihre Primalität untersuchte - die größte bekannte Primzahl dieser Form ist $34790! - 1$. Auch für andere Arten von Zahlen existieren Listen für die größten bekannten Primzahlen, so z.B. ist der größte bekannte Primzahlzwilling $2003663613 \cdot 2^{195000} \pm 1$. Für Zahlen n mit gewissen Eigenschaften (z.B. $n-1$ läßt sich leicht in ihre Primfaktoren zerlegen bzw. die Zerlegung von $n-1$ ist bekannt) existieren schnelle Methoden, um die Primalität einer Zahl zu entscheiden. Diese Tests werden in den Kapiteln 3 und 5 behandelt, weitere Test sind unter anderem in [7] zu finden.

In den siebziger Jahren begann man Algorithmen für beliebige Zahlen zu konstruieren und deren Laufzeit anzugeben. Miller, Solovay, Strassen und Rabin verwendeten elementare Ergebnisse der Zahlentheorie, um „schnell“ funktionierende Tests zu erzeugen. (Siehe Kapitel 4.) Diese waren entweder auf unbewiesene Hypothesen aufgebaut beziehungsweise nahmen das Restrisiko eines Fehlers bezüglich der Feststellung der Primalität in Kauf, jedoch falls eine Zahl als zusammengesetzt erkannt worden ist, konnte man den Beweis ohne den Test neu laufen lassen zu müssen schnell nachvollziehen. Diese Tests funktionieren für allgemeine Zahlen bis heute am schnellsten und die Fehlerwahrscheinlichkeit kann beliebig minimiert werden. Deshalb werden diese Tests heute in der Praxis noch immer sehr häufig verwendet.

Adleman, Pomerance und Rumely in [10] und Cohen und Lenstra Jr. in [11] gaben ohne auf Vermutungen zurückgreifen zu müssen Algorithmen in fast polynomieller Laufzeit an. Diese Tests hatten den Nachteil, daß die Primalität der Zahl nicht anhand eines kurzen Beweises nachvollzogen werden konnte. Die Konstruktion eines Primzahltests aufbauend auf elliptische Kurven konnte diesen Nachteil ausmerzen (siehe Kapitel 6), aber auch dieser Test funktionierte nicht allgemein in polynomieller Laufzeit.

Die wohl wichtigste Erkenntnis in den letzten Jahren ist die Lösung des Problems, ob die Feststellung der Primalität einer beliebigen Zahl in polynomialer Laufzeit möglich ist. Nach Vermutungen (z.B. aufbauend auf die Riemannsche Hypothese) und Konstruktion verschiedenen Tests mit nahezu polynomialer Laufzeit wurde durch die Vorstellung eines konkreten Beweises von Agrawal, Kayal und Saxena (AKS-Test) diese Problemstellung gelöst, siehe Kapitel 7.

Kapitel 2

Primzahltests

Diese Arbeit soll auf das Erkennen von Primzahlen eingehen. Bei dieser Problemstellung ist es vollkommen irrelevant, welche Zahl (abseits von 1 und der Zahl selbst) eine gegebene Zahl teilt, die alleinige Frage ist, ob es einen echten Teiler gibt. Dementsprechend wird auf Methoden zur Auffindung von Faktoren oder gar zur kompletten Faktorisierung einer Zahl nicht eingegangen.

Es kann natürlich bei einigen Tests passieren, daß ein Primteiler durch einen Primzahltest gefunden wird, dies ist jedoch nur ein „Zufall“. So ist z.B. in vielen Tests eine Zahl $a < n$ mit der Eigenschaft $\mathrm{ggT}(a, n) = 1$ zufällig zu wählen. Wenn der größte gemeinsame Teiler nicht 1 ist, ist ein Teiler gefunden und der Test bricht mit der Meldung, daß die Zahl n nicht prim ist, ab. Die Tests zielen in diesen Fällen aber prinzipiell nicht darauf, daß man auf dieser Art und Weise einen Teiler findet.

2.1 Arten von Primzahltests

Jede Aussage über Primzahlen kann zu einem Test ausgebaut werden, wobei nicht alle Tests für alle Zahlen funktionieren, manche Tests können nur ausschließen, daß eine Zahl zusammengesetzt ist oder umgekehrt. Oft wird ein Zufallselement in den Tests eingebaut, wodurch entweder eine kleine Unsicherheit in der Aussage des Tests entsteht oder es nicht sicher ist, daß ein Test terminiert.

Folgende Tests werden unterschieden:

- „Echte“ Primzahltests: Diese erkennen immer, ob eine Zahl prim oder zusammengesetzt ist. Diese Tests beruhen auf Aussagen der Form „n ist genau dann ein Primzahl, falls folgende Bedingungen gelten“.

- Tests auf Zusammengesetztheit: Liefert ein Test dieser Art als Ergebnis „zusammengesetzt“, dann ist die zu prüfende Zahl mit Sicherheit zusammengesetzt, jedoch kann man nicht davon ausgehen, daß wenn man nicht „zusammengesetzt“ bekommt, die Zahl auch tatsächlich prim ist. Diese Tests beruhen auf Aussagen der Form „Gelten bestimmte Bedingungen, so ist n zusammengesetzt“ bzw. „Ist n prim, so gilt das Folgende“.
- Test auf Primalität: Diese Tests erkennen nur mit Sicherheit, daß eine Zahl prim ist. Diese Tests beruhen auf Aussagen der Form „Gelten bestimmte Bedingungen, so ist n prim“ bzw. „Ist n zusammengesetzt, so gilt das Folgende“.
- Probabilistische Tests: diese Tests bauen auf Tests auf Zusammengesetztheit bzw. auf Test auf Primalität auf. Grundidee hinter diesem Tests ist, daß die Tests als Grundlage gut funktionieren und nur wenige Zahlen werden nicht als prim bzw. als zusammengesetzt erkannt. Sind diese Tests auch noch verschieden parametrisierbar und erkennen diese für verschiedene Parameter verschiedene Zahlen nicht, so kann man durch mehrmaliges Wiederholen mit zufällig gewählten Parametern die Wahrscheinlichkeit, daß man eine Zahl nicht richtig erkennt, minimieren.

2.2 Augenzeugen und Zertifikate

Oft sind Primzahltests sehr aufwendig, da man in einem Zahlenraum die richtigen Parameter finden muß. Hat man diese Parameter aber gefunden, ist der Beweis für die Zusammengesetztheit sehr einfach. So ist z.B. die Angabe eines Teilers im allgemeinen Fall zeitaufwendig, hat man aber einen gefunden, so ist es sehr schnell nachvollziehbar, ob diese Zahl tatsächlich ein Teiler ist. Kurze Operationen - wie z.B. die Angabe eines Teilers, die eine Zahl eindeutig und schnell nachvollziehbar als zusammengesetzt erkennen läßt, nennt man Augenzeugen.

Auch für Primzahlen existieren Augenzeugen, sprich sehr kurze Rechnungen, die die Primalität einer Zahl bestätigen. In diesem Fall spricht man von einem Zertifikat. Der bekannteste Test mit gleichzeitiger Erzeugung von einem Zertifikat ist der Primzahltest beruhend auf elliptische Kurven (siehe Kapitel 6).

2.3 Laufzeit

Ein wichtiger Faktor bei Primzahltests ist deren Laufzeit, da davon die praktische Anwendbarkeit abhängt. Was dabei als „schnell“, „brauchbar“ etc. gilt, ist relativ, hängt auch sehr stark damit zusammen, wieviel Zeitaufwand ein einzelner

Schritt benötigt, wie schnell ein Rechner ist. Da die Schnelligkeit der Rechner stets zunimmt, wird die Frage der Schnelligkeit eines Algorithmus auf die Anzahl der Operationen heruntergebrochen. Dabei soll in Abhängigkeit von der Länge der Zahl (in der Binärdarstellung) angegeben werden, wie sich der Zeitaufwand entwickelt. Da die exakte Berechnung der Anzahl der durchzuführenden Operationen oft schwierig bis praktisch unmöglich ist, wird das asymptotische Verhalten der Laufzeit in Abhängigkeit der Länge der zu prüfenden Zahl untersucht. Die Angabe der Laufzeit wird in der Landauschen-O-Notation angegeben, die Definition und die für diese Arbeit wichtige Eigenschaften wurden im Anhang A zusammengefasst.

Da der Zufall bei einigen Tests eine nicht geringe Rolle spielt, ist auch die Angabe des asymptotischen Verhaltens für alle möglichen zu testenden Zahlen nur schwer möglich, in einigen Fällen kann nicht einmal gesagt werden, ob ein Test für alle Zahlen überhaupt terminiert. Viele Tests funktionieren in der Praxis hingegen wesentlich schneller und oft wird gar vermutet, daß die Laufzeit kürzer sein muß, jedoch ist kein exakter Beweis bekannt.

Bei Laufzeitbetrachtungen will man die Anzahl der nötigen Schritte in Abhängigkeit von der binären Länge der zu prüfenden Zahl p angeben, und die Anzahl der Zeichen von p in der Binärdarstellung weicht von $\log p$ höchstens um 1 ab[1].

Des Weiteren gibt es viele Möglichkeiten, Primzahlen zu identifizieren, die wegen der langen Rechenzeit als Primzahltest in der Praxis nicht in Frage kommen. Hier sei darauf hingewiesen, daß in diesen Fällen nie ausgeschlossen werden kann, daß diese Tests irgendwann interessant werden können, wenn man Methoden findet, die Berechnungen auf eine „brauchbare“ Zeit zu beschränken. Solche Tests finden in dieser Arbeit keine Berücksichtigung, aus historischen gründen wird jedoch auf den Test nach Wilson eingegangen.

2.4 Wilsontest

Satz 1 (Wilson)**.** $(p-1)! \equiv -1 \mod p$ *gilt dann und nur dann, wenn* p *eine Primzahl ist.*

Beweis: Der Beweis wird in 2 Schritten geführt:

- p prim $\Rightarrow (p-1)! \equiv -1 \mod p$

  Links steht das Produkt über alle Elemente der multiplikativen Gruppe des Körpers $\mathbb{Z}_p$. Da mit jedem $a \in \mathbb{Z}_p^*$ auch $a^{-1} \in \mathbb{Z}_p^*$ ist, lassen sich die Faktoren

[1] Wird in dieser Arbeit log angegeben, so ist stets der Logarithmus zum Basis 2 gemeint. (Verwendet wird in dieser Arbeit noch der natürliche Logarithmus, in Zeichen ln.)

zu Paaren $a \cdot a^{-1} = 1$ zusammenfassen mit Ausnahme der Elemente, welche zu sich selbst invers sind, d.h. $a^2 = 1$ erfüllen. Da die Gleichung $x^2 = 1$ im Körper $\mathbb{Z}_p$ genau die Lösungen ± 1 hat, folgt daraus die Behauptung.

- $(p-1)! \equiv -1 \mod p \Rightarrow p$ prim

  Ist $p = n \cdot m$, wobei n und m echte Teiler von p mit $m \neq n$ sind, dann sind beide Faktoren in $(p-1)!$ vorhanden und damit ist $(p-1)!$ durch $m \cdot n$ teilbar, also ist $(p-1)! \equiv 0 \mod p$. Sei nun $p = c^2$, so ist

  $$(p-1)! = (p-1) \cdot (p-2) \cdot \ldots \cdot (2c) \cdot \ldots \cdot c \cdot \ldots \cdot 2 \cdot 1.$$

  Ist nun $(p-1) > 2c = 2\sqrt{p}$, dann enthält $(p-1)!$ mindestens 2mal den Faktor c, und damit gilt auch $(p-1)! \equiv 0 \mod p$. Die Ungleichung $(p-1) > 2c = 2\sqrt{p}$ ist für alle $p \geq 6$ erfüllt, da $6-1 > 2\sqrt{6} \sim 4,9$ und die linke Seite wächst schneller als die rechte. Damit ist die einzige noch zu behandelnde Quadratzahl die 4, für p=4 gilt jedoch $(p-1)! \equiv 3! \equiv 6 \equiv 2 \mod 4$.

□

Bemerkung. *Oft wird auch nur die erste Hälfte des Satzes als Satz von Wilson bezeichnet. In der Erstveröffentlichung in [12] lautete der Satz wie folgt: Für eine Primzahl p gilt, daß*

$$\frac{1 \cdot 2 \cdot \ldots \cdot (p-1) + 1}{p}$$

eine ganze Zahl ist. Für weitere historische Bemerkungen und eine ausführliche Behandlung des Satzes siehe [13].

Den Satz von Wilson kann man eins zu eins in einem Primzahltest umwandeln, indem man für eine gegeben Zahl p das Produkt $(p-1)! \mod p$ berechnet. Ist das Ergebnis -1, dann ist p eine Primzahl, sonst nicht.

Leider ist dieser Test sehr ineffizient, da für die Berechnung der Fakultät keine schnellen Methoden bekannt sind, man müsste $p-2$ Multiplikationen durchführen, um eine Zahl als Primzahl zu entlarven. Dadurch wird dieser Test praktisch unanwendbar.

Kapitel 3

Fermattests

Pierre de Fermat erwähnte in einem mit 18. Oktober 1640 datierten Brief an Frénicle de Bessy erstmals den heute unter dem Namen Kleiner Fermatscher Satz bekannten Zusammenhang. (Siehe [14].) Dieser Satz samt weiteren Verfeinerungen stellen den Hintergrund für viele Test auf Primalität. Die in diesem Kapitel angeführten Test stammen aus [7].

3.1 Kleiner Fermatscher Satz

Satz 2 (Kleiner Fermatscher Satz)**.** *Sei p eine Primzahl und a eine beliebige natürliche Zahl mit $ggT(a,p) = 1$, so gilt*

$$a^{p-1} \equiv 1 \mod p. \tag{3.1}$$

Beweis: Multipliziert man (3.1) mit a, so erhält man

$$a^p \equiv a \mod p. \tag{3.2}$$

Da $\mathrm{ggT}(a,p) = 1$ gilt, ist (3.2) identisch zu (3.1). (Die Beschränkung $\mathrm{ggT}(a,p) = 1$ ist für (3.2) hingegen nicht mehr notwendig, denn für $a \equiv 0 \mod p$ ist die Gleichung trivialerweise erfüllt.) Die Aussage (3.2) bedeutet, daß p den Ausdruck $a^p - a$ teilt. Für $a = 0$ ist dies trivialerweise erfüllt.

$$(a+1)^p = \sum_{i=0}^{p} \binom{p}{i} a^i, \tag{3.3}$$

diesen Ausdruck modulo p betrachtet fallen alle Terme mit den Binomialkoeffizienten $\binom{p}{i} = \frac{p!}{i!\cdot(p-i)!}$ mit $0 < i < p$ weg, da der Zähler den Faktor p enthält, der Nenner aber nicht, somit gilt

$$(a+1)^p \equiv a^p + 1 \mod p. \tag{3.4}$$

Gilt nun $p \mid a^p - a$ für einen Wert von a, so gilt

$$(a+1)^p - (a+1) \equiv a^p + 1 - (a+1) \equiv a^p - a \mod p, \tag{3.5}$$

woraus folgt, daß p auch $(a+1)^p - (a+1)$ teilt. Mit der vollständigen Induktion folgt daraus der Kleine Fermatsche Satz. □

Bemerkung. *Es existieren auch weitere Beweise des Satzes, erwähnenswert ist der gruppentheoretische Beweis. Der Satz von Euler-Fermat, eine Verfeinerung des Kleinen Fermatschen Satzes, wird im Kapitel 3.3 anhand eines gruppentheoretischen Ansatzes bewiesen. Ein Sonderfall dieses Beweises ist der gruppentheoretische Beweis des Kleinen Fermatschen Satzes.*

Diese Identität kann leicht für einen Test auf Zusammengesetztheit ausgenützt werden:

Sei ein $n \in \mathbb{N}$ gegeben. Man nehme ein a mit $\mathrm{ggT}(a, n) = 1$. Ist nun

$$a^{n-1} \not\equiv 1 \mod n,$$

dann ist n zusammengesetzt.

Die Umkehrung des Kleinen Fermatschen Satzes gilt nicht, wie es im Folgenden durch ein Gegenbeispiel gezeigt wird. Dies führt dazu, daß der Kleine Fermatscher Satz nicht (ohne weiteres) geeignet ist, die Primalität einer Zahl nachzuweisen.

Beispiel. *Sei $n = 341 = 11 \cdot 31$ und $a = 2$. n ist klarerweise nicht prim, jedoch gilt:*

$$2^{340} \equiv 1 \mod 341$$

Definition (pseudoprim)**.** *Eine zusammengesetzte Zahl n für welche*

$$a^{n-1} \equiv 1 \mod n \tag{3.6}$$

gilt wird (fermat-)pseudoprim zur Basis a genannt.

Bemerkung. *Es gibt verschiedene Arten von Pseudoprimheit, wird aber nur das Wort „pseudoprim“ verwendet, meint man damit fermat-pseudoprim.*

Bemerkung. *Geschichtlich gesehen wurden alle Zahlen, die die Eigenschaft (3.6) erfüllen pseudoprim genannt. Primzahlen (die diese Eigenschaft ja immer erfüllen) werden heute nicht als pseudoprim angesehen. Für Zahlen, die zwar die Eigenschaft (3.6) erfüllen, von denen aber nicht bekannt ist, ob sie prim oder nichtprim sind, wird der Begriff „wahrscheinlich prim“ in der modernen Literatur verwendet.*

Für zusammengesetzte Zahlen kann man meist relativ leicht irgendeine Basis finden, sodaß die Zahl als zusammengesetzt erkannt wird. Jedoch existieren zusammengesetzte Zahlen n, die für alle möglichen Basen bei der Auswertung von (3.6) als Ergebnis 1 ergeben:

Definition (Carmichael-Zahlen)**.** *Zusammengesetzte Zahlen n, die für alle a mit $ggT(a,n) = 1$ die Gleichung $a^{n-1} \equiv 1 \mod n$ erfüllen, werden Carmichael-Zahlen genannt.*

Die kleinste dieser Zahlen ist $561 = 3 \cdot 11 \cdot 17$ es folgen 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, ... Alford, Granville und Pomerance haben 1992 in [16] bewiesen, daß es unendlich viele dieser Zahlen gibt, auch auf ihre Häufigkeit wurden Aussagen getroffen. Dementsprechend kann der Fermat-Test nicht durch wiederholtes Prüfen mit verschiedenen Basen und eventuell eine Angabe von einer endlichen Liste von Ausnahmen als Primzahltest verwendet werden.

3.2 Euler-Kriterium

Satz 3 (Euler-Kriterium)**.** *Falls p prim ist und für ein beliebiges a $ggT(a,p) = 1$ gilt, so gilt:*

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

wobei $\left(\frac{a}{p}\right)$ für das Legendre-Symbol steht. (Für die Berechnung des Legendre-Symbol siehe Anhang B.)

Dieser Satz kann genauso wie der Satz von Fermat für einen Test auf Zusammengesetztheit ausgenützt werden:

Sei n ungerade und $\mathrm{ggT}(a,n) = 1$. Ist nun

$$a^{\frac{n-1}{2}} \not\equiv \pm 1 \mod n,$$

so ist n zusammengesetzt. Ist $a^{\frac{n-1}{2}} \equiv \pm 1 \mod n$, so muß dieser Wert noch mit $\left(\frac{a}{p}\right)$ verglichen werden.

Leider gilt auch hier genau das Gleiche wie für den Fermat-Test: Aus

$$a^{\frac{n-1}{2}} \equiv \pm 1 \mod n$$

folgt nicht, daß n prim ist.

Dies legt die folgende Definition nahe:

Definition (euler-pseudoprim)**.** *Eine ungerade zusammengesetzte Zahl n, die*

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right)$$

erfüllt nennt man euler-pseudoprim zur Basis a.

Euler-Pseudoprimzahlen sind klarerweise auch fermat-pseudoprim, jedoch gilt das Umgekehrte nicht, so können viele Carmichael-Zahlen als zusammengesetzt erkannt werden!

Beispiel. *341 ist pseudoprim zur Basis 2 (s.o),* $2^{170} \equiv 1 \mod 341$*, jedoch ist* $\left(\frac{2}{341}\right) \equiv -1$*. Auch die Carmichaelzahl 561 kann mit dem Euler-Kriterium als nichtprim erkannt werden:* $5^{280} \equiv 67 \mod 561$*, hier braucht man das Jacobi-Symbol gar nicht berechnen.*

Eine fermat-pseodoprime Zahl n zur Basis a genügt der Gleichung $a^{n-1}-1 \equiv 0 \mod n$. n ist ungerade, kann also in Form $n = 2m + 1$ geschrieben werden

$$a^{n-1} - 1 \equiv (a^m - 1)(a^m + 1) \equiv 0 \mod n$$

Falls n prim ist, muß eine der Faktoren ein Vielfaches von n sein. Falls jetzt m gerade ist, kann man wieder in weitere Faktoren zerlegen und so weiter und so fort. Hat $n - 1$ also die Darstellung $d \cdot 2^s$ mit ungeradem d, so gilt

$$a^{n-1} - 1 = (a^d - 1)(a^d + 1)(a^{2d} + 1) \cdot \cdots \cdot (a^{2^{s-1}d} + 1).$$

Ist n prim, muß einer der Faktoren $\equiv 0 \mod n$ sein, aber auch hier gilt die Umkehrung nicht.

Definition (Starke Pseudoprimzahlen zur Basis a)**.** *Sei a eine beliebige natürliche Zahl > 1. Eine zusammengesetzte Zahl n mit $n - 1 = d \cdot 2^s$ (d ungerade) wird stark-pseudoprim zur Basis a genannt, falls entweder*

$$a^d \equiv 1 \mod n$$

oder

$$a^{d \cdot 2^r} \equiv -1 \mod n \text{ für ein } r \text{ aus } \{0, 1, \ldots, s-1\}$$

gilt.

Klarerweise gibt es weniger starke Pseudoprimzahlen als Fermat- bzw. Euler-Pseudoprimzahlen, es gibt zu jeder zusammengesetzten Zahl n eine Basis a, sodaß n keine starke Pseudoprimzahl zur Basis a ist. Damit gibt es keine zur Carmichaelzahlen ähnlichen „starken Pseudoprimzahlen“. Die kleinsten starken Pseudoprimzahlen zur Basis 2 sind 2047, 3277, 4033, 4681, 8321, ... Überprüft man

eine gewünschte Zahl zusätzlich mit den Basen 3 und 5, so gibt es bis $2.5 \cdot 10^{10}$ nur mehr 13 zusammengesetzte Zahlen, die nicht erkannt werden (die kleinste ist $25326001 = 2251 \cdot 11251$), nimmt man zusätzlich 7 dazu, nur mehr eine und beachtet man auch noch die Zahl 11, so werden bis $2.5 \cdot 10^{10}$ alle Zahlen richtig klassifiziert! Für „kleinere Zahlen“ wird in vielen Programmen wie z.B. Mathematica oder Derive die vorher erwähnte Methode verwendet. Je mehr Basen man verwendet, desto weniger zusammengesetzte Zahlen werden nicht erkannt, diese Eigenschaft wird auch für probabilistische Tests verwendet, siehe Kapitel 4. Aber egal wie viele Basen man auch in Betracht zieht, es werden immer Zahlen existieren, welche bei diesem Test nicht richtig erkannt werden - leider unendlich viele sogar. Es bleibt die Frage, ob man in Abhängigkeit der Größe der zu testenden Zahl die Anzahl der zu testenden Basen angeben kann, weiterführende Analysen und Bemerkungen dazu wiederum im Kapitel 4.

3.3 Umkehrung des Satzes von Fermat

Definition (Eulersche φ-Funktion)**.** *Die Eulersche φ-Funktion wird für eine natürliche Zahl durch*

$$\varphi(n) =\mid \{1 \leq a \leq n \mid ggT(a,n) = 1\} \mid$$

definiert.

Lemma 1. *Es gilt*

$$\varphi(n) =\mid \mathbb{Z}_n^* \mid,$$

wobei

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid \exists a^{-1} : a \cdot a^{-1} = 1\}$$

ist.

Beweis: Zu einer Zahl a in $\mathbb{Z}_n$ existiert ein inverses Element, falls in $\mathbb{Z}$ ein b mit $a \cdot b = n \cdot k + 1$, k beliebig, existiert. Für gegebenes a und n ist also die diophantische Gleichung $ax + ny = 1$ zu lösen. Diese Gleichung ist genau dann lösbar (siehe [4]), falls $\mathrm{ggT}(a,n) = 1$ ist, womit die Aussage gezeigt ist. □

Für eine Primzahl p ist klarerweise $\varphi(p) = p - 1$, allgemein gilt für eine Zahl $n = \prod p_i^{\alpha_i}$ mit paarweise verschiedenen p_i

$$\varphi(n) = \prod p_i^{\alpha_i - 1}(p_i - 1).$$

Für eine Herleitung der Berechnung der Eulersche φ-Funktion siehe z.B. [17].

Satz 4 (Euler-Fermat). *Für eine beliebige Zahl $m \in \mathbb{N}$ und für ein a mit $ggT(a, m) = 1$ gilt*

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

Beweis: Laut Definition der Eulerschen φ-Funktion hat die Menge $\mathbb{Z}_m^*$ genau $\varphi(m)$ Elemente,

$$\mathbb{Z}_m^* = \{a_1, a_2, \ldots, a_{\varphi(m)}\}.$$

Für ein beliebiges Element $b \in \mathbb{Z}_m^*$ sei $f(x) = bx$. Gilt für zwei Elemente x und y aus $\mathbb{Z}_m^*$, daß deren Bilder unter der Funktion f identisch sind, so gilt auch $x = y$ in $\mathbb{Z}_m^*$:

$$f(x) = f(y)$$

bedeutet

$$bx \equiv by \mod m,$$

und da b aus $\mathbb{Z}_m^*$ ist gilt $\mathrm{ggT}(b, m) = 1$, womit durch b gekürzt werden kann. Daraus folgt $x = y$ in $\mathbb{Z}_m^*$. $\mathbb{Z}_m^*$ ist bezüglich der Multiplikation abgeschlossen, deshalb ist f eine Permutation auf $\mathbb{Z}_m^*$. Damit gilt in $\mathbb{Z}_m^*$

$$\prod_{i=1}^{\varphi(m)} a_i = \prod_{i=1}^{\varphi(m)} f(a_i) = \prod_{i=1}^{\varphi(m)} (ba_i) = b^{\varphi(m)} \prod_{i=1}^{\varphi(m)} a_i.$$

Da in $\mathbb{Z}_m^*$ jedes Element ein Inverses hat, kann durch $\prod a_i$ dividiert werden, womit für alle $b \in \mathbb{Z}_m^*$

$$1 = b^{\varphi(m)}$$

gilt. Im Beweis wurde sowohl $m \in \mathbb{N}$ als auch $b \in \mathbb{Z}_m^*$ beliebig gewählt, woraus der Satz von Euler-Fermat folgt. □

Satz 5 (Umkehrung des Satzes von Fermat (Lehmer)). *Sei $n \geq 3$ ungerade und*

$$n - 1 = \prod_{i=1}^{r} p_i^{v_i}$$

die Primfaktorzerlegung von $n-1$ mit paarweise verschiedene Primzahlen p_i und ihrer Vielfachheit v_i. n ist dann und nur dann eine Primzahl, falls es ein a mit

$$a^{n-1} \equiv 1 \mod n$$

und

$$a^{\frac{n-1}{p_i}} \not\equiv 1 \mod n$$

für alle $i = 1, \ldots, r$ existiert.

Beweis: Für eine beliebige natürliche Zahl m gilt nach dem Satz von Euler-Fermat

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

Ist nun n keine Primzahl, so gilt stets $\varphi(n) < n-1$. Findet man ein a mit $a^{n-1} \equiv 1 \mod n$, so ist $\varphi(n)$ ein echter Teiler von $n-1$ und so auch ein Teiler von mindestens einen der $\frac{n-1}{p_i}$. Für dieses i gilt:

$$a^{\frac{n-1}{p_i}} \equiv a^{k\varphi(n)} \equiv (a^{\varphi(n)})^k \equiv 1^k \equiv 1 \mod n.$$

Damit kann kein a für ein zusammengesetztes n gefunden werden, welches die Bedingungen erfüllt.

Ist anderseits n prim, so gibt es in $\mathbb{Z}_n^*$ ein erzeugendes Element, und dieses hat genau die Eigenschaften, die von a gefordert werden. Um Eulers Satz zu verwenden, braucht man zusätzlich $\mathrm{ggT}(a,n) = 1$. Das stellt aber in keinster Weise ein Problem dar, denn falls $\mathrm{ggT}(a,n) > 1$, ist, so ist n nicht prim. (Da a nur modulo n betrachtet wird, kann man von $0 < a < n$ ausgehen.) □

Damit kann man (im Gegensatz zu den anderen Tests beruhend auf den Kleinen Fermatschen Satz) eindeutig feststellen, ob eine Zahl prim ist, falls man ein entsprechendes a findet. Diese Methode hat zwei Probleme: Man braucht die Zerlegung von $n-1$ und es gibt keine deterministische Methode, ein Primitivwurzel zu finden (es bleibt nur das Durchprobieren über).

Um sich die Suche nach Primitivwurzeln zu ersparen, kann man folgenden Satz verwenden:

Satz 6. *Sei $n \geq 3$ ungerade und*

$$n - 1 = \prod_{i=1}^{r} p_i^{v_i}$$

die Primfaktorzerlegung von $n-1$ in paarweise verschiedene Primzahlen p_i mit ihrer Vielfachheit v_i. Wenn für jedes p_i ein a_i existiert, sodaß

$$a_i^{n-1} \equiv 1 \mod n$$

und

$$a_i^{\frac{n-1}{p_i}} \not\equiv 1 \mod n,$$

dann ist n prim.

Beweis: Sei e_i die Ordnung von a_i (in $\mathbb{Z}_n^*$). $e_i|n-1$ und $e_i \nmid \frac{n-1}{p_i}$, daher gilt $p_i^{v_i}|e_i$. Für alle i gilt aber, daß $e_i|\varphi(n-1)$, damit gilt für alle i: $p_i^{v_i}|\varphi(n-1)$. Dies impliziert $\prod p_i^{v_i} = n-1$ teilt $\varphi(n-1)$, also muß $\varphi(n-1) = n-1$ sein und dies gilt nur, falls n prim ist. □

Was kann man machen, wenn man $n-1$ nicht in Primfaktoren zerlegen kann? Wenn man zumindest einen (genügend großen) Teil von $n-1$ zerlegen kann, kann man untersuchen, ob n prim ist (oder nicht).

Satz 7. *Sei* $n-1 = R \cdot F = R\prod_{i=1}^{r} p_i^{v_i}$, *mit* $ggT(p_i, p_j) = 1$ *für alle* $1 \leq i \neq j \leq r$, $ggT(R, F) = 1$ *und* $R < F$. *Gibt es nun ein* a *mit*

$$ggT(a^{\frac{n-1}{p_j}} - 1, n) = 1 \tag{3.7}$$

für alle $1 \leq i \leq r$ *und*

$$a^{n-1} \equiv 1 \mod n, \tag{3.8}$$

dann ist n *prim.*

Beweis: Sei q ein beliebiger Primfaktor von n. Aus (3.7) folgt für alle $1 \leq j \leq r$

$$\mathrm{ggT}(a^{\frac{n-1}{p_j}} - 1, q) = 1$$

und aus (3.8) folgt

$$a^{n-1} \equiv 1 \mod q.$$

Sei nun d die Ordnung von a in $\mathbb{Z}_q$. $d|n-1 = R\prod_{i=1}^{r} p_i^{v_i}$, jedoch gilt

$$d \nmid \frac{n-1}{p_j} = Rp_j^{v_j-1} \prod_{i=1, i\neq j}^{r} p_i^{v_i}.$$

$p_j^{v_j}|d$ gilt für alle j also muß auch $F|d$ gelten. Da d ein Teiler von $q-1$ sein muß, gilt auch $F|q-1$. Damit muß $q \geq F+1$ sein. Aus $n = RF$ mit $R < F$ folgt, daß $F > \sqrt{n}$ und daraus daß $q > \sqrt{n}$ ist. Wäre n nicht prim, wäre dies nun ein Widerspruch, womit die Primalität von n bewiesen ist. □

Die oben erwähnten Tests beruhen auf die Zerlegung von $n-1$ in ihre Primfaktoren, und auch wenn wie oben gezeigt man für n nicht alle Faktoren finden muß, es wird Zahlen geben, wo auch diese teilweise Faktorisierung Probleme bereitet. Auch das Finden entsprechenden Werte für a bleibt bestehen. Zumindest das erste Problem kann man umgehen, wenn man Tests entwickelt, die nur für bestimmte n-Werte konzipiert ist, nämlich solche, wo die Zerlegung von $n-1$ einfach ist. Die einfachste Form von n (für die Zerlegung) ist $2^m + 1$.

3.4 Pepintest

Satz 8 (Pepin)**.** *Eine Fermatzahl $F_n = 2^{2^n} + 1$ mit $n \geq 1$ ist dann und nur dann prim, wenn die folgende Eigenschaft gilt:*

$$3^{2^{2^n}-1} \equiv -1 \mod F_n$$

bzw. in einer leserlicheren Form

$$3^{\frac{F_n-1}{2}} \equiv -1 \mod F_n.$$

Beweis: Sei F_n prim, so folgt aus dem Euler-Kriterium

$$\left(\frac{3}{F_n}\right) \equiv 3^{\frac{F_n-1}{2}} \mod F_n.$$

Potenzen von 2 sind für gerade Potenzen 1 modulo 3 (und für ungerade 2 modulo 3), daher

$$F_n \equiv 2 \mod 3$$

Daraus läßt sich $(\frac{F_n}{3})$ berechnen:

$$\left(\frac{F_n}{3}\right) = \left(\frac{F_n \mod 3}{3}\right) = \left(\frac{2}{3}\right) = -1,$$

da modulo 3 nur 1 ein quadratischer Rest ist. Aus dem quadratischen Reziprozitätsgesetz und aus der Tatsache, daß $F_n \equiv 1 \mod 4$ ist, folgt nun

$$\left(\frac{3}{F_n}\right) = (-1)^{\frac{3-1}{2}\frac{F_n-1}{2}} \left(\frac{F_n}{3}\right) = 1 \cdot \left(\frac{F_n}{3}\right) = -1,$$

womit nur mehr zu zeigen ist, daß wenn $3^{2^{2^n}-1} \equiv -1 \mod F_n$ gilt F_n prim ist. Sei also nun

$$3^{\frac{F_n-1}{2}} \equiv -1 \mod F_n.$$

Quadriert man beide Seiten, ergibt dies

$$3^{F_n-1} \equiv 1 \mod F_n.$$

Daraus folgt, daß die Ordnung von 3 in $\mathbb{Z}_{F_n}^*$ ein Teiler von $F_n - 1$ ist. Aus der Ausgangsbedingung $3^{\frac{F_n-1}{2}} \equiv -1 \mod F_n$ folgt aber, daß die Ordnung von 3 gleich $F_n - 1$ sein muß, und damit ist F_n prim.

□

Kapitel 4

Probabilistische Tests

Probabilistische Tests beruhen auch auf den Kleinen Fermatschen Satz. Die Feststellung, ob eine große Zahl eine Primzahl ist oder nicht, ist eine zeitintensive Berechnung. In der Praxis ist jedoch oft genug ausreichend, wenn man nur mit einer (großen) Wahrscheinlichkeit diese Aussage trifft. Dafür wurden probabilistische Tests entwickelt, und wegen ihrer praktischen Bedeutung werden diese Tests in einem eigenen Kapitel behandelt.

4.1 Solovay-Strassen-Test

Für eine Primzahl p und eine ganze Zahl a gilt stets

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

Wie schon im vorigen Kapitel (Euler-Kriterium) erwähnt existieren leider auch zusammengesetzte Zahlen, die diese Eigenschaft erfüllen. Idee des von Solovay und Strassen in [19] vorgestellten Tests ist, diese Gleichung mit möglichst vielen Zahlen als Basis zu probieren. Je mehr Zahlen getestet werden und für diese die Gleichung erfüllt ist, desto wahrscheinlicher ist die Zahl p prim.

Algorithmus: Gegeben sei eine auf Primalität zu prüfende ungerade Zahl n und die Anzahl i der maximal zu testenden Basen:

1. $i^* = 1$ (Anzahl der geprüften Basen)
2. Wähle ein zufälliges a mit $1 < a < n$
3. Berechne $c = \mathrm{ggT}(a, n)$

4. Ist $c > 1$, Ausgabe „Zusammengesetzt“ und Beendigung des Algorithmus
5. Berechne $d \equiv a^{\frac{n-1}{2}} \mod n$
6. Ist $d \not\equiv \pm 1 \mod n$, Ausgabe „Zusammengesetzt“ und Beendigung des Algorithmus
7. Berechne $j = (\frac{a}{n})$
8. Ist $d \not\equiv j \mod n$, Ausgabe „Zusammengesetzt“ und Beendigung des Algorithmus
9. Ist $i^* < i$, so erhöhe i^* um 1 und gehe zu Schritt 2
10. Ausgabe „n ist wahrscheinlich eine Primzahl“

Die Richtigkeit des Algorithmus ist offensichtlich, da an allen Stellen, wo die Zahl als eine zusammengesetzte Zahl erkannt wird das Euler-Kriterium verletzt wird. Die offene Frage ist nur, mit welcher Wahrscheinlich n eine Primzahl ist.

Lemma 2. *Ist n ungerade und keine Primzahl, so existiert mindestens ein a, sodaß*

$$a^{\frac{p-1}{2}} \equiv \left(\frac{a}{n}\right) \mod n \tag{4.1}$$

nicht gilt.

Beweis: Es wird nun eine Konstruktion von so einem a angegeben.

- Fall 1: n ist nicht quadratfrei, sprich, es existiert eine Primzahl p mit $p^2 \mid n$: Sei $a = 1 + \frac{n}{p}$, so gilt

$$\begin{aligned} a^p &= \left(1 + \tfrac{n}{p}\right)^p \\ &= 1 + \tbinom{p}{1}\tfrac{n}{p} + \tbinom{p}{2}\left(\tfrac{n}{p}\right)^2 + \cdots + \tbinom{p}{p-1}\left(\tfrac{n}{p}\right)^{p-1} + \left(\tfrac{n}{p}\right)^p. \end{aligned}$$

  Für $i > 1$ gilt $(\frac{n}{p})^i \equiv 0 \mod n$, da $n = p^2 k$ für irgendein k und so

$$\begin{aligned} \left(\tfrac{n}{p}\right)^i &= \tfrac{n}{p} \cdot \tfrac{n}{p} \cdot \left(\tfrac{n}{p}\right)^{i-2} \\ &= \tfrac{n}{p} \cdot pk \cdot \left(\tfrac{n}{p}\right)^{i-2} \\ &= nk \cdot \left(\tfrac{n}{p}\right)^{i-2}. \end{aligned}$$

  Für $i = 1$ gilt

$$\binom{p}{1}\frac{n}{p} = p \cdot \frac{n}{p} = n$$

und somit gilt

$$a^p \equiv 1 \mod n.$$

Die Ordnung von a modulo n teilt daher p, und da p eine Primzahl ist, ist die Ordnung von a gleich p. Da nun $p \mid n$ gilt, teilt die Ordnung von a $n-1$ nicht, und deshalb ist

$$a^{n-1} \not\equiv 1 \mod n.$$

Aber es gilt (mit $n = p^2k$)

$$\begin{aligned}\left(\frac{a}{n}\right) &= \left(\frac{1+\frac{n}{p}}{p^2k}\right) \\ &= \left(\frac{1+pk}{p}\right)\left(\frac{1+pk}{pk}\right) \\ &= \left(\frac{1}{p}\right)\left(\frac{1}{pk}\right) \\ &= 1 \cdot 1 = 1.\end{aligned}$$

- Fall 2: Ist n quadratfrei, dann sei p ein Primteiler. Man wähle einen beliebigen quadratischen Nichtrest c aus $\mathbb{Z}_p^*$. Der Chinesische Restsatz garantiert, daß man ein $a \in \mathbb{Z}_p^*$ wählen kann mit

$$\begin{aligned}a &\equiv c \mod p \\ a &\equiv 1 \mod \frac{n}{p}.\end{aligned} \tag{4.2}$$

(Hierfür ist notwendig, daß n quadratfrei ist, da dadurch $\mathrm{ggT}(p, \frac{n}{p}) = 1$ gewährleistet ist.) Für dieses a gilt

$$\begin{aligned}\left(\frac{a}{n}\right) &= \left(\frac{a}{p}\right)\left(\frac{a}{\frac{n}{p}}\right) \\ &= \left(\frac{c}{p}\right)\left(\frac{1}{\frac{n}{p}}\right) \\ &= -1 \cdot 1 = -1,\end{aligned}$$

aus (4.2) folgt

$$a^{\frac{n-1}{2}} \equiv 1 \mod \frac{n}{p}. \tag{4.3}$$

Kann nun

$$a^{\frac{n-1}{2}} \equiv -1 = \left(\frac{a}{n}\right) \mod n \tag{4.4}$$

gelten? (4.3) bedeutet $a^{\frac{n-1}{2}} = 1 + k_1\frac{n}{p}$ für irgendein k_1. Damit (4.4) gilt muß es ein ganzzahliges k_2 geben mit $a^{\frac{n-1}{2}} = -1 + k_2 n$. Um k_2 zu finden werden diese zwei Gleichungen gleichgesetzt, es muß also

$$1 + k_1\frac{n}{p} = -1 + k_2 n$$

gelten. k_2 herausgehoben bedeutet dies

$$k_2 = \frac{k_1 n + 2p}{np}$$

Damit k_2 ganzzahlig ist, muß $k_1 n + 2p = k_3 np$ bzw. $k_1 n - k_3 np = 2p$ sein. So ein Zahlenpaar (k_1, k_3) läßt sich nur finden, falls $\mathrm{ggT}(n, np) = n \mid 2p$ gilt. Die einzige Möglichkeit für ein nicht primes n ist $n = 2p$, jedoch werden hier laut Voraussetzung nur ungerade Zahlen behandelt, also kann (4.4) nicht gelten.

Damit kann man zu einem beliebigen ungeraden n stets ein a finden, welches die Gleichung (4.1) nicht erfüllt. □

Satz 9. *Sei n ungerade und zusammengesetzt. Für mindestens die Hälfte der Zahlen a aus $\mathbb{Z}_n^*$ gilt:*

$$a^{\frac{n-1}{2}} \not\equiv \left(\frac{a}{n}\right) \mod n \tag{4.5}$$

Beweis: Sei

$$A = \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \mod n\} \tag{4.6}$$

Lemma 2 stellt sicher, daß es ein $b \in \mathbb{Z}_n^*$ jedoch nicht aus A existiert. Für jedes $a \in A$ gilt nun klarerweise $\mathrm{ggT}(ab, n) = 1$, des Weiteren

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right) \not\equiv a^{\frac{n-1}{2}} b^{\frac{n-1}{2}}$$

Da hier keine Einschränkungen für a getroffen worden sind, gilt für alle $a \in A$, daß $ab \notin A$, womit mindestens so viele Zahlen nicht in A sind wie in A. □

Dies bedeutet nun für diesen Primzahltest, daß, wenn nach dem Test mit einer Basis n nicht als zusammengesetzt entlarvt worden ist, n mit der Wahrscheinlichkeit von mindestens $\frac{1}{2}$ prim ist. Dadurch ist nach k Schritten, ohne n als zusammengesetzt erkannt zu haben, die Wahrscheinlichkeit, daß n eine Primzahl ist, mindestens $1 - \frac{1}{2^k}$. Damit kann man mit relativ wenig Testschritten eine relativ hohe Wahrscheinlichkeit erreichen, daß dieser Test die richtige Antwort gibt, und man kann auch zu jeder vorgegebenen maximalen Fehleranfälligkeit leicht die Anzahl der Testschritte angeben.

4.2 Rabin-Miller-Test

Ähnlich wie der Solovay-Strassen-Test geht auch dieser Test von Verallgemeinerung des Kleinen Fermatschen Satzes aus, jedoch ist dieser Test insgesamt besser,

dieser Test wird auch in der Praxis wohl am häufigsten verwendet. Auch der Name Rabin-Selfridge-Miller-Test wird öfters gebraucht[1], da schon Selfridge 1974 vor Miller diesen Test verwendete (siehe [20]). Miller selber veröffentlichte 1976 in [21] einen Primzahltest, wobei er zuersteinmal auf die Erweiterte Riemannsche Hypothese aufbauend einen deterministischen Test angab. Erst 1980 veröffentlichte Rabin in [22] die hier vorgestellte Fehlerabschätzung des probabilistischen Tests.

Algorithmus: Gegeben sei eine auf Primalität zu prüfende ungerade Zahl n und die Anzahl m der maximal zu testenden Basen:

1. Berechne die Werte m und h mit $n-1 = 2^h \cdot m$ und m ungerade
2. $i = 1$ (Anzahl der geprüften Basen)
3. Wähle ein zufälliges a mit $1 < a < n$
4. Berechne $c = \mathrm{ggT}(a, n)$
5. Ist $c > 1$, Ausgabe „Zusammengesetzt“ und Beendigung des Algorithmus
6. Berechne $d \equiv a^m \mod n$
7. Ist $d \equiv 1 \mod n$, gehe zu Schritt 13
8. $r = 0$
9. Berechne $e_r \equiv a^{2^r m} \mod n$
10. Ist $e_r \equiv -1 \mod n$, gehe zu Schritt 13
11. Ist $r < h-1$, so erhöhe r um eins und gehe zu Schritt 9
12. Ausgabe „Zusammengesetzt“ und Beendigung des Algorithmus
13. Ist $i^* < i$, so erhöhe i^* um 1 und gehe zu Schritt 3
14. Ausgabe „n ist wahrscheinlich eine Primzahl“

Für die Berechnung der Werte e_r kann $e_0 = d$ und $e_{r+1} = (e_r)^2$ herangezogen werden.

Ist n eine zusammengesetzte Zahl, so sind mindestens 3/4 der Werte für a Zeugen, daß m zusammengesetzt ist. Dementsprechend kann man mit k verschiedenen zufällig gewählten a-Werten die Fehlerwahrscheinlichkeit, daß eine zusammengesetzte Zahl nicht als solche erkannt wird, auf unter $\frac{1}{4^k}$ drücken.

[1] Auch die Reihenfolge der Namen im Namen des Tests ist nicht eindeutig, es existiert sowohl die Bezeichnung Miller-Rabin-Tests als auch die Bezeichnung Rabin-Miller-Test.

Lemma 3. *Ist p eine ungerade Primzahl, so hat die Gleichung*

$$x^2 \equiv 1 \mod p \tag{4.7}$$

genau zwei Lösungen, nämlich ± 1.

Beweis: Aus (4.7) folgt

$$(x-1)(x+1) \equiv 0 \mod p \tag{4.8}$$

Die Gleichung (4.8) kann nur dann erfüllt werden, wenn $(x-1)(x+1)$ durch p teilbar ist, und da p eine Primzahl ist, muß p entweder $x-1$ oder $x+1$ teilen, woraus $x \equiv 1$ oder $x \equiv -1 \mod p$ folgt. □

Lemma 4. *Falls p prim ist, wird der Rabin-Miller-Test p nie als zusammengesetzt erkennen.*

Beweis: Ist p eine Primzahl, dann muß laut dem Kleinen Fermatschen Satz $a^{n-1} = a^{2^t m} \equiv 1 \mod n$ sein und wegen Lemma 3 muß $a^{2^{t-1}m} \equiv \pm 1 \mod n$ sein. Ist der Wert gleich -1, so ist man fertig, ist dieser Wert gleich 1, kann Lemma 3 erneut angewendet werden. Wenn die Reihe nie abbricht, sprich nie -1 wird, so muß dann $a^m \equiv 1 \mod n$ sein. □

Um über die Qualität des Rabin-Miller-Testes zu urteilen, bleibt die Frage, wie viele Zahlen „falsche Zeugen“ sind, und mit einer Angabe diesbezüglich ist eine Abschätzung möglich, wie viele Testschritte man für eine vorgegebene Genauigkeit mindestens braucht.

Satz 10. *Für ungerade zusammengesetzte Zahlen n ungleich 9 sind höchstens $1/4$ der zu n teilerfremden Zahlen falsche Zeugen, anders formuliert sind $3/4$ der Zahlen a mit $ggT(a,n) = 1$ sind Zeugen, daß n keine Primzahl ist.*

Beweis: Existiert für eine zusammengesetzte Zahlen kein falscher Zeugen, so ist der Beweis fertig. Existiert ein falscher Zeuge, so existiert auch ein a mit

$$a^{2^r m} \equiv -1 \mod n \tag{4.9}$$

für $0 \leq r \leq h-1$. (Würde so ein a nicht existieren, muss für einen falschen Zeugen a stets $a^m \equiv 1 \mod n$ gelten, hierbei wäre für $-a$ jedoch $(-a)^m \equiv -1$, was ein Widerspruch wäre.) Damit existiert ein größtes $s \in \mathbb{N}$, sodaß mindestens für ein a

$$a^{2^s m} \equiv -1 \mod n \tag{4.10}$$

gilt.

Nun werden 4 Hilfsmengen für den Beweis definiert. Sei $d = 2^s m$ und $\prod p_i^{\alpha_i}$ die Primzerlegung von n mit paarweise verschiedenen p_i. Damit seien die Mengen A, B, C und D wie folgt definiert:

$$\begin{aligned} A &= \{a \in \mathbb{Z}_n \mid a^d \equiv 1 \mod n\} \\ B &= \{a \in \mathbb{Z}_n \mid a^d \equiv \pm 1 \mod n\} \\ C &= \{a \in \mathbb{Z}_n \mid a^d \equiv \pm 1 \mod p_i^{\alpha_i} \text{ für alle } i\} \\ D &= \{a \in \mathbb{Z}_n \mid a^{n-1} \equiv 1 \mod n\}. \end{aligned}$$

Für diese Mengen gilt

$$A \subseteq B \subseteq C \subseteq D \subseteq \mathbb{Z}_n^*.$$

Ist nun a nicht Zeuge für die Zusammengesetztheit von n, so ist $a \in B$. Es gilt, eine Größenabschätzung zwischen diesen Mengen anzugeben.

Zuerst wird der Index $(C : A)$ betrachtet. Hierzu sei die Gruppe

$$G = \prod_{p_i} \{-1, 1\}$$

und die Abbildung f von C nach G gegeben durch

$$a \to (a^d \mod p_1^{\alpha_1}, \quad a^d \mod p_2^{\alpha_2}, \ldots, \quad a^d \mod p_t^{\alpha_t}),$$

wobei n genau t verschiedene Primzahlen als Teiler hat. Auf die Gruppe G läßt sich eine Multiplikation durch $(a_1, a_2, \ldots, a_t) \cdot (b_1, b_2, \ldots, b_t) = (a_1 b_1, a_2 b_2, \ldots, a_t b_t)$ definieren, als neutrale Element bezüglich dieser Multiplikation dient $(1, 1, \ldots, 1)$. Die Bedingung $f(a \cdot b) = f(a) \cdot f(b)$ ist wie einfach zu sehen erfüllt, deswegen ist f ein Homomorphismus von C nach G. Auf das neutrale Element werden die Zahlen abgebildet, für die $a^d \equiv 1 \mod p_i^{\alpha_i}$ für alle i gilt. Laut dem Chinesischen Restsatz läßt sich diese Bedingung auf $a^d \equiv 1 \mod n$ zusammenfassen, womit der Kern der Abbildung genau die Menge A ist.

Der Homomorphiesatz besagt, daß falls f ein Homomorphismus von X nach Y ist, so ist $X/\ker(f) \cong \mathrm{im}(f)$. Wählt man nun für X die Menge C und für Y die Menge G und kann man das Bild von f angeben, so hat man den Index $(C : A)$. Um diesen Schritt zu vervollständigen wählen wir ein b mit $b^d \equiv -1 \mod n$ (so ein b existiert, da d genau auf diese Bedingung hin konzipiert worden ist), was $b^d \equiv -1 \mod p_i^{\alpha_i}$ für alle i impliziert. Sei nun a_i für $0 \leq i \leq t$ beliebig $+1$ oder -1, so kann man mit Hilfe des Chinesischen Restsatzes ein $x \in \mathbb{Z}$ mit $x \equiv b^{\frac{1-a_i}{2}} \mod p_i^{\alpha_i}$ konstruieren.

Es gilt

$$x^d \equiv \left(b^{\frac{1-a_i}{2}}\right)^d \equiv \left(b^d\right)^{\frac{1-a_i}{2}} \equiv (-1)^{\frac{1-a_i}{2}} \equiv a_i \mod p_i^{\alpha_i},$$

womit x in C liegt und $f(x) = (a_1, a_2, \cdots, a_t)$. Da alle a_i-Werte beliebig aus der Menge $\{-1, 1\}$ gewählt worden sind, sind damit alle Werte aus G im Bild von f. G hat 2^t Elemente. Zusammenfassend ergibt dies

$$C/\ker(f) = C/A \cong \mathrm{im}(f) = G,$$

womit der Index

$$(C : A) = 2^t \tag{4.11}$$

ist, wobei t die Anzahl der verschiedenen Primteiler von n ist.

Nun wird der Index $(B : A)$ ermittelt. Hierzu sei

$$H = \{a \in \mathbb{Z}_n \mid a^d \equiv -1 \mod n\}.$$

Klarerweise ist

$$H \cup A = B \text{ und } H \cap A = \varnothing, \tag{4.12}$$

und H kann nicht die leere Menge sein, es existiert ein b mit $b^d \equiv -1 \mod n$. Ist b ein Element von H und $a \in A$, so ist $a \cdot b$ wegen $(a \cdot b)^d \equiv a^d \cdot b^d \equiv 1 \cdot (-1) \equiv -1 \mod n$ aus H. Sei nun

$$H_2 = \{c \cdot b \mid c \in A\},$$

so hat diese neue Menge genauso viele Elemente wie A. Angenommen es existiert ein $z \in H$, sodaß $z \notin H_2$. $z \cdot b$ ist ein Element aus A, womit $z \cdot b^2$ wiederum ein Element aus H_2 ist. Dies läßt sich so fortsetzen, für gerade y wird $z \cdot b^y$ stets aus H_2 sein. Damit ist $z \cdot b^{2m} \in H_2$, jedoch gilt

$$z \cdot b^{2m} \equiv z \cdot (b^m)^2 \equiv z \cdot (-1)^2 \equiv z \mod n,$$

woraus $z \in H_2$ folgt, was ein Widerspruch zur Annahme ist. Damit gilt $H_2{=}H$, und somit ist $\mid A \mid{=}\mid H \mid$ und und aus (4.12) folgt $\mid B \mid{=} 2 \mid A \mid$, was weiterführend

$$(B : A) = 2 \tag{4.13}$$

bedeutet. Aus (4.11) und (4.13) folgt

$$(C : B) = 2^{t-1}. \tag{4.14}$$

Um über den Index $(\mathbb{Z}_n^* : B)$ eine Aussage zu treffen, werden verschiedenen Fälle in Abhängigkeit von der Anzahl der verschiedenen Primteiler von n betrachtet:

- Hat nun n mindestens 3 verschiedene Primteiler, so ist man fertig, da wegen $B \subseteq C \subseteq D \subseteq \mathbb{Z}_n^*$ und a kein Zeuge für die Zusammengesetztheit $a \in B$ impliziert schon der Index $(C : B) = 2^{3-1} = 4$ ist.

- Hat n 2 verschiedene Primteiler, so ist $(C : B) = 2$. Jedoch gibt es keine Carmichaelzahlen mit nur 2 verschiedenen Primfaktoren, somit ist $D \neq \mathbb{Z}_n^*$ und der Index $(\mathbb{Z}_n^* : D)$ ist mindestens 2, womit $(\mathbb{Z}_n^* : B)$ mindestens 4 ist und damit ist dieser Fall auch abgeschlossen.

- Es bleibt der Fall $n = p^\alpha$ zu behandeln. $\mathbb{Z}_{p^\alpha}^*$ ist zyklisch und hat die Ordnung $\varphi(p^\alpha) = p^{\alpha-1}(p-1)$. Sei g eine Primitivwurzel von $\mathbb{Z}_{p^\alpha}^*$, so ist $j = \varphi(p)$ der erste Exponent mit $g^j \equiv 1 \mod n$. Ein $b \in \mathbb{Z}_{p^\alpha}^*$ läßt sich durch $b = g^i$ für ein bestimmtes $1 \leq i \leq p^{\alpha-1}(p-1)$ ausdrücken. Für wie viele b gilt $b^{n-1} \equiv 1 \mod n$? Man kann b stets als eine Potenz der Primitivwurzel ausdrücken,

  $$b^{n-1} = \left(g^i\right)^{n-1} = b^{(n-1)\cdot i},$$

  somit ist die Frage gleichbedeutend mit für wie viele $1 \leq i \leq p^{\alpha-1}(p-1)$ gilt

  $$p^{\alpha-1}(p-1) \mid (n-1) \cdot i.$$

  Da $n - 1 = p^\alpha - 1$ ist, ist $\mathrm{ggT}(p^{\alpha-1}, n-1) = 1$, womit i den Faktor $p^{\alpha-1}$ enthalten muß, es gilt $i = x \cdot p^{\alpha-1}$. Da i zwischen 1 und $p^{\alpha-1}(p-1)$ liegt, kommen genau $p-1$ verschiedene Werte für i in Frage. Damit ist der Index $(\mathbb{Z}_n^* : B)$ mindestens $\frac{p^{\alpha-1}}{(p-1)}(p-1) = p^{\alpha-1}$. Für alle Primzahlpotenzen von n ist diese Summe stets größer als 4, ausgenommen für $n = 3^2$.

□

Bemerkung. *Die Ausnahme 9 bereitet keine Schwierigkeit, da diese Tests für große n konzipiert sind, des Weiteren hat $\mathbb{Z}_{3^2}^*$ genau 6 Elemente wovon nur 1 und 8 falsche Zeugen sind, läßt man den Rabin-Miller-Test mindestens mit 3 verschiedenen Basen durchlaufen, wird 9 als zusammengesetzt erkannt.*

Eine weitere direkte Folge dieser Aussage ist, daß, wenn man mehr als 1/4 aller möglichen Basen durchprobiert und n nicht als zusammengesetzt erkannt worden ist, n eine Primzahl ist und man hat einen echten Primzahltest. Diese Berechnung würde leider viel zu lange dauern und erweist sich deshalb als unpraktikabel. Verschiedene Personen haben Tabellen aufgestellt (siehe z.B. [7]), bis zu welcher Größe von n gewisse Basen ausreichen, um n einwandfrei zu klassifizieren, wodurch der Test durch die geringe Anzahl der Basen sehr schnell funktioniert und aus diesem Grund auch oft verwendet wird.

Miller zeigte in [21], unter der Annahme, die unbewiesene Erweiterte Riemannsche Hypothese sei richtig, daß der Miller-Rabin-Test auch als nichtprobabilistischer Test verwendbar ist, nämlich wenn für a alle Werte aus dem Intervall $[2; \min(n-1, \lfloor 2(\ln n)^2 \rfloor)]$ getestet werden, das Testen dieser Basen benötigt eine polynomiale Laufzeit.

Kapitel 5

Lucas-Lehmer-Tests

Definition (Lucas-Folgen)**.** *Gegeben sei die Gleichung* $x^2 - Px + Q = 0$ *mit* $P, Q \in \mathbb{Z}$*, die Lösungen seien die reellen Zahlen* a *und* b*,* $a \neq b$*.* U_n *und* V_n *werden durch*

$$U_n = \frac{a^n - b^n}{a - b}, \quad n \geq 0$$
$$V_n = a^n + b^n, \quad n \geq 0$$

definiert und Lucas-Folgen genannt[1]*.*

Es seien 2 Identitäten (Vietà) erwähnt, welche in der Folge die Lesbarkeit verbessern:

$$a + b = P, \quad ab = Q.$$

5.1 Berechnung der Folgeglieder

Die Folgenglieder für $n = 0$ und $n = 1$ kann man leicht ablesen:

$$U_0 = 0, \quad U_1 = 1$$
$$V_0 = 2, \quad V_1 = a + b = P.$$

Die Lucas-Folgen erfüllen die Rekursion

$$U_{n+2} = PU_{n+1} - QU_n$$

bzw.

$$V_{n+2} = PV_{n+1} - QV_n.$$

[1] Dieses Kapitel ist anhand [7] aufgebaut.

Diese Folgen haben weitere Eigenschaften, welche das Berechnen der Folgeglieder vereinfacht und ermöglicht, daß man um einen Folgeglied zu erhalten nicht alle Folgeglieder bis zum gewünschten Folgeglied (mit der Rekursionsformel) berechnen muß. Man gehe davon aus, daß $m \geq n$ ist:

$$\begin{aligned} U_{m+n} &= \tfrac{a^{m+n}-b^{m+n}}{a-b} \\ &= \tfrac{(a^m-b^m)(a^n+b^n)}{a-b} - \tfrac{a^n b^n (a^{m-n}-b^{m-n})}{a-b} \\ &= U_m V_n - Q^n U_{m-n}, \\ V_{m+n} &= a^{m+n} + b^{m+n} \\ &= (a^m+b^m)(a^n+b^n) - a^n b^n (a^{m-n}+b^{m-n}) \\ &= V_m V_n - Q^n V_{m-n}. \end{aligned}$$

Setzt man nun $m = n$:

$$\begin{aligned} U_{2n} &= U_n V_n - Q^n U_0 = U_n V_n \\ V_{2n} &= V_n V_n - Q^n V_0 = V_n^2 - 2Q^n. \end{aligned}$$

Weiters bekommt man bei $m = n + 1$:

$$\begin{aligned} U_{2n+1} &= U_{n+1} V_n - Q^n U_1 = U_{n+1} V_n - Q^n \\ V_{2n+1} &= V_{n+1} V_n - Q^n V_1 = V_{n+1} V_n - Q^n P. \end{aligned}$$

Mit diesen Rekursionen lassen sich alle Folgeglieder (wie leicht ersichtlich) berechnen, es ist damit auch bewiesen, daß alle Folgeglieder aus $\mathbb{Z}$ sind.

Es stellt sich nun die Frage, wie viele Schritte man benötigt, um U_n beziehungsweise V_n auszurechnen.

Berechnung von V_n für ein gegebenes n: (Bemerkung: dies ist die einfachere Folge, da man bei der Rekursion nur Werte der Folge V_n berechnen muß.)

Ist n ein Zweierpotenz:

$$V_1 \to V_2 \to V_4 \to V_8 \to V_{16} \ldots$$

Man muß sich jeweils nur das Ergebnis merken, denn die Vorgänger werden bei der weiteren Berechnung nicht benötigt. Ist n keine Zweierpotenz, so erreicht man dieses Folgeglied V_n nur durch verdoppeln klarerweise nicht. Zum Beispiel braucht man für V_{13} die Folgeglieder V_6 und V_7. Für V_{13} müssen also zwei Werte berechnet werden. Mehr als zwei Werte müssen aber nie gleichzeitig gemerkt werden. Will man V_{2n+1} und V_{2n} berechnen, braucht man nur die Werte V_{n+1}

und V_n (s.o.). Auch wenn der größere Index gerade ist, braucht man nur zwei Folgeglieder berechnen (wie z.B. bei $V_7 = V_4V_3 - Q^nP$):

$$V_{2n} = V_n^2 - 2Q^n, \quad V_{2(n-1)+1} = V_nV_{n-1} - Q^{n-1}P.$$

Damit ist gewährleistet, daß der Speicherbedarf bei der Berechnung nicht mit dem Index immer größer wird.

Nun zum Algorithmus, um V_n zu berechnen. Sei

$$n = \prod_{i=0}^{s} a_i 2^i,$$

wobei a_i 0 oder 1 ist mit $a_s = 1$ (sprich die Binärdarstellung von n).

Aus einem Wertepaar (V_{j+1}, V_j) kann man mit den oben beschriebenen Rekursionen die Werte V_{2j+2}, V_{2j+1} und V_{2j} berechnen. Diese können entweder zum Wertepaar (V_{2j+2}, V_{2j+1}) oder zum Wertepaar (V_{2j+1}, V_{2j}) zusammengefasst werden. Startet man vom bekannten Wertepaar (V_1, V_0) und nimmt man als Indikator, welches Wertepaar man berechnet nacheinander $a_s, a_{s-1}, \ldots, a_0$ aus der Binärdarstellung von n, wobei man bei $a_i = 1$ den Wertepaar mit dem größeren Index und dementsprechend bei $a_i = 0$ den Wertepaar mit dem kleineren Index nimmt, so erhält man nach dem letzten Schritt (V_n, V_{n-1}).

Da man nur den Wert V_n berechnen will, kann man auf die Berechnung von V_{n-1} im letzten Schritt verzichten. Je nachdem, durch welche Zweierpotenz n teilbar ist, kann man die Berechnung von jeweils zwei Folgengliedern auf einen reduzieren.

Beispiel: Berechne V_{844}.

$$844 = 512 + 256 + 64 + 8 + 4 = (1101001100)_2$$

$$\begin{aligned}
&(V_1, V_0) \longrightarrow (V_2, V_1) \longrightarrow (V_4, V_3) \longrightarrow \\
&\longrightarrow (V_7, V_6) \longrightarrow (V_{14}, V_{13}) \longrightarrow (V_{27}, V_{26}) \longrightarrow \\
&\longrightarrow (V_{53}, V_{52}) \longrightarrow (V_{106}, V_{105}) \longrightarrow (V_{212}, V_{211}) \longrightarrow \\
&\longrightarrow (V_{423}, V_{422}) \longrightarrow (V_{485}, V_{844})
\end{aligned}$$

Es ist nicht nötig die Folgenglieder V_{212}, V_{423} und V_{845} zu berechnen, dementsprechend kann man diese Schritte vernachlässigen.

Will man einen Wert aus der Folge U_n berechnen, geht das Verfahren ähnlich, es reicht aber nun nicht mehr aus, nur Werte der Reihe U_n zu berechnen, auch Werte der Folge V_n müssen ermittelt werden. (Soweit n keine Zweierpotenz ist.)

Vom Prinzip her ist die Berechnung gleich wie die Ermittlung der Werte V_n, jedoch werden zum Wertepaar (V_{j+1}, V_j) auch jeweils der Wert U_{j+1} berechnet.

Beispiel: Berechne U_{844}.

$$\begin{aligned}
&(U_1, V_1, V_0) \longrightarrow (U_2, V_2, V_1) \longrightarrow \\
&\longrightarrow (U_4, V_4, V_3) \longrightarrow (U_7, V_7, V_6) \longrightarrow \\
&\longrightarrow (U_{14}, V_{14}, V_{13}) \longrightarrow (U_{27}, V_{27}, V_{26}) \longrightarrow \\
&\longrightarrow (U_{53}, V_{53}, V_{52}) \longrightarrow (U_{106}, V_{106}, V_{105}) \longrightarrow \\
&\longrightarrow (U_{212}, V_{212}, V_{211}) \longrightarrow (U_{423}, V_{423}, V_{422}) \longrightarrow \\
&\longrightarrow (U_{485}, V_{485}, V_{844})
\end{aligned}$$

5.2 Teilbarkeitseigenschaften der Folgenglieder U_n

Definition (Quadratischer Zahlenkörper)**.** *Für ein quadratfreies D heißt die Menge*

$$\mathbb{Q}(\sqrt{D}) = \{a + b\sqrt{D} \mid a, b \in \mathbb{Q}\}$$

quadratischer Zahlenkörper. Für ein $a = a_1 + a_2\sqrt{D}$ ist $\overline{a} = a_1 - a_2\sqrt{D}$.

Sei $P^2 - 4Q = c^2 D$, mit $c, D \in \mathbb{N}$ und D quadratfrei. Ist nun $D > 1$, so sind die zwei Lösungen a und b der Gleichung $x^2 - Px + Q = 0$ irrationale Zahlen aus dem quadratischen Zahlenkörper $\mathbb{Q}(\sqrt{D})$.

Es existiert ein Pendant zum Kleinen Fermatschen Satz in quadratischen Zahlenkörpern. Ist p eine ungerade Primzahl mit

$$p \nmid D, \tag{5.1}$$

so gilt für alle $a \in \mathbb{Q}(\sqrt{D})$:

$$a^p \equiv a \mod p, \text{ falls } \left(\frac{D}{p}\right) = 1 \tag{5.2}$$

bzw.

$$a^p \equiv \overline{a} \mod p, \text{ falls } \left(\frac{D}{p}\right) = -1. \tag{5.3}$$

Für die Herleitung dieser Formel siehe z.B. [7]., Appendix 4.

Falls $\mathrm{ggT}(a, p) = 1$ in $\mathbb{Q}(\sqrt{D})$ ist, dann folgt aus (5.2)

$$a^{p-1} \equiv 1 \mod p, \text{ falls } \left(\frac{D}{p}\right) = 1. \tag{5.4}$$

Aus (5.3) folgt stets

$$a^{p+1} \equiv a\overline{a} \mod p, \text{ falls } \left(\frac{D}{p}\right) = -1. \tag{5.5}$$

Lemma 5. *Die Bedingung*

$$ggT(a,p) = 1 \text{ in } \mathbb{Q}(\sqrt{D}) \tag{5.6}$$

ist identisch mit der Bedingung

$$ggT(Q,p) = 1 \text{ in } \mathbb{N}. \tag{5.7}$$

Diese Eigenschaft gilt allgemein für jedes a aus $\mathbb{Q}(\sqrt{D})$. Für Lucas-Folgen ist diese Eigenschaft nur für die Lösungen der Gleichung $x^2-Px+Q=0$ interessant, deswegen beschränkt sich die Arbeit nur auf diesen Sonderfall.

Beweis: Im quadratischen Zahlenkörper ist für alle Zahlen a mit $N(a) = a\overline{a}$ eine Norm definiert. Durch einfaches Ausmultiplizieren ist leicht nachvollziehbar, daß

$$N(a \cdot b) = N(a) \cdot N(b)$$

gilt. Teilt nun eine Zahl d sowohl a als auch p, so muß $N(d)$ sowohl $N(a)$ als auch $N(p)$ teilen, woraus aus (5.6)

$$\mathrm{ggT}(N(a), N(p)) = 1 \tag{5.8}$$

folgt. $N(a) = a\overline{a}$ ist nach dem Satz von Vietà für die Lösung der Gleichung $x^2 - Px + Q = 0$ gleich Q, womit (5.8) gleichbedeutend mit

$$\mathrm{ggT}(Q, p^2) = 1 \tag{5.9}$$

ist. Somit muß auch $\mathrm{ggT}(Q,p) = 1$ gelten. □

Aus den Identitäten ergeben sich folgende Berechnungen für U_{p-1} bzw. U_{p+1} (beachte: $b = \overline{a}$):

$$U_{p-1} = \frac{a^{p-1} - \overline{a}^{p-1}}{a - \overline{a}} \equiv \frac{1-1}{a-\overline{a}} \equiv 0 \mod p, \text{ falls } \left(\frac{D}{p}\right) = 1, \tag{5.10}$$

$$U_{p+1} = \frac{a^{p+1} - \overline{a}^{p+1}}{a - \overline{a}} \equiv \frac{a\overline{a} - \overline{a}a}{a-\overline{a}} \equiv 0 \mod p, \text{ falls } \left(\frac{D}{p}\right) = -1. \tag{5.11}$$

Bemerkung. *Es muß darauf geachtet werden, daß $a-\overline{a}$ nicht 0 sein darf, denn sonst ist die Division nicht durchführbar. $(a-\overline{a})^2 = c^2 D$, also darf c den Faktor p nicht erhalten, es muß daher*

$$ggT(c,p) = 1 \tag{5.12}$$

gelten. Zusammen mit der Bedingung (5.7) muß $ggT(cQ,p) = 1$ erfüllt sein.

Nun wird betrachtet, wie Potenzen von a bzw. U_i sich bezüglich der Teilbarkeit durch p^n verhalten. Für den Fall $\left(\frac{D}{p}\right) = 1$ ist $a^p = a + kp$ für ein bestimmtes k und daher gilt

$$a^{p^n} = (a^p)^{p^{n-1}} = (a+kp)^{p^{n-1}} \equiv a^{p^{n-1}} \mod p^n,$$

da alle Terme abseits des ersten ($a^{p^{n-1}}$) stets den Faktor p^n enthalten. Genauso wie aus (5.2) (5.4) folgt, kann man hier unter der Bedingung $\mathrm{ggT}(a,p) = 1$ in $\mathbb{Q}(\sqrt{D})$ durch Division durch $a^{p^{n-1}}$ das Folgende erhalten:

$$a^{p^{n-1}(p-1)} \equiv 1 \mod p^n. \tag{5.13}$$

Analog erhält man

$$a^{p^{n-1}(p+1)} \equiv (a\overline{a})^{p^{n-1}} \mod p^n, \text{ falls } \left(\frac{D}{p}\right) = -1. \tag{5.14}$$

Mit diesen Identitäten können die folgenden Folgeglieder U_i berechnet werden:

$$U_{p^{n-1}(p-1)} \equiv 0 \mod p^n \quad \text{für } \mathrm{ggT}(a,p) = 1 \text{ in } \mathbb{Q}(\sqrt{D}) \text{ und } \left(\frac{D}{p}\right) = 1, \tag{5.15}$$

$$U_{p^{n-1}(p+1)} \equiv 0 \mod p^n \quad \text{für } \left(\frac{D}{p}\right) = -1. \tag{5.16}$$

Die gewonnenen Aussagen lassen sich wie folgt zusammenfassen:

$$U_{p^{n-1}\left(p-\left(\frac{D}{p}\right)\right)} \equiv 0 \mod p^n \quad \text{für } \mathrm{ggT}(2QcD,p) = 1. \tag{5.17}$$

Die Beschränkung $\mathrm{ggT}(2QcD,p) = 1$ setzt sich aus (5.1), (5.7), (5.12) und aus der Beschränkung, daß p eine ungerade Zahl sein muß, zusammen.

5.3 Lucas-Lehmer-Test

Satz 11 (Lucas-Lehmer)**.** *Sei* $n+1 = \prod_{j=1}^{r} q_j^{\beta_j}$ *mit* q_j *paarweise verschiedenen Primzahlen. Sei* U_i *eine Lucas-Folge mit* $ggT(2QcD, n) = 1$ *und*

$$ggT(U_{\frac{n+1}{q_j}}, n) = 1 \quad \forall j = 1, 2, \ldots, r. \tag{5.18}$$

Ist nun

$$U_{n+1} \equiv 0 \mod n, \tag{5.19}$$

so ist n *prim.*

Bevor direkt zum Beweis übergegangen wird, hier noch eine Grundeigenschaft von Lucas-Folgen, welche zum Beweis des Satzes notwendig ist.

Lemma 6. *Sei* $ggT(Q, n) = 1$*. Es existiert ein* $d \geq 1$*, sodaß die Menge*

$$G = \{t \mid U_t \equiv 0 \mod n\}$$

die Form

$$G = d \cdot \mathbb{N}$$

hat.

Beweis: Der Beweis wird in zwei Schritten geführt, im ersten Schritt wird bewiesen, daß falls $U_p \equiv 0 \mod n$ ist, auch alle $U_{kp} \equiv 0 \mod n$ sind. Im zweiten Schritt wird bewiesen, daß als Indizes nur $d = \min_t\{U_t \equiv 0 \mod n, t > 0\}$ und dessen Vielfache $U_x \equiv 0 \mod n$ erfüllen.

1. Dieser Schritt läßt sich am einfachsten mit vollständiger Induktion beweisen. Es gelte $U_p \equiv 0 \mod n$.

   $$U_{2p} \equiv U_p \cdot V_p \equiv 0 \cdot V_p \equiv 0 \mod n$$

   $$U_{3p} \equiv U_{2p+p} \equiv U_{2p} \cdot V_p - Q^p U_p \equiv 0 \mod n$$

   Seien nun $U_{xp} \equiv U_{(x-1)p} \equiv 0 \mod n$:

   $$U_{(x+1)p} \equiv U_{xp+p} \equiv U_{xp} V_p - Q^p U_{xp-p} \equiv U_{xp} V_p - Q^p U_{(x-1)p} \equiv 0 \mod n.$$

   Die vollständige Induktion liefert damit, daß aus $U_p \equiv 0 \mod n$ für alle $k > 1$ die Aussage $U_{kp} \equiv 0 \mod n$ gilt.

2. Nur $d = \min_t\{U_t \equiv 0 \mod n, t > 0\}$ und deren Vielfache erfüllen $U_x \equiv 0 \mod n$.

   Angenommen $\exists t : U_t \equiv 0 \mod n$ mit $d \nmid t$. Sei

   $$r = \min_t\{U_t \equiv 0 \mod n \wedge d \nmid t\}.$$

   Wähle man nun $s > 0$ kleinstmöglich, sodaß $d \mid r + s$, also $r + s = kd$ für ein bestimmtes k, so gilt

   $$0 \equiv U_{kd} \equiv U_{r+s} \equiv U_r V_s - Q^s U_{r-s} \equiv 0 \cdot V_s - Q^s U_{r-s} \mod n.$$

   Ohne den Zwischenschritten bedeutet dies

   $$0 \equiv -Q^s U_{r-s} \mod n,$$

   woraus wegen $\mathrm{ggT}(Q, n) = 1$

   $$U_{r-s} \equiv 0 \mod n$$

   folgt.

   Da r die kleinste Zahl mit $U_r \equiv 0 \mod n$ und $d \nmid r$ ist, muß $d \mid r - s$ gelten. Durch die Wahl von s (dieses wurde kleinstmöglichst gewählt) gilt für dieses $0 < s < d$, deshalb muß $r - s = (k-1)d$ gelten, anders geschrieben $r = (k-1)d + s$.

   $$0 \equiv U_r \equiv U_{(k-1)d+s} \equiv U_{(k-1)d} V_s - Q^s U_{(k-1)d-s} \equiv 0 - Q^s U_{(k-1)d-s} \mod n$$

   Aus $\mathrm{ggT}(Q, n) = 1$ folgt wiederum

   $$U_{(k-1)d-s} \equiv 0 \mod n.$$

   $(k-1)d - s < r$ und $d \nmid (k-1)d - s$, was ein Widerspruch zur Annahme ist.

□

Beweis: (Lucas-Lehmer) Idee des Beweises ist, zu zeigen, daß falls n zusammengesetzt ist, ein $d < n + 1$ existiert, sodaß $U_d \equiv 0 \mod n$ erfüllt wird, wobei für primes n das erste Folgenglied der Lucas-Folge mit $U_d \equiv 0 \mod n$ genau $d = n + 1$ ist.

Die im Lemma 6 bewiesene Eigenschaft samt der Bedingung (5.18) sichert, daß U_{n+1} das erste Folgenglied kongruent 0 modulo n ist. Es wird nun gezeigt,

daß falls n eine zusammengesetzte Zahl ist, U_{n+1} entweder nicht kongruent 0 sein kann beziehungsweise $n+1$ nicht der erste Index mit $U_{n+1} \equiv 0 \mod n$ ist.

Sei $n = \prod_{i=1}^{m} p_i^{\alpha_i}$ mit p_i paarweise verschiedenen Primzahlen und $\mathrm{ggT}(2QcD, n) = 1$. Aus (5.18) bzw. (5.19) folgen

$$\mathrm{ggT}(U_{(N+1)/q_j}, p_i) = 1 \quad \forall i, j \tag{5.20}$$

und

$$U_{n+1} \equiv 0 \mod p_i \quad \forall i. \tag{5.21}$$

Wegen (5.17) gilt

$$U_{p_i^{\alpha_i - 1}\{p_i - (D/p_i)\}} \equiv 0 \mod p_i^{\alpha_i}.$$

Da für alle Primteiler von n ein n_i mit $U_{n_i} \equiv 0 \mod p_i^{\alpha_i}$ existiert und wie weiter oben gezeigt worden ist $\{t \mid U_t \equiv 0 \mod X\} = d \cdot \mathbb{N}$ gilt, existiert ein s mit

$$U_s \equiv 0 \mod p_i^{\alpha_i} \forall i. \tag{5.22}$$

Das kleinste s, das die Aussage (5.22) auf jeden erfüllt, ist klarerweise der kleinste gemeinsame Vielfache aller n_i, sprich

$$s = \mathrm{kgV}[n_i] = \mathrm{kgV}\left[p_i^{\alpha_i - 1}\left(p_i - \left(\frac{D}{p_i}\right)\right)\right]. \tag{5.23}$$

Nun gilt es eine Abschätzung für s zu erreichen und zu zeigen, daß s bei zusammengesetzten Zahlen stets kleiner als $n+1$ ist.

Ist $n = p_1^{\alpha_1}$ ein Primpotenz, so ist

$$\begin{aligned} s = \mathrm{kgV}[n_i] &= \mathrm{kgV}\left[p_i^{\alpha_i - 1}\left(p_i - \left(\tfrac{D}{p_i}\right)\right)\right] \\ &= p_1^{\alpha_1 - 1}\left(p_1 - \left(\tfrac{D}{p_1}\right)\right) \\ &= p_1^{\alpha_1} \pm p_1^{\alpha_1 - 1} \\ &= n \pm p_1^{\alpha_1 - 1}. \end{aligned}$$

Für den Fall $s = n - p_1^{\alpha_1 - 1}$ ist $s < n+1$ und U_{n+1} ist nicht das erste Folgenglied mit $U_i \equiv 0 \mod n$, womit dieser Fall erledigt ist.

Für den Fall $s = n + p_1^{\alpha_1 - 1}$ ist sowohl $U_s \equiv 0 \mod n$ als auch laut Voraussetzung des Satzes $U_{n+1} \equiv 0 \mod n$, und wegen $\{t \mid U_t \equiv 0 \mod n\} = d \cdot \mathbb{N}$ und da $n+1$ das erste Folgenglied mit $U_i \equiv 0 \mod n$ ist, muß $n+1 \mid n + p_1^{\alpha_1 - 1}$ gelten. Da $n + p_1^{\alpha_1 - 1} < 2(n+1)$ ist, gilt diese Eigenschaft nur für $\alpha_1 = 1$, was ein Widerspruch dazu wäre, daß n eine Primpotenz ist.

Besteht nun n aus verschiedenen Primfaktoren, so ist eine weitere Abschätzung notwendig.

Wegen $(2QcD, n) = 1$ sind alle p_i ungerade und damit $p_i - \left(\frac{D}{p_i}\right)$ gerade.

$$\begin{aligned}
s &= 2 \cdot \mathrm{kgV}\left[p_i^{\alpha_i - 1} \frac{p_i - \left(\frac{D}{p_i}\right)}{2}\right] \\
&\leq 2 \prod \frac{1 - \frac{1}{p_i}\left(\frac{D}{p_i}\right)}{2} p_i^{\alpha_i} \\
&\leq 2n \prod \tfrac{1}{2}\left(1 + \tfrac{1}{p_i}\right) \\
&= 2n \tfrac{1}{2}\left(1 + \tfrac{1}{p_1}\right) \tfrac{1}{2}\left(1 + \tfrac{1}{p_2}\right) \prod_{i>2} \tfrac{1}{2}\left(1 + \tfrac{1}{p_i}\right) \\
&\leq \tfrac{1}{2} n \left(1 + \tfrac{1}{3}\right)\left(1 + \tfrac{1}{5}\right) \\
&= \tfrac{4}{3}\tfrac{6}{5}\tfrac{1}{2} n = \tfrac{4}{5} n \\
&< n + 1.
\end{aligned}$$

Damit kann ein n, welches die Bedingungen (5.18) und (5.19) erfüllt, nur prim sein. □

Bemerkung. *Für die Verwendung dieses Satzes als Test auf Primalität ist die Bedingung $ggT(2QcD, n) = 1$ keine allzu große Einschränkung, denn wenn der größte gemeinsame Teiler nicht 1 ist, ist ein Teiler von n gefunden worden. (Ausgenommen n teilt eine der Größen Q, c oder D. Jedoch sind diese Größen meist klein gewählt, und diese Tests auf Primalität sind für relativ große n interessant.)*

5.4 Lucas-Lehmer-Test für Mersennezahlen

Die Schwierigkeit besteht darin, die Primfaktoren von $n+1$ und die entsprechende Lucas-Folge zu finden. Die Mersenne-Zahlen $M_m = 2^m - 1$ haben die Eigenschaft, daß $n + 1$ in diesem Fall eine Potenz von 2 ist. Damit ist die Zerlegung von $n + 1$ offensichtlich. Kann eine Lucas-Folge mit

$$U_{n+1} \equiv 0 \mod n \tag{5.24}$$

bei

$$U_{\frac{n+1}{2}} \not\equiv 0 \mod n \tag{5.25}$$

gefunden werden, wobei $n = M_m = 2^m - 1$ ist, so ist n prim.

Lemma 7. *U_k und V_k haben höchstens $2Q^k$ als gemeinsamen Teiler, sprich*

$$ggT(U_k, V_k) \mid 2Q^k. \tag{5.26}$$

Beweis: Bedingung (5.26) gilt genau dann, wenn eine Linearkombination mit

$$V_k y + U_k x = 2Q^k$$

existiert.

$$\begin{aligned} V_k U_{k+1} - U_k V_{k+1} &= \left(a^k + b^k\right) \tfrac{a^{k+1}-b^{k+1}}{a-b} - \tfrac{a^k-b^k}{a-b}\left(a^{k+1} + b^{k+1}\right) \\ &= \tfrac{a^{2k+1}-a^k b^{k+1}+b^k a^{k+1}-b^{2k+1}}{a-b} - \tfrac{a^{2k+1}+a^k b^{k+1}-b^k a^{k+1}-b^{2k+1}}{a-b} \\ &= \tfrac{1}{a-b} \cdot \left(-a^k b^{k+1} + b^k a^{k+1} - a^k b^{k+1} + b^k a^{k+1}\right) \\ &= \tfrac{a^k b^k}{a-b} \cdot (2a - 2b) = 2Q^k, \end{aligned}$$

womit das Lemma bewiesen ist. □

Die zwei Bedingungen (5.24) und (5.25) können zu einer zusammengefasst werden. Es gilt

$$U_{n+1} = U_{\frac{n+1}{2}} V_{\frac{n+1}{2}},$$

womit $V_{\frac{n+1}{2}} \equiv 0 \mod n$ sein muß. Durch die allgemeine Bedingung $\mathrm{ggT}(2QcD, n) = 1$ an Lucas-Folgen für Primzahltests und Lemma 7 ist auch klar, daß wenn $V_{\frac{n+1}{2}}$ den Faktor n erhält, $U_{\frac{n+1}{2}}$ ungleich 0 modulo n sein muß. Es genügt demnach, $V_{\frac{n+1}{2}} = V_{2^{m-1}} \equiv 0 \mod n$ zu überprüfen.

Die Notation wird in eine leserlichere Form gebracht:

$$V_{2^s} = v_s.$$

Die Berechnung $V_{2^s} = V_{2^{s-1}}^2 - 2Q^{2^{s-1}}$ hat dadurch die Form

$$v_s = v_{s-1}^2 - 2Q^{2^{s-1}}.$$

Begonnen wird die Rekursion mit $v_0 = V_1 = P$ und die Frage bezüglich der Primalität von M_m läßt sich wie folgt formulieren:

$$v_{m-1} \equiv 0 \mod M_m?$$

Eine entsprechende Lucas-Folge läßt sich auch finden, bezüglich der Berechnung ist es hilfreich, wenn die Werte P und Q möglichst klein sind. Es ist leicht zu zeigen, daß die Werte $a = 1 + \sqrt{3}$ und $b = 1 - \sqrt{3}$ eine passende Lucas-Folge ergeben, dies führt zu

$$P = 2, Q = -2, \quad P^2 - 4Q = 12, \quad D = 3, \quad c = 2.$$

Dadurch schaut die Rekursion wie folgt aus:

$$v_s = v_{s-1}^2 - 2 \cdot 2^{2^{s-1}}$$

mit dem Startwert

$$v_1 = v_0^2 + 4 = 8.$$

Um die Berechnung einfacher zu machen, wäre es wünschenswert, $Q = 1$ oder $Q = -1$ zu haben, da dadurch das Potenzieren von 2 in jedem Berechnungsschritt wegfallen würde. Leider ist keine Lucas-Folge zu finden, die mit dieser gewünschten Zusatzbedingung alle anderen Bedingungen auch erfüllt. Dieses Problem wurde aber durch Einfügen weiterer Folgenglieder in der Lucas-Folge gelöst. Die Details der Beweisführung, daß die Folge noch immer als Primzahltest benutzt werden kann, ist kompliziert, jedoch wurden seither schon einfachere Beweise gefunden. Auf diese soll hier eingegangen werden.

5.5 Verbesserter Lucas-Lehmer-Test für Mersennezahlen

Sei die Folge v_i wie folgt definiert:

$$v_i = v_{i-1}^2 - 2$$

mit dem Startwert

$$v_0 = 4.$$

Satz 12. *M_p ist dann und nur dann prim, falls p eine ungerade Primzahl[2] ist und*

$$v_{p-2} \equiv 0 \mod M_p.$$

Beweis: Daß nur prime p-Werte in Frage kommen, wird durch

$$2^{mn} - 1 = (2^m - 1)(2^{(n-1)m} + 2^{(n-2)m} + \cdots + 2^{2m} + 2^m + 1)$$

klar.

Die Differenzengleichung v_i hat die geschlossene Form

$$v_i = x^{2^i} + \overline{x}^{2^i}$$

mit

$$x = 2 + \sqrt{3}, \text{ und daher } \overline{x} = 2 - \sqrt{3}.$$

[2]Für $p = 2$ ist $M_2 = 3$ und damit eine Primzahl

Wohl am einfachsten kann man die Richtigkeit der geschlossenen Form durch vollständige Induktion zeigen. Dazu, und auch in einigen Schritten des Beweises weiter unten, brauchen wir das Produkt $x\overline{x}$.

$$x\overline{x} = (2+\sqrt{3})(2-\sqrt{3}) = 2^2 - \sqrt{3}^2 = 1$$

$i = 0$:

$$v_0 = x^{2^0} + \overline{x}^{2^0} = (2+\sqrt{3}) + (2-\sqrt{3}) = 4$$

Sei nun $v_i = x^{2^i} + \overline{x}^{2^i}$

$$\begin{aligned} v_{i+1} &= v_i^2 - 2 = (x^{2^i} + \overline{x}^{2^i})^2 - 2 \\ &= (x^{2^i})^2 + 2x^{2^i}\overline{x}^{2^i} + (\overline{x}^{2^i})^2 - 2 \\ &= x^{2^{i+1}} + 2(x\overline{x})^{2^i} + \overline{x}^{2^{i+1}} - 2 \\ &= x^{2^{i+1}} + 2(1)^{2^i} + \overline{x}^{2^{i+1}} - 2 \\ &= x^{2^{i+1}} + \overline{x}^{2^{i+1}} \end{aligned}$$

- Aus $v_{p-2} \equiv 0 \mod M_p$ folgt: M_p ist prim. Dieser Beweisschritt lehnt sich an die Arbeit [23] an.

  Sei nun q ein echter Teiler von M_p. Da M_p stets ungerade ist, ist $q \geq 3$. Sei nun weiters

  $$X = \{a + b\sqrt{3} \mid a, b \in \mathbb{Z}_q\}.$$

  X hat q^2 Elemente. Auf diese Gruppe läßt sich auch eine naheliegende Multiplikation definieren:

  $$(a+b\sqrt{3})(b+c\sqrt{3}) = [(ac+3bd \mod q)] + [bc+ad \mod q]\sqrt{3}.$$

  Bezüglich dieser Multiplikation ist X keine Gruppe, da 0 kein inverses Element hat. Sei nun

  $$X^* = \{x \in X \mid \exists y \in X : xy = 1\}.$$

  X und X^* unterscheiden sich zumindest um das Element 0, somit hat X^* maximal $q^2 - 1$ Elemente.

  Aus $v_{p-2} \equiv 0 \mod M_p$ folgt

  $$\begin{aligned} x^{2^{p-2}} + \overline{x}^{2^{p-2}} &= kM_p \text{ für irgendein } k \\ x^{2^{p-2}} &= kM_p - \overline{x}^{2^{p-2}} \quad | \cdot x^{2^{p-2}} \\ (x^{2^{p-2}})^2 &= kM_p x^{2^{p-2}} - (x\overline{x})^{2^{p-2}} \end{aligned}$$

$$x^{2^{p-1}} = kM_p x^{2^{p-2}} - 1. \tag{5.27}$$

Da $x \in X$ und q ein Teiler von M_p, sprich $M_p \equiv 0 \mod q$, ist, ist

$$kM_p x^{2^{p-2}} = 0 \text{ in } X.$$

Dies in (5.27) eingesetzt ergibt

$$x^{2^{p-1}} = -1 \text{ in } X \quad |^2 \tag{5.28}$$

$$x^{2^p} = 1$$

Daraus ist ersichtlich, daß x ein inverses Element, nämlich $x^{2^{p-1}}$, in X hat, und somit $x \in X^*$. Des Weiteren ist daraus ersichtlich, daß die Ordnung von x ein Teiler von 2^p und daher eine Zweierpotenz sein muß. Aus (5.28) ist ersichtlich, daß die Ordnung größer als 2^{p-1} sein muß, und daraus folgt, daß die Ordnung von x genau 2^p ist.

Die Ordnung einer Gruppe ist mindestens so groß wie die Ordnung eines Elementes, daher gilt für X^*

$$2^p \leq q^2 - 1 < q^2. \tag{5.29}$$

q ist laut Annahme ein nichttrivialer Teiler von M_p, woraus

$$q^2 \leq M_p = 2^p - 1 \tag{5.30}$$

folgt.

(5.29) und (5.30) zusammengefügt ergibt dies

$$2^p \leq q^2 - 1 < q^2 \leq M_p = 2^p - 1 \tag{5.31}$$

$$2^p \leq 2^p - 1, \tag{5.32}$$

womit bewiesen ist, daß unter der Annahme $v_{p-2} \equiv 0 \mod M_p$ kein echter Teiler von M_p existieren kann.

- Aus M_p prim folgt $v_{p-2} \equiv 0 \mod M_p$. Dieser zweite Teil des Beweises ist ein Sonderfall des Beweises in [24].

  Hierzu ist es zuersteinmal notwendig, festzustellen, ob M_p bezüglich den Modulen 2 und 3 ein quadratischer Rest bzw. Nichtrest ist.

  Für ungerade Primzahlen q gilt[3]:

$$\left(\frac{3}{q}\right) = \begin{cases} +1 & \text{für } q = 1 \text{ oder } 11 \mod 12 \\ -1 & \text{für } q = 5 \text{ oder } 7 \mod 12 \end{cases}$$

[3] siehe Anhang B

$2^n \mod 12$ ist für ungerade Potenzen ($n = 1$ ausgenommen) 8 modulo 12, da $2^3 = 8$ und falls $2^n \equiv 8 \mod 12$ ist, so folgt $2^{n+2} = 2^n \cdot 2^2 \equiv 8 \cdot 4 \equiv 32 \equiv 8 \mod 12$. Damit gilt für ungerade Primzahlen p

$$M_p \equiv 2^p - 1 \equiv 8 - 1 \equiv 7 \mod 12,$$

woraus

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

folgt. Eulers Kriterium liefert aus diesem Ergebnis

$$3^{\frac{M_p-1}{2}} \equiv -1 \mod M_p.$$

2 ist ein quadratischer Rest modulo M_p, da aus der Definition von M_p

$$2^p \equiv 1 \mod M_p$$

und in weiterer Folge

$$2 \equiv 2^{p+1} = \left(2^{\frac{p+1}{2}}\right)^2$$

folgt, womit 2 ein quadratischer Rest ist. Diese Tatsache in Eulers Kriterium eingebettet ergibt

$$2^{\frac{M_p-1}{2}} \equiv 1 \mod M_p.$$

Ähnlich wie im ersten Teil des Beweises sei nun

$$X = \{a + b\sqrt{3} \mid a, b \in \mathbb{Z}_{M_p}\}$$

mit der Multiplikation

$$(a + b\sqrt{3})(b + c\sqrt{3}) = [(ac + 3bd \mod M_p)] + [bc + ad \mod M_p]\sqrt{3}.$$

Im Folgenden werden zwei Äquivalenzen erwähnt, die das Rechnen in X erleichtern und angewendet werden:

$$a^{M_p} \equiv a \mod M_p \text{ (Satz v. Fermat)}$$

$$(a + b)^{M_p} \equiv a^{M_p} + b^{M_p} \mod M_p,$$

da in den restlichen Termen in den Binomialkoeffizienten der Faktor M_p stets erhalten bleibt und diese durch die Modulorechnung wegfallen.

$x = 2 + \sqrt{3}$ in X und in X gilt:

$$
\begin{aligned}
x^{\frac{M_p+1}{2}} &= \left(2+\sqrt{3}\right)^{\frac{M_p+1}{2}} \\
&= \left(\tfrac{48+24\sqrt{3}}{24}\right)^{\frac{M_p+1}{2}} \\
&= \left(\tfrac{36+2*6*2\sqrt{3}+12}{24}\right)^{\frac{M_p+1}{2}} \\
&= \left(\tfrac{\left(6+2\sqrt{3}\right)^2}{24}\right)^{\frac{M_p+1}{2}} \\
&= \frac{\left(6+2\sqrt{3}\right)^{M_p+1}}{24^{\frac{M_p+1}{2}}} \\
&= \frac{\left(6+2\sqrt{3}\right)^{M_p}\left(6+2\sqrt{3}\right)}{24\cdot 24^{\frac{M_p-1}{2}}} \\
&= \frac{\left(6^{M_p}+2^{M_p}\sqrt{3}^{M_p}\right)\left(6+2\sqrt{3}\right)}{24\cdot(2^3\cdot 3)^{\frac{M_p-1}{2}}} \\
&= \frac{\left(6+2\cdot 3^{\frac{M_p-1}{2}}\sqrt{3}\right)\left(6+2\sqrt{3}\right)}{24\left(2^{\frac{M_p-1}{2}}\right)^3\left(3^{\frac{M_p-1}{2}}\right)} \\
&= \frac{\left(6+2(-1)\sqrt{3}\right)\left(6+2\sqrt{3}\right)}{24(1)^3(-1)} \\
&= \frac{\left(6-2\sqrt{3}\right)\left(6+2\sqrt{3}\right)}{-24} \\
&= \tfrac{24}{-24} = -1
\end{aligned}
$$

$M_p+1 = 2^p$ und da $p > 2$ ist, ist diese Größe stets durch 4 teilbar, wodurch stets

$$\overline{x}^{(M_p+1)/4} \in X.$$

Beide Seiten der obigen Gleichung werden mit diesem Wert multipliziert:

$$
\begin{aligned}
x^{\frac{M_p+1}{2}} \cdot \overline{x}^{\frac{M_p+1}{4}} &= -\overline{x}^{\frac{M_p+1}{4}} \\
x^{\frac{M_p+1}{2}} \cdot \overline{x}^{\frac{M_p+1}{4}} + \overline{x}^{\frac{M_p+1}{4}} &= 0 \\
(x\overline{x})^{\frac{M_p+1}{4}} \cdot x^{\frac{M_p+1}{4}} + \overline{x}^{\frac{M_p+1}{4}} &= 0 \\
1 \cdot x^{\frac{2^p-1+1}{4}} + \overline{x}^{\frac{2^p-1+1}{4}} &= 0 \\
x^{2^{p-2}} + \overline{x}^{2^{p-2}} &= 0 \\
v_{p-2} &= 0
\end{aligned}
$$

$v_{p-2} = 0$ in X bedeutet

$$v_{p-2} \equiv 0 \mod M_p.$$

Damit ist der Satz bewiesen. □

5.6 The Great Internet Mersenne Prime Search (GIMPS)

Der Lucas-Lehmer-Test ist zwar nur auf die Mersennezahlen spezialisiert, für diese funktioniert der Test aber sehr schnell. Des Weiteren vermutet man, daß unter diesen Zahlen die Wahrscheinlichkeit (verglichen mit anderen Zahlen ähnlicher Größe) „hoch“ ist, daß eine Mersennezahl prim ist. Über die Häufigkeit von Mersenne-Primzahlen erschienen unter anderem die Arbeiten [25], [26] und [27]. Dies ist der Grund, wieso gerade diese Zahlen auf der Suche nach noch größeren Primzahlen immer wieder systematisch untersucht worden sind und oft eine Mersenne-Primzahl die größte bekannte Primzahl war.

Auch heute führen Mersenne-Primzahlen die Liste der größten bekannten Primzahlen an, zur Zeit sind die 6 größten bekannten Primzahlen von der Form $2^p - 1$. (Eine Liste der 100 höchsten bekannten Primzahlen kann man in [8] einsehen.) Lange Zeit verfolgte man die Strategie, möglichst leistungsstarke Computer mit der Primzahlsuche zu beschäftigen, jedoch gewann in letzter Zeit die Vernetzung von „einfachen“ Computern zu einem großen Cluster immer mehr an Bedeutung.

Die Prozessoren der meisten Heimrechner sind nicht ausgelastet, nur die wenigsten beanspruchen den Prozessor mit rechenintensiven Programmen, während z.B. eMails Schreiben kaum Rechenressourcen verbraucht. Jede Person kann freiwillig die nicht verbrauchte Prozessorleistung des eigenen Rechners zur Verfügung stellen. So rechnet der eigene Rechner in Zeiten, wo der Benützer selbst die Prozessorleistung nicht beziehungsweise nicht voll braucht, an der Rechenaufgabe des Clusters. (Dabei läuft das Programm mit der niedrigsten Priorität, damit andere Programme des Users nicht benachteiligt werden und so der User vom Programm nicht in den alltäglichen Arbeiten behindert wird.)

Das GIMPS-Projekt ist eines der Projekte, die sich die Überprüfung der Mersenne-Zahlen auf ihre Primalität zur Aufgabe gemacht haben. Das Programm hinter GIMPS heißt Prime95/MPrime, das Programm gibt es für verschiedene Plattformen (Windows, Unix, ...). Jeder kann das Programm auf seinem Computer installieren, via Internet werden bestimmte prime p-Werte vergeben und das Programm führt für M_p einen Lucas-Lehmer-Test durch und meldet am Projekt das Ergebnis der Berechnung. Grundsätzlich wird nur ein Lucas-Lehmer-Test wie vorgestellt und bewiesen verwendet. Um eventuelle langwierige unnötige Arbeiten zu vermeiden, werden vor der Überprüfung mit dem Lucas-Lehmer-Test etliche kleine Primfaktoren als Teiler ausgeschlossen.

Das Programm wurde im Jänner des Jahres 1996 gestartet, in den ersten 2 Monaten beteiligten sich lediglich 50 Computer an der Suche. Im November 1996 wurde so die erste Mersenne-Primzahl mit GIMPS gefunden. Insgesamt wurden so bis heute 10 Mersenne-Primzahlen gefunden, alle zur Zeit der Berechnung die damals größte bekannte Primzahl. Die letzte Primzahl wurde im September des Jahres 2006 mit 9,8 Millionen Stellen (Dezimaldarstellung) gefunden. Zur Zeit sind auf der Homepage des Projektes 53186 Personen gelistet, die bis jetzt dem Projekt Rechenkapazität zur Verfügung gestellt haben.

Kapitel 6

Primzahltests basierend auf elliptischen Kurven

6.1 Elliptische Kurven

In diesem Kapitel liegt das Hauptaugenmerk auf einen Primzahltest basierend auf elliptische Kurven, vorgestellt von Goldwasser und Kilian in [28]. Hierzu braucht man etliche Eigenschaften von elliptischen Kurven, über die zunächst ein kurzer Überblick gegeben wird.

Definition (Ebene affine Kurve)**.** *Sei $\mathbb{F}$ ein beliebiger Körper und $p(x, y)$ ein Polynom in zwei Variablen, so heißt*

$$C = \{(x, y) \in \mathbb{F}^2 \mid p(x, y) = 0\}$$

eine ebene affine Kurve.

Definition (Ebene projektive Kurve)**.** *Sei $\mathbb{F}$ ein beliebiger Körper und $p(X, Y, Z)$ ein homogenes*[1] *Polynom von Grad d in drei Variablen, so heißt*

$$C = \{(X, Y, Z) \in \mathbb{F}^3 \mid p(X, Y, Z) = 0\}$$

eine ebene projektive Kurve.

Dadurch, daß $p(X, Y, Z)$ ein homogenes Polynom von Grad d ist, gilt

$$p(tX, tY, tZ) = t^d \cdot p(X, Y, Z).$$

[1] Die Summe der Exponenten der einzelnen Termen ist stets gleich.

Damit können verschiedene Polynome die gleiche ebene projektive Kurve darstellen. Durch die Transformation

$$X \to \frac{X}{Z}, \quad Y \to \frac{Y}{Z}, \quad Z \to \frac{Z}{Z} = 1$$

wird die Schar der Polynome, die die gleiche ebene projektive Kurve darstellen, auf die gleiche Form zusammengefasst:

$$p(X, Y, Z) \to p(\frac{X}{Z}, \frac{Y}{Z}, 1)$$

Das Polynom $p(\frac{X}{Z}, \frac{Y}{Z}, 1) = p(x_1, y_1)$ ist nur mehr ein Polynom in 2 Variablen und repräsentiert dadurch eine ebene affine Kurve. $p(x_1, y_1)$ heißt das affine Polynom zu $p(X, Y, Z)$. Zu jeder projektiven ebenen Kurve gehört so eine affine ebene Kurve. Die projektive ebene Kurve kann durch $Z = 0$ zusätzliche Lösungen aufweisen, wodurch die Punkte im Unendlichen in der affinen Ebene besser behandeln lassen. Für eine tiefergehende Behandlung von ebenen affinen und projektiven Kurven siehe z.B. [29].

Definition (Nichtsinguläre Kurve)**.** *Eine von von $f(x, y)$ erzeugte Kurve wird als nichtsingulär im Punkt (x_1, y_1) bezeichnet, wenn der Punkt auf f liegt und beide Ableitungen von f an der Stelle (x_1, y_1) nicht gleichzeitig gleich 0 sind. Eine Kurve heißt nichtsingulär, wenn für alle Punkte (x_1, y_1) aus dem algebraischen Abschluss des betrachteten Körpers nichtsingulär sind.*

Definition (Elliptische Kurve)**.** *Eine singularitätsfreie projektive ebene Kurve erzeugt vom Polynom*

$$f(X, Y, Z) = Y^2Z + a_1XYZ + a_3YZ^2 - X^3 - a_2X^2Z - a_4XZ^2 - a_6Z^3 \quad (6.1)$$

heißt elliptische Kurve. (Die nicht durchgehende Nummerierung der Koeffizienten hat historische Gründe und wurden hier so übernommen.)

Die Kurve kann auf beliebige Körper $\mathbb{F}$ definiert werden, um die Kurve auch graphisch anschaulich interpretieren zu können wird zuerst der Fall $\mathbb{F} = \mathbb{R}$ betrachtet, für die Anwendung der elliptischen Kurven für einen Test auf Primalität wird der Körper $\mathbb{F} = \mathbb{Z}_p$ betrachtet.

Da die Gleichung (6.1) vom homogenen Grad 3 ist, läßt sich durch Division von Z^3 und danach durch die Transformationen $\frac{X}{Z} \to x$ und $\frac{Y}{Z} \to y$ in die affine Form

$$f(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 \quad (6.2)$$

bringen.

Diese Form kann durch

$$(x, y) \to (x, y - \frac{a_1}{2}x - \frac{a_3}{2})$$

weiter transformiert werden und man erhält die Form

$$f(x,y) = y^2 - x^3 - b_2x^2 - b_4x - b_6.$$

Durch die weitere Umformung

$$(x,y) \to (\frac{x - 3b_2}{3^2}, \frac{y}{3^3})$$

und eine abschließende Multiplikation mit 3^6 erhält man die Form

$$f(x,y) = y^2 - x^3 - ax - b.$$

Mit dieser Form elliptischer Kurven wird in der Folge gearbeitet, die dazugehörige projektive Form ist

$$f(X,Y,Z) = Y^2Z - X^3 - aXZ^2 - bZ^3.$$

Im Folgenden werden nun die Punktemengen (x,y) bzw. die Punktemengen (X,Y,Z) betrachtet, die die Gleichung

$$y^2 = x^3 + ax + b \tag{6.3}$$

beziehungsweise

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \tag{6.4}$$

erfüllen. Um Verwechslungen zu vermeiden werden Koordinaten in der affinen stets mit Kleinbuchstaben, in der projektiven Ebene mit Großbuchstaben referenziert.

Hierzu soll angemerkt werden, daß diese Umformungen klarerweise die Existenz der Zahlen 2 und 3 im entsprechenden Körper voraussetzen (da durch diese dividiert wird), die Charakteristik von $\mathbb{F}$ darf also nicht 2 oder 3 sein.

Die Nichtsingularität einer Kurve wird für die Anwendung von Wichtigkeit sein, da man an der Kurve in jedem Punkt eine eindeutige Tangente legen können muß, und dies wird durch diese Eigenschaft sichergestellt. Die partielle Ableitung nach y ist $2y$, und dies kann nur für $y = 0$ verschwinden. Somit kann eine elliptische Kurve nur in Punkten der Form $(x,0)$ singulär sein, wobei sowohl $x^3 + ax + b = 0$ als auch $(x^3 + ax + b)' = 0$ sein müssen. Aus der 2. Bedingung kann man x ausdrücken und diese dann in die erste Bedingung einsetzen, damit erhält man als Voraussetzung für die Nichtsingularität der elliptischen Kurve

$$4a^3 + 27b^2 \neq 0.$$

(Die Bedingung $g(x) = g'(x) = 0$ bedeutet, daß die Kurve in so einem Punkt eine doppelte Nullstelle hat. Wäre im Falle einer elliptischen Kurve $4a^3+27b^2 = 0$, so wäre $\frac{\sqrt{-3a}}{3}$ bzw. $-\frac{\sqrt{-3a}}{3}$ eine doppelte Nullstelle.)

Was ist nun genau der Unterschied zwischen der Betrachtung der affinen und projektiven elliptischen Kurve?

Ist (x, y) Lösung von (6.3), so sind für $t \neq 0$ alle (tx, ty, t) Lösungen von (6.4). Alle Lösungen von (6.4) der Form (tx, ty, t) mit $t \neq 0$ lassen sich also auf ein und dieselbe Lösung transformieren, der Vektor $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ repräsentiert den Punkt (x, y) in der projektiven Sichtweise. Es bleibt der Fall $Z = 0$ zu betrachten, man erhält

$$0 = X^3. \tag{6.5}$$

Diese Gleichung wird für Y beliebig und $X = 0$ erfüllt, die Lösung wird also repräsentiert durch den Vektor $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$. Diese Lösung entspricht aber nicht einer Lösung der Gleichung (6.3), wir bezeichnen diesen Punkt als Lösung im Unendlichen (Richtung y), in Zeichen $\mathcal{O}$.

Im Folgenden werden die Mengen

$$E_{a,b}(\mathbb{F}) = \{(x, y) \in \mathbb{F} \times \mathbb{F} \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\} \tag{6.6}$$

bzw.

$$E_{a,b}(\mathbb{F}) = \{(X, Y, Z) \in \mathbb{F} \times \mathbb{F} \mid (Y^2 = X^3 + aX + b, Z = 1) \vee (X = Z = 0, Y = 1)\} \tag{6.7}$$

mit $\mathbb{F}$ ein Körper verwendet, wobei auf die Angabe des Index und des Körpers bei Klarheit bzw. bei allgemeinen Formulierungen oft verzichtet wird.

Lemma 8. *Nimmt man nun zwei Punkte auf einer elliptischen Kurve in $\mathbb{F}$ und legt eine Gerade durch diese, so schneidet die Gerade die elliptische Kurve in einem weiteren dritten Punkt.*

Beweis: Sei die Gerade gegeben mit $y = kx + d$, damit wären alle möglichen Geraden bis auf die Geraden parallel zur y-Achse abgedeckt. Diese Gerade in (6.3) eingesetzt ergibt

$$\begin{aligned} (kx + d)^2 &= x^3 + ax + b \\ k^2x^2 + 2kdx + d^2 &= x^3 + ax + b \\ 0 &= x^3 - k^2x^2 + (a - 2kd)x + b - (d^2) \end{aligned} \tag{6.8}$$

Die Gerade $y = kx + d$ schneidet also die elliptische Punkte bei den x-Werten, welche die Gleichung (6.8) erfüllen. Eine Kurve dritten Grades hat aber genau eine oder drei Nullstellen (mehrfache Nullstellen auch mehrfach gezählt). Da aber laut Voraussetzung zumindest zwei Schnittpunkte gibt, hat (6.8) zumindest zwei Lösungen, und damit auch eine dritte. Anders formuliert, die Gerade schneidet die elliptische Kurve in einem weiteren dritten Punkt. □

Die Gerade parallel zur y-Achse hat wegen der Symmetrie einer Kurve der Form (6.3) bezüglich der x-Achse nur zwei (falls überhaupt) Schnittpunkte, nämlich (x_1, y_1) und $(x_1, -y_1)$. Hier muß man einen „Kunstgriff“ anwenden und wieder über die projektive Darstellung gehen. Im projektiven Raum entsprechen diese Lösungen den Vektoren $v_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$ und $v_2 = \begin{pmatrix} x_1 \\ -y_1 \\ 1 \end{pmatrix}$, die Gerade im projektiven Raum also $g = \lambda \cdot v_1 + \mu \cdot v_2$. Durch die Wahl $\lambda = \frac{1}{2y_1}$ und $\mu = -\frac{1}{2y_1}$ erhält man, daß auch $(0, 1, 0)$ auf der Geraden liegt, also definieren wir in diesem Fall im affinen den Punkt $\mathcal{O}$ als dritten Schnittpunkt. (Anschaulich gesehen wird die Steigung einer elliptischen Kurve immer größer, asymptotisch geht die Steigung nach unendlich, und damit ist sie parallel zu der y-Achse und damit auch zu der Gerade, und zwei parallele Geraden schneiden sich im Unendlichen.)

Damit gerüstet läßt sich auf der Menge E eine Addition definieren:

Definition (Punktaddition (auf elliptischen Kurven))**.** *Seien P und Q zwei Elemente aus E. So läßt sich der Punkt $R = P + Q$ wie folgt konstruieren: Man lege eine Gerade durch P und Q und ermittle den dritten Schnittpunkt mit der elliptischen Kurve und spiegle diesen Punkt an der x-Achse. So erhält man den Punkt R, welcher (aus Symmetriegründen) wieder auf der elliptischen Kurve liegt.*

Der Zwischenpunkt, also der dritte Punkt auf der von P und Q erzeugten Geraden wird mit $-R$ bezeichnet. Ist $P = Q$, dann ist die Gerade die Tangente an die Kurve in diesem Punkt.

Diese Definition ist zwar anschaulich, jedoch zum Rechnen ungeeignet. Die oben angegebene Additionskonstruktion läßt sich wie folgt in Formeln gießen, für die Herleitung der Formeln siehe z.B. [30].

Sei $P = (x_1, y_1)$ und $Q = (x_2, y_2)$. Die Koordinaten (x_3, y_3) von $R = P + Q$ lassen sich falls $x_1 \neq x_2$ wie folgt berechnen:

$$x_3 = k^2 - x_1 - x_2 \quad \text{und} \quad y_3 = k(x_1 - x_3) - y_1,$$

wobei k die Steigung der Verbindungsgeraden P und Q ist und des gilt

$$k = \frac{y_1 - y_2}{x_1 - x_2}.$$

Gilt hingegen $P = Q$ und $y_1 = y_2 \neq 0$, so lautet die Berechnung für $P + Q = 2P = R$

$$x_3 = k^2 - 2x_1 \quad \text{und} \quad y_3 = k(x_1 - x_3) - y_1,$$

wobei in diesem Fall k die Steigung der Tangente im Punkt P ist und berechnet

$$k = \frac{3x_1^2 + a}{2y_1}$$

ergibt.

Für die Fälle $P = Q$ mit $y_1 = y_2 = 0$ beziehungsweise $P \neq Q$ mit $x_1 = x_2$ (woraus $y_1 = -y_2$ folgt) ist die Tangente bzw. Verbindungsgerade der Punkte parallel zur y-Achse und es gilt

$$P + Q = \mathcal{O}.$$

Des Weiteren wird das n-malige Addieren von P zu sich selbst $(P+P+\ldots P)$ mit nP abgekürzt. Diese Größe läßt sich durch wiederholtes Verdoppeln leicht berechnen, hierzu ist es notwendig, daß bezüglich der Addition das Assoziativgesetz gilt, hierzu siehe [30].

$$nP = \begin{cases} P & \text{für } n = 1 \\ (P+P) \cdot \frac{n}{2} & \text{für gerade } n \\ P + (n-1)P & \text{für ungerade } n \end{cases}$$

$E(\mathbb{F})$ bildet mit dieser Addition eine abelsche Gruppe. Wichtig dabei ist, daß die Addition in der Gruppe abgeschlossen ist und $P+Q$ stets existiert. Dies wird sichergestellt, da $\mathbb{F}$ ein Körper ist und deswegen die Steigung der Geraden wegen der Existenz eines Inversen stets berechenbar ist.

Würde man für die elliptische Kurve keinen Körper verwenden, so gäbe es Definitionslücken, und nicht für alle Punkte P und Q wäre $P + Q$ definiert.

Definition (Elliptische Pseudokurve)**.** *Seien $a, b \in \mathbb{Z}_n$ und erfüllen diese Zahlen die Bedingungen $ggT(a, 6) = 1$ und $ggT(4a^3 + 27b^2, n) = 1$, so heißt*

$$E_{a,b}(\mathbb{Z}_n) = \{(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_n \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

elliptische Pseudokurve über $\mathbb{Z}_n$.

Ist n eine Primzahl, so ist $\mathbb{Z}_n$ ein Körper und die Addition ist abgeschlossen. Ist n zusammengesetzt, so gilt wie bereits erwähnt die Abgeschlossenheit nicht mehr. Bildet man für eine Zahl n die Punktemenge $E(\mathbb{Z}_n)$ und kann man beweisen, daß die Addition abgeschlossen ist, so hat man einen Beweis für die Primalität von n bzw. wenn man eine Definitionslücke findet, hat man einen Beweis für die Zusammengesetztheit von n.

Zwar ist das deterministische direkte Auffinden in praktikabler Zeit nicht möglich, jedoch stellten Goldwasser und Kilian aufbauend auf diese Eigenschaften einen Primzahltest vor.

6.2 Goldwasser-Kilian

Satz 13 (Goldwasser-Kilian)**.** *Sei $n \in \mathbb{N}$ und nicht durch 2 oder 3 teilbar, des Weiteren seien $a, b \in \mathbb{Z}_n$ mit $ggT(4a^3+27b^2, n) = 1$ und $P \in E_{a,b}(\mathbb{Z}_n)$ mit $P \neq \mathcal{O}$. Falls für eine Primzahl $q > n^{\frac{1}{2}} + 2n^{\frac{1}{4}} + 1$ gilt, daß $qP = \mathcal{O}$ ist, so ist n prim.*

Um diesen Satz zu beweisen werden zuerst noch einige Definitionen und Zusammenhänge vorgestellt.

Definition. *Sei $x \in \mathbb{Z}_n$ und $p > 3$ ein Primteiler von n, so bezeichnen wir mit x_p die natürliche Projektion von $\mathbb{Z}_n$ nach $\mathbb{Z}_p$. (Ist $x = t \cdot p + d$, so ist $x_p = d$.) Ist nun $P = (x, y) \in E_{a,b}(\mathbb{Z}_n)$ ein Punkt auf einer elliptischen Pseudokurve, so sei $P_p = (x_p, y_p)$ und es gelte $\mathcal{O}_p = \mathcal{O}$.*

Lemma 9. *Ist $P, Q \in E_{a,b}(\mathbb{Z}_n)$, $p > 3$ ein Primteiler von n und $P + Q$ definiert, so ist*

$$(P + Q)_p = P_p + Q_p$$

Beweis: Die Addition zweier Zahlen auf elliptischen Kurven ist eine rationale Funktion, und so ist es gleichgültig, ob in jedem einzelnen Schritt die Modulorechnung durchgeführt wird oder nur abschließend. Für die Addition werden abhängig von den betroffenen Punkten aber verschiedene Rechenregeln angegeben. Es könnte deshalb vorkommen, daß die Summe in $\mathbb{Z}_p$ anders als in $\mathbb{Z}_n$ zu berechnen ist, in diesem Fall ist $P + Q$ in $\mathbb{Z}_n$ nicht definiert. Um die Behauptung zu zeigen, seien nun $P = (x_1, y_1)$ und $Q = (x_2, y_2)$. (Hier sollen die Indizes nicht die Projektionen kennzeichnen!)

- Fall $P = Q$, so ist auch $P_p = Q_p$ und die Addition wird nach der gleichen Regel berechnet.
- Fall $x_1 = x_2$ und $y_1 = -y_2$, so gilt auch $(x_1)_p = (x_2)_p$ und $(y_1)_p = (-y_2)_p$, und damit wird die Addition ebenfalls nach dem gleichen Regel berechnet.
- Fall $x_1 \neq x_2$: Hier kann $(x_1)_p = (x_2)_p$ sein. In diesem Fall gilt $p \mid (x_1 - x_2)$, und so kann man in $\mathbb{Z}_n$ für $(x_1 - x_2)$ kein inverses Element berechnen und damit ist $P + Q$ nicht definiert.

□

Satz 14 (Hasse)**.** *Die Ordnung einer elliptischen Kurve $E_{a,b}(\mathbb{Z}_p)$ läßt sich durch*

$$|| E | - (p + 1) | \leq 2\sqrt{p} \tag{6.9}$$

abschätzen.

Für einen Beweis siehe z.B. [15].

Bemerkung. *Aus dem Satz von Hasse folgt direkt*

$$\mid E \mid \leq (\sqrt{p}+1)^2 \tag{6.10}$$

Beweis: [Satz von Goldwasser-Kilian] Angenommen die Vorgaben sind alle erfüllt. Wäre nun n zusammengesetzt, dann existiert ein Primteiler $p \neq 2,3$ von n mit $p \leq \sqrt{n}$. Die Punktemenge $E_{a,b}(\mathbb{Z}_p)$ ist eine elliptische Kurve, da aus $\mathrm{ggT}(4a^3 + 27b^2, n) = 1$ unmittelbar $4a^3 + 27b^2 \not\equiv 0 \mod p$ folgt. $P_p \in E_{a,b}(\mathbb{Z}_p)$ ist ungleich $\mathcal{O}$, und aus Lemma 9 folgt $q \cdot P_p = \mathcal{O}$. Die Ordnung von P_p in $E_{a,b}(\mathbb{Z}_p)$ muß also q teilen, und da q eine Primzahl ist, ist die Ordnung gleich q. (Die Ordnung kann wegen $P_p \neq \mathcal{O}$ nicht 1 sein.) Damit hat auch $E_{a,b}(\mathbb{Z}_p)$ zumindest die Ordnung q, jedoch gilt nach dem Satz von Hasse, daß die Ordnung kleiner gleich $(\sqrt{p}+1)^2$ sein muß, und da $p \leq \sqrt{n}$ gilt, folgt daraus

$$\mid E_{a,b}(\mathbb{Z}_p) \mid \leq p + 2\sqrt{p} + 1 \leq n^{\frac{1}{2}} + 2n^{\frac{1}{4}} + 1 < q,$$

was ein Widerspruch ist, womit n nicht zusammengesetzt sein kann. □

Von diesem Satz bis zu einem „brauchbaren“ Primzahltest sind noch einige Schwierigkeiten aus dem Weg zu räumen. Die Größen a, b, P und q müssen gefunden werden, wobei q durch die Konstruktion mit der Vorgabe $q > n^{\frac{1}{2}} + 2n^{\frac{1}{4}} + 1$ selbst relativ hoch sein muß. Um die Primalität von q sicherzustellen, wird zuerst durch „schnell“ funktionierende probabilistische Tests die Primalität von q wahrscheinlich gemacht. Hat man nun alle Größen so gefunden, daß der Satz von Goldwasser-Kilian erfüllt ist, macht man sich mit einer neuerlichen Anwendung des Satzes von Goldwasser-Kilian an den Beweis der Primalität von q. Zu diesem Beweis braucht man wiederum eine Primzahl, und das Verfahren beginnt wieder von vorne. Somit wird die Primalität einer Zahl stets durch die Primalität einer kleineren Zahl bewiesen, dies wird so oft wiederholt, bis die Zahl, deren Primalität zu zeigen ist, so klein ist, daß andere Methoden für den Beweis ausreichen.

Wie generiert man nun a, b, P und q so, daß die geforderten Bedingungen erfüllt sind? Es werden so lange zufällig Zahlen a und b gewählt, bis $\mathrm{ggT}(4a^3 + 27b^2, n) = 1$ ist. Die Ordnung einer elliptischen Kurve, in diesem Fall von $E_{a,b}(\mathbb{Z}_p)$, kann mit dem sogenannten Algorithmus von Schoof, welchem Schoof 1985 in [31] vorstellte, berechnet werden. Ist die Ordnung in der Form $2 \cdot q$ darstellbar und eine schnelle Prüfung mit probabilistischen Tests ergibt, daß q sehr wahrscheinlich prim ist, so wird mit diesem q weitergerechnet, sonst geht man einen Schritt zurück und sucht andere Werte für a und b. Dies wird so lange fortgesetzt, bis ein primes q gefunden worden ist. Damit hat man dann die elliptische Kurve durch die Parameter a und b und q festgelegt. Um einen Punkt P zu finden wird unabhängig und zufällig ein x aus $\mathbb{Z}_p$ so oft gewählt, bis $x^3 + ax + b$ ein quadratischer

Rest modulo p ist. Danach wird $y = \sqrt{x^3 + ax + b}$ berechnet und eine der beiden Lösungen genommen, mit diesem dann qP berechnet. (Zum Wurzelziehen in quadratischen Zahlenkörpern haben Adleman, Manders und Miller in [32] einen Algorithmus vorgestellt.)

Ist dies gleich $\mathcal{O}$ ist man fertig, sonst wird dieser Schritt nach der Suche für ein geeignetes P wiederholt.

Dieser Test auf Primalität hat einen sehr großen Vorteil verglichen mit anderen, nämlich daß man, falls der Test erfolgreich abgelaufen ist, durch die Angabe der Größen a, b, P und q für jeden Schritt ein Zertifikat für die Primalität von n ausstellen kann.

6.3 Zeitaufwand

Die Zeitaufwandberechnung macht nur für Primzahlen Sinn, denn bei diesem Test versucht man die Primalität einer Zahl zu beweisen. Versucht man eine elliptische Kurve für eine zusammengesetzte Zahl mit den oben angegebenen Eigenschaften zu konstruieren, wird man nicht fündig. Stößt man nicht während der Berechnung auf eine Definitionslücke, terminiert der Algorithmus nicht. Dementsprechend ist vor der Durchführung eines Schrittes stets notwendig, mit einem probabilistischen Test die Primalität der zu prüfenden Zahl nahezulegen, des Weiteren muß ein Schritt nach zu langem erfolglosen Laufen samt probabilistischem Primzahltest neu gestartet werden.

Die Laufzeit beruht vor allem auf der folgenden Vermutung:

Vermutung. *Es existieren positive Konstanten c_1 und c_2, sodaß*

$$\pi(x + \sqrt{x}) - \pi(x) \geq \frac{c_2 \sqrt{x}}{\log^{c_1} x}$$

für genügend große x erfüllt ist.

Der Primzahlsatz stützt diese Vermutung:

Satz 15 (Primzahlsatz)**.**

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

Für einen Beweis des Primzahlsatzes siehe z.B. [34].

Geht man von der Annäherung $\pi(x) = \frac{x}{\ln x} = \frac{x}{2\ln(\sqrt{x})}$ aus, so ist

$$\pi(x + \sqrt{x}) = \frac{x + \sqrt{x}}{\ln(x + \sqrt{x})} = \frac{x + \sqrt{x}}{\ln(\sqrt{x} \cdot (\sqrt{x} + 1))} \approx \frac{x + \sqrt{x}}{2\ln(\sqrt{x})},$$

womit

$$\pi(x+\sqrt{x})-\pi(x) \approx \frac{x+\sqrt{x}}{2\ln(\sqrt{x})} - \frac{x}{2\ln(\sqrt{x})} = \frac{\sqrt{x}}{2\ln(\sqrt{x})} = \frac{\sqrt{x}}{\ln x}$$

wäre, was für die Vermutung spricht. Ein Beweis ist dies jedoch nicht, aufbauend auf die Arbeit [33] von Heath-Brown über die Verteilung von Primzahlen in Intervallen zeigten Goldwasser und Kilian in [28], daß die Vermutung - falls überhaupt - nur für sehr wenige Intervalle nicht stimmt.

Sei nun

$$S(p) = \left\{ q \,\middle|\, \frac{p+1-\lfloor\sqrt{p}\rfloor}{2} \leq q \leq \frac{p+1+\lfloor\sqrt{p}\rfloor}{2} \text{ und } q \text{ prim} \right\}. \tag{6.11}$$

Stimmt Vermutung 6.3, so ist $\mid S(p) \mid$ von der Größenordnung $\frac{\sqrt{p}}{\log^{c_1} p}$, sprich

$$\mid S(p) \mid = \mathcal{O}\left(\frac{\sqrt{p}}{\log^{c_1} p}\right). \tag{6.12}$$

Satz 16 (Lenstra)**.** *Sei $p > 5$ eine Primzahl und sei*

$$T \subseteq \{t \mid p+1-\lfloor\sqrt{p}\rfloor \leq t \leq p+1+\lfloor\sqrt{p}\rfloor\}.$$

Ist $E_{a,b}(\mathbb{Z}_p)$ eine elliptische Kurve mit zufällig gewählten a und b, so läßt sich die Wahrscheinlichkeit, daß die Anzahl der Punkte von $E_{a,b}(\mathbb{Z}_p)$ ein Element von T ist, für ein fixes c wie folgt abschätzen:

$$P\left(\mid E_{a,b}(\mathbb{Z}_p) \mid \in T\right) > \frac{c}{\log p} \cdot \frac{\mid T(p) \mid -2}{2\lfloor\sqrt{p}\rfloor + 1}$$

Für den Beweis siehe [46].

Aufbauend auf den Satz von Lenstra läßt sich die Wahrscheinlichkeit, durch zufällig gewählte Parameter eine geeignete elliptische Kurve zu erhalten, abschätzen:

Lemma 10. *Sei $p > 5$ und prim und $E_{a,b}(\mathbb{Z}_p)$ eine elliptische Kurve mit zufällig gewählten a und b, und sei S eine Menge wie weiter oben definiert. So gilt*

$$P(\mid E_{a,b}(\mathbb{Z}_p) \mid = 2q \text{ mit } q \text{ prim}) > \frac{c}{\log p} \cdot \frac{\mid S(p) \mid -2}{2\lfloor\sqrt{p}\rfloor + 1}$$

Beweis: Durch die Bijektion $t \to 2t$ läßt sich die Menge

$$\left[\frac{p+1-\lfloor\sqrt{p}\rfloor}{2}, \frac{p+1+\lfloor\sqrt{p}\rfloor}{2}\right]$$

in

$$[p+1-\lfloor\sqrt{p}\rfloor, p+1+\lfloor\sqrt{p}\rfloor]$$

überführen, dabei werden Primzahlen aus der Ursprungsmenge auf Zahlen der Form $2q$ mit q prim abgebildet. Dadurch läßt sich der Satz von Lenstra direkt anwenden und die Abschätzung folgt direkt. □

In diese Abschätzung wird nun (6.12) eingesetzt und die asymptotische Entwicklung betrachtet:

$$\frac{c}{\log p} \cdot \frac{\mid S(p) \mid -2}{2\lfloor\sqrt{p}\rfloor + 1} \longrightarrow \frac{c}{\log p} \cdot \frac{\frac{\sqrt{p}}{\log^{c_1} p}}{2\sqrt{p}} \longrightarrow \frac{c}{2\log p \cdot \log^{c_1} p}.$$

Der Kehrwert dieser Größe ergibt eine Abschätzung dafür, wie oft man probieren muß, bis eine passende elliptische Kurve gefunden wird, es müssen also $\mathcal{O}(\log p^{1+c_1})$ probiert werden. Die Aufstellung und das Prüfen der Gruppengröße einer Kurve benötigt die Schritte Erzeugung der Zufallsvariablen a und b, Berechnung von $\mathrm{ggT}(4a^3+27b^2, p)$ und die Überprüfung der Gruppengröße. (Man kann davon ausgehen, daß zufällig gewählte Variablen a und b eine elliptische Kurve ergeben, da für alle Werte von b höchstens zwei falsche Werte für a in Frage kommen, nämlich $\pm\sqrt{\frac{4a^3}{27}}$). Bei den aufgezählten Schritten ist die Erzeugung von Zufallsvariablen in $\mathcal{O}(1)$ und die Berechnung des größten gemeinsamen Teilers in $\mathcal{O}(\log p)$ durchführbar. Für die Ermittlung der Gruppengröße gab Schoof einen Algorithmus in $\mathcal{O}(\log^8 p)$ an, dieser Teil dominiert den Zeitlauf und damit ist dies auch die Größe für die Laufzeit für Erstellung und das Prüfen einer Kurve. Insgesamt braucht man bis zur Aufstellung einer geeigneten Kurve $\mathcal{O}(\log^{c+9} p)$ Zeit.

Der zweite Teil des Tests ist das Auffinden eines geeigneten Punktes. Eine elliptische Kurve ist isomorph zu einem Produkt zweier zyklisch additiver Gruppen $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ mit $m_1 \mid m_2$. Da die elliptische Kurve die Größe $2q$ hat, muß $m_1 m_2 = 2q$ sein, und für $q > 2$ folgt wegen der Teilbarkeit $m_1 = 1$ und $m_2 = 2q$. Dadurch ist $E_{a,b}$ isomorph zu $\mathbb{Z}_{2q}$, und diese Gruppe hat $q-1$ Elemente von Ordnung q. Damit erfüllt ungefähr jeder zweite Punkt die Voraussetzung, erwartungsgemäß reicht es also 2 x-Werte zu wählen und dazu den dazugehörigen y-Wert zu berechnen. Hierzu muß man ermitteln, ob $x^3 + ax + b$ ein quadratischer Rest ist. Dies wird von der Hälfte der Werte erfüllt. Das Berechnen der Quadratwurzel benötigt $\mathcal{O}(\log^4 p)$ Schritte. Zwei Punkte zu addieren benötigt $\mathcal{O}(\log^3 p)$ Schritte, und um $qP = \mathcal{O}$ zu prüfen muß man $\mathcal{O}(\log p)$ Mal das wiederholte Verdoppeln anwenden, womit die Laufzeit für diesen Teil mit $\mathcal{O}(\log^4 p)$ angegeben werden kann.

Insgesamt dominiert die Erzeugung einer geeigneten elliptischen Kurve, somit ist für einen Schritt die Zeitangabe $\mathcal{O}(\log^{c+9} p)$. Bei jedem Schritt wird die neue

Primzahl verglichen mit der Zahl, deren Primalität mit dieser gezeigt werden soll, ungefähr halb so groß sein. Es müssen also $\log p$ geeignete elliptische Kurven samt eines dazugehörigen Punktes gefunden werden. Da die zu prüfenden Zahlen kleiner werden, wird der Zeitaufwand von Schritt zu Schritt geringer, die Komplexität $\mathcal{O}(\log^{c+9} p)$ pro Schritt wird also nicht überschritten, womit die Zeitaufwandfunktion des Algorithmus in $\mathcal{O}(\log^{c+10} p)$ liegt. Bis jetzt wurde nicht betrachtet, daß der probabilistische Test zufällig eine zusammengesetzte Zahl als solche nicht erkennt. Goldwasser und Kilian zeigten in [28], daß dieser Fall die asymptotische Gesamtaufwandberechnung nicht beeinflusst, falls der probabilistische Primzahltest jeweils mit höchstens einer Wahrscheinlichkeit von $\frac{1}{p}$ fehlerhaft funktioniert.

6.4 Verbesserungen

Der langsamster Teil des Tests ist Schoofs Algorithmus. Dieser Algorithmus wurde seit der Vorstellung schon wesentlich effizienter gestaltet, verbesserte Versionen findet man in [35], [36], [37] und [38].

Im vorgestellten Algorithmus wird stets eine elliptische Kurve der Ordnung $2q$ gesucht. Es ist jedoch auch möglich, daß die Ordnung $r \cdot q$ ist, so lange q noch groß genug bleibt. Damit verringert sich der Erwartungswert der Anzahl der aufzustellenden elliptischen Kurven um eine passende Kurve zu finden. Diesen Ansatz verfolgend konnten Adleman und Huang in [39] und Lenstra Jr., Pila und Pomerance in [40] Verbesserungen erzielen.

Atkin verbesserte den Test durch die Einsparung des Schoofschen Algorithmus. In [41] entwickelte er eine alternative Methode, indem er die elliptische Kurve und deren Ordnung gleichzeitig festlegte. Diese Methode wurde in den Arbeiten [42] und [43] weiter verfeinert.

Morain erstellte ein Computerprogramm für den Primzahltest basierend auf elliptische Kurven. Dieses Programm ist unter [44] frei herunterladbar, in [45] gibt Morain einen Überblick über die praktische Anwendung seines Programmes an herkömmlichen Heim-PCs.

Kapitel 7

Primes in P

Am 6. August 2002 veröffentlichten Manindra Agrawal, Neeraj Kayal und Nitin Saxena vom Indian Institut of Technology Kanpur einen Algorithmus mit polynomialer Laufzeit, welcher definitiv sagen kann, ob eine beliebig vorgegebene Zahl prim ist oder nicht. Eine durch die Autoren verbesserte Version wurde 2004 in [47] veröffentlicht, dies bildet auch die Grundlage dieses Kapitels. Bis zu diesem Zeitpunkt war die Frage, ob es einen Algorithmus mit dieser Eigenschaft gibt, ungeklärt, obwohl Primzahltests in beinahe polynomialer Laufzeit bzw. in polynomialer Laufzeit aufgebaut auf die unbewiesene Riemannhypothese schon konzipiert wurden.

7.1 Grundidee

Das Verfahren beruht auf der folgenden Identität

Lemma 11. *Sei a relativ prim zu p. Dann und nur dann ist p eine Primzahl, falls*

$$(x-a)^p \equiv (x^p - a) \mod p$$

ist.

Zwei Polynome sind genau dann gleich modulo einer Zahl p, falls termenweise die Koeffizienten in der gleichen Restklasse bezüglich p liegen. Es genügt nicht, daß die Werte der Polynome modulo p gleich sind. So z.B. liefern für $p = 2$ die Polynome x und x^2 die gleichen Ergebnisse, die Polynome selbst sind jedoch nicht kongruent.

Lemma 11 ist eine Verallgemeinerung des Kleinen Fermatschen Satzes, welchen man durch $x = 0$ erhält. Will man von einer Zahl nun wissen, ob diese prim

ist oder nicht, genügt es „nur“ diese Identität anzuwenden. Um diese Identität zu verifizieren, müssen jedoch alle Koeffizienten des Polynoms berechnet werden, was den Algorithmus unbrauchbar macht. Die weiterführende Idee ist, daß man nicht alle Koeffizienten berechnet, sondern die Identität modulo eines Polynoms und nur für gewisse a-Werte berechnet. Klar ist, daß falls die zu prüfende Zahl eine Primzahl ist, die Identität stets erfüllt sein wird, jedoch ist dies umgekehrt nicht der Fall.

Die Herausforderung besteht darin, die Anzahl der durchgerechneten a-Werte zu minimieren und ein entsprechendes Polynom für die Modulorechnung zu finden, sodaß für eine zusammengesetzte Zahl mindestens eine der Berechnungen die Zahl als nicht prim identifiziert. Nach der Erstveröffentlichung des Algorithmus wurde dann vorallem durch Hendrik Lenstra Jr. und Carl Pomerance erfolgreich versucht, den Aufwand des Algorithmus zu senken.

Beweis: (Lemma 11) $(x-a)^p$ hat an der Stelle i den Koeffizienten $\binom{p}{i}a^{p-i}$

- ist p prim, so ist laut Fermat $a^p \equiv a \mod p$, x^n hat als Koeffizient 1. Für $0 < i < p$ gilt $\binom{p}{i} \equiv 0 \mod p$, da $\binom{p}{i} = \frac{p!}{i!(p-i)!}$, der Zähler enthält den Faktor p, der Nenner jedoch nicht.
- Ist p zusammengesetzt, so gibt es ein q mit $q \mid p$. Man bestimme k so, daß $q^k \mid p$ aber $q^{k+1} \nmid p$.

  $$\binom{p}{q} = \frac{p(p-1)\dots(p-q+1)}{q!} \Rightarrow q^k \nmid \binom{p}{q} \Rightarrow p \nmid \binom{p}{q} \Rightarrow \binom{p}{q} \not\equiv 0 \mod p.$$

  Aus $\mathrm{ggT}(a,p) = 1$ folgt unmittelbar, daß dann auch $\binom{p}{q}a^{p-q} \not\equiv 0$ sein muß. (Sei $\mathrm{ggT}(a,p) = 1$ und $d \not\equiv 0 \mod p$ und t beliebig. $a^t d \equiv 0(p) \Rightarrow a^t d = hp$. Da $\mathrm{ggT}(a,p) = 1$, gilt auch $\mathrm{ggT}(a^t,p) = 1 \Rightarrow p \mid d \Rightarrow d \equiv 0(p) \Rightarrow$ Widerspruch.)

□

Wie schon angedeutet, wird diese Identität nur modulo eines Polynoms und der zu prüfenden Zahl evaluiert. Als Polynom wird $x^r - 1$ mit einem geeignet gewählten r herangenommen, wodurch $x^r \equiv 1$ ist und sich so die Modulorechnung für Terme x^t mit $t > r$ „relativ“ einfach gestaltet. Geprüft wird also (für verschiedene a):

$$(x-a)^n \equiv (x^n - a) \mod x^r - 1, n. \tag{7.1}$$

Im folgenden wird falls eindeutig auf die Anführung des Polynoms $x^r - 1$ und des Moduls p desöfteren verzichtet.

7.2 Introspektivität

Für den Beweis wird zuerst die Eigenschaft introspektiv definiert und es werden weitere Eigenschaften abgeleitet, und um diese Eigenschaft herum werden zwei Gruppen definiert.

Definition. *Ein $c \in \mathbb{N}$ heißt introspektiv für ein Polynom $f(x)$ (und eine Zahl r und die Primzahl p), falls*

$$(f(x))^c = f(x^c) \mod x^r - 1, p$$

gilt.

Lemma 12. *Sind c und d introspektiv für $f(x)$, so ist auch $c \cdot d$ introspektiv für $f(x)$.*

Beweis: Da c introspektiv für $f(x)$ ist, gilt laut Definition:

$$(f(x))^c = f(x^c) \mod x^r - 1, p$$

$$(f(x))^{c \cdot d} = (f(x^c))^d \mod x^r - 1, p. \tag{7.2}$$

Nun wird die Introspektiveigenschaft von d ausgenutzt, gleichzeitig wird x durch x^c ersetzt:

$$(f(x^c))^d = f(x^{c \cdot d}) \mod x^{c \cdot r} - 1, p.$$

Da $x^r - 1$ ein Teiler von $x^{c \cdot r} - 1$ ist, folgt

$$(f(x^c))^d = f(x^{c \cdot d}) \mod x^r - 1, p. \tag{7.3}$$

Die Zeilen (7.2) und (7.3) zusammengesetzt ergibt dies

$$(f(x))^{c \cdot d} = (f(x^{c \cdot d})) \mod x^r - 1, p,$$

was zu beweisen war. □

Lemma 13. *Ist c introspektiv für $f(x)$ und $g(x)$, so ist c auch introspektiv für $f(x) \cdot g(x)$.*

Beweis:

$$(f(x) \cdot g(x))^c = (f(x))^c \cdot (g(x))^c = f(x^c) \cdot g(x^c)$$

□

Lemma 14. *Ist c introspektiv für $f(x) \cdot g(x)$ und $f(x)$, so ist c auch introspektiv für $g(x)$.*

Beweis: Aus den Ausgangsbedingungen gilt einerseits

$$(f(x) \cdot g(x))^c = f(x^c) \cdot g(x^c)$$

und andererseits

$$(f(x) \cdot g(x))^c = (f(x))^c \cdot (g(x))^c = f(x^c) \cdot (g(x))^c$$

Diese zwei Zeilen zusammengefasst bekommt man

$$f(x^c) \cdot g(x^c) = f(x^c) \cdot (g(x))^c,$$

woraus die Behauptung folgt. □

7.3 Der AKS-Algorithmus

Definition. *Ist $a \in \mathbb{Z}$ und $r \in \mathbb{N}$ mit $ggT(a, r) = 1$ so heißt das kleinste k mit $a^k \equiv 1 \mod r$ die Ordnung von a modulo r, in Zeichen $o_r(n) = k$.*

Der AKS-Algorithmus in einem Pseudocode:

1. Eingabe: n
2. Falls $n = a^b$ (für irgendein $a, b > 1$): Ausgabe „Zusammengesetzt“
3. Finde das kleinste r mit $o_r(n) > \log^2 n$
4. Falls $1 < \mathrm{ggT}(a, n) < n$ für irgendein $a \leq r$: Ausgabe „Zusammengesetzt“
5. Falls $n \leq r$: Ausgabe „Prim“
6. Falls für ein ein a mit $1 \leq a \leq \lfloor \sqrt{\varphi(r)} \log n \rfloor$ $(x - a)^n \not\equiv x^n - a \mod x^r - 1, n$: Ausgabe „Zusammengesetzt“
7. Ausgabe „Prim“

Nach einer Ausgabe kann das Programm selbstverständlich abgebrochen werden.

Ist die Zahl n eine Primzahl, ist klar, daß die Ausgaben „Zusammengesetzt“ in den Schritt 2, 4 und 6 nicht zustande kommen kann und dementsprechend das Programm spätestens im letzten Schritt „Prim“ ausgibt.

Ist nun die Zahl n zusammengesetzt, so kann im Schritt 5 nicht „Prim“ ausgegeben werden, da falls $n \leq r$ gilt, würde man im Schritt 4 schon einen Faktor finden und schon die Ausgabe „Zusammengesetzt“ bekommen.

Allein die Ausgabe im letzten Schritt für den Fall, daß n zusammengesetzt ist, ist nicht trivial und muß bewiesen werden, daß es so weit bei einer zusammengesetzten Zahl nicht kommen kann.

Nehmen wir nun an, wir haben eine zusammengesetzte Zahl n mit dem Primfaktor p, und das Programm läuft bis zum Schluss durch und man bekommt als Ergebnis „Prim“. Dann folgt daraus, daß für $0 \leq a \leq l = \lfloor \sqrt{\varphi(r)} \log n \rfloor$ die folgende Gleichung gilt:

$$(x-a)^n \equiv x^n - a \mod x^r - 1, n.$$

(Für $a = 0$ ist diese Gleichung trivialerweise wahr.)

Da $p \mid n$, gilt deshalb auch (stets für alle a mit $0 \leq a \leq l$)

$$(x-a)^n \equiv x^n - a \mod x^r - 1, p, \tag{7.4}$$

und da p eine Primzahl ist gilt wegen Lemma (11)

$$(x-a)^p \equiv x^p - a \mod x^r - 1, p. \tag{7.5}$$

Aus (7.4) und (7.5) folgt

$$(x-a)^{\frac{n}{p}} \equiv x^{\frac{n}{p}} - a \mod x^r - 1, p. \tag{7.6}$$

p und $\frac{n}{p}$ sind introspektiv für die Polynome $x+a$ mit $0 \leq a \leq l$. Aus Lemma 13 folgt daher, daß p und $\frac{n}{p}$ auch introspektiv für $(x+a)\cdot(x+a)$ sind und in weiterer Folge auch für alle weitere Potenzen von $(x+a)$ Sprich p und $\frac{n}{p}$ sind introspektiv für alle $(x+a)^{k_a}$, $k_a > 0$ beliebig. Des Weiteren sind diese beiden Zahlen auch für alle Multiplikationen zwischen $(x+a)^{k_a}$ mit $0 \leq a \leq l$ introspektiv. Sei nun

$$P = \left\{ \prod_{a=0}^{l} (x+a)^{k_a} \mid k_a \geq 0 \right\}$$

Wie oben gesehen sind p und $\frac{n}{p}$ introspektiv für alle Elemente von P, wegen Lemma 12 ist nun sowohl p^i als auch $(\frac{n}{p})^j$ mit $i, j \geq 0$ introspektiv für alle $f(x) \in P$, und aus Lemma 12 folgt weiters, daß auch $p^i \cdot (\frac{n}{p})^j$ mit $i, j \geq 0$ introspektiv für $f(x)$ sind. So wie die Polynome werden auch diese Zahlen zusammengefasst:

$$I = \left\{ p^i \cdot \left(\frac{n}{p}\right)^j \mid i, j \geq 0 \right\}$$

Zusammenfassend: Sei $m \in I$ und $f(x) \in P$, so ist m introspektiv für $f(x)$.

Aufbauend darauf werden nun zwei Gruppen definiert. Sei G die Menge der Restklassen der Zahlen aus I modulo r. Diese Gruppe wird wie aus dem Aufbau von I ersichtlich von p und $\frac{n}{p}$ erzeugt, und da Schritt 4 des Algorithmus für sowohl $\mathrm{ggT}(p, r) = 1$ als auch $\mathrm{ggT}(\frac{n}{p}, r) = 1$ sorgt, ist G eine Subgruppe von $\mathbb{Z}_r^*$. Sei die Mächtigkeit von $\mid G \mid = t$, wegen $n \in G$ und $o_r(n) > \log^2 n$ (siehe Schritt 3) folgt $t > \log^2 n$. Um die zweite Gruppe H zu definieren, müssen einige Hilfsfunktionen und Hilfsgrößen eingeführt werden. Sei $Q_r(x)$ das r-te Kreisteilungspolynom über $\mathbb{Z}_p$. $Q_r(x)$ teilt $x^r - 1$ und faktoriert diese in irreduzible Faktoren vom Grad $o_r(p) > 1$. Sei $h(x)$ eine dieser Faktoren. H ist nun die Menge der Restklassen der Polynome aus P modulo $h(x)$ und p. H wird durch die Polynome $x, x+1, \ldots, x+l$ multiplikativ erzeugt und ist eine Subgruppe der multiplikativen Gruppe von $\mathbb{F} = \mathbb{Z}_p[x]/(h(x))$.

Es werden nun die folgenden drei Abschätzungen für H gezeigt:

Lemma 15. *Es gilt stets*

$$\mid H \mid \geq \binom{t+l}{t-1} \tag{7.7}$$

Lemma 16. *Falls n keine Potenz von p ist, gilt für $\mid H \mid$*

$$\mid H \mid \leq n^{\sqrt{t}} \tag{7.8}$$

Lemma 17. *Für $l = \lfloor \sqrt{\varphi(r)} \log n \rfloor$ gilt*

$$\mid H \mid > n^{\sqrt{t}}$$

Aus Lemma 16 und Lemma 17 folgt, daß die Voraussetzungen für eine der Lemmata nicht erfüllt sein kann, dies ist aber nicht der Fall, da Schritt 2 des Algorithmus sicherstellt, daß n keine Potenz einer Zahl ist und Schritt 6 sorgt dafür, daß l mindestens $\lfloor \sqrt{\varphi(r)} \log n \rfloor$ ist. Somit kann nur die Annahme, daß n zusammengesetzt ist, falsch sein.

Beweis: (Lemma 15) Zuerst wird gezeigt, daß falls $f(x)$ und $g(x)$ Polynome mit einem Grad kleiner t in P sind, diese in $\mathbb{F}$ auf zwei verschiedene Elemente abgebildet werden. Nehmen wir an, dies wäre nicht der Fall und es gilt

$$f(x) = g(x) \text{ in } \mathbb{F}.$$

$h(x)$ ist ein Faktor des Kreisteilungspolynomes, deshalb ist x eine primitive Einheitswurzel in $\mathbb{F}$. Sei nun $m \in I$, damit ist m introspektiv für sowohl $f(x)$ als auch $g(x)$, und da $h(x)$ ein Teiler von $x^r - 1$ ist gilt auch

$$f(x^m) = g(x^m) \text{ in } \mathbb{F}$$

Daraus folgt, daß x^m für alle $m \in G$ eine Nullstelle des Polynoms $k(y) = f(y) - g(y)$ ist. Da wie weiter oben erwähnt G eine Subgruppe von $\mathbb{Z}_r^*$ ist, also $\text{ggT}(m, r) = 1$ gilt, ist jedes x^m auch eine r-te Primitiveinheitswurzel. $k(y)$ hat also $\mid G \mid = t$ verschiedene Nullstellen in $\mathbb{F}$. $k(y)$ ist aber die Differenz zweier Polynome mit dem Grad kleiner t, also muß auch der Grad von $k(y)$ kleiner als t sein und damit kann $k(y)$ nicht t verschiedene Nullstellen haben. Daraus folgt, daß die Annahme nicht stimmt und $f(x)$ und $g(x)$ auch auf verschiedene Elemente in H abgebildet werden.

Da $l = \lfloor \sqrt{t} \log n \rfloor < \sqrt{r} \log n < r$ und $p > r$, ist für $0 \leq i \neq j \leq l$ auch in $\mathbb{Z}_p$ $i \neq j$. Damit sind auch die Elemente $x, x+1, \ldots, x+l$ in $\mathbb{Z}_p$ verschieden. Grad von $h(x)$ ist größer als 1, damit kann auch ein $x + a$ nicht 0 in $\mathbb{Z}_p$ sein, woraus folgt, daß mindestens $l + 1$ verschiedene Polynome mit dem Grad 1 in H existieren. Die Anzahl der Polynome mit Grad kleiner t erhält man durch eine Kombination (mit Zurücklegen). Die Formel für eine Kombination mit Zurücklegen für n aus k ist $\binom{n+k-1}{k}$, wobei in diesem Fall $n = t$ und $k = l + 1$ ist:

$$\mid H \mid \geq \binom{t+l+1-1}{l+1} = \binom{t+l}{l+1} = \binom{t+l}{t+l-(l+1)} = \binom{t+l}{t-1} \tag{7.9}$$

□

Beweis: (Lemma 16) Sei

$$I' = \{p^i \cdot (\frac{n}{p})^j \mid 0 \leq i, j \leq \lfloor \sqrt{t} \rfloor\}.$$

n keine Potenz von p garantiert, daß p^i und $(\frac{n}{p})^j$ stets verschiedene Zahlen (bis auf den Fall $i = j = 0$) sind. I' hat damit $(\lfloor \sqrt{t} \rfloor + 1)^2 > t$ verschiedene Elemente. Da $\mid G \mid = t$ ist, sind mindestens 2 Zahlen aus I' gleich modulo r, seien dies die Zahlen m_1 und m_2 und sei $m_1 > m_2$. Es gilt daraus

$$x^{m_1} = x^{m_2} \mod x^r - 1.$$

Sei nun $f(x)$ ein beliebiges Polynom aus P, dann gilt für $f(x)$

$$(f(x))^{m_1} = f(x^{m_1}) = f(x^{m_2}) = (f(x))^{m_2} \mod x^r - 1, p.$$

Daraus folgt also

$$(f(x))^{m_1} = (f(x))^{m_2} \text{ in } \mathbb{F}.$$

$f(x) \in H$ ist damit eine Nullstelle des Polynoms

$$k(y) = y^{m_1} - y^{m_2}$$

in $\mathbb{F}$. Da $f(x)$ ein beliebiges Element von H ist, können alle Elemente von H Nullstellen sein, $k(y)$ hat also mindestens $\mid H \mid$ verschiedene Nullstellen. Der Grad von $k(y)$ ist $m_1 \leq (p \cdot \frac{n}{p})^{\lfloor\sqrt{t}\rfloor} \leq n^{\lfloor\sqrt{t}\rfloor}$. Zusammengesetzt folgt daraus

$$\mid H \mid \leq n^{\lfloor\sqrt{t}\rfloor}$$

□

Beweis: (Lemma 17) Wie weiter oben gezeigt gilt

$$\mid H \mid \geq \binom{t+l}{t-1}. \tag{7.10}$$

t wurde so gewählt, daß $t > \log^2 n$ und daher ist $t > \sqrt{t}\log n$. Wird nun also in der oberen Ungleichung t durch $\lfloor\sqrt{t}\log n\rfloor + 1$ ersetzt, so werden die Größen des Binomialkoeffizienten entweder gleichbleiben oder beide um die gleiche Summe verringert werden:

$$\binom{t+l}{t-1} \geq \binom{l + \lfloor\sqrt{t}\log n\rfloor + 1}{\lfloor\sqrt{t}\log n\rfloor}. \tag{7.11}$$

G ist eine Subgruppe von $\mathbb{Z}_r^*$, und so ist $\mid G \mid = t \leq \varphi(r)$, woraus $\lfloor\sqrt{t}\log n\rfloor \leq \lfloor\sqrt{\varphi(r)}\log n\rfloor = l$ folgt. Diese Ungleichung führt zu

$$\binom{l + \lfloor\sqrt{t}\log n\rfloor + 1}{\lfloor\sqrt{t}\log n\rfloor} \geq \binom{2\lfloor\sqrt{t}\log n\rfloor + 1}{\lfloor\sqrt{t}\log n\rfloor}. \tag{7.12}$$

Hier nun eine Abschätzung des Binomialkoeffizienten, wobei wegen der Leserlichkeit $\lfloor\sqrt{t}\log n\rfloor$ durch b ersetzt wird:

$$\binom{2b+1}{b} = \frac{(2b+1)!}{b!(b+1)!} = \prod_{i=1}^{b} \frac{(b+1+i)}{i} > \frac{b+2}{1}\prod_{i=2}^{b} 2 = (b+2)2^{b-1}. \tag{7.13}$$

Da $b = \lfloor\sqrt{t}\log n\rfloor > \lfloor\log^2 n\rfloor \geq 1$ gilt, ist (b+2) stets mindestens 4 groß, womit

$$\binom{2b+1}{b} > (b+2)2^{b-1} \geq 2^{b+1} \tag{7.14}$$

gilt. Eingesetzt in (7.12) ergibt dies

$$\binom{2\lfloor\sqrt{t}\log n\rfloor + 1}{\lfloor\sqrt{t}\log n\rfloor} > 2^{\lfloor\sqrt{t}\log n\rfloor + 1} \geq 2^{\sqrt{t}\log n} = n^{\sqrt{t}}, \tag{7.15}$$

insgesamt also

$$\mid H \mid > n^{\sqrt{t}}.$$

□

7.4 Laufzeit

Es ist also nun gezeigt, daß der Algorithmus funktioniert und damit ein korrekter Primzahltest ist. Es bleibt also nur noch zu zeigen, daß er tatsächlich eine polynomiale Laufzeit besitzt. Bevor man die Arbeitsschritte (Multiplikationen und Additionen) zählen kann, muß man erst zeigen, daß ein r im Schritt 3 existiert und man muß eine obere Schranke angeben, damit man weiß, für wieviel Werte $o_r(n) > \log^2 n$ evaluiert werden muß.

Lemma 18. *Für alle $n > 1$ existiert ein r mit $r \leq \max\{3, \lceil \log^5 n \rceil\}$, sodaß $o_r(n) > \log^2 n$ ist.*

Beweis: Für $n = 2$ wird das Lemma durch $r = 3$ erfüllt, sei daher im weiteren $n > 2$ und $S = \{r_1, r_2, \ldots, r_t\}$ die Menge alle Zahlen mit $o_{r_i}(n) \leq \log^2 n$ oder $r_i \mid n$. Sei weiters

$$\Pi = n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} (n^i - 1).$$

Aus der Definition von $r_i \in S$ geht hervor, daß für alle i gilt

$$r_i \mid \Pi,$$

da r_i direkt n oder $n^{o_{r_i}(n)} - 1$ teilt. Für Π gilt die Abschätzung

$$\begin{aligned}
\Pi &= n \cdot \textstyle\prod_{i=1}^{\lfloor \log^2 n \rfloor} (n^i - 1) \\
&< n \cdot \textstyle\prod_{i=1}^{\lfloor \log^2 n \rfloor} n^i \\
&= n \cdot n^{1+2+\cdots+n^{\lfloor \log^2 n \rfloor}} \\
&= n \cdot n^{\frac{\lfloor \log^2 n \rfloor (\lfloor \log^2 n \rfloor + 1)}{2}} \\
&= n^{\frac{\lfloor \log^2 n \rfloor^2 + \lfloor \log^2 n \rfloor + 2}{2}}.
\end{aligned}$$

Nun wird gezeigt, daß der Exponent mit $\log^4 n$ abgeschätzt werden kann. Es gilt stets $\lfloor \log^2 n \rfloor + 2 < \lfloor \log^2 n \rfloor^2$, da der Logarithmus eine stark monoton steigende Funktion ist und ab dem Funktionswert 1 wächst die rechte Seite wegen dem Quadrat schneller. Die Ungleichung ist für $n = 3$ schon erfüllt, also wird sie für alle $n > 2$ gelten, woraus

$$n^{\frac{\lfloor \log^2 n \rfloor^2 + \lfloor \log^2 n \rfloor + 2}{2}} < n^{\frac{\lfloor \log^2 n \rfloor^2 + \lfloor \log^2 n \rfloor^2}{2}} = n^{\lfloor \log^2 n \rfloor^2} < n^{(\log^2 n)^2} = n^{\log^4 n}$$

folgt. Für Π bedeutet dies

$$\Pi < n^{\log^4 n} = (2^{\log n})^{\log^4 n} = 2^{\log^5 n}$$

Für das kleinste gemeinsame Vielfache aller Zahlen bis m gilt, daß für m größer 7 dieser Wert stets größer als 2^m ist. (Für einen Beweis siehe z.B. [48].) Daher gilt für die ersten $\lceil \log^5 n \rceil$ Zahlen (für $n = 3$ ist $\lceil \log^5 n \rceil \approx 10.002$, also die oben erwähnte Abschätzung anwendbar), daß deren kleinstes gemeinsames Vielfaches mindestens $2^{\lceil \log^5 n \rceil}$ ist, also größer als Π. Daher muß eine Zahl s unter $\lceil \log^5 n \rceil$ existieren, die Π nicht teilt, und da $s \nmid \Pi$ gilt auch $s \notin S$. Ist nun $\mathrm{ggT}(s, n) = 1$, so ist die Ordnung von s definiert und ist größer als $\log^2 n$ und damit ist das gewünschte $r = s$ gefunden. Ist nun $\mathrm{ggT}(s, n) > 1$, so sei $r = \frac{s}{\mathrm{ggT}(s,n)}$. s teilt nicht n und $\mathrm{ggT}(s, n) \in S$, und damit ist $r \notin S$ und daher gilt $o_r(n) > \log^2 n$. □

Für die Angaben der Laufzeit wird hier auf die Arbeit [49] zurückgegriffen.

Eine Addition, Multiplikation bzw. Division von zwei in Binärdarstellung m-stelligen Zahlen kann in $\mathcal{O}^\sim(m)$ erledigt werden[1], die gleichen Funktionen für Polynome mit dem Grad d und Koeffizienten mit höchsten m (dualen) Stellen kann nach [49] in $\mathcal{O}^\sim(d \cdot m)$ erledigt werden.

Nun zu den einzelnen Berechnungen in den jeweiligen Schritten:

- Schritt 2: Laufzeit beträgt laut [49] $\mathcal{O}^\sim(\log^3 n)$
- Schritt 3: Gesucht wird ein r mit $o_r(n) > \log^2 n$, dazu wird für verschiedene r-Werte durch fortlaufende Multiplikation mit r die Potenzen r^k mit $k \leq \log^2 n$ berechnet. Für jeweils ein r braucht man höchstens $\log^2 n$ Multiplikationen modulo r und daher einen Zeitaufwand von $\mathcal{O}^\sim(\log^2 n \log r)$. Lemma 7.4 stellt sicher, daß höchstens $\log^5 n$ Zahlen zu testen sind. Damit ist r höchstens $\log^5 n$ und der Gesamtzeitaufwand für diesen Schritt ist $\mathcal{O}^\sim(\log^7 n)$.
- Schritt 4: Eine Berechnung des größten gemeinsamen Teilers kann laut [49] in $\mathcal{O}(\log n)$ erledigt werden, Gesamtaufwand ist daher $\mathcal{O}(r \log n)$, also $\mathcal{O}(\log^6 n)$.
- Schritt 5: Ist bloß ein Vergleich von 2 Zahlen, Aufwand ist $\mathcal{O}(\log n)$.
- Schritt 6: Es werden $\lfloor \sqrt{\varphi(r)} \log n \rfloor$ Gleichungen evaluiert. Jede dieser Gleichungen benötigt $\mathcal{O}(\log n)$ Multiplikationen von Polynomen mit dem Grad r und Koeffizienten mit der Größe $\mathcal{O}(\log n)$. Eine Überprüfung der Gleichung benötigt dementsprechend $\mathcal{O}^\sim(r \cdot \log^2 n)$. Die gesamte Evaluation braucht dementsprechend eine Laufzeit von $\mathcal{O}^\sim(r\sqrt{\varphi(r)} \cdot \log^3 n) = \mathcal{O}^\sim(r^{\frac{3}{2}} \cdot \log^3 n) = \mathcal{O}^\sim(\log^{\frac{21}{2}} n)$.

[1] Für die Definition von $\mathcal{O}^\sim$ siehe Anhang A.

Schritt 6 ist der aufwendigste und ist daher für den Gesamtlaufzeit ausschlaggebend, damit ist die Laufzeit der Algorithmus mit der oberen Schranke $\mathcal{O}^{\sim}(\log^{\frac{21}{2}} n)$ anzugeben.

7.5 Verbesserungen der Laufzeit

Die Zeitabschätzung hängt stark von der Abschätzung für r ab, kann man eine kleinere Schranke für das Finden eines geeigneten r angeben, so läßt sich die Zeitkomplexität verringern. Es existieren viele verschieden Vermutungen, die es nahelegen, daß für r tatsächlich eine niedrigere Schranke angegeben werden kann.

Vermutung. *(Artin) Sei $n \in \mathbb{N}$ beliebig aber keine Quadratzahl. Die Anzahl der Primzahlen $q < m$ mit $o_q(n) = q - 1$ läßt sich asymptotisch durch $A(n) \cdot \frac{m}{\ln m}$ abschätzen, wobei $A(n) > 0,35$ für die Artin-Konstante steht.*

Es wurden Fortschritte Richtung Beweis dieser Vermutung in [51], [52] und [53] gemacht; des Weiteren ist die Erweiterte Riemannsche Hypothese richtig, so auch Atkins Vermutung.

Vermutung. *(Vermutung über die Häufigkeit von Sophie-Germain-Primzahlen) Eine Primzahl p wird Sophie-Germain-Primzahl genannt, falls auch $2p + 1$ eine Primzahl ist. Die Anzahl der Sophie-Germain-Primzahlen bis m läßt sich asymptotisch durch $\frac{2C_2 m}{\ln^2 m}$ abschätzen, wobei C_2 die Zwillingskonstante ist und ungefähr $0,66$ beträgt.*

Aus beiden Vermutungen folgt eine bessere Abschätzung für r, die Zeitkomplexität, unter der Annahme einer der beiden Vermutungen ist richtig, beträgt $\mathcal{O}^{\sim}(\log^6 n)$.

Auch ohne Vermutungen kann r weiter begrenzt werden, Fouvry hat in [54] ein Lemma veröffentlicht, mit dessen Hilfe der AKS-Test eine Laufzeit von $\mathcal{O}^{\sim}(\log^{7,5} n)$ besitzt:

Lemma 19. *Es existiert ein $c > 0$ und ein n_0, sodaß für alle $x > n_0$*

$$| \{q \mid q \leq x \text{ ist eine Primzahl und } q - 1 \text{ hat einen Primteiler } p > q^{\frac{2}{3}}\} | \geq c \cdot \frac{x}{\ln x}$$

gilt.

Dieses Lemma ist ein relativ großes Geschütz der Mathematik, die Angabe der Konstante c ist recht schwierig und allgemein zur Zeit nicht möglich. Nach Berechnungen von Baker und Hartman in [55] gilt das Lemma für $c \leq 0,6683$.

Lenstra und Pomerance wählten in [56] einen anderen Ansatz zur Verbesserung des Testes, indem sie eine andere Funktion für die Modulorechnung verwendet haben. Damit läßt sich der AKS-Test, ohne auf Vermutungen zurückgreifen zu müssen, auf eine Laufzeit von $\mathcal{O}^{\sim}(\log^6 n)$ verbessern.

Die Entdecker schlugen eine weitere mögliche Verbesserung der Laufzeit des AKS-Tests auf eine Zeitkomplexität von $\mathcal{O}^{\sim}(\log^3 n)$ vor, welche wiederum auf eine von Bhattacharjee und Pandey in [57] vorgestellte Vermutung beruht.

Vermutung. *Falls r eine Primzahl ist und n nicht teilt und ist*

$$(x-1)^n = x^n - 1 \mod x^r - 1, n,$$

so ist entweder n prim oder $n^2 = 1 \mod r$.

Kayal und Saxena zeigten in [58], daß die Vermutung für $r \leq 100$ und $n \leq 10^{10}$ gilt, jedoch deuten heuristische Untersuchungen von Lenstra und Pomerance in [59] darauf hin, daß diese Vermutung allgemein nicht gilt.

Anhang A

Landau-Symbol $\mathcal{O}$

Definition (Landau-Symbol, Groß-O)**.**

$$\mathcal{O}(f) = \left\{g(n) : \mathbb{N} \to \mathbb{R}_0^+ \mid \exists c \in \mathbb{R}^+, \exists n_o \in \mathbb{N} : \forall n \geq n_0 \quad g(n) \leq c \cdot f(n)\right\}$$

Durch die sogenannte „Groß-O-Notation“ wird eine obere asymptotische Schranke für eine Funktion angegeben. Abgesehen vom $\mathcal{O}$-Symbol gibt es weitere Landau-Symbole (z.B. für untere asymptotische Schranken), in dieser Arbeit werden diese aber nicht benötigt und dementsprechend auch nicht behandelt. Für weiterführende Informationen siehe z.B. [9].

Definition. *Als Vereinfachung wird* $\mathcal{O}^{\sim}(t(n))$ *für* $\mathcal{O}(t(n) \cdot p(\log t(n)))$ *verwendet, wobei* $p(x)$ *für ein beliebiges Polynom steht.*

Obwohl $\mathcal{O}(f(x))$ eine Menge von Funktionen darstellt, wird statt der Schreibweise $g(x) \in \mathcal{O}(f(x))$ üblicherweise $g(x) = \mathcal{O}(f(x))$ verwendet. Diese Schreibweise wird auch in dieser Arbeit verwendet. Die Bezeichnung $\mathcal{O}(f) = \mathcal{O}(g)$ bedeutet nicht, daß diese zwei Mengen gleich sind sondern $\mathcal{O}(f) \subseteq \mathcal{O}(g)$.

Aus der Definition ist es leicht ersichtlich, daß für alle positiven Konstanten k

$$\mathcal{O}(f(x)) = \mathcal{O}(k \cdot f(x))$$

gilt. Weiters, falls für eine Funktion $g(x)$ für alle $n \geq n_1$ stets $g(n) \leq f(n)$ erfüllt ist, so ist

$$\mathcal{O}(g(x)) = \mathcal{O}(f(x)).$$

Damit gilt für ein beliebiges Polynom $f(x)$ von Grad d

$$\mathcal{O}(f(x)) = \mathcal{O}(x^d).$$

Auf dem ersten Blick scheint bei einer Laufzeitbetrachtung das Weglassen von beliebigen Koeffizienten eine zu große Verallgemeinerung zu sein. Bei der Betrachtung durch das $\mathcal{O}$-Symbol fallen z.B. Algorithmen mit n^3 Schritten in die gleiche Kategorie wie Algorithmen mit $1000 \cdot n^3$ Schritten. In der Praxis macht dies sehr wohl einen großen Unterschied, weshalb oft Algorithmen von höherer Ordnung vorgezogen werden. Das $\mathcal{O}$-Symbol verrät jedoch, wie ein Algorithmus bei z.B. Verdopplung der Eingabenlänge reagiert. Egal ob ein Algorithmus $1000 \cdot n^3$ oder n^3 Schritte benötigt, bei Verdopplung des Wertes wird der Aufwand verachtfacht. Diese Betrachtung ist auch deswegen interessant, da die Frage, wie viel Zeit (in Sekunden, Stunden, etc.) ein Algorithmus benötigt auch davon abhängt, wie schnell man einen einzelnen Schritt (Multiplikation, etc.) ausführen kann. Weiß man für einen Eingangswert n wie lange der Algorithmus rennt, kann man aus dem $\mathcal{O}$-Symbol darauf schließen, bis zu welchem Eingangswert der Algorithmus für den Verwendungszweck brauchbar ist. Wird von Laufzeit gesprochen, meint man dabei mathematisch die Anzahl der benötigten Operationen für den Algorithmus.

Definition. *Eine Funktion wächst polynomiell, falls $f(n) = \mathcal{O}(n^k)$ für ein $k \in \mathbb{N}$ gilt. Ein Algorithmus wird polynomieller Algorithmus bzw. ein Algorithmus mit polynomialer Laufzeit bezeichnet, falls man eine Funktion f in Abhängigkeit mit der Eingangsvariable für den Algorithmus für die Laufzeit angeben kann und f polynomiell wächst.*

Anhang B

Das Legendre- und Jacobi-Symbol

Definition (Quadratischer Rest)**.** *Sei $m \in \mathbb{N}$, dann heißt eine ganze Zahl a mit $ggT(a, m) = 1$ quadratischer Rest modulo m, falls ein x mit*

$$x^2 \equiv a \mod m \tag{B.1}$$

existiert. Falls für eine ganze Zahl a mit $ggT(a, m) = 1$ kein x gefunden werden kann, sodaß (B.1) erfüllt werden kann, so heißt a quadratischer Nichtrest modulo m.

Definition (Legendre-Symbol)**.** *Für eine Primzahl p und eine ganze Zahl a ist der Legendre-Symbol $\left(\frac{a}{p}\right)$ wie folgt definiert:*

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \textit{falls } a \textit{ ein Vielfaches von } p \textit{ ist} \\ 1 & \textit{falls } a \textit{ ein quadratischer Rest modulo } p \textit{ ist} \\ -1 & \textit{falls } a \textit{ ein quadratischer Nichtrest modulo } p \textit{ ist} \end{cases}$$

Das Legendre-Symbol läßt sich durch das Eulersche Kriterium[1] berechnen, daneben existieren zahlreiche Rechenregel für dieses Symbol, für Beweise und weiterführende Literatur siehe [60]:

- Falls $a \equiv b \mod p$ gilt, so ist

$$\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$$

[1] $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \mod p$

- $$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$$

- $$\left(\frac{-1}{p}\right) = \begin{cases} 1 & \text{falls } p \equiv 1 \mod 4 \\ -1 & \text{falls } p \equiv -1 \mod 4 \end{cases}$$

- Für ungerade Primzahlen p gilt

$$\left(\frac{2}{p}\right) = \begin{cases} 1 & \text{falls } p \equiv \pm 1 \mod 8 \\ -1 & \text{falls } p \equiv \pm 3 \mod 8 \end{cases}$$

- Für ungerade Primzahlen p gilt

$$\left(\frac{3}{p}\right) = \begin{cases} 1 & \text{für } p = 1 \text{ oder } 11 \mod 12 \\ -1 & \text{für } p = 5 \text{ oder } 7 \mod 12 \end{cases}$$

- Sind p und q zwei verschiedene Primzahlen ungleich 2, so gilt (Quadratisches Reziprozitätsgesetz)

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = \begin{cases} -1 & \text{falls } p \equiv q \equiv -1 \mod 4 \\ 1 & \text{sonst} \end{cases}$$

  Dies ist mit den folgenden zwei Formulierungen äquivalent:

$$\left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{q}{p}\right) & \text{falls } p \equiv q \equiv -1 \mod 4 \\ \left(\frac{q}{p}\right) & \text{sonst} \end{cases}$$

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

Das Legendre-Symbol ist nur für prime p definiert, bei den Rechenregeln verursacht dies insoweit eine Schwierigkeit, als bei der Anwendung des quadratischen Reziprozitätsgesetzes zuerst die vollständige Primzerlegung durchzuführen ist. Um diesen Schritt einzusparen und dadurch allgemeiner rechnen zu können, wurde das Jacobi-Symbol eingeführt.

Definition (Jacobi-Symbol)**.** *Sei $b > 1$ und ungerade, $ggT(a, b) = 1$ und $b = p_1 p_2 \cdot \ldots \cdot p_t$ mit (nicht unbedingt verschiedenen) primen p_i, so wird das Jacobi-Symbol mithilfe des Legendre-Symbols wie folgt definiert:*

$$\left(\frac{a}{b}\right) = \left(\frac{a}{p_1}\right) \cdot \left(\frac{a}{p_2}\right) \cdot \ldots \cdot \left(\frac{a}{p_t}\right)$$

Die Rechenregel für das Legendre-Symbol sind dadurch auch anwendbar, beim quadratischen Reziprozitätsgesetz muß wie schon erwähnt nicht getestet werden, ob beide Zahlen Primzahlen sind. Die beiden Symbole werden mit den gleiche Zeichen versehen, dies stellt aber kein Problem dar, da der Wert des Jacobi-Symbols mit dem Wert des Legendre-Symbol übereinstimmt, falls dieses berechenbar ist.

Bemerkung. *Es ist zu beachten, daß man aus dem Jacobi-Symbol nicht schließen kann, ob eine Zahl ein quadratischer Rest ist. Ist aber das Jacobi-Symbol gleich* -1*, so handelt es sich definitiv um einen quadratischen Nichtrest.*

Anhang C

RSA

Das zur Zeit wichtigste Einsatzgebiet von Primzahltests ist im Bereich der Kryptographie. In diesem Kapitel soll das RSA-Verfahren behandelt werden. Dieses Verfahren wurde Ende der siebziger Jahre von Rivest, Shamir und Adleman entwickelt und in [61] veröffentlicht, zur Benennung des Verfahrens wurden die Anfangsbuchstaben dieser Personen herangenommen.

Definition (Einwegfunktion)**.** *Eine Funktion $f(x)$ wird Einwegfunktion genannt, falls $f(x)$ für alle Werte aus dem Definitionsbereich „schnell“ zu berechnen sind, jedoch bei einem aus dem Wertebereich gegebenen y das Urbild x ($y = f(x)$) trotz Kenntnis der Funktion $f(x)$ „praktisch unmöglich“ (=„zeitintensivst“) zu berechnen ist.*

Die Begriffe „schnell“ und „praktisch unmöglich“ können nicht genau definiert werden, da diese Begriffe von neuen Algorithmen und vorallem von stetig schneller werdenden (Super-)Computern abhängig sind. Unter „schnell“ soll verstanden werden, daß man im täglichen Gebrauch einsetzen können soll, ohne dabei unnötig lange zu warten (z.B. bei Abfrage am Bankomat soll man nicht minutenlang warten müssen). „Praktisch unmöglich“ soll beinhalten, daß man mit den besten Computern zur Entschlüsselung jahrelang braucht, sodaß bis zur Ermittlung der Ausgangszahl die Information veraltet ist. (Eine Bankomatkarte wird alle drei Jahre ausgetauscht, benötigt man zum „Knacken“ mehr als drei Jahre, dann ist die Arbeit unnötig gewesen.)

Beispiel. *Die Multiplikation (mit bestimmten Einschränkungen) kann als Einwegfunktion betrachtet werden. Es ist leicht zwei Zahlen zu multiplizieren aber wesentlich schwerer aus einer Zahl herauszufinden, welche zwei Zahlen zusammenmultipliziert worden sind. An die zwei Zahlen muß man einige Kriterien stellen, damit die Funktion tatsächlich eine Einwegfunktion wird. Die verwendeten Zahlen sollten relativ groß sein und müssen selbst Primzahlen sein. Um*

die Sicherheit, daß große Zahlen schwer zu faktorisieren sind, zu demonstrieren, wurde von RSA Laboratories ein „RSA Factoring Challenge“ ins Leben gerufen. Bei diesem Challenge wurde u.a. eine 663-bit-Zahl in ihren zwei Primfaktoren zerlegt, lt. Angaben von RSA Laboratories in [50] braucht man mit einem 2.2 Gigahertz Opteron CPU 55 Jahre für die Faktorisierung.

Bemerkung. *Verschlüsselt werden stets Zahlen und nicht Texte. Um Texte zu verschlüsseln müssen diese zuerst in Zahlen umgewandelt werden. Bei den meisten Verschlüsselungsalgorithmen (so auch RSA) werden die Zahlen in Blöcke zu einer gewissen Anzahl von Ziffern unterteilt und die Verschlüsselung wird auf die Blöcke angewendet und danach werden die einzelnen Blöcke wieder zu einer Zahlenkolonne zusammengeschweißt.*

Definition (Falltürfunktionen)**.** *Falltürfunktionen sind Einwegfunktionen, die mit Zusatzinformationen effizient umgekehrt werden können. Solche Funktionen werden auch noch Einwegfunktionen mit Falltür oder Trapdoor-Einwegfunktionen genannt.*

Bemerkung. *Der Einsatz von Falltürfunktionen wird in der Kryptographie verwendet, damit man sich durch eine Falltürfunktion verschlüsselte Botschaften zuschicken kann, wobei bei der Konstruktion dieser Funktion die Falltüre mitkonstruiert wird, jedoch diese nur für sich behalten wird. Dies hat zum Vorteil, daß die Verschlüsselung nicht von der Geheimhaltung der Methode abhängt, sondern von der Geheimhaltung der Falltür. Dadurch können alle Menschen mit der gleichen Funktion die Nachricht verschlüsseln, man muß nicht mit jedem einzelnen Kommunikationspartner geheime Schlüssel austauschen.*

Das RSA-Verfahren - Nachrichten schicken:

Person A will sich Nachrichten verschlüsselt zusenden lassen.

1. A wählt zwei Primzahlen p und q
2. A berechnet $n = p \cdot q$
3. A wählt ein s mit der Bedingung $\mathrm{ggT}(s, (p-1)(q-1)) = 1$
4. A veröffentlicht das Wertepaar (n, s)

Person B will die Nachricht a an A schicken.

1. B berechnet $m = a^s \mod n$
2. B schickt m an A

Laut dem Satz von von Euler-Fermat gilt $a^{\varphi(n)} = a^{(p-1)(q-1)} \equiv 1 \mod n$.

A muß ein t finden, sodaß $s \cdot t \equiv 1 \mod (p-1)(q-1)$, damit kann man aus a^s wieder die ursprüngliche Nachricht a gewinnen:

$$\begin{aligned}
(a^s)^t &= a^{st} \\
&= a^{k(p-1)(q-1)+1} \\
&= a \cdot a^{k(p-1)(q-1)} \\
&= a \cdot (a^{(p-1)(q-1)})^k \\
&\equiv a \cdot 1^k \\
&\equiv a \mod n
\end{aligned}$$

Satz 17. *Seien $a, b > 0 \in \mathbb{Z}$ und sei $d = ggT(a, b)$. Dann ist*

$$\{xa + yb \mid x, y \in \mathbb{Z}\} = d\mathbb{Z} = \{md \mid m \in \mathbb{Z}\}.$$

d ist also die kleinste natürliche Zahl, die sich als ganzzahlige Linearkombination von a und b schreiben läßt. Wenn insbesondere a und b teilerfremd sind, hat die Gleichung $xa + yb = 1$ eine ganzzahlige Lösung.

Da laut Voraussetzung s und $(p-1)(q-1)$ teilerfremd sind, existieren x und y mit

$$xs + y(p-1)(q-1) = 1.$$

Anders geschrieben:

$$xs \equiv 1 \mod (p-1)(q-1)$$

x mit t substituiert ergibt die oben gewünschte Form. Dieses x bzw. t auszurechnen, geht sehr schnell mit Hilfe des Euklidschen Algorithmus, soferne man $(p-1)(q-1)$ kennt, was aus n nur durch Faktorisierung bzw. durch ähnlich aufwendige Methoden gewonnen werden kann. (Es kann nicht ausgeschlossen werden, daß es schnellere Algorithmen gefunden werden können.) Der einzige, der t in akzeptabler Zeit ausrechnen kann ist nur A.

Das RSA-Verfahren - Nachrichten entschlüsseln:

1. A berechnet t aus $t \cdot s \equiv 1 \mod (p-1)(q-1)$ mit Hilfe des Euklidschen Algorithmus
2. A berechnet m^t und erhält die unverschlüsselte Nachricht

Damit man RSA praktisch verwenden kann, muß man noch sicherstellen, daß man eine beliebige Potenz einer Zahl modulo n auf einem herkömmlichen Computer berechnen kann.

Satz 18. *Es existiert ein Algorithmus $A(m, e, x)$, welches für gegebenes m, e und x den Wert $A(m, e, x) \equiv x^e \mod m$ berechnet, wobei höchstens $\lfloor \log_2 e \rfloor$ Divisionen, Multiplikationen und Quadrierungen gebraucht werden und die Zwischenergebnisse stets kleiner als m^3 bleiben.*

Beweis: Man betrachte e in der Binärdarstellung, d.h.

$$e = 2^n e_n + 2^{n-1} e_{n-1} + \ldots + 2e_1 + e_0,$$

wobei $n = \lfloor \log_2 e \rfloor$ ist und $e_i \in \{0, 1\}$ gilt. So kann man die Berechnung wie folgt durchführen:

$$\begin{aligned}
x^e &= x^{2^n e_n + 2^{n-1} e_{n-1} + 2^{n-2} e_{n-2} + \ldots + 2e_1 + e_0} \\
&= x^{2^n e_n} \cdot x^{2^{n-1} e_{n-1}} \cdot x^{2^{n-2} e_{n-2}} \cdot \ldots \cdot x^{2e_1} \cdot x^{e_0} \\
&= \left((x^{e_n})^2\right)^{2^{n-1}} \cdot \left(x^{e_{n-1}}\right)^{2^{n-1}} \cdot x^{2^{n-2} e_{n-2}} \cdot \ldots \cdot x^{2e_1} \cdot x^{e_0} \\
&= \left((x^{e_n})^2 \cdot x^{e_{n-1}}\right)^{2^{n-1}} \cdot x^{2^{n-2} e_{n-2}} \cdot \ldots \cdot x^{2e_1} \cdot x^{e_0} \\
&= \left(\left((x^{e_n})^2 \cdot x^{e_{n-1}}\right)^2\right)^{2^{n-2}} \cdot \left(x^{e_{n-2}}\right)^{2^{n-2}} \cdot \ldots \cdot x^{2e_1} \cdot x^{e_0} \\
&= \left(\left((x^{e_n})^2 \cdot x^{e_{n-1}}\right)^2 \cdot x^{e_{n-2}}\right)^{2^{n-2}} \cdot \ldots \cdot x^{2e_1} \cdot x^{e_0} \\
&= \left(\ldots\left(\left((x^{e_n})^2 \cdot x^{e_{n-1}}\right)^2 \cdot x^{e_{n-2}}\right)^2 \cdot x^{e_{n-2}}\right)^2 \cdot \ldots \cdot x^{e_1}\right)^2 \cdot x^{e_0}
\end{aligned}$$

Eine mögliche Umsetzung der Berechnung in einer Meta-Programmiersprache:

1. Eingabe: e, m, x
2. Berechne n und die Bits $e_0, e_1, \ldots, e_n$, so daß
$$e = 2^n e_n + 2^{n-1} e_{n-1} + \cdots + 2e_1 + e_0 \text{ mit } e_n \neq 0$$
3. y=x
4. for $i = n-1, n-2, \ldots, 0$: $\quad y \equiv y^2 x^{e_i} \mod m$
5. Ausgabe y

n ist $\lfloor \log_2 e \rfloor$, die Schleife in Schritt 3 wird genau n mal durchlaufen und jedes mal ist eine Quadrierung, eine Division für die Modulorechnung und, falls $e_i = 1$, eine Multiplikation durchzuführen. □

Damit gerüstet kann man ein Beispiel für die RSA-Rechnung durchführen. Die verwendeten Zahlen sollen den Algorithmus veranschaulichen, für Verwendungszwecke sind diese natürlich höchst unbrauchbar.

Beispiel. *B will an A die Zahl 37 verschlüsselt senden und A will diese Nachricht entschlüsseln. Dazu wählt A z.B.* $p = 67$ *und* $q = 47$.

$$n = pq = 67 \cdot 47 = 3149,$$

$$(p-1)(q-1) = 66 \cdot 46 = 3036,$$

s könnte z.B. 13 *sein.*
A berechnet t wie folgt:

$$\begin{aligned} 3036 &= 233 \cdot 13 + 7 \\ 13 &= 1 \cdot 7 + 6 \\ 7 &= 1 \cdot 6 + 1 \end{aligned}$$

$$1 = 7-6 = 7-(13-7) = 2\cdot 7 - 13 = 2\cdot(3036 - 233\cdot 13) - 13 = 2\cdot 3036 - 467\cdot 13,$$

also

$$-467 \cdot 13 \equiv 1 \mod 3036$$

und daher ist $t = -467$ *bzw.* $t = -467 + 3036 = 2569$.

B berechnet nun $37^{13} \mod 3149$

$$13 = 8 + 4 + 1 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

also $n = 3$, $e_3 = e_2 = e_0 = 1$ *und* $e_1 = 0$

i	e_i	$y \equiv y^2 \cdot 37^{e_i} \mod 3149$	*Ergebnis*
2	*1*	$37^2 \cdot 37$	*269*
1	*0*	269^2	*3083*
0	*1*	$3083^2 \cdot 37$	*573*

Also ist $37^{13} \equiv 573 \mod 3149$

Diese Zahl kann A durch $573^t = 573^{2569} \mod 3149$ *wieder entschlüsseln und so die Botschaft lesen:*

$$\begin{aligned} 2569 &= 2048 + 512 + 8 + 1 \\ &= 1 \cdot 2^{11} + 0 \cdot 2^{10} + 1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + \\ &\quad 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \end{aligned}$$

also $n = 11$, $e_{11} = e_9 = e_3 = e_0 = 1$, *die restlichen* e_i *sind gleich 0*

i	e_i	$y \equiv y^2 \cdot 573^{e_i} \mod 3149$	*Ergebnis*
10	*0*	573^2	*833*
9	*1*	$833^2 \cdot 573$	*2508*
8	*0*	2508^2	*1511*
7	*0*	1511^2	*96*
6	*0*	96^2	*2918*
5	*0*	2918^2	*2977*
4	*0*	2977^2	*1243*
3	*1*	$1243^2 \cdot 573$	*68*
2	*0*	68^2	*1475*
1	*0*	1475^2	*2815*
0	*1*	$2815^2 \cdot 573$	*37*

Und damit hat A die nur für ihn bestimmte Nachricht.

Bemerkung. *Diese bzw. ähnliche Verschlüsselungen werden in Passwortabfragesystemen verwendet. Würde man auf einem Computer die Passwörter unverschlüsselt speichern, so würde man diese von der Festplatte lesen können. (Dies kann auf verschiedene Weisen passieren, z.B. wenn man zum Computer mit einem anderen Benutzerprofil Zugang hat (z.B. als Systemadministrator) oder zur Festplatte des Computers gelangt und diese man sich an einen anderen Rechner als eine zusätzliche Datenspeicherplatte anschließt.) Ist nun x das Passwort, so wird auf der Festplatte statt x der Wert $f(x)$ gespeichert. In diesem Fall kann man bei der Prüfung einer Eingabe des Passwortes y auf Richtigkeit so prüfen, indem man $f(y)$ mit $f(x)$ vergleicht. Dadurch kann man die Entschlüsselung sparen, die Falltür ist zum Passwortvergleich nicht notwendig.*

Anhang D

Primzahlrekorde

Eine chronologische Entwicklung der Suche nach immer größeren Primzahlen[1].

Zahl	Jahr	Finder
$2^{19}-1$	1588	Cataldi
$2^{31}-1$	1772	Euler
999999000001	1851	Looff
67280421310721	1855	Clausen
$2^{127}-1$	1876	Lucas
$934(2^{127}-1)+1$	1951	Miller und Wheeler
$(2^{148}+1)/17$	1951	Ferrier
$180(2^{127}-1)^2+1$	1951	Miller und Wheeler
$2^{521}-1$	1952	Robinson
$2^{607}-1$	1952	Robinson
$2^{1279}-1$	1952	Robinson
$2^{2203}-1$	1952	Robinson
$2^{2281}-1$	1952	Robinson
$2^{3217}-1$	1957	Riesel
$2^{4253}-1$	1961	Hurwitz und Selfridge
$2^{4423}-1$	1961	Hurwitz und Selfridge
$2^{9689}-1$	1963	Gillies
$2^{9941}-1$	1963	Gillies
$2^{11213}-1$	1963	Gillies
$2^{19937}-1$	1971	Tuckerman
$2^{21701}-1$	1978	Noll und Nickel
$2^{23209}-1$	1979	Noll
$2^{44497}-1$	1979	Nelson und Slowinski

[1] Als Quellen wurden [18] und [6] herangezogen.

$2^{86243} - 1$	1982	Slowinski
$2^{132049} - 1$	1983	Slowinski
$2^{216091} - 1$	1985	Slowinski
$391581 \cdot 2^{216193} - 1$	1989	Noll, G. Smith, Zarantonello, Brown, Parady, J. Smith
$2^{756839} - 1$	1992	Slowinski und Gage
$2^{859433} - 1$	1994	Slowinski und Gage
$2^{1257787} - 1$	1996	Slowinski und Gage
$2^{1398269} - 1$	1996	Armengaud (GIMPS)
$2^{2976221} - 1$	1997	Spence (GIMPS)
$2^{3021377} - 1$	1998	Clarkson (GIMPS)
$2^{6972593} - 1$	1999	Hajratwala (GIMPS)
$2^{13466917} - 1$	2001	Cameron (GIMPS)
$2^{20996011} - 1$	2003	Shafer (GIMPS)
$2^{24036583} - 1$	2004	Findley (GIMPS)
$2^{25964951} - 1$	2005	Nowak (GIMPS)
$2^{30402457} - 1$	2005	Cooper und Boone (GIMPS)
$2^{32582657} - 1$	2006	Cooper und Boone (GIMPS)
$2^{43112609} - 1$	2008	E. Smith (GIMPS)

Literaturverzeichnis

[1] http://www.dbgwiehl.de/prim/geschi.htm

[2] http://turnbull.mcs.st-and.ac.uk/history/HistTopics/Prime_numbers.html

[3] J. Schönbeck; Euklid; ISBN 3-764-36584-6

[4] R. Schulze-Pillot-Ziemen; Elementare Algebra und Zahlentheorie; ISBN 3-540-79569-3

[5] http://www.prothsearch.net/fermat.html

[6] http://www.mersenne.org/

[7] H. Riesel; Prime Numbers and Computer Methods for Factorization; ISBN 978-0-8176-3291-5

[8] http://primes.utm.edu/primes/search.php?Number=100

[9] C. Wagenknecht; Algorithmen und Komplexität; ISBN 3-446-22314-2

[10] L. M. Adleman, C. Pomerance und R. Rumely; On distinguishing prime numbers from omposite numbers; Annals of Math. 117: 173-206 (1983)

[11] H. Cohen und H. W. Lenstra Jr.; Primality testing and Jacobi sums; Math. Comp. 42 (1984)

[12] Edward Waring; Meditationes Algebraicae; 1770

[13] O. Ore; Number Theory and Its History; ISBN 0-486-65620-9

[14] A. Shell-Gellasch und D. Jardine; From Calculus to Computers: Using the Last 200 Years of Mathematics History in the Classroom; ISBN 0-883-85178-4

[15] J.H. Silverman; The Arithmetic of Elliptic Curves; ISBN 978-0-3879-6203-0

[16] W. R. Alford, A. Granville und C. Pomerance; There are infinitely many Carmichael numbers; Annals Math. 140 (1994), 703-722

[17] S. Müller-Stach und J. Piontkowsky; Elementare und algebraische Zahlentheorie; ISBN 978-3-8348-0211-8

[18] E. Bach und Jeffrey Shallit; Algorithmic Number Theorie; ISBN 0-262-02405-05

[19] R. Solovay und V. Strassen; A fast Monte-Carlo test for primality; SIAM J. Comp. 6(1): 84-85 (1977)

[20] R.A. Mollin; Fundamental Number Theory with Applications; ISBN 0-849-33987-1

[21] G. Miller; Riemann's Hypothesis and Tests for Primality; Journal of Computer and System Sciences anno 1976, Nr 3

[22] M. O. Rabin; Probabilistic algorithm for testing primality; Journal of Number Theory 12 (1980), no. 1, pp. 128-138

[23] J. W. Bruce; A Really Trivial Proof of the Lucas-Lehmer Test; The American Mathematical Monthly April 1993, Volume 100, No. 4

[24] Ö. J. R. Ödseth; A note on primality tests for $N = h \cdot 2n - 1$; http://www.uib.no/People/nmaoy/papers/luc.pdf

[25] D. B. Gillies; Three new Mersenne primes and a statistical theory; Math. Comp., 18 (1964) 93–95. Corrigendum in Math. Comp. 31 (1977), 1051

[26] C. Pomerance; Recent developments in primality testing; Math. Intelligencer, 3:3 (1980/81) 97–105

[27] S. Wagstaff; Divisors of Mersenne numbers; Math. Comp., 40:161 (January 1983) 385–397

[28] S. Goldwasser und J. Kilian; Almost all primes can be quickly certied; In proceedings, 18th Annual ACM Symposium on Theory of Computing, 316-329 (1986)

[29] A. Beutelspacher und U. Rosenbaum; Projektive Geometrie: Von den Grundlagen bis zu den Anwendungen; ISBN 978-3-5281-7241-1

[30] A. Werner; Elliptische Kurven in der Kryptographie; ISBN 3-540-42518-7

[31] R. Schoof; Elliptic curves over finite fields and the computation of square roots modulo p; Math. Comp. 44:483-494 (1985)

[32] L. Adleman, K. Manders und G. Miller; On taking roots in finite fields; sfcs, pp. 175-178, 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), 1977

[33] D. R. Heath-Brown; The differences between consecutive primes; J. London Math. Soc. 2(18): 7-13, (1978)

[34] K. Pracher; Primzahlverteilung (1957)

[35] A. O. L. Atkin; Schoof's algorithm; Manuscript (1986)

[36] A. O. L. Atkin; The number of points on an elliptic curve modulo a prime; Manuscript (1988)

[37] A. O. L. Atkin; The number of points on an elliptic curve modulo a prime (ii); Manuscript (1992)

[38] N. D. Elkies; Elliptic and modular curves over finite fields and related computational issues; In D. A. Buell und J. T. Teitelbaum (ed.), Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A. O. L. Atkin, AMS/IP Studies in Advanced Mathematics, 7: 21-76,AMS, IP (1998)

[39] L. M. Adleman und M. Huang; Primality Testing and Abelian Varieties over Finite Fields; Lecture Notes in Mathematics, 1512, Springer-Verlag, Berlin-Heidelberg-New York (1992)

[40] H. W. Lenstra Jr., J. Pila und C. Pomerance; A hyperelliptic smoothness test, III

[41] A. O. L. Atkin; Manuscript (1986)

[42] A. O. L. Atkin und F. Morain; Elliptic curves and primality proving; Math. Comp. 61(203): 29-68, July (1993)

[43] E. Kaltofen, T. Valente und N. Yui; An improved Las Vegas primality test; In G. H. Gonnet, editor, Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation: ISSAC 26-33, ACM Press (1989)

[44] http://www.lix.polytechnique.fr/∼morain/Prgms/getecpp.english.html

[45] F. Morain; Primality proving using elliptic curves: an update; 1998, Proc. of ANTS-III, Lecture Notes in Comput. Sci. 1423

[46] H. W. Lenstra Jr.; Factoring integers with elliptic curves; Annals of Math. 126: 649-673 (1987)

[47] M. Agrawal, N. Kayal und N. Saxena; Primes in P; Annals of Mathematics 160: 781-793 (2004)

[48] M. Nair; On Chebyshev-type inequalities for primes; Amer. Math. Monthly 89 (1982),126-129.

[49] J. von zur Gathen und J. Gerhard; Modern Computer Algebra; ISBN 978-0-5218-2646-4

[50] http://www.rsa.com/rsalabs/node.asp?id=2879

[51] R. Gupta und M. Ram Murty; A remark on Artin's conjecture; Invent. Math. 78 (1984), 127-130

[52] R. Gupta, V. K. Murty und M. R. Murty; The Euclidian algorithm for S-integers, Number Theory; (Montreal, Que., 1985), CMS Conf. Proc. 7 (1987), 189-202

[53] D. R. Heath-Brown; Artin's conjecture for primitive roots; Quart. J. Math. Oxford 37 (1986), 27-38

[54] E. Fouvry; Théorème de Brun-Titchmarsh; application au théorème de Fermat; Invent. Math. 79 (1985), 383-407

[55] R. C. Baker und G. Harman; The Brun-Titchmarsh Theorem on average, in Analytic Number Theory, Volume I; (Allerton Park, IL, 1995), Progr. Math. 138, 39-103, Birkhäuser Boston, Boston, MA, 1996

[56] H. W. Lenstra Jr. und C. Pomerance; Primality testing with gaussian periods; Private communication, March 2003

[57] R. Bhattacharjee und P. Pandey; Primality testing; Technical report, IIT Kanpur, 2001

[58] N. Kayal und N. Saxena; Towards a deterministic polynomial-time test; Technical report, IIT Kanpur, 2002

[59] http://www.aimath.org/WWN/primesinp/articles/html/50a/

[60] V. Shoup; A Computational Introduction to Number Theory and Algebra; ISBN: 978-0-5218-5154-1

[61] http://people.csail.mit.edu/rivest/Rsapaper.pdf