Faktorisieren mit dem Quadratischen Sieb

Ein Beitrag zur Didaktik der Algebra und Kryptologie

Ralph-Hardo Schulz und Helmut Witten

Eines der zur Zeit schnellsten Verfahren zur Faktorisierung ganzer Zahlen ist das "Quadratische Sieb" (engl. "quadratic sieve factorization method"), das 1981 von Carl Pomerance entwickelt wurde. Mit einer Variante dieses Siebes konnte z.B. 1994 die 129-dezimalstellige Zahl RSA-129 unter Leitung von D. Atkins, M. Graff, A. Lenstra und P. Leyland parallel unter Mithilfe von 600 Freiwilligen (in 8 Monaten mit rund 10^{17} Operationen) faktorisiert werden (s. [Atkins et al.], 'RSA-129' in [Wikipedia en] und [Pomerance 1996]). Diese Zahl war nicht Bestandteil der berühmten RSA Factoring Challenge, die von der Firma RSA Security am 18. März 1991 gestartet und im Jahr 2007 beendet wurde (s. 'RSA_Factoring_Challenge' in [Wikipedia en]). RSA-129 wurde bereits im Jahr 1976 von Martin Gardner in der Zeitschrift Scientific American in einem Artikel über das damals neu entwickelte RSA-Kryptosystem veröffentlicht und erhielt später seinen Namen sozusagen ehrenhalber. Ron Rivest schätzte seinerzeit, dass man 40 Billiarden Jahre (engl. quadrillion! years) brauchen würde, um diese Zahl zu faktorisieren (s. Squeamish_ossifrage in [Wikipedia en]). Die Faktorisierung gelang dann aber schon nach 18 Jahren. (Zu weiteren Einzelheiten s. [Witten & Schulz 2006], S. 55 ff. sowie 'The_Magic_Words_are_Squeamish_Ossifrage' in [Wikipedia en].) Mit der Entschlüsselung war klar, dass die für das RSA-Kryptosystem benötigte sichere Schlüssellänge wesentlich größer sein musste.
Das Quadratische Sieb eignet sich besonders für Zahlen unter 100 dezimalen Stellen, das konkurrierende "Zahlkörpersieb" (engl. "number field sieve") (s. 'General number field sieve' in [Wikipedia en] , 'Zahlkörpersieb' in [Wikipedia de] und [Pomerance 1996]) bei über 130 Stellen. (Man beachte auch unsere Zeit-Experimente zur Faktorisierung, s. [Schulz & Witten 2010].)
Wir beschreiben im Folgenden die Basisversion des Quadratischen Siebs sowie die Variante des Quadratischen Siebs mit mehrfachen Polynomen, das sogenannte "Multiple Polynomial Quadratic Sieve" MPQS, das unabhängig von J. Davis und D. Holdridge bzw. P. Montgomery gefunden wurde. Bei der Darstellung der Verfahren orientieren wir uns an [Buchmann 2010], [Crandall & Pomerance 2005], [Esslinger et al. 2011], [Pomerance 1996], 'Quadratisches Sieb' in [Wikipedia de] und 'quadratic sieve' in [Wikipedia en].

1 Die Grundidee

Sei die natürliche Zahl n zu faktorisieren. Dazu bestimme man ganze Zahlen a und b mit

$$(1.1) \qquad a^2 \equiv b^2 \pmod{n} \qquad \text{und} \qquad (1.2) \quad a \not\equiv \pm b \,(\bmod n).$$

Denn dann gilt: n teilt a^2-b^2, aber nicht $a-b$ und $a+b$; es existiert nun eine ganze Zahl t mit $(a-b)(a+b) = tn$; wären $(a-b)$ und n teilerfremd, so wäre n ein Teiler von $a+b$, ein Widerspruch; somit hat man mit $\mathrm{ggT}\,(a-b, n)$ einen (weil n nicht $a-b$ teilt) echten Teiler von n gefunden. (Analog ist $\mathrm{ggT}\,(a+b, n)$ echter Teiler.) Solche Zahlen a und b sind aber im Allgemeinen nicht einfach zu bestimmen.

Anmerkung: Das beschriebene Prinzip wird auch bei anderen Faktorisierungsverfahren angewandt, z.B. beim "Verfahren von Fermat", bei dem $a^2 - b^2$ gleich n statt einem beliebigen Vielfachen von n gewählt wird (s.u.), und beim schon erwähnten "Zahlkörpersieb". Der Unterschied liegt im Vorgehen beim Bestimmen der Zahlen a und b.
Im Folgenden beschreiben wir zunächst das Faktorisieren nach Fermat, das schon wesentliche Elemente des Quadratischen Siebs enthält.

2 Das Faktorisierungsverfahren von Fermat

2.1 Der Algorithmus

Gesucht ist eine natürliche Zahl a derart, dass a^2-n gleich einem Quadrat b^2 mit $b \in \mathbb{N}$ ist (dies bedingt $a > \sqrt{n}$). Definiert man $q(x) := x^2 - n$ (als Kraitchik's quadratisches Polynom bezeichnet), so kann man $q(x)$ für verschiedene $x > \sqrt{n}$ solange berechnen, bis man ein Quadrat $b^2 = q(a) = a^2 - n$ gefunden hat. Ähnlich wie oben erhält man dadurch die Zerlegung

$$n = a^2 - b^2 = (a-b)(a+b) = (a - \sqrt{q(a)})(a + \sqrt{q(a)}). \tag{2.1}$$

Damit $q(x)$ nicht größer als nötig wird, wählt man x nahe bei $\sqrt{n}$. Statt $q(x) := x^2 - n$ betrachtet man daher oft das Polynom $g(\hat{x}) := (\hat{x} + \lceil \sqrt{n} \rceil)^2 - n$. (Im Folgenden bezeichnet $m := \lceil \sqrt{n} \rceil$ die kleinste natürliche Zahl, die größer oder gleich $\sqrt{n}$ ist.)
Man untersucht nun nacheinander für $x = m$, $x = m+1$, $x = m+2$, $x = m+3$ usw., ob $q(x)$ ein Quadrat ist, solange, bis man ein Quadrat gefunden hat, was aber lange dauern kann (s.u.). Die Hauptschleife des Algorithmus und ein Python-Programm findet man in den beiden nächsten Kästen.

Hauptschleife des Verfahrens von Fermat
(nach [Crandall & Pomerance 2005] (5.1.1); zur rechten Laufgrenze s.u. § 2.3)

For $(\lceil \sqrt{n} \rceil] \le a \le (n+9)/6)$
{ if ($b = \sqrt{a^2 - n}$ is an integer) return $a - b$; }
return "n ist Primzahl".

Faktorisierung nach dem Verfahren von Fermat
(Programm in Python Version 2.7.2)
(Das Programm liefert bei zusammengesetzten positiven ganzen Zahlen einen nichttrivialen Teiler, bei Primzahlen wird 1 zurückgegeben. Die geraden Zahlen werden am Anfang gesondert behandelt. Außerdem wird sichergestellt, dass nur positive ganze Zahlen größer 1 übergeben werden.)

```
# -*- coding: cp1252 -*-
from math import sqrt, ceil
def fermat(n):
 if n <= 1 : return -1          # Unzulässige Eingabe für n
 n = int(n)                     # ggf. Nachkommastellen abschneiden
 if n%2 == 0: return 2          # Eingabewert ist gerade!
 a = ceil(sqrt(n))              # Startwert für die Schleife
 while a <= (n + 9)/6 :
        b =sqrt(a * a - n)
        if (b). is_integer():
           return int(a - b)    # int unterdrückt ".0" am Ende der Ausgabe
        a = a+1                 # nächster Versuch
 return 1                       # n ist Primzahl
```

Die angegebene Python-Funktion funktioniert aufgrund der Rundungsfehler bei der Umwandlung von 'biginteger' in 'float' und umgekehrt nur dann zuverlässig, wenn der Eingabeparameter kleiner als 2^{56} ist. Deshalb kann man mit dieser Funktion auch nicht die Semiprimzahl aus dem Kryptorätsel "NICHT SO GEHEIME NACHRICHT AUS MALAWI TEIL I (RSA)" (s. http://www.mysterytwisterc3.org/images/challenges/mtc3-schaefer-01-rsa-de.pdf) zerlegen, die 309 Stellen hat bzw. 1025 Bit lang ist. Zur Zerlegung einer so großen Semiprimzahl mit dem Fermat-Verfahren benötigt man eine höhere Genauigkeit bei der Umwandlung, die sich z.B. mit dem 'open source' Computer-Algebra-System Sage (http://www.sagemath.org/) bzw. einer anderen Python-Mathematik-Bibliothek realisieren lässt.

2.2 Kleine Beispiele zum Verfahren von Fermat

- n=703:

x	$27(=\lceil\sqrt{703}\rceil)$	28
x^2	729	784
$q(x)=x^2-n$	26	$81=9^2$

  Dies liefert $703=(28-9)\cdot(28+9)=19\cdot 37$.

- n=1.649 (vgl. 'Quadratisches Sieb' in [Wikipedia de]):
  Nach Beginn mit $x=41=\lceil 40,6\ldots\rceil=\lceil\sqrt{1649}\rceil$ liefert erst im 17. Schritt $x=57:=a$ ein Quadrat: $q(a)=40^2=:b^2$. Es folgt $1649=(57+40)(57-40)=97\cdot 17$. (Fortsetzung des Beispiels s.u.).

- $n = 15.229$
  Beginnend mit $x = 124$ erhält man im 4. Schritt mit $127 := a$ die Beziehung $q(a) = 30^2$ und damit
  $15.229 = 127^2 - 30^2 = (127 - 30)(127 + 30) = 97 \cdot 157$.

- Weitere Beispiele ($n = 1.729$ und $n = 290.377$) findet man z.B. unter 'Faktorisierungsmethode von Fermat' in [Wikipedia de].

2.3 Zur Laufzeit des Fermatschen Algorithmus

Die Darstellung als Differenz zweier Quadrate ist i.A. nicht eindeutig: $105 = 11^2 - 4^2 = 19^2 - 16^2$. Für ungerade Semiprimzahlen, also natürliche Zahlen der Form $n = p \cdot q$ mit ungeraden Primteilern p und q, folgt aber aus $pq = (a + b)(a - b)$ und $p > q$ sowie $a, b \in \mathbb{N}$, dass $p = (a + b)$ und $q = (a - b)$ und damit $a = \frac{p+q}{2}$ und $b = \frac{p-q}{2}$ gilt; umgekehrt erfüllen (übrigens auch für andere als Semiprimzahlen) diese a und b die Gleichungen $b \neq a \geq \sqrt{n}$ (das arithmetische Mittel positiver reeller Zahlen ist immer größer gleich dem geometrischen Mittel) und $a^2 - b^2 = (a + b)(a - b) = p \cdot q = n$, also (2.1). Kann das Verfahren bei Semiprimzahlen früher enden? Nein! Denn aus $y^2 - n = c^2$, also $n = (y + c)(y - c)$, für $y, c \in \mathbb{N}$ folgt, dass $p = y + c$ und $q = y - c$ und damit $y = a$ ist. Also kommt der Algorithmus, der $x = m, x = m + 1, \ldots x = a = m + d$ testet und erst mit $x = a$ erfolgreich ist, nach $d + 1 = a - m + 1 = \frac{p+q}{2} - \lceil\sqrt{n}\rceil + 1$ Schritten zu einem Ende.
Nun kennt man p und q nicht vor erfolgreicher Anwendung des Algorithmus; daher nimmt man als obere Laufgrenze die Zahl $A = \frac{n+9}{6}$; es gilt für $p \geq q \geq 3$ nämlich die Ungleichung $q(p - 3) \geq 3(p - 3)$ und daher $A = \frac{pq+9}{6} \geq \frac{p+q}{2} = a$.
Diese Schranke wird auch erreicht: Der 'worst case' tritt bei $n = 3p$ mit ungerader Primzahl p ein; dann führen nämlich erst $A = (n + 9)/6$ und $B = (n - 9)/6$ wegen $n = (A + B)(A - B)$ (gilt immer für reelle A und B und $A, B \in \mathbb{N}$) und nicht frühere Werte zur Faktorisierung: Es ist ja $a = (p + 3)/2 = (3p + 9)/6 = A$.
Damit ist die (maximale) Laufzeit des Algorithmus mindestens von der Größenordnung von n, also exponentiell bzgl. der Stellenzahl von n. (In Fall $n = 3p$ ist natürlich die 'brute-force' Faktorisierung mit Probedivisionen ungleich schneller.)

3 Von Fermat zum Quadratischen Sieb

3.1 Kombination von Kongruenzen

Um schneller zum Ziel zu kommen (und subexponentiell zu bleiben), sucht man erstens (angeregt durch Maurice Kraitchik 1920, aufbauend auf Arbeiten von Gauß und Seelhoff) nach Zahlen a und b derart, dass $a^2 - b^2$ ein beliebiges Vielfache von n ist. (Vgl. die Kongruenz (1.1) !)

Dies ermöglicht zweitens, statt direkt ein a zu bestimmen, für das $q(a)$ ein Quadrat ist, ein Quadrat durch geeignete Kombination von Kongruenzen zu erreichen: Aus $q(x_i) = {x_i}^2 - n \equiv {x_i}^2 \pmod{n}$ $(i = 1, \ldots, k)$ ergibt sich ja

$$(3.1) \qquad q(x_1) \cdot q(x_2) \cdot \ldots \cdot q(x_k) \equiv {x_1}^2 \cdot {x_2}^2 \cdot \ldots \cdot x_k^2 \pmod{n},$$

wobei die rechte Seite auf jeden Fall ein Quadrat ist, nämlich $(x_1 \cdot x_2 \cdot \ldots \cdot x_k)^2$. Wählt man nun verschiedene Werte $x_1, \ldots, x_k$ so aus, dass

$$q(x_1) \cdot q(x_2) \cdot \ldots \cdot q(x_k) \mod n$$

ein Quadrat b^2 ist, so gilt für $a := x_1 \cdot x_2 \ldots \cdot x_k$ und b die Bedingung (1.1). Die (oft erfüllte) Hoffnung ist, dass auch Gleichung (1.2) erfüllt ist und damit $\mathrm{ggT}(a + b, n)$ und $\mathrm{ggT}(a - b, n)$ echte Teiler von n sind (vgl. §1!)

Wie findet man nun geeignete Werte x_i ? Dazu sieht man sich die Primfaktorzerlegungen von $q(x_i)$ für mehrere x_i an und kombiniert geeignete Kongruenzen derart, dass die Primfaktordarstellung des Produkts nur gerade Exponenten enthält, das Produkt also ein Quadrat ist. Man will also erreichen, dass sich die Exponenten entsprechender Primfaktoren der ausgewählten $q(x_i)$ zu einer geraden Zahl, mod 2 also zu 0 addieren..

Ein erstes kleines Beispiel zum Quadratischen Sieb:

$n = 1.649$ (s.o.)

x	$q(x) = x^2 - n$	Primfaktorzerlegung: Exponenten der Primfaktoren				Exponenten mod 2 der Primfaktoren			
		2	3	5	23	2	3	5	23
41	32	5	0	0	0	1	0	0	0
42	115	0	0	1	1	0	0	1	1
43	200	3	0	2	0	1	0	0	0

Die Exponentenvektoren zu 41 und 43 addieren sich mod 2 zum Nullvektor. Daher wählt man $a = 41 \cdot 43 = 1.763$ und $b = \sqrt{q(41) \cdot q(43)} = \sqrt{2^5 \cdot 2^3 \cdot 5^2} = 2^4 \cdot 5 = 80$. Erfreulicherweise ist $a \pm b \not\equiv 0 \pmod{n}$. Einen Teiler von 1.649 erhält man als $\mathrm{ggT}(a - b, n) = \mathrm{ggT}(1.763 - 80, 1.649)$, der sich z.B. mit Hilfe des Euklidischen Algorithmus zu 17 berechnet.

In diesem Beispiel konnte man durch genaues Hinsehen erkennen, mit welchen Kongruenzen man die Exponentenvektoren zum Nullvektor kombinieren kann. Wie kann man dies im allgemeinen Fall systematisch erreichen? Eine entsprechende Strategie wurde (gemäß C. Pomerance) von John Brillhart und Michael Morrison entwickelt:

Zunächst sucht man eine Reihe von Werten $q(x_i) \mod n \quad (i = 1, \ldots k)$, bei denen die Primfaktorzerlegung von $q(x_i)$ jeweils bekannt ist und höchstens kleine Primzahlen $p_1, \ldots, p_r$ (s.u.§3.2) enthält. Der zu $q(x_i) = \prod_{j=1}^{r} p_j^{h_j}$ gehörige Exponentenvektor mod 2, nämlich $(h_1 \mod 2, \ldots, h_r \mod 2)$ (mit Komponenten 0 oder 1) lässt sich dann auffassen als Vektor $\vec{v_i}$ aus dem Vektorraum $V = \mathbb{F}_2^r$ der Dimension r über $\mathbb{F}_2$ (dem Körper mit 2 Elementen, der

auch als Galoisfeld GF(2) bezeichnet wird). Gesucht sind dann Koeffizienten $c_1, \ldots, c_k \in \{0, 1\}$, nicht alle 0, derart, dass $\sum_{i=1}^{k} c_i \vec{v_i} = 0$ (über $\mathbb{F}_2$) gilt. Falls es solche c_i gibt, lassen sie sich als Lösungen eines linearen Gleichungssystems in diesen Variablen (z.B. mit der Gaußschen Eliminationsmethode) bestimmen. *Anmerkung:* Das lineare Gleichungssystem hat eine Lösung ungleich $\vec{0}$, wenn die Vektoren $\vec{v_1}, \ldots \vec{v_k}$ linear abhängig sind. Aus der Linearen Algebra wissen wir, dass dies der Fall ist, wenn die Anzahl k der Vektoren größer als die Dimension r des Raumes ist, evtl. schon früher.

3.2 Wahl einer Faktorbasis

Da bei dem zu behandelnden Algorithmus die Faktorisierung einer großen Zahl auf viele kleinere Faktorisierungen zurückgeführt wird, und da die Dimension r nicht zu groß sein sollte, wählt man aus der Menge $\mathbb{P}$ aller Primzahlen eine Teilmenge von kleinen, durch eine Zahl B beschränkten Primzahlen aus. Zur Vermeidung zu großer Zahlen will man von m auch Schritte rückwärts gehen können und lässt negative $q(x_i)$ (s. §3.3) und somit den Faktor (-1) zu. Man legt daher eine sogenannte *Faktorbasis* (engl. 'factor base') fest:

$$F_B := \{p \in \mathbb{P} \,|\, p \leq B\} \cup \{-1\}. \tag{3.2}$$

Eine Zahl $q(x)$ heißt dann *B-glatt* (engl. 'B-smooth') oder auch nur *glatt*, falls sie lediglich Primfaktoren aus F_B (und evtl. Faktor (-1) hat. Man betrachtet nun nur solche x_i, bei denen $q(x_i)$ glatt ist. Für kleine B existieren dann effiziente Faktorisierungsalgorithmen zur Bestimmung der Primfaktorzerlegung von $q(x_i)$. (Oft läßt man noch ein oder zwei, manchmal drei weitere bei der Faktorisierung auftretende Primzahlen zu, die größer als B sind.)

Anmerkung: Es ist p Teiler von $q(x) = x^2 - n$ genau dann, wenn $x^2 - n \equiv 0 \pmod{p}$, also $x^2 \equiv n \pmod{p}$ gilt, d.h. n sogenannter Quadratischer Rest mod p (mit Wurzel x) ist (siehe den Kasten "Quadratische Reste"). Von Interesse für F_B sind also nur Primzahlen p dieser Eigenschaft. Entsprechend kann man F_B durch Streichen der Nicht-Quadrate zu einer Teilmenge $\hat{F_B} := F$ abändern.

Eine weitere entscheidende Vereinfachung erhält man dadurch, dass man aus gegebenen Zerlegungen auf andere Zahlen schließt. Weiß man z.B., dass p Teiler von $q(x_\ell) = x_\ell^2 - n$ ist, so folgt aus

$$q(x_\ell + j \cdot p) = (x_\ell + j \cdot p)^2 - n \equiv x_\ell^2 - n = q(x_\ell) \pmod{p}, \tag{3.3}$$

dass p auch alle Zahlen aus $\{q(x_\ell + j \cdot p) \,|\, j \in \mathbb{Z}\}$ teilt. Diese Tatsache ermöglicht das weiter unten beschriebene Sieben.

Einige Beispiele von Faktorbasisgrößen (auch für Variationen des Quadratischen Siebs) sind in folgender Tabelle angegeben (cf. [Atkins et al.], [Esslinger et al. 2011], [Leyland et al. 2002] und 'Quadratisches Sieb' in [Wikipedia de]):

Anzahl der Dezimalstellen von n	50	120	129 (RSA)	135
Anzahl der Primzahlen der Faktorbasis	3.000	245.000	524.338	550.000

3.3 Wahl eines Siebintervalls

Auch die Folge der betrachteten x_i beschränkt man oft durch die Bedingung $|x - \lceil\sqrt{n}\rceil\,| \leq s$ auf ein sogenanntes *Siebintervall* (engl. 'sieve interval' oder 'sieving range') also auf

$$S = \{x \in \mathbb{N}\,|\,|x - m| \leq s\}.$$

Hierbei sind auch negative Werte für $x - \lceil\sqrt{n}\rceil$, also $x < m$, zugelassen. Der Vorteil dabei ist, dass die Werte von $q(x)$ im Mittel etwas kleiner sind, der Nachteil, dass auch negative $q(x)$ auftreten. Letzteres lässt sich hingegen leicht durch Erweiterung der Exponentenvektoren um eine weitere Komponente ausgleichen, die 0 im positiven Fall und 1 im negativen Fall gesetzt wird; auch diese Komponente muss sich bei der (multiplikativen) Kombination der Kongruenzen mod 2 zu 0 summieren, da b^2 positiv sein muss.
Bei Bedarf wird während des Experiments das Siebintervall noch erweitert.
Übliche Anzahlen der Elemente eines Siebintervalls sind zwischen $0,2$ Millionen für 50-dezimalstellige und, vom Experiment abhängig, zwischen 2 Millionen (bei günstiger Polynomwahl, s.u.) und 100 Millionen für 135-dezimalstellige Zahlen (s.[Leyland et al. 2002]).

3.4 Der Siebschritt

Betrachtet werden nun alle x aus dem Siebintervall, die glatt bzgl. der Faktorbasis F_B sind. Wie erkennt man, ob $q(x)$ glatt ist oder nicht? Für jede Zahl x aus dem Siebintervall könnte man $q(x)$ für alle Werte $p \in \hat{F}_B$ durch die jeweils höchstmögliche p-Potenz teilen. Die Primfaktorzerlegung durch Probedivisionen herausfinden zu wollen, ist aber sehr aufwendig. Eine bessere Möglichkeit, die Glattheit festzustellen, ist das folgende von Carl Pomerance durch Komplexitätsüberlegungen entwickelte Siebverfahren.
(In seinem hübschen Aufsatz [Pomerance 1996] beschreibt Pomerance diese Überlegungen. Auch erwähnt er seine Anfangsmotivation, nämlich dass ihm auf der High-School in einem Wettbewerb nicht die Faktorisierung der Zahl 8051 innerhalb der gesetzten Zeit von 5 Minuten gelang. Die Lösung wäre wieder die Darstellung als Differenz zweier Quadrate gewesen:

$$8051 = 8100 - 49 = 90^2 - 7^2 = (90 + 7)(90 - 7).\ )$$

Zu $p \in F_B$ sucht man diejenigen Zahlen $x \in \{m, m+1, \ldots m + (p-1)\}$ (mit $m = \lceil\sqrt{n}\rceil$), für die $q(x)$ durch p teilbar ist und wendet (3.3) an; sind diese Zahlen nicht offensichtlich, so kann man wie folgt verfahren: Das Polynom $q(X) := X^2 - n$ hat modulo p (also im Körper $\mathbb{F}_p$) wegen $p \in \hat{F}_B$ zwei Nullstellen. (Zur Bestimmung der Lösungen gibt es spezielle dafür geeigneten Algorithmen). Geht man von diesen Nullstellen durch Addition von

geeigneten Vielfachen von p ins Siebintervall und dort in Schritten der Länge p nach rechts und links durch das Siebintervall, so findet man alle Werte $x \in S$, für die $q(x)$ durch p teilbar ist (vgl. die Kongruenz (3.3), s. § 3.2). Die entsprechenden Zahlen $q(x)$ teilt man jeweils durch p, statt sie, wie beim Sieb des Eratosthenes, zu streichen. Diesen Vorgang nennt man *"Sieben mit p"*. Man vermeidet so erfolglose Probedivisionen. Ein Problem sind höhere Potenzen von p, die man ebenfalls durch Sieben erkennt und durch die man $q(x)$ gegebenenfalls dividiert; dabei sind wieder nur diejenigen $q(x)$ zu betrachten, die schon durch p teilbar waren. Falls nach allen solchen Divisionen mit jedem $p \in \hat{F}_B$ von $q(x)$ nur 1 oder -1 übrig bleibt, enthält die Primfaktorzerlegung von $q(x)$ nur Potenzen von Primzahlen aus der Faktorbasis, und $q(x)$ ist glatt. Alle anderen Werte x und $q(x)$ können gestrichen werden.

3.5 Zusammenfassung

Der Algorithmus besteht aus vier wesentlichen Schritten:

1. Wahl des Siebintervalls $S = \{m-s, m-s+1, \ldots, m, m+1, \ldots, m+s\}$ und Berechnung der Liste der $q(x) = x^2 - n$ mit $x \in S$; siehe § 3.3 !

2. Wahl einer Faktorbasis $F_B = \{p \in \mathbb{P} \,|\, p \leq B\} \cup \{-1\}$ und Sieben mit allen Primzahlen der Faktorbasis; siehe § 3.2 und § 3.4 !
   Ergebnis ist eine Liste aller Zahlen $x \in S$ und $q(x)$ mit glattem $q(x)$.

3. Auswahl der zu kombinierenden Kongruenzen durch Lösung eines Linearen Gleichungssystems; siehe § 3.1 !

4. Überprüfung der Bedingungen $a^2 \equiv b^2 \pmod{n}$ und $a \not\equiv \pm b \pmod{n}$ und Berechnung der Faktoren von n mittels $\mathrm{ggT}(a \pm b, n)$.

Eine ausführliche Darstellung und Zusammenfassung findet man z.B. in [Crandell & Pomerance 2005] § 6.1. Dort wird auch erläutert, dass (bei geeigneter Wahl von B, von S und des Lösungsverfahrens für das lineare Gleichungssystem) die Laufzeit des Algorithmus (für ungerades zusammengesetztes n, das keine Potenz ist,) $e^{(1+o(1))\sqrt{\ln n \ln\ln n}}$ beträgt und damit subexponentiell ist, d.h. von der Form $n^{o(1)}$.

Quadratische Reste

Eine ganze Zahl a heißt *quadratischer Rest* (engl. quadratic residue) modulo m (für eine ungerade natürliche Zahl m), falls sie zu m teilerfremd ist und es eine ganze Zahl x gibt mit

$$x^2 \equiv a \pmod{m}.$$

Existiert keine solche Zahl, so heißt ein zu m teilerfremdes a ein "quadratischer Nichtrest" mod m.

Beispiele: $a = 2$ ist quadratischer Rest modulo 7 mit "Wurzel" ± 3 (wegen $(\pm 3)^2 \equiv 2 \pmod{7}$).
$a = 3$ ist quadratischer Nichtrest mod 5 (wegen $x^2 \bmod 5 \in \{0, 1, 4\}$).

Reduktion auf Primzahl-Moduln: Man kann zeigen, dass *für ungerades m a genau dann quadratischer Rest* mod m *ist, wenn a auch für alle Primteiler p von m quadratischer Rest* mod p *ist.*

Für einfacheres Rechnen mit quadratischen Resten dienen das Legendre- und das Jacobi-Symbol. Dabei ist für eine ganze Zahl a und eine Primzahl $p > 2$ das **Legendre-Symbol** $\left(\frac{a}{p}\right)$ (gelesen als "a nach p" oder "a für p") definiert als:

$$\left(\frac{a}{p}\right) := \begin{cases} 1 & \text{falls } a \text{ quadratischer Rest modulo } p \text{ ist} \\ -1 & \text{falls } a \text{ quadratischer Nichtrest modulo } p \text{ ist} \\ 0 & \text{falls } p \text{ Teiler von } a \text{ ist.} \end{cases}$$

Aus dem kleinen Satz von Fermat ($a^p \equiv a \pmod{p}$) folgt u.a. $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$, falls p nicht a teilt, und daraus das **Eulersche Kriterium:**

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p} \quad \text{(für Primzahlen } p > 2\text{)}.$$

Das **Jacobi-Symbol** definiert man mit Hilfe des Legendre-Symbols:

$$\left(\frac{a}{m}\right) := \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{r_i} \quad \text{für die Primfaktorzerlegung } m = \prod_{i=1}^{k} p_i^{r_i} \text{ von } m.$$

Das Legendere-Symbol ist also der Spezialfall des Jacobi-Symbols mit $m = p$ für eine Primzahl p, stimmt damit in diesem Fall mit diesem überein.
Man beachte, dass $\left(\frac{a}{m}\right) = 1$ auch für Nichtreste, dass $\left(\frac{a}{m}\right) = -1$ aber nicht für quadratische Reste gelten kann. Das Jacobi-Symbol und damit auch das Legendre-Symbol genügen (für positive ungerade Zahlen P bzw. P und Q mit $\mathrm{ggT}\,(P, Q) = 1$) folgenden Gesetzen:

- $\left(\frac{a \cdot b}{P}\right) = \left(\frac{a}{P}\right) \cdot \left(\frac{b}{P}\right)$ und $\left(\frac{a}{P}\right) = \left(\frac{a \bmod P}{P}\right)$.
- **Quadratisches Reziprozitätsgesetz** (law of quadratic reciprocity):
$$\left(\frac{P}{Q}\right) \cdot \left(\frac{Q}{P}\right) = (-1)^{(P-1)(Q-1)/4}.$$
- "Ergänzungssätze" zum Quadratischen Reziprozitätsgesetz:
$$\left(\frac{-1}{P}\right) \equiv (-1)^{(P-1)/2} \text{ und } \left(\frac{2}{P}\right) \equiv (-1)^{(P^2-1)/8}.$$

3.6 Ein einfaches Beispiel mit Sieben

Zu faktorisieren sei die Zahl $n = 22.213$.
(Weitere Beispiele findet man in der Literatur, z.B. die Faktorisierung von $n = 87.463$ in [Hulpke 2004] oder in 'Quadratisches Sieb' von [Wikipedia de], von $n = 2.041$ in [Esslinger et al. 2011], von $n = 4.309$ und $n = 7.729$ in [Müller & Piontkowski 2011] und von $n = 15.347$ in 'quadratic sieve' von [Wikipedia en].

Mit Probedivisionen (wie sich später zeigt, wären 25 Divisionen durch Primzahlen bis zu einem der Faktoren nötig) könnte man das Problem lösen. Wir wollen hier hingegen das Quadratische Sieb verwenden. Dass man das Beispiel mit Hilfe eines Taschenrechners lösen kann, ist nicht so entscheidend, man kann auch die einzelnen Operationen mit mathematischer Software ausführen, aber ohne Faktorisierungs-Befehl. Es geht darum, Schritt für Schritt einen konkreten Einblick in die Methode zu erhalten.

Zunächst bestimmen wir $m = \lceil \sqrt{n} \rceil = \lceil 149,04\ldots \rceil = 150$ und versuchen folgende Setzung: $B = 19$. Damit ergibt sich als Faktorbasis (unter Berücksichtigung der Anmerkung von § 3.2) :

$$F = \{-1\} \cup \{p \leq 19 \mid p \text{ prim und } n \text{ quadratischer Rest modulo } p\} \subseteq F_B.$$

Zunächst bestimmen wir F explizit, indem wir prüfen, für welche Primzahlen p aus $F_B = \{-1\} \cup \{p \mid p \text{ prim und } p \leq 19\}$ die Zahl n quadratischer Rest mod p ist. Durch Reduktion modulo p sehen wir, dass

$n \equiv 1 \pmod 3$ ist (mit Wurzel ± 1) und $n \equiv 1 \pmod 9$,
$n \equiv 3 \pmod 5$ ist, also kein quadratischer Rest,
$n \equiv 2 \pmod 7$ ist (mit Wurzel ± 3),
$n \equiv 4 \pmod{11}$ ist (mit Wurzel ± 2),
$n \equiv 9 \pmod{13}$ ist (mit Wurzel ± 3),
$n \equiv 11 \pmod{17}$ (s..u.) und
$n \equiv 2 \pmod{19}$.

In den letzen beiden Fällen liegen keine quadratischen Reste vor, was man direkt sehen kann oder mit Hilfe des Satzes, dass für ungerade Primzahlen p gilt: $n^{(p-1)/2} \mod p$ ist gleich 1, falls n quadratischer Rest mod p ist, und gleich -1 für quadratische Nicht-Reste; (s. das Euler-Kriterium im Kasten "Quadratische Reste"). Hier ist $11^8 \equiv 2^4 \equiv -1 \pmod{17}$ und $2^9 + 1 \equiv 0 \pmod{19}$. Somit ergibt sich

$$F = \{-1, 2, 3, 7, 11, 13\}.$$

Damit haben die Exponentvektoren Länge 6. In der Hoffnung, dass genügend Werte $q(x)$ glatt sein werden und linear abhängige Exponentenvektoren haben, verwenden wir das Siebintervall

$$S = \{x \in \mathbb{N} \mid 150 - 7 \leq x \leq 150 + 7\} = \{143, 144, 145, 146, 147, \ldots, 157\}.$$

3.6.1 Vorbereitung zum Sieben

Für x aus dem Intervall S sind genau alle $q(x) = x^2 - 22.213$

- mit x ungerade durch 2 teilbar,
- mit $x \in \{150 \pm 1 + t \cdot 3\} = \{143, 145, 146, 148, 149, 151, 152, 154, 155, 157\}$ durch 3 teilbar,
  Begründung: 150 ist durch 3 teilbar und im Siebintervall und $x = 150 \pm 1 \equiv \pm 1 \pmod 3$ erfüllt $q(x) \equiv x^2 - n \equiv (\pm 1)^2 - 1 \equiv 0 \pmod 3$. Nun wird (3.3) angewandt.
- mit $x \in \{150 \pm 1 + t \cdot 9\} = \{143, 145, 152, 154\}$ durch 3^2 teilbar,
- mit $x \in \{147 \pm 3 + t \cdot 7\} = \{143, 144, 150, 151, 157\}$ durch 7 teilbar,
- mit $x \in \{154 \pm 2 + t \cdot 11\} = \{145, 152, 156\}$ durch 11 teilbar,
- mit $x \in \{143 \pm 3 + t \cdot 13\} = \{146, 153\}$ durch 13 teilbar.

Man erhält:

3.6.2 Liste nach dem Sieben

x	$q(x) = x^2 - 22.213$	
143	-1764	$= (-1) \cdot 2^2 \cdot 3^2 \cdot 7^2$ glatt
144	-1477	$= (-1) \cdot 7 \cdot 211$ nicht glatt
145	-1188	$= (-1) \cdot 2^2 \cdot 3^3 \cdot 11$ glatt
146	-897	$= (-1) \cdot 3 \cdot 13 \cdot 23$ (evtl. 23 in die Faktorbasis aufnehmen)
147	- 604	nicht glatt: durch kein p aus F außer durch 2 teilbar (s.§ 3.6.1); keine 2-Potenz
148	-309	nicht glatt: durch kein Element von $F \setminus \{3\}$ teilbar (s.§ 3.6.1); keine 3-Potenz
149	-12	$= (-1) \cdot 2^2 \cdot 3$ glatt
150	287	nicht glatt: durch kein p aus F außer durch 7 teilbar; keine 7-Potenz
151	588	$= 2^2 \cdot 3 \cdot 7^2$ glatt
152	891	$= 3^4 \cdot 11$ glatt
153	1196	$= 2^2 \cdot 13 \cdot 23$ (evtl. 23 in die Faktorbasis aufnehmen)
154	1503	nicht glatt: durch kein Element von F außer 3 teilbar, keine 3-Potenz)
155	1812	$= 2^2 \cdot 3 \cdot 151$ nicht glatt
156	2123	nicht glatt: durch kein Element von F außer 11 teilbar; keine 11-Potenz
157	2436	$= 2^2 \cdot 3 \cdot 7 \cdot 29$ nicht glatt

3.6.3 Kombination von Kongruenzen

Es ergeben sich so als binäre Exponentenvektoren der glatten Zahlen die Vektoren der folgenden Tabelle:

x / p	-1	2	3	7	11	13
143	1	0	0	0	0	0
145	1	0	1	0	1	0
149	1	0	1	0	0	0
151	0	0	1	0	0	0
152	0	0	0	0	1	0

Z.B. die Exponentenvektoren zu $x_1 = 145, x_2 = 149$ und $x_3 = 152$ addieren sich zum Nullvektor. Daher kann man $a = 145 \cdot 149 \cdot 152 = 3.283.960$ setzen und b wie folgt bestimmen: $b^2 = (-1) \cdot 2^2 \cdot 3^3 \cdot 11 \cdot (-1) \cdot 2^2 \cdot 3 \cdot 3^4 \cdot 11 = 2^4 \cdot 3^8 \cdot 11^2$ und somit $b = 2^2 \cdot 3^4 \cdot 11 = 3.564$. Leider stellt sich nun heraus, dass $a + b = 3.287.524$ durch n teilbar, also die Bedingung (1.2) verletzt ist. Ebenfalls führt die Kombination der Exponentialvektoren zu $x = 143, 145, 151$ und 152 zu keinem Erfolg: Mit $a = 143 \cdot 145 \cdot 151 \cdot 152 = 475.909.720$, $b^2 = (-1) \cdot 2^2 \cdot 3^2 \cdot 7^2 \cdot (-1) \cdot 2^2 \cdot 3^3 \cdot 11 \cdot 2^2 \cdot 3 \cdot 7^2 \cdot 3^4 \cdot 11 = 2^6 \cdot 3^{10} \cdot 7^4 \cdot 11^2$ und $b = 2^3 \cdot 3^5 \cdot 7^2 \cdot 11 = 1.047.816$ gilt erneut $a + b\,(= 476.957.536) \equiv 0 \pmod{n}$.
Auch nach Erweiterung von F zu $F \cup \{23\}$ und Wahl von $x = 146, 149$ und 153 erhält man mit $a = 146 \cdot 149 \cdot 153 = 3.328.362, b = 2^2 \cdot 3 \cdot 13 \cdot 23 = 3.588$ und $a + b = 3.331.950 \equiv 0 \pmod{n}$ keine Lösung.
Anstatt nun nach anderen Mengen linear abhängiger Exponentenvekoren bei erweitertem Siebintervall und vergrößerter Faktorbasis zu suchen, werden wir das Beispiel im nächsten Paragraphen mit einem anderen Polynom fortsetzen.

4 Sieben mit mehrfachen Polynomen (MPQS)

4.1 Das Verfahren

Da beim Quadratischen Sieb die Zahlen $q(x)$ mit wachsendem Abstand von x zu $\sqrt{n}$ schnell wachsen und da man auch parallel rechnen will, benutzt man statt des einzigen Polynoms $q(x) = x^2 - n$ beim "Multiple Polynomial Quadratic Sieve (MPQS)" mehrere geeignet gewählte Polynome der Form: $q_g(x) = gx^2 + 2hx + j \quad \text{mit} \quad g, h, j \in \mathbb{Z} \quad \text{und} \quad h^2 - gj = n$. Lässt sich dann die Kongruenz $h^2 \equiv n \pmod{g}$ lösen, so ist $j = (h^2 - n)/g$ wählbar. So erhält man

$$Q_g(x) := g \cdot q_g(x) = g^2x^2 + 2ghx + gj = (gx + h)^2 - n \tag{4.1}$$

und daraus

$$(gx + h)^2 \equiv Q_g(x) \pmod{n}. \tag{4.2}$$

Diese Kongruenzen kann man wieder multiplizieren (sogar für verschiede Q_g), sodass wieder eine Kombination von Kongruenzen (nach dem Sieben mit

Primzahlen aus F_B) möglich ist. Wie beim Quadratischen Sieb kommen als ungerade Primteiler von $Q_g(x)$ nur solche Primzahlen p in Frage, für die nach (4.1)

$$0 \equiv g \cdot q_g(x) = (gx+h)^2 - n \pmod{p}$$

gilt und für die daher n quadratischer Rest mod p ist. Damit kann man die beim Sieben verwendeten Primzahlen unabhängig von den Polynomen wählen.

Sei nun g Produkt eines Quadrats mit einer glatten Zahl; z.B. wählt man $g = p^2$ für eine Primzahl p, modulo der die Zahl n quadratischer Rest ist; die Kongruenz $h^2 \equiv n \pmod{g}$ ist dann lösbar, da n dann auch quadratischer Rest modulo p^2 ist (s. den Kasten "Quadratische Reste"). Von den beiden Lösungen wählt man eine bzw. eine dazu kongruente aus.
Ergibt sich für ein x ein glattes $q_g(x)$, so erhalten wir mittels $(gx+h)^2 - n$ einen der gesuchten modulo 2 reduzierten Exponentenvektoren für unser Gleichungssystem, durch das wir wie beim Quadratischen Sieb eine Darstellung von n oder eines Vielfachen von n als Differenz zweier Quadrate finden und so evtl. zu einer Faktorisierung gelangen. Mit den Werten $\tilde{x} := gx + h$ statt x geht man also wie beim Quadratischen Sieb vor.

4.2 Fortsetzung des Beispiels aus §3.6

Wie in §3.6 sei $n = 22.213$ zu faktorisieren. Statt des Polynoms $x^2 - n$ verwenden wir nun ein anderes Polynom (wir hoffen, mit einem auszukommen) der Form $(gx+h)^2 - n$. Wieder ist $m = 150$. Wir wählen $g = 4$ und ein h mit $h^2 \equiv 22.213 \pmod{g}$, also $h^2 \equiv 1 \pmod 4$ d.h. $h \equiv \pm 1 \pmod 4$. Damit wir nicht zu große Zahlen erhalten, sei $(4 \cdot 150 + h)^2 \approx n$. also $h \approx \sqrt{n} - 600 \approx -450$; daher setzen wir $h = -451$ und somit

$$Q_g(x) = g \cdot q_g(x) = (gx+h)^2 - n = (4x - 451)^2 - 22.213.$$

Als Faktorbasis wählen wir wieder $F = \{-1, 2, 3, 7, 11, 13\}$ und als Siebintervall diesmal $S = \{x \in \mathbb{Z} \mid 140 \le x \le 160\}$. Durch Sieben mit den Zahlen aus F erhalten wir die Zerlegungen von $Q_g(x)$ der folgenden Tabelle; (diese enthält mit $\tilde{x} := gx + h$ auch Teile der Liste aus § 3.6.2).

x	4x-451	$Q_g(x)$	$=(4x-451)^2-22.213=4\cdot q_4(x)$
140	109	-10.332	$=(-1)\cdot 2^2\cdot 3^2\cdot 7\cdot 41$ glatt bzgl. $F\cup\{41\}$
141	113	-9.444	$=(-1)\cdot 2^2\cdot 3\cdot 787$ nicht glatt
142	117	-8.524	$=(-1)\cdot 2^2\cdot 2131$ nicht glatt
143	121	-7.572	$=(-1)\cdot 2^2\cdot 3\cdot 631$ nicht glatt
144	125	-6.588	$=(-1)\cdot 2^2\cdot 3^3\cdot 61$ nicht glatt
145	129	-5.572	$=(-1)\cdot 2^2\cdot 7\cdot 199$ nicht glatt
146	133	-4.524	$=(-1)\cdot 2^2\cdot 3\cdot 13\cdot 29$ glatt bzgl. $F\cup\{29\}$
147	137	-3.444	$=(-1)\cdot 2^2\cdot 3\cdot 7\cdot 41$ glatt bzgl. $F\cup\{41\}$
148	141	-2.332	$=(-1)\cdot 2^2\cdot 11\cdot 53$ nicht glatt
149	145	-1.188	$=(-1)\cdot 2^2\cdot 3^3\cdot 11$ glatt
150	149	-12	$=(-1)\cdot 2^2\cdot 3$ glatt
151	153	1.196	$=2^2\cdot 13\cdot 23$ glatt bzgl. $F\cup\{23\}$
152	157	2.436	$=2^2\cdot 3\cdot 7\cdot 29$ glatt bzgl. $F\cup\{29\}$
153	161	3.708	$=2^2\cdot 3^2\cdot 103$ nicht glatt
154	165	5.012	$=2^2\cdot 7\cdot 179$ nicht glatt
155	169	6.348	$=2^2\cdot 3\cdot 23^2$ glatt bis auf Quadrat
156	173	7.716	$=2^2\cdot 3\cdot 643$ nicht glatt
157	177	9.116	$=2^2\cdot 43\cdot 53$ nicht glatt
158	181	10.548	$=2^2\cdot 3^2\cdot 293$ nicht glatt
159	185	12.012	$=2^2\cdot 3\cdot 7\cdot 11\cdot 13$ glatt
160	189	13.508	$=2^2\cdot 11\cdot 307$ nicht glatt

Da die Exponentenvektoren der bis auf ein Quadrat glatten Werte linear unabhängig sind, erweitern wir die Faktorbasis zu $\tilde{F} := F\cup\{23,29,41\}$ und erhalten die Exponentenvektoren aus folgender Liste:

x	$gx+h$	-1	2	3	7	11	13	23	29	41
140	109	1	0	0	1	0	0	0	0	1
146	133	1	0	1	0	0	1	0	1	0
147	137	1	0	1	1	0	0	0	0	1
149	145	1	0	1	0	1	0	0	0	0
150	149	1	0	1	0	0	0	0	0	0
151	153	0	0	0	0	0	1	1	0	0
152	157	0	0	1	1	0	0	0	1	0
155	169	0	0	1	0	0	0	0	0	0
159	185	0	0	1	1	1	1	0	0	0

Schon die Exponentenvektoren zu $x_1=140, x_2=147$ und $x_3=155$ addieren sich zum Nullvektor. So erhalten wir: $a=\prod_{i=1}^{3}(gx_i+h)=109\cdot 137\cdot 169=2.523.677$ und $b^2=(-1)\cdot 2^2\cdot 3^2\cdot 7\cdot 41\cdot(-1)\cdot 2^2\cdot 3\cdot 7\cdot 41\cdot 2^2\cdot 3\cdot 23^2$, somit $b=2^3\cdot 3^2\cdot 7\cdot 23\cdot 41=475.272$. Wieder hoffen wir, durch $\mathrm{ggT}(a\pm b,n)$ echte Teiler von n zu finden. Tatsächlich erhalten wir mit $a+b=2.998.949\equiv 194=2\cdot 97 \pmod{n}$ in $\mathrm{ggT}(a+b,n)=97$ einen echten Teiler von $n=22.213$ und in $n/97=229$ den komplementären Teiler. (Mit $a-b=2.048.405\equiv 4809=3\cdot 7\cdot 229$ finden wir letzteren ebenfalls.)

Damit haben wir 22.213 faktorisiert, und zwar mit dem Ergebnis:

$$22.213 = 97 \cdot 229.$$

4.3 Implementierung

Eine Implementierung des Quadratischen Siebs steht (neben alternativen Verfahren) z.B. im freien E-Learning CrypTool Programmpaket 1.4.30 (s. [Cryptool] Download → Einzelverfahren → RSA-Kryptosystem → Faktorisieren einer Zahl) zur Verfügung. Die Quelltextdatei dazu findet man im 'Subversion-Repository' der Download-Seite (als Benutzer 'anonymous' mit leerem Passwort) unter 'IntegerArithmetic.cpp'.

Danksagung
Herrn Bernhard Esslinger (Universität Siegen und CrypTool) danken wir herzlich für Verbesserungsvorschläge bei Entstehung dieses Artikels.

Literatur und Internetquellen

[**Atkins et al.**] Derek Atkins, Michael Graff, Arjen Lenstra und Paul Leyland: "factor RSA-129" (Mitteilung der Zerlegung von RSA-129)
http://www.mit.edu:8001/people/warlord/RSA129-announce.txt

[**Buchmann 2010**] Johannes Buchmann: Einführung in die Kryptographie. Springer Verlag, Wiesbaden, Heidelberg, 1999, 2010[5].

[**Crandall & Pomerance 2005**] Richard Crandall und Carl Pomerance: Prime numbers. A Computational Perspective. Springer V. 2001, 2005[2].

[**CrypTool**] Cryptool Lernprogramm für Kryptographie und Analyse – Download
http://www.cryptool.de/index.php/de/download-topmenu-63.html

[**Esslinger et al. 2011**] Bernhard Esslinger et.al.: CrypTool-Skript: Mathematik und Kryptographie.
http://www.CrypTool.de

[**Hulpke 2004**] Alexander Hulpke: Factorization of $n = 87463$ with the Quadratic Sieve.
http://www.math.colostate.edu/∼hulpke/lectures/m400c/quadsieve.pdf

[**Leyland et al. 2002**] Paul Leyland, Arjen Lenstra, Bruce Dodson, Alec Muffett and Sam Wagstaff: MPQS with Three Large Primes. C.Fieker and D.R.Kohel (Eds.), Algorithmic number theory. 5th international symposium, ANTS-V, Sydney, Australia, July 7–12,

2002. Proceedings. Berlin: Springer-V. Lect. Notes Comput. Sci. 2369, 446-460 (2002).

[**Müller-Stach & Piontkowski 2011**] Stefan Müller-Stach und Jens Piontkowski: Elementare und algebraische Zahlentheorie. Ein moderner Zugang zu klassischen Themen. Vieweg+Teubner 2006, 2011[2].

[**Pomerance 1996**] Carl Pomerance: A Tale of Two Sieves. Notices of the AMS 43 (1996) 1473-1485.
http://www.ams.org/notices/199612/pomerance.pdf

[**Schulz & Witten 2010**] Ralph-Hardo Schulz und Helmut Witten: Zeit-Experimente zur Faktorisierung. Ein Beitrag zur Didaktik der Kryptologie. Log In 166/167 (2010) 113-120.

[**Wikipedia de**] Wikipedia (deutsch) http://de.wikipedia.org/wiki/

[**Wikipedia en**] Wikipedia (englisch) http://en.wikipedia.org/wiki/

[**Witten & Schulz 2006**] Helmut Witten und Ralph-Hardo Schulz: RSA & Co in der Schule. Moderen Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge Teil 2: RSA für große Zahlen. Log In 143 (2006) 50-58.

Die Internetadressen wurden am 27.6.2011 oder später überprüft.

Adresse der Autoren:

Prof. Dr. Ralph-Hardo Schulz
c/o Inst. f. Mathematik der FU Berlin
Arnimallee 3
14195 Berlin
E-Mail: rhschulz@zedat.fu-berlin.de

StudDir a.D. Helmut Witten
Brandenburgische Straße 23
10707 Berlin

E-Mail: helmut@witten-berlin.de