Vorlesung Kryptographie

WiSe '23/'24, hhu
K. Halupczok

K9: Faktorisierung mit Kettenbrüchen

Stichworte: Faktorisierungsproblem, Zahlkörpersieb, Zerlegung mit □en nach Fermat, Faktorbasis, glatte Zahlen, Algorithmus von Brillhart & Morrison

---

Bestimmte Kryptographie-Anwendungen wie z.B. das RSA-Verfahren nutzen die Schwierigkeit des (zahlentheoretischen) Faktorisierungsproblems (vgl. 2.21) aus:

9.1. **Faktorisierungsproblem:** Sei N eine beliebige, große zusammengesetzte nat. Zahl. Man zerlege N in die Primfaktoren, d.h. gib die PFZ an.

Dafür genügt es, ein Verfahren zur Auffindung eines nichttrivialen Teilers t von N anzugeben ($1 \neq t \neq N$) und dieses wiederholt auf $\frac{N}{t}, \frac{N}{t_1 t_2}, \ldots, t_1, t_2, \ldots$ anzuwenden, bis alle Primfaktoren ermittelt sind. (Beachte: $\Omega(N) = \sum_{p^n \| N} k$ erfüllt $\Omega(N) = O(\log(N))$, im Mittelwert, d.h. "meistens", ist $\Omega(N)$ etwa $\log\log(N)$ nach dem Satz von Erdös–Kac aus dem Jahr 1940.)

Man sollte vorher bereits sichergestellt haben, dass N nicht prim=zus.gesetzt ist:

9.2. **Primzahltestproblem:** Sei N eine beliebige, große nat. Zahl. Entscheide, ob N eine Primzahl oder zusammengesetzt ist.

9.3. Aufgrund der Ergebnisse aus Kapitel K7, K8 gehen wir davon aus, dass das PZtestproblem leicht zu lösen ist und untersuchen das Problem für eine (laut Test) zusgs. Zahl. Die KBE von $\sqrt{N}$ liefert ein Verfahren, N relativ rasch zu faktorisieren, und geht auf [Morrison/Brillhart 1975] zurück. Bis in die 1980er Jahre war es das am schnellsten bekannte Verfahren, das für N mit etwa ≤ 50 Dezimalziffern funktionierte: das erste mit einer erwarteten subexponentiellen Laufzeit (d.h. $o(N^\varepsilon) = o(e^{\varepsilon \log N})$ für jedes $\varepsilon > 0$, hier ist $O(\log N)$ die Inputgröße, d.h. die Anzahl Stellen/Ziffern von N).

9.4. Heute ist nach wie vor das sogenannte Zahlkörpersieb (um 1990) am schnellsten mit einer Laufzeit von $\exp(O(\sqrt[3]{(\log N)\cdot(\log\log N)^2}))$; mittlerweile sind auch schnelle Faktorisierungsalgorithmen mit elliptischen Kurven bekannt (ECM = elliptic curve method; nach [Lenstra 1987]. Das Zahlkörpersieb ist eine der Weiterentwicklungen des KBE-Faktorisierungsalgorithmus von Morrison/Brillhart, das wir in diesem § behandeln. Im Zahlkörpersieb wird intensiv die Arithmetik von Zahlringen algebraischer Zahlkörper (aus ZT I, Z3–Z19) benutzt.

9.5. Eine alte Idee von Fermat: Gibt es $x,y\in\mathbb{Z}$ mit $x^2\equiv y^2\ (N)$ und $x\not\equiv\pm y\ (N)$, dann ist $(N, x+y)$ ein nichttrivialer Faktor von N.

Denn: $0\equiv x^2-y^2=\underbrace{(x-y)}_{\not\equiv 0(N)}\underbrace{(x+y)}_{\not\equiv 0(N)}\ (N)$ $\Rightarrow$ $x-y$, $x+y$ haben gem. Teiler mit N, der jeweils $\neq N$, dann also auch jeweils $\neq 1$ ist.

Wie verschafft man sich solche Zahlen $x,y\in\mathbb{Z}$?
Man suche nach hinreichend vielen Quadratzahlen in derselben Restklasse mod N, so dass Kombinationen von ihnen auf nichttriviale Teiler führen:

9.6. Idee: Gibt es genügend viele Kongruenzen $X_j^2\equiv(-1)^{\varepsilon_{0j}}\ell_1^{\varepsilon_{1j}}\ell_2^{\varepsilon_{2j}}\cdots\ell_m^{\varepsilon_{mj}}\ (\mathrm{mod}\ N)$, $j\in k$, wo die ℓ_i kleine Pzen und ε_{ij} bestimmte Exponenten sind, so hofft man, mit einer Gaußelimination auf Zahlen $\delta_j\in\{0,1\}$, $j\in k$, zu kommen, so dass $\sum_{j\in k}\delta_j(\varepsilon_{0j},\dots,\varepsilon_{mj})^T\equiv(0,\dots,0)^T\ (\mathrm{mod}\ 2)$ (*) gilt.

Denn dann ist mit (+) $x=\prod_{j\in k}X_j^{\delta_j}$, $y=(-1)^{v_0}\ell_1^{v_1}\ell_2^{v_2}\cdots\ell_m^{v_m}$, $\sum_{j\in k}\delta_j(\varepsilon_{0j},\dots,\varepsilon_{mj})=2(v_0,\dots,v_m)$ richtig, dass

$$x^2\equiv\prod_{j\in k}(-1)^{\delta_j\varepsilon_{0j}}\ell_1^{\delta_j\varepsilon_{1j}}\cdots\ell_m^{\delta_j\varepsilon_{mj}}\equiv\prod_{j\in k}(-1)^{2v_0}\cdot\ell_1^{2v_1}\cdots\ell_m^{2v_m}=y^2\ (N)$$ gilt.

Für $x\not\equiv\pm y\ (N)$ liefert dies wie in 9.5 einen nichttrivialen Teiler von N.

9.7. **Def.:** Eine Menge $\mathcal{B} := \{-1, \ell_1, \dots, \ell_m\}$ mit Primzahlen $\ell_i \le B$ für ein $B > 0$ heißt Faktorbasis für N.

9.8. **Bem.:** Nat. Zahlen, die nur Pzen $\le B$ in ihrer PFZ besitzen (d.h. $p \mid N \Rightarrow p \le B$), heißen B-glatt. Die Zählfunktion $\Psi(x, B) := \#\{n \in \mathbb{N};\ n \le x,\ p \mid n \Rightarrow p \le B\}$ muss für Fragestellungen der Laufzeitbestimmung in der algor. ZT häufig abgeschätzt werden, etwa zur Laufzeitbestimmung des KBE-Algorithmus 9.10 oder des ZKSiebs.

Die anal. ZT hat dafür sehr präzise Antworten parat, ein fundamentales Ergebnis für $\Psi(x, B)$ ist folgendes.

9.9. **Satz (Dickman–de Bruijn):** Für $x \ge 2$, $U > 0$ gilt glm. in $0 \le u \le U$:
$\Psi(x, x^{1/u}) = \rho(u)x + O\left(\frac{x}{\log x}\right)$, wobei die Dickman-Funktion ρ def. ist durch
$\rho(u) \equiv 1$ für $0 \le u \le 1$ und $u\rho'(u) = -\rho(u-1)$ für $u > 1$.
Wir haben $\frac{\hat{1}}{2\Gamma(2u+1)} \le \rho(u) \le \frac{\hat{1}}{\Gamma(u+1)}$ für $u > 0$. Für $A > 1$ ist $\Psi(x, \log^A x) = x^{1-1/A+o(1)}$.

Beweis-Skizze (für die Asymptotik):
Da ρ für $u > 1$ streng monoton fällt und $\rho(u) \ge 0$, folgt $0 < \rho(u) < 1$ für alle $u > 1$.

1.) Zeige $\Psi(x,y) = \Psi(x,z) - \sum_{y < p \le z} \Psi\left(\frac{x}{p}, p\right)$ für alle $x \ge 2$, $z \ge y \ge 2$.
"Buchstaben-Identität"

2.) Schreibe $u = \frac{\log(x)}{\log(y)}$, führe eine Induktion über $k \in \mathbb{N}$ für $u \le k$ durch.
$k = 1$: Beh. trivial, denn $u \ge 1$ heißt $y \ge x$ und $\Psi(x,y) = \lfloor x \rfloor = x + O(1)$.
$k \rightsquigarrow k+1$: Sei die Beh. für $u \le k \Leftrightarrow y \ge x^{1/u}$ bekannt. Für $k < u < k+1$ folgt aus 1.) dann $\Psi(x,y) = \Psi(x, x^{1/2}) - \sum_{y < p \le x^{1/2}} \Psi\left(\frac{x}{p}, p\right)$
(Ind.vor. anwendbar für $p > y \ge x^{\hat{1}/k+1}$)

- Falls $k = 1$: $\Psi\left(\frac{x}{p}, p\right) = \lfloor \frac{x}{p} \rfloor$ für $x^{1/2} < p \le x$, also $\Psi(x,y) = x\left(1 - \sum_{y < p \le x} \hat{\frac{1}{p}}\right) + O\left(\frac{x}{\log(x)}\right)$,
$= \log\log x - \log\log y + O\left(\hat{\frac{1}{\log y}}\right)$, $\log(u) = \log\log x - \log\log y$
also $\rho(u) = 1 - \log(u)$.

Der Fall $k \ge 2$ ist komplizierter, s. [Brüdern, Anal. ZT, S. 140ff.]. (Mertens) s. Anz 21.4 $\square$

Wir beschreiben nun den bekannten KBE-Algo zur Faktorisierung von N; dabei bezeichne "m mod N" die ganze Zahl m mit $0 \le m < N$.

9.10. **Algorithmus von Brillhart & Morrison:** Für $j = 0,1,2,3,\ldots$ führe aus:

1.) Berechne den j-ten NB $\frac{c_j}{d_j}$ in der KBE von $\sqrt{N}$.

2.) Berechne c_j^2 mod N. Nachdem dies für mehrere j bereits geschehen ist, schaue auf die $\pm c_j^2$ mod N, die in ein Produkt aus lauter kleinen PZen zerfallen.
Def. $\mathcal{B}$ als die Menge -1 und aller PZen, die in mehr als einem der c_j^2 mod N aufgehen, oder in genau einem c_j^2 mod N und darin mit geradem Exponenten.

3.) Erstelle eine Liste mit allen c_j^2 mod N, die als Produkt von Zahlen in der Faktorbasis $\mathcal{B}$ dargestellt werden können.
Falls möglich, erstelle eine Teilmenge von ℓ's in $\mathcal{B}$ so, dass die zugehörigen Exponenten ε in $\mathcal{B}$ sich zu 0 mod 2 addieren gemäß ⊛ in 9.6.
Def. x, y wie in ⊕. Ist $x \not\equiv \pm y$ mod N, so ist $(x+y, N)$ nichttriv. Faktor von N. ✓
Ist dies unmöglich, erweitere die Faktorbasis $\mathcal{B}$ um weitere c_j und c_j^2 mod N.

Für die Zähler c_j der NBe von $\sqrt{N}$ ist c_j^2 mod N ein kleiner Rest, so dass wir davon ausgehen können, dass das Verfahren 9.10 rasch funktioniert:

9.11. **Satz:** Sei $\alpha > 1$ irrational. Dann erfüllen die NBe $\frac{c_j}{d_j}$ an α die Ungl. $|d_j^2\alpha^2 - c_j^2| < 2\alpha$.
Ist speziell $\alpha = \sqrt{N}$, wo $N \in \mathbb{N}$ keine Quadratzahl, dann ist c_j^2 mod N kleiner als $2\sqrt{N}$.

Bew.: Haben $|d_j^2\alpha^2 - c_j^2| = d_j^2 \left|\alpha - \frac{c_j}{d_j}\right| \cdot \left|\alpha + \frac{c_j}{d_j}\right| \le d_j^2 \underbrace{\left|\alpha - \frac{c_j}{d_j}\right|}_{< \frac{1}{d_j d_{j+1}}} \cdot \left(2\alpha + \underbrace{\left|\alpha - \frac{c_j}{d_j}\right|}_{< \frac{1}{d_j d_{j+1}}}\right)$ nach Lemma 22.1.3 in ZT I.

Also ist $|d_j^2\alpha^2 - c_j^2| < \frac{d_j^2}{d_j d_{j+1}}\left(2\alpha + \frac{1}{d_j d_{j+1}}\right)$,
also $|d_j^2\alpha^2 - c_j^2| - 2\alpha < 2\alpha\left(-1 + \frac{d_j}{d_{j+1}} + \frac{1}{2\alpha d_{j+1}^2}\right) < 2\alpha\left(-1 + \frac{d_j + 1}{d_{j+1}}\right) \le 0$. □

9.12. Bem.: Würde man eher große $\square_e \bmod N$ bekommen, würde $|x^2 - N|$ rasch wachsen, was die Chance mindert, mit einer solchen Faktorbasis $x^2 - N$ faktorisieren zu können.

9.13. Bsp.: Betr. $N = 8777$, NBe: $\frac{93}{1}, \frac{94}{1}, \frac{281}{3}, \frac{1499}{16}, \ldots \to \sqrt{8777} = [93; 1,2,5,1,\ldots]$.

Somit:

j	0	1	2	3
$c_j \bmod N$	93	94	281	1499
$c_j^2 \bmod N$	-128	59	-32	89

$\to$ Kleine $\square_e$ leicht zu faktorisieren $\to \mathcal{B} = \{-1, 2\}$.

Damit sind die $c_j^2 \bmod N$ zerlegbar über $\mathcal{B}$ für $j = 0$, $j = 2$:

$$-128 = (-1)^1 \cdot 2^7, \quad -32 = (-1)^1 \cdot 2^5$$

Die zugehörigen Exponenten ε_{ij} hier sind $(\varepsilon_{00}, \varepsilon_{10}) = (1,7)$, $(\varepsilon_{02}, \varepsilon_{12}) = (1,5)$.

Die Summen dieser Vektoren addieren sich in jeder Komponente zu $0 \bmod 2$, also $\delta_1 = \delta_2 = 1$ und $x = \underbrace{93}_{c_0 = x_0} \cdot \underbrace{281}_{c_2 = x_2} \equiv -198 \bmod 8777$, $y = \underbrace{-2^6}_{= (-1)^{\frac{1+1}{2}} \cdot 2^{\frac{7+5}{2}}} = -64$.

Es folgt $198^2 \equiv 64^2 \bmod 8777$, da $198 \not\equiv \pm 64 \bmod 8777$ ist

$(198 + 64, 8777) = 131$ ein nichttriv. Teiler von 8777!
(↑ euklid. Algo)

Probe-Division bestätigt $8777 = 67 \cdot 131$.

9.14. Bem.: 1.) Bekannt ist, dass 9.10 nicht für Primpotenzen $N = p^k$ funktioniert; dies kann aber mit einem (schnellen) Test leicht entschieden werden. [$2 \le k \le \log(N)$, $\sqrt[k]{\ }$, Pz-test]

2.) Hat $\sqrt{N}$ zu kurze Periode, kann 9.10 nur $\mathcal{B}$ mit $\#\mathcal{B}$ klein produzieren, was die Chancen zum Erfolg schmälert. Unter Umständen kann die Betrachtung von $\sqrt{cN}$ helfen.

3.) Algo hat subexponentielle Laufzeit, kann mit zahlth. glatter Zahlen heuristisch abgeschätzt werden als $O\left(\exp\left((\sqrt{2} + o(1)) \sqrt{\log(N) \log\log(N)}\right)\right)$.

4.) Weitere, vorteilhafte Faktorisierungsalgorithmen sind Pollards Rho-Methode mit Laufzeit $O(\sqrt{p} \log^2(N)) = O(N^{1/4} \log^2(N))$ für einen Primteiler $p = O(\sqrt{N})$ von N, und Pollards $(p-1)$-Methode, die besonders schnell ist wenn ein $p \mid N$ mit glattem $p-1$ existiert. Lenstras ECM-Methode hat Laufzeit $O\left(\exp\left(\sqrt{(\sqrt{2} + o(1)) \log(p) \log\log(p)}\right) \log^2(N)\right)$, wo $p \mid N$ minimal.