Algoritmi u teoriji brojeva

Andrej Dujella

PMF – MO, Sveučilište u Zagrebu, 2021.

Sadržaj

Poglavlje 1

Osnovni algoritmi iz teorije brojeva

1.1 Složenost algoritama

Često se za neke matematičke probleme kaže da su "laki" ili "teški". Primjerice, za sigurnost i efikasnost najpoznatijih kriptosustava s javnim ključem važno je da je testiranje prostosti "lakše" od faktorizacije, te da je u nekim konačnim grupama potenciranje "lakše" od logaritmiranja . Da bismo mogli formalizirati te vrlo neformalne tvrdnje, trebamo imati način za usporedbu efikasnosti različitih algoritama.

Pod algoritmom smatramo metodu (proceduru) za rješavanje neke klase problema, koja za ulazne podatke određenog tipa daje odgovor (izlazne podatke) u konačnom vremenu.

Algoritme ćemo uspoređivati obzirom na broj "osnovnih koraka" potrebnih za njihovo izvršavanje, te ponekad i obzirom na potreban prostor (memoriju). Pod osnovnim korakom podrazumijevamo jednu "bitnu operaciju", tj. logičku operaciju disjunkcije, konjukcije ili negacije na bitovima - nulama i jedinicama. Veličinu ulaznih podataka ćemo mjeriti brojem bitova potrebnih za njihov prikaz. Na primjer, veličina prirodnog broja N je $\lfloor \log_2 N \rfloor + 1$. Ovdje $\lfloor x \rfloor$ označava najveći cijeli broj koji je manji ili jednak x ("najveće cijelo" od x, "pod" od x). S $\lceil x \rceil$ ćemo označavati najmanji cijeli broj koji je veći ili jednak x ("strop" od x).

Često je teško, pa i nemoguće, izvesti egzaktnu formulu za broj operacija nekog algoritma. Zato proučavamo asimptotsko ponašanje broja operacija kad veličina ulaznih podataka neograničeno raste. Pritom koristimo sljedeću notaciju.

Definicija 1.1. Neka su $f, g : S \to \mathbb{R}$ dvije funkcije (S je neki podskup od $\mathbb{R}$; kod nas će najčešće biti $S = \mathbb{N}$). Tada pišemo:

(1) $f(n) = O(g(n))$ ako postoje $B, C > 0$ tako da je $|f(n)| \leq C|g(n)|$ za sve $n \in S$ takve da je $n > B$;

(2) $f(n) \sim g(n)$ ako je $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = 1$;

(3) $f(n) = o(g(n))$ ako je $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = 0$.

Kod ocjene broja operacija, razlikujemo ocjene za

- broj operacija u najlošijem slučaju (za proizvoljan ulazni podatak) - to ćemo nazivati *složenost algoritma*;
- prosječan broj operacija (prosjek za sve ulazne podatke fiksne duljine) - to ćemo nazivati *prosječna složenost algoritma*.

Definicija 1.2. *Polinomijalan algoritam* je algoritam čiji je broj operacija u najlošijem slučaju funkcija oblika $O(n^k)$, gdje je n duljina ulaznog podatka (u bitovima), a k je konstanta. Algoritme koji nisu polinomijalni, nazivamo *eksponencijalnim*.

Često za polinomijalne algoritme kažemo da su "dobri" ili "efikasni", dok za eksponencijalne algoritme kažemo da su "neefikasni". Ipak, kod praktičnih primjena (npr. u kriptografiji) treba imati u vidu da je stupanj polinoma (konstanta k) jako važan. Na primjer, iako je algoritam složenosti $O(n^{\ln\ln n})$ asimptotski sporiji od algoritma složenosti $O(n^{100})$, za praktične vrijednosti od n, prvi će algoritam biti brži. Pored toga je kod primjena u kriptografiji često je važniji prosječan broj operacija od broja operacija u najlošijem slučaju. Naime, želimo da nam kriptosustav bude zasnovan na problemu koji je težak u prosječnom slučaju (ili, još bolje, u svakom slučaju), a ne samo u nekim izoliranim slučajevima.

Definicija 1.3. *Subeksponencijalni algoritam* je algoritam čija je složenost funkcija oblika $O(e^{o(n)})$, gdje je n duljina ulaznog podatka.

Najbolji poznati algoritmi za faktorizaciju prirodnog broja N su subeksponencijalni i njihova složenost je funkcija oblika

$$L_N(v,c) = O\Big(e^{c(\ln N)^v(\ln\ln N)^{1-v}}\Big)$$

za $v = \frac{1}{2}$ ili $v = \frac{1}{3}$. Uočimo da za $v = 0$ imamo $L_N(0,c) = O((\ln N)^c)$, dok za $v = 1$ imamo $L_N(1,c) = O(N^c)$. Dakle, subeksponencijalni algoritmi koji odgovaraju vrijednostima v, $0 < v < 1$, su asimptotski sporiji od polinomijalnih, ali su brži od potpuno eksponencijalnih, tj. onih čija je složenost funkcija oblika $O(e^{n^k})$.

Dosad smo promatrali algoritme i dijelili ih ugrubo na efikasne i neefikasne. Sada bismo htjeli same probleme koje rješavaju ti algoritmi podijeliti (opet ugrubo) na lake i teške. Zbog jednostavnosti, promatrat ćemo samo *probleme odluke*, tj. probleme na koje je odgovor DA ili NE.

Definicija 1.4. *Klasa složenosti* **P** se sastoji od svih problema odluke za koje postoji polinomijalni algoritam. *Klasa složenosti* **NP** se sastoji od svih problema odluke za koje se odgovor DA može provjeriti u polinomijalnom vremenu korištenjem neke dodatne informacije, tzv. certifikata. *Klasa složenosti* **co-NP** se definira na isti način za odgovor NE.

Primjer 1.1. Neka je n prirodan broj. Promotrimo sljedeći problem odluke: "Je li broj n složen?"

Vidjet ćemo kasnije da je pitanje pripada li ovaj problem klasi **P** vrlo teško i zanimljivo pitanje. No, vrlo lako se vidi da ovaj problem pripada klasi **NP**. Zaista, certifikat je u ovom slučaju bilo koji netrivijalni djeljitelj od n.

To također pokazuje da problem odluke "Je li broj n prost?" pripada i klasi **co-NP**. $\diamondsuit$

Jasno je da vrijedi **P** $\subseteq$ **NP** i **P** $\subseteq$ **co-NP**. Opće prihvaćena slutnja je da **P** $\neq$ **NP**. To se smatra jednim od najvažijih neriješenih matematičkih problema i spada među sedam *Millenium Prize Problems*, za čije je rješavanje *Clay Mathematics Institute* raspisao nagradu od milijun dolara.

Definicija 1.5. Neka su L_1 i L_2 dva problema odluke. Kažemo da se L_1 *može u polinomijalnom vremenu reducirati na* L_2, i pišemo $L_1 \leq_P L_2$, ako postoji polinomijalni algoritam za rješavanje L_1 koji koristi kao potprogram (*oracle*) algoritam za rješavanje problema L_2, pri čemu je broj poziva tog potprograma također polinomijalan. Neformalno rečeno, ako je $L_1 \leq_P L_2$, onda L_1 nije bitno teži od L_2.

Primjer 1.2. Neka je $L_1 =$ "Postoji li za polinom $p(x)$ s cjelobrojnim koeficijentima interval na kojem $p(x)$ pada?", $L_2 =$ "Postoji li za polinom $q(x)$ s cjelobrojnim koeficijentima interval na kojem je $q(x)$ negativan?".

Tada je $L_1 \leq_P L_2$. Zaista, dovoljno je izračunati $q(x) = p'(x)$ i na $q(x)$ primijeniti potpogram za L_2. $\diamondsuit$

Primjer 1.3. Neka je $L_1 =$ "Naći netrivijalan faktor M prirodnog broja N ili zaključiti da takav faktor ne postoji.", $L_2 =$ "Ima li prirodan broj N faktor M takav da je $2 \leq M \leq k$?".

Pokazat ćemo da je $L_1 \leq_P L_2$ i $L_2 \leq_P L_1$, tj. da su ovi problemi (računski) ekvivalentni. Uočimo da je L_2 problem odluke, dok L_1 to nije.

Rješenje: Pokažimo najprije da je $L_2 \leq_P L_1$. Primijenimo algoritam za L_1 na broj N. Tako dobijemo faktor M. Potom ponovo primijenimo isti algoritam na brojeve M i N/M, itd. sve dok N ne prikažemo kao produkt prostih brojeva. Pogledamo je li najmanji prosti faktor od N manji ili jednak k i riješimo problem L_2.

Pokažimo sada da je $L_1 \leq_P L_2$. Naći ćemo faktor od N bit po bit, krenuvši od vodećeg bita, pomoću tzv. binarnog pretraživanja. Neka je n

broj bitova od N. Najprije primijenimo algoritam L_2 za $k = 2^{n-1} - 1$. Ako je odgovor NE, onda znamo da je N prost i problem je riješen. Ako je odgovor DA, onda primijenimo algoritam L_2 za $k = 2^{n-2} - 1$. Ako je odgovor NE, onda N ima faktor oblika $1 \cdot 2^{n-2} + \varepsilon_{n-3} \cdot 2^{n-3} + \varepsilon_0$, a ako je odgovor DA, onda N ima faktor oblika $0 \cdot 2^{n-2} + \varepsilon_{n-3} \cdot 2^{n-3} + \varepsilon_0$. Da bismo odredili sljedeći bit, primijenimo L_2 za $k = 2^{n-2} + 2^{n-3} - 1$ ako je odgovor bio NE, odnosno za $k = 2^{n-3} - 1$ ako je odgovor bio DA. Ako sada odgovor bude NE, onda uzimamo $\varepsilon_{n-3} = 1$, dok u protivnom uzimamo $\varepsilon_{n-3} = 0$. Nastavljajući ovaj postupak, u n poziva algoritma L_2 nalazimo netrivijalni faktor od N.

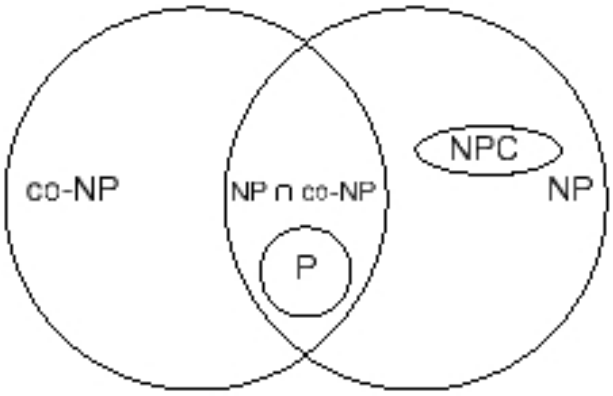
Ilustrirajmo gornji dokaz na konkretnom primjeru $N = 91$. U pozivima algoritma za L_2, pitamo se redom: "Ima li 91 faktor između 2 i 63 (DA), između 2 i 31 (DA), između 2 i 15 (DA), između 2 i 7 (DA), između 2 i 3 (NE), između 2 i 5 (NE), između 2 i 6 (NE)?" Zaključujemo da broj 91 ima faktor čiji je binarni zapis 000111, tj. da je 7 faktor od 91. ♢

Definicija 1.6. Problem odluke L je **NP**-*potpun* ako je $L \in$ **NP** i $L_1 \leq_P L$ za svaki $L_1 \in$ **NP**. Klasa svih **NP**-potpunih problema označava se s **NPC**.

Možemo reći da su **NP**-potpuni problemi najteži problemi u klasi **NP**. Jedan primjer **NP**-potpunog problema je problem ruksaka, na kojem je zasnovan Merkle-Hellmanov kriptosustav. Drugi primjer je problem trgovačkog putnika, koji treba naći najkraću rutu koja prolazi svim gradovima na njegovoj karti. Primjer **NP**-potpunog problema je također i bojenje geografske karte trima bojama.

Postojanje polinomijalnog algoritma za bilo koji od **NP**-potpunih problema povlačilo bi da vrijedi **P** = **NP**. Kao što već napomenuli, vjeruje se da ova jednakost ne vrijedi, pa je stoga i postojanje takvog polinomijalnog algoritma jako malo vjerojatno.

Hipotetski, odnos promatranih klasa izgleda ovako:



Dosad promatrani algoritmi bili su *deterministički*, što znači da za isti ulazni podatak uvijek slijede isti niz operacija. Za razliku od njih, tzv. *vjerojatnosni (randomizirani) algoritmi* prilikom izvođenja rade neke slučajne izbore.

Definicija 1.7. Kažemo da je problem odluke L rješiv u *vjerojatnosno (randomizirano) polinomijalnom vremenu*, i pišemo $L \in \mathbf{RP}$, ako postoji polinomijalni algoritam koji uključuje slučajan izbor jednog ili više cijelih brojeva, i ovisno o tom izboru, daje odgovor DA ili NE, s time da je odgovor DA sigurno točan, dok je odgovor NE točan s vjerojatnošću barem $1/2$.

Ako je $L \in \mathbf{RP}$, onda uzimajući k nezavisnih iteracija dobivamo algoritam za kojeg je odgovor NE točan s vjerojatnošću većom od $1 - 2^{-k}$.

1.2 Množenje prirodnih brojeva

Prije proučavanja algoritama iz teorije brojeva, recimo nešto o složenosti osnovnih računskih operacija s prirodnim brojevima.

Iz same definicije bitnih operacija, jasno je da se zbrajanje i oduzimanje dvaju prirodnih brojeva x i y, takvih da je $x, y \leq N$, može obaviti u $O(\ln N)$ bitnih operacija. Neka su $x = (x_n, \ldots, x_1, x_0)_b$ i $y = (y_n, \ldots, y_1, y_0)_b$ dva prirodna broja zapisana u bazi b. Tada prikaz broja

$$x + y = (w_{n+1}, w_n, \ldots, w_1, w_0)_b$$

u bazi b računamo na sljedeći način (oduzimanje je vrlo slično):

Algoritam 1 Algoritam za zbrajanje

$c = 0$
for $(0 \leq i \leq n)$ do
    if $(x_i + y_i + c < b)$ then $w_i = x_i + y_i + c$; $c = 0$
        else $w_i = x_i + y_i + c - b$; $c = 1$
$w_{n+1} = c$

Algoritmi za najjednostavnije "školsko" (ili "naivno") množenje i dijeljenje imaju složenost $O(\ln^2 N)$. Podsjetimo se tih algoritama. Neka je

$$x = (x_n, \ldots, x_1, x_0)_b, \quad y = (y_t, \ldots, y_1, y_0)_b.$$

Želimo izračunati $x \cdot y = (w_{n+t+1}, \ldots, w_0)_b$ (ako x ima $n+1$ znamenku, a y $t+1$-znamenku, onda produkt $x \cdot y$ ima najviše $n+t+2$ znamenke). Uočimo da ako su x_j i y_i znamenke u bazi b, onda $x_j \cdot y_i$ možemo zapisati kao $(uv)_b$, gdje su u i v znamenke u bazi b, s time da u može biti 0.

Algoritam 2 Algoritam za "školsko" množenje

1: for $(0 \leq i \leq n+t+1)$ do $w_i = 0$
2: for $(0 \leq i \leq t)$ do
3: $\quad c = 0$
4: $\quad$ for $(0 \leq j \leq n)$ do
5: $\quad\quad (uv)_b = w_{i+j} + x_j \cdot y_i + c$; $w_{i+j} = v$; $c = u$
6: $\quad w_{n+t+1} = u$

Neka su x i y kao gore i pretpostavimo da je $n \geq t \geq 1$. Želimo naći kvocijent $q = (q_{n-t}, \ldots, q_0)_b$ i ostatak $r = (r_t, \ldots, r_0)_b$ pri dijeljenju broja x s y, tj. brojeve q i r koji zadovoljavaju $x = qy + r$, $0 \leq r < y$.

Algoritam 3 Algoritam za dijeljenje s ostatkom

1: for $(0 \leq j \leq n-t)$ do $q_j = 0$
2: while $(x \geq yb^{n-t})$ do $q_{n-t} = q_{n-t} + 1$; $x = x - yb^{n-t}$
3: for $(n \geq i \geq t+1)$ do
4: $\quad$ if $(x_i = y_t)$ then $q_{i-t-1} = b - 1$
5: $\quad\quad$ else $q_{i-t-1} = \lfloor (x_i b + x_{i-1})/y_t \rfloor$
6: $\quad$ while $(q_{i-t-1}(y_t b + y_{t-1}) > x_i b^2 + x_{i-1} b + x_{i-2})$ do
7: $\quad\quad q_{i-t-1} = q_{i-t-1} - 1$
8: $\quad x = x - q_{i-t-1} y b^{i-t-1}$
9: $\quad$ if $(x < 0)$ then $x = x + yb^{i-t-1}$; $q_{i-t-1} = q_{i-t-1} - 1$
10: $r = x$

Za razliku od zbrajanja i oduzimanja, kod množenja i dijeljenja postoje algoritmi koji su, barem teoretski, puno efikasniji od gore navedenih "školskih" algoritama. Najbolji poznati algoritmi za množenje i dijeljenje imaju složenost

$$O(\ln N (\ln \ln N)(\ln \ln \ln N))$$

(Schönhage-Strassenov algoritam iz 1971. godine, Fürerov algoritam iz 2007.). Uočimo da je

$$\ln N (\ln \ln N)(\ln \ln \ln N) = O((\ln N)^{1+\varepsilon}) \quad \text{za svaki } \varepsilon > 0.$$

Dakle, možda pomalo i iznenađujuće, množenje je tek neznatno složenije od zbrajanja. No, to je ipak teoretski zaključak koji ignorira ogromnu konstantu koja se krije iza "velikog O". Ti (teoretski) najbolji poznati algoritmi koriste *brzu Fourierovu transformaciju* (FFT). Njihova primjena je od praktične važnosti tek za brojeve od nekoliko tisuća znamenaka. No, postoje algoritmi koji su bolji od "školskog", a koji imaju praktičnu važnost za brojeve od stotinjak znamenaka, kakvi se danas uglavnom rabe u kriptografiji.

Opisat ćemo *Karacubinu metodu* (iz 1962. godine) za množenje prirodnih brojeva.

Neka su $x = (x_{2n-1}, \ldots, x_1, x_0)_2$ i $y = (y_{2n-1}, \ldots, y_1, y_0)_2$ dva $2n$-bitna prirodna broja. Zapišimo ih u obliku

$$x = 2^n u_1 + u_0, \qquad y = 2^n v_1 + v_0$$

(u_1 i v_1 su "lijeve polovice", a u_0 i v_0 "desne polovice" od x, odnosno y). Sada je

$$x \cdot y = 2^{2n} u_1 v_1 + 2^n (u_1 v_0 + u_0 v_1) + u_0 v_0.$$

Zasad još nemamo nikakvu prednost od ovakvog zapisa – umjesto jednog produkta $2n$-bitnih brojeva, trebamo izračunati četiri produkta n-bitnih brojeva. Međutim, i to je glavna poanta Karacubine metode, u stvari je dovoljno izračunati samo 3 produkta, jer vrijedi

$$u_1 v_0 + u_0 v_1 = u_1 v_1 + u_0 v_0 - (u_1 - u_0)(v_1 - v_0).$$

Dakle,

$$x \cdot y = (2^{2n} + 2^n) u_1 v_1 + 2^n (u_1 - u_0)(v_0 - v_1) + (2^n + 1) u_0 v_0.$$

Ovaj proces sada možemo nastaviti rekurzivno, tj. tri nova produkta možemo računati na isti način, tako da svaki faktor rastavimo na dva dijela podjednake veličine.

Postavlja se pitanje koliko je ovakav algoritam efikasniji od "naivnog" množenja. Označimo s $T(n)$ broj bitnih operacija potrebnih za množenje dvaju n-bitnih brojeva Karacubinom metodom. Tada vrijedi

$$T(2n) \leq 3T(n) + cn \tag{1.1}$$

za neku konstantu c. Iz (1.1) slijedi da, uz dovoljno veliku konstantu C, vrijedi

$$T(2^k) \leq C(3^k - 2^k), \tag{1.2}$$

za $k \geq 1$. Zaista, neka je konstanta $C \geq c$ odabrana tako da (1.2) vrijedi za $k = 1$, te pretpostavimo da (1.2) vrijedi za neki $k \in \mathbb{N}$. Tada imamo:

$$T(2^{k+1}) \leq 3T(2^k) + c \cdot 2^k \leq C \cdot 3^{k+1} - 3C \cdot 2^k + c \cdot 2^k \leq C(3^{k+1} - 2^{k+1}),$$

pa tvrdnja vrijedi po principu matematičke indukcije. Sada je

$$T(n) \leq T(2^{\lceil \log_2 n \rceil}) \leq C\left(3^{\lceil \log_2 n \rceil} - 2^{\lceil \log_2 n \rceil}\right) < 3C \cdot 3^{\log_2 n} = 3C n^{\log_2 3}.$$

Stoga je složenost Karacubinog algoritma $O(n^{\log_2 3})$, tj. brojevi $x, y \leq N$ se mogu pomnožiti uz $O((\ln N)^{\log_2 3})$ bitnih operacija. U usporedbi s "naivnim" množenjem, umjesto $(\ln N)^2$ imamo približno $(\ln N)^{1.585}$ operacija.

1.3 Modularno množenje i potenciranje

U većini kriptosustava s javnim ključem, šifriranje i dešifriranje je opisano pomoću operacija u prstenu $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$, za neki veliki prirodni broj m. Zbrajanje u $\mathbb{Z}_m$ je vrlo jednostavno. Naime, za $x, y \in \mathbb{Z}_m$, shvatimo x i y kao nenegativne cijele brojeve manje od m, i tada je

$$x +_m y = \begin{cases} x + y & \text{ako je } x + y < m, \\ x + y - m & \text{ako je } x + y \geq m. \end{cases}$$

S druge strane, množenje u $\mathbb{Z}_m$ nije tako jednostavno. Posebno, ono je bitno kompliciranije od običnog množenja prirodnih brojeva, zato što pored množenja uključuje i (netrivijalnu) modularnu redukciju.

Direktna metoda za računanje produkta $x \cdot_m y$ u $\mathbb{Z}_m$ je da, pomoću algoritama iz prethodnog poglavlja, izračunamo najprije $x \cdot y$, a potom izračunamo ostatak r pri djeljenju $x \cdot y$ s m. Tada je $x \cdot_m y = r$.

Postoji nekoliko poboljšanja ove metode. Opisat ćemo *Montgomeryjevu redukciju* (iz 1985. godine) čija je glavna ideja izbjegavanje klasične modularne redukcije, tj. dijeljenja.

Neka su m, R i T prirodni brojevi takvi da je $R > m$, $\mathrm{nzd}(m, R) = 1$ i $0 \leq T < mR$. Ako je m prikazan u bazi b i ima u tom prikazu n znamenaka, onda se obično uzima $R = b^n$. Pokazat ćemo da se $TR^{-1} \bmod m$ može izračunati bez klasičnog dijeljenja. Preciznije, dijeljenje s m zamjenjuje se puno jednostavnijim dijeljenjem s R, koje je zapravo (u slučaju $R = b^n$) jednostavni pomak za n znamenaka.

Lema 1.1. *Neka je $m' = -m^{-1} \bmod R$, te $U = Tm' \bmod R$. Tada je $V = (T + Um)/R$ cijeli broj i $V \equiv TR^{-1} \pmod m$. Nadalje, $TR^{-1} \bmod m = V$ ili $TR^{-1} \bmod m = V - m$.*

Dokaz: Iz definicije brojeva m' i U slijedi da postoje $k, l \in \mathbb{Z}_m$ takvi da je $mm' = -1 + kR$, $U = Tm' + lR$. Sada je

$$\frac{T + Um}{R} = \frac{T + Tmm' + lRm}{R} = \frac{T + T(-1 + kR) + lRm}{R} = kT + lm \in \mathbb{Z}.$$

Očito je $V \equiv (T + Um)R^{-1} \equiv TR^{-1} \pmod m$. Konačno, iz $T < mR$ i $U < R$ slijedi $0 \leq V < (mR + mR)/R = 2m$, pa iz $V \equiv TR^{-1} \pmod m$ slijedi $V - (TR^{-1} \bmod m) = 0$ ili m. □

Izraz $TR^{-1} \bmod m$ naziva se *Montgomeryjeva redukcija* od T modulo m u odnosu na R, dok se $xR \bmod m$ naziva *Montgomeryjev prikaz* od x. *Montgomeryjev produkt* brojeva x i y je broj $\mathrm{Mont}(x, y) = xyR^{-1} \bmod m$. Ovo je dobro definirano, jer je $xy < m^2 < mR$. Vrijedi:

$$\mathrm{Mont}(xR \bmod m, yR \bmod m) = (xR)(yR)R^{-1} = xyR \bmod m.$$

Dakle, za brojeve u Montgomeryjevom prikazu modularno se množenje može provesti bez modularne redukcije modulo m. Naravno, modularnu redukciju trebamo da bismo uopće dobili Montgomeryjev prikaz. No, ukoliko više puta koristimo jedan te isti broj, kao što je slučaj kod potenciranja, Montgomeryjeva metoda je znatno efikasnija od obične modularne redukcije.

Jedna od mogućnosti za pojednostavljenje modularne redukcije je izbor modula specijalnog oblika. Tu se ponovo koristi činjenica da je dijeljenje s brojevima oblika b^n vrlo jednostavno. Zato se (ako je to u konkretnoj situaciji moguće) biraju moduli oblika $m = b^n - a$, gdje je a mali prirodni broj. Tada se $x \bmod m$ može izračunati pomoću ovog algoritma:

Algoritam 4 Redukcija modulo $b^n - a$

1: $q_0 = \lfloor x/b^n \rfloor$
2: $r_0 = x - q_0 b^n$
3: $r = r_0$; $i = 0$
4: while ($q_i > 0$) do
5: $\quad q_{i+1} = \lfloor q_i a / b^n \rfloor$; $r_{i+1} = q_i a - q_{i+1} b^n$;
6: $\quad i = i + 1$; $r = r + r_i$
7: while ($r \geq m$) do $r = r - m$

U najpopularnijim kriptosustavima s javnim ključem (RSA, ElGamal) šifriranje se provodi pomoću modularnog potenciranja, tj. funkcije oblika $x^n \bmod m$. Štoviše, činjenica da se takva funkcija može puno efikasnije izračunati od njezinog inverza, predstavlja osnovu za korištenje problema diskretnog logaritma u kriptografiji.

Modularno potenciranje predstavlja specijalni slučaj potenciranja u Abelovim grupama. Stoga ćemo reći nešto o općim metodama za računanje potencije x^n u Abelovoj grupi G. Naravno, trivijalni algoritam u kojem bismo x^n izračunali kao $x \cdot x \cdot \ldots \cdot x$ pomoću $n - 1$ množenja vrlo je neefikasan.

Najjednostavnija i najstarija među efikasnim metodama je binarna metoda ili metoda uzastopnog kvadriranja (još se naziva i metoda "kvadriraj i množi" ili "binarne ljestve") koja koristi binarni zapis broja n. Recimo da želimo izračunati x^{13}. Binarni zapis od 13 je 1101. Sada x^{13} možemo izračunati kao

$$x^{13} = x \cdot (x^2)^2 \cdot ((x^2)^2)^2.$$

Mogli bismo reći da smo binarni zapis čitali s desna na lijevo. Ako isti zapis pročitamo s lijeva na desno, onda imamo

$$x^{13} = x \cdot ((x \cdot x^2)^2)^2.$$

Dakle, imamo sljedeća dva algoritma za računanje $z = x^n$, gdje je

$$n = (n_d, \ldots, n_0)_2.$$

Algoritam 5 Binarna metoda (s desna na lijevo)

1: $z = 1$; $y = x$
2: for $(0 \leq i \leq d-1)$ do
3: if $(n_i = 1)$ then $z = z \cdot y$
4: $y = y^2$
5: $z = z \cdot y$

Algoritam 6 Binarna metoda (s lijeva na desno)

1: $z = x$
2: for $(d-1 \geq i \geq 0)$ do
3: $z = z^2$
4: if $(n_i = 1)$ then $z = x \cdot z$

Obje varijante binarne metode imaju isti broj operacija: d kvadriranja, te množenja onoliko koliko ima jedinica u binarnom zapisu od n (što je $\leq d+1$, a u prosječnom slučaju je oko $d/2$). Dakle, u slučaju modularnog potenciranja složenost je $O(\ln n \ln^2 m)$ (ako za složenost množenja i dijeljenja brojeva manjih od m uzmemo da je $O(\ln^2 m)$, što ćemo i ubuduće raditi).

Prednost druge varijante (s lijeva na desno) je u tome da se u koraku $z = z \cdot x$ množi uvijek s istim brojem x. Često je u primjenama taj x mali (čak jednak 2), pa je u tom slučaju ova operacija vrlo brza. Ova prednost je tim veća, što je veći broj jedinica u binarnom zapisu od n.

Jedno poboljšanje binarne metode sastoji se u promatranju grupa binarnih znamenaka (tzv. prozora). Na primjer, ako gledamo grupe od po dvije znamenke (tj. radimo u bazi 4), onda ćemo prethodno izračunati x, x^2, x^3, pa potom npr. x^{79} možemo izračunati kao

$$x^{79} = x^3 \cdot ((x^4)^4 \cdot x^3)^4.$$

U primjenama u kriptografiji, često su ili baza x, ili eksponent n, fiksni. U tim situacijama moguća su dodatna poboljšanja.

U slučaju fiksne baze x, unaprijed izračunamo $x_i = x^{2^i}$, te x^n računamo kao $x^n = \prod_{i=0}^{d} x_i^{n_i}$. Ako umjesto baze 2 koristimo bazu b, onda se unaprijed izračunaju vrijednosti $x_{ij} = x^{j \cdot b^i}$, $1 \leq j \leq b-1$, $1 \leq i \leq d$. Sada se x^n računa kao $x^n = \prod_{i=0}^{d} x_{in_i}$, što znači uz $d \sim (\ln n)/(\ln b)$ množenja.

U slučaju fiksnog eksponenta n, jedna ideja za moguće poboljšanje je korištenje "lanca zbrojeva". To je niz $u_0, u_1, \ldots, u_s$ s pridruženim nizom $w_1, \ldots, w_s$ parova $w_i = (i_1, i_2)$, sa svojstvom da je

$$u_0 = 1,\ u_s = n,\ u_i = u_{i_1} + u_{i_2},\ \text{za } 1 \leq i \leq s.$$

Npr. za $n = 15$, jedan lanac zbrojeva je $u_0 = 1$, $u_1 = 2$, $u_2 = 3$, $u_3 = 6$, $u_4 = 12$, $u_5 = 15$. Ako je poznat lanac zbrojeva za n duljine s, onda se x^n može izračunati uz s množenja: $x_0 = x$, $x_i = x_{i_1} \cdot x_{i_2}$ za $i = 1, \ldots, s$, pa je $x_s = x^n$.

Sve što smo dosad rekli o potenciranju, odnosi se na potenciranje u bilo kakvoj Abelovoj grupi. Kao što smo već prije spomenuli, jedan od načina za dodatno poboljšanje modularnog potenciranja jest korištenje Montgomeryjeve redukcije. Montgomeryjevu redukciju možemo kombinirati s bilo kojom od metoda za potenciranje koje smo ranije naveli. Prikažimo to na primjeru binarne metode (s lijeva na desno).

Algoritam 7 Montgomeryjevo potenciranje

1: $y = \text{Mont}(x, R^2 \bmod m)$
2: $z = R \bmod m$
3: for $(d \geq i \geq 0)$ do
4: $\quad z = \text{Mont}(z, z)$
5: $\quad$ if $(n_i = 1)$ then $z = \text{Mont}(z, y)$
6: $z = \text{Mont}(z, 1)$

Zaista, $y = xR \bmod m$ pa se u petlji izračuna $x^n R \bmod m$, dok je

$$\text{Mont}(x^n R \bmod m, 1) = x^n \bmod m.$$

1.4 Euklidov algoritam

Razmotrit ćemo sada problem nalaženja najvećeg zajedničkog djelitelja dva cijela broja a i b, u oznaci $\text{nzd}(a, b)$ (ili $\gcd(a, b)$ ili (a, b)). Jedna, naivna, mogućnost jest faktorizacija brojeva a i b na proste faktore. Ako je $a = \prod_p p^{\alpha(p)}$, $b = \prod_p p^{\beta(p)}$, onda je

$$\text{nzd}(a, b) = \prod_p p^{\min(\alpha(p), \beta(p))}.$$

No, kako je teškoća faktorizacije velikih prirodnih brojeva jedna od najvažijih činjenica u algoritamskoj teoriji brojeva, jasno je da trebamo bolju metodu za računanje najvećeg zajedničkog djelitelja.

Problem, koji je usko povezan s ovim, jest problem računanja modularnog inverza $x^{-1} \bmod m$, tj. broja $y \in \{1, \ldots, m-1\}$ takvog da je $xy \equiv 1 \pmod{m}$. (Taj problem se pojavljuje primjerice kod RSA kriptosustava, gdje su eksponenti d i e povezani relacijom $de \equiv 1 \pmod{\varphi(n)}$.) Ova dva problema povezuje sljedeći teorem.

Teorem 1.1. *Postoje cijeli brojevi x, y takvi da je $ax + by = \text{nzd}(a, b)$.*

Dokaz: Provjerimo najprije da skup $S = \{ax + by : x, y \in \mathbb{Z}\}$ ima barem jedan pozitivni element. Bez smanjenja općenitosti možemo pretpostaviti da je $a \neq 0$. Ako je $a > 0$, onda je $a = a \cdot 1 + b \cdot 0$ pozitivan, a ako je $a < 0$, onda je $|a| = a \cdot (-1) + b \cdot 0$ pozitivan.

Neka je $g = \mathrm{nzd}(a, b)$ te neka je l najmanji pozitivni član skupa $S = \{ax + by : x, y \in \mathbb{Z}\}$. To znači da postoje cijeli brojevi x_0 i y_0 takvi da je $l = ax_0 + by_0$.

Pokažimo da $l \mid a$ i $l \mid b$. Pretpostavimo da npr. $l \nmid a$. Tada postoje cijeli brojevi q i r takvi da je $a = lq + r$ i $0 < r < l$. Sada je

$$r = a - lq = a - q(ax_0 + by_0) = a(1 - qx_0) + b(-qy_0) \in S,$$

što je u suprotnosti s minimalnosti od l. Dakle, $l \mid a$, a na isti način se pokazuje da $l \mid b$. To znači da je $l \leq g$.

Budući da je $g = \mathrm{nzd}(a, b)$, to postoje $\alpha, \beta \in \mathbb{Z}$ takvi da je $a = g\alpha$, $b = g\beta$, pa je $l = ax_0 + by_0 = g(\alpha x_0 + \beta y_0)$. Odavde slijedi da je $g \leq l$, pa smo dokazali da je $g = l$. □

Sada je jasna veza s traženjem inverza. Ako su a i b relativno prosti cijeli brojevi, onda postoje cijeli brojevi x, y takvi da je

$$ax + by = 1$$

i pritom je $x \bmod b = a^{-1} \bmod b$, dok je $y \bmod a = b^{-1} \bmod a$.

Euklidov algoritam je jedan od najstarijih, ali ujedno i jedan od najvažnijih algoritama u teoriji brojeva. Zasnovan je na činjenici da je $\mathrm{nzd}(a, b) = \mathrm{nzd}(b, a \bmod b)$. Kako je $\mathrm{nzd}(a, b) = \mathrm{nzd}(|a|, |b|)$, možemo pretpostaviti da $a > b \geq 0$.

Algoritam 8 Euklidov algoritam

```
1: while (b > 0) do
2:     (a, b) = (b, a mod b)
3: return a
```

Da bismo analizirali složenost ovog algoritma, raspišimo ga po koracima:

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= q_2 r_1 + r_2 \\ &\vdots \\ r_{n-3} &= q_{n-1} r_{n-2} + r_{n-1} \\ r_{n-2} &= q_n r_{n-1} \end{aligned}$$

Ocijenimo broj koraka, tj. broj n, u najlošijem slučaju. Neka su a i b prirodni brojevi takvi da Euklidov algoritam za (a, b) treba n koraka. Tada je $a \geq F_{n+2}$, $b \geq F_{n+1}$, gdje F_k označava k-ti Fibonaccijev broj (podsjetimo se

definicije Fibonaccijevih brojeva: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ za $n \geq 2$). Dokažimo to matematičkom indukcijom po n. Za $n = 1$ je $b \geq 1 = F_2$, $a \geq 2 = F_3$. Pretpostavimo da tvrdnja vrijedi za $n - 1$ koraka. Za brojeve b i r_1 Euklidov algoritam treba $n - 1$ koraka. Stoga je po pretpostavci indukcije $b \geq F_{n+1}$, $r_1 \geq F_n$. No, tada je $a = q_1 b + r_1 \geq b + r_1 \geq F_{n+2}$.

Budući da je $\lfloor F_{k+1}/F_k \rfloor = 1$ i $F_{k+1} \bmod F_k = F_{k-1}$, svi su kvocijenti u Euklidovu algoritmu za F_{n+2} i F_{n+1} jednaki 1 i algoritam treba točno n koraka. Odatle i iz Binetove formule za Fibonaccijeve brojeve

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

slijedi

Teorem 1.2. *Neka su $a, b \leq N$. Tada je broj koraka u Euklidovu algoritmu za računanje (a, b) manji ili jednak*

$$\left\lceil \frac{\ln(\sqrt{5}N)}{\ln((1+\sqrt{5})/2)} \right\rceil - 2 \approx 2.078 \ln N + 1.672\,.$$

Može se pokazati da je prosječan broj koraka u Euklidovu algoritmu za brojeve a i b iz skupa $\{1, \ldots, N\}$ približno jednak

$$\frac{12 \ln 2}{\pi^2} \ln N + 0.14 \approx 0.843 \ln N + 0.14.$$

Pojednostavljeno rečeno, broj koraka je $O(\ln N)$. Kako svaki korak Euklidova algoritma zahtijeva jedno dijeljenje brojeva $\leq N$, dobivamo da je složenost Euklidova algoritma $O(\ln^3 N)$. Ovu ocjenu možemo poboljšati ako uočimo da u svakom koraku radimo sa sve manjim brojevima. Tako da je broj operacija

$$\begin{aligned} O(\ln a \cdot \ln q_1 \,+\, & \ln b \cdot \ln q_2 + \ln r_1 \cdot \ln q_3 + \cdots + \ln r_{n-2} \cdot \ln q_n) \\ &= O(\ln N \cdot (\ln q_1 + \ln q_2 + \cdots + \ln q_n)) \\ &= O(\ln N \cdot \ln(q_1 q_2 \cdots q_n)) = O(\ln^2 N) \end{aligned}$$

(posljednju jednakost dobijemo množeći sve lijeve i sve desne strane u jednakostima u Euklidovu algoritmu).

Euklidov algoritam se može iskoristiti i za nalaženje cijelih brojeva x, y takvih da je $ax + by = \mathrm{nzd}(a, b)$. Dakle, možemo ga koristiti za rješavanje linearnih diofantskih jednadžbi. Kao što smo već vidjeli, u slučaju da je $\mathrm{nzd}(a, b) = 1$ na taj se način može dobiti modularni inverz. Ta verzija Euklidova algoritma se obično naziva *prošireni Euklidov algoritam*.

Algoritam 9 Prošireni Euklidov algoritam

1: $(x, y, g, u, v, w) = (1, 0, a, 0, 1, b)$
2: while $(w > 0)$ do
3: $\quad q = \lfloor g/w \rfloor$
4: $\quad (x, y, g, u, v, w) = (u, v, w, x - qu, y - qv, g - qw)$
5: return (x, y, g)

Prikazat ćemo još jedan algoritam za računanje najvećeg zajedničkog djelitelja, *binarni gcd algoritam*. Kod njega se umjesto dijeljenja koriste samo operacije oduzimanja i pomaka (dijeljenja s 2). Kao rezultat dobivamo algoritam koji ima veći broj koraka, ali su ti koraci jednostavniji. U samom algoritmu susrećemo dvije ideje. Prva je da iako smo u početku rekli da je faktorizacija brojeva težak problem, izdvajanje potencija broja 2 je vrlo jednostavno. Druga ideja je zamjena dijeljenja oduzimanjem, a povezana je s činjenicom da u originalnom Euklidovu algoritmu vrlo često umjesto dijeljenja zapravo imamo oduzimanje, jer je pripadni kvocijent jednak 1. Može se pokazati da vjerojatnost da je Euklidov kvocijent jednak q iznosi

$$P(q) = \log_2 \Big(1 + \frac{1}{(q+1)^2 - 1}\Big).$$

Tako je $P(1) \approx 0.415$, $P(2) \approx 0.170$, $P(3) \approx 0.093$, ... Dakle, u 41.5% slučajeva kvocijent je jednak 1. Ovi rezultati su u uskoj vezi s ranije navedenim prosječnim brojem koraka u Euklidovu algoritmu.

Označimo s $v_2(k)$ najveću potenciju broja 2 koja dijeli k.

Algoritam 10 Binarni gcd algoritam

1: $\beta = \min\{v_2(a), v_2(b)\}$
2: $a = a/2^{v_2(a)}$; $b = b/2^{v_2(b)}$
3: while $(a \neq b)$ do
4: $\quad (a, b) = (\min\{a, b\}, |b - a|/2^{v_2(b-a)})$
5: return $2^\beta a$

1.5 Kineski teorem o ostatcima

Kineski teorem o ostatcima (engl. Chinese Remainder Theorem - CRT) govori o rješenju sustava linearnih kongruencija. Ime mu se vezuje uz kineskog matematičara iz trećeg stoljeća Sun Tzua. Smatra se da je teorem već tada korišten u kineskoj vojsci za prebrojavanje vojnika. Pretpostavimo da treba prebrojiti grupu od približno 1000 vojnika. Vojnici se rasporede npr. u 3, 4, 5 i 7 kolona, te se zabilježi koliko je vojnika ostalo kao "višak" u zadnjem redu. Tako dobivamo sustav od četiri kongruencije s modulima 3, 4, 5 i 7, a

taj sustav, prema sljedećem teoremu, ima jedinstveno rješenje između 800 i 1200.

Teorem 1.3 (Kineski teorem o ostatcima)**.** *Neka su $m_1, m_2, \ldots, m_k$ u parovima relativno prosti prirodni brojevi, tj.* $\mathrm{nzd}(m_i, m_j) = 1$ *za $i \neq j$. Tada za proizvoljne cijele brojeve $x_1, \ldots, x_k$ postoji cijeli broj x takav da vrijedi*

$$x \equiv x_i \pmod{m_i}, \quad i = 1, \ldots, k.$$

Broj x je jedinstven modulo $M = m_1 \cdots m_k$.

Broj x iz teorema možemo naći na sljedeći način. Neka je $M_i = \dfrac{M}{m_i}$. Kako je $\mathrm{nzd}(M_i, m_i) = 1$, slijedi da pomoću Euklidova algoritma možemo naći a_i takav da je $a_i M_i \equiv 1 \pmod{m_i}$. Sada

$$x = \sum_{i=1}^{k} a_i M_i x_i \bmod m$$

zadovoljava uvjete teorema.

Složenost algoritma kojeg smo upravo opisali je $O(\ln^2 M)$.

Kineski teorem o ostatcima ima brojne primjene. Jedan od razloga jest taj da on omogućava zamjenu računanja po jednom velikom modulu s nekoliko neovisnih računanja po puno manjim modulima, što je jako dobra osnova za "paralelizaciju" računanja.

U primjenama su često m_i-ovi fiksni, dok x_i-ovi variraju. U takvoj je situaciji dobro onaj dio algoritma, koji ne ovisi o x_i-ovima, izračunati unaprijed. Sljedeći algoritam koristi tu ideju, a također vodi računa o racionalnom korištenju brojeva M_i koji mogu biti jako veliki (za razliku od brojeva a_i koji su sigurno manji od m_i).

Algoritam 11 Garnerov algoritam za CRT

1: for $(1 \leq i \leq k-1)$ do
2: $\quad \mu_i = \prod_{j=1}^{i} m_j$
3: $\quad c_i = \mu_i^{-1} \bmod m_{i+1}$
4: $M = \mu_{k-1} m_k$

5: $x = x_1$
6: for $(1 \leq i \leq k-1)$ do
7: $\quad y = ((x_{i+1} - x) c_i) \bmod m_{i+1}$
8: $\quad x = x + y\mu_i$
9: $x = x \bmod M$

Ovaj algoritam "rješava" module jedan po jedan. Tako da nakon i-tog koraka u petlji, x zadovoljava $x \equiv x_j \pmod{m_j}$ za $j = 1, 2, \ldots, i+1$.

Ukoliko treba riješiti neki jednokratni problem, onda naravno nema koristi od prethodnog računanja s m_i-ovima. U takvoj se situaciji preporuča induktivna uporaba originalnog algoritma za sustav od dvije kongruencije. Naime, ako želimo riješiti sustav

$$x \equiv x_1 \pmod{m_1}, \quad x \equiv x_2 \pmod{m_2},$$

onda jednom primjenom Euklidova algoritma dobivamo oba željena inverza iz $um_1 + vm_2 = 1$. Tada je $x = um_1x_2 + vm_2x_1 \bmod m_1m_2$ rješenje sustava.

Algoritam 12 Induktivni algoritam za CRT

1: $m = m_1$; $x = x_1$
2: for $(2 \leq i \leq k)$ do
3: nađi u, v takve da je $um + vm_i = 1$
4: $x = umx_i + vm_ix$
5: $m = mm_i$
6: $x = x \bmod m$

1.6 Verižni razlomci

U ovom poglavlju vidjet ćemo još jednu primjenu Euklidova algoritma. Iz prvog koraka u Euklidovu algoritmu imamo

$$\frac{a}{b} = q_1 + \frac{1}{b/r_1}.$$

Drugi korak nam daje

$$\frac{a}{b} = q_1 + \frac{1}{q_2 + \frac{1}{r_1/r_2}}.$$

Na kraju dobivamo prikaz racionalnog broja a/b u obliku

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \ddots + \cfrac{1}{q_n}}}.$$

Ovaj se prikaz naziva *razvoj broja $\frac{a}{b}$ u jednostavni verižni razlomak*. Općenito, za $\alpha \in \mathbb{R}$ se prikaz broja α u obliku

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots}},$$

gdje je $a_0 \in \mathbb{Z}$, te $a_1, a_2, \ldots \in \mathbb{N}$, naziva *razvoj broja α u jednostavni verižni (ili neprekidni) razlomak*. Verižni razlomak kraće zapisujemo kao $[a_0; a_1, a_2, \ldots]$. Brojevi $a_0, a_1, a_2, \ldots$ nazivaju se *parcijalni kvocijenti*, a definiraju se na sljedeći način:

$$a_0 = \lfloor \alpha \rfloor, \quad \alpha = a_0 + \frac{1}{\alpha_1}, \quad a_1 = \lfloor \alpha_1 \rfloor, \quad \alpha_1 = a_1 + \frac{1}{\alpha_2}, \quad a_2 = \lfloor \alpha_2 \rfloor, \ldots .$$

Postupak se nastavlja sve dok je $a_k \neq \alpha_k$. Razvoj u jednostavni verižni razlomak broja α je konačan ako i samo ako je α racionalan broj.

Kao što smo vidjeli gore, razvoj u verižni razlomak racionalnog broja može se odrediti pomoću Euklidova algoritma. Kod iracionalnog broja, najčešće znamo samo njegovu racionalnu aproksimaciju (obično binarni ili decimalni zapis s fiksnim brojem točnih znamenaka). To znači da će i u njegovom razvoju biti točni samo neki početni a_i-ovi.

Primjer 1.4. Izračunajmo razvoj u jednostavni verižni razlomak broja $\sqrt{101}$, ali tako da u svim računima koristimo samo 10 najznačajnijih decimalnih znamenaka. Dobije se

$$\sqrt{101} \approx 10.04987562 = [10; 20, 20, 20, 8, 6, 1, \ldots].$$

Postavlja se pitanje koji su od ovih a_i-ova stvarno točni. Kao što ćemo uskoro vidjeti, točan razvoj je

$$\sqrt{101} = [10; 20, 20, 20, 20, 20, 20, \ldots]. \qquad \diamondsuit$$

Željeli bismo da nam algoritam za računanje verižnog razlomka kaže kada točno treba stati. Neka je dan $\alpha \in \mathbb{R}$, te racionalni brojevi a/b i a'/b' takvi da je

$$\frac{a}{b} \leq \alpha \leq \frac{a'}{b'}\,.$$

Sljedeći algoritam računa razvoj od α i staje točno onda kada više nije moguće odrediti sljedeći a_i iz a/b i a'/b' (pripadni a_i-ovi u razvojima od a/b i a'/b' se ne podudaraju), te također daje gornju i donju ogradu za taj a_i.

Algoritam 13 Razvoj u verižni razlomak

$i = 0$
$q = \lfloor a/b \rfloor$; $r = a - bq$; $r' = a' - b'q$
while ($0 \leq r' < b'$ and $b \neq 0$) do
    $a_i = q$
    $i = i + 1$
    $a = b$; $b = r$; $a' = b'$; $b' = r'$
    $q = \lfloor a/b \rfloor$; $r = a - bq$; $r' = a' - b'q$
if ($b = 0$ and $b' = 0$) then return $[a_0; a_1, \ldots, a_i]$
if ($b \neq 0$ and $b' = 0$) then return $[a_0; a_1, \ldots, a_{i-i}]$, $a_i \geq q$
$q' = \lfloor a'/b' \rfloor$
if ($b = 0$ and $b' \neq 0$) then return $[a_0; a_1, \ldots, a_{i-i}]$, $a_i \geq q'$
if ($bb' \neq 0$) then return $[a_0; a_1, \ldots, a_{i-i}]$, $\min\{q, q'\} \leq a_i \leq \max\{q, q'\}$

U slučaju kada je α kvadratna iracionalnost, tj. iracionalan broj koji je rješenje neke kvadratne jednadžbe s racionalnim koeficijentima, tada je njegov razvoj u jednostavni verižni razlomak periodičan. Razvoj se može dobiti sljedećim algoritmom. Prikažemo α u obliku

$$\alpha = \alpha_0 = \frac{s_0 + \sqrt{d}}{t_0},$$

gdje su $d, s_0, t_0 \in \mathbb{Z}$, $t_0 \neq 0$, d nije potpun kvadrat i $t_0 | (d - s_0^2)$. Zadnji uvjet se uvijek može zadovoljiti množenjem brojnika i nazivnika s prikladnim cijelim brojem. Sada parcijalne kvocijente a_i, $i = 0, 1, 2, \ldots$ računamo rekurzivno na sljedeći način:

$$a_i = \lfloor \alpha_i \rfloor, \quad s_{i+1} = a_i t_i - s_i, \quad t_{i+1} = \frac{d - s_{i+1}^2}{t_i}, \quad \alpha_{i+1} = \frac{s_{i+1} + \sqrt{d}}{t_{i+1}}.$$

Uočimo da iako je α iracionalan broj, ovaj algoritam radi samo s cijelim brojevima, jer iz svojstava funkcije najveće cijelo, uz pretpostavku da je $t_i > 0$, slijedi da se a_i zapravo može izračunati pomoću

$$a_i = \left\lfloor \frac{s_{i+1} + a_0}{t_{i+1}} \right\rfloor.$$

Može se pokazati da su s_i-ovi i t_i-ovi ograničeni, pa stoga mogu poprimiti samo konačno mnogo vrijednosti. To znači da postoje indeksi j, k, $j < k$, takvi da je $s_j = s_k$ i $t_j = t_k$. No, tada je $\alpha_j = \alpha_k$, što znači da je

$$\alpha = [a_0; a_1, \ldots, a_{j-1}, \overline{a_j, a_{j+1}, \ldots, a_{k-1}}],$$

gdje povlaka označava blok koji se periodički ponavlja.

U slučaju, koji je najvažniji za primjene, kada je $\alpha = \sqrt{d}$ može se i preciznije reći kako izgleda razvoj u verižni razlomak. Naime, vrijedi

$$\sqrt{d} = [a_0; \overline{a_1, a_2, \ldots, a_{r-1}, 2a_0}],$$

gdje je $a_0 = \lfloor \sqrt{d} \rfloor$, a $a_1, \ldots, a_{r-1}$ su palindromni, tj. vrijedi $a_i = a_{r-i}$ za $i = 1, 2, \ldots, r-1$. Na primjer,

$$\sqrt{7} = [2; \overline{1, 1, 1, 4}], \quad \sqrt{101} = [10; \overline{20}], \quad \sqrt{1037} = [32; \overline{4, 1, 15, 3, 3, 15, 1, 4, 64}].$$

Može se pokazati da za duljinu perioda r vrijedi da je $r = O(\sqrt{d} \log d)$.

Racionalne brojeve

$$\frac{p_k}{q_k} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots + \cfrac{1}{a_k}}} = [a_0; a_1, \ldots, a_k]$$

nazivamo *konvergente verižnog razlomka*. Brojnici i nazivnici konvergenti zadovoljavaju sljedeće rekurzije:

$$p_0 = a_0, \quad p_1 = a_0 a_1 + 1, \quad p_{k+2} = a_{k+2} p_{k+1} + p_k,$$
$$q_0 = 1, \quad q_1 = a_1, \quad q_{k+2} = a_{k+2} q_{k+1} + q_k.$$

Uz dogovor da je $p_{-2} = 0$, $p_{-1} = 1$, $q_{-2} = 1$, $q_{-1} = 0$, rekurzivne relacije vrijede i za $k = -1$, $k = -2$. Indukcijom se lako dokazuje sljedeća važna relacija:

$$q_k p_{k-1} - p_k q_{k-1} = (-1)^k. \tag{1.3}$$

Relacija (1.3) povlači da je $\frac{p_{2k}}{q_{2k}} \leq \alpha$ i $\alpha \leq \frac{p_{2k+1}}{q_{2k+1}}$ za svaki k. Nadalje, ako je α iracionalan, onda je $\lim_{k\to\infty} \frac{p_k}{q_k} = \alpha$.

Racionalni brojevi $\frac{p_k}{q_k}$ jako dobro aproksimiraju α. Preciznije,

$$\left| \alpha - \frac{p_k}{q_k} \right| < \frac{1}{q_k^2}.$$

Vrijedi i svojevrsni obrat ove tvrdnje (Legendreov teorem): ako je $\frac{p}{q}$ racionalan broj koji zadovoljava nejednakost

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

onda je $\frac{p}{q} = \frac{p_k}{q_k}$ za neki k. Ova je činjenica osnova za Wienerov napad na RSA kriptosustav s malim tajnim eksponentom. Mi ćemo dati dokaz nešto općenitije tvrdnje (Worleyev teorem), koja se također koristi u proširenjima Wienerova napada.

Teorem 1.4. *Neka je α proizvoljan realan broj, te c pozitivan realan broj. Ako racionalan broj $\frac{p}{q}$ zadovoljava nejednakost*

$$\left|\alpha - \frac{p}{q}\right| < \frac{c}{q^2}, \tag{1.4}$$

onda je

$$\frac{p}{q} = \frac{rp_k \pm sp_{k-1}}{rq_k \pm sq_{k-1}},$$

za neke nenegativne cijele brojeve k, r, s takve da je $rs < 2c$ (i neki izbor predznaka).

Dokaz: Pretpostavit ćemo da je $\alpha < \frac{p}{q}$. U slučaju $\alpha > \frac{p}{q}$, dokaz je analogan. Također ćemo pretpostaviti da je α iracionalan (za α racionalan potrebna je mala modifikacija dokaza). Neka je k najveći neparan broj takav da je

$$\alpha < \frac{p}{q} \leq \frac{p_k}{q_k}.$$

(Ako je $\frac{p}{q} > \frac{p_1}{q_1}$, onda uzimamo $k = -1$.) Definirajmo brojeve r i s s

$$\begin{aligned} p &= rp_{k+1} + sp_k, \\ q &= rq_{k+1} + sq_k. \end{aligned}$$

Prema relaciji (1.3), determinanta ovog sustava je ± 1, pa su r, s cijeli brojevi, a kako je $\frac{p_{k+1}}{q_{k+1}} < \frac{p}{q} \leq \frac{p_k}{q_k}$, vrijedi $r \geq 0$ i $s > 0$.

Zbog maksimalnosti od k, imamo

$$\left|\frac{p_{k+2}}{q_{k+2}} - \frac{p}{q}\right| < \left|\alpha - \frac{p}{q}\right| < \frac{c}{q^2}.$$

Nadalje,

$$\begin{aligned} \left|\frac{p_{k+2}}{q_{k+2}} - \frac{p}{q}\right| &= \frac{(a_{k+2}q_{k+1} + q_k)(rp_{k+1} + sp_k) - (a_{k+2}p_{k+1} + p_k)(rq_{k+1} + sq_k)}{qq_{k+2}} \\ &= \frac{sa_{k+2} - r}{qq_{k+2}}. \end{aligned}$$

Stoga je

$$q(sa_{k+2} - r) < cq_{k+2} = \frac{c}{s}((sa_{k+2} - r)q_{k+1} + q),$$

tj.

$$(sa_{k+2} - r)(q - \frac{c}{s}q_{k+1}) < \frac{c}{s}\,q.$$

Dalje imamo

$$\frac{1}{sa_{k+2} - r} > \frac{q - \frac{c}{s}q_{k+1}}{\frac{c}{s}q} = \frac{s}{c} - \frac{1}{r + \frac{sq_k}{q_{k+1}}} \geq \frac{s}{c} - \frac{1}{r}.$$

Tako smo dobili nejednakost (kvadratnu nejednadžbu po r):

$$r^2 - sra_{k+2} + ca_{k+2} > 0. \tag{1.5}$$

Razlikujemo sada dva slučaja:

1) $s^2a_{k+2} \geq 4c$

   Uz ovu pretpostavku je $s^4a_{k+2}^2 - 4cs^2a_{k+2} \geq (s^2a_{k+2} - 4c)^2$, pa za rješenje nejednadžbe (1.5) vrijedi da je

   $$r < \frac{1}{2s}\left(s^2a_{k+2} - \sqrt{s^4a_{k+2}^2 - 4cs^2a_{k+2}}\right) \leq \frac{2c}{s},$$

   ili

   $$r > \frac{1}{2s}\left(s^2a_{k+2} + \sqrt{s^4a_{k+2}^2 - 4cs^2a_{k+2}}\right) \geq \frac{1}{s}(s^2a_{k+2} - 2c).$$

   Prva mogućnost povlači $rs < 2c$. Ako je nastupila druga mogućnost, uvodimo supstituciju $t = sa_{k+2} - r$. Broj t je prirodan i vrijedi

   $$p = rp_{k+1} + sp_k = (sa_{k+2} - t)p_{k+1} + sp_k = sp_{k+2} - tp_{k+1},$$
   $$q = sq_{k+2} - tq_{k+1}$$

   i $st = s^2a_{k+2} - rs < 2c$.

2) $s^2a_{k+2} < 4c$

   Ako je $r < \frac{1}{2}sa_{k+2}$, onda $rs < \frac{1}{2}s^2a_{k+2} < 2c$. Ako je $\frac{1}{2}sa_{k+2} \leq r < sa_{k+2}$, onda ponovo definiramo $t = sa_{k+2} - r$ i vrijedi $st \leq \frac{1}{2}s^2a_{k+2} < 2c$.

□

Završit ćemo ovo poglavlje s jednom primjenom verižnih razlomaka. Već je Fermat znao da se neparan prost broj može prikazati kao zbroj kvadrata dva cijela broja ako i samo ako je $p \equiv 1 \pmod 4$. Međutim, ostaje pitanje kako za dani prost broj p oblika $4k + 1$ naći cijele brojeve x, y takve da je $x^2 + y^2 = p$. Prikazat ćemo Hermiteovu konstrukciju koja koristi verižne razlomke.

Neka je z neko rješenje kongruencije $z^2 \equiv -1 \pmod p$ (o rješavanju ovakvih kongruencija bit će riječi u sljedećem poglavlju). Dakle, $z^2 + 1$ je neki višekratnik od p kojeg znamo prikazati kao zbroj dva kvadrata. Želimo pomoću njega naći prikaz broja p u tom obliku. Promotrimo verižni razlomak

$$\frac{z}{p} = [a_0; a_1, \ldots, a_m].$$

Postoji jedinstveni cijeli broj n takav da je $q_n < \sqrt{p} < q_{n+1}$. Budući da $\frac{z}{p}$ leži između susjednih konvergenti $\frac{p_n}{q_n}$ i $\frac{p_{n+1}}{q_{n+1}}$, slijedi da je

$$\left|\frac{z}{p} - \frac{p_n}{q_n}\right| < \left|\frac{p_n}{q_n} - \frac{p_{n+1}}{q_{n+1}}\right| = \frac{1}{q_n q_{n+1}}\,.$$

Dakle, $\frac{z}{p} = \frac{p_n}{q_n} + \frac{\varepsilon}{q_n q_{n+1}}$, gdje je $|\varepsilon| < 1$. Odavde je $zq_n - pp_n = \frac{\varepsilon p}{q_{n+1}}$, pa je $(zq_n - pp_n)^2 < \frac{p^2}{q_{n+1}^2} < p$. Konačno,

$$(zq_n - pp_n)^2 + q_n^2 \equiv q_n^2(z^2+1) \equiv 0 \pmod{p} \text{ i } 0 < (zq_n - pp_n)^2 + q_n^2 < 2p,$$

što povlači da je $(zq_n - pp_n)^2 + q_n^2 = p$.

1.7 Kvadratne kongruencije

Definicija 1.8. Neka su a i m relativno prosti cijeli brojevi i $m \geq 1$. Kažemo da je a *kvadratni ostatak* modulo m ako kongruencija $x^2 \equiv a \pmod{m}$ ima rješenja. Ako ova kongruencija nema rješenja, onda kažemo da je a *kvadratni neostatak* modulo m.

Pretpostavimo sada da je modul kongruencije neparan prost broj p. Tada kvadratnih ostataka modulo p ima jednako mnogo kao i kvadratnih neostataka, tj. $(p-1)/2$. Ova tvrdnja neposredno slijedi iz činjenice da je grupa $(\mathbb{Z}_p^*, \cdot_p)$ ciklička. To znači da postoji element $g \in \mathbb{Z}_p$ čiji je red jednak $p-1$. Taj element g se naziva *primitivni korijen* modulo p. Sada je jasno da su $g^0, g^2, g^4, \ldots, g^{p-3}$ kvadratni ostatci, a $g^1, g^3, g^5, \ldots, g^{p-2}$ kvadratni neostatci.

Kad smo se već prisjetili definicije primitivnog korijena, recimo nešto o tome kako se on nalazi. Možemo krenuti redom i testirati je li $g = 2$, $g = 3$, $\ldots$ primitivni korijen. Pritom ne treba testirati brojeve oblika g_0^k, $k \geq 2$, jer ako g_0 nije primitivni korijen, onda to ne može biti ni g_0^k. Testiranje je li g primitivni korijen se zasniva na sljedećoj očitoj činjenici: g je primitivni korijen ako i samo ako za svaki prosti faktor q od $p-1$ vrijedi $g^{(p-1)/q} \not\equiv 1 \pmod{p}$.

Može se postaviti pitanje kolika je vjerojatnost da već $g = 2$ bude primitivan korijen. S tim u vezi spomenimo poznatu Artinovu slutnju koja kaže da za prirodan broj a, koji nije potencija nekog prirodnog broja, vrijedi $\nu_a(N) \sim A \cdot \pi(N)$, gdje je $\pi(N)$ broj prostih brojeva $\leq N$, $\nu_a(N)$ broj prostih brojeva $\leq N$ za koje je a primitivni korijen, dok je A Artinova konstanta

$$\prod_{p \text{ prost}} \left(1 - \frac{1}{p(p-1)}\right) \approx 0.3739558.$$

Poznato je da tzv. generalizirana Riemannova slutnja (GRH) povlači Artinovu slutnju, a također povlači i ocjenu $O(\ln^6 p)$ za najmanji primitivni korijen modulo p.

Vratimo se sada kvadratnim ostatcima modulo p.

Definicija 1.9. Neka je p neparan prost broj. *Legendreov simbol* $(\frac{a}{p})$ jednak je 1 ako je a kvadratni ostatak modulo p; jednak je -1 ako je a kvadratni neostatak modulo p; a jednak je 0 ako je $a \equiv 0 \pmod{p}$.

Vidimo da je broj rješenja kongruencije $x^2 \equiv a \pmod{p}$ jednak $1 + (\frac{a}{p})$.

Postavlja se pitanje kako izračunati Legendreov simbol. Jedna mogućnost je pomoću *Eulerova kriterija* koji glasi:

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

Budući da smo već vidjeli kako se može efikasno potencirati, Eulerov kriterij nam omogućava da Legendreov simbol izračunamo uz $O(\ln^3 p)$ bitnih operacija. No, postoji i efikasniji algoritam, čija je složenost $O(\ln^2 p)$, a koji je vrlo sličan Euklidovu algoritmu. Taj algoritam je zasnovan na *Gaussovu kvadratnom zakonu reciprociteta*, koji glasi

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4}$$

za različite proste brojeve p i q. Dakle, ovaj zakon nam omogućava da $(\frac{p}{q})$ zamijenimo s $(\frac{q}{p})$, što je posebno korisno ukoliko je $p < q$. Međutim, za primjenu ovog zakona oba parametra moraju biti prosti brojevi, što dolazi od zahtjeva u definiciji Legendreova simbola da jedan od parametara (donji) bude prost. To nas vodi do potrebe uvođenja poopćenja Legendreova simbola kod kojeg parametri neće morati biti prosti.

Definicija 1.10. Neka je m neparan prirodan broj i $m = \prod_{i=1}^{k} p_i^{\alpha_i}$ njegov rastav na proste faktore, te neka je a cijeli broj. *Jacobijev simbol* $(\frac{a}{m})$ se definira s

$$\left(\frac{a}{m}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{\alpha_i},$$

gdje $(\frac{a}{p_i})$ predstavlja Legendreov simbol.

Jasno je da ako je m prost, onda se Jacobijev i Legendreov simbol podudaraju. Ako je $\mathrm{nzd}(a, m) > 1$, onda je $(\frac{a}{m}) = 0$. Ako je a kvadratni ostatak modulo m, onda je a kvadratni ostatak modulo p_i za svaki i. Zato je $(\frac{a}{p_i}) = 1$ za svaki i, pa je i $(\frac{a}{m}) = 1$. Međutim, $(\frac{a}{m}) = 1$ ne povlači da je a kvadratni ostatak modulo m. Da bi a bio kvadratni ostatak modulo m, nužno je i dovoljno da svi $(\frac{a}{p_i})$ budu jednaki 1.

Navodimo osnovna svojstva Jacobijeva simbola koja se koriste u njegovom računanju:

1) $a \equiv b \pmod{m} \;\Rightarrow\; \left(\frac{a}{m}\right) = \left(\frac{b}{m}\right)$.

2) $\left(\frac{ab}{m}\right) = \left(\frac{a}{m}\right)\left(\frac{b}{m}\right), \quad \left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$

3) $\left(\frac{-1}{m}\right) = (-1)^{(m-1)/2}, \quad \left(\frac{2}{m}\right) = (-1)^{(m^2-1)/8}$

4) $\left(\frac{m}{n}\right)\left(\frac{n}{m}\right) = (-1)^{(m-1)(n-1)/4}$ ako su m i n relativno prosti.

Algoritam 14 Algoritam za računanje Jacobijeva simbola $(\frac{a}{m})$

```
1:  a = a mod m; t = 1
2:  while (a ≠ 0) do
3:      while (a paran) do
4:          a = a/2
5:          if (m ≡ 3,5 (mod 8)) then t = −t
6:      (a, m) = (m, a)
7:      if (a ≡ m ≡ 3 (mod 4)) then t = −t
8:      a = a mod m
9:  if (m = 1) then return t
10:     else return 0
```

Vidimo da je ovaj algoritam vrlo sličan Euklidovu algoritmu. Tu je ključno svojstvo 4) (generalizacija kvadratnog zakona reciprociteta), koje nam omogućava da zamijenimo gornji i donji parametar u Jacobijevu simbolu, a nakon toga veći parametar možemo zamijeniti s njegovim ostatkom pri dijeljenju s onim manjim. Jedina bitna razlika od Euklidova algoritma je u posebnom tretiranju faktora 2, kojeg moramo izlučiti prije nego što zamijenimo gornji i donji parametar.

Pretpostavimo sada da je $(\frac{a}{p}) = 1$. To znači da postoji cijeli broj x takav da je

$$x^2 \equiv a \pmod{p}\,. \tag{1.6}$$

Postavlja se pitanje kako naći taj broj x, tj. kako efikasno izračunati kvadratni korijen od a modulo p. Ako je p vrlo mali, to možemo napraviti tako da ispitamo redom sve moguće ostatke modulo p. No, za imalo veće p-ove, to je vrlo neefikasan algoritam.

Odgovor na postavljeno pitanje je vrlo lak za brojeve specijalnog oblika $p = 4k + 3$ (pa se zato takvi brojevi koriste u Rabinovom kriptosustavu). Zapravo, mogli bismo reći da taj oblik i nije jako specijalan, budući da pola prostih brojeva ima takav oblik.

Propozicija 1.1. *Ako je* $p \equiv 3 \pmod{4}$*, onda je* $x = a^{(p+1)/4}$ *rješenje kongruencije (1.6).*

Dokaz: Budući da je a kvadratni ostatak modulo p, iz Eulerova kriterija imamo $a^{(p-1)/2} \equiv 1 \pmod{p}$, pa je

$$x^2 \equiv a^{(p+1)/2} \equiv a \cdot a^{(p-1)/2} \equiv a \pmod{p}. \qquad \square$$

Prethodnu je propoziciju moguće modificirati i na preostale proste brojeve, uz poznavanje barem jednog kvadratnog neostatka modulo p. Ako je $p \equiv 5 \pmod{8}$, onda je broj 2 kvadratni neostatak modulo p. Upravo se ta činjenica koristi u sljedećoj propoziciji.

Propozicija 1.2. *Ako je $p \equiv 5 \pmod{8}$, onda je jedan od brojeva*

$$a^{(p+3)/8}, \ 2^{(p-1)/4}a^{(p+3)/8}$$

rješenje kongruencije (1.6).

Dokaz: Ako je $p = 8k+5$, onda je $a^{4k+2} \equiv 1 \pmod{p}$. Odavde je $a^{2k+1} \equiv \pm 1 \pmod{p}$, pa je $a^{2k+2} \equiv \pm a \pmod{p}$. Ako u posljednjoj kongruenciji imamo predznak $+$, onda je $x = a^{k+1} = a^{(p+3)/8}$ rješenje kongruencije (1.6).

Ukoliko imamo predznak $-$, onda iskoristimo činjenicu da je $(\frac{2}{p}) = -1$. To povlači da je $2^{4k+2} \equiv -1 \pmod{p}$, pa za $x = 2^{(p-1)/4}a^{(p+3)/8}$ vrijedi

$$x^2 \equiv 2^{4k+2}a^{2k+2} \equiv (-1)(-a) \equiv a \pmod{p}. \qquad \square$$

Preostao je slučaj $p \equiv 1 \pmod{8}$. Taj slučaj je i najteži, zato što ne možemo eksplicitno napisati jedan kvadratni neostatak modulo p (iako znamo da ih ima "puno", tj. $(p-1)/2$). Opisat ćemo Tonellijev algoritam za nalaženje kvadratnog korijena u tom slučaju. Pretpostavimo da nam je poznat jedan kvadratni neostatak d modulo p. Ovo je teoretski najproblematičniji dio algoritma. Naime, nije poznat niti jedan (bezuvjetni) deterministički polinomijalni algoritam za nalaženje kvadratnog neostatka. Pretpostavimo li da vrijedi proširena Riemannova slutnja (ERH), onda postoji kvadratni neostatak manji od $2\ln^2 p$, pa nam jednostavno pretraživanje daje polinomijalni algoritam. U praksi ovo nije problem, jer je vjerojatnost da je slučajno izabrani broj kvadratni neostatak jednaka $1/2$. Tako je vjerojatnost da od 20 slučajno izabranih brojeva niti jedan nije kvadratni neostatak manja od 10^{-6}.

Neka je $p = 2^s t + 1$, gdje je t neparan. Prema Eulerovu kriteriju imamo:

$$a^{2^{s-1}t} \equiv 1 \pmod{p}, \quad a^{2^{s-2}t} \equiv \pm 1 \pmod{p}, \quad d^{2^{s-1}t} \equiv -1 \pmod{p}.$$

Dakle, postoji $t_2 \geq 0$ takav da je

$$a^{2^{s-2}t}d^{t_2 2^{s-1}} \equiv 1 \pmod{p}, \quad a^{2^{s-3}t}d^{t_2 2^{s-2}} \equiv \pm 1 \pmod{p}.$$

Analogno zaključujemo da postoji $t_3 \geq 0$ takav da je

$$a^{2^{s-3}t}d^{t_3 2^{s-2}} \equiv 1 \pmod{p}, \quad a^{2^{s-4}t}d^{t_3 2^{s-3}} \equiv \pm 1 \pmod{p}.$$

Nastavljajući ovaj postupak, na kraju dobijemo $t_s \geq 0$ takav da je

$$a^t d^{2t_s} \equiv 1 \pmod{p},$$

pa je $x = a^{(t+1)/2} d^{t_s}$ rješenje kongruencije (1.6).

Spojivši gornje dvije propozicije i Tonellijev algoritam, dobivamo sljedeći algoritam za računanje rješenja kongruencije $x^2 \equiv a \pmod{p}$.

Algoritam 15 Kvadratni korijen modulo p

$a = a \bmod p$
if ($p \equiv 3, 7 \pmod 8$) then
    $x = a^{(p+1)/4} \bmod p$;
    return x
if ($p \equiv 5 \pmod 8$) then
    $x = a^{(p+3)/8} \bmod p$
    $c = x^2 \bmod p$
    if ($c \neq a \bmod p$) then $x = x \cdot 2^{(p-1)/4} \bmod p$
    return x
nađi broj $d \in \{2, 3, \cdots, p-1\}$ takav da je $\left(\frac{d}{p}\right) = -1$
prikaži $p - 1 = 2^s t$, t neparan
$A = a^t \bmod p$
$D = d^t \bmod p$
$m = 0$
for ($0 \leq i \leq s-1$) do
    if ($(AD^m)^{2^{s-1-i}} \equiv -1 \pmod p$) then $m = m + 2^i$
$x = a^{(t+1)/2} D^{m/2} \bmod p$
return x

Neka je d prirodan broj, te p neparan prost broj. Promotrimo diofantsku jednadžbu

$$x^2 + dy^2 = p.$$

Zanima nas ima li ova jednadžba rješenja u cijelim brojevima, te ako ih ima, kako ih naći. Ovaj problem se između ostalog pojavljuje kod primjene eliptičkih krivulja u testiranju prostosti. Specijalni slučaj $d = 1$ smo već obradili u poglavlju o verižnim razlomcima, gdje smo prikazali konstrukciju za rješavanje problema zbroja kvadrata. Sada ćemo navesti jednu modifikaciju te konstrukcije, *Cornacchia-Smithov algoritam*, koja rješava ovaj općenitiji problem.

Nužan uvjet za postojanje rješenja je da je $-d$ kvadratni ostatak modulo p. Zaista, iz $x^2 + dy^2 = p$ slijedi $(xy^{-1})^2 \equiv -d \pmod p$. Napomenimo da obrat općenito ne vrijedi (osim u slučaju kada je *broj klasa* $h(-4d)$ jednak 1,

tj. kada su sve binarne kvadratne forme s diskriminantom $-4d$ međusobno ekvivalentne). Neka je dakle z cijeli broj takav da je

$$z^2 \equiv -d \pmod{p}$$

i neka je $\frac{p}{2} < z < p$. Primjenjujemo Euklidov algoritam na (p, z) sve dok ne dođemo do ostatka $r < \sqrt{p}$. Promotrimo broj $t = (p - r^2)/d$. Ako je $t = s^2$ za neki $s \in \mathbb{Z}$, onda je $p = r^2 + ds^2$. U protivnom jednadžba nema rješenja.

Algoritam 16 Cornacchia-Smithov algoritam

1: if $((\frac{-d}{p}) = -1)$ then return nema rješenja
2: $z = \sqrt{-d} \bmod p$
3: if $(2z < p)$ then $z = p - z$
4: $(a, b) = (p, z)$
5: $c = \lfloor \sqrt{p} \rfloor$
6: $t = p - b^2$
7: if $(t \not\equiv 0 \pmod{d})$ then return nema rješenja
8: if (t/d nije potpun kvadrat) then return nema rješenja
9: return $(b, \sqrt{t/d})$

1.8 Kvadrati i kvadratni korijeni

U ovom poglavlju razmatramo pitanje kako za dani prirodni broj n što efikasnije odrediti je li n potpun kvadrat ili nije, te ako jest potpun kvadrat, kako izračunati njegov kvadratni korijen. Općenitije, pitamo se kako za proizvoljni prirodni broj n izračunati cjelobrojni dio kvadratnog korijena od n, tj. broj $\lfloor \sqrt{n} \rfloor$. Ove probleme smo već susreli u Cornacchia-Smithovu algoritmu.

Algoritam za računanje $\lfloor \sqrt{n} \rfloor$ je zapravo varijanta poznate Newtonove iterativne metode za približno računanje korijena jednadžbe. Podsjetimo se da se u Newtonovoj metodi aproksimacije rješenja jednadžbe $f(x) = 0$ računaju po formuli

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Kod nas je $f(x) = x^2 - n$, a algoritam ćemo modificirati tako da radi samo s cijelim brojevima.

Algoritam 17 Algoritam za $\lfloor\sqrt{n}\rfloor$

```
1: x = n
2: y = ⌊(x + ⌊n/x⌋)/2⌋
3: while (y < x) do
4:       x = y
5:       y = ⌊(x + ⌊n/x⌋)/2⌋
6: return x
```

Dokažimo da ovaj algoritam stvarno računa $\lfloor\sqrt{n}\rfloor$. Neka je $q = \lfloor\sqrt{n}\rfloor$. Budući da je $\frac{1}{2}(t + \frac{n}{t}) \geq \sqrt{n}$, za svaki pozitivan broj t, imamo da je $x \geq q$ u svim koracima algoritma. U zadnjem koraku imamo $y = \lfloor(x + \lfloor n/x\rfloor)/2\rfloor = \lfloor(x + \frac{n}{x})/2\rfloor \geq x$. Želimo dokazati da je $x = q$. Pretpostavimo suprotno, tj. da je $x \geq q + 1$. Tada je

$$y - x = \left\lfloor \frac{x + \frac{n}{x}}{2} \right\rfloor - x = \left\lfloor \frac{\frac{n}{x} - x}{2} \right\rfloor = \left\lfloor \frac{n - x^2}{2x} \right\rfloor .$$

Iz $x \geq q + 1 > \sqrt{n}$ slijedi $n - x^2 < 0$ i $y - x < 0$, što je kontradikcija.

Složenost ovog algoritma je $O(\ln^3 n)$. Efikasnost algoritma se može poboljšati ako se za inicijalnu vrijednost, umjesto $x = n$, uzme broj koji bolje aproksimira (odozgo) broj $\sqrt{n}$. Na primjer, ako je $2^e \leq n < 2^{e+1}$, onda možemo uzeti $x = 2^{\lfloor(e+2)/2\rfloor}$. Tako se može dobiti algoritam složenosti $O(\ln^2 n)$.

Neka je sada n prirodan broj. Želimo testirati je li n potpun kvadrat ili nije. Jedna mogućnost je izračunati $q = \lfloor\sqrt{n}\rfloor$ i provjeriti je li q^2 jednako n. No, "većina" prirodnih brojeva nisu kvadrati. Stoga bi bilo dobro barem neke od tih "nekvadrata" eliminirati na neki jednostavniji način. Ideja je iskoristiti činjenicu da ako je n potpun kvadrat, onda je n kvadratni ostatak modulo m za svaki m koji je relativno prost s n. Dakle, ako je n kvadratni neostatak po nekom modulu, onda n sigurno nije kvadrat. Ova se ideja u praksi realizira tako da se izabere nekoliko konkretnih modula, te se unaprijed izračunaju kvadrati u pripadnom prstenu.

Za modul m, generiramo pripadnu tablicu qm ovako ($qm[k]$ označava k-ti element tablice):

for $(0 \leq k \leq m-1)$ do $qm[k] = 0$
for $(0 \leq k \leq \lfloor m/2\rfloor)$ do $qm[k^2 \bmod m] = 1$

Jedna preporučena kombinacija modula je 64, 63, 65, 11. Broj kvadrata u $\mathbb{Z}_{64}$, $\mathbb{Z}_{63}$, $\mathbb{Z}_{65}$, $\mathbb{Z}_{11}$ je redom 12, 16, 21, 6. Budući da je

$$\frac{12}{64} \cdot \frac{16}{63} \cdot \frac{21}{65} \cdot \frac{6}{11} = \frac{6}{715} < 0.01 ,$$

vidimo da na ovaj način za više od 99% brojeva ne moramo računati kvadratni korijen da bismo zaključili da nisu kvadrati. Redoslijed kojim testiramo

module dolazi od

$$\frac{12}{64} < \frac{16}{63} < \frac{21}{65} < \frac{6}{11},$$

tako da će se za većinu prirodnih brojeva program zaustaviti već nakon prvog testa modulo 64. Naravno da su mogući i drugi izbori modula.

Algoritam 18 Algoritam za detekciju kvadrata

1: $t = n \bmod 64$
2: if ($q64[t] = 0$) then return n nije kvadrat
3: $r = n \mod 45045$
4: if ($q63[r \bmod 63] = 0$) then return n nije kvadrat
5: if ($q65[r \bmod 65] = 0$) then return n nije kvadrat
6: if ($q11[r \bmod 11] = 0$) then return n nije kvadrat
7: $q = \lfloor \sqrt{n} \rfloor$
8: if ($n \neq q^2$) then return n nije kvadrat
9: else return n je kvadrat, $\sqrt{n} = q$

1.9 LLL-algoritam

Iz svojstava verižnih razlomaka znamo da za dani iracionalni broj α nejednadžba

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

ima beskonačno mnogo rješenja u racionalnim brojevima p/q. Ta tvrdnja se naziva Dirichletov teorem. Poznata su i poopćenja Dirichletovog teorema za simultane racionalne aproksimacije više od jednog racionalnog broja. Ako je barem jedan od brojeva $\alpha_1, \ldots, \alpha_n$ iracionalan, onda postoji beskonačno mnogo n-torki racionalnih brojeva $\frac{p_1}{q}, \ldots, \frac{p_n}{q}$ sa svojstvom

$$\left|\alpha_i - \frac{p_i}{q}\right| < \frac{1}{q^{1+1/n}}, \quad i = 1, \ldots, n. \tag{1.7}$$

Nadalje, ako su $1, \alpha_1, \cdots, \alpha_n$ linearno nezavisni nad $\mathbb{Q}$ (svaka njihova linearna kombinacija s racionalnim koeficijentima koji nisu svi jednaki 0 je različita od 0), onda postoji beskonačno mnogo $(n+1)$-torki relativno prostih brojeva $(q_1, \ldots, q_n, p)$ sa svojstvom

$$q = \max(|q_1|, \ldots, |q_n|) > 0 \quad \text{i} \quad |\alpha_1 q_1 + \cdots + \alpha_n q_n - p| < \frac{1}{q^n}. \tag{1.8}$$

Iako su Dirichletovi teoremi o simultanim aproksimacijama izravna poopćenja (običnog) Dirichletova teorema za $n = 1$, kada je u pitanju efektivna algoritamska realizacija pronalaženja racionalnih brojeva čiju egzistenciju jamče ti teoremi, postoji bitna razlika između slučajeva $n = 1$ i $n \geq 2$. Naime, u slučaju $n = 1$, kao što smo već vidjeli, racionalne aproksimacije sa željenim svojstvom možemo dobiti s pomoću verižnih razlomaka. U slučaju $n \geq 2$ situacija je u tom pogledu složenija. Postoje različita poopćenja verižnih razlomaka, odnosno njihovih svojstava, na više dimenzije. Mi ćemo obraditi jednu, vjerojatno najpoznatiju, i za brojne primjene najvažniju, a to je tzv. LLL-algoritam. On će nam efikasno (u polinomijalnom vremenu) dati racionalne aproksimacije slične kvalitete kao one koje jamče Dirichletovi teoremi (pojavit će se neki dodatni faktori u odnosu na kvalitetu optimalnih aproksimacija koje daje teorija).

LLL-algoritam je povezan s problemom nalaženja najkraćeg nenul-vektora u rešetki.

Definicija 1.11. *Neka je n prirodni broj te neka su $b_1, \ldots, b_n$ linearno nezavisni vektori u $\mathbb{R}^n$.* Rešetka *($\mathbb{Z}$-modul) L razapeta ovim vektorima je skup svih njihovih cjelobrojnih linearnih kombinacija*

$$L = \left\{ \sum_{i=1}^{n} x_i\, b_i \,:\, x_i \in \mathbb{Z} \right\}.$$

Kaže se da je $B = \{b_1, \ldots, b_n\}$ baza rešetke *L.*

Npr. u $\mathbb{R}^2$, ako je $b_1 = (1, 0)$, $b_2 = (0, 1)$, onda je L rešetka svih točaka u ravnini s cjelobrojnim koordinatama.

S B ćemo označavati i matricu čiji su stupci vektori $b_1, \ldots, b_n$. Determinanta rešetke L se definira s $\Delta(L) = |\det(B)|$. Ova definicija je dobra jer je baza rešetke jedinstvena do na množenje zdesna s matricama iz $GL_n(\mathbb{Z})$, tj. matricama s cjelobrojnim koeficijentima i determinantom ± 1.

Sa $\langle\ ,\ \rangle$ ćemo označavati standardni skalarni produkt u $\mathbb{R}^n$. Primijetimo da kvadrat euklidske norme vektora $v = \sum_{i=1}^{n} x_i b_i$ inducira kvadratnu formu

$$\|v\|^2 = v^\tau v = x^\tau B^\tau B x = Q(x).$$

Budući da je rešetka diskretan skup, dobro je definiran pojam duljine najkraćeg nenul-vektora rešetke. Iz Teorema Minkowskog o konveksnom tijelu slijedi da ako je C kompaktan, konveksan skup, simetričan s obzirom na ishodište te ako je $\mu(C) \geq 2^n \Delta(L)$, onda C sadržava nenul-vektor iz L. Ako sada za C uzmemo n-dimenzionalnu kuglu, dobivamo gornju ocjenu za duljinu najkraćeg nenul-vektora u rešetki.

Propozicija 1.3. *Postoji konstanta γ_n takva da vrijedi*

$$\min_{v \in L \setminus \{0\}} \|v\| \leq \sqrt{\gamma_n} \Delta(L)^{1/n}.$$

Optimalne vrijednosti od γ_n su poznate za $n \leq 8$. Primjerice, $\gamma_1 = 1$, $\gamma_2 = \frac{4}{3}$, $\gamma_3 = 2$.

Jedna rešetka može imati više različitih baza, pa se pitamo možemo li izabrati bazu koja bi imala neko dodatno dobro svojstvo. Jasno je da B predstavlja bazu vektorskog prostora $\mathbb{R}^n$. Znamo da Gram-Schmidtovim postupkom možemo dobiti ortogonalnu bazu za isti vektorski prostor ($b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*$, $i = 1, \ldots, n$, gdje je $\mu_{ij} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$). No ta nova baza ne mora razapinjati istu rešetku kao polazna baza B jer koeficijenti μ_{ij} ne moraju biti cijeli brojevi. Općenito, rešetka i ne mora imati ortogonalnu bazu. A. K. Lenstra, H. W. Lenstra i L. Lovász uveli su 1982. godine pojam *LLL-reducirane baze*, koja ima svojstva:

1) $|\mu_{i,j}| \leq \frac{1}{2}$, $1 \leq j < i \leq n$;

2) $\|b_i^*\|^2 \geq (\frac{3}{4} - \mu_{i,i-1}^2) \|b_{i-1}^*\|^2$.

Prvi uvjet se može interpretirati tako da se kaže da je LLL-reducirana baza "skoro ortogonalna", dok drugi uvjet govori da niz normi vektora $\|b_i^*\|$ "skoro raste". Dodatno važno svojstvo LLL-reducirane baze je da je prvi vektor u toj bazi vrlo kratak, tj. ima malu normu. Može se dokazati da uvijek vrijedi da je $\|b_1\| \leq 2^{(n-1)/2} \|x\|$, za sve nenul-vektore $x \in L$, no u primjenama se vrlo često događa da je $\|b_1\|$ upravo najkraći nenul-vektor iz L. To ćemo precizirati u sljedećoj lemi.

Lema 1.2. *Neka je $\{b_1, \ldots, b_n\}$ LLL-reducirana baza te $\{b_1^*, \ldots, b_n^*\}$ pripadna Gram-Schmidtova baza. Tada vrijedi:*

1) $\|b_j\|^2 \leq 2^{i-1}\|b_i^*\|^2$, $1 \leq j \leq i \leq n$;

2) $\Delta(L) \leq \prod_{i=1}^{n} \|b_i\| \leq 2^{n(n-1)/4}\Delta(L)$;

3) $\|b_1\| \leq 2^{(n-1)/4}\left(\Delta(L)\right)^{1/n}$;

4) Za svaki $x \in L$, $x \neq 0$, vrijedi $\|b_1\|^2 \leq c_1\|x\|^2$, gdje je

$$c_1 = \max\left\{\frac{\|b_1\|^2}{\|b_i^*\|^2} \,:\, 1 \leq i \leq n\right\} \leq 2^{n-1}.$$

5) Za vektor $y \notin L$ definiramo $\sigma = B^{-1}y$, gdje je B matrica čiji su stupci $b_1, \ldots, b_n$. Neka je i_0 najveći indeks takav da $\sigma_{i_0} \notin \mathbb{Z}$, te $\|\sigma_{i_0}\|$ udaljenost od σ_{i_0} do najbližeg cijelog broja. Tada za svaki $x \in L$ vrijedi

$$\|x - y\|^2 \geq c_1^{-1}\|\sigma_{i_0}\|^2\|b_1\|^2.$$

Dokaz:

1) Iz definicije LLL-reducirane baze slijedi

$$\|b_i^*\|^2 \geq (\frac{3}{4} - \mu_{i,i-1}^2)\|b_{i-1}^*\|^2 \geq \frac{1}{2}\|b_{i-1}^*\|^2$$

za $i = 1, 2, \ldots, n$. Odavde indukcijom slijedi $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$ za $1 \leq j \leq i \leq n$. Sada iz definicije Gram-Schmidtove baze dobivamo

$$\begin{aligned}\|b_i\|^2 &= \|b_i^*\|^2 + \sum_{j=1}^{i-1}\mu_{i,j}^2\|b_j^*\|^2 \leq \left(1 + \sum_{j=1}^{i-1}2^{i-j-2}\right)\|b_i^*\|^2 \\ &= \left(1 + \frac{1}{4}(2^i - 2)\right)\|b_i^*\|^2 \leq 2^{i-1}\|b_i^*\|^2.\end{aligned}$$

Dakle, za $1 \leq j \leq i \leq n$ vrijedi

$$\|b_j\|^2 \leq 2^{j-1}\|b_j^*\|^2 \leq 2^{j-1+i-j}\|b_i^*\|^2 = 2^{i-1}\|b_i^*\|^2.$$

2) Iz ortogonalnosti vektora b_i^* slijedi

$$\Delta(L) = |\det(b_1^*, \ldots, b_n^*)| = \prod_{i=1}^{n}\|b_i^*\|.$$

Iz tvrdnje 1) i nejednakosti $\|b_i^*\| \leq \|b_i\|$ dobivamo

$$\Delta(L) \leq \prod_{i=1}^{n}\|b_i\| \leq \prod_{i=1}^{n}2^{(i-1)/2}\|b_i^*\| \leq 2^{n(n-1)/4}\prod_{i=1}^{n}\|b_i^*\| = 2^{n(n-1)/4}\Delta(L).$$

3) Uvrstimo $j = 1$ u tvrdnju 1) i uzmimo produkt po svim mogućim i-ovima, pa dobivamo

$$\|b_1\|^{2n} \leq \prod_{i=1}^{n} 2^{i-1}\|b_i^*\|^2 \leq 2^{n(n-1)/2}\Delta(L)^2,$$

otkud izravno slijedi tvrdnja 3).

4) Neka je

$$x = \sum_{i=1}^{n} r_i b_i = \sum_{i=1}^{n} r_i' b_i^*, \quad r_i \in \mathbb{Z},\ r_i' \in \mathbb{R}.$$

Neka je i_0 najveći indeks za koji je $r_{i_0} \neq 0$. Budući da $b_1, \ldots, b_i$ razapinju isti vektorski prostor kao $b_1^*, \ldots, b_i^*$, za sve i te da je b_{i+1}^* projekcija od b_{i+1} na ortogonalni komplement tog prostora, zaključujemo da je $r_{i_0}' = r_{i_0}$. Stoga imamo

$$\|x\|^2 = \sum_{i=1}^{n} r_i'^2\|b_i^*\|^2 \geq {r_{i_0}'}^2\|b_{i_0}^*\|^2 \geq \|b_{i_0}^*\|^2 \geq c_1^{-1}\|b_1\|^2.$$

5) Neka je

$$x = \sum_{i=1}^{n} r_i b_i = \sum_{i=1}^{n} r_i' b_i^*, \quad y = \sum_{i=1}^{n} \sigma_i b_i = \sum_{i=1}^{n} \sigma_i' b_i^*.$$

Ako je i_1 najveći indeks takav da je $r_{i_1} \neq \sigma_{i_1}$, onda je $r_{i_1} - \sigma_{i_1} = r_{i_1}' - \sigma_{i_1}'$, pa imamo

$$\|x - y\|^2 \geq |r_{i_1} - \sigma_{i_1}|^2\|b_{i_1}^*\|^2 \geq |r_{i_1} - \sigma_{i_1}|^2 c_1^{-1}\|b_1\|^2.$$

Ako je $i_1 < i_0$, onda vrijedi $\sigma_{i_0} = r_{i_0} \in \mathbb{Z}$, što je kontradikcija. Ako je $i_1 = i_0$, onda imamo $|r_{i_1} - \sigma_{i_1}| = |r_{i_0} - \sigma_{i_0}| \geq \|\sigma_{i_0}\|$ te dobivamo traženu nejednakost. Naposljetku, ako je $i_1 > i_0$, onda je $\sigma_{i_1} \in \mathbb{Z}$, pa iz $\sigma_{i_1} \neq r_{i_1}$ i $\|\sigma_{i_0}\| \leq \frac{1}{2}$, dobivamo $|r_{i_1} - \sigma_{i_1}| \geq 1 \geq \|\sigma_{i_0}\|$. □

U svom članku iz 1982. godine A. K. Lenstra, H. W. Lenstra i L. Lovász prikazali su polinomijalni algoritam za konstrukciju LLL-reducirane baze iz proizvoljne baze rešetke (prema njima nazvan *LLL-algoritam*). Algoritam je ubrzo našao brojne primjene, npr. u faktorizaciji polinoma s racionalnim koeficijentima, kriptoanalizi kriptosustava RSA s malim javnim ili tajnim eksponentom, problemu ruksaka te diofantskim aproksimacijama i diofantskim jednadžbama.

De Weger je 1989. godine predložio varijantu LLL-algoritma koja se koristi samo cjelobrojnom aritmetikom (ako su ulazni podaci cjelobrojni) te izbjegava probleme vezane uz numeričku stabilnost algoritma.

U programskom paketu PARI je LLL-algoritam implementiran s pomoću funkcije `qflll`(x), koja kao rezultat vraća transformacijsku matricu T takvu da je xT LLL-reducirana baza rešetke generirane stupcima matrice x. Naredbe za nalaženje LLL-reducirane baze postoje i u drugim programskim paketima (npr. `lattice` (Maple), `LatticeReduce` (Mathematica), `LLL` (Magma)).

Prije nego što prijeđemo na primjene rešetki na simultane diofantske aproksimacije u više dimenzija, pogledajmo možemo li dobro poznati nam problem aproksimacije jednog iracionalnog broja racionalnima interpretirati u terminima rešetki. Znamo da se dobre racionalne aproksimacije iracionalnog broja α mogu dobiti s pomoću konvergenti $\frac{p_i}{q_i}$ verižnog razlomka od $\alpha = [a_0, a_1, a_2, \ldots]$. Konvergente zadovoljavaju rekurzije

$$p_{i+1} = a_i p_i + p_{i-1}, \quad q_{i+1} = a_i q_i + q_{i-1}.$$

Uvedemo li oznaku

$$M(i) = \begin{pmatrix} q_i & q_{i+1} \\ p_i & p_{i+1} \end{pmatrix}, \tag{1.9}$$

rekurzije možemo matrično zapisati kao

$$M(i) = M(i-1) \cdot \begin{pmatrix} 0 & 1 \\ 1 & a_i \end{pmatrix}, \tag{1.10}$$

uz $M(-1) = I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Matrice $M(i)$ imaju determinantu ± 1, tj. $M(i) \in GL_2(\mathbb{Z})$. Neka je C pozitivni realni broj. Označimo s $L_C(\alpha)$ rešetku generiranu stupcima matrice $\begin{pmatrix} 1 & 0 \\ -C\alpha & C \end{pmatrix}$. Vrijedi

$$\begin{pmatrix} 1 & 0 \\ -C\alpha & C \end{pmatrix} M(i) = \begin{pmatrix} q_i & q_{i+1} \\ C(p_i - q_i\alpha) & C(p_{i+1} - q_{i+1}\alpha) \end{pmatrix}.$$

Prisjetimo se da je $|p_i - q_i\alpha| < \frac{1}{q_i}$. Ako sada izaberemo $C \approx q_i^2$, onda vidimo da je prvi stupac u zadnjoj matrici vektor $(q_i, C(p_i - q_i\alpha))^\tau$ iz rešetke $L_C(\alpha)$ koji je bitno kraći od vektora polazne baze te rešetke. Nadalje, q_i je prva komponenta tog kratkog vektora.

Propozicija 1.4. *Neka su α i $C > 0$ realni brojevi. Ako je $(u, C(w - \alpha u))^\tau$, $u > 0$, najkraći vektor u rešetki $L_C(\alpha)$, onda je broj u nazivnik neke konvergente verižnog razlomka od α te vrijedi $u \leq \frac{2}{\sqrt{3}}\sqrt{C}$.*

Dokaz: Pretpostavimo suprotno te neka je n najveći indeks sa svojstvom da je $q_n < u$. Tada iz svojstva najboljih aproksimacija slijedi $|p_n - \alpha q_n| < |w - \alpha u|$. Odavde je

$$q_n^2 + C^2(p_n - \alpha q_n)^2 < u^2 + C^2(w - \alpha u)^2,$$

što je u suprotnosti s minimalnosti od $(u, C(w-\alpha u))^\tau$.

Budući da je $\Delta(L_C(\alpha)) = C$, iz Propozicije 1.3 i $\gamma_2 = \frac{4}{3}$ slijedi da je $u \leq \frac{2}{\sqrt{3}}\sqrt{C}$. □

Zaključujemo da se LLL algoritam može iskoristiti za dobivanje dobrih racionalnih aproksimacija. Za razliku od algoritma verižnog razlomka koji nam daje cijeli niz dobrih aproksimacija, ovdje za fiksni C dobivamo jednu dobru aproksimaciju, pa za dobivanje više dobrih aproksimacija treba varirati C.

Sada ćemo ovu ideju primijeniti na problem simultanih diofantskih aproksimacija.

Teorem 1.5. *Neka su $\alpha_1, \ldots, \alpha_n$ realni brojevi te $Q > 1$ prirodni broj. Postoji polinomijalni algoritam koji pronalazi cijele brojeve $q, p_1, \ldots, p_n$ takve da je*

$$1 \leq q \leq 2^{n/4}Q^n \quad te \quad |\alpha_i q - p_i| \leq \frac{\sqrt{5}\cdot 2^{(n-4)/4}}{Q},\ i = 1, \ldots, n. \tag{1.11}$$

Dokaz: Za realni broj x, s $\lfloor x \rceil$ ćemo označavati najbliži cijeli broj broju x. Neka je $C = Q^{n+1}$. Promotrimo matricu

$$B = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\lfloor C\alpha_1 \rceil & C & 0 & \cdots & 0 \\ -\lfloor C\alpha_2 \rceil & 0 & C & \cdots & 0 \\ \vdots & 0 & 0 & \ddots & \vdots \\ -\lfloor C\alpha_n \rceil & 0 & 0 & \cdots & C \end{pmatrix}$$

te rešetku L generiranu njezinim stupcima. Dimenzija rešetke je $n+1$, a volumen C^n. Prema Lemi 1.2.3), LLL-algoritam nalazi vektor

$$r = (q, p_1, \ldots, p_n)^\tau$$

iz $\mathbb{Z}^{n+1}$, takav da je $v = Br$ vektor iz L čija je norma

$$\leq \Lambda := 2^{n/4}\cdot C^{n/(n+1)} = 2^{n/4}Q^n.$$

Komponente vektora v su $v_1 = q$, $v_{i+1} = Cp_i - q\lfloor C\alpha_i \rceil$, $i = 1, \ldots, n$. Vrijedi

$$q^2 + \sum_{i=1}^{n}(Cp_i - q\lfloor C\alpha_i \rceil)^2 \leq \Lambda^2.$$

Dakle, $q \leq \Lambda$ i $\max_{1\leq i\leq n} |Cp_i - q\lfloor C\alpha_i \rceil| \leq \Lambda$. Budući da je $|Cp_i - q\lfloor C\alpha_i \rceil| \geq C|p_i - q\alpha_i| - q/2$, dobivamo

$$|p_i - q\alpha_i| \leq C^{-1}(|Cp_i - q\lfloor C\alpha_i \rceil| + q/2).$$

Naposljetku, iskoristimo još ovu jednostavnu činjenicu: za realne brojeve x, y vrijedi

$$2x + y \leq \sqrt{5(x^2 + y^2)}. \tag{1.12}$$

Zaista, kvadriranjem dobivamo $(x-2y)^2 \geq 0$, što je očito točno. Primijenimo li (1.12) na $x = |Cp_i - q\lfloor C\alpha_i \rceil|$, $y = q$, dobivamo

$$|p_i - q\alpha_i| \leq C^{-1} \cdot \sqrt{5(|Cp_i - q\lfloor C\alpha_i \rceil|^2 + q^2)}/2 \leq \frac{\sqrt{5}}{2C}\Lambda = \frac{\sqrt{5}}{2}2^{n/4}Q^{-1}. \qquad \square$$

Primjer 1.5. Neka je $\alpha_1 = \sqrt{2}$, $\alpha_2 = \sqrt{3}$, $\alpha_3 = \sqrt{5}$. Uzmimo $Q = 1000$ te primijenimo algoritam iz Teorema 1.5. Formiramo matricu B kao u dokazu teorema te s pomoću PARI-ja izračunamo `qflll`(B). Iz prvog stupca dobivene matrice pročitamo brojeve $q = 118452669$, $p_1 = 167517371$, $p_2 = 205166041$, $p_3 = 264868220$. Provjerom dobivamo da je $q < 1.2 \cdot Q^3$, $|q\sqrt{2} - p_1| < 0.91 \cdot Q^{-1}$, $|q\sqrt{3} - p_2| < 0.14 \cdot Q^{-1}$, $|q\sqrt{5} - p_3| < 0.29 \cdot Q^{-1}$,

$$\max\left(\left|\sqrt{2} - \frac{p_1}{q}\right|, \left|\sqrt{3} - \frac{p_2}{q}\right|, \left|\sqrt{5} - \frac{p_3}{q}\right|\right) < q^{-4/3}.$$

Dakle, dobivene simultane racionalne aproksimacije su čak i bolje nego što Teorem 1.5 jamči te zadovoljavaju nejednakost (1.7) iz Dirichletova teorema o simultanim aproksimacijama bez ikakvih dodatnih faktora.

Primjer 1.6. *Nađimo kubni polinom s cjelobrojnim koeficijentima (malim po apsolutnoj vrijednosti) kojem je jedan korijen blizu $\pi = 3.14159\ldots$, a drugi blizu $e = 2.71828\ldots$*

Rješenje: Problem možemo shvatiti kao nalaženje koeficijenata $x_1, \ldots, x_4$ takvih da linearne forme $|x_1\pi^3 + x_2\pi^2 + x_3\pi + x_4|$ i $|x_1e^3 + x_2e^2 + x_3e + x_4|$ budu (istodobno) male, a da pritom $|x_i|$ ne budu preveliki (recimo reda veličine 10^2). Zato promotrimo rešetku generiranu stupcima matrice

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 3101 & 987 & 314 & 100 \\ 2009 & 739 & 272 & 100 \end{pmatrix}$$

(elementi u trećem retku su $\lfloor 100 \cdot \pi^{4-i} \rceil$, a u četvrtom $\lfloor 100 \cdot e^{4-i} \rceil$). Želimo dobiti LLL-reduciranu bazu za ovu rešetku. Točnije, ovdje nas zanima matrica prijelaza T takva da stupci matrice AT čine LLL-reduciranu bazu. Dobivamo

$$\begin{pmatrix} -3 & 0 & -8 & 1 \\ 2 & 1 & -1 & -9 \\ 66 & -6 & 214 & 27 \\ -134 & 9 & -414 & -27 \end{pmatrix}.$$

Sada prvi stupac od T daje koeficijente željenog polinoma

$$f(x) = 3x^3 - 2x^2 - 66x + 134.$$

Njegovi korijeni su približno 3.147875, 2.725321 i -5.20653.

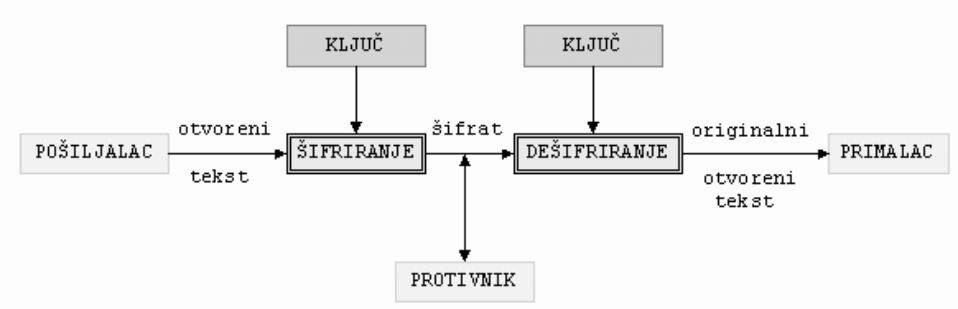
Poglavlje 2

Kriptografija javnog ključa

2.1 Kratki uvod u kriptografiju

Kako uspostaviti sigurnu komunikaciju preko nesigurnog komunikacijskog kanala? Metode za rješavanje ovog problema pročava znanstvena disciplina koja se zove *kriptografija* (ili *tajnopis*). Osnovni zadatak kriptografije je omogućavanje komunikacije dvaju osoba (zovemo ih *pošiljalac* i *primalac* - u kriptografskoj literaturi za njih su rezervirana imena *Alice* i *Bob*) na takav način da treća osoba (njihov *protivnik* - u literaturi se najčešće zove *Eva* ili *Oskar*), koja može nadzirati komunikacijski kanal, ne može razumjeti njihove poruke.

Poruku koju pošiljalac želi poslati primaocu zovemo *otvoreni tekst*. Pošiljalac transformira otvoreni tekst koristeći unaprijed dogovoreni *ključ* K. Taj se postupak zove *šifriranje*, a dobiveni rezultat *šifrat*. Nakon toga pošiljalac pošalje šifrat preko nekog komunikacijskog kanala. Protivnik prisluškujući može saznati sadržaj šifrata, ali kako ne zna ključ, ne može odrediti otvoreni tekst. Za razliku od njega, primalac zna ključ kojim je šifrirana poruka, pa može *dešifrirati* šifrat i odrediti otvoreni tekst.



Ove pojmove ćemo formalizirati u sljedećoj definiciji.

Definicija 2.1. Kriptosustav *je uređena petorka* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, gdje je* $\mathcal{P}$ *konačan skup svih otvorenih tekstova,* $\mathcal{C}$ *konačan skup svih šifrata,* $\mathcal{K}$ *konačan skup svih mogućih ključeva,* $\mathcal{E}$ *skup svih funkcija šifriranja i* $\mathcal{D}$ *skup svih funkcija dešifriranja. Za svaki* $K \in \mathcal{K}$ *postoji* $e_k \in \mathcal{E}$ *i odgovarajući* $d_K \in \mathcal{D}$*. Pritom su* $e_K : \mathcal{P} \to \mathcal{C}$ *i* $d_K : \mathcal{C} \to \mathcal{P}$ *funkcije sa svojstvom da je* $d_K(e_K(x)) = x$ *za svaki* $x \in \mathcal{P}$*.*

Shema koju smo u uvodu opisali predstavlja tzv. *simetrični ili konvencionalni kriptosustav*. Funkcije koje se koriste za šifriranje e_K i dešifriranje d_K ovise o ključu K kojeg Alice i Bob moraju tajno razmjeniti prije same komunikacije. Kako njima nije dostupan siguran komunikacijski kanal, ovo može biti veliki problem.

Godine 1976. Diffie i Hellman su ponudili jedno moguće rješenje problema razmjene ključeva, zasnovano na činjenici da je u nekim grupama potenciranje puno jednostavnije od logaritmiranja. O ovom algoritmu ćemo detaljnije govoriti u jednom od sljedećih poglavlja.

Diffie i Hellman se smatraju začetnicima *kriptografije javnog ključa*. Ideja javnog ključa se sastoji u tome da se konstruiraju kriptosustavi kod kojih bi iz poznavanja funkcije šifriranja e_K bilo praktički nemoguće (u nekom razumnom vremenu) izračunati funkciju dešifriranja d_K. Tada bi funkcija e_K mogla biti javna. Dakle, u kriptosustavu s javnim ključem svaki korisnik K ima dva ključa: javni e_K i tajni d_K. Ako Alice želji poslati Bobu poruku x, onda je ona šifrira pomoću Bobovog javnog ključa e_B, tj. pošalje Bobu šifrat $y = e_B(x)$. Bob dešifrira šifrat koristeći svoj tajni ključ d_B, $d_B(y) = d_B(e_B(x)) = x$. Uočimo da Bob mora posjedovati neku dodatnu informaciju (tzv. *trapdoor* - skriveni ulaz) o funkciji e_B, da bi samo on mogao izračunati njezin inverz d_B, dok je svima drugima (a posebno Eve) to nemoguće. Takve funkcije čiji je inverz teško izračunati bez poznavanja nekog dodatnog podatka zovu se *osobne jednosmjerne funkcije*.

Napomenimo da su kriptosustavi s javnim ključem puno sporiji od modernih simetričnih kriptosustava (DES, IDEA, AES), pa se stoga u praksi ne koriste za šifriranje poruka, već za šifriranje ključeva, koji se potom koriste u komunikaciji pomoću nekog simetričnog kriptosustava.

Druga važna primjena kriptosustava s javnim ključem dolazi od toga da oni omogućavaju da se poruka "*digitalno potpiše*". Naime, ako Alice pošalje Bobu šifrat $z = d_A(e_B(x))$, onda Bob može biti siguran da je poruku poslala Alice (jer samo ona zna funkciju d_A), a također jednakost $e_A(z) = e_B(x)$ predstavlja i dokaz da je poruku poslala Alice, pa ona to ne može kasnije zanijekati.

2.2 Kriptosustavi zasnovani na problemu faktorizacije

U konstrukciji kriptosustava s javnim ključem, tj. osobnih jednosmjernih funkcija, obično se koriste neki "teški" matematički problemi. Jedan od takvih problema je *problem faktorizacije* velikih prirodnih brojeva. O metodama faktorizacije ćemo detaljno govoriti kasnije. Za sada kažimo da je danas praktički nemoguće rastaviti na faktore pažljivo odabran broj s više od 300 znamenaka.

Najpoznatiji kriptosustav s javnim ključem je RSA kriptosustav iz 1977. godine, nazvan po svojim tvorcima Rivestu, Shamiru i Adlemanu. Njegova sigurnost je zasnovana upravo na teškoći faktorizacije velikih prirodnih brojeva. Slijedi precizna definicija RSA kriptosustava.

RSA kriptosustav: Neka je $n = pq$, gdje su p i q prosti brojevi. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$, te

$$\mathcal{K} = \{(n, p, q, d, e) \,:\, n = pq,\ de \equiv 1 \pmod{\varphi(n)}\}.$$

Za $K \in \mathcal{K}$ definiramo

$$e_K(x) = x^e \bmod n, \qquad d_K(y) = y^d \bmod n, \quad x, y \in \mathbb{Z}_n.$$

Vrijednosti n i e su javne, a vrijednosti p, q i d su tajne, tj. (n, e) je javni, a (p, q, d) je tajni ključ.

Ovdje je $\varphi(n)$ Eulerova funkcija. U našem slučaju je $\varphi(n) = \varphi(pq) = (p-1)(q-1) = n - p - q + 1$.

U dokazu da je d_K inverz od e_K koristimo *Eulerov teorem*:

$$x^{\varphi(n)} \equiv 1 \pmod{n}, \quad \text{za } \operatorname{nzd}(x, n) = 1.$$

Uvjerimo se da su funkcije e_K i d_K jedna drugoj inverzne.

Imamo: $d_K(e_K(x)) \equiv x^{de} \pmod{n}$. Iz $de \equiv 1 \pmod{\varphi(n)}$ slijedi da postoji prirodan broj k takav da je $de = k\varphi(n) + 1$. Pretpostavimo da je $\operatorname{nzd}(x, n) = 1$. Sada je

$$x^{de} = x^{k\varphi(n)+1} = (x^{\varphi(n)})^k \cdot x \equiv x \pmod{n}$$

(prema Eulerovom teoremu). Ako je $\operatorname{nzd}(n, x) = n$, onda je $x^{de} \equiv 0 \equiv x \pmod{n}$. Ako je $\operatorname{nzd}(n, x) = p$, onda je $x^{de} \equiv 0 \equiv x \pmod{p}$ i $x^{de} = (x^{q-1})^{(p-1)k} \cdot x \equiv x \pmod{q}$, pa je $x^{de} \equiv x \pmod{n}$. Slučaj $\operatorname{nzd}(n, x) = q$ je potpuno analogan. Prema tome, zaista je $x^{de} \equiv x \pmod{n}$, što znači da je $d_K(e_K(x)) = x$.

Sigurnost RSA kriptosustava leži u pretpostavci da je funkcija $e_K(x) = x^e \bmod n$ jednosmjerna. Dodatni podatak (trapdoor) koji omogućava dešifriranje je poznavanje faktorizacije $n = pq$. Zaista, onaj tko zna faktorizaciju broja n, taj može izračunati $\varphi(n) = (p-1)(q-1)$, te potom dobiti eksponent d rješavajući linearnu kongruenciju

$$de \equiv 1 \pmod{\varphi(n)}$$

(pomoću Euklidovog algoritma).

Postoji i efikasan (vjerojatnosni) algoritam koji iz poznavanja tajnog eksponenta d, računa faktorizaciju $n = pq$. Opišimo ukratko ideju tog algoritma. Za paran broj $m = ed - 1$ vrijedi $a^m \equiv 1 \pmod{n}$ za sve $a \in \mathbb{Z}_n^*$, tj. takve da je $\mathrm{nzd}(a, n) = 1$. To znači da je m neki zajednički višekratnik od $p - 1$ i $q - 1$. Dakle, poznavanje broja m je slabije od poznavanja broja $\varphi(n) = (p - 1)(q - 1)$, ali se ipak može iskoristiti za faktorizaciju broja n. Provjerimo je li $m/2$ zadovoljava isto svojstvo kao m. Ako zadovoljava, onda m zamijenimo s $m/2$ i nastavimo postupak. Ako postoji a takav da $a^{m/2} \not\equiv 1 \pmod{n}$, onda se može pokazati da takvih a-ova ima barem 50% među svim $a \in \mathbb{Z}_n^*$. Ako je a jedan takav broj, onda možemo očekivati da je $\mathrm{nzd}(a^{m/2} - 1, n)$ netrivijalni faktor od n (jer je s vjerojatnošću 50% imamo da je $a^{m/2} - 1$ djeljiv s jednim od brojeva p i q, a nije djeljiv s drugim).

No, otvoreno pitanje je da li je razbijanje RSA kriptosustava, tj. određivanje x iz poznavanja $x^e \bmod n$, ekvivalentno faktorizaciji od n.

Recimo sada nekoliko riječi o izboru parametara u RSA kriptosustavu.

1. Izaberemo tajno dva velika prosta broja p i q slične veličine (oko 150 znamenaka). To radimo tako da najprije generiramo slučajan prirodan broj m s traženim brojem znamenaka, pa zatim pomoću nekog testa prostosti tražimo prvi prosti broj veći ili jednak m. (Po teoremu o distribuciji prostih brojeva, možemo očekivati da ćemo trebati testirati približno $\ln m$ brojeva dok ne nađemo prvi prosti broj.) Treba paziti da $n = pq$ bude otporan na metode faktorizacije koje su vrlo efikasne za brojeva specijalnog oblika. Tako bi brojevi $p \pm 1$ i $q \pm 1$ trebali imati barem jedan veliki prosti faktor, jer postoje efikasne metode za faktorizaciju brojeva koji imaju prosti faktor p takav da je jedan od brojeva $p - 1$, $p + 1$ "gladak", tj. ima samo male proste faktore. Također, p i q ne smiju biti jako blizu jako blizu jedan drugome, jer ih se onda može naći koristeći činjenicu da su približno jednaki $\sqrt{n}$.

2. Izračunamo $n = pq$ i $\varphi(n) = (p - 1)(q - 1) = n - p - q + 1$.

3. Izaberemo broj e takav da je $\mathrm{nzd}(e, \varphi(n)) = 1$, te pomoću Euklidovog algoritma izračunamo d takav da je $de \equiv 1 \pmod{\varphi(n)}$. Obično se uzima da je $e < \varphi(n)$. Broj e se može izabrati slučajno, a ima smisla izabrati ga i što manjim, tako da bi šifriranje $x^e \bmod n$ (tzv. modularno

popenciranje) bilo što brže. Broj operacija u šifriranju ovisi o veličini broja e, te o broju jedinica u binarnom zapisu od e. Stoga je dugo vremena $e = 3$ bio popularan izbor. No, vidjet ćemo da izbor vrlo malog eksponenta e predstavlja opasnost za sigurnost, te se danas preporuča izbor $e = 2^{16} + 1 = 65537$.

4. Stavimo ključ za šifriranje (n, e) u javni direktorij.

Usko povezan s problemom faktorizacije je *problem računanja kvadratnog korijena* u $\mathbb{Z}_n$. Neka je $n = pq$, gdje su p, q prosti brojevi. Za $1 \leq a \leq n-1$ treba naći $x \in \mathbb{Z}$ takav da je $x^2 \equiv a \pmod{n}$, uz pretpostavku da takav x postoji, tj. da je a kvadratni ostatak modulo n. Vidjeli smo da postoji efikasan algoritam za rješavanje kongruencije $x^2 \equiv a \pmod{p}$. Algoritam je posebno jednostavan ako je $p \equiv 3 \pmod{4}$. Naime, tada je rješenje $x \equiv \pm a^{(p+1)/4} \pmod{p}$. Kombinirajući dva rješenja $\pm r$ kongruencije $x^2 \equiv a \pmod{p}$ i dva rješenja $\pm s$ kongruencije $x^2 \equiv a \pmod{q}$, po Kineskom teoremu o ostatcima dobivamo cetiri rješenja kongruencije $x^2 \equiv a \pmod{pq}$.

Obrnuto, ako znamo riješiti problem kvadratnog korijena, onda znamo riješiti i problem faktorizacije. Neka je dan složen broj n. Odaberimo slučajan broj x takav da je $\mathrm{nzd}(x, n) = 1$ (ako je $\mathrm{nzd}(x, n) > 1$, onda smo našli faktor od n) i izračunajmo $a = x^2 \bmod n$. Primijenimo algoritam za problem kvadratnog korijena na broj a. Tako dobijemo broj y. Ako je $y \equiv \pm x \pmod{n}$, onda biramo novi x. Ako je $y \not\equiv \pm x \pmod{n}$, onda iz $n | (x-y)(x+y)$ slijedi da je $\mathrm{nzd}(x - y, n)$ netrivijalni faktor od n. Kako postoje četiri kvadratna korijena od a modulo n, to je vjerojatnost uspjeha ovog algoritma u jednom koraku $1/2$, a očekivani broj potrebnih koraka je dva.

Rabinov kriptosustav (Rabin, 1979.) zasnovan je na teškoći računanja kvadratnog korijena modulo fiksni složeni broj. Štoviše, za njega vrijedi da je njegovo razbijanje ekvivalentno rješavanju problema kvadratnog korijena, pa je, u skladu s gore pokazanim, ekvivalentno i problemu faktorizacije. Ova činjenica pokazuje jednu, barem teoretsku, prednost ovog kriptosustava pred RSA kriptosustava.

Rabinov kriptosustav: Neka je $n = pq$, gdje su p i q prosti brojevi takvi da je $p \equiv q \equiv 3 \pmod{4}$. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$, te

$$\mathcal{K} = \{(n, p, q) : n = pq\}.$$

Za $K \in \mathcal{K}$ definiramo

$$e_K(x) = x^2 \bmod n, \qquad d_K(y) = \sqrt{y} \bmod n.$$

Vrijednost n je javna, a vrijednosti p i q su tajne.

Ovdje $a = \sqrt{b} \bmod n$ znači da je $a^2 \equiv b \pmod{n}$. Uvjet $p \equiv q \equiv 3 \pmod{4}$ se može izostaviti. No, uz ovaj uvjet je dešifriranje jednostavnije i efikasnije.

Jedan nedostatak Rabinovog kriptosustava je da funkcija e_K nije injekcija. Naime, postoje četiri kvadratna korijena modulo n, pa dešifriranje nije moguće provesti da jednoznačan način (osim ako je otvoreni tekst neki smisleni tekst, a to nije slučaj kod razmjene ključeva, za što se kriptosustavi s javnim ključem prvenstveno koriste). Jedan način za rješavanje ovog problema je da se u otvoreni tekst na umjetan način ubaci izvjesna pravilnost. To se može napraviti npr. tako da se posljednja 64 bita dupliciraju (ponove). Tada možemo očekivati da će samo jedan od 4 kvadratna korijena dati rezultat koji ima zadanu pravilnost.

Williams je 1980. dao jednu modifikaciju Rabinovog kriptosustava kojom se također eliminira ovaj nedostatak. U toj modifikaciji se kreće od prostih brojeva p, q sa svojstvom $p \equiv 3 \pmod{8}$, $q \equiv 7 \pmod{8}$. Tada je Jacobijev simbol $(\frac{2}{pq}) = -1$, pa se svojstva Jacobijevog simbola mogu iskoristiti za identifikaciju "pravog" kvadratnog korijena.

2.3 Kriptosustavi zasnovani na problemu diskretnog logaritma

Neka je G konačna abelova grupa. Da bi bila prikladna za primjene u kriptografiji javnog ključa, grupa G bi trebala imati svojstvo da su operacije množenja i potenciranja u njoj jednostavne, dok je logaritmiranje (inverzna operacija od potenciranja) vrlo teško. Također bi trebalo biti moguće generirati slučajne elemente grupe na gotovo uniforman način. Ipak, centralno pitanje jest koliko je težak tzv. *problem diskretnog logaritma* u grupi G.

> **Problem diskretnog logaritma:** Neka je $(G, *)$ konačna grupa, $g \in G$, $H = \{g^i : i \geq 0\}$ podgrupa od G generirana s g, te $h \in H$. Treba naći najmanji nenegativni cijeli broj x takav da je $h = g^x$, gdje je $g^x = \underbrace{g * g * ... * g}_{x \text{ puta}}$. Taj broj x se zove *diskretni logaritam* i označava se s $\log_g h$.

Činjenicu da postoje grupe u kojima je problem diskretnog logaritma težak, iskoristili su Diffie i Hellman u svom rješenju problema razmjene ključeva.

Pretpostavimo da se Alice i Bob žele dogovoriti o jednom tajnom slučajnom elementu u grupi G, kojeg bi onda poslije mogli koristi kao ključ za šifriranje u nekom simetričnom kriptosustavu. Oni taj svoj dogovor moraju

provesti preko nekog nesigurnog komunikacijskog kanala, bez da su prethodno razmjenili bilo kakvu informaciju. Jedina informacija koju imaju jest grupa G i njezin generator g (pretpostavimo zbog jednostavnosti da je grupa G ciklička).

Slijedi opis Diffie-Hellmanovog protokola. Sa $|G|$ ćemo označavati broj elemenata u grupi G.

Diffie-Hellmanov protokol za razmjenu ključeva:

1. Alice generira slučajan prirodan broj $a \in \{1, 2, ..., |G| - 1\}$. Ona pošalje Bobu element g^a.
2. Bob generira slučajan prirodan broj $b \in \{1, 2, ..., |G| - 1\}$, te pošalje Alice element g^b.
3. Alice izračuna $(g^b)^a = g^{ab}$.
4. Bob izračuna $(g^a)^b = g^{ab}$.

Sada je njihov tajni ključ $K = g^{ab}$.

Njihov protivnik (Eve), koji može prisluškivati njihovu komunikaciju preko nesigurnog komunikacijskog kanala, zna sljedeće podatke: G, g, g^a, g^b. Eve treba iz ovih podataka izračunati g^{ab} (kaže se da Eve treba riješiti *Diffie-Hellmanov problem* (DHP)). Ako Eve iz poznavanja g i g^a može izračunati a (tj. ako može riješiti problem diskretnog logaritma (DLP)), onda i ona može pomoću a i g^b izračunati g^{ab}. Vjeruje se da su za većinu grupa koje se koriste u kriptografiji ova dva problema, DHP i DLP, ekvivalentni (tj. da postoje polinomijalni algoritmi koji svode jedan problem na drugi).

U originalnoj definiciji Diffie-Hellmanovog protokola za grupu G se uzima multiplikativna grupa $\mathbb{Z}_p^*$ svih ne-nul ostataka modulo p, gdje je p dovoljno velik prost broj. Poznato je da je grupa $\mathbb{Z}_p^*$ ciklička. Generator ove grupe se naziva *primitivni korijen* modulo p.

Sada ćemo opisati *ElGamalov kriptosustav* iz 1985. godine, koji zasnovan na teškoći računanja diskretnog logaritma u u grupi $(\mathbb{Z}_p^*, \cdot_p)$.

Pokazuje se da je ovaj problem približno iste težine kao problem faktorizacije složenog broja n (ako su p i n istog reda veličine), a i neke od metoda koje koriste u najboljim poznatim algoritmima za rješavanje tih problema su vrlo slične.

ElGamalov kriptosustav: Neka je p prost broj i $\alpha \in \mathbb{Z}_p^*$ primitivni korijen modulo p. Neka je $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ i

$$\mathcal{K} = \{(p, \alpha, a, \beta) \,:\, \beta \equiv \alpha^a \pmod{p}\}.$$

Vrijednosti p, α, β su javne, a vrijednost a je tajna.

Za $K \in \mathcal{K}$ i tajni slučajni broj $k \in \{0, 1, \ldots, p-1\}$ definiramo

$$e_K(x, k) = (\alpha^k \bmod p,\, x\beta^k \bmod p).$$

Za $y_1, y_2 \in \mathbb{Z}_p^*$ definiramo

$$d_K(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod p.$$

Mogli bismo reći da se otvoreni tekst x "zamaskira" množeći s β^k. Onaj tko poznaje tajni eksponent a može iz α^k izračunati β^k i "ukloniti masku".

Da bi eksponent a stvarno bio tajan, prost broj p mora biti dovoljno velik da bi u $\mathbb{Z}_p^*$ problem diskretnog logaritma bio praktički nerješiv. Stoga se danas preporuča korištenje prostih brojeva od barem 1024 bita. Također bi red grupe, tj. broj $p-1$, trebao imati barem jedan veliki prosti faktor (od barem 160 bitova).

No, nije $\mathbb{Z}_p^*$ jedina grupa kod koje je potenciranje puno lakše od logaritmiranja. Dapače, ima grupa, poput grupe eliptičke krivulje nad konačnim poljem, kod kojih je razlika u težini ova dva problema (potenciranja i logaritmiranja) još veća.

Ideju o tome da bi eliptičke krivulje mogle biti korisne u konstrukciji kriptosustava s javnim ključem prvi su javno iznijeli Koblitz i Miller 1985. godine.

Definicija 2.2. *Neka je K polje karakteristike različite od 2 i 3.* Eliptička krivulja *nad poljem K je skup svih uređenih parova $(x, y) \in K \times K$ koji zadovoljavaju jednadžbu*

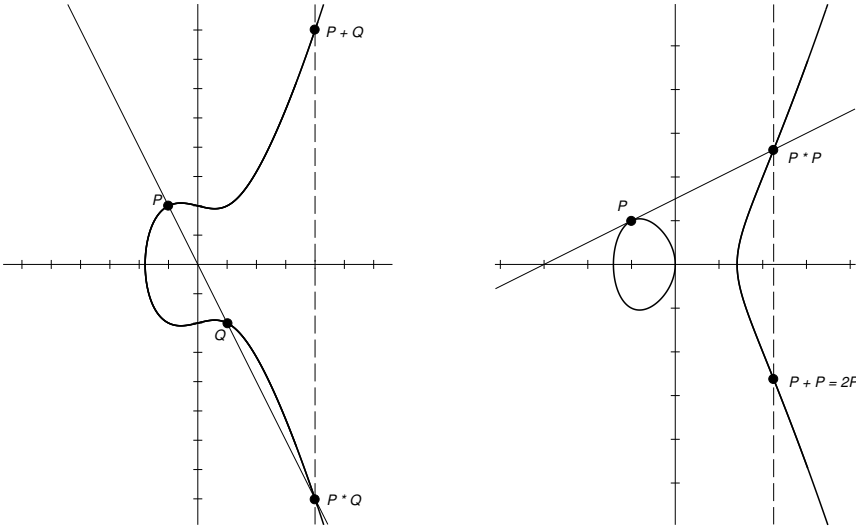
$$E: \qquad y^2 = f(x) = x^3 + ax + b,$$

gdje su $a, b \in K$ i polinom $f(x)$ nema višestrukih korijena, zajedno s "točkom u beskonačnosti" koju ćemo označavati sa $\mathcal{O}$. Taj skup označavamo s $E(K)$.

Vrlo slično se definira eliptička krivulja i nad poljima karakteristike 2 ili 3.

Jedno od najvažnijih svojstava eliptičkih krivulja jest da se na njima može, na prirodan način, uvesti operacija uz koju one postaju abelove grupe. Da bi to objasnili, uzmimo da je $K = \mathbb{R}$ polje realnih brojeva. Tada eliptičku krivulju $E(\mathbb{R})$ (bez točke u beskonačnosti) možemo prikazati kao podskup ravnine.

Definirat ćemo operaciju zbrajanja na $E(\mathbb{R})$. Neka su $P, Q \in E(\mathbb{R})$. Povucimo pravac kroz točke P i Q. On siječe krivulju E u tri točke. Treću točku označimo s $P * Q$. Sada definiramo da je $P + Q$ osnosimetrična točka točki $P * Q$ s obzirom na os x. Ako je $P = Q$, onda umjesto sekante povlačimo tangentu kroz točku P. Po definiciji stavljamo da je $P + \mathcal{O} = \mathcal{O} + P = P$ za svaki $P \in E(\mathbb{R})$.



sekanta tangenta

Pokazuje se da skup $E(\mathbb{R})$ uz ovako definiranu operaciju zbrajanja postaje abelova grupa. Očito je $\mathcal{O}$ neutralni element, dok je $-P$ osnosimetrična točka točki P u odnosu na os x. Možemo zamišljati da je $\mathcal{O}$ treća točka presjeka od E s (vertikalnim) pravcem kroz P i $-P$. Komutativnost je također očita, a najteže je provjeriti asocijativnost.

Analitička geometrija nam omogućava da operaciju zbrajanja, koju smo definirali geometrijski, zapišemo pomoću algebarskih formula. Te formule nam omogućavaju da definiramo zbrajanje točaka na eliptičkoj krivulji nad proizvoljnim poljem K (uz malu modifikaciju za slučaj polja s karakteristikom 2 i 3). Ponovo je skup $E(K)$, uz tako definirano zbrajanje, abelova grupa.

Za primjene eliptičkih krivulja u kriptografiji posebno je važan slučaj kada je polje $K = \mathbb{Z}_p$, ili općenitije kada je K konačno polje. Među konačnim poljima, pored polja $\mathbb{Z}_p$, najvažija su polja karakteristike 2. Eliptičke krivulje imaju važnu primjenu i na probleme faktorizacije i dokazivanja prostosti.

Kao što smo već spomenuli, glavni razlog za uvođenje eliptičkih krivulja u kriptografiju javnog ključa jest taj da je problem diskretnog logaritma u grupi $E(\mathbb{Z}_p)$ još teži od problema diskretnog logaritma u grupi $\mathbb{Z}_p^*$.

To pak znači da se ista sigurnost može postići s manjim ključem. Tako je npr. umjesto ključa duljine duljine 1024 bita, dovoljan ključ duljine 160 bitova. To je osobito važno kod onih primjena (kao što su npr. čip-kartice)

kod kojih je prostor za pohranu ključeva vrlo ograničen.

Spomenimo da postoje i post-kvantni kriptografski algoritmi (dakle, oni koji bi trebali biti otporni i na napade pomoću kvantnih računala) temeljeni na svojstvima izogenija tzv. supersingularnih eliptičckih krivulja, od kojih je najpoznatiji protokol za razmjenu ključeva SIDH (supersingular isogeny Diffie- Hellman key exchange)

Multiplikativna grupa konačnog polja i grupa točaka na eliptičkoj krivulji nad konačnim poljem su dva najvažija tipa grupa koje se koriste u kriptografiji javnog ključa. Pored njih, još su dva tipa grupa proučavana u ovom kontekstu. Prvi tip su grupe klasa ideala u imaginarnim kvadratnim poljima (ili, što je ekvivalentno, grupe klasa pozitivno definitnih kvadratnih formi). No, nakon što je McCurley 1989. godine pronašao efikasan algoritam za problem diskretnog logaritma u njima, interes za primjenu ovih grupa u kriptografiji je znatno smanjen.

Drugi tip su tzv. Jacobijani hipereliptičkih krivulja, o kojima čemo pokušati nešto reći.

Hipereliptička krivulja genusa g nad poljem K ima jednadžbu

$$C: \quad y^2 + h(x)y = f(x),$$

gdje je f normirani polinom stupnja $2g+1$, a h polinom stupnja najviše g. Polinomi f i h imaju koeficijente u polju K. Ako je K karakteristike 2, onda možemo uzeti da je $h = 0$. Dakle, eliptičke krivulje možemo shvatiti kao krivulje genusa 1. Za razliku od eliptičkih krivulja, na točkama na krivulji genusa većeg od 1, ne možemo da prirodan način uvesti grupovnu strukturu. Ipak, hipereliptičkim krivuljama možemo pridružiti jednu važnu grupu, tzv. Jacobijan, koja se može shvatiti kao analogon grupe točaka na eliptičkoj krivulji.

Jacobijan $J_K(C)$ krivulje C nad poljem K se može opisati na više načina. Mi ćemo ovdje dati prikaz (tzv. Mumfordova reprezentacija) koji vodi k efikasnom algoritmu za grupovnu operaciju. Također ćemo pretpostaviti da je K karakteristike različite od 2. Elemente grupe $J_K(C)$ ćemo reprezentirati s parom (a, b) polinoma $a, b \in K[x]$, takvih da je $\deg b < \deg a \leq g$ i $b^2 \equiv f \pmod a$.

Cantor-Koblitzov algoritam za zbrajanje u Jacobijanu:

Algoritam računa $(a_3, b_3) = (a_1, b_1) + (a_2, b_2)$.

Dvostrukom primjenom Euklidovog algoritma izračunaj
$d = \gcd(a_1, a_2, b_1 + b_2) = s_1a_1 + s_2a_2 + s_3(b_1 + b_2)$
$a_3 = a_1a_2/d^2$
$b_3 = (s_1a_1b_2 + s_2a_2b_1 + s_3(b_1b_2 + f))/d \mod a_3$
while $(\deg a_3 > g)$ {
$a_3 = (f - b_3^2)/a_3$
$b_3 = -b_3 \mod a_3$ }

Pokazuje se da je $J_K(C)$ uz ovako definirano zbrajanje abelova grupa. Kao i u slučaju eliptičkih krivulja, najteže je dokazati asocijativnost. Neutralni element je par $(1, 0)$.

Druga interpretacija Jacobijana je pomoću tzv. divizora. *Divizor* na krivulji je formalna suma točaka

$$D = \sum_{P \in C(\overline{K})} n_P P,$$

gdje su n_P cijeli brojevi i svi osim konačno mnogo od njih su jednaki 0. Uz zbrajanje po komponentama, skup svih divizora postaje grupa, koju označavamo sa $Div(C)$. Veza između gornjeg prikaza Jacobijana i divizora je sljedeća. Ako su a i b polinomi kao gore, te ako su x_1, ..., x_t nultočke od a, onda točke $(x_i, b(x_i))$ leže na krivulji C, pa paru (a, b) možemo pridružiti divizor

$$div(a, b) = \sum_{i=1}^{t} (x_i, b(x_i)).$$

Ako je $g = 1$, onda je $\deg a = 1$, $\deg b = 0$, pa elemente Jacobijana eliptičke krivulje možemo identificirati s točkama na eliptičkoj krivulji.

Ako za K uzmemo konačno polje s q elemenata, onda je red od $J_K(C)$ približno jednak q^g. U usporedbi s eliptičkim krivuljama, to znači da za manje vrijednosti od q možemo dobiti grupe dovoljno velikog reda, da bi problem diskretnog logaritma u $J_K(C)$ bio težak. S druge strane, grupovna operacija na eliptičkoj krivulji je puno jednostavnija za implementaciju, i to je glavni razlog što da za sada u praksi ne preporuča primjena hipereliptičkih, umjesto eliptičkih krivulja.

2.4 Wienerov napad na RSA kriptosustav

Budući da je broj operacija za modularno potenciranje linearan u broju bitova eksponenta, na prvi pogled čini se kao dobra ideja pokušati izabrati parametre RSA kriptosustava tako da jedan od eksponenata e ili d bude mali. To bi moglo smanjiti vrijeme potrebno za šifriranje, odnosno dešifriranje, što bi posebno moglo biti od interesa u situacijama kad postoji veliki nesrazmjer u snazi dvaju uređaja koji sudjeluju u komunikaciji, kao što je npr. slučaj kad "pametna kartica" komunicira s centralnim računalom. U toj situaciji bismo možda poželjeli kartici dodijeliti mali tajni eksponent, a računalu mali javni eksponent, da bismo minimizirali onaj dio računanja koje treba provesti kartica. Međutim, vidjet ćemo da takav izbor eksponenata ipak nije dobar. Sljedeći teorem M. Wienera iz 1990. godine pokazuje da u slučaju izbora relativno malog tajnog eksponenta d (u odnosu na n) postoji efikasan algoritam za razbijanje RSA šifre.

Teorem 2.1. *Neka je $n = pq$ i $p < q < 2p$, te neka je $e < \varphi(n)$ i $d < \frac{1}{3}n^{0.25}$. Tada postoji polinomijalni algoritam koji iz poznavanja n i e izračunava d.*

Dokaz: Iz $ed \equiv 1 \pmod{\varphi(n)}$ slijedi da postoji prirodan broj k takav da je $ed - k\varphi(n) = 1$. Odavde je

$$\left|\frac{e}{\varphi(n)} - \frac{k}{d}\right| = \frac{1}{d\varphi(n)}. \tag{2.1}$$

Dakle, $\frac{k}{d}$ je dobra aproksimacija od $\frac{e}{\varphi(n)}$. Međutim, mi ne znamo $\varphi(n)$. Stoga ćemo $\varphi(n)$ aproksimirati s n. Iz $\varphi(n) = n - p - q + 1$ i $p + q - 1 < 3\sqrt{n}$ slijedi $|n - \varphi(n)| < 3\sqrt{n}$. Zamijenimo $\varphi(n)$ s n u (2.1), pa dobivamo:

$$\begin{aligned}\left|\frac{e}{n} - \frac{k}{d}\right| &= \left|\frac{ed - k\varphi(n) - kn + k\varphi(n)}{nd}\right| = \left|\frac{1 - k(n - \varphi(n))}{nd}\right| \\ &\leq \frac{3k\sqrt{n}}{nd} = \frac{3k}{d\sqrt{n}}.\end{aligned}$$

Sada je $k\varphi(n) = ed - 1 < ed$, pa iz $e < \varphi(n)$ (to je standardna pretpostavka u RSA kriptosustavu), slijedi $k < d < \frac{1}{3}n^{0.25}$, te dobivamo

$$\left|\frac{e}{n} - \frac{k}{d}\right| \leq \frac{1}{d\sqrt[4]{n}} < \frac{1}{2d^2}. \tag{2.2}$$

Iz Legenderovog teorema o verižnim razlomcima slijedi da relacija (2.2) povlači da je k/d neka konvergenta razvoja u verižni razlomak od e/n. Iz rekurzije za nazivnike konvergenti $\frac{p_k}{q_k}$ verižnog razlomka, slijedi da je $q_k \geq F_k$, gdje je F_k k-ti Fibonaccijev broj, što znači da nazivnici konvergenti rastu eksponencijano. U našem slučaju dakle slijedi da ima $O(\log n)$ konvergenti od $\frac{e}{n}$. Jedna od njih je $\frac{k}{d}$. Dakle, izračunamo sve konvergente od $\frac{e}{n}$ i testiramo koja od njih zadovoljava uvjet $(x^e)^d \equiv x \pmod{n}$ za slučajno odabran broj x. To daje polinomijalni algoritam za otkrivanje tajnog ključa d.

Drugi način za testiranje točnosti pretpostavke da je neka konkretna konvergenta jednaka $\frac{k}{d}$, jest da se, uz tu pretpostavku, izračuna $\varphi(n) = (p-1)(q-1) = (ed-1)/k$. Tada se može izračunati $\frac{p+q}{2}$ iz identiteta

$$\frac{pq - (p-1)(q-1) + 1}{2} = \frac{p+q}{2},$$

te $\frac{q-p}{2}$ iz identiteta $(\frac{p+q}{2})^2 - pq = (\frac{q-p}{2})^2$. Ako se na ovaj način dobije da su brojevi $\frac{p+q}{2}$ i $\frac{q-p}{2}$ cijeli, onda zaključujemo da je promatrana konvergenta stvarno jednaka $\frac{k}{d}$. Tada iz $\frac{p+q}{2}$ i $\frac{q-p}{2}$ možemo lako dobiti i faktorizaciju modula $n = pq$. □

Primjer 2.1. Pretpostavimo da su u RSA kriptosustavu zadani modul

$$n = 7978886869909,$$

javni eksponent

$$e = 3594320245477,$$

te da je poznato da tajni eksponent d zadovoljava $d < \frac{1}{3}n^{0.25} < 561$. Da bismo primijenili Wienerov napad, računamo razvoj broja $\frac{e}{n}$ u verižni razlomak. Dobivamo:

$$[0, 2, 4, 1, 1, 4, 1, 2, 31, 21, 1, 3, 1, 16, 3, 1, 114, 10, 1, 4, 5, 1, 2].$$

Potom računamo pripadne konvergente:

$$0, \frac{1}{2}, \frac{4}{9}, \frac{5}{11}, \frac{9}{20}, \frac{41}{91}, \frac{50}{111}, \frac{141}{313}, \frac{4421}{9814}, \ldots .$$

Konačno, provjeravamo koji od nazivnika 2, 9, 11, 20, 91, 111, 313 zadovoljava kongruenciju $(x^e)^d \equiv x \pmod{n}$ za npr. $x = 2$. Tako dobivamo da je tajni eksponent $d = 313$. ♢

U ovom primjeru smo vidjeli da je prava konvergenta bila upravo zadnja koja je zadovoljavala uvjet za veličinu nazivnika. To nam sugerira da možda uopće nije nužno testirati sve konvergente u zadanom rasponu, već da bi moglo biti moguće karakterizirati pravu konvergentu. Zaista, to se može napraviti preciznijom ocjenom za $\varphi(n)$. Uz razumnu pretpostavku da je $n > 10^8$, dobije se da je $\frac{k}{d}$ jedinstvena konvergenta koja zadovoljava nejednakost

$$\frac{2e}{n\sqrt{n}} < \frac{k}{d} - \frac{e}{n} < \frac{2.122\, e}{n\sqrt{n}}.$$

Verheul i van Tilborg (1997), te Dujella (2004) prikazali su dvije varijante Wienerova napada na RSA u kojem je tajni ključ veći od $\sqrt[4]{n}$. Neka je $d = D\sqrt[4]{n}$. Ako D nije jako velik, onda možemo pokušati $\frac{k}{d}$ prikazati u obliku $\frac{rp_{m+1} \pm sp_m}{rq_{m+1} \pm sq_m}$, gdje su r, s nenegativni cijeli brojevi i $\frac{p_m}{q_m}$ konvergenta verižnog razlomka od $\frac{e}{n}$. Koristeći poopćenje Legendreova teorema (Worleyev teorem 1.4), mogu se dobiti ocjene za broj parova (r, s) koje treba ispitati u najlošijem slučaju. Te su ocjene ugrubo $O(D^2)$, dakle eksponencijalne u D. Preciznije, broj mogućih parova (r, s) u Verheul - van Tilborgovom napadu je $O(D^2A^2)$, gdje je $A = \max\{a_i : i = m+1, m+2, m+3\}$, dok je u Dujellinoj varijanti $O(D^2 \log A)$ (a_i su parcijalni kvocijenti u razvoju u verižni razlomak).

Ilustrirat ćemo tu varijantu Wienerovog napada na sljedećem primjeru.

Primjer 2.2. Neka je $n = 7978886869909$, $e = 4603830998027$, i pretpostavimo da je $d < 10000000$. Razvoj u verižni razlomak broja $\frac{e}{n}$ je

$$[0, 1, 1, 2, 1, 2, 1, 18, 10, 1, 3, 3, 1, 6, 57, 2, 1, 2, 14, 7, 1, 2, 1, 4, 6, 2],$$

a prvih nekoliko konvergenti je

$$0,\ 1,\ \frac{1}{2}, \frac{3}{5}, \frac{4}{7}, \frac{11}{19}, \frac{15}{26}, \frac{281}{487}, \frac{2825}{4896}, \ldots .$$

Tražimo dvije susjedne neparne konvergente između kojih se nalazi $\frac{e}{n} + \frac{2.122e}{n\sqrt{n}}$. Dobivamo:

$$\frac{281}{487} < \frac{e}{n} + \frac{2.122e}{n\sqrt{n}} < \frac{11}{19}.$$

Sada tajni eksponent d tražimo među brojevima nekog od oblika $26r + 19s$ ili $487s - 26t$ ili $4896r' + 487s'$. Primjenjujući kriterij za testiranje kandidata za razlomak $\frac{k}{d}$ opisan u dokazu Teorema 2.1, nalazimo da je $d = 5936963$, što se dobiva za $s = 12195$, $t = 77$. ♢

Vremenska složenost ovog napad može poboljšati primjenom metode "susret u sredini" (*meet-in-the-middle*) za testiranje kandidata. Želimo testirati je li

$$2^{e(rq_{m+1}+sq_m)} \equiv 2 \pmod{n}.$$

Primijetimo da je indeks m skoro fiksiran. Naime, ako je m' najveći neparan prirodan broj takav da je

$$\frac{p_{m'}}{q_{m'}} > \frac{e}{n} + \frac{2.122e}{n\sqrt{n}},$$

onda je $m \in \{m', m'+1, m'+2\}$.

Uvedimo oznake:

$$2^{eq_{m+1}} \bmod n = a, \quad (2^{eq_m})^{-1} \bmod n = b.$$

Tada zapravo možemo testirati kongruenciju

$$a^r \equiv 2b^s \pmod{n}.$$

To možemo napraviti tako da izračunamo $a^r \bmod n$ za sve r, sortiramo rezultate, a potom računamo $2b^s \bmod n$ redom za svaki s i provjeravamo pojavljuje li se rezultat u prethodno dobivenoj sortiranoj listi. Na ovaj način broj koraka u testiranju postaje ugrubo (broj mogućnosti za r) + (broj mogućnosti za s). Preciznije, vremenska složenost faze testiranja smanjuje sa $O(D^2)$ na $O(D \log D)$ (uz prostornu (memorijsku) složenost $O(D)$). Ovaj napad radi efikasno za vrijednosti od D do 2^{30}, tj. za $d < 2^{30} n^{0.25}$.

2.5 Napadi na RSA koji koriste LLL-algoritam

Postoje i napadi koji, umjesto verižnih razlomaka, koriste Coppersmithovu metodu za nalaženje rješenja polinomijalnih kongruencija. Naime, radi se o sljedećem problemu. Neka je zadan polinom $f(x) \in \mathbb{Z}[x]$ stupnja d i neka je poznato da postoji "malo" rješenje kongruencije $f(x) \equiv 0 \pmod{N}$, tj. rješenje x_0 za koje vrijedi $|x_0| < N^{1/d}$. Pitanje je možemo li efikasno naći x_0. Coppersmith je pokazao da je odgovor na ovo pitanje potvrdan. Osnovna

ideja je konstruirati novi polinom $h(x) = h_0 + h_1 x + \cdots h_n x^n \in \mathbb{Z}[x]$ za kojeg će također vrijediti $h(x_0) \equiv 0 \pmod N$, ali koji će imati male koeficijente. Preciznije, traži se da "norma" $\|h(x)\| := \left(\sum_{i=0}^{n} h_i^2\right)^{1/2}$ bude mala. Tada se može iskoristiti sljedeća jednostavna činjenica: ako za prirodan broj X vrijedi

$$\|h(xX)\| < \frac{N}{\sqrt{n+1}}$$

i $|x_0| < X$ zadovoljava kongruenciju $h(x_0) \equiv 0 \pmod N$, onda je x_0 nultočka polinoma h, tj. vrijedi ne samo kongruencija, već i jednakost $h(x_0) = 0$.

Polinom $h(x)$ s traženim svojstvom može se naći pomoću LLL-algoritma. Naime, koeficijenti polinoma $h(x)$ mogu se dobiti kao komponente prvog vektora LLL-reducirane baze određene rešetke koja se dobije pomoću koeficijenata polaznog polinoma $f(x)$.

Boneh i Durfee su opisali jedan napad na RSA ovakvog tipa koji je primjenjiv u slučaju da je $d < n^{0.292}$. Slično kao kod Weinerova napada, kreće se od jednakosti $ed - k\varphi(n) = 1$, koja se može zapisati i kao

$$ed - k(n + 1 - p - q) = 1.$$

Stavimo $s = p + q$, $a = n + 1$. Sada je nalaženje malog tajnog eksponenta d, recimo $d < n^{\delta}$, ekvivalentno nalaženju malih rješenja k i s kongruencije

$$f(k, s) = k(s - a) \equiv 1 \pmod e.$$

Zaista, za k i e imamo sljedeće ocjene:

$$|s| < 3\sqrt{n} \approx e^{0.5}, \quad |k| < \frac{de}{\varphi(n)} \approx e^{\delta}.$$

Dakle, situacija je slična kao kod gore navedenog Coppersmithova rezultata, samo što se ovdje radi o polinomu od dvije varijable, pa se Coppersmithov teorem ne može direktno primijeniti da bi se strogo dokazala korektnost ovog napada. Ipak, pokazalo se da on u praksi radi sasvim zadovoljavajuće.

Savjet je da se izbjegava slučaj kada je $d < \sqrt{n}$, jer je poznato da su svi ovi gore spomenuti napadi sasvim neprimjenjivi ako je $d > \sqrt{n}$.

Također postoje i napadi na RSA uz pretpostavku da je eksponent e mali, pa bi i to trebalo izbjegavati. U ranijim implementacijama RSA kriptosustava, često se uzimalo $e = 3$, da bi se minimiziralo vrijeme potrebno za šifriranje. Pokazat ćemo zašto taj izbor za e nije dobar.

Pretpostavimo da imamo tri korisnika s različitim vrijednostima javnog modula n_1, n_2, n_3, te pretpostavimo da svi oni koriste isti javni eksponent $e = 3$. Nadalje, pretpostavimo da im netko želi poslati identičnu poruku m. Tada njihov protivnik može doznati sljedeće šifrate:

$$c_1 \equiv m^3 \pmod{n_1}, \quad c_2 \equiv m^3 \pmod{n_2}, \quad c_3 \equiv m^3 \pmod{n_3}.$$

Nakon toga, on može, koristeći Kineski teorem o ostatcima naći rješenje sustava linearnih kongruencija

$$x \equiv c_1 \pmod{n_1}, \quad x \equiv c_2 \pmod{n_2}, \quad x \equiv c_3 \pmod{n_3}.$$

Na taj način, dobit će broj x sa svojstvom $x \equiv m^3 \pmod{n_1 n_2 n_3}$. No, kako je $m^3 < n_1 n_2 n_3$, zapravo vrijedi jednakost $x = m^3$, pa protivnik može izračunati originalnu poruku m tako na nađe treći korijen iz x.

Upravo opisani napad može se izbjeći tako da se porukama prije šifriranja doda neki "slučajni dodatak" (engl. random pad). Na taj način, nikad nećemo različitim primateljima slati potpuno identične poruke. No, postoje napadi (zasnovani na gore spomenutu Coppersmithovu rezultatu i LLL-algoritmu) koji pokazuju da ni u tom slučaju RSA kriptosustav s vrlo malim eksponentom e nije siguran. Prikazat ćemo sada Håstadov napad (1985).

Pretpostavimo da je, prije šifriranja, na početku svake poruke dodan neki podatak ovisan o korisniku. Npr.

$$c_i = (i \cdot 2^h + m)^e \bmod n_i, \;\; i = 1, \ldots, k.$$

Dakle, imamo k polinoma $g_i(x) = (i \cdot 2^h + x)^e - c_i$, te tražimo m sa svojstvom da je

$$g_i(m) \equiv 0 \pmod{n_i}.$$

Neka je $n = n_1 n_2 \cdots n_k$. Pomoću Kineskog teorema o ostacima možemo naći t_i tako da je

$$g(x) = \sum_{i=1}^{k} t_i g_i(x) \;\text{ i }\; g(m) \equiv 0 \pmod{n}$$

($t_i \equiv 1 \pmod{n_i}$, $t_i \equiv 0 \pmod{n_j}$ za $j \neq i$). Polinom g je normiran i stupnja e. Ako je $k \geq e$, tj. imamo barem onoliko korisnika (presretnutih šifrata) koliki je javni eksponent, onda je $m < \min_i n_i < n^{1/k} \leq n^{1/e}$, pa se m može efikasno naći primjenom gore navedenog Coppersmithovog rezultata.

Može se preporučiti uporaba eksponenta $e = 65537$, koji je dovoljno velik da bi onemogućio sve poznate napade na RSA s malim eksponentom, a prednost mu je vrlo brzo šifriranje jer ima malo jedinica u binarnom zapisu. Naime, $65537 = 2^{16} + 1$.

2.6 Coppersmithov teorem

Recimo sada nešto malo više o Coppersmithovom rezultatu korištenom u napadima opisanim u prethodnom potpoglavlju.

Neka je N veliki složen broj s nepoznatom faktorizacijom. Nadalje, neka je f (normirani) polinom

$$f(x) = f_0 + f_1 x + \cdots + f_{d-1} x^{d-1} + x^d$$

s cjelobrojnim koeficijentima stupnja d, za kojeg je poznato da postoji "malo" rješenje kongruencije $f(x) \equiv 0 \pmod N$, tj. rješenje x_0 za koje vrijedi $|x_0| < N^{1/d}$. Ovdje pretpostavka da je polinom f normiran nije gubitak općenitosti, jer ga inače možemo pomnožiti s inverzom modulo N od vodećeg koeficijenta (a ukoliko inverz ne postoji, onda smo našli netrivijalni faktor od N).

Želimo (efikasno) naći x_0. Kao što smo već spomenuli, osnovna ideja je konstruirati novi polinom $h(x) = h_0 + h_1 x + \cdots h_n x^n \in \mathbb{Z}[x]$ za kojeg će također vrijediti $h(x_0) \equiv 0 \pmod N$, ali koji će imati male koeficijente. Tada se kongruencija modulo N može zamijeniti običnom jednakošću, te problem riješiti nalaženjem cjelobrojnih nultočaka polinoma h.

Lema 2.1. *Neka je $h(x) = h_0 + h_1 x + \cdots h_n x^n \in \mathbb{Z}$ polinom stupnja n, te neka su X i N prirodni brojevi. Pretpostavimo da vrijedi*

$$\|h(xX)\| < \frac{N}{\sqrt{n+1}}.$$

Tada ako $|x_0| < X$ zadovoljava kongruenciju $h(x_0) \equiv 0 \pmod N$, onda je $h(x_0) = 0$.

Dokaz: Iz nejednakosti trokuta, te nejednakosti aritmetičke i kvadratne sredine, imamo:

$$\begin{aligned} |h(x_0)| &\le |h_0| + |h_1|X + \cdots + |h_n|X^n \\ &\le \sqrt{n+1} \cdot \sqrt{h_0^2 + h_1^2 X^2 + \cdots + h_n^2 X^{2n}} \\ &= \sqrt{n+1} \cdot \|h(xX)\| < \sqrt{n+1} \cdot \frac{N}{\sqrt{n+1}} = N. \end{aligned}$$

Sada iz $h(x_0) \equiv 0 \pmod N$ i $|h(x_0)| < N$ slijedi da je $h(x_0) = 0$. □

Vratimo se sada na polazni polinom f i kongruenciju $f(x_0) \equiv 0 \pmod N$. Iz ove konkruencije slijedi da je i $f(x_0)^k \equiv 0 \pmod{N^k}$ za svaki $k \ge 1$. Nadalje, ako za dani prirodni broj m definiramo polinome

$$g_{u,v}(x) = N^{m-v} x^u f(x)^v,$$

onda za sve $0 \le u < d$ i $0 \le v \le m$ vrijedi

$$g_{u,v}(x_0) \equiv 0 \pmod{N^m}.$$

Sada ćemo traženi polinom h tražiti u obliku

$$h(x) = \sum_{u=0}^{d-1} \sum_{v=0}^{m} a_{u,v} g_{u,v}(x),$$

gdje su $a_{u,v} \in \mathbb{Z}$.

Dakle, želimo naći cijele brojeve $a_{u,v}$ tako da dobiveni polinom h zadovoljava nejednakost

$$\|h(xX)\| \leq \frac{N^m}{\sqrt{d(m+1)}}.$$

Ovaj je problem može shvatiti kao problem nalaženja malog vektora u odgovarajućoj rešetki, i stoga ga se može riješiti pomoću LLL-algoritma. Ilustrirat ćemo to na primjeru polinoma malog stupnja ($d = 2$).

Zadan je polinom

$$f(x) = x^2 + ax + b$$

i želimo naći (mali) x_0 takav da je $f(x_0) \equiv 0 \pmod{N}$. Uzmimo $m = 2$, te konstruirajmo polinome $g_{u,v}$ na gore opisani način:

$$\begin{aligned}
g_{0,0}(xX) &= N^2, \\
g_{1,0}(xX) &= XN^2x, \\
g_{0,1}(xX) &= bN + aXNx + NX^2x^2, \\
g_{1,1}(xX) &= bNXx + aNX^2x^2 + NX^3x^3, \\
g_{0,2}(xX) &= b^2 + 2abXx + (a^2+2b)X^2x^2 + 2aX^3x^3 + X^4x^4, \\
g_{1,2}(xX) &= b^2Xx + 2abX^2x^2 + (a^2+2b)X^3x^3 + 2aX^4x^4 + X^5x^5.
\end{aligned}$$

Cilj nam je naći linearnu kombinaciju ovih šest polinoma tako da dobiveni polinom ima male koeficijente. Drugim riječima, tražimo kratki vektor u rešetki generiranoj stupcima sljedeće matrice, u kojoj redci odgovaraju potencijama od x, a stupci polinomima $g_{u,v}$:

$$A = \begin{pmatrix}
N^2 & 0 & bN & 0 & b^2 & 0 \\
0 & XN^2 & aNX & bNX & 2abX & Xb^2 \\
0 & 0 & NX^2 & aNX^2 & (a^2+2b)X^2 & 2abX^2 \\
0 & 0 & 0 & NX^3 & 2aX^3 & (a^2+2b)X^3 \\
0 & 0 & 0 & 0 & X^4 & 2aX^4 \\
0 & 0 & 0 & 0 & 0 & X^5
\end{pmatrix}.$$

Determinanta ove matrice je $\det(A) = N^6X^{15}$. Primjenom LLL-algoritma na rešetku generiranu stupcima matrice A, dobivamo LLL bazu za tu rešetku. Po Lemi 1.2.3), prvi vektor b_1 te zadovoljava

$$\|b_1\| \leq 2^{5/4} \det(A)^{1/6} = 2^{5/4} N X^{5/2}.$$

Prema tome, polinom

$$h(x) = b_1^{(1)} g_{0,0}(x) + b_1^{(2)} g_{1,0}(x) + \cdots + b_1^{(6)} g_{1,2}(x)$$

zadovoljava

$$\|h(xX)\| \leq 2^{5/4} N X^{5/2}.$$

Sada možemo primijeniti Lemu 2.1 ako je zadovoljen uvjet

$$2^{5/4} N X^{5/2} < N^2/\sqrt{6},$$

tj.

$$X < \frac{N^{0.4}}{6^{0.2} \cdot 2^{0.5}}.$$

Za dovoljno veliki N, to znači da ćemo moći naći rješenje x_0 kongruencije $f(x_0) \equiv 0 \pmod{N}$ ako je

$$|x_0| \leq N^{0.39}.$$

Analogna tehnika se može primijeniti i na polinome proizvoljnog stupnja d. Postoje različiti izbori pripadnih rešetki (obično rešetke veće dimenzije daju bolje eksponente). Ovdje navodimo originalni Coppersmithov teorem koji za dobivanje eksponenta $1/d - \epsilon$ koristi rešetku dimenzije $d(2m-1)$, gdje je $m \geq \max(\frac{d-1+\epsilon d}{\epsilon d^2}, \frac{7}{d})$.

Teorem 2.2 (Coppersmith (1997))**.** *Neka je $f(x) \in \mathbb{Z}[x]$ normirani polinom stupnja d, te neka je N prirodan broj. Ako postoji rješenje kongruencije $f(x_0) \equiv 0 \pmod{N}$ koje zadovoljava nejednakost $|x_0| \leq N^{1/d-\epsilon}$, tada postoji algoritam koji pronalazi x_0, a čija je složenost polinomijalna u $\log N$ i $1/\epsilon$ (za fiksirani d).*

Poglavlje 3

Testiranje i dokazivanje prostosti

3.1 Distribucija prostih brojeva

Definicija 3.1. Za prirodan broj p, koji ima točno dva djelitelja 1 i p, kažemo da je *prost*. Za prirodan broj $n > 1$, koji nije prost, kažemo da je *složen*.

Prvi dokaz da prostih brojeva ima beskonačno mnogo dao je Euklid oko 300. godine prije Krista. Njegov dokaz se zasniva na činjenici da svaki prirodan broj ima barem jedan prosti djelitelj. Pa ako bi $p_1, p_2, \ldots, p_m$ svi bili prosti brojevi, onda broj $n = p_1 p_2 \cdots p_m + 1$ ne bi imao niti jedan prosti djelitelj.

Za $x \in \mathbb{R}$, sa $\pi(x)$ ćemo označavati broj prostih brojeva koji su $\leq x$. Osnovni rezultat o distribuciji prostih brojeva je *teorem o prostim brojevima* (engl. Prime Number Theorem - PNT) koji kaže da je

$$\pi(x) \sim \frac{x}{\ln x}\,.$$

Ovu činjenicu je prvi naslutio Gauss, a dokazali su je neovisno Hadamard i de la Vallée Poussin 1896. godine.

Još bolja aproksimacija za funkciju $\pi(x)$ je *logaritamsko-integralna funkcija*

$$\mathrm{li}(x) = \int_2^x \frac{1}{\ln t}\, dt.$$

Po l'Hôpitalovu pravilu neposredno dobivamo da je

$$\lim_{x\to\infty} \frac{\mathrm{li}(x)}{x/\ln(x)} = 1.$$

Stoga je teorem o prostim brojevima ekvivalentan sa $\pi(x) \sim \mathrm{li}(x)$.

Sljedeći važan rezultat kojeg ćemo spomenuti je *Dirichletov teorem o prostim brojevima u aritmetičkom nizu* prema kojemu svaki aritmetički niz, kojem su početni član i diferencija relativno prosti, sadrži beskonačno mnogo prostih brojeva. Preciznije, ako $\pi(x; d, a)$ označava broj prostih brojeva $p \leq x$ koji zadovoljavaju $p \equiv a \pmod{d}$, onda vrijedi

$$\pi(x; d, a) \sim \frac{1}{\varphi(d)} \cdot \frac{x}{\ln x}\,,$$

gdje je φ Eulerova funkcija. Dakle, možemo reći da svaka klasa reduciranih ostataka modulo d sadrži podjednako mnogo prostih brojeva.

Važno sredstvo u proučavanju distribucije prostih brojeva je *Riemannova zeta funkcija*, koja je definirana s

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}\,.$$

Ovaj red konvergira apsolutno za $\mathrm{Re}(s) > 1$. Ovako definirana funkcija može se proširiti do meromorfne funkcije na cijeloj kompleksnoj ravnini, koja jedini pol (jednostruki) ima u $s = 1$ i pripadni reziduum je jednak 1. To znači da je funkcija $f(s) = (s-1)\zeta(s)$ analitička na $\mathbb{C}$ i $f(1) = 1$. Proširenje se najprije napravi za $\mathrm{Re}(s) > 0$ pomoću formule

$$\zeta(s) = \frac{s}{s-1} - s\int_1^{\infty} \frac{x - \lfloor x \rfloor}{x^{s+1}}\,dx\,,$$

dok se za $\mathrm{Re}(s) \leq 0$ koristi funkcionalna jednadžba

$$\zeta(s) = 2^s \pi^{s-1} \Gamma(1-s)\zeta(1-s)\sin(\frac{\pi s}{2})\,.$$

Ova funkcija ima očite nultočke u $s = -2, -4, -6, \ldots$, dok se sve ostale nultočke nalaze unutar "kritične trake" $0 < \mathrm{Re}(s) < 1$. Znamenita *Riemannova slutnja* (engl. Riemann Hypothesis - RH) glasi da sve nultočke funkcije $\zeta(s)$ u kritičnoj traci leže na pravcu $\mathrm{Re}(s) = \frac{1}{2}$.

Veza Riemannove zeta funkcije i prostih brojeva dolazi preko *Eulerove produktne formule*

$$\zeta(s) = \prod_{p \text{ prost}} (1 - p^{-s})^{-1},$$

koja je neposredna posljedica teorema o jednoznačnoj faktorizaciji na proste faktore. Svaka informacija o funkciji ζ nam stoga daje neku informaciju o distribuciji prostih brojeva. Tako činjenica da $\zeta(s)$ nema nultočaka na pravcu $\mathrm{Re}(s) = 1$ povlači teorem o prostim brojevima, dok je Riemannova slutnja ekvivalentna sa sljedećom ocjenom

$$\pi(x) = \mathrm{li}(x) + O(x^{1/2+\varepsilon}), \quad \forall \varepsilon > 0.$$

Kod ocjene veličine najmanjeg kvadratnog neostatka modulo p, spominjali smo *proširenu Riemannovu slutnju* (engl. Extended Riemann Hypothesis - ERH). Ona se odnosi na funkcije koje se dobiju modifikacijom funkcije ζ. Neka je $d \in \mathbb{N}$ i $\chi : \mathbb{Z} \to \mathbb{C}$ funkcija sa svojstvima

1) $\chi(mn) = \chi(m)\chi(n)$ za sve $m, n \in \mathbb{Z}$
2) χ je periodična s periodom d
3) $\chi(n) = 0$ ako i samo ako je $\mathrm{nzd}(n, d) > 1$.

Tada se χ naziva *Dirichletov karakter* modulo d. Jedan primjer Dirichletovog karaktera je Jacobijev simbol $(\frac{n}{d})$ za neparan broj $d > 1$. Općenito, ako je $\mathrm{nzd}(n, d) = 1$, onda je $\chi(n)^{\varphi(d)} = \chi(n^{\varphi(d)}) = \chi(1) = 1$. Stoga je $\chi(n)$ neki korijen iz jedinice. Postoji točno $\varphi(d)$ Dirichletovih karaktera modulo d. Uvjerimo se u to u slučaju kada je broj $d = p$ prost. Neka je g primitivni korijen modulo p. Tada je, zbog multiplikativnosti, χ u potpunosti određen s $\chi(g)$. No, za $\chi(g)$ možemo izabrati bilo koji kompleksan broj η takav da je $\eta^{\varphi(p)} = 1$, a takvih brojeva ima $\varphi(p) = p - 1$.

Dirichletova L-funkcija se definira s

$$L(s, \chi) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s},$$

gdje je χ Dirichletov karakter. Sada proširena Riemannova slutnja glasi: za proizvoljan Dirichletov karakter χ, sve nultočke funkcije $L(s, \chi)$ u poluravnini $\mathrm{Re}(s) > 0$ leže na pravcu $\mathrm{Re}(s) = \frac{1}{2}$.

3.2 Pseudoprosti brojevi

Vidjeli smo da se u konstrukciji većine kriptosustava s javnim ključem kreće od jednog ili više velikih prostih brojeva. U RSA kriptosustavu pomoću velikih tajnih prostih brojeva p i q gradimo javni modul n. S druge strane, u ElGamalovu kriptosustavu veliki javni prosti broj p određuje pripadno konačno polje $\mathbb{Z}_p$ u kojem će se vršiti šifriranje. Uloga prostih brojeva u ovim dvama kriptosustavima nije sasvim ista. U slučaju ElGamalova kriptosustava, gdje je p javna veličina, mi slobodno možemo za p koristiti neki broj preporučen u literaturi. S druge strane, znamo da je za sigurnost RSA kriptosustava nužno da brojevi p i q budu tajni (a k tome bi još trebali i zadovoljavati neka dodatna svojstva).

Vidimo da je u kriptografiji javnog ključa važno pitanje kako za dani prirodan broj odrediti je li prost, ili je složen. U ovom poglavlju ćemo govoriti o tzv. testovima prostosti. To su kriteriji koje broj p mora zadovoljiti da bi bio prost. Dakle, ako p ne zadovolji neki od tih kriterija, onda je sigurno složen, a ako ih zadovolji, onda je "vjerojatno prost", što znači da je vrlo velika vjerojatnost da je p prost. Nešto kasnije ćemo prikazati i neke od metoda pomoću kojih je moguće egzaktno dokazati da je neki broj prost. No,

u primjenama se najčešće zadovoljavamo s brojevima za koje je vrlo velika vjerojatnost da su prosti. Važno je napomenuti da su ovi vjerojatnosni testovi puno brži od svih poznatih metoda za dokazivanje prostosti.

Razlog za ovo razlikovanje testiranja i dokazivanja prostosti leži u tome što teoremi, koji u potpunosti karakteriziraju proste brojeve (npr. *Wilsonov teorem*: p je prost ako i samo ako p dijeli $(p-1)!+1$), nisu jednostavni za provjeru. S druge strane, neka važna svojstva prostih brojeva su vrlo jednostavna za provjeru, ali ne karakteriziraju proste brojeve, tj. postoje i neki složeni koji imaju to svojstvo. Tipičan i važan primjer takvog svojstva je *Mali Fermatov teorem* koji kaže da ako je p prost broj, onda za svaki cijeli broj b vrijedi

$$b^p \equiv b \pmod{p}, \tag{3.1}$$

tj. $b^{p-1} \equiv 1 \pmod{p}$ ako p ne dijeli b. Za efikasnost provjere ovog svojstva važna je činjenica da se modularno potenciranje može vrlo efikasno provesti. No, obrat ovog teorema ne vrijedi. Naime, p može biti složen, a da ipak za neki (pa čak i za svaki) b vrijedi relacija (3.1).

Definicija 3.2. Ako je n neparan složen broj, te b cijeli broj takav da je $\mathrm{nzd}(b,n)=1$ i

$$b^{n-1} \equiv 1 \pmod{n}, \tag{3.2}$$

onda kažemo da je n *pseudoprost* u bazi b (ili da je n je psp(b)).

Primjer 3.1. Broj $341 = 11 \cdot 31$ je psp(2), jer je $2^{340} \equiv 32^{68} \equiv (-1)^{68} \equiv 1 \pmod{11}$ i $2^{340} \equiv 32^{68} \equiv 1^{68} \equiv 1 \pmod{31}$, pa je $2^{340} \equiv 1 \pmod{341}$.

Slično, $91 = 7 \cdot 13$ je psp(3), jer je $3^{90} \equiv 27^{30} \equiv (-1)^{30} \equiv 1 \pmod{7}$ i $3^{90} \equiv 27^{30} \equiv 1^{30} \equiv 1 \pmod{13}$, pa je $3^{90} \equiv 1 \pmod{91}$.

Teorem 3.1. *Neka je n neparan složen broj. Tada vrijedi:*

(1) n je pseudoprost u bazi b, gdje je $\mathrm{nzd}(b,n)=1$, ako i samo ako red od b modulo n (tj. najmanji prirodan broj a takav da je $b^a \equiv 1 \pmod{n}$) dijeli $n-1$.

(2) Ako je n pseudoprost u bazama b_1 i b_2, onda je n pseudoprost i u bazama $b_1 \cdot b_2$ i $b_1 \cdot b_2^{-1}$ (gdje b_2^{-1} označava inverz od b_2 modulo n).

(3) Ako n ne zadovoljava (3.2) za neki b, onda n ne zadovoljava (3.2) za barem pola mogućih baza b (brojeva između 1 i n koji su relativno prosti s n).

Dokaz:

(1) Neka je a red od b. Pretpostavimo da a ne dijeli $n-1$, tj. $n-1 = a\cdot q + r$, $0 < r < a$. Tada je

$$b^r = b^{n-1-aq} \equiv b^{n-1}(b^a)^{-q} \equiv 1 \pmod{n},$$

što je u suprotnosti s minimalnošcu od a.

Obrnuto, neka je $n-1 = a \cdot k$. Tada je $b^{n-1} = (b^a)^k \equiv 1 \pmod{n}$.

(2) Iz $b_1^{n-1} \equiv b_2^{n-1} \equiv 1 \pmod{n}$, slijedi $(b_1 b_2)^{n-1} \equiv b_1^{n-1} b_2^{n-1} \equiv 1 \pmod{n}$ i $(b_1 b_2^{-1})^{n-1} \equiv b_1^{n-1} (b_2^{n-1})^{-1} \equiv 1 \pmod{n}$.

(3) Neka je $\{b_1, b_2, \ldots, b_s\}$ skup svih baza za koje vrijedi (3.2), te neka je b neka baza za koju ne vrijedi (3.2). Tada baze $bb_1, bb_2, \ldots, bb_s$ ne zadovoljavaju (3.2), jer ako bi bb_i zadovoljavala (3.2), onda bi po (2) to vrijedilo i za $b = (bb_i)b_i^{-1}$. Prema tome, postoji barem s baza za koje ne vrijedi (3.2), što je i trebalo dokazati.

□

Postavlja se pitanje koliko ima pseudoprostih brojeva u bazi b. Erdős je dokazao da za broj psp(b) brojeva koji su $\leq x$ vrijedi ocjena $o(\pi(x))$, tj. pseudoprosti brojevi su "rjeđi" od prostih brojeva. Pa ipak i njih ima beskonačno mnogo.

Teorem 3.2. *Za svaki prirodan broj $b \geq 2$ postoji beskonačno mnogo pseudoprostih brojeva u bazi b.*

Dokaz: Neka je p bilo koji neparan prost broj koji ne dijeli $b^2 - 1$. Promotrimo broj $n = \frac{b^{2p}-1}{b^2-1}$. Budući da je

$$n = \frac{b^p - 1}{b-1} \cdot \frac{b^p + 1}{b+1},$$

vidimo da je n složen. Iz Malog Fermatovog teorema slijedi $b^{2p} \equiv b^2 \pmod{p}$. Dakle, p dijeli $b^{2p} - b^2 = (n-1)(b^2-1)$. No, kako p ne dijeli $b^2 - 1$, zaključujemo da $p|n-1$. Nadalje, $n - 1 = b^{2p-2} + b^{2p-4} + \cdots + b^2$ je suma od $p-1$ pribrojnika iste parnosti, pa je stoga $n-1$ paran broj. Dakle, $2p|n-1$, pa kako n dijeli $b^{2p} - 1$, slijedi da n mora dijeliti i $b^{n-1} - 1$. Stoga je $b^n \equiv b \pmod{n}$. □

Postojanje pseudoprostih brojeva nam pokazuje da testiranje samo s jednom bazom nije dovoljno da bismo zaključili da je broj prost. Zato možemo pokušati kombinirati više baza. Tako je npr. 341 psp(2), a nije psp(3), dok je 91 psp(3), a nije psp(2). No, broj $561 = 3 \cdot 11 \cdot 17$ je pseudoprost u svakoj bazi. Takvi brojevi se nazivaju *Carmichaelovi brojevi*. Ako je poznata faktorizacija od n, onda je lako ustanoviti je li on Carmichaelov broj. Naime, *Korseltov kriterij* kaže da je n Carmichaelov ako i samo ako je n složen, kvadratno slobodan i za svaki prosti faktor p od n vrijedi da $p-1$ dijeli $n-1$. Odavde neposredno slijedi da n mora biti produkt od barem tri različita prosta broja. Zaista, kako je n kvadratno slobodan, on mora biti produkt različitih prostih brojeva. Ostaje za vidjeti zašto n ne može biti produkt dvaju prostih brojeva. Pretpostavimo da je $n = pq$, $p < q$. Tada je $n - 1 = pq - 1 \equiv p - 1 \not\equiv 0 \pmod{(q-1)}$, što je u suprotnosti s Korseltovim kriterijem.

Poznato je da postoji beskonačno mnogo Carmichaelovih brojeva. Označimo s $C(x)$ broj Carmichaelovih brojeva koji su $\leq x$. Alford, Granville i Pomerance su 1994. godine dokazali da je $C(x) > x^{2/7}$. Erdős je postavio slutnju da za svaki $\varepsilon > 0$ postoji $x_0(\varepsilon)$ takav da je $C(x) > x^{1-\varepsilon}$ za $x \geq x_0(\varepsilon)$.

Postojanje Carmichaelovih brojeva pokazuje važan nedostatak testiranja prostosti na osnovu Malog Fermatova teorema. Sada ćemo pokazati kako se malim modificiranjem testa taj nedostatak može ukloniti.

3.3 Solovay-Strassenov test

Neka je p neparan prost broj. Podsjetimo se da za broj b, koji je relativno prost s p, kažemo da je *kvadratni ostatak* modulo p ako kongruencija $x^2 \equiv b \pmod{p}$ ima rješenja. U protivnom, kažemo da je b kvadratni neostatak. *Legendreov simbol* $(\frac{b}{p})$ se definira ovako: $(\frac{b}{p}) = 1$ ako je b kvadratni ostatak modulo p; $(\frac{b}{p}) = -1$ ako je b kvadratni neostatak modulo p; $(\frac{b}{p}) = 0$ ako p dijeli b.

Neka je n neparan broj i $n = p_1 \cdots p_k$ njegov rastav na (ne nužno različite) proste faktore. Tada se *Jacobijev simbol* $(\frac{b}{n})$ definira sa $(\frac{b}{n}) = (\frac{b}{p_1}) \cdots (\frac{b}{p_k})$. Ako je n prost, onda se Jacobijev i Legendreov simbol podudaraju. Važno svojstvo Legendreovog simbola je Eulerov kriterij:

$$\left(\frac{b}{p}\right) \equiv b^{(p-1)/2} \pmod{p}.$$

Definicija 3.3. Ako je n neparan složen broj, te b cijeli broj takav da je $\mathrm{nzd}(b, n) = 1$ i vrijedi

$$\left(\frac{b}{n}\right) \equiv b^{(n-1)/2} \pmod{n}, \tag{3.3}$$

gdje je $(\frac{b}{n})$ Jacobijev simbol, onda n zovemo *Eulerov pseudoprosti broj u bazi* b (ili da je n epsp(b)) (ponekad se kaže i Euler-Jacobijev pseudoprost).

Kvadriranjem relacije (3.3) dobivamo $b^{n-1} \equiv 1 \pmod{n}$, tj. relaciju (3.2). To znači da je svaki epsp(b) ujedno i psp(b). Obrat ove tvrdnje ne vrijedi.

Primjer 3.2. Vidjeli smo u Primjeru 3.1 da je 91 pseudoprost u bazi 3. Međutim,

$$3^{45} \equiv 729^7 \cdot 27 \equiv 27 \pmod{91},$$

pa 91 nije Eulerov pseudoprost broj u bazi 3. Ipak, 91 jest Eulerov pseudoprost u bazi 10 jer je

$$10^{45} \equiv 1000^{15} \equiv (-1)^{15} \equiv -1 \pmod{91} \quad \text{i} \quad \left(\frac{10}{91}\right) = -1.$$

Neka je $\mathbb{Z}_n^* = \{b \in \mathbb{Z}_n \ : \ \text{nzd}(b,n) = 1\}$. Tada je $|\mathbb{Z}_n^*| = \varphi(n)$. Neka je nadalje

$$P_n = \{b \in \mathbb{Z}_n^* \ : \ \left(\frac{b}{n}\right) = b^{(n-1)/2} \pmod{n}\}.$$

Sljedeći teorem predstavlja osnovu za Solovay-Strassenov test prostosti.

Teorem 3.3. *Za sve neparne složene brojeve n vrijedi $|P_n| \leq \varphi(n)/2$.*

Uočimo važnu razliku izmedu Teorema 3.1.3) i 3.3. Naime, kod Eulerovih pseudoprostih brojeva nema analogona Carmichaelovih brojeva, već za svaki složen broj n, relacija (3.3) nije zadovoljena za barem pola mogućih baza.

Prije dokaza Teorema 3.3, dokažimo dvije leme. Prva lema će nam dati potrebne informacije o kongruencijama oblika $x^m \equiv \pm 1 \pmod{n}$. S $\nu_m(t)$ ćemo označavati najveći cijeli broj k takav da m^k dijeli t.

Lema 3.1. *Neka je m prirodan broj, $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ neparan broj, te neka je*

$$\nu = \min\{\nu_2(p_i - 1) \ : \ i = 1, \ldots, r\} \ \textit{i} \ S = \prod_{i=1}^{r} \text{nzd}(m, \varphi(p_i^{\alpha_i})).$$

Tada vrijedi:

(1) Kongruencija $x^m \equiv 1 \pmod{n}$ ima točno S rješenja.

(2) Kongruencija $x^m \equiv -1 \pmod{n}$ ima rješenja ako i samo ako je $\nu_2(m) < \nu$.

(3) Ako kongruencija $x^m \equiv -1 \pmod{n}$ ima rješenja, onda ih ima točno S.

Dokaz: Neka je g_i primitivni korijen modulo $p_i^{\alpha_i}$. To znači da za svaki $x \in \{1, ..., p_i^{\alpha_i} - 1\}$ koji nije djeljiv s p_i postoji j takav da je $g_i^j \equiv x \pmod{p_i^{\alpha_i}}$. Označimo taj j s $\text{ind}_{g_i} x$. Sada je kongruencija $x^m \equiv c \pmod{n}$ ekvivalentna sustavu kongruencija

$$m \cdot \text{ind}_{g_i} x \equiv \text{ind}_{g_i} c \pmod{\varphi(p_i^{\alpha_i})}, \quad i = 1, \ldots, r.$$

Za $c = 1$ je $\text{ind}_{g_i} 1 = 0$, pa sustav postaje

$$m \cdot \text{ind}_{g_i} x \equiv 0 \pmod{\varphi(p_i^{\alpha_i})}, \quad i = 1, \ldots, r.$$

Ovo su linearne kongruencije i i-ta ima $\text{nzd}(m, \varphi(p_i^{\alpha_i}))$ rješenja. Stoga je ukupan broj rješenja sustava jednak S.

Za $c = -1$ je $\text{ind}_{g_i}(-1) = \frac{\varphi(p_i^{\alpha_i})}{2}$, jer je $g_i^{2\text{ind}_{g_i}(-1)} \equiv (-1)^2 \equiv 1 \pmod{p_i^{\alpha_i}}$, pa gornji sustav postaje

$$m \cdot \text{ind}_{g_i} x \equiv \frac{\varphi(p_i^{\alpha_i})}{2} \pmod{\varphi(p_i^{\alpha_i})}, \quad i = 1, \ldots, r.$$

Koristimo osnovnu činjenicu o linearnim kongruncijama, koja kaže da kongruencija $ax \equiv b \pmod{M}$ ima rješenja ako i samo ako broj $d = \mathrm{nzd}(a, M)$ dijeli b, te ako je ovaj uvjet ispunjen, onda kongruencija ima točno d rješenja modulo M. Ako naš sustav kongruencija ima rješenja, onda zaključujemo da ih ima S, a nužan i dovoljan uvjet za postojanje rješenja je da za svaki i broj $\mathrm{nzd}(m, \varphi(p_i^{\alpha_i}))$ dijeli $\frac{\varphi(p_i^{\alpha_i})}{2}$. Odavde imamo da $(2^{\nu_2(m)}m_1, p_i^{\alpha_i-1} \cdot 2^{\nu_2(p_i-1)} \cdot q_i)$ dijeli $p_i^{\alpha_i-1} \cdot 2^{\nu_2(p_i-1)-1} \cdot q_i$, gdje su m_1 i q_i neparni brojevi. No, ovo je očito ekvivalentno s $\nu_2(m) < \nu_2(p_i - 1)$ za sve $i = 1, \ldots, r$, tj. s $\nu_2(m) < \nu$. □

Lema 3.2. *Neka je n neparan broj, te $p_1, \ldots, p_r$ svi različiti prosti faktori od n. Tada je*

$$|P_n| = \delta_n \cdot \prod_{i=1}^{r} \mathrm{nzd}\left(\frac{n-1}{2}, p_i - 1\right),$$

gdje je $\delta_n \in \{1/2, 1, 2\}$.

Dokaz: Neka je

$$\begin{aligned} K_n^{+,-} &= \{b \in \mathbb{Z}_n^* \,:\, b^{(n-1)/2} \equiv \pm 1 \pmod{n}\}, \\ L_n^{+,-} &= \{b \in \mathbb{Z}_n^* \,:\, \left(\frac{b}{n}\right) = \pm 1\}, \\ M_n^{+,-} &= \{b \in \mathbb{Z}_n^* \,:\, \left(\frac{b}{n}\right) b^{(n-1)/2} \equiv \pm 1 \pmod{n}\}. \end{aligned}$$

Tada je $P_n = M_n^+$. Iz Leme 3.2 imamo: $|K_n^+| = \prod_{i=1}^{r} \mathrm{nzd}((n-1)/2, p_i - 1)$. S druge strane, $M_n^+ = (K_n^+ \cap L_n^+) \cup (K_n^- \cap L_n^-)$. Ako je $K_n^- \cap L_n^- = \emptyset$, onda je $|M_n^+| = |K_n^+ \cap L_n^+|$. Ako je $K_n^- \cap L_n^- \neq \emptyset$ i $b_0 \in K_n^- \cap L_n^-$, onda je preslikavanje $f : K_n^+ \cap L_n^+ \to K_n^- \cap L_n^-$, definirano s $f(b) = bb_0$, bijekcija, pa je $|M_n^+| = 2|K_n^+ \cap L_n^+|$. Na isti način, iz relacije $K_n^+ = (K_n^+ \cap L_n^+) \cup (K_n^+ \cap L_n^-)$ slijedi $|K_n^+ \cap L_n^+| = |K_n^+|$ ako je $K_n^+ \cap L_n^- = \emptyset$, te $|K_n^+ \cap L_n^+| = \frac{1}{2}|K_n^+|$ ako je $K_n^+ \cap L_n^- \neq \emptyset$. Stoga je zaista $|P_n| = |M_n^+| = \delta_n |K_n^+|$, gdje je $\delta_n \in \{1/2, 1, 2\}$. □

Dokaz Teorema 3.3: Neka je $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$. Iz Leme 3.2 slijedi

$$\frac{|P_n|}{\varphi(n)} = \delta_n \prod_{i=1}^{r} \frac{\mathrm{nzd}(\frac{n-1}{2}, p_i - 1)}{p_i^{\alpha_i - 1}(p_i - 1)}. \tag{3.4}$$

Ako je $\alpha_i \geq 2$ za neki i, onda je desna strana od (3.4) $\leq \frac{\delta_n}{3} \leq \frac{2}{3}$. Skup P_n je jezgra homomorfizma $h_n : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$ definiranog s $h_n(a) = (\frac{a}{n})a^{(n-1)/2} \bmod n$, pa je P_n podgrupa od $\mathbb{Z}_n^*$, a zbog $|P_n| < \varphi(n)$ je prava podgrupa od $\mathbb{Z}_n^*$. Stoga je $|P_n| \leq \frac{\varphi(n)}{2}$.

Sada možemo pretpostaviti da je $\alpha_i = 1$ za svaki i, tj. $n = p_1 \cdots p_r$, $r \geq 2$. Pretpostavimo da tvrdnja teorema ne vrijedi. Tada bi bilo $P_n = \mathbb{Z}_n^*$. Neka je

g primitivni korijen modulo p_1 (ili bilo koji kvadratni neostatak modulo p_1). Po Kineskom teoremu o ostatcima, možemo naći $a \in \mathbb{Z}_n^*$ takav da je

$$a \equiv g \pmod{p_1} \quad \text{i} \quad a \equiv 1 \pmod{n/p_1}.$$

Budući da je $\mathbb{Z}_n^* = P_n$, to je $a^{(n-1)/2} \equiv (\frac{a}{n}) \pmod{n}$. Međutim, $(\frac{a}{n}) = (\frac{a}{p_1})(\frac{a}{n/p_1}) = (\frac{g}{p_1}) = -1$. Zato je $a^{(n-1)/2} \equiv -1 \pmod{n/p_1}$, što je u suprotnosti s $a \equiv 1 \pmod{n/p_1}$. □

Korolar 3.1. *Neka je $n \equiv 3 \pmod 4$ složen broj s r različitih prostih faktora. Tada je*

$$|P_n| = \frac{\varphi(n)}{2^{r-1}}.$$

Dokaz: Iz relacije (3.4) imamo:

$$\frac{|P_n|}{\varphi(n)} \leq \delta_n \prod_{i=1}^{r} \frac{(p_i - 1)/2}{p_i - 1} \leq \frac{1}{2^{r-1}}\,.$$

□

Solovay-Strassenov test prostosti:

Neka je n neparan broj za kojeg želimo ustanoviti je li prost ili složen. Odaberimo na slučajan način k brojeva b takvih da je $0 < b < n$. Za svaki od tih b-ova izračunamo obje strane od (3.3). Za računanje $b^{(n-1)/2} \bmod n$ treba $O(\ln^3 n)$ operacija metodom kvadriraj i množi. Za računanje Jacobijevog simbola $(\frac{b}{n})$, koristeći kvadratni zakon reciprociteta, treba $O(\ln^2 n)$ operacija. Ako ova dva broja nisu kongruentna modulo n, onda znamo da je n složen i test staje. Inače uzmemo idući b. Ako n prođe test (3.3) za k različitih b-ova, onda je vjerojatnost da je n složen $\leq \frac{1}{2^k}$. Stoga, ako je k dovoljno velik, n proglašavamo "vjerojatno prostim". U praksi se uzima $k = 50$ ili $k = 100$ ($2^{-50} \approx 10^{-15}$, $2^{-100} \approx 10^{-30}$).

Solovay-Strassenov test je primjer tzv. *Monte Carlo algoritma* koji uvijek daje odgovor, ali on može biti netočan. Drugi tip vjerojatnosnog algoritma je tzv. *Las Vegas algoritam* koji ne daje uvijek odgovor, ali kada ga da, onda je odgovor sigurno točan. Primjer takvog algoritma smo vidjeli kod faktorizacije RSA modula n ako je poznat tajni eksponent d.

3.4 Miller-Rabinov test

Neka je n neparan prirodan broj, $\text{nzd}(b, n) = 1$, te $b^{n-1} \equiv 1 \pmod{n}$. Budući da je $n - 1$ paran, možemo pokušati "vaditi drugi korijen" iz ove kongruencije, tj. računati $b^{(n-1)/2}$, $b^{(n-1)/4}, \ldots$. Pretpostavimo da u i-tom koraku prvi put dobijemo na desnoj strani nešto različito od 1, recimo $b^{(n-1)/2^i} \equiv a \pmod{n}$. Tada, ako je n prost, onda mora biti $a = -1$ jer je $b^{(n-1)/2^{i-1}} \equiv 1$

$(\mathrm{mod}\ n)$, a jedina rješenja kongruencije $x^2 \equiv 1\ (\mathrm{mod}\ n)$, ako je n prost, su $x \equiv \pm 1\ (\mathrm{mod}\ n)$. Dakle, kombinirajući Mali Fermatov teorem sa svojstvom kongruencije $x^2 \equiv 1\ (\mathrm{mod}\ p)$ dobivamo jači zahtjev od onog iz definicije pseudoprostih brojeva.

Definicija 3.4. Neka je n neparan složen broj, te neka je $n-1 = 2^s \cdot t$, gdje je t neparan. Ako za cijeli broj b vrijedi

$$b^t \equiv 1\ \ (\mathrm{mod}\ n) \text{ ili postoji } r < s \text{ takav da je } b^{2^r \cdot t} \equiv -1\ \ (\mathrm{mod}\ n), \quad (3.5)$$

onda kažemo da je n *jak pseudoprost broj u bazi* b (ili da je n spsp(b)).

Ako uvjet (3.5) nije ispunjen za neki b, $0 < b < n$, tada je broj n složen. U tom slučaju broj b nazivamo *svjedok složenosti od* n.

Primjer 3.3. Svaki spsp(b) je ujedno i psp(b). Obrat ne vrijedi. Npr. $n = 341$ je psp(2), ali nije spsp(2). Zaista, $340 = 2^2 \cdot 85$, dok je $2^{85} \equiv 32\ (\mathrm{mod}\ 341)$ i $2^{170} \equiv 1\ (\mathrm{mod}\ 341)$.

Može se pokazati da je svaki spsp(b) ujedno i epsp(b). Obrat ne vrijedi. Npr. $n = 561$ je epsp(2) jer je $2^{280} \equiv (\frac{2}{561}) \equiv 1\ (\mathrm{mod}\ 561)$, ali n nije spsp(2) jer je $2^{280} \equiv 1\ (\mathrm{mod}\ 561)$, a $2^{140} \equiv 67\ (\mathrm{mod}\ 561)$.

Kao primjer jakog pseudoprostog broja navedimo npr. da je 91 spsp(10) jer je $10^{45} \equiv -1\ (\mathrm{mod}\ 91)$.

Teorem 3.4. *Ako je* $n \equiv 3\ (\mathrm{mod}\ 4)$*, onda je* n *spsp(b) ako i samo ako je* n *epsp(b).*

Dokaz: U ovom slučaju je $s = 1$, $t = \frac{n-1}{2}$. Ako je n epsp(b), onda je

$$b^t \equiv b^{(n-1)/2} \equiv \left(\frac{b}{n}\right) \equiv \pm 1 \quad (\mathrm{mod}\ n),$$

pa je n spsp(b).

Neka je sada n spsp(b). To znači da je $b^{(n-1)/2} \equiv \pm 1\ (\mathrm{mod}\ n)$. Budući da je $n \equiv 3\ (\mathrm{mod}\ 4)$, to je $(\frac{\pm 1}{n}) = \pm 1$, pa je

$$\left(\frac{b}{n}\right) = \left(\frac{b \cdot (b^2)^{(n-3)/4}}{n}\right) \equiv \left(\frac{b^{(n-1)/2}}{n}\right) \equiv b^{(n-1)/2} \quad (\mathrm{mod}\ n).$$

□

Pojam jakog pseudoprostog broja uveo je Selfridge 1974. godine. No, pravu snagu testu zasnovanom na njemu daje sljedeći teorem koji su neovisno dokazali Monier i Rabin 1980. godine, a koji nam pokazuje da i u slučaju jakih pseudoprostih brojeva ne postoji analogon Carmichaelovih brojeva, ali i da će pripadni test biti još efikasniji od Solovay-Strassenovog testa jer je broj baza u kojima složen broj može biti jak pseudoprost broj manji od broja baza u kojima može biti Eulerov pseudoprost broj.

Teorem 3.5. *Neka je n neparan složen broj. Tada je n jak pseudoprost broj u bazi b za najviše $(n-1)/4$ baza b, $0 < b < n$.*

Dokaz:

1. SLUČAJ: n nije kvadratno slobodan.

Neka je $n = p^2 q$, gdje je p prost. Ako je n spsp(b), onda vrijedi $b^{n-1} \equiv 1 \pmod{n}$. Tada je i $b^{n-1} \equiv 1 \pmod{p^2}$. Po Lemi 3.1, ova kongruencija ima $d = \operatorname{nzd}(p(p-1), n-1)$ rješenja. Budući da $p|n$, to p ne dijeli $n-1$. Zato je $d \leq p-1$. Stoga je broj baza b za koje je n psp(b) manji ili jednak

$$d \cdot q \leq (p-1)q = \frac{(p^2-1)q}{p+1} \leq \frac{n-1}{4}\,.$$

Ovdje jednakost vrijedi samo za $n = 9$.

2. SLUČAJ: $n = p \cdot q$, gdje su p i q različiti prosti brojevi.

Neka je $p-1 = 2^u \cdot v$, $q-1 = 2^w \cdot z$, gdje su v i z neparni, te $u \leq w$. Neka je $n-1 = 2^s t$, gdje je t neparan. Pretpostavimo da je $b^t \equiv 1 \pmod{n}$. Po Lemi 3.1, takvih baza ima $\operatorname{nzd}(t,v) \cdot \operatorname{nzd}(t,z) \leq vz$. Pretpostavimo sada da je $b^{2^r t} \equiv -1 \pmod{n}$ za neki r, $0 \leq r < s$. Po Lemi 3.1, ova kongruencija ima rješenja ako i samo ako je $r < u$, a broj rješenja je $2^r \operatorname{nzd}(t,v) \cdot 2^r \operatorname{nzd}(t,z) \leq 4^r vz$. Budući da je $n-1 > \varphi(n) = 2^{u+w} vz$, slijedi da prirodnih brojeva b, $0 < b < n$, za koje je n spsp(b) ima najviše

$$vz + \sum_{r=0}^{u-1} 4^r vz = vz\Big(1 + \frac{4^u - 1}{3}\Big) < (n-1) \cdot 2^{-u-w} \cdot \frac{4^u + 2}{3}\,.$$

Ako je $u < w$, onda je desna strana ove nejednakosti

$$\leq (n-1) \cdot 2^{-2u-1}\Big(\frac{2}{3} + \frac{4^u}{3}\Big) \leq (n-1) \cdot \Big(\frac{1}{8} \cdot \frac{2}{3} + \frac{1}{6}\Big) = \frac{n-1}{4}\,.$$

Ako je $u = w$, onda barem jedna od nejednakosti $\operatorname{nzd}(t,v) \leq v$, $\operatorname{nzd}(t,z) \leq z$ mora biti stroga, jer bismo inače imali $0 \equiv 2^s t \equiv pq - 1 \equiv q - 1 \pmod{v}$, pa bi iz $v|q-1 = 2^w z$ slijedilo da $v|z$. Analogno bismo dobili da $z|v$, što bi značilo da je $v = z$ i $p = q$, što je kontradikcija.

Dakle, u gornjim ocjenama možemo zamijeniti vz s $\frac{vz}{3}$. To dovodi do sljedeće gornje ograde za broj baza b za koje je n spsp(b):

$$(n-1) \cdot \frac{1}{3} \cdot 2^{-2u}\Big(\frac{2}{3} + \frac{4^u}{3}\Big) \leq (n-1)\Big(\frac{1}{18} + \frac{1}{9}\Big) = \frac{n-1}{6} < \frac{n-1}{4}\,.$$

3. SLUČAJ: $n = p_1 p_2 \cdots p_k$, gdje je $k \geq 3$, a p_i-ovi su različiti prosti brojevi.

Neka je $p_j - 1 = 2^{s_j} t_j$, t_j neparan. Postupimo kao u 2. slučaju. Možemo pretpostaviti da je $s_1 \leq s_j$, $\forall j$. Dobivamo sljedeću gornju ogradu za broj

baza b takvih da je n spsp(b):

$$\begin{aligned}
&(n-1)2^{-s_1-s_2-\cdots-s_k}\Big(1+\frac{2^{ks_1}-1}{2^k-1}\Big) \le (n-1)2^{-ks_1}\Big(\frac{2^k-2}{2^k-1}+\frac{2^{ks_1}}{2^k-1}\Big)\\
&= (n-1)\Big(2^{-ks_1}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1}\Big) \le (n-1)\Big(2^{-k}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1}\Big)\\
&= (n-1)\cdot\frac{1}{2^{k-1}} \le \frac{n-1}{4}\,.
\end{aligned}$$

□

Miller-Rabinov test prostosti. Neka je n neparan broj za kojeg želimo ustanoviti je li prost ili složen. Neka je $n-1=2^s t$, gdje je t neparan. Na slučajan način izaberemo b, $0<b<n$. Izračunamo $b^t \bmod n$. Ako dobijemo ± 1, zaključujemo da je n prošao test, te biramo sljedeći b. U protivnom, uzastopno kvadriramo b^t modulo n sve dok ne dobijemo rezultat -1. Ako dobijemo -1, onda je n prošao test. Ako nikad ne dobijemo -1, tj. ako dobijemo da je $b^{2^{r+1}t} \equiv 1 \pmod n$, ali $b^{2^r t} \not\equiv -1 \pmod n$, onda sigurno znamo da je n složen. Ako n prođe test za k b-ova, onda je vjerojatnost da je n složen $\le \frac{1}{4^k}$.

Npr. za $k=20$ je vjerojatnost da je n složen manja od 10^{-12}. Tako dobiveni "vjerojatno prosti brojevi" se nazivaju još i "industrijski prosti brojevi" (koliko smo za njih spremni "platiti", tj. koliko veliki k uzmemo, toliku "kvalitetu" možemo i očekivati). Poznato je da ne postoji niti jedan broj manji od 10^{12} koji je istovremeno spsp(b) za $b=2,3,5,7$ i 11. Napomenimo još da se ocjena iz Teorema 3.5 može značajno poboljšati za velike brojeve n. Tako je vjerojatnost da je 500-bitni broj, koji prođe samo jedan test, složen manja od $1/4^{28}$.

Složenost jednog Miller-Rabinova testa je $O(\ln^3 n)$. Naime, $b^t \bmod n$ se može izračunati u $O(\ln^3 n)$ bitnih operacija, a potom za računanje $b^{2t}, b^{4t}, \ldots, b^{2^{s-1}t}$ uzastopnim kvadriranjem trebamo također $O(\ln^3 n)$ bitnih operacija.

Uz pretpostavku da vrijedi proširena Riemannova slutnja (ERH), Miller-Rabinov test postaje polinomijalni deterministički algoritam za dokazivanje prostosti. Naime, može se pokazati da ako je n složen broj, onda uz pretpostavku da vrijedi ERH postoji barem jedna baza $b < 2\ln^2 n$ za koju ne vrijedi (3.5). Dakle, uz pretpostavku da vrijedi ERH, složenost ovog algoritma je $O(\ln^5 n)$. To je i bio originalni Millerov test iz 1976. godine. Dokaz se zasniva na sljedećem teoremu.

Teorem 3.6 (Ankeny). *Pretpostavimo da vrijedi ERH. Tada za svaki $d \in \mathbb{N}$ i Dirichletov karakter $\chi \neq 1$ modulo d postoji $n < 2\ln^2 d$ takav da je $\chi(n) \neq 1$.*

Gornja tvrdnja se dobije primjenom Ankenyjeva teorema na Dirichletove karaktere $\chi_1(m) = \left(\frac{m}{p_1 p_2}\right)$ i $\chi_2 = \left(\frac{m}{p_2}\right)$, gdje su p_1, p_2 neparni prosti faktori od n.

Postoje i druge vrste pseudoprostih brojeva, te odgovarajući testovi prostosti zasnovani na njima. Spomenut ćemo još samo Lucasove pseudoproste brojeve. Neka su α i β korijeni polinoma $x^2 - ax + b = 0$, $a, b \in \mathbb{Z} \setminus \{0\}$. Definiramo nizove $U_k(a,b) = \frac{\alpha^k - \beta^k}{\alpha - \beta}$, $V_k(a,b) = \alpha^k + \beta^k$. Ovi nizovi se nazivaju *Lucasovi nizovi*. Za $a = 1$, $b = -1$, U_k su Fibonaccijevi brojevi, a V_k su (obični) Lucasovi brojevi. Može se pokazati da za proste brojeve p, takve da p ne dijeli $2bD$, gdje je $D = a^2 - 4b$, vrijedi

$$U_{\delta(p)} \equiv 0 \pmod{p}, \tag{3.6}$$

gdje je $\delta(p) = p - (\frac{D}{p})$. (Za Fibonaccijeve brojeve to znači da $p|F_{p-1}$ ako je $p \equiv \pm 1 \pmod{10}$, dok $p|F_{p+1}$ ako je $p \equiv \pm 3 \pmod{10}$.) Stoga se ovo svojstvo prostih brojeva može iskoristiti za definiciju nove vrste pseudoprostih brojeva, te za konstrukciju testa prostosti zasnovanog na njima. Ako za neparan složen broj n vrijedi $U_{\delta(n)} \equiv 0 \pmod{n}$, onda kažemo da je n *Lucasov pseudoprost broj s parametrima* a, b (n je lpsp(a,b)). Pokazuje se u praksi da je kombinacija testova s jakim pseudoprostim brojevima i Lucasovim pseudoprostim brojevima jako dobra i vjeruje se da broj koji prođe po jedan test sa spsp i lpsp, s prikladno odabranim parametrima, mora biti prost.

3.5 Polinomijalni AKS algoritam za dokazivanje prostosti

Ako broj n prođe nekoliko dobrih testova prostosti (npr. Miller-Rabinov test za nekoliko različitih baza), onda možemo biti prilično sigurni da je n prost. Međutim, ti testovi nam ne daju *dokaz* da je n prost. Što se tiče relevantnosti ovog problema za primjene u kriptografiji, treba razlikovati dva različita načina na koje se pojavljuje potreba za velikim prostim brojevima. Npr. kod izbora tajnih prostih brojeva p i q za RSA kriptosustav, želimo što brže generirati takve brojeve i tu se zadovoljavamo s time da je vrlo velika vjerojatnost da su prosti. S druge strane, kod izbora polja koje će se koristiti za šifriranje u npr. ElGamalovom kriptosustavu, radi se o prostom broju koji će se preporučiti kao standard za uporabu možda i na nekoliko godina, pa tu želimo biti sigurni (imati dokaz) da je broj stvarno prost. Sada ćemo reći nešto o metodama kojima se može dokazati da je dani broj prost.

Sljedeći teorem nam omogućava dokazivanje prostosti broja n ako je poznata faktorizacija broja $n - 1$.

Teorem 3.7 (Pocklington). *Neka je s djelitelj od $n - 1$ koji je veći od $\sqrt{n}$. Pretpostavimo da postoji prirodan broj a takav da vrijedi*

$$a^{n-1} \equiv 1 \pmod{n},$$

$$\mathrm{nzd}(a^{(n-1)/q} - 1, n) = 1 \quad \textit{za svaki prosti djelitelj } q \textit{ od } s.$$

Tada je n prost.

Dokaz: Pretpostavimo suprotno, tj. da je n složen. Tada on ima prosti faktor $p \leq \sqrt{n}$. Stavimo $b = a^{(n-1)/s}$. Tada je

$$b^s \equiv a^{n-1} \equiv 1 \pmod{n},$$

pa je i $b^s \equiv 1 \pmod{p}$. Tvrdimo da je s red od b modulo p. Zaista, pretpostavimo da za neki djelitelj q od s vrijedi $b^{s/q} \equiv 1 \pmod{p}$. Tada bi p dijelio n i $b^{s/q} - 1$, tj. $a^{(n-1)/q} - 1$, što je u suprotnosti s pretpostavkom da su n i $a^{(n-1)/q} - 1$ relativno prosti. Kako je iz Malog Fermatova teorema $b^{p-1} \equiv 1 \pmod{p}$, zaključujemo da s dijeli $p - 1$. No, to je nemoguće budući da je $s > \sqrt{n}$, a $p \leq \sqrt{n}$. □

Da bismo dokazali prostost broja n pomoću Pocklingtova teorema, moramo poznavati barem djelomičnu faktorizaciju broja $n - 1$. No, faktorizacija velikih brojeva je općenito težak problem. Postoje metode za dokazivanje prostosti koje se zasnivaju na faktorizaciji od $n+1$, umjesto od $n-1$. Spomenimo samo *Lucas-Lehmerovu metodu* za dokazivanje prostosti Mersenneovih brojeva. Brojevi $M_p = 2^p - 1$, gdje je p prost, nazivaju se *Mersenneovi brojevi*. Neki Mersennovi brojevi su prosti, kao npr. $M_7 = 127$, a neki su složeni, kao npr. $M_{11} = 2047 = 23 \cdot 89$. Slutnja je da prostih Mersennovih brojeva ima beskonačno mnogo.

Teorem 3.8 (Lucas-Lehmer)**.** *Neka je niz* (v_k) *zadan sa*

$$v_0 = 4, \quad v_{k+1} = v_k^2 - 2.$$

Neka je p *neparan prost broj. Tada je* $M_p = 2^p - 1$ *prost ako i samo ako* M_p *dijeli* v_{p-2}*.*

Najveći poznati prosti Mersennov broj je $M_{82589933}$. To je ujedno i najveći danas poznati prosti broj (ima $24\,862\,048$ znamenaka; otkriven je 2018. godine u sklopu projekta *Great Internet Mersenne Prime Search* (GIMPS)).

Kao što smo već napomenuli, problem s primjenom Pocklingtonova teorema je u tome što zahtjeva (djelomičnu) faktorizaciju broja $n - 1$. Ovaj broj $n-1$ se može shvatiti kao red grupe $\mathbb{Z}_n^*$ (ako je n prost). Jedna od ideja kako riješiti ovaj problem je zamjena grupe $\mathbb{Z}_n^*$ s grupom $E(\mathbb{Z}_n)$, gdje je E neka eliptička krivulja nad $\mathbb{Z}_n$. Naime, kod mogućih redova grupe $E(\mathbb{Z}_n)$ imamo veću fleksibilnost, pa se možemo nadati da ćemo naći eliptičku krivulju čiji će red biti lako faktorizirati. Ideju o korištenju eliptičkih krivulja za dokazivanje prostosti su uveli Goldwasser i Killian 1986. godine.

Godine 2002. Agrawal, Kayal i Saxena, znanstvenici s *Indian Institute of Technology Kanpur*, pronašli su prvi polinomijalni algoritam za dokazivanje prostosti, po njima nazvan *AKS algoritam* (članak su objavili 2004. godine u jednom od najprestižnijih matematičkih časopisa "Annals of Mathematics"). Dakle, dokazali su da problem odluke "Je li broj n prost?" pripada klasi **P**, čime su riješili dugogodišnji otvoreni problem. Prije toga bio

je poznat Adleman-Pomerance-Rumelyjev algoritam iz 1983. godine (njihov članak je također bio objavljen u "Annals of Mathematics"), čija je složenost $O((\ln n)^{c \ln\ln\ln n})$, dakle "skoro polinomijalna". Kao i većina algoritama za testiranje ili dokazivanje prostosti, i AKS algoritam se zasniva na Malom Fermatovu teoremu. Točnije, polazište mu je sljedeća lema.

Lema 3.3. *Neka je $a \in \mathbb{Z}$, $n \in \mathbb{Z}$, $n \geq 2$ i $\operatorname{nzd}(a, n) = 1$. Tada je broj n prost ako i samo ako vrijedi*

$$(X + a)^n \equiv X^n + a \pmod{n}, \tag{3.7}$$

tj. ako i samo ako su odgovarajući koeficijenti polinoma na lijevoj i desnoj strani kongruencije (3.7) kongruentni modulo n.

Dokaz: Za $0 < i < n$ je koeficijent od X^i u polinomu $((X+a)^n - (X^n+a))$ jednak $\binom{n}{i} a^{n-1}$, dok je slobodni član jednak $a^n - a$. Ako je n prost, onda je $\binom{n}{i} \equiv 0 \pmod{n}$ i $a^n - a \equiv 0 \pmod{n}$, pa vrijedi (3.7).

Pretpostavimo sada da je n složen. Neka je q neki prosti faktor od n i neka $q^k \| n$, tj. neka je k najveća potencija od q koja dijeli n. Promotrimo binomni koeficijent

$$\binom{n}{q} = \frac{n(n-1)\cdots(n-q+1)}{1 \cdot 2 \cdots q}.$$

Vidimo da $q^{k-1} \| \binom{n}{q}$. Nadalje je $\operatorname{nzd}(q, a^{n-q}) = 1$. Zaključujemo da koeficijent od X^q nije djeljiv s q^k, pa stoga nije kongruentan 0 modulo n. □

Doslovna primjena prethodne leme neće dati efikasan algoritam za dokazivanje prostosti jer bismo trebali izračunati n koeficijenata na lijevoj strani od (3.7). Jednostavan način za reduciranje broja koeficijenata koje treba izračunati jest da se obje strane kongruencije (3.7) reduciraju modulo $X^r - 1$, za prikladno odabrani mali broj r. Dakle, provjerava se vrijedi li

$$(X + a)^n \equiv X^n + a \pmod{X^r - 1,\ n}. \tag{3.8}$$

Ovdje nam oznaka $f(X) \equiv g(X) \pmod{h(X),\ n}$ znači da je $f(X) = g(X)$ u prstenu $\mathbb{Z}_n[X]/(h(X))$.

Iz Leme 3.3 je jasno da ako je n prost, onda (3.8) vrijedi za sve a i r. Pokazuje se da vrijedi i djelomični obrat, tj. za prikladno odabran r vrijedi: ako je (3.8) zadovoljeno za nekoliko a-ova, onda n mora biti potencija nekog prostog broja. Pokazuje se da se i broj a-ova i veličina od r mogu ograničiti s polinomom u $\ln n$, pa se na taj način dobiva polinomijalni algoritam za dokazivanje prostosti.

Za dokazivanje korektnosti algoritma, treba pokazati da broj n koji prođe sve testove u 5. koraku, mora biti prost. To nećemo dokazivati, već samo recimo da dokaz koristi svojstva ciklotomskih polinoma (korijeni su im primitivni korijeni iz jedinice) nad konačnim poljima.

Algorithm 19 AKS algoritam

1: if (n je potencija prirodnog broja) then return n je složen
2: nađi r takav da je $o_r(n) > \ln^2 n$
3: if ($1 < \mathrm{nzd}(a, n) < n$ za neki $a \leq r$) then return n je složen
4: if ($n \leq r$) then return n je prost
5: for ($1 \leq a \leq \lfloor 2\sqrt{\varphi(r)} \ln n \rfloor$) do
6: if ($(X + a)^n \not\equiv X^n + a \pmod{X^r - 1,\ n}$) then
7: return n je složen
8: return n je prost

Recimo nešto o složenosti AKS algoritma. Najprije trebamo testirati je li n k-ta potencija nekog prirodnog broja, za $k = 2, 3, \ldots, \lfloor \log_2 n \rfloor$. Za svaku konkretnu potenciju k to se može efikasno napraviti modifikacijom Newtonove metode, kao što smo već bili pokazali za slučaj $k = 2$ u Poglavlju 1.8. Na taj način se ovaj test može obaviti u $O(\ln^4 n)$ operacija.

Za složenost preostalog dijela algoritma, ključna je polinomijalna ocjena za veličinu najmanjeg broja r takvog da za red od n modulo r vrijedi $o_r(n) > \ln^2 n$. Najprije dokažimo jedan pomoćni rezultat. S $[a_1, \ldots, a_m]$ označavat ćemo najmanji zajednički višekratnik brojeva $a_1, \ldots, a_m$.

Lema 3.4. *Neka je* $d_n = \mathrm{nzv}(1, 2, \ldots, n)$. *Tada za* $n \geq 7$ *vrijedi* $d_n \geq 2^n$.

Dokaz: Za $1 \leq m \leq n$ promotrimo integral

$$I(m,n) = \int_0^1 x^{m-1}(1-x)^{n-m}dx = \int_0^1 \sum_{r=0}^{n-m} \binom{n-m}{r}(-1)^r x^{m+r-1}$$
$$= \sum_{r=0}^{n-m}(-1)^r \binom{n-m}{r}\frac{1}{m+r}\,.$$

Odavde je jasno da je $d_n \cdot I(m,n)$ cijeli broj. S druge strane, parcijalnom se integracijom dobije

$$I(m,n) = \frac{1}{m\binom{n}{m}}\,.$$

Dakle, $m\binom{n}{m}$ dijeli d_n, za svaki $m = 1, 2, \ldots, n$. Posebno,

$$(n+1)\binom{2n+1}{n+1} = (2n+1)\binom{2n}{n}$$

dijeli d_{2n+1}, dok $n\binom{2n}{n}$ dijeli d_{2n}, pa stoga dijeli i d_{2n+1}. Zaključujemo da broj $n(2n+1)\binom{2n}{n}$ dijeli d_{2n+1}, pa je

$$d_{2n+1} \geq n(2n+1)\binom{2n}{n}\,.$$

Budući da je $\binom{2n}{n}$ najveći pribrojnik u razvoju $(1+1)^{2n} = \sum_{i=0}^{2n} \binom{2n}{i}$, slijedi da je $(2n+1)\binom{2n}{n} \geq 2^{2n}$. Dakle, dokazali smo da je

$$d_{2n+1} \geq n \cdot 2^{2n}.$$

Odavde za $n \geq 2$ dobivamo $d_{2n+1} \geq 2^{2n+1}$, a za $n \geq 4$ dobivamo $d_{2n+2} \geq d_{2n+1} \geq 2^{2n+2}$. □

Lema 3.5 (prema korekciji iz 2019). *Postoji* $r \leq \max\{4, \lceil \ln^5 n \rceil$ *takav da je* $o_r(n) > \ln^2 n$.

Dokaz: Za $n = 2$, možemo uzeti da je $r = 3$. Pretpostavimo da je $n > 2$. Tada je $\lceil \ln^5 n \rceil > 10$ pa možemo primijeniti Lemu 3.4. Označimo $B = \lceil \ln^5 n \rceil$. Neka je s najmanji prirodan broj koji ne dijeli

$$n^{\lfloor \ln B \rfloor} \prod_{i=1}^{\lfloor \ln^2 n \rfloor} (n^i - 1).$$

Imamo da je

$$n^{\lfloor \ln B \rfloor} \prod_{i=1}^{\lfloor \ln^2 n \rfloor} (n^i - 1) < n^{\lfloor \ln B \rfloor + \frac{1}{2} \ln^2 n (\ln^2 n - 1)} < n^{\frac{1}{2} \ln^4 n} < 2^{\ln^5 n} \leq 2^B.$$

Prema Lemi 3.4 je

$$\operatorname{nzv}(1, 2, \ldots, B) \geq 2^B.$$

Zato je $s \leq B$.

Definirajmo sada r sa

$$r = \frac{s}{\operatorname{nzd}(s, n^{\lfloor \ln B \rfloor})}.$$

Tada je r relativno prost s n, te zbog izbora od s, r ne dijeli produkt

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

Stoga r ne dijeli niti jedan od $n^i - 1$ za $1 \leq i \leq \lfloor \ln^2 n \rfloor$, a to znači da je $o_r(n) > \ln^2 n$. □

Koristeći Lemu 3.5 možemo ocijeniti broj operacija potrebnih za nalaženje broja r s traženim svojstvom. Za konkretni r, provjeravamo relaciju $n^k \not\equiv 1 \pmod{r}$ za $k \leq 4 \ln^2 n$. Za to nam treba $O(\ln^2 n)$ množenja modulo r. Kako treba provjeriti najviše $O(\ln^5 n)$ r-ova, slijedi da 2. korak algoritma ima složenost $O(\ln^{7+\varepsilon} n)$.

U 3. koraku, r puta računamo najveći zajednički djelitelj dvaju brojeva. Složenost ovog dijela je $O(r \ln^2 n) = O(\ln^7 n)$. Složenost 4. koraka je $O(\ln n)$.

U 5. koraku imamo $\lfloor 2\sqrt{\varphi(r)} \ln n \rfloor = O(\sqrt{r} \ln n) = O(\ln^{3.5} n)$ provjera. Svaka provjera zahtijeva $O(\ln n)$ množenja polinoma stupnja r s koeficijentima veličine $O(\ln n)$. Stoga se svaka provjera može izvršiti u $O(r^2 \ln^3 n) = O(\ln^{13} n)$ operacija. Konačno, dobivamo da je ukupan broj operacija u 5. koraku, a također i u cijelom algoritmu $O(\ln^{16.5} n)$.

Ova se ocjena može poboljšati preciznijom analizom gornje ograde za r, koja se može dobiti korištenjem preciznijih rezultata o distribuciji prostih brojeva (umjesto Leme 3.4). Ako se još k tome, umjesto "školskog množenja", koriste algoritmi za brzo množenje, dobiva se ocjena $O(\ln^{7.5+\varepsilon} n)$.

Poglavlje 4

Metode faktorizacije

Ako prirodan broj n ne prođe neki od testova prostosti, onda znamo da je n sigurno složen. Međutim, ti nam testovi uglavnom ne daju niti jedan netrivijalni faktor od n. Stoga se postavlja pitanje kako naći netrivijalni faktor velikog složenog broja. To se smatra teškim problemom i na njegovoj su težini zasnovani neki od najvažnijih kriptosustava s javnim ključem.

Metode faktorizacije možemo podijeliti na opće i specijalne. Kod općih metoda očekivani broj operacija ovisi samo o veličini broja n, dok kod specijalnih ovisi također i o svojstvima faktora od n.

Naivna metoda faktorizacije broja n jest dijeljenje broja n sa svim prostim brojevima $\leq \sqrt{n}$. Broj potrebnih dijeljenja je u najlošijem slučaju oko $\frac{2\sqrt{n}}{\ln n}$, pa je složenost ove metode $O(\sqrt{n}\ln n)$. Kod ove smo ocjene pretpostavili da nam je dostupna tablica svih prostih brojeva $\leq \sqrt{n}$. U protivnom, dijelili bismo s 2, te sa svim neparnim brojevima, ili samo s neparnim brojevima koji zadovoljavaju određene kongruencije (npr. $\equiv 1, 5 \pmod 6$ ili $\equiv 1, 7, 11, 13, 17, 19, 23, 29 \pmod{30}$). U svakom slučaju, ova metoda je vrlo neefikasna za velike n-ove. Međutim, dobro ju je koristiti u kombinaciji s boljim metodama faktorizacije, za uklanjanje eventualnih malih faktora od n.

4.1 Pollardova ρ metoda

Jedna od najjednostavnijih metoda faktorizacije čija je složenost bolja od $O(\sqrt{n})$ je *Pollardova ρ metoda* iz 1975. godine. Ideja za nalaženje faktora p broja n je sljedeća:

1. Konstruiramo niz (x_i) cijelih brojeva koji je periodičan modulo p.
2. Nađemo i, j takve da je $x_i \equiv x_j \pmod p$.
3. Odredimo p kao $\mathrm{nzd}(x_i - x_j, n)$.

Niz (x_i) se konstruira pomoću preslikavanja $f : \mathbb{Z}_n \to \mathbb{Z}_n$. Od tog se preslikavanja traži određena "slučajnost". Pokazuje se da zbog toga f ne smije biti linearni polinom. No, kvadratni polinomi, npr. $f(x) = x^2 - 1$, su sasvim dobar izbor. Nadalje, izaberemo vrijednost od x_0 (npr. $x_0 = 2$), te računamo iterirane vrijednosti funkcije f:

$$x_1 = f(x_0), \quad x_2 = f(f(x_0)), \ldots.$$

Općenito, $x_{i+1} = f(x_i)$. Jasno je da u beskonačnom nizu $x_0, x_1, x_2, \ldots$, čiji elementi poprimaju samo konačno mnogo vrijednosti, mora doći do ponavljanja. No, iz definicije niza slijedi da ako je $x_i = x_j$, onda je $x_{i+k} = x_{j+k}$ za $k = 0, 1, 2, \ldots$. Stoga je niz (x_i) periodičan počevši od nekog mjesta. Odavde dolazi naziv ρ metoda, jer "rep" od ρ označava pretperiod, a "okrugli dio" od ρ označava čisto periodični dio niza.

Postavlja se pitanje kada možemo očekivati prvo ponavljanje u nizu (x_i). Pitanje možemo preformulirati tako da pitamo koliko velik treba biti broj k da bi između slučajno izabranih k cijelih brojeva postojala bar dva broja koja su međusobno kongruentna modulo p, s vjerojatnošću većom od 1/2. Vjerojatnost da je k slučajno izabranih brojeva međusobno nekongruentno modulo p jednaka je

$$\begin{aligned}\Big(\frac{p-1}{p}\Big)\Big(\frac{p-2}{p}\Big)\cdots\Big(\frac{p-k+1}{p}\Big) &= \Big(1-\frac{1}{p}\Big)\Big(1-\frac{2}{p}\Big)\cdots\Big(1-\frac{k-1}{p}\Big)\\ &\approx \Big(1-\frac{k}{2p}\Big)^{k-1} \approx e^{-\frac{k^2}{2p}}.\end{aligned}$$

Iz uvjeta da je $e^{-\frac{k^2}{2p}} \approx 1/2$, dobivamo $k \approx \sqrt{2p\ln 2} \approx 1.18\sqrt{p}$. Ova činjenica, da je očekivani broj koraka puno manji od p, naziva se *paradoks rođendana*. Originalni paradoks rođendana kaže da u društvu od barem 23 osobe, s vjerojatnošću većom od 50 %, postoje bar dvije osobe koje imaju rođendan istog dana.

Ako bismo faktor od n tražili tako da računamo $\mathrm{nzd}(x_i - x_j, n)$ za sve i, j, to bi bilo vrlo neefikasno. Puno efikasnije je računati samo $\mathrm{nzd}(x_{2i} - x_i, n)$. Naime, ako je $x_i \equiv x_j \pmod p$, onda za $t = j - i$, $m = t \cdot \lceil \frac{j}{t} \rceil$ vrijedi $x_m \equiv x_{m+t} \equiv x_{m+2t} \equiv \cdots \equiv x_{m+m} \pmod p$. Ovdje je važno uočiti da za računanje $y_i = x_{2i}$ ne treba računati međuvrijednosti $x_{i+1}, x_{i+2}, \ldots, x_{2i-1}$. Vrijednosti x_i, x_{2i} se računaju simultano:

$$\begin{aligned}x_i &= f(x_{i-1}) \bmod n,\\ y_i &= f(f(y_{i-1})) \bmod n.\end{aligned}$$

Primjer 4.1. *Faktorizirajmo broj* $n = 1817$.

Uzmimo $f(x) = x^2 - 1$, $x_0 = 2$. Imamo:

$$x_1 = 2^2 - 1 = 3,\ y_1 = 3^2 - 1 = 8,\ \mathrm{nzd}(y_1 - x_1, n) = \mathrm{nzd}(5, 1817) = 1;$$
$$x_2 = 8,\ y_2 = f(63) = 334,\ \mathrm{nzd}(y_2 - x_2, n) = \mathrm{nzd}(326, 1817) = 1;$$
$$x_3 = 63,\ y_3 = f(334^2 - 1) \bmod n = 1312,\ \mathrm{nzd}(y_3 - x_3, n) = \mathrm{nzd}(1249, 1817) = 1;$$
$$x_4 = 334,\ y_4 = f(1312^2 - 1) \bmod n = 459,\ \mathrm{nzd}(y_4 - x_4, n) = \mathrm{nzd}(125, 1817) = 1;$$
$$x_5 = 718,\ y_5 = f(459^2 - 1) \bmod n = 1195,\ \mathrm{nzd}(y_5 - x_5, n) = \mathrm{nzd}(477, 1817) = 1;$$
$$x_6 = 1312,\ y_6 = f(1195^2 - 1) \bmod n = 873,\ \mathrm{nzd}(y_6 - x_6, n) = \mathrm{nzd}(439, 1817) = 1;$$
$$x_7 = 644,\ y_7 = f(873^2 - 1) \bmod n = 1172,\ \mathrm{nzd}(y_7 - x_7, n) = \mathrm{nzd}(528, 1817) = 1;$$
$$x_8 = 459,\ y_8 = f(1172^2 - 1) \bmod n = 1126,\ \mathrm{nzd}(y_8 - x_8, n) = \mathrm{nzd}(667, 1817) = 23.$$

Dakle, 23 je djelitelj od 1817. Zaista, $1817 = 23 \cdot 79$. ♢

Očekivani broj operacija za nalaženje faktora p broja n Pollardovom ρ metodom je $O(\sqrt{p}\ln^2 n)$. U najlošijem slučaju, kada je $p = O(\sqrt{n})$, dobivamo složenost $O(n^{1/4}\ln^2 n)$, što je znatno bolje od običnog dijeljenja, ali je još uvijek eksponencijalna složenost. Ipak, važno je uočiti da složenost algoritma ovisi o najmanjem faktoru od n. Stoga se može reći da ova metoda spada u specijalne metode.

Malom modifikacijom ove metode su Brent i Pollard 1980. godine uspjeli faktorizirati osmi Fermatov broj

$$2^8 + 1 = 1238926361552897 \cdot p_{63},$$

gdje je p_{63} prost broj sa 63 znamenke.

Spomenimo još da se vrlo slična ideja koristi i u ρ algoritmu za računanje diskretnog logaritma, koji je jedan od najboljih poznatih algoritama za problem diskretnog logaritma u općoj konačnoj Abelovoj grupi.

4.2 Pollardova $p - 1$ metoda

Pollardova $p - 1$ *metoda* iz 1974. godine spada u specijalne metode faktorizacije. Njezino polazište je ponovno Mali Fermatov teorem. Neka je n složen broj koji želimo faktorizirati, te neka je p neki njegov prosti faktor. Tada je $a^{p-1} \equiv 1 \pmod{p}$ za $\mathrm{nzd}(a, p) = 1$. Štoviše, vrijedi $a^m \equiv 1 \pmod{p}$ za svaki višekratnik od $p - 1$. Ako nađemo m, onda nam $\mathrm{nzd}(a^m - 1, n)$ daje faktor (nadamo se netrivijalni) od n. No, pitanje je kako naći višekratnik od $p - 1$ kad ne znamo p. To možemo efikasno napraviti u slučaju kada broj $p - 1$ ima samo male proste faktore. Za prirodan broj kažemo da je B-*gladak* ako su mu svi prosti faktori $\leq B$. Pretpostavimo dodatno da su sve potencije prostih brojeva, koje dijele $p - 1$, manje ili jednake B. Tada za m možemo uzeti najmanji zajednički višekratnik brojeva $1, 2, \ldots, B$. Za ovako odabrani

m, broj operacija za računanje $a^m \bmod n$ je $O(B \ln B \ln^2 n + \ln^3 n)$. U najgorem slučaju, a to je kada je broj $\frac{p-1}{2}$ prost, ova metoda nije ništa bolja od običnog dijeljenja.

Primjer 4.2. Neka je $n = 846631$. Izaberimo $B = 8$ i $a = 2$. Tada je $m = 2^3 \cdot 3 \cdot 5 \cdot 7 = 840$. Imamo da je $2^{840} \bmod n = 346905$ i $\mathrm{nzd}(346904, n) = 421$. Zaista, $n = 421 \cdot 2011$. ♢

Pomoću $p-1$ metode je Baillie 1980. godine našao 25-znamenkasti faktor Mersenneovog broja $2^{257} - 1$.

Uspjeh $p-1$ metode direktno ovisi o glatkoći broja $p-1$. Postoje varijante ove metode koje koriste glatkoću brojeva $p+1$, p^2+p+1, p^2+1 ili p^2-p+1. No, najvažnija modifikacija $p-1$ metode je Lenstrina metoda faktorizacije pomoću eliptičkih krivulja. U njoj se, ponovo, grupa $\mathbb{F}_p^*$ reda $p-1$ zamjenjuje grupom $E(\mathbb{Z}_p)$, čiji red varira unutar intervala $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$, pa se možemo nadati da ćemo pronaći eliptičku krivulju nad $\mathbb{Z}_p$ dovoljno glatkog reda.

4.3 Metoda verižnog razlomka

Kao i kod metode faktorizacije pomoću eliptičkih krivulja, i ovdje je opća metoda motivirana jednom specijalnom metodom. U ovom slučaju ta specijalna metoda je *Fermatova faktorizacija*, koja je primjenjiva na brojeve n koji su produkti dvaju bliskih brojeva. Takav n je tada razlika kvadrata dvaju prirodnih brojeva, od kojih je jedan jako mali, a drugi je jako blizak broju $\sqrt{n}$. Naime, ako je $n = ab$, onda je $n = t^2 - s^2$, gdje je $t = \frac{a+b}{2}$, $s = \frac{a-b}{2}$, pa ako su a i b bliski, onda je s jako mali, a t samo malo veći od $\sqrt{n}$. Stoga uvrštavanjem za t redom $t = \lfloor\sqrt{n}\rfloor + 1$, $\lfloor\sqrt{n}\rfloor + 2$, ... možemo pronaći t, a time i traženu faktorizaciju.

Primjer 4.3. Neka je $n = 201703$. Imamo: $\lfloor\sqrt{201703}\rfloor + 1 = 450$. Sada je $450^2 - 201703 = 797$, što nije kvadrat. Nastavljamo s $t = 451$. Imamo $451^2 - 201703 = 1698$, što ponovo nije kvadrat. Međutim, $452^2 - 201793 = 2601 = 51^2$, pa je $201703 = 452^2 - 51^2 = (452+51)(452-51) = 503 \cdot 401$. ♢

Mnoge moderne metode faktorizacije koriste sljedeću modifikaciju Fermatove faktorizacije. Umjesto da tražimo brojeve s i t takve da je $n = t^2 - s^2$, pokušajmo naći brojeve s i t takve da $n | t^2 - s^2$, tj. da je $s^2 \equiv t^2 \pmod{n}$. Ako je pritom $s \not\equiv t \pmod{n}$, onda su $\mathrm{nzd}(s+t, n)$ i $\mathrm{nzd}(t-s, n)$ netrivijalni faktori od n.

Prva takva metoda koju ćemo prikazati jest *metoda verižnog razlomka*, koja se još naziva i *Brillhart-Morrisonova metoda*, budući da su je oni 1970. godine iskoristili za faktorizaciju Fermatovog broja $2^{2^7}+1$. No, osnovna ideja

se može naći već u radovima Kraitchika, Lehmera i Powersa 20-tih i 30-tih godina 20. stoljeća.

Neka je n složen broj kojeg želimo faktorizirati. Možemo pretpostaviti da n nije potpun kvadrat. Tada je razvoj broja $\sqrt{n}$ u verižni razlomak periodičan. Preciznije,

$$\sqrt{n} = [a_0; \overline{a_1, a_2, \ldots, a_{r-1}, 2a_0}].$$

Neka je $\frac{p_i}{q_i} = [a_0; a_1, \ldots, a_i]$. Tada vrijedi

$$p_i^2 - nq_i^2 = (-1)^{i+1} \cdot t_{i+1} \text{ i } 0 < t_i < 2\sqrt{n},$$

gdje je niz (t_i) definiran Poglavlju 1.6. Dakle, konvergente verižnog razlomka zadovoljavaju kongruencije oblika

$$p_i^2 \equiv w_i \pmod{n},$$

gdje je w_i relativno mali. Ako uspijemo pronaći neke w_i-ove čiji je produkt potpun kvadrat, recimo $w_{k_1} \cdots w_{k_m} = w^2$, onda smo pronašli željenu kongruenciju

$$(p_{k_1} \cdots p_{k_m})^2 \equiv w^2 \pmod{n},$$

te se možemo nadati da će nam $\text{nzd}(p_{k_1} \cdots p_{k_m} + w, n)$ dati netrivijalni faktor od n.

Primjer 4.4. *Faktorizirati broj* $n = 9073$.

Rješenje: Računamo:

i	0	1	2	3	4	5
s_i	0	95	49	90	92	82
t_i	1	**48**	139	7	87	**27**
a_i	95	3	1	26	2	6
$p_i \bmod n$	95	286	381	1119	2619	7760

Očito je $w_0 w_4 = (-1)^1 t_1 \cdot (-1)^5 t_5 = 36^2$. Stoga je

$$p_0^2 p_4^2 = (95 \cdot 2619)^2 \equiv 3834^2 \equiv 36^2 \pmod{9073}.$$

Izračunamo: $\text{nzd}(3834 + 36, 9073) = 43$. Zaista, $9073 = 43 \cdot 211$. ♢

Da bi metoda verižnog razlomka postala stvarno relativno efikasna (subeksponencijalna) metoda, gore opisanu ideju treba kombinirati s korištenjem *faktorske baze* za nalaženje relacija oblika $w_{k_1} \cdots w_{k_m} = w^2$. Dakle, formiramo faktorsku bazu $\mathcal{B}$ koja se sastoji od broja -1, te svih prostih brojeva $\leq B_1$, gdje je B_1 prikladno odabrana granica. Sada svaki od gore dobivenih w_i-ova pokušamo prikazati kao produkt elemenata iz $\mathcal{B}$. Recimo da $\mathcal{B}$

ima m elemenata. Tada, nakon što uspijemo faktorizirati barem $m+1$ w_i-ova, pogledamo odgovarajuće vektore parnosti eksponenata. To su vektori u $\mathbb{Z}_2^m$. Budući da tih vektora ima više od dimenzije pripadnog vektorskog prostora, oni su linearno zavisni. Gaussovom eliminacijom (ili, još bolje, nekom od specijalnih metoda za "rijetke" matrice, npr. tzv. Lanczosovom metodom) pronađemo njihovu netrivijalnu linearnu kombinaciju koja daje nul-vektor. Drugim riječima, nađemo podskup w_i-ova takav da je suma pripadnih vektora parnosti parna. No, to znači da je produkt tih w_i-ova potpun kvadrat.

Kod metode verižnog razlomka, otprilike se pola prostih brojeva može izostaviti iz faktorske baze $\mathcal{B}$. Naime, ako $p|w_i$, onda $p|(p_i^2 - nq_i^2)$, pa je p kvadratni ostatak modulo n. Zato se iz faktorske baze mogu izbaciti svi oni p-ovi za koje je $(\frac{n}{p}) = -1$.

U našem gornjem primjeru imamo: $w_0 = (-1)^1 \cdot 2^4 \cdot 3^1$, $w_4 = (-1)^1 \cdot 3^3$, pa bi za $\mathcal{B} = \{-1, 2, 3\}$ pripadni vektori parnosti bili $[1, 0, 1]$ i $[1, 0, 1]$, čija je suma $[0, 0, 0]$.

Što se tiče složenosti metode verižnog razlomka (slično kao kod ECM), pokazuje se da je optimalan izbor granice $B_1 \approx e^{\sqrt{\ln n \ln\ln n}}$, što daje ocjenu za očekivani broj operacija

$$O\Big(e^{(\sqrt{2}+\varepsilon)\sqrt{\ln n \ln\ln n}}\Big).$$

Ipak, za razliku od ECM, ove ocjene nisu sasvim strogo dokazane, već se zasnivaju na nekim nedokazanim "heurističkim" slutnjama.

Kod svoje poznate faktorizacije broja $2^{2^7}+1$, Brillhart i Morrison su koristili jednu modifikaciju gore opisane metode. Naime, umjesto razvoja $\sqrt{n}$, predložili su korištenje razvoja broja $\sqrt{kn}$ za neki mali broj k. Ova modifikacija je posebno korisna ako razvoj od $\sqrt{n}$ ima mali period. U ovom konkretnom slučaju, oni su koristili $k = 257$.

Od drugih uspješnih faktorizacija metodom verižnih razlomaka, spomenimo još faktorizaciju 56-znamenkastog broja N'_{11} (Naur, 1982). Ovdje je $N'_{11} = \frac{N_{11}}{1307}$, a brojevi N_i se definiraju rekurzivno s $p_1 = 2$, $N_i = p_1 p_2 \cdots p_{i-1} + 1$, gdje je p_j najveći prosti faktor od N_j (motivacija za promatranje ovih brojeva dolazi iz Euklidova dokaza beskonačnosti skupa prostih brojeva).

4.4 Metoda kvadratnog sita

Kvadratno sito je varijanta metode faktorske baze koju je uveo Pomerance 1982. godine. Ovdje za faktorsku bazu $\mathcal{B}$ uzimamo

$$\mathcal{B} = \{p \,:\, p \text{ neparan prost broj}, p \le B,\ \left(\frac{n}{p}\right) = 1\} \cup \{2\},$$

gdje je B broj odabran na neki prikladan način. Skup S u kojem tražimo $\mathcal{B}$-brojeve (to su oni koji su djeljivi samo s prostim brojevima iz $\mathcal{B}$) bit će isti

kao u Fermatovoj faktorizaciji, tj.

$$S = \{t^2 - n \,:\, \lfloor\sqrt{n}\rfloor + 1 \leq t \leq \lfloor\sqrt{n}\rfloor + A\},$$

za neki prikladno odabrani A.

Glavna ideja ove metode je da umjesto da za svaki $s \in S$ dijeleći ga s prostim brojevima $p \in \mathcal{B}$ provjeravamo je li $\mathcal{B}$-broj, uzimamo jedan po jedan $p \in \mathcal{B}$ i ispitujemo djeljivost s p za sve $s \in S$. Odavde dolazi i naziv "sito", po analogiji s Eratostenovim sitom za generiranje tablice prostih brojeva. Opisat ćemo glavne korake u faktorizaciji neparnog složenog broja n metodom kvadratnog sita (QS).

Algoritam kvadratnog sita:

1. Odaberimo brojeve B i A, oba reda veličine $e^{\sqrt{\ln n \, \ln\ln n}}$. Obično se uzima da je $B < A < B^2$.

2. Za $t = \lfloor\sqrt{n}\rfloor + 1$, $\lfloor\sqrt{n}\rfloor + 2$, ... , $\lfloor\sqrt{n}\rfloor + A$, napravimo listu brojeva $t^2 - n$ (i stavimo ih u jedan stupac).

3. Za svaki prost broj $p \leq B$, provjerimo je li $(\frac{n}{p}) = 1$. Ako nije, izbacimo p iz faktorske baze.

4. Za neparan prost broj p iz baze $\mathcal{B}$, rješavamo kongruenciju $t^2 \equiv n \pmod{p^\beta}$ za $\beta = 1, 2, \ldots$. Neka je β najveći prirodan broj za kojeg postoji t, $\lfloor\sqrt{n}\rfloor + 1 \leq t \leq \lfloor\sqrt{n}\rfloor + A$, takav da je $t^2 \equiv n \pmod{p^\beta}$. Neka su t_1 i t_2 dva rješenja od $t^2 \equiv n \pmod{p^\beta}$ takva da je $t_2 \equiv -t_1 \pmod{p^\beta}$ (t_1 i t_2 nisu nužno oba iz S).

5. Za p iz 4. koraka, pogledamo listu iz 2. koraka. U stupcu ispod p stavimo 1 kod svih vrijednosti od $t^2 - n$ kod kojih $p|t - t_1$; promijenimo 1 u 2 kod onih kod kojih $p^2|t - t_1$; promijenimo 2 u 3 ako $p^3|t - t_1$, itd. sve do p^β. Potom napravimo sve isto s t_2 umjesto t_1. Najveći broj koji će se pojaviti u ovom stupcu bit će β.

6. Svaki put kad u 5. koraku stavimo 1 ili promijenimo 1 u 2, ili 2 u 3, ili itd., podijelimo odgovarajući $t^2 - n$ sa p i zabilježimo rezultat.

7. U stupcu ispod $p = 2$, ako $n \not\equiv 1 \pmod 8$, onda stavimo 1 kod svih $t^2 - n$ u kojima je t neparan, te podijelimo $t^2 - n$ s 2. Ako je $n \equiv 1 \pmod 8$, onda rješavamo kongruenciju $t^2 \equiv n \pmod{2^\beta}$ i radimo sve isto kao za neparne p (osim što će za $\beta \geq 3$ biti 4 različita rješenja t_1, t_2, t_3, t_4 modulo 2^β).

8. Kad obradimo sve proste brojeve p iz $\mathcal{B}$, odbacimo sve $t^2 - n$, osim onih koji su postali jednaki 1 nakon dijeljenja s potencijama prostih brojeva u prethodna dva koraka. Dobit ćemo tako tablicu u kojoj će jedan stupac sadržavati vrijednosti elemenata $t^2 - n$ iz S koji su $\mathcal{B}$-brojevi, a

ostali stupci će sadržavati potencije od $p \in \mathcal{B}$ u rastavu brojeva $t^2 - n$ na proste faktore.

9. Ostatak algoritma je isti kao kod opće metode faktorske baze.

Primjer 4.5. Neka je $n = 1042387$. Uzmimo $B = 80$ i $A = 500$. Ovdje je $\lfloor \sqrt{n} \rfloor = 1020$. Naša faktorska baza se sastoji od 12 prostih brojeva $\{2, 3, 11, 17, 19, 23, 43, 47, 53, 61, 67, 71\}$. Budući da je $n \not\equiv 1 \pmod 8$, u stupcu pod $p = 2$ stavljamo 1 kod svih neparnih brojeva između 1021 i 1520.

Opisat ćemo detaljno formiranje stupca pod $p = 3$. Želimo naći jedno rješenje kongruencije $t_1^2 \equiv 1042387 \pmod{3^\beta}$, gdje je β definiran u 4. koraku algoritma. Rješenje tražimo u obliku $t_1 = t_{1,0} + t_{1,1} \cdot 3 + t_{1,2} \cdot 3^2 + \cdots + t_{1,\beta-1} \cdot 3^{\beta-1}$, $t_{1,j} \in \{0, 1, 2\}$. Henselova lema nam osigurava da ako postoji rješenje kongruencije modulo 3, onda postoji rješenje modulo 3^m za svaki prirodan broj m. Za rješenje modulo 3, možemo uzeti $t_{1,0} = 1$. Zatim gledamo kongruenciju modulo 9, pa imamo: $(1 + 3t_{1,1})^2 \equiv 7 \pmod 9$, odakle je $t_{1,1} = 1$. Modulo 27 imamo: $(1+3+9t_{1,2})^2 \equiv 25 \pmod{27}$, odakle je $t_{1,2} = 2$. Nastavljajući ovaj postupak do 3^7, dobivamo $t_1 = (210211)_3 = 589 \bmod 3^7$. Budući da je $589 < 1021$, a $3^7 - 589 = 1598 > 1520$, zaključujemo da je $\beta = 6$ i možemo uzeti $t_1 = 589 \equiv 1318 \pmod{3^6}$ i $t_2 = 3^6 - 589 = 140$ ($t_2 \equiv 1112 \pmod{3^5}$).

Sada konstruiramo "sito" za $p = 3$. Krenuvši od 1318, skačemo za po 3 na dolje do 1021 i na gore do 1519, te svaki put stavimo 1 u stupac i podijelimo odgovarajući $t^2 - n$ s 3. Tada napravimo sve isto sa skokovima po 9, 27, 81, 243 i 729 (ustvari, za 729 nemamo skokova, već samo promijenimo 5 u 6 kod 1318 i podijelimo $1318^2 - 1042387$ još jednom sa 3). Nakon toga ponovimo sve krenuvši u skokove od 1112 umjesto 1318. Ovaj se put zaustavljamo kod skoka za 243.

Nakon što ovaj postupak primijenimo na preostalih 10 brojeva u faktorskoj bazi, dobit ćemo tablicu 500×12 u kojoj redci odgovaraju t-ovima između 1021 i 1520. Izbacimo li sve retke za koje se $t^2 - n$ nije reducirao na 1, tj. zadržimo li samo one retke za koje je $t^2 - n$ $\mathcal{B}$-broj, dobivamo sljedeću tablicu:

t	t^2-n	2	3	11	17	19	23	43	47	53	61	67	71
1021	54	1	3										
1027	12342	1	1	2	1								
1030	18513		2	2	1								
1061	83334	1	1		1	1		1					
1071	104654	1		1								1	1
1112	194157		5		1				1				
1129	232254	1	3	1	1		1						
1148	275517		2	3			1						
1175	338238	1	2			1	1	1					
1213	428982	1	1							1			1
1217	438702	1	1	1	2		1						
1390	889713		2	2		1		1					
1520	1268013		1		1		2		1				

Sada tražimo relacije modulo 2 između redaka u ovoj matrici, tj. retke čija je suma jednaka nul-vektoru u $\mathbb{Z}_2^{12}$. Jedan takav slučaj nalazimo u prva tri retka. Tako dobivamo

$$(1021 \cdot 1027 \cdot 1030)^2 \equiv (2 \cdot 3^3 \cdot 11^2 \cdot 17)^2 \pmod{1042387}.$$

Nažalost, brojevi pod kvadratom na obje strane ove kongruencije kongruentni su 111078 modulo 1042387, tako da ne dobivamo ništa korisno. Međutim, ako kombiniramo šesti i zadnji redak, dobivamo

$$\begin{aligned}(1112 \cdot 1520)^2 &\equiv (3^3 \cdot 17 \cdot 23 \cdot 47)^2 \pmod{1042387},\\ 647853^2 &\equiv 496179^2 \pmod{1042387},\end{aligned}$$

odakle dobivamo netrivijalni faktor $\operatorname{nzd}(647853 - 496179, 1042387) = 1487$. Drugi faktor je 701. ♢

Očekivani broj operacija metodom kvadratnog sita je

$$O\Big(e^{\sqrt{\log n \, \log\log n}}\Big),$$

dakle, vrlo slično kao kod faktorizacije pomoću eliptičkih krivulja. Spomenimo da je 1994. godine metodom kvadratnog sita faktoriziran tzv. RSA-129. To je broj od 129 znamenaka koji je produkt dva prosta broja od 64 i 65 znamenaka.

Trenutno najbolja poznata metoda faktorizacije je *metoda sita polja brojeva* (engl. Number Field Sieve - NFS) koja kombinira ideje iz metode kvadratnog sita i algebarsku teoriju brojeva. Kod ove je metode očekivani broj

operacija

$$O\Big(e^{c(\log n)^{1/3}\,(\log\log n)^{2/3}}\Big),$$

gdje je $c = \sqrt[3]{\frac{64}{9}} \approx 1.92$. Metodu je prvi put upotrijebio Pollard 1990. godine za faktorizaciju Fermatova broja $2^{2^9}+1$. Ovom su metodom faktorizirani brojevi iz natječaja *RSA Factoring Challenge*: RSA-130 (1996. godine), RSA-140, RSA-155 (oba 1999.), RSA-160, RSA-576 (oba 2003.), RSA-200, RSA-640 (oba 2005.), RSA-210 (2013.), RSA-220 (2016.), RSA-230 (2018.), RSA-240 (2019.), RSA-250 (2020.). Npr. broj RSA-200 ima 200 znamenaka, dok broj RSA-640 ima 640 bitova (193 znamenke) (terminologija sa znamenaka na bitove je promijenjena 2001. godine). Brojevi su dizajnirani tako da slijede pravila za odabir modula u RSA algoritmu, te uspjesi u njihovoj faktorizaciji daju dobru informaciju o tome koliko velik RSA modul bi trebalo birati da bi nam komunikacija bila sigurna (danas, ali i u skoroj budućnosti). Broj RSA-250 je u veljači 2020. godine faktorizirala grupa istraživača (F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé, P. Zimmermann). Evo tog broja i njegove faktorizacije:

$$\begin{aligned}
&2140324650240744961264423072839333563008614715144755017797754920881418023447140136643345519095804679610992851872470914587687 3\backslash\\
&9626192155736304745477052080511905649310668769159001975940569345745223058932597669747168173806936489469987157849497593749793 7 =\\
&6413528947707158027879019017057738908482501474294344720811685963202453234463023862359875266834770873766192558569463979885336 7\times\\
&\times 3337202759497815655622601060535511422794076034476755466678452098702384172921003708025744867329688187756571898625803693206271 1.
\end{aligned}$$

Bibliografija

[1] H. COHEN, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1993.

[2] R. CRANDALL, C. POMERANCE, *Prime Numbers. A Computational Perspective*, Springer-Verlag, 2001.

[3] A. DAS, *Computational Number Theory*, CRC Press, 2013.

[4] A. DUJELLA, *Teorija brojeva*, Školska knjiga, Zagreb, 2019.

[5] A. DUJELLA, *Number Theory*, Školska knjiga, Zagreb, 2021.

[6] A. DUJELLA, M. MARETIĆ, *Kriptografija*, Element, Zagreb, 2007.

[7] N. KOBLITZ, *A Course in Number Theory and Cryptography*, Springer-Verlag, 1994.

[8] A. J. MENEZES, P. C. OORSCHOT, S. A. VANSTONE, *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] D. R. STINSON, *Cryptography. Theory and Practice*, CRC Press, 1996, 2002, 2005.