Primzahlen: Theorie und Anwendung

Don Zagier

Max-Planck-Institut für Mathematik, Bonn

Die Theorie der Primzahlen, die die Mathematiker seit Jahrhunderten fasziniert hat, ist voll von Paradoxen. Wir nennen hier vier von ihren scheinbaren Widersprüchen:

1. Die Primzahlen besitzen in hohem Maße die Eigenschaft der Gesetzmäßigkeit, gleichzeitig aber und genauso stark die der Willkür.

2. Man kann leicht erkennen, ob eine gegebene große Zahl prim ist, dies aber schwer beweisen.

3. Es ist möglich festzustellen, ob eine gegebene Zahl prim ist oder nicht, ohne ihre Faktoren zu kennen.

4. Man kann die Primzahlen unter einer gegebenen Grenze x zählen, ohne die einzelnen zu kennen.

Zu der ersten dieser Behauptungen möchte ich hier nicht viel sagen, sondern auf einen früheren Vortrag von mir verweisen,[1] dessen Hauptthema sie bildete. Ich entleihe jenem Vortrag nur ein Zitat

> Sie wachsen wie Unkraut unter den natürlichen Zahlen, scheinbar keinem anderen Gesetz als dem Zufall unterworfen, und kein Mensch kann voraussagen, wo wieder eine sprießen wird, noch einer Zahl ansehen, ob sie prim ist oder nicht.

und zwei Bilder (Fig. 1 und 2), die die Regelmäßigkeit sowie die Unregelmäßigkeiten der Primzahlen frappant illustrieren. Beide Bilder zeigen das Wachstum der Funktion

---

[1] "Die ersten 50 Millionen Primzahlen," in *Lebendige Zahlen*, Mathematische Miniaturen 1, Birkhäuser-Verlag, Boston–Basel 1981, 39–73.

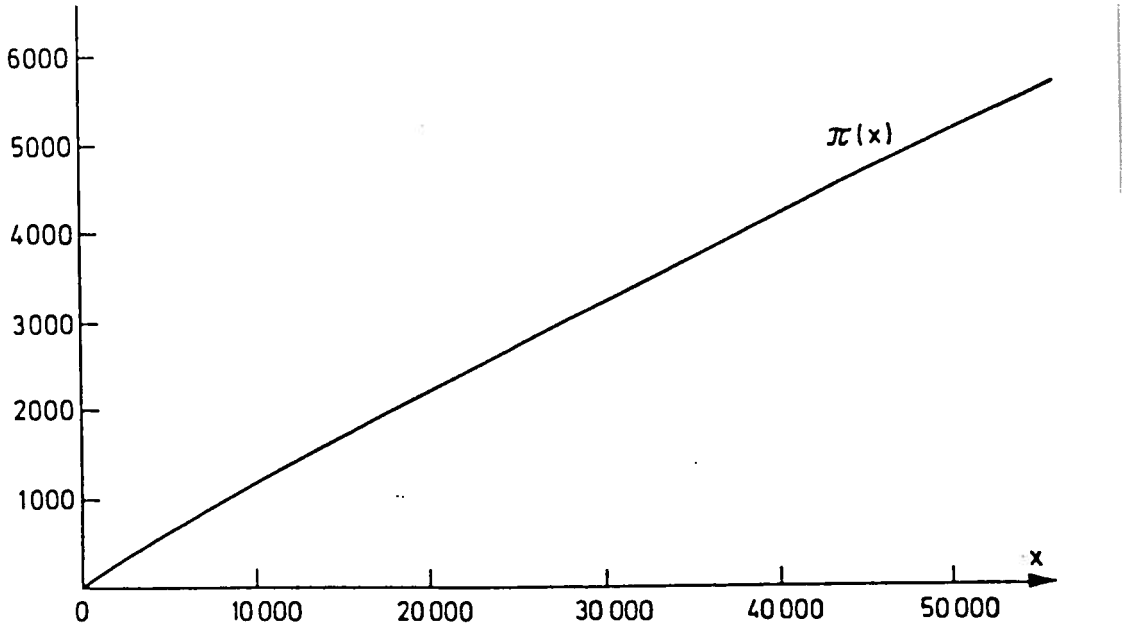

Fig. 1. $\pi(x)$, die Anzahl der Primzahlen $\leq x$, im Bereich $1 \leq x \leq 50000$

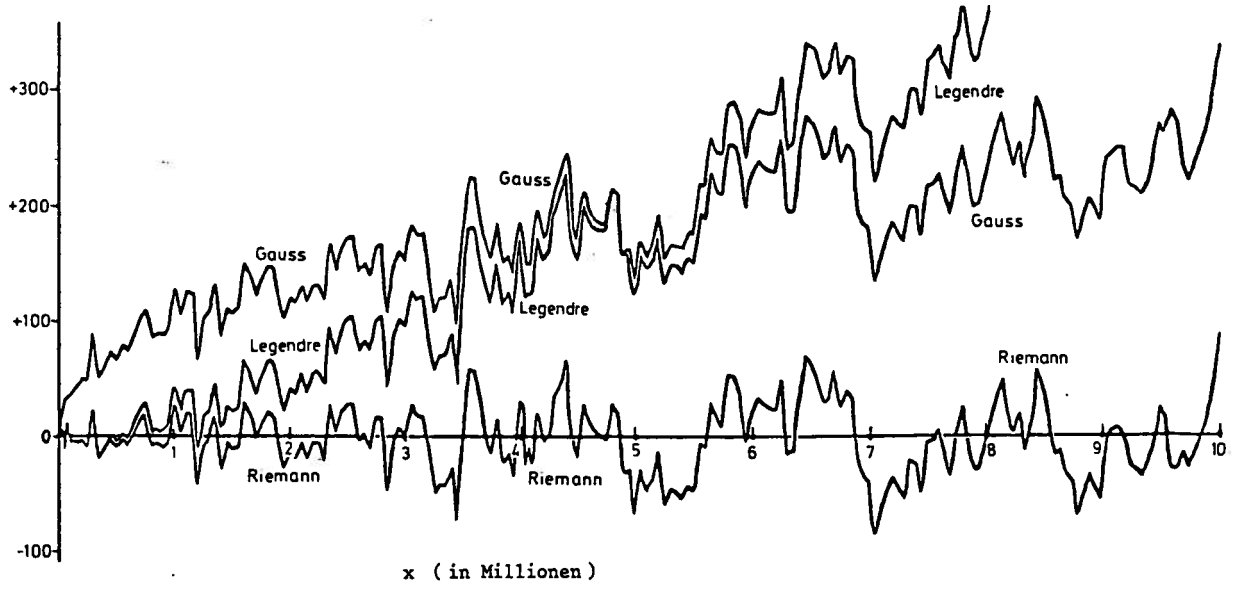

Fig. 2. Differenz zwischen $\pi(x)$ und den von Legendre, Gauß und Riemann gegebenen Approximationen. Die Approximation von Gauß lautet z.B. $\pi(x) \approx \frac{x}{\log x - 1}$. Der sogenannte Primzahlsatz, $\pi(x) \approx \frac{x}{\log x}$, gibt eine viel schlechtere Annäherung (Abweichung 44158 bei $x = 10000000$).

Funktion $\pi(x) =$ Anzahl der Primzahlen unterhalb einer Zahl x; im zweiten sieht man, wie die Funktion $\pi(x)$ im Bereich bis 10 Millionen von gewissen glatten Approximationen abweicht, die von Legendre, Gauß und Riemann vorgeschlagen wurden.

Zu der Behauptung **4.** erwähne ich einige Berechnungen von $\pi(x)$ für große Werte von x, die gemacht worden sind, ohne alle Primzahlen bis x einzeln zu bestimmen. 1885 hat Meißel die erste solche Methode erfunden und damit den Wert $\pi(1\,000\,000\,000)=50\,847\,478$ errechnet; allerdings hat er sich vertan, und es gibt tatsächlich 50 847 534 Primzahlen unter einer Milliarde, wie man heute weiß. 1958 errechnete Lehmer mit einer Varianten der Meißelschen Methode den Wert $\pi(10\,000\,000\,000)=455\,052\,512$. Auch er hat sich vertan, denn es gibt nur 455 052 511 Primzahlen unter zehn Milliarden. Schließlich haben Lagarias, Miller und Odlyzko 1985 mit Hilfe einer neuen Methode und einer sehr großen Rechenanlage den Wert $\pi(40\,000\,000\,000\,000\,000)=1\,075\,292\,778\,753\,150$ errechnet; ob auch sie sich vertan haben, ist unbekannt.

Der eigentliche Gegenstand dieses Vortrags hängt mit den Behauptungen **2.** und **3.** oben zusammen, das heißt, mit den Fragen der Primzahlerkennung und der Faktorisierung, zu denen Gauß in den *Disquisitiones Arithmeticae* (§329) schrieb:

> Problema, numeros primos a compositis diagnoscendi, hosque in factores suos primos resolvendi, ad gravissima ac utilissima totius arithmeticae pertinere, et geometrarum tum veterum tum recentiorum industriam ac sagacitatem occupavisse, tam notum est, ut de hac re copiose loqui superfluum foret.
> ["Daß das Problem, die Primzahlen von den zusammengesetzten zu unterscheiden und letztere in ihre Primfaktoren zu zerlegen, zu den wichtigsten und auch nützlichsten der ganzen Arithmetik gehört und den Fleiß und die Weisheit der Geometer der Antike und der Neuzeit beschäftigt hat, ist so bekannt, daß es überflüssig ist, viel darüber zu sagen."]

Die letzten Worte dieses Zitats treffen nicht mehr zu, denn es ist nicht so bekannt, daß sich die Mathematiker in letzter Zeit wieder sehr aktiv mit diesem alten Problem beschäftigen — einerseits, weil man Möglichkeiten entdeckt hat, zu seiner Lösung **tiefe arithmetische Theorien** einzusetzen, andererseits, weil man **unerwartete Anwendungen im Bereich der Erstellung von Geheimcodes** entdeckt hat. Bevor ich auf diese beiden Aspekte eingehe, möchte ich zur Illustration einige spezielle Primzahltypen erwähnen, die in der Geschichte eine besondere Rolle spielten.

Zahlen, die gleich der Summe ihrer echten Teiler sind (etwa $28=1+2+4+7+14$) heißen *vollkommen* und waren in der Antike wegen ihrer vermeintlichen mystischen Eigenschaften sehr beliebt. Euklid zeigte bereits, daß die Zahl $2^{n-1}p$, insofern $p=2^n-1$ prim ist, stets vollkommen ist.[2] Primzahlen der Gestalt 2^n-1 heißen *Mersennesche Primzahlen*, nach dem französischen Priester Mersenne, der 1644 die Primalität von 2^n-1 für $n=2$, 3, 5, 7, 11, 13, 17, 19, 31, 67, 127 und 257 behauptet hat. Seine Liste ist nicht ganz richtig: 67 sollte 61 heißen,

[2]Für gerade vollkommene Zahlen gilt auch die Umkehrung, wie Euler bewiesen hat. Ob es ungerade vollkommene Zahlen gibt, ist bis heute unbekannt.

89 und 107 fehlen, und $2^{257}-1$ ist nicht prim. Man kennt heute 29 Mersennsche Primzahlen.

Eine Idee, die die Mathematiker immer gelockt hat, war, eine Formel aufzustellen, die nur Primzahlen liefert. So hat der große französische Mathematiker Fermat 1650 behauptet, daß die Zahlen $F_k=2^{2^k}+1$ immer prim seien, was er immerhin für $k\leq 4$ verifiziert hat ($2^1+1=3$, $2^2+1=5$, $2^4+1=17$, $2^8+1=257$, $2^{16}+1=65537$). Seine Behauptung war besonders unglücklich: schon der allernächste Wert $2^{32}+1$ hat den kleinen Faktor 641, wie Euler bemerkte, und man glaubt heute sogar, daß alle F_k für $k\geq 5$ zusammengesetzt sind. Immerhin spielen die *Fermatschen Primzahlen* die Heldenrolle in dem berühmten Satz von Gauß, wonach das regelmäßige n-Eck genau dann mit Zirkel und Lineal konstruierbar ist, wenn n sich aus einer Zweierpotenz und einem Produkt verschiedener solcher Primzahlen zusammensetzt (also $n=3, 4, 5, 6, 8, 10, 12, 15, 16, 17, 20, 24,\ldots$).

Wir können jetzt einige Daten angeben, die ein Gefühl für die Grenzen bei Faktorisierung und Primzahlerkennung zu verschiedenen Zeitpunkten vermitteln. 1876 bewies E. Lucas, daß $2^{127}-1$ prim ist (was, wie wir gesehen haben, auch von Mersenne behauptet worden war, aber ohne Beweis). Diese 39-stellige Zahl blieb 76 Jahre lang die größte bekannte Primzahl; sie wurde erst 1952 nach der Einführung elektronischer Rechenmachinen durch die 687-stellige Zahl $2^{2281}-1$ übertroffen. Der Weltrekord—der immer von einer Mersennschen Primzahl besetzt war, weil diese besonders leicht zu testen sind—stieg danach mehrmals, er lag 1971 zum Beispiel bei $2^{19937}-1$ (6002 Ziffern) und steht heute bei $2^{216091}-1$ (65050 Ziffern). In der entgegensetzten Richtung hat man bewiesen, daß die noch viel größere Zahl F_{20} (315653 Ziffern) *nicht* prim ist, allerdings ohne daß man einen Faktor von ihr gefunden hätte (vergleiche Bemerkung 3. oben).

Wie kann man so große Zahlen auf Primalität testen? Um eine erste Antwort hierauf zu geben, müssen wir den *kleinen Fermatschen Satz*, einen der ersten und auch der wichtigsten theoretischen Resultate über Primzahlen, formulieren. Dafür brauchen wir wiederum den Begriff der *Kongruenz*, der für das ganze Weitere eine Schlüsselrolle spielt.

Ist n eine fest gewählte natürliche Zahl, so können wir Zahlen vom Standpunkt ihres Restes nach Division durch n betrachten. Wir nennen zwei Zahlen *kongruent modulo* n, falls sie denselben Rest liefern. So sind zum Beispiel zwei (positive) Zahlen genau dann kongruent modulo 100, wenn ihre beiden letzten Dezimalstellen übereinstimmen, etwa 87 und 587. Man bezeichnet Kongruenz modulo n mit dem Zeichen $\underset{n}{\equiv}$. Das wichtigste an dem Begriff der Kongruenz ist einerseits, daß man

nur endlich viele (genauer: n) Restklassen zu betrachten hat, da der Rest nach Division durch n stets unter den Zahlen 0, 1, 2,..., $n-1$ zu finden ist, andererseits, daß man mit diesen Klassen die elementaren arithmetischen Operationen (also Addition, Subtraktion, Multiplikation) ausführen kann. So ist zum Beispiel $21 \times 33 \underset{100}{\equiv} 93$, weil das Produkt je zweier Zahlen, die mit 21 bzw. 33 enden, eine Zahl mit den letzten beiden Ziffern 93 ist.

Mit diesem Begriff gewappnet, könnten wir den kleinen Fermatschen Satz aufstellen: *Ist* p *eine Primzahl und* x *eine beliebige nicht durch* p *teilbare Zahl, so ist die* (p−1)*-te Potenz* x^{p-1} *von* x *modulo* p *zu eins kongruent.* Für $p=5$ zum Beispiel gilt stets $x^4 \underset{5}{\equiv} 1$, denn die möglichen Reste von der (nicht durch 5 teilbaren) Zahl x nach Division durch 5 sind 1, 2, 3 und 4, und die vierten Potenzen $1^4=1$, $2^4=16$, $3^4=81$, $4^4=256$ enden alle mit 1 oder 6.

Als Variante des Fermatschen Satzes haben wir die Aussage, daß die $\frac{p-1}{2}$-te Potenz von x immer kongruent 1 oder $p-1$ modulo p sein muß, wobei ein tiefer Satz aus der Zahlentheorie (Quadratisches Reziprozitätsgesetz) uns a priori sagt, welche der Alternativen auftreten wird. Zum Beispiel ist $5^{(p-1)/2} \underset{p}{\equiv} 1$, falls die letzte Ziffer von p entweder 1 oder 9 ist, aber $5^{(p-1)/2} \underset{p}{\equiv} p-1$, falls p mit 3 oder 7 endet.

Den kleinen Fermatschen Satz können wir jetzt als Primalitätstest einsetzen, denn wenn wir für gegebenes p eine nicht durch p teilbare Zahl x finden, für die die Fermatsche Behauptung $x^{p-1} \underset{p}{\equiv} 1$ nicht gilt, kann p nicht prim gewesen sein. Zum Beispiel ist

$$2^{14} = 16384 = 15 \times 1092 + 4 \underset{15}{\equiv} 4 \underset{15}{\not\equiv} 1,$$

also ist die Zahl 15 nicht prim. Dies sieht als Methode ziemlich umständlich aus, weil für große Zahlen p die Zahl x^{p-1} sehr groß sein wird, aber wir können die Berechnung schnell durchführen, indem wir eine Methode verwenden, die manchmal die "russische Bauernmethode" heißt, weil sie auf dem Prinzip beruht, das angeblich die russischen Muzhiken zur Multiplikation größerer Zahlen benutzten. Sie besteht darin, durch sukzessives Quadrieren und Berechnen der Reste die Kongruenzklassen der Zahlen x^2, $x^4=(x^2)^2$, $x^8=(x^4)^2$, $x^{16}=(x^8)^2, \cdots$ modulo p zu ermitteln und sodann x^{p-1} mit Hilfe der binären Darstellung von $p-1$ als Produkt von Zahlen x^{2^r} zu berechnen. Wollen wir zum Beispiel testen, ob 101 prim ist, indem wir die Richtigkeit von $2^{50} \underset{101}{\equiv} 1$ oder 100 nachprüfen (d.h., wir benutzen die oben angegebene Variante des Fermatschen Satzes), so berechnen wir sukzessiv

$$2^1 = 2, \quad 2^2 = 4, \quad 2^4 = 4^2 = 16, \quad 2^8 = 16^2 = 256 \underset{101}{\equiv} 54,$$

$$2^{16} \underset{101}{\equiv} 54^2 = 2916 \underset{101}{\equiv} 88, \quad 2^{32} \underset{101}{\equiv} 88^2 = 7744 \underset{101}{\equiv} 68,$$

also (da 50 gleich 32+16+2 ist)

$$2^{50} \underset{101}{\equiv} 2^{32} \times 2^{16} \times 2^2 \underset{101}{\equiv} 68 \times 88 \times 4 = 23936 \underset{101}{\equiv} 100.$$

Dies ist für die kleine Zahl 101 recht kompliziert, aber der Aufwand nimmt bei größeren Zahlen p nur sehr langsam zu. So berechnet man für die 78-stellige Zahl

$$p = 2^{257} - 1 =$$

231584178474632390847141970017375815706539969331281128078915168015826259279871

die Zahl 3^{p-1} mit nur ungefähr 500 Multiplikationen (mein Laptop-Computer macht das in 8 Sekunden) und findet

$$3^{(p-1)/2} \underset{p}{\equiv}$$

379120102079384371927416061199043151230143648698746541750663412285272223158 65

Da das Ergebnis weder 1 noch $p-1$ ist, haben wir, im Gegensatz zur Mersenneschen Behauptung, bewiesen, daß p zusammengesetzt ist—ohne allerdings die geringste Ahnung zu haben, was ihre Faktoren sein könnten.

Wenn eine Zahl den Fermat-Test nicht besteht, so ist sie definitiv nicht prim. Die Umkehrung gilt nicht, z.B. ist $2^{(p-1)/2}$ für $p = 2^{257}-1$ kongruent 1 modulo p und doch ist p, wie wir gerade gesehen haben, keine Primzahl.[3] Man weiß aber, daß *für eine Nicht-Primzahl* p *der verfeinerte Fermat-Test* $x^{(p-1)/2} \underset{p}{\equiv} 1$ oder $p-1$ *für mindestens die Hälfte aller Zahlen* x *versagt*. Man kann also für gegebenes p den Test mit verschiedenen, zufällig gewählten Werten von x wiederholen, bis p entweder durchgefallen oder mit an Sicherheit grenzender Wahrscheinlichkeit als prim erkannt worden ist. (Im zweiten Fall nennt man p eine "Pseudoprimzahl" oder eine "Primzahl für industrielle Zwecke".)

Vom praktischen Standpunkt her gesehen ist es also sehr leicht, die Primalität auch einer sehr großen Zahl zu bestätigen oder zu widerlegen. Stellt sich die Zahl aber als zusammengesetzt heraus, so ist die Ermittlung ihrer Primfaktoren eine ungleich schwierigere Aufgabe. So konnten wir vorhin das Nicht-prim-sein einer 78-stelligen Zahl in 8 Sekungen auf einem Minicomputer feststellen; die entsprechende Rechenzeit auf einer Großrechenanlage wäre auch bei einer Zahl von mehreren hundert Ziffern nur ein winziger Bruchteil von einer Sekunde. Für die Faktorisierung dagegen hat man folgende sehr approximative Tabelle von

---

[3] In gewissen Fällen gilt allerdings eine partielle Umkehrung, z.B. ist eine Fermatsche Zahl $p = F_k$ genau dann prim, wenn $3^{(p-1)/2} \underset{p}{\equiv} p-1$.

erforderlichen Rechenzeiten auf den größten heute verfügbaren Computern und mit den schnellsten heute verfügbaren Methoden:

Anzahl der Ziffern	50	70	90	100
Typische Rechenzeit	1 Minute	7 Stunden	4 Tage	1 Monat

Diese Kluft zwischen der Schwierigkeit der beiden Probleme "Primzahlen erkennen" und "Zahlen faktorisieren" ist die Basis für die Verwendbarkeit der Primzahlen bei der Erstellung von Geheimcodes. Dies werden wir jetzt erläutern und danach zur Theorie der Primzahlen zurückkehren.

Die Codes, um die es geht, gehören zu einem Typ, den man "public key cryptosystem" nennt. Dies bedeutet, daß jeder Benutzer A des Systems mit einem Verfahren ϕ_A ausgestattet ist, welches Nachrichten x (etwa Ketten von Buchstaben oder von Zahlen) in andere umwandelt. Dieses Verfahren ist öffentlich bekannt, das heißt jeder interessierte Benutzer hat Zugang zu ihm (daher das Wort "public key"), und umkehrbar, das heißt es gibt ein eindeutiges Verfahren $\overleftarrow{\phi}_A$, das die verschlüsselte Nachricht $\phi_A(x)$ in x zurückverwandelt. Der springende Punkt ist, daß dieses inverse Verfahren nur A bekannt ist und von einem anderen nicht, oder nicht ohne sehr großen Aufwand, gefunden werden kann. Will nun Benutzer A an Benutzer B eine Nachricht x senden, so wendet er auf x zunächst das (nur ihm bekannte) Verfahren $\overleftarrow{\phi}_A$ und danach das (jedem bekannte) Verfahren ϕ_B an. Er schickt also die verschlüsselte Nachricht $\phi_B(\overleftarrow{\phi}_A(x))$ an B, der sie lesen kann, indem er umgekehrt erst $\overleftarrow{\phi}_B$ und danach ϕ_A anwendet. Die Kommunikation ist doppelt geschützt: A weiß, daß seine Mitteilung nur von B gelesen werden kann, weil man zu ihrer Entschlüsselung das nur B bekannte Verfahren $\overleftarrow{\phi}_B$ benötigt; B weiß, daß die Mitteilung tatsächlich von A stammt, weil nur A den ersten Schritt $\overleftarrow{\phi}_A$ hätte durchführen können.

Dieses Schema ist natürlich sehr allgemein und kann auf verschiedenste Weisen realisiert werden. Wir beschreiben eine besonders einfache Version der Realisierung mit Hilfe von Primzahlen. Jeder Benutzer wählt zwei sehr große, sagen wir 100-stellige, Primzahlen p und q und macht das Produkt $N=pq$ öffentlich bekannt. Die Zahlen p und q zu finden, ist leicht: A testet verschiedene 100-stellige Zahlen mit der oben beschriebenen Wahrscheinlichkeitsmethode, bis er zwei (Pseudo-)Primzahlen gefunden hat; dies wird nicht allzu lange dauern, da fast 1% der ungeraden 100-stelligen Zahlen prim sind und der Test jeweils nur einen Bruchteil von einer Sekunde dauert. Für die Version, die wir beschreiben, nehmen wir an, daß A nur unter Zahlen der Gestalt $3k+2$ sucht, also $p=3a+2$, $q=3b+2$. Das

Codierungsverfahren von A lautet jetzt:

(1) $\qquad \phi_A(x) = \text{Rest von } x^3 \text{ nach Division durch } N,$

wobei die ursprüngliche Nachricht eine Zahl x zwischen 0 und $N-1$ sein soll, was man erreichen kann, indem man Buchstaben in Ziffern umsetzt und die entstehende Ziffernfolge in Blöcke der Länge 100 einteilt. Da die Zahl N und die Formel (1) öffentlich bekannt sind, kann jeder das A'sche Verfahren $x \mapsto \phi_A(x)$ nachvollziehen. Das inverse Verfahren zu (1) lautet aber

(2) $\qquad \overleftarrow{\phi}_A(y) = \text{Rest von } y^M \text{ nach Division durch } N,$

wobei $M = 6ab+2a+2b+1$ ist. Da nur A die beiden Primfaktoren $p = 3a+2$ und $q = 3b+2$ von N kennt, kann nur er das Verfahren $y \mapsto \overleftarrow{\phi}_A(y)$ durchführen. Damit leisten die durch (1) und (2) festgelegten Prozesse das Gewünschte, insofern sie tatsächlich zueinander invers sind, was wir jetzt als kleine Übung in der Benutzung des Fermatschen Satzes verifizieren:

$$\overleftarrow{\phi}_A\big(\phi_A(x)\big) \underset{N}{\equiv} \big(\phi_A(x)\big)^M \underset{N}{\equiv} \big(x^3\big)^M = x^{3M} = x^{2(p-1)(q-1)+1} \underset{N}{\equiv} x,$$

weil

$$x^{p-1} \underset{p}{\equiv} 1 \;\Rightarrow\; x^{2(p-1)(q-1)+1} = (x^{p-1})^{2(q-1)} \cdot x \underset{p}{\equiv} x,$$

$$x^{q-1} \underset{q}{\equiv} 1 \;\Rightarrow\; x^{2(p-1)(q-1)+1} = (x^{q-1})^{2(p-1)} \cdot x \underset{q}{\equiv} x,$$

und weil zwei Zahlen, die modulo den Primzahlen p und q kongruent sind, automatisch auch modulo dem Produkt $N=pq$ kongruent sind.

Wir kommen jetzt zu der arithmetischen Theorie zurück. Da der Vortrag eigentlich über Primzahlen und nicht über zusammengesetzte Zahlen geht, werden wir uns dabei auf Primalitätstests statt auf Faktorisierungsmethoden konzentrieren. Wie wir gesehen haben, kann man in vernachlässigbar kurzer Rechenzeit die Nichtprimalität einer Zahl nachweisen oder sich umgekehrt mit Pseudoprimalitätsmethoden von der Primalität einer Zahl überzeugen. Für den Mathematiker ist das aber natürlich nicht gut genug. Wie kann man von einer wirklich großen Zahl *beweisen*, daß es sich um eine Primzahl handelt? Wie wir am Anfang des Vortrags erwähnt haben, hat man zur Lösung dieser Aufgabe in den allerletzten Jahren neue Ansätze entdeckt, die auf schwierigen und weittragenden Hilfsmitteln aus der Zahlentheorie basieren. Diese Anwendbarkeit tiefliegender Theorien auf das algorithmische und elementar anmutende Problem der Primzahlerkennung war verblüffend und aufsehenerregend.

Heute sind zwei Methoden zur schnellen Primalitätsprüfung bekannt, die auf solchen höheren Theorien aufbauen. Die erste wurde von Adleman-Pomerance-

Rumely 1980 erfunden und von Cohen und Lenstra 1982 verfeinert und in einen wirklich praktischen Algorithmus umgesetzt. Diese APR/CL-Methode basiert auf den sogenannten höheren Reziprozitätsgesetzen aus der Klassenkörpertheorie. Das sind Sätze, deren Formulierungen (geschweige denn Beweise) wir hier nicht geben können. Es handelt sich um weitgehende Verallgemeinerungen des im Zusammenhang mit dem Fermat-Test erwähnten Gaußschen Reziprozitätsgesetzes, die, ganz grob gesagt, es erlauben, nicht nur die Klasse von $x^{(p-1)/2}$ modulo p, sondern auch die Klasse von $x^{(p-1)/d}$ modulo p für verschiedene Teiler d von $p-1$ vorauszusagen. Diese zusätzliche Freiheit verschafft uns irgendwann soviel Information über die Zahl p, daß wir sie als Primzahl erkennen können.

Typische Rechenzeiten für die APR/CL-Methode werden durch folgende Tabelle gegeben.

Anzahl der Ziffern	100	200	500
Typische Rechenzeit	20 Sekunden	2 Minuten	1 Stunde

Wie man sieht, sind sie viel länger als die Millisekunden, die man für den Pseudoprimalitätstest braucht, aber um viele Größenordnungen kürzer als die zur Faktorisierung erforderlichen Zeiten, die wir in der ersten Tabelle angegeben haben. Die theoretische Analyse der Methode zeigt, daß sie fast polynomial in der Anzahl der Ziffern der zu testenden Zahl p ist (genauer: die Rechenzeit wächst asymptotisch wie $(\log p)^{C \log\log\log p}$ für eine gewisse Konstante C).

Die zweite Methode ist noch jüngeren Datums: sie wurde 1986 von Goldwasser und Kilian und — in der Gestalt, die wir beschreiben wollen — von Atkin entwickelt. Sie benutzt nicht die Reziprozitätsgesetze aus der Klassenkörpertheorie, sondern die Theorie der elliptischen Kurven, ein sehr aktuelles Forschungsgebiet, das sich im übrigen nach Entdeckungen von Lenstra und anderen auch auf das Problem der Faktorisierung gut anwenden läßt. Die GK/A-Methode ist in der heutigen Implementierung etwas langsamer als die APR/CL-Methode (die geschätzte Rechenzeit für eine 500-stellige Primzahl wäre ungefähr 10 Stunden), wäre aber für genügend große Zahlen irgendwann schneller, da die Rechenzeit asymptotisch wie eine feste Potenz (ungefähr die fünfte) von $\log p$ wächst. Im Rest des Vortrags geben wir eine stark vereinfachte Beschreibung des GK/A-Algorithmus, wobei die mathematischen Ansprüche etwas höher sein werden als bisher.

Erst erläutern wir aber eine andere Methode, die viel einfacher ist, allerdings nur dann zum Erfolg führt, wenn man die volle Faktorisierung der Zahl $p-1$ kennt. Nehmen wir zum Beispiel an, wir wollen die Primalität der Zahl $p=577$ beweisen.

Durch sukzessives Teilen durch 2 und 3 stellt man fest, daß $p-1$ nur diese beiden Primfaktoren hat: $p-1=576=2^6\times 3^2$. Mit der russischen Bauernmethode findet man schnell

$$5^{p-1} = 5^{576} = 5^{512}\times 5^{64} \underset{p}{\equiv} 546\times 335 \underset{p}{\equiv} 1,$$

$$5^{(p-1)/2} = 5^{288} = 5^{256}\times 5^{32} \underset{p}{\equiv} 435\times 256 \underset{p}{\equiv} 576 \underset{p}{\not\equiv} 1,$$

$$5^{(p-1)/3} = 5^{192} = 5^{128}\times 5^{64} \underset{p}{\equiv} 287\times 335 \underset{p}{\equiv} 363 \underset{p}{\not\equiv} 1$$

Wenn man also sukzessiv die Klassen modulo p von 5^1, 5^2, 5^3,...,5^n,... berechnet, so wiederholen sie sich immer nach 576 Schritten, weil $5^{n+576} \underset{p}{\equiv} 5^n\times 5^{576} \underset{p}{\equiv} 5^n$, aber nicht nach weniger Schritten, da jeder echte Teiler von 576 ein Teiler von 288 oder 192 ist und die Potenzen 5^{288} und 5^{192} noch nicht zu 1 modulo p kongruent sind. Es folgt, daß die Klassen von 5^1, 5^2,...,5^{p-1} modulo p alle verschieden sind. Aber dann muß 577 prim sein, denn die Anzahl der zu einer Zahl p teilerfremden Klassen modulo p ist offenbar nur dann gleich $p-1$, wenn p prim ist (sonst würden unter den Klassen $1, 2, \ldots, p-1$ mindestens die Teiler von p entfallen). Der Nachteil der Methode ist klar: sie funktioniert nur, wenn wir die Zahl $p-1$, die ja kaum kleiner als p ist, voll faktorisieren können. Bei der speziell gewählten Zahl 577 hatten wir damit Glück, im allgemeinen aber haben wir mit unserem Verfahren das Problem des Primalitätsnachweises nur auf das viel schwierigere Problem der Faktorisierung "reduziert".

Um diese Schwierigkeit zu umgehen, muß man das Verfahren von einem abstrakteren Standpunkt aus interpretieren. Was wir eigentlich gebraucht haben, war nur, daß die zu einer Primzahl p teilerfremden Klassen modulo p eine *Gruppe* bilden, das heißt, es gibt eine Operation (hier natürlich die Multiplikation), die aus jeweils zwei Klassen eine dritte produziert. Von diesem Standpunkt aus gesehen bestand unsere Schwierigkeit darin, daß diese "multiplikative Gruppe" durch p eindeutig bestimmt ist und ihre Ordnung (Anzahl der Elemente) gleich der eben nicht unbedingt hoch faktorisierbaren Zahl $p-1$. Wenn wir aber andere Gruppen hätten — also andere endliche Mengen, die unter einer binären Operation abgeschlossen sind und auch die sonstigen Gruppenaxiome erfüllen — so könnten wir hoffen, unter ihnen solche zu finden, deren Ordnung sich faktorisieren läßt; dann würde dieselbe Beweisidee wie im Fall der multiplikativen Gruppe zu einem Beweis der Primalität von p führen. Solche Gruppen werden eben durch die elliptischen Kurven geliefert.

Die elliptischen Kurven sind Strukturen, die durch eine Gleichung

(3) $$y^2 = x^3 + Ax + B \qquad (A, B \text{ feste Konstanten})$$

gegeben werden können. Die Punkte der Kurve sind die Paare (x,y), die die Glei-

chung erfüllen, zusammen mit einem zusätzlichen Punkt "∞". Erstaunlicherweise bilden sie eine Gruppe unter dem Gesetz: sind P und Q Punkte der Kurve, so ist $P+Q$ der Punkt, den man erhält, wenn man den dritten Schnittpunkt der Verbindungsgeraden $\overline{PQ}$ mit der Kurve bildet und diesen an der x-Achse spiegelt (s. Figur 3).

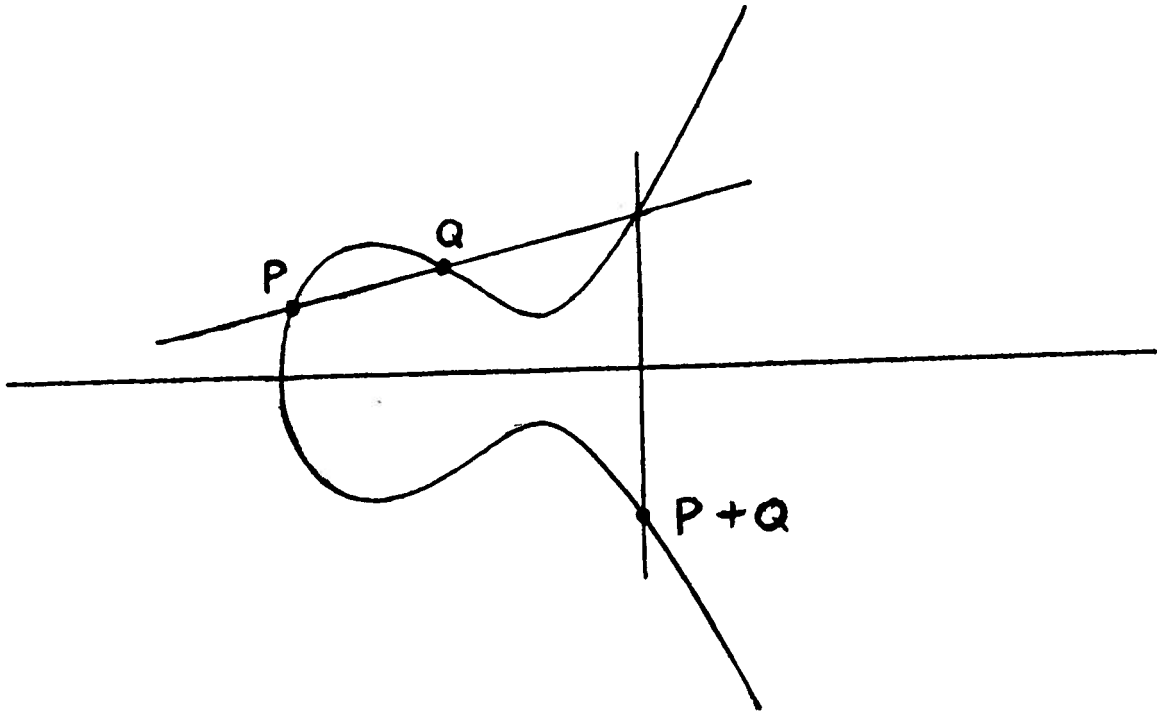

Fig. 3. Additionsgesetz auf einer elliptischen Kurve

Wie man leicht nachrechnet, wird diese geometrisch definierte Gruppenoperation durch die Formel[4]

$$(x_1,y_1)+(x_2,y_2)=(x_3,y_3) \quad \text{mit} \quad x_3=\left[\frac{y_2-y_1}{x_2-x_1}\right]^2-x_1-x_2, \quad y_3=-\frac{y_2-y_1}{x_2-x_1}(x_3-x_1)-y_1$$

(zusammen mit den Ergänzungsgesetzen $(x,y)+\text{"}\infty\text{"}=(x,y)$ und $(x,y)+(x,-y)=\text{"}\infty\text{"}$) beschrieben. Diese rein algebraische Schreibweise erlaubt es, das Ganze in die Welt der Restklassen modulo einer Primzahl p zu transponieren. Somit erhalten wir eine neue, von der multiplikativen Gruppe verschiedene endliche Gruppe und sogar — wenn wir die Koeffizienten A und B in (3) variieren — ganz viele.

Dies würde allerdings nichts nutzen, wenn wir die Ordnungen unserer Gruppen nicht kennen würden. Im allgemeinen ist die Bestimmung der Anzahl der Elemente

---

[4]Im Falle $(x_1,y_1)=(x_2,y_2)$ muß man in dieser Formel den sinnlosen Ausdruck $\frac{y_2-y_1}{x_2-x_1}$ durch $\frac{3x_1^2+A}{2y_1}$ ersetzen.

auf einer elliptischen Kurve modulo p ein schwieriges Problem. Es gibt aber eine spezielle Klasse, für die man die Antwort kennt. Das sind die sogenannten CM-Kurven ("CM" steht für die englischen Worte "complex multiplication", die wir nicht weiter erläutern). Ohne auf die Definition einzugehen, sagen wir nur, daß das gewisse elliptische Kurven E_d $(d=1,2,3,\ldots)$ sind, deren Gleichungen explizit bekannt sind, zum Beispiel

$$E_4\colon\ y^2 = x^3 - x,$$

$$E_7\colon\ y^2 = x^3 - 35x + 98,$$

$$E_5\colon\ y^2 = x^3 - (30+9\sqrt{5})x + (56+36\sqrt{5}).$$

Wie man sieht, sind die Koeffizienten nicht unbedingt gewöhnliche, sondern manchmal auch algebraische Zahlen, was aber auch in der Welt modulo p einen Sinn hat (zum Beispiel kann man modulo 11 für $\sqrt{5}$ die Zahl 7 nehmen, da $7^2=49 \underset{11}{\equiv} 5$). Die algebraischen Zahlen, die in der Gleichung von E_d vorkommen, haben einen Grad h_d, der die *Klassenzahl* von d heißt und die man immer ausrechnen kann (in unseren Beispielen oben ist $h_4=1$, $h_7=1$, $h_5=2$). Das Besondere an den CM-Kurven ist, daß für gewisse "gute" Primzahlen die Anzahl der Restklassen modulo p, die die Gleichung von E_d erfüllen, durch eine einfache Formel gegeben wird. Diese guten Primzahlen sind die Primzahlen der Gestalt

$$p = a^2 + db^2 \tag{4}$$

und die besagte Formel für die Anzahl $N_d(p)$ von Punkten auf E_d modulo p lautet[5]

$$N_d(p) = p - 2a + 1. \tag{5}$$

Zum Beispiel ist $N_4(13)=8$, da die Gleichung $y^2 \underset{13}{\equiv} x^3 - x$ die 8 Lösungen $(x,y)=(0,0)$, $(1,0)$, $(5,4)$, $(5,9)$, $(8,6)$, $(8,7)$, $(12,0)$ und "∞" hat, und tatsächlich ist $13=3^2+4\times1^2$, $13-6+1=8$.

Die guten Primzahlen bilden einen positiven Prozentsatz von der Gesamtheit aller Primzahlen, genauer, sie haben in der Menge aller Primzahlen die Dichte $\frac{1}{2h_d}$. So gilt in unseren Beispielen

$$\begin{aligned} &p \text{ ist gut für } E_4 \iff p = a^2+4b^2 \iff p \underset{4}{\equiv} 1 \quad (50\% \text{ aller Primzahlen}) \\ &p \text{ ist gut für } E_7 \iff p = a^2+7b^2 \iff p \underset{14}{\equiv} 1,9 \text{ oder } 11 \quad (50\%), \\ &p \text{ ist gut für } E_5 \iff p = a^2+5b^2 \iff p \underset{20}{\equiv} 1 \text{ oder } 9 \quad (25\%). \end{aligned} \tag{6}$$

Umgekehrt gibt es für eine gegebene Primzahl p viele Kurven E_d, für die p gut ist. Außerdem kann man mit bekannten Algorithmen die Darstellung einer guten Primzahl

[5] In Formeln (4) und (5) ist die Zahl a nicht unbedingt positiv zu nehmen; es gibt aber ein einfaches Rezept, um zwischen $|a|$ und $-|a|$ richtig zu wählen.

p in der Gestalt (4) schnell finden, auch wenn p sehr groß ist.

Sei p die Zahl, deren Primalität wir nachweisen wollen. Man probiert dann der Reihe nach kleine Zahlen d (oder noch besser: Zahlen d, für die h_d klein ist), bis man eine findet, für die gilt:

(i) p ist gut für E_d (dies ist mit Kriterien wie in (6) leicht nachzuprüfen) und

(ii) die mit Hilfe der Formeln (4) und (5) berechnete Anzahl $N_d(p)$ hat eine Faktorisierung $N_d(p) = sq$ mit s nicht allzugroß und q prim.

Da es unter großen Zahlen verhältnismäßig viele gibt, die das Produkt von einer relativ kleinen Zahl und einer Primzahl sind, wird man mit hoher Wahrscheinlichkeit ein solches d finden können. Natürlich wird man in (ii) zunächst nur verifizieren können, daß q eine Pseudoprimzahl ist; wir können aber ihre Primalität vorläufig annehmen, den Primalitätstest für p unter dieser Annahme zuende durchführen und anschließend mit genau derselben Methode die Primalität der viel kleineren Zahl q nachweisen (bootstrapping).

Wir wählen jetzt einen beliebigen Punkt P auf der Kurve E_d modulo p und berechnen (mit der russischen Bauernmethode, also mittels sukzessivem Verdoppeln) den Punkt $Q=sP$, das heißt, die s-fache Summe $P+P+\ldots+P$, wobei die Addition die der elliptischen Kurve ist. Wir berechnen dann, wieder nach der russischen Bauernmethode, das Vielfache qQ. Wenn unsere Zahl p tatsächlich prim war, so muß für qQ das Nullelement "∞" der Gruppe E_d modulo p herauskommen, da dann die Zahl sq tatsächlich die Ordnung dieser Gruppe ist. Nehmen wir an, dies sei der Fall (sonst haben wir die Primalität von p wiederlegt). Weil q prim ist, hat Q die *genaue* Ordnung q. Wir wissen noch nicht, daß die Ordnung der Gruppe E_d modulo p gleich der durch (5) errechneten Zahl $N_d(p)$ ist, da diese Formel nur für Primzahlen gilt. Die Existenz von Q impliziert aber, daß diese Ordnung durch q teilbar ist. Wenn p nicht prim, sondern das Produkt von Primzahlen $p_1,\ldots,p_n$ ist, so ist die richtige Ordnung der Kurve E_d modulo p gleich dem Produkt $N_d(p_1)\ldots N_d(p_m)$, wobei $N_d(p_i)$ jeweils durch Formel (5) gegeben wird. Da die Primzahl q diese Ordnung teilt, muß mindestens einer der Faktoren, etwa $N_d(p_1)$, durch q teilbar sein. Dann ist $N_d(p_1)$ mindestens so groß wie q und damit auch p_1 nicht wesentlich kleiner als q, da (4) und (5) implizieren, daß p_1 und $N_d(p_1)$ ungefähr gleich groß sind. Dies impliziert aber, daß das Produkt der restlichen Faktoren $p_2\ldots p_n = p/p_1$ nicht wesentlich größer als $p/q \approx N_d(p)/q = s$ ist. Damit sind wir fertig, denn s wurde als "nicht allzugroß" vorausgesetzt und wir können annehmen, daß die Existenz von kleinen Faktoren von p durch direktes Suchen schon vorher ausgeschlossen wurde.

Beispiel: Sei $p=9241$ die zu testende Zahl. Sie ist pseudoprim (z.B. ist $2^{9240/2} \underset{p}{\equiv} 1$). Da $p \underset{4}{\equiv} 1$, muß p nach (6) als a^2+4b^2 darstellbar sein, und tatsächlich

ist $9241 = 5^2 + 4 \times 48^2$. Die Formel (5) gibt jetzt den Wert

$$N_4(p) = 9241 - 10 + 1 = 9232 = 16 \times 547$$

für die Ordnung der Kurve E_4 modulo p, falls p tatsächlich prim ist. Die Zahl 16 ist nicht allzu groß, und die Primalität von 577 haben wir schon nachgewiesen. Der Punkt $Q = (-19, 49)$ liegt auf der Kurve E_4 modulo p, da $49^2 \underset{9241}{\equiv} (-19)^3 - (-19)$. Außerdem ist $577Q = 0$, denn man findet mit der russischen Bauernmethode sukzessiv

$$2Q = (4536, -2249),\ 4Q = (1296, -1145), \ldots,\ 64Q = (2482, -338), \ldots, 512Q = (-2009, 2197),$$

$$576Q = 512Q + 64Q = (2482, -338) + (-2009, 2197) = (-19, -149) = -Q.$$

(Den Punkt Q haben wir als $sP = 16P$ gefunden, wobei $P = (-527, 155)$ ein zufällig gewählter Punkt auf E_4 modulo 9241 war.) Da 577 prim ist, ist die Ordnung von Q genau 577, also ist die Ordnung von E_4 modulo p durch 577 teilbar und p muß mindestens einen Primfaktor p_1 haben, für den $N_a(p_1)$ durch 577 teilbar ist. Dann gilt

$$577 \leq N_a(p_1) \leq p_1 + 2\sqrt{p_1} + 1, \qquad p_1 \geq 530, \quad p/p_1 \leq 9241/530 < 18.$$

Da 9241 keine Faktoren kleiner als 18 hat, muß sie tatsächlich prim sein.

Für den Leser, der mehr über die in diesem Vortrag angeschnitten Themen erfahren möchte, empfehlen wir den sehr schönen Übersichtsartikel "Elliptic curves and number theoretic algorithms" von H. Lenstra in den Proceedings des Berkeley Internationalen Mathematiker-Kongresses (AMS, 1986) sowie das Buch "Prime Numbers and Computer Methods for Factorization" von H. Riesel (Progress in Math. 57, Birkhäuser-Verlag, Basel 1985), das nicht nur die Theorie, sondern auch eine Fülle von numerischem Material bringt.