III. Quantencomputer

Die Quantentheorie ist nicht nur in der Lage, Kryptographie sicherer zu machen, sie stellt potentiell auch eine ernste Bedrohung existierender Kryptoverfahren dar, die mit einem sogenannten Quantencomputer möglicherweise leicht gebrochen werden können. Besonders gefährdet sind die sogenannten asymmetrischen oder *public key*-Verfahren, die wir daher (auch wenn sie zur Gegenwart gehören) zumindest kurz rekapitulieren müssen.

1. Kryptographie mit öffentlichen Schlüsseln

In der klassischen Kryptographie gibt es zu jedem Kryptoverfahren einen Schlüssel, dessen Kenntnis sowohl zur Verschlüsselung als auch zur Entschlüsselung notwendig ist. 1976 veröffentlichten MARTIN HELLMANN, Elektrotechniker an der Stanford University, und WHITFIELD DIFFIE, ein *graduate student,* in den *IEEE Transactions on Information Theory* eine Arbeit mit dem Titel *New directions in cryptography,* in der sie behaupteten, man könne auch Verfahren entwickeln, bei denen zwar jedermann eine Nachricht verschlüsseln könne, aber trotzdem nur der legitime Empfänger in der Lage sei, sie zu *ent*schlüsseln. Inzwischen ist bekannt, daß in Großbritannien COCKS und ELLIS bereits zwei Jahre vorher ein entsprechendes System entwickelten, das aber aus Gründen der „nationalen Sicherheit“ nie veröffentlicht wurde.

DIFFIE und HELLMANN stellten noch kein explizites derartiges Verfahren vor, aber zwei Jahre später publizierten RON RIVEST, ADI SHAMIR und LEONARD ADLEMAN vom MIT in den *Communications of the ACM* ihre Arbeit *A Method for Obtaining Digital Signatures and Public Key*

Cryptosystems, die das heute nach ihren Initialen als RSA bekannte und kommerziell sehr erfolgreiche Kryptosystem vorstellte.

Ausgangspunkt ist die Beobachtung, daß es zwar sehr leicht ist, zwei Zahlen miteinander zu multiplizieren, daß aber die inverse Operation, die Faktorzerlegung, erheblichen Aufwand erfordert.

Man wähle sich daher zwei große Primzahlen p und q und berechne deren Produkt N; außerdem wähle man einen zu $(p-1)(q-1)$ primen Exponenten e zwischen 2 und $N-2$. Eine zu übermittelnde Nachricht wird aufgespalten in Blöcke, die sich durch Zahlen zwischen 0 und $N-1$ kodieren lassen, und ein solcher Block a wird chiffriert als

$$b = a^e \bmod N ,$$

was sich auch für große Werte von N und e noch ziemlich effizient berechnen läßt.

Zur Berechnung von a aus b beachte man, daß für die beiden Primzahlen p und q

$$x^p \equiv x \bmod p \quad \text{und} \quad x^q \equiv x \bmod q \quad \text{für alle } x$$

gilt: Nach dem binomischen Lehrsatz ist $(u+v)^p$ gleich

$$u^p + \binom{p}{1} u^{p-1} v + \binom{p}{2} u^{p-2} v^2 + \cdots + \binom{p}{p-2} u^2 v^{p-2} + \binom{p}{p-1} u v^{p-1} + v^p ,$$

und hierin sind wegen

$$\binom{p}{k} = \frac{p!}{k!(p-k)!}$$

alle Binomialkoeffizienten mit $1 \le k \le p-1$ durch p teilbar. Somit ist

$$(u+v)^p \equiv u^p + v^p \bmod p ,$$

woraus die Behauptung leicht durch vollständige Induktion folgt.

Falls x nicht durch p oder q teilbar ist, ist dann aber auch

$$x^{p-1} \equiv 1 \bmod p \quad \text{und} \quad x^{q-1} \equiv 1 \bmod q ,$$

also auch

$$x^{(p-1)(q-1)} \equiv 1 \bmod p \quad \text{und} \quad x^{(p-1)(q-1)} \equiv 1 \bmod q$$

und somit

$$x^{(p-1)(q-1)} \equiv 1 \bmod N .$$

Allgemeiner gilt für jede ganze Zahl k

$$x^{k(p-1)(q-1)} \equiv 1 \bmod N .$$

Multiplikation mit x führt auf

$$x^{k(p-1)(q-1)+1} \equiv x \bmod N ,$$

wobei diese Formel wieder für *alle* x gilt.

Nach Voraussetzung ist e teilerfremd zu $(p-1)(q-1)$; mit dem EUKLIDischen Algorithmus lassen sich also ganze Zahlen d und k finden, so daß

$$de + k(p-1)(q-1) = 1$$

ist. Da sich diese Gleichung nicht ändert, wenn man ein Vielfaches der Gleichung

$$(p-1)(q-1)e - e(p-1)(q-1) = 0$$

addiert, läßt sich leicht erreichen, daß dabei $d > 0$ ist. Alsdann ist

$$b^d = a^{ed} = a^{1-k(p-1)(q-1)} = a \cdot a^{-k(p-1)(q-1)} \equiv a \cdot 1 = a \bmod N .$$

Veröffentlicht man also N und e, so kann damit jedermann Nachrichten verschlüsseln; zum Entschlüsseln benötigt man aber d, in dessen Berechnung wiederum die Zahl $(p-1)(q-1)$ einging, deren Kenntnis gleichbedeutend mit der von p und q ist. d wäre hier also ein zusätzlicher *geheimer* Schlüssel.

Dieser kann auch zur Erzeugung elektronischer Unterschriften benutzt werden: Als elektronische Unterschrift zur Nachricht a mit $2 \le a \le N-2$ dient

$$u = a^d \bmod N .$$

Jedermann kann nachrechnen, daß

$$u^e = a^{de} \equiv a \bmod N$$

ist und somit die Unterschrift überprüfen; erzeugen läßt sie sich nur mit Kenntnis von d.

Von DIFFIE und HELLMANN stammt ein Verfahren zur Schlüsselvereinbarung über eine unsichere Leitung. Auch hier geht es um eine Umkehrung der Funktion $a^e \bmod N$, nun aber bezüglich des Exponenten: Für eine Primzahl p finde man für festes a zwischen zwei und $p-2$ und gegebenes b

einen Exponenten e, für den $b \equiv a^e \bmod p$ ist, den sogenannten diskreten Logarithmus von b zur Basis a modulo p. Es gibt eine ganze Reihe von Algorithmen, mit denen sich e berechnen läßt; sie sind aber allesamt deutlich aufwendiger als die Berechnung von $a^e \bmod p$.

Davon ausgehend, schlagen DIFFIE und HELLMANN folgendes Verfahren vor: Die beiden Partner A und B einigen sich auf eine Primzahl p und auf $2 \le a \le p-2$; sodann erzeugt A eine Zufallszahl u, berechnet die Potenz $x = a^u \bmod p$ und schickt diese an B; dieser erzeugt eine Zufallszahl v, berechnet $y = a^v \bmod p$ und schickt dies an A. Mit dieser Information kann jeder der beiden die Zahl

$$S = a^{uv} \bmod p = x^v \bmod p = y^u \bmod p$$

berechnen, ein Lauscher, der nur p, a, x und y kennt, aber weder u noch v, kann dies nicht. S kann also als Schlüssel dienen.

Man beachte den großen Unterschied zwischen dieser Form der Schlüsselvereinbarung und dem entsprechenden Algorithmus auf Grundlage der Quantenkryptographie: Letzterer ist sicher auch gegenüber einem Gegner mit unbegrenzten rechnerischen Ressourcen; das hier vorgestellte Verfahren aber bietet gegenüber einem solchen Gegner keinerlei Schutz, da er beispielsweise aus a und x problemlos u durch systematisches Ausprobieren berechnen könnte.

Auch das RSA-Verfahren bietet keinerlei Schutz gegen einen solchen Gegner, denn für ihn ist auch Faktorisieren kein Problem. Der Schutz der hier vorgestellten Verfahren beruht also ausschließlich darauf, daß ein Gegner mit *beschränkten* rechnerischen Möglichkeiten eine prinzipiell lösbare mathematische Aufgabe nicht in akzeptabler Zeit lösen kann.

Quantencomputer, so sie je existieren, können die rechnerischen Möglichkeiten eines Gegners so dramatisch erweitern, daß alle diese Verfahren wertlos werden könnten.

2. Quantenmechanische Grundlagen

Zum Verständnis eines Quantencomputers müssen wir uns etwas stärker mit dem formalen Apparat der Quantenmechanik befassen als dies bei der Quantenkryptographie notwendig war.

Grundlage der Beschreibung eines quantenmechanischen Systems ist immer sein *Zustandsraum,* ein Vektorraum über dem Körper $\mathbb{C}$ der komplexen Zahlen mit einem HERMITEschen Skalarprodukt. Dieses Produkt wird in der Form $\langle u|v\rangle$ geschrieben, und von daher hat sich die DIRACsche Konvention eingebürgert, die Elemente von V in der Form $|v\rangle$ zu schreiben. Da $\langle u|v\rangle$ eine *bracket* ist, wird $|v\rangle$ allein meist kurz als ein *ket* bezeichnet.

Diese Schreibweise mag einem Mathematiker auf den ersten Blick verwirrend erscheinen; er sollte sich aber klarmachen, daß es hier nur um ein Symbol geht und sich an der Mathematik natürlich nichts ändert, egal ob man einen Vektor als $\mathfrak{v}$, $\vec{v}$ oder eben $|v\rangle$ schreibt.

Die Vektoren aus V beschreiben die möglichen Zustände des Systems, wobei zueinander proportionale Vektoren denselben Zustand beschreiben; der tatsächliche Zustandsraum ist also der projektive Raum $\mathbb{P}(V)$. Da es somit auf die Länge des Zustandsvektors nicht ankommt, wird diese vielfach auf eins normiert.

Meßbare physikalische Größen werden durch HERMITEsche Operatoren $A\colon V \to V$ beschrieben. Ist das System vor der Messung im Zustand, der durch den Einheitsvektor $|v\rangle$ beschrieben wird, so ist sein Zustand nach der Messung ein Eigenvektor von A und der Meßwert ist der zugehörige Eigenwert. Zu einem HERMITEschen Operator gibt es bekanntlich immer eine Orthonormalbasis $\mathfrak{B}$ von V aus Eigenvektoren von A; falls die zugehörigen Eigenwerte alle verschieden sind, ist die Wahrscheinlichkeit für den Eigenwert λ und den zugehörigen Eigenvektor $|w\rangle$ der Länge eins gleich dem Betragsquadrat des Produkts von $|v\rangle$ und $|w\rangle$.

Betrachten wir als Beispiel die Photonen aus dem vorigen Kapitel. Der Zustandsraum für ein Photon ist ein zweidimensionaler Vektorraum, in dem die Polarisationsrichtung φ etwa durch den Vektor $\binom{\cos\varphi}{\sin\varphi}$ repräsentiert wird; im Falle eines reellen Vektorraums wäre jeder Vektor proportional zu einem solchen Einheitsvektor. Daß wir komplexe Vektorräume betrachten, bedeutet, daß zum Zustand auch noch ein Phasenfaktor gehört, der für die vorgestellten Protokolle zur Quantenkryptographie aber keine Rolle spielt.

Die Messung der Polarisationsrichtung mit einem Polarisationsfilter mit Durchlaßrichtung ψ wird durch den Operator

$$A_\psi = \begin{pmatrix} \cos 2\psi & \sin 2\psi \\ \sin 2\psi & -\cos 2\psi \end{pmatrix} = \begin{pmatrix} \cos^2\psi - \sin^2\psi & 2\sin\psi\cos\psi \\ 2\sin\psi\cos\psi & \sin^2\psi - \cos^2\psi \end{pmatrix}$$

modelliert; seine Eigenwerte sind ± 1 mit Eigenvektoren

$$|v_1\rangle = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix} \quad \text{und} \quad |v_{-1}\rangle = \begin{pmatrix} -\sin\psi \\ \cos\psi \end{pmatrix}.$$

Bei einem in Richtung φ polarisierten Photon liegt also die Wahrscheinlichkeit für $+1$ beim Quadrat von

$$\begin{pmatrix}\cos\varphi\\ \sin\varphi\end{pmatrix}\begin{pmatrix}\cos\psi\\ \sin\psi\end{pmatrix} = \cos\varphi\cos\psi + \sin\varphi\sin\psi = \cos(\varphi-\psi)\,,$$

die für -1 entsprechend bei $\sin^2(\varphi-\psi)$.

3. Quantenregister und QBits

In einem klassischen Computer werden Bits durch die beiden Zustände eines bistabilen Schaltelements wie etwa eines Flipflops oder durch eine Magnetisierungsrichtung oder etwas ähnliches realisiert. Unabhängig von der physikalisch-technischen Realisierung hat man immer ein Element, das sich stets in einem von zwei wohldefinierten Zuständen befindet; diese werden traditionell als 0 und 1 bezeichnet.

Angenommen, wir verwenden stattdessen ein Photon. Das ist beim heutigen Stand der Technologie zwar völlig unrealistisch, aber es ist wohl die anschaulichste Art, das Prinzip eines Quantencomputers zu verstehen; weiter unten werden auch realitätsnähere Ansätze diskutiert werden.

Zur Kodierung eines Bits können wir etwa vereinbaren, daß die horizontale Polarisation einer Null und die vertikale einer Eins entsprechen soll; mit einem Polarisationsfilter lassen sich Photonen in jedem der beiden Zustände unschwer produzieren.

Wir können aber auch anders vorgehen: Wir produzieren ein Photon mit Polarisationsrichtung 45° und lassen es so auf einen Doppelspat fallen, daß der eine der beiden austretenden Strahlen horizontal polarisiert ist, der andere vertikal.

Da nur ein Photon in den Kristall eintritt, kann auch nur eines austreten; indem wir zwei Detektoren in die beiden austretenden Strahlengänge stellen, können wir leicht feststellen, in welchem der beiden Strahlen sich das Photon befindet.

Dieses Aufstellen von Detektoren ist allerdings eine Messung, der Zustandsvektor des Photons wird also auf einen Eigenraum eines Operators projiziert. Was wir messen, ist daher nicht wirklich das aus dem Kristall austretende Photon. Dessen Zustand ist nach der experimentell sehr gut bestätigten Lehre der Quantentheorie, solange es keiner Messung unterworfen wird, eine Überlagerung der beiden möglichen Meßergebnisse: Bezeichnen wir die beiden möglichen Zustände *nach* der Messung mit $|0\rangle$ und $|1\rangle$, so befindet sich das Photon *vor* der Messung in einem Zustand, der durch den (auf Länge eins normierten) Vektor

$$\frac{\sqrt{2}}{2}\,|0\rangle + \frac{\sqrt{2}}{2}\,|1\rangle$$

beschrieben wird. Hier sind also die beiden Zustände $|0\rangle$ und $|1\rangle$ überlagert, und falls der Winkel 45° durch einen anderen ersetzt wird, lassen sich auch die Koeffizienten vor $|0\rangle$ und $|1\rangle$ beliebig verändern.

Ein solches quantenmechanisches System mit einem zweidimensionalen Zustandsraum bezeichnen wir als ein *QBit.* Im Gegensatz zu einem gewöhnlichen Bit, das nur die Werte 0 und 1 annehmen kann, sind für ein QBit also beliebige Werte der Form

$$\alpha\,|0\rangle + \beta\,|1\rangle \quad \text{mit} \quad \alpha, \beta \in \mathbb{C}$$

möglich.

Setzt man mehrere QBits zusammen, entsteht ein Quantenregister. Wichtig ist dabei, daß dieses Quantenregister ein einziges quantenmechanisches System ist, es ist also beispielsweise durchaus möglich, daß der Zustand „alle QBits sind $|0\rangle$“ mit dem Zustand „alle QBits sind $|1\rangle$“ überlagert ist. In diesem Fall wüßten wir zwar nicht im voraus, welches Ergebnis eine Messung eines QBits aus dem Register hätte, wir könnten uns aber sicher sein, daß nach der Messung des ersten QBits jede Messung eines weiteren QBits zum selben Ergebnis führte.

In Dirac-Notation schreibt man

$$|\alpha_1 \alpha_2 \ldots \alpha_n\rangle$$

für den Zustand, in dem das i-te QBit den Wert $\alpha_i \in \{0, 1\}$ hat; die 2^n Vektoren, die man auf diese Weise erhält, bilden offenbar eine Basis des Zustandsraums für das Quantenregister. (Mathematisch betrachtet ist dieser das Tensorprodukt $V_1 \otimes V_2 \otimes \cdots \otimes V_n$ der Zustandsräume V_i der einzelnen QBits.) Der Zustand eines Quantenregisters wird also beschrieben durch eine Linearkombination von Vektoren der obigen Form wie etwa

$$\frac{\sqrt{2}}{2}\,|00\ldots0\rangle\,\frac{\sqrt{2}}{2}\,|11\ldots1\rangle$$

für das Beispiel aus dem vorigen Absatz.

4. Quantencomputer

Genau wie ein klassischer Computer den Inhalt klassischer Register manipuliert, manipuliert ein Quantencomputer den Inhalt von Quantenregistern.

Das fundamentale Grundgesetz der Quantenmechanik, die SCHRÖDINGER-Gleichung, sagt uns, wie sich deren Inhalt verändern kann: Ist $|\psi(t)\rangle$ der Zustandsvektor zur Zeit t und wird das System durch keine äußeren Einflüsse gestört, so ist

$$i\hbar \frac{\partial\,|\psi(t)\rangle}{\partial t} = H\,|\psi(t)\rangle\ ,$$

wobei H den HAMILTONschen Operator des Systems bezeichnet, jenen HERMITEschen Operator also, der die Gesamtenergie des Systems beschreibt, und $\hbar = h/2\pi \approx 1{,}054589 \times 10^{-34}$ Js ist das durch 2π dividierte PLANCKsche Wirkungsquantum.

Die SCHRÖDINGER-Gleichung ist ein System linearer Differentialgleichungen mit konstanten Koeffizienten; seine Lösung läßt sich mit Hilfe der Matrixexponentialfunktion sofort hinschreiben:

$$|\psi(t)\rangle = e^{-iHt/\hbar}\,|\psi(0)\rangle = U(t)\,|\psi(0)\rangle \quad \text{mit} \quad U(t) = e^{-iHt/\hbar}\,.$$

Dabei ist

$$U(t)\cdot \overline{U(t)}^{T} = e^{-iHt/\hbar}\cdot e^{i\overline{H}^{T}t/\hbar} = E$$

die Einheitsmatrix, denn $\overline{H}^{T} = H$ nach Definition eines HERMITEschen Operators. Somit ist der Operator $U(t)$, der die zeitliche Entwicklung des Systems beschreibt, unitär und insbesondere auch invertierbar.

Diese Invertierbarkeit ist ein großer Unterschied zu klassischen Computern: Die Addition $3 + 5 = 8$ etwa läßt sich nicht invertieren, denn das Ergebnis „8“ enthält keine Information mehr darüber, wie es zustandegekommen ist. Langfristig werden aber möglicherweise auch klassische Computer bei immer weitergehender Miniaturisierung der Bauelemente mit reversibler Logik rechnen müssen, denn die einzige Stelle beim Rechnen, bei der Energieverbrauch aus physikalischen Gründen nicht vermieden werden kann, ist die Vernichtung von Information.

Von den klassischen Logikoperationen, mit denen heutige Computer arbeiten, ist nur die Negation reversibel und natürlich auch unitär: Bezüglich der Basis $\{|0\rangle, |1\rangle\}$ des Zustandsraums eines Photons wird sie durch die Matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

beschrieben. Konjunktion und Disjunktion sind aus demselben Grund wie die Addition nicht reversibel: Für eine reversible Logikoperation muß die Anzahl der Ausgangsbits gleich der der Eingangsbits sein. Ein Beispiel mit zwei Eingangsbits ist die kontrollierte Negation

$$(x,y) \mapsto \begin{cases} (x, \neg y) & \text{falls } x = 1 \\ (x, \ \ y) & \text{falls } x = 0 \end{cases} = (x, x \oplus y) \, ,$$

wobei $\oplus$ die Addition modulo 2 bezeichnet.

Bezüglich der Basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ wird die kontrollierte Negation durch die unitäre Matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

beschrieben, und sie ist auch quantenmechanisch realisierbar.

Wichtiger ist das TOFFOLI-Gate, das auf drei QBits operiert:

$$(x,y,z) \mapsto T(x,y,z) \underset{\text{def}}{=} \big((x,y,z \oplus (x \wedge y)\big) \, ,$$

denn mit seiner Hilfe lassen sich auf Kosten eines zusätzlichen Bits „und“ und „oder“ realisieren:

$$T(x,y,|0\rangle) = (x,y,x \wedge y) \quad \text{und} \quad T(\neg x, \neg y, |1\rangle) = (\neg x, \neg y, x \vee y) \, .$$

In der Standardbasis des Zustandsraums für drei QBits vertauscht das TOFFOLI-Gate einfach die beiden Vektoren $|110\rangle$ und $|111\rangle$, es wird also durch eine unitäre Matrix beschrieben. Seine quantenmechanische Realisierung *ohne* Phasenverschiebung ist etwas trickreich, aber möglich. Insbesondere kann es als eine Folge von Operationen mit jeweils höchstens zwei Eingangsvariablen realisiert werden, was für die technische Realisierung von großer Bedeutung ist: Interaktionen zwischen drei Quantenbits gleichzeitig wären zu weit jenseits des derzeit Vorstellbaren.

Ein Quantencomputer kann also mit der Negation und dem TOFFOLI-Gate alle logischen Berechnungen durchführen. Arithmetische Operationen können nach den klassischen Regeln der Schaltalgebra auf logische zurückgeführt werden; auch sie sind somit in einem Quantencomputer realisierbar.

5. Der Algorithmus von Shor

Als Beispiel für die kryptographische Relevanz eines Quantencomputers wollen wir die Faktorisierung einer ganzen Zahl N betrachten.

Der dümmste Ansatz zur Faktorisierung von Hand beseht darin, daß wir x und y unabhängig voneinander die Zahlen von 2 bis N durchlaufen lassen und jeweils das Produkt xy berechnen; falls dieses gleich N ist, haben wir eine Zerlegung von N gefunden.

Für eine etwa hundertstellige Zahl N (mit deren Faktorisierung ein heutiger Computer keine größeren Schwierigkeiten hat) wären hierzu rund 10^{200} Multiplikationen notwendig, mit klassischen Computern ein Ding der Unmöglichkeit: Selbst wenn alle heutigen Computer seit Beginn des Universums daran gerechnet hätten, wäre erst ein verschwindend kleiner Bruchteil der Produkte berechnet.

Für einen Quantencomputer dagegen sind diese 10^{200} Multiplikationen überhaupt kein Problem: Da $10^{100} \approx 2^{332}$ ist, nehmen wir zwei Quantenregister x, y aus etwa 350 QBits und bringen beide in den Zustand, in dem alle Basisvektoren denselben Koeffizienten haben. Sodann berechnen wir das Produkt der beiden Registerinhalte auf reversible Weise, d.h. wir berechnen das Tripel (x, y, xy). Dieses ist in einem Zustand, in dem alle Kombinationen (x, y, xy) mit $0 \leq x, y < 2^{350}$ überlagert sind, insbesondere also auch die, für die $xy = N$ ist. Der Quantencomputer kann also all diese Multiplikationen gleichzeitig durchführen.

Damit ist allerdings leider das Faktorisierungsproblem noch nicht gelöst, denn wir müssen das Tripel ja auch noch messen. Dabei kollabiert der überlagerte Zustand und wir erhalten als Ergebnis ein Tripel (x, y, xy) aus natürlichen Zahlen, über das wir keinerlei Kontrolle haben. Insbesondere ist die Wahrscheinlichkeit, daß an dritter Stelle die Zahl N steht, verschwindend gering, so daß dieser sehr einfache Ansatz definitiv nicht zum Ziel führt.

Ein Quantencomputer kann also zwar sehr viele Operationen gleichzeitig durchführen, aber dies läßt sich nur ausnutzen, wenn man das Ziel der Rechnung anschließend auch wirklich messen kann.

Deshalb geht der Algorithmus von SHOR das Faktorisierungsproblem völlig anders an: Wie von FERMAT bis hin zu den modernsten Siebalgorithmen immer wieder ausgenutzt wird, hat man dann eine gute Chancen, eine Zahl zu faktorisieren, wenn man sie selbst oder ein Vielfaches als Differenz zweier Quadrate darstellen kann: Ist

$$kN = x^2 - y^2 = (x - y)(x + y) \,,$$

so kann man hoffen, daß $\operatorname{ggT}(x-y, N)$ und $\operatorname{ggT}(x+y, N)$ echte Teiler von N sind.

Der Algorithmus von SHOR nutzt dies aus, indem er für eine zufällig gewählte Zahl x zwischen 2 und $N-2$ ihre Ordnung modulo N berechnet, d.h. die kleinste natürliche Zahl r, für die

$$x^r \equiv 1 \bmod N$$

ist. Eine solche Zahl r muß es nicht geben; für $N = 4$ und $x = 2$ beispielsweise gibt es keine. Elementare zahlentheoretische Betrachtungen zeigen, daß die zu N teilerfremde Zahlen bezüglich der Multiplikation modulo N eine zyklische Gruppe bilden; genau für diese Zahlen gibt es also ein solches r, und für alle anderen ist der ggT von x und N ein echter Teiler von N.

Falls r existiert und ungerade ist, hat man Pech gehabt und beginnt noch einmal mit einem neuen x; andernfalls ist

$$(x^{r/2+1} + 1)(x^{r/2} - 1) \equiv 0 \bmod N\ ,$$

wir sind also genau in der obigen Situation und können ggTs berechnen. Falls dies keine echten Teiler sind, haben wir ebenfalls Pech gehabt und müssen mit einem neuen x von vorne anfangen. Da r die kleinste Zahl ist, für die $x^r - 1$ durch N teilbar ist, passiert das genau dann, wenn $x^{r/2} + 1$ durch N teilbar ist, also im Fall, daß $x^{r/2} \equiv -1 \bmod N$. Der Algorithmus ist somit genau dann nützlich, wenn dieser Fall nicht allzu oft (oder gar immer) eintritt.

Ist etwa p eine ungerade Primzahl und $N = p^n$, so kann bei geraden r die Primzahl p keinesfalls Teiler *beider* Faktoren

$$x^{r/2+1} + 1 \quad \text{und} \quad x^{r/2} - 1$$

sein, da sich die beiden nur um zwei unterscheiden. Also ist einer der beiden durch p^n teilbar, was natürlich nur der erste sein kann. Somit versagt der Algorithmus von SHOR in diesem Fall für *jedes* x.

Im kryptographisch besonders interessanten Fall, daß $N = pq$ Produkt zweier ungerader Primzahlen ist, sieht es aber – je nach Standpunkt leider oder zum Glück – ganz anders aus: Da modulo einer Primzahl $y \equiv \pm 1$ die beiden einzigen Lösungen der Gleichung $y^2 \equiv 1$ sind, zeigt eine einfache, wenn auch etwas langwierige Argumentation über den chinesischen Restesatz, daß für mindestens die Hälfte aller zu N primen Zahlen die Ordnung r gerade und $x^{r/2}$ nicht kongruent -1 modulo N ist. Hier liegt also die Erfolgswahrscheinlichkeit bei über 50%; wiederholt man die die Rechnung mit einem anderen x, kommt man bereits auf 75%, nach sieben zufällig gewählten Werten von x auf über 99%, nach zehn auf über 99,9%. Falls es also gelingt, r effizient zu berechnen, ist dieses Verfahren extrem gefährlich für das RSA-System.

Die Berechnung von r stellt uns vor ein ähnliches Problem wie die Berechnung eines diskreten Logarithmus, und in der Tat führt der Algorithmus von SHOR zur Bestimmung von r auch auf einen Algorithmus zur Berechnung diskreter Logarithmen; auch die Verallgemeinerung auf entsprechende Probleme für elliptische Kurven stellt keine prinzipiellen Probleme.

SHOR geht folgendermaßen vor: Für $N < 2^L$ braucht er einen Quantencomputer mit zwei Quantenregistern der Längen $2L$ beziehungsweise L. Im ersten speichert er die Überlagerung aller Zahlen a von 0 bis $2^{2L} - 1$, für das zweite berechnet er den Zustand x hoch erstes Register modulo N. Der Aufwand hierfür liegt für die klassische Berechnungsmethode durch fortgesetztes Quadrieren in der Größenordnung von L^3 Operationen.

Der Computer befindet sich dann in einem Zustand, in dem alle Paare $(a, x^a \bmod N)$ mit $0 \leq a < 2^{2L}$ überlagert sind. Wird nun der Inhalt des zweiten Registers gemessen, ist die Chance für das Messen der Zahl eins natürlich verschwindend gering; das Meßergebnis wird *irgendeine* Zahl b zwischen 0 und $N - 1$ sein.

Durch die Messung des zweiten Registers hat sich aber der Inhalt des ersten verändert: Der *gesamte* Computer ist *ein* quantenmechanisches System, das in einen Zustand gebracht wurde, in dem der Inhalt des zweiten Registers gleich x hoch dem Inhalt des ersten Registers ist. Die Messung des zweiten Registers kann an dieser Tatsache nichts ändern, und da das zweite Register nach der Messung die Zahl b enthält, kann eine Messung des ersten Registers nun nur noch ein Ergebnis a liefern, für das $x^a \bmod N = b$ ist. Ein solches Ergebnis nützt uns aber nichts, und deshalb dürfen wir das erste Register keinesfalls messen.

Da das erste Register doppelt so lang ist wie das zweite, befindet es sich immer noch in einem überlagerten Zustand, nämlich in der Überlagerung aller Zahlen $0 \leq a < 2^{2L}$, für die $x^a \equiv b \bmod N$ ist. Ist a_0 die kleinste solche Zahl, sind dies genau diejenigen Zahlen der Form $a_0 + kr$ mit $k \in \mathbb{N}_0$, die kleiner sind als 2^{2L}; hierbei ist r die gesuchte Ordnung.

Bei den Zahlen im überlagerten Zustand im Register handelt es sich also um eine periodische Folge von $m = \left[\frac{2^{2L} - a_0}{r}\right]$ Zahlen; was uns interessiert, ist die Periode r.

Die Periode eines Gitters läßt sich optisch aus dem Beugungsspektrum des Gitters bestimmen; genauso wollen wir auch hier vorgehen und nicht den *Inhalt* des ersten Registers messen, sondern so etwas wie sein *Beugungsspektrum.* Ein wesentlicher Aspekt der Quantentheorie ist schließlich, daß auch einander ausschließende Zustände eines Teilchens miteinander interferieren: Falls etwa ein Photon zwei mögliche Wege zu einem Punkt zurücklegen kann, tritt auch im Falle eines einzigen Photons Interferenz auf.

Das rechnerische Analogon zum Beugungsspektrum ist die *Quanten-*FOURIER-*Transformation.* Ersteres leitet man bekanntlich aus dem HUYGENSschen Prinzip ab, wonach jeder Punkt, in dem das Gitter durchlässig ist, Ausgangspunkt einer neuen Welle ist; diese wird beschrieben durch eine komplexe Exponentialfunktion, und das Beugungsspektrum ist gegeben durch die Summe aller dieser Exponentialfunktionen.

Ganz entsprechend ordnet die Quanten-FOURIER-Transformation eines Registers der Länge $2L$ ordnet dem Inhalt $|x\rangle$ dieses Registers (aufgefaßt als Zahl zwischen 0 und $2^{2L}-1$ oder als Überlagerung mehrerer solcher Zahlen) den Zustand

$$2^{-L}\sum_{\nu=0}^{2^{2L}-1} e^{2\pi|x\rangle\nu/2^{2L}}\,|\nu\rangle$$

zu. Man überzeugt sich leicht, daß dies eine unitäre Abbildung definiert; dies liegt einfach an der wohlbekannten Formel

$$\sum_{\nu=1}^{m} e^{k\nu/m} = \begin{cases} n & \text{falls } n|k \\ 0 & \text{sonst} \end{cases} ,$$

hinter der die Symmetrie der n-ten Einheitswurzeln zum Nullpunkt der komplexen Zahlenebene steckt – falls $n = m/\mathrm{ggT}(m,k)$ größer als eins ist.

Schwieriger ist es, diese Abbildung quantenmechanisch zu realisieren; dazu braucht man die sogenannte schnelle FOURIER-Transformation, die durch geschickte Gruppierung der Summanden nach dem Prinzip *Teile und herrsche* eine deutlich effizientere Berechnung der Abbildung erlaubt und deren einzelne Rechenschritte sich leicht in Quantenprozesse übersetzen lassen, deren Ergebnis jeweils nur von zwei QBits abhängt.

Diese Quanten-FOURIER-Transformation wird nun also auf das erste Register angewandt; dadurch kommt dieses in den Zustand

$$\begin{aligned} &2^{-L} \sum_{\nu=0}^{2^{2L}-1} \frac{1}{\sqrt{m}} \sum_{\mu=1}^{m} e^{2\pi i (a_0+\mu r)\nu/2^{2L}} |\nu\rangle \\ &= \frac{1}{2^L\sqrt{m}} \sum_{\nu=0}^{2^{2L}-1} e^{2\pi i a_0} \left(\sum_{\mu=1}^{m} e^{2\pi i r\mu\nu/2^{2L}} \right) |\nu\rangle \end{aligned} .$$

Da $m = \left[\frac{2^{2L}-a_0}{r}\right] \approx 2^{2L}/r$ ist, läßt sich die Klammer approximieren durch

$$\sum_{\mu=1}^{m} e^{2\pi i r\mu\nu/2^{2L}} \approx \sum_{\mu=1}^{m} e^{2\pi i \mu\nu/m} = \begin{cases} m & \text{falls } m|\nu \\ 0 & \text{sonst} \end{cases} .$$

Der Zustand des ersten Registers ist also ungefähr gleich

$$\frac{1}{2^L\sqrt{m}} \sum_{\nu=0}^{m-1} e^{2\pi i a_0} m\, |r\nu\rangle = \frac{e^{2\pi i a_0}\sqrt{m}}{2^L} \sum_{\nu=0}^{m-1} |r\nu\rangle \ .$$

(Durch Aufsummieren der geometrischen Reihe kann man natürlich auch leicht den genauen Wert der geometrischen Reihe berechnen.)

Falls wir ein Register messen, dessen Inhalt durch die approximative Zustandsfunktion beschrieben wird, erhalten wir mit *Sicherheit* ein Vielfaches von r als Ergebnis, wobei die verschiedenen Vielfachen allerdings gleiche Wahrscheinlichkeit haben.

Die exakte Zustandsfunktion führt zu leichten Abweichungen davon: Die Betragsmaxima der Koeffizienten liegen nun bei den Vielfachen der wirklichen Periode $2^{2L}/r$, aber weiter davon entfernt liegende Werte sind nicht mehr unmöglich, sondern nur noch unwahrscheinlich.

Wir wiederholen deshalb das Experiment mit demselben x mehrfach (eine genauere Analyse zeigt, daß etwa L Wiederholungen mit hoher Wahrscheinlichkeit genügen) und haben dann verschiedene Meßwerte für ν. Für die meisten, wenn nicht gar alle dieser Werte gibt es eine Zahl λ, so daß

$$\nu \approx \lambda \frac{2^{2L}}{r} \quad \text{oder} \quad \frac{\nu}{2^{2L}} \approx \frac{\lambda}{r}$$

ist.

In der hinteren Gleichung ist die linke Seite bekannt; von der rechten kennen wir weder Zähler noch Nenner. Wir wissen aber, daß der Nenner r

als Ordnung von x modulo N kleiner als N ist und damit sehr viel kleiner als der Nenner 2^{2L} auf der linken Seite, der ja größer als N^2 ist. Zur Bestimmung von r müssen wir also für die verschiedenen Brüche $\lambda/2^{2L}$ Approximationen finden, deren Nenner kleiner ist als N.

Die besten rationalen Approximationen einer reellen Zahl liefert deren Kettenbruchentwicklung. Was das ist und wie man es berechnet, sieht man am besten an einem Beispiel: Angenommen, wir wollen die Zahl π durch eine rationale Zahl approximieren. Als ganzzahlige Approximation können wir $[\pi] = 3$ nehmen; als erste echt gebrochene Approximation versuchen wir es mit einem Ansatz der Form

$$\pi \approx 3 + \frac{1}{a_1} \quad \text{oder} \quad a_1 \approx \frac{1}{\pi - 3};$$

der Kettenbruchalgorithmus wählt

$$a_1 = \left[\frac{1}{\pi - 3}\right] = [7{,}0625\ldots] = 7\,.$$

Also ist $\pi \approx 3\frac{1}{7} = \frac{22}{7}$, eine seit langem bekannte für den kleinen Nenner sehr gute Approximation.

Als nächstbessere Approximation versuchen wir es mit

$$\pi \approx 3 + \cfrac{1}{7 + \cfrac{1}{a_2}} \quad \text{oder} \quad \frac{1}{\pi - 3} \approx 7 + \frac{1}{\pi};$$

wir wählen

$$a_2 = \left[\cfrac{1}{\cfrac{1}{\pi - 3} - 7}\right] = [15{,}996599\ldots] = 15$$

und so weiter. Allgemein setzen wir also zur Approximation einer reellen Zahl x in einem nullten Schritt

$$b_0 = x \quad \text{und} \quad a_0 = [b_0]\,.$$

Im i-ten Schritt, für $i \geq 1$, testen wir zunächst, ob $a_{i-1} = b_{i-1}$ ist; wenn ja, sind wir fertig (und x war eine rationale Zahl). Andernfalls setzen wir

$$b_i = \frac{1}{b_{i-1} - a_{i-1}} \quad \text{und} \quad a_i = [b_i]\,.$$

Als n-te Konvergente des Kettenbruchs bezeichnen wir die rationale Zahl

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \cfrac{1}{\ddots\, a_n}}}}} \, .$$

Für uns interessant ist dieser Kettenbruchalgorithmus wegen des folgenden Satzes:

Ist für eine reelle Zahl x *und eine rationale Zahl* p/q

$$\left| x - \frac{p}{q} \right| < \frac{1}{2q^2} \, ,$$

so ist p/q *ein nicht weiter kürzbarer Bruch und eine Konvergente des Kettenbruchs von* x*.*

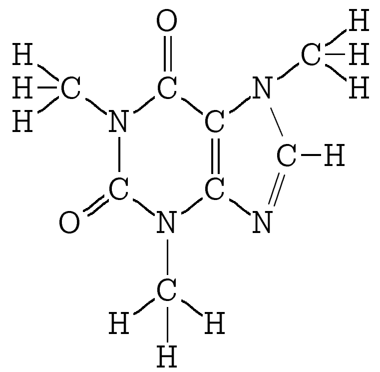
Der Kettenbruchalgorithmus liefert also die besten rationalen Approximationen mit Nennern einer vorgegebenen Größenordnung, und deshalb berechnen wir für jeden Meßwert μ alle Konvergenten des Kettenbruchs zu $\frac{\mu}{2^{2l}}$ mit Nenner kleiner N und suchen dann nach einem Nenner r, der bei möglichst vielen Meßwerten auftritt. Dieser ist ein Kandidat für die gesuchte Ordnung von x, und wir verfahren damit wie im SHORschen Algorithmus angegeben.

6. Experimentelle Realisierung

Das große Problem beim Bau von Quantencomputern ist die *Dekohärenz:* Ein überlagerter Zustand bleibt nur erhalten, wenn keinerlei Wechselwirkung mit äußeren Systemen eintritt; dies läßt sich auch mit großem Aufwand nur für sehr kurze Zeitspannen sicherstellen. SHOR entwickelte *Quantencodes,* die nach dem Vorbild klassischer fehlerkorrigierender Codes ein gewisses Maß an Dekohärenz verkraften können, aber auch damit läßt sich diese Zeitspanne nur begrenzt ausdehnen. Ein Quantencomputer auf der Basis von Photonen mit ihren extrem kurzen Dekohärenzzeiten erscheint deshalb im Augenblick jenseits jeder technischen Realisierbarkeit.

Am vielversprechendsten ist im Augenblick ein Ansatz, der mit NMR-Spektroskopie arbeitet. Bei dieser handelt es sich um eine schon seit über dreißig Jahren gebräuchliche chemische Analysemethode, die inzwischen auch für bildgebende Verfahren in der Medizintechnik eingesetzt wird. Sie beruht auf der magnetischen Resonanz gewisser Atomkerne. Zwar zeigen nur wenige Isotopen Kernresonanz, aber es handelt sich dabei um Isotope häufiger und wichtiger Atome: ^1H,^2H,^{10}B,^{11}B,^{13}C,^{14}N,^{15}N,^{17}O,^{19}F,^{29}Si und ^{31}Si.

Als Beispiel eines für einen ersten Quantencomputer geeigneten Moleküls wurde (nicht unbedingt ganz ernsthaft) das *Koffein* vorgeschlagen, das in der Tat genug solche Atome enthält, um einen kleinen „Supercomputer in der Kaffeetasse“ zu realisieren:



Protonen und Neutronen in einem Atom haben jeweils Spin $\pm 1/2$, wobei sich benachbarte Protonen sowie benachbarte Neutronen jeweils antiparallel ausrichten. Bleibt dabei ein Gesamtspin übrig, richtet sich das Atom in einem umgebenden Magnetfeld der Stärke von etwa 9 bis 15 Tesla in einer von wenigen wohldefinierten Richtungen aus, zwischen denen es durch Aufnahme beziehungsweise Abgabe von Strahlungsquanten wechseln kann. Mit geeigneten Radiowellen läßt sich also zwischen verschiedenen Zuständen hin- und herschalten; durch Aufnahme eines Resonanzspektrums lassen sich die (über viele Atome gemittelten) Zustände ablesen. Dadurch, daß man hier nicht mit Quantenzuständen einzelner Partikel arbeitet, sondern über viele Partikel mittelt, wird auch das Problem der Dekohärenz entschärft. Man rechnet damit, daß ein solcher Computer schon bald die Zahl fünfzehn faktorisieren kann.

QBits in einzelnen Atomen benutzt die um 1995 in Innsbruck konzipierte *Ionenfalle.* Hier werden Ionen durch elektromagnetische Felder in einer linearen Anordnung gehalten; die QBits werden kodiert durch Anregungszustände der Ionen (von denen jedes durch einen eigenen Laser kontrolliert wird) und der Gitterschwingungen (Phononen) zwischen den einzelnen Ionen. Überlagerte Zustände, die mehrere QBits umfassen, werden über die Vibrationsenergie ihres Schwerpunkts kontrolliert.

Am National Institut of Standards in Boulder, Colorado, wurde so die kontrollierte Negation implementiert; Faktorisierung kleiner Zahlen erscheint durch Weiterentwicklung und Vergrößerung der Apparatur möglich, allerdings müssen dazu noch einige technische Probleme gelöst werden.

Für weiteres und insbesondere auch andere physikalische Ansätze sei auf die zitierte Literatur verwiesen.

7. Literaturhinweise

Auch hier sind als erstes die beiden schon im vorigen Kapitel erwähnten und kurz charakterisierten Bücher

MICHAEL BROOKS[Hrsg.]: *Quantum computing and communications,* Springer, 1999 *und*

COLIN P. WILLIAMS, SCOTT H. CLEARWATER: *Explorations in quantum computing,* Springer Telos, 1997

zu nennen. Ein Buch, das speziell auf die algorithmische Seite Bezug nimmt (oft eher anhand von Beispielen als einer klaren Darstellung des Algorithmus) ist

ARTHUR O. PITTENGER: *An introduction to quantum computing algorithms,* Birkhäuser, Boston, 1999

Die Originalarbeit zum Algorithmus von SHOR wurde kürzlich in überarbeiteter Form nachgedruckt:

PETER W. SHOR: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41** (1999), S. 303–332

Einen weiteren Übersichtsartikel, in dem auch auf das hier nicht behandelte Problem der Fehlerkorrektur in Quantencomputern eingegangen wird, ist SHORS Vortrag auf dem internationalen Mathematikerkongreß 1998 in Berlin:

PETER W. SHOR: Quantum Computing, *Proceedings of the international congress of mathematicians Berlin 1998, Documenta Mathematica, Extra volume ICM* 1998 · I, DEUTSCHE MATHEMATIKER-VEREINIGUNG, 1998, S. 467–486 *oder*
www.mathematik.uni-bielefeld.de/documenta/xvol-icm/00/00.html

Eine Darstellung des Kettenbruchalgorithmus für Schüler, die ab etwa Klasse elf lesbar sein sollte, ist

N.M. BESKIN: *Fascinating Fractions,* Mir, Moskau, 1986.