Teorija brojeva u kriptografiji

Andrej Dujella

Poslijediplomski kolegij 2003/2004.

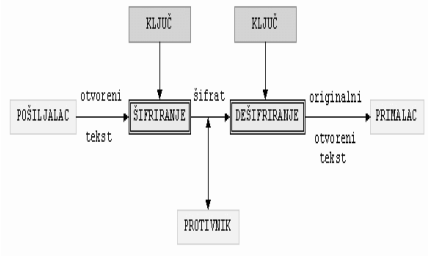
Poglavlje 1

Kriptografija javnog ključa

1.1 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 *Eve* 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 1.1. 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 poptiše"*. 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.

1.2 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 200 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*:

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

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 $(x, n) = 1$. Sada je

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

(prema Eulerovom teoremu). Ako je $(n, x) = n$, onda je $x^{de} \equiv 0 \equiv x \pmod{n}$. Ako je $(n, x) = p$, onda je $x^{de} \equiv 0 \equiv x \pmod{p}$ i $x^{de} = (x^{q-1})^{(p-1)k}$. $x \equiv x \pmod{q}$, pa je $x^{de} \equiv x \pmod{n}$. Slučaj $(n, x) = q$ 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 $m \in \mathbb{Z}_n^*$, tj. takve da je $(m, n) = 1$. Može se pokazati da je $a^{m/2} \not\equiv \pm 1 \pmod{n}$ za barem 50% svih $a \in \mathbb{Z}_n^*$. Ako je a jedan takav broj, onda je $(a^{m/2} - 1, n)$ netrivijalni faktor od n.

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 100 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 $(e, \varphi(n)) = 1$, 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. Vidjet ćemo 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 $x \equiv \pm a^{(p+1)/4} \pmod{p}$. Zaista, $x^2 \equiv a^{(p+1)/2} = a^{(p-1)/2} \cdot a \equiv a \pmod{p}$, po Eulerovom kriteriju za kvadratne ostatke. 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 $(x, n) = 1$ (ako je $(x, n) > 1$, 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 $(x-y, n)$ 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.

1.3 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. Broj $g \in \{1, 2, ..., p - 1\}$ je primitivni korijen modulo p ako je g^{p-1} najmanja potencija broja g koja daje ostatak 1 pri djeljenju s 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 oko 1024 bita. Također bi, zbog razloga koje ćemo kasnije objasniti, 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 1.2. *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 $E(K)$.

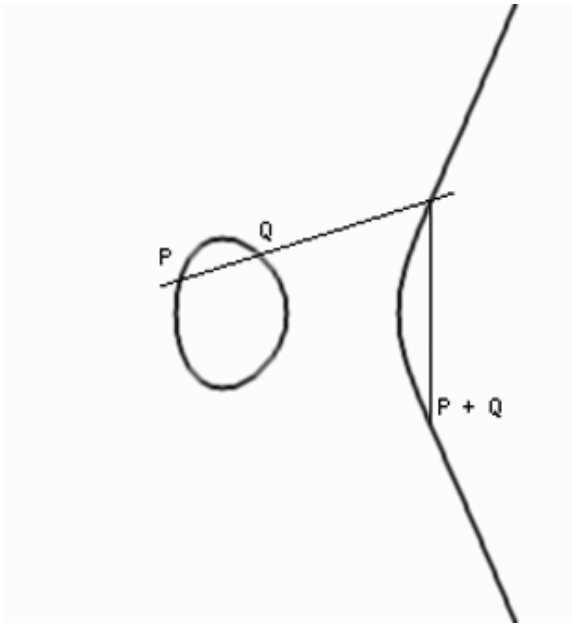
Vrlo slično se definira eliptička krivulja i nad poljima karakteristike 2 ili 3.

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 $P \in E(\mathbb{R})$.

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. O svojstvima eliptičkih krivulja nad konačnim poljem ćemo kasnije govoriti opširnije. Vidjet ćemo također da eliptičke krivulje imaju važnu primjenu i na probleme koje smo spominjali u prethodnom poglavlju, a to su faktorizacija i dokazivanje prostosti.

Svi kriptosustavi koji u svojoj originalnoj definiciji koriste grupu $\mathbb{Z}_p^*$, kao što je npr. ElGamalov, mogu se vrlo lako modificirati tako da koriste grupu $E(\mathbb{Z}_p)$. No, doslovno prevođenje ElGamalovog kriptosustava u eliptičke krivulje ima nekoliko nedostataka.

Prvi je da prije šifriranja moramo elemente otvorenog teksta prebaciti u točke na eliptičkoj krivulji. Za to ne postoji zadovoljavajući deterministički algoritam. No, postoji vjerojatnosni algoritam, koji koristi činjenicu da kvadrati u konačnom polju predstavljaju 50% svih elemenata. To znači da s približnom vjerojatnošću $1-\frac{1}{2^k}$ možemo očekivati da ćemo iz k pokušaja pronaći broj x takav da je $x^3 + ax + b$ kvadrat u $\mathbb{Z}_p$. Za $k = 30$ to je sasvim zadovoljavajuća vjerojatnost. Pretpostavimo sada da su nam osnovne je-

dinice otvorenog teksta cijeli brojevi između 0 i M. Pretpostavimo nadalje da je $p > Mk$. Sada otvorenom tekstu m pridružujemo točku na eliptičkoj krivulji $E(\mathbb{Z}_p)$ na sljedeći način. Za brojeve x oblika $mk + j$, $j = 1, 2, \ldots, k$ provjeravamo da li je $x^3 + ax + b$ kvadrat u $\mathbb{Z}_p$. Kad nađemo takav broj, izračunamo y koji zadovoljava da je $y^2 \equiv x^3 + ax + b \pmod{p}$, te broju m pridružimo točku (x, y) na $E(\mathbb{Z}_p)$. Obrnuto, iz točke (x, y) pripadni otvoreni tekst m možemo dobiti po formuli

$$m = \left\lfloor \frac{x-1}{k} \right\rfloor.$$

Drugi problem je da se šifrat jednog elementa otvorenog teksta kod ove varijante ElGamalovog kriptosustava sastoji od uređenog para točaka na eliptičkoj krivulji. To znači da, prilikom šifriranja, poruka postane otprilike 4 puta dulja.

Navest ćemo jednu varijantu ElGamalovog kriptosustava koja koristi eliptičke krivulje. Zove se *Menezes-Vanstoneov kriptosustav*. U njemu se eliptičke krivulje koriste samo za "maskiranje", dok su otvoreni tekstovi i šifrati proizvoljni uređeni parovi elemenata iz polja (a ne nužno parovi koji odgovaraju točkama na eliptikoj krivulji). Kod ovog kriptosustava, šifrirana poruka je (samo) 2 puta dulja od originalne poruke.

Menezes-Vanstoneov kriptosustav: Neka je E eliptička krivulja nad $\mathbb{Z}_p$ ($p > 3$ prost), te H ciklička podgrupa od E generirana s α. Neka je $\mathcal{P} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, $\mathcal{C} = E \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ i

$$\mathcal{K} = \{(E, \alpha, a, \beta) \,:\, \beta = a\alpha\},$$

gdje $a\alpha$ označava $\alpha + \alpha + \cdots + \alpha$ (a puta), a $+$ je zbrajanje točaka na eliptičkoj krivulji.

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

Za $K \in \mathcal{K}$ i tajni slučajni broj $k \in \{0, 1, \ldots, |H| - 1\}$, te za $x = (x_1, x_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ definiramo

$$e_K(x, k) = (y_0, y_1, y_2),$$

gdje je $y_0 = k\alpha$, $(c_1, c_2) = k\beta$, $y_1 = c_1 x_1 \bmod p$, $y_2 = c_2 x_2 \bmod p$.

Za šifrat $y = (y_0, y_1, y_2)$ definiramo

$$d_K(y) = (y_1 (c_1)^{-1} \bmod p,\ y_2 (c_2)^{-1} \bmod p),$$

gdje je $ay_0 = (c_1, c_2)$.

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.

Multiplikativna grupe konačnog polja i grupa točaka na eliptičkoj krivulji nad konačnim polje 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 (roda) 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 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
$d = \gcd(a_1, a_2, b_1 + b_2) = s_1 a_1 + s_2 a_2 + s_3(b_1 + b_2)$
$a_3 = a_1 a_2 / d^2$
$b_3 = (s_1 a_1 b_2 + s_2 a_2 b_1 + s_3(b_1 b_2 + f))/d \bmod a_3$
while $(\deg a_3 > g)$ {
$a_3 = (f - b_3^2)/a_3$
$b_3 = -b_3 \bmod a_3$ }

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.

1.4 Ostali kriptostavi s javnim ključem

U ovom poglavlju ćemo vrlo kratko spomenuti još neke kriptosustave s javnim ključem, te također ukratko opisati matematičke probleme na kojima se oni zasnivaju. Recimo odmah da se, zbog različitih razloga, ovi kriptosustavi u praksi upotrebljavaju puno rjeđe nego kriptosustavi opisani u prethodna dva poglavlja.

Merkle-Hellmanov kriptosustav (Merkle i Hellman, 1978) za osnovu ima tzv. *problem ruksaka*. Pretpostavimo da imamo n predmeta s volumenima v_1, v_2, ... , v_n s kojima želimo napuniti ruksak volumena V. Dakle, želimo naći $J \subseteq \{1, 2, \ldots, n\}$ tako da je $\sum_{j \in J} v_j = V$ (ako takav podskup postoji). Ekvivalentna formulacija je:

> **Problem ruksaka:** Za dani skup $\{v_1, v_2, \ldots, v_n\}$ od n prirodnih brojeva i prirodan broj V, naći niz $m = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ od n binarnih znamenaka ($\varepsilon_i \in \{0, 1\}$) tako da je
>
> $$\varepsilon_1 v_1 + \varepsilon_2 v_2 + \cdots + \varepsilon_n v_n = V,$$

ako takav m postoji.

Poznato je da je ovaj opći problem ruksaka vrlo težak. On spada u tzv. *NP-potpune* probleme. To, pored ostalog, znači da nije poznat polinomijalni algoritam za njegovo rješavanje. Međutim, jedan njegov specijalni slučaj, tzv. *superrastući problem ruksaka*, je puno lakši. To je slučaj kad je niz v_1, v_2, ... , v_n rastući i vrijedi

$$v_j > v_1 + v_2 + \cdots v_{j-1} \quad \text{za } j = 2, 3, ..., n.$$

Primjer superrastućeg niza je niz $v_i = 2^{i-1}$. Jasno je da u slučaju superrastućeg niza, u svakom koraku u ruksak moramo staviti najveći predmet koji u njega stane. To vodi do sljedećeg algoritma:

Algoritam za superrastući problem ruksaka:

for $(n \geq i \geq 1)$ {
  if $(V \geq v_i)$ then $V = V - v_i$; $\varepsilon_i = 1$
    else $\varepsilon_i = 0$ }
if $(V = 0)$ then $(\varepsilon_1, \ldots, \varepsilon_n)$ *je rješenje*
  else *nema rješenja*

Ideja Merkle-Hellmanovog kriptosustava je "zamaskirati" superrastući niz tako da izgleda kao sasvim slučajan niz. Onaj kome je poruka namjenjena (Bob) zna kako ukloniti masku, pa može pročitati poruku rješavajući supperrastući problem ruksaka. Svi drugi moraju rješavati, puno teži, opći problem ruksaka, pa ne mogu pročitati poruku. "Maskiranje" se provodi pomoću modularnog množenja.

Merkle-Hellmanov kriptosustav: Neka je $v = (v_1, v_2, \ldots, v_n)$ superrastući niz prirodnih brojeva, te neka je $p > v_1 + v_2 + \cdots + v_n$ prost broj i $1 \leq a \leq p - 1$. Za $i = 1, 2, \ldots n$, definiramo

$$t_i = av_i \bmod p$$

i označimo $t = (t_1, t_2, \ldots, t_n)$. Neka je

$$\mathcal{P} = \{0,1\}^n, \quad \mathcal{C} = \{0, 1, \ldots, n(p-1)\} \quad \text{i} \quad \mathcal{K} = \{(v, p, a, t)\},$$

gdje su v, p, a i t konstruirani na gore opisani način.

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

$$e_K(x_1, x_2, \ldots, x_n) = x_1t_1 + x_2t_2 + \cdots + x_nt_n.$$

Za $0 \leq y \leq n(p-1)$ definiramo $z = a^{-1}y \bmod p$, riješimo (superrastući) problem ruksaka za skup $\{v_1, v_2, \ldots, v_n, z\}$ i tako dobivamo

$$d_K(y) = (x_1, x_2, \ldots, x_n).$$

Vrijednost t je javna, dok su vrijednosti p, a i v tajne.

Merkle-Hellmanov kriptosustav je imao jednu vrlo veliku prednost u odnosu na ostale kriptosustave s javnim ključem. Naime, šifranje pomoću njega je znatno brže, te je on po brzini bio usporediv s najboljim simetričnim kriptosustavima. No, godine 1982. je usljedilo razočaranje, kad je Shamir pronašao polinomijalni algoritam za razbijanje Merkle-Hellmanovog kriptosustava. Pokazalo se da se ovako jednostavnim maskiranjem vrlo specijanog niza ipak ne dobiva sasvim slučajan niz. U razbijanju se koriste algoritmi za diofantske aproksimacije (verižni razlomci i LLL-algoritam) koje ćemo kasnije detaljno opisati. Prema tome, ovaj sustav se ne može više smatrati sigurnim kriptosustavom. Ipak, ideja na kojoj je zasnovan je vrlo zanimljiva. Ta ideja je korištenje u dešifriranju nekog jednostavnog specijalnog slučaja nekog teškog (NP-potpunog) problema, s time da se taj specijalni slučaj prikrije tako da izgleda kao opći.

Ova ideja se koristi i u *McElieceovom kriptosustavu* (McEliece, 1978). Ovdje je pripadni NP-potpuni problem dekodiranje općih linearnih kodova za ispravljanje grešaka. Kao osnova u ovom kriptosustavu koristi se specijalna klasa tzv. *Goppa kodova* za koje postoji polinomijalni algoritam za dekodiranje.

Neka su k i n prirodni brojevi, $k \leq n$. *Linearni* $[n,k]$*-kod* je k-dimenzionalni potprostor od $\mathbb{Z}_2^n$. Generirajuća matrica za linearni kod C je $k \times n$ matrica G čiji redci tvore bazu za C.

Za $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n) \in \mathbb{Z}_2^n$, definiramo *Hammingovu udaljenost* sa

$$d(x,y) = |\{i \,:\, 1 \leq i \leq n, x_i \neq y_i\}|,$$

tj. kao broj koordinata u kojima se x i y razlikuju. Neka je C neki linearni $[n,k]$-kod. Definiramo udaljenost od C sa

$$d(C) = \min\{d(x,y) \,:\, x,y \in C, x \neq y\}.$$

Tada za C kažemo da je $[n,k,d]$-kod.

Primjer jednog $[7,4,3]$-koda dan je sljedećom matricom:

$$G = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,0 \\ 0\,1\,0\,0\,1\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,1\,1\,1 \end{pmatrix}$$

Svrha kodova za ispravljanje grešaka jest da isprave slučajne greške koje mogu nastati prilikom prenošenja binarnih podataka preko kanala sa "šumom". Neka je G generirajuća matrica za $[n,k,d]$-kod C. Pretpostavimo da je x binarna k-torka koju Alice želi prenijeti Bobu preko kanala sa šumom. Tada Alice *kodira* x kao n-torku $y = xG$, te pošalje y preko kanala. Bob primi n-torku r koja se, budući da kanal ima šum, ne mora podudarati sa y. Da

bi *dekodirao* r, Bob traži element ("kodnu riječ") $y' \in C$ koja ima najmanju Hammingovu udaljenost od r, te potom izračuna k-torku x' takvu da je $y' = x'G$. Tada Bob može očekivati da je $y' = y$, pa onda i $x' = x$, tj. da je uspio ispraviti sve greške nastale prilikom prijenosa. Nije teško vidjeti da ukoliko broj grešaka nije veći od $(d-1)/2$, onda se na ovaj način mogu ispraviti sve greške.

Pokazuje se da je problem nalaženja najbliže kodne riječi vrlo težak problem za opće linearne kodove. No, postoje kodovi za koje postoje efikasni algoritmi za dekodiranje. McEliece je 1978 predlozio da se jedna klasa takvih kodova, tzv. Goppa kodovi, iskoriste u konstrukciji kriptosustava s javnim ključem. Parametri Goppa kodova imaju oblik

$$n = 2^m,\ d = 2t+1,\ k = n - mt.$$

McEliece je predložio korištenje Goppa kodova s parametrima $[1024, 524, 101]$. Otvoreni tekst je binarna 524-torka, a odgovarajući šifrat je binarna 1024-torka. Javni ključ je 524×1024 binarna matrica. Do danas nije poznat niti jedan efikasan napad na McEliecov kriptosustav. Međutim, ovaj kriptosustav nije korišten u praksi, prvenstveno zbog ogromne veličine javnog ključa.

Jedan od najzanimljivijih novijih kriptosustava, koji je još uvijek predmet intenzivnog proučavanja, je *NTRU kriptosustav*, koji su 1997. godine predložili Hoffstein, Pipher i Silverman. (Skraćenica NTRU dolazi od "Number Theory Research Unit"). U ovom se kriptosustavu kod šifriranja koriste polinomi, preciznije koristi se prsten $R = \mathbb{Z}[X]/(X^n - 1)$. Na elementima od R definira se operacija cikličke konvolucije, tj. za $F = \sum_{i=0}^{n-1} F_i x^i$, $G = \sum_{i=0}^{n-1} G_i x^i$, definira se $H = F \circledast G$ sa

$$H_k = \sum_{i+j \equiv k \pmod n} F_i G_j.$$

Pored toga koristi se redukcija ovako dobivenih polinoma modulo dva relativno prosta broja p i q. Sigurnost ovog kriptosustava se upravo zasniva na "nezavisnosti" te dvije redukcije.

Šifriranje i dešifriranje kod NTRU kriptosustava je znatno brže nego kod npr. RSA kriptosustava, što svakako predstavlja jednu njegovu potencijalnu prednost. No, poznato je nekoliko mogućih napada na NTRU koji koriste LLL-algoritam za nalaženje najkraćeg elementa u rešetki. Za sada nije sasvim jasno koliko su ti napadi ozbiljna prijetnja na sigurnost ovog kriptosustava. Svakako će na to pitanje trebati dati odgovor, prije nego što dođe do eventualnog ulaska ovog kriptosustava u najširu uporabu.

Poglavlje 2

Osnovni algoritmi u teoriji brojeva

2.1 Složenost algoritama

Do sada smo više puta za određene matematičke probleme govorili da su "laki" ili "teški". Da bi 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 s obzirom na broj "osnovnih koraka" potrebnih za njihovo izvršavanje, te ponekad i s obzirom na potreban prostor (memoriju). Pod osnovnim korakom podrazumjevamo 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 $\lfloor \log_2 N \rfloor + 1$.

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 2.1. Neka su $f, g :\to \mathbb{R}$ dvije funkcije. Tada pišemo:

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

(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 input) - to ćemo zvati *složenost algoritma*;
- prosječan broj operacija (prosjek za sve inpute fiksne duljine) - to ćemo zvati *prosječna složenost algoritma*.

Definicija 2.2. *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, zovemo *eksponencijalni*.

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 onaj prvi algoritam će biti brži. Pored toga, 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 2.3. *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 totalno eksponencijalnih, tj. onih čija je složenost funkcija oblika $O(e^{n^k})$.

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

Definicija 2.4. *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 2.1. 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 klasi **co-NP**.

Jasno je vrijedi $\mathbf{P} \subseteq \mathbf{NP}$ i $\mathbf{P} \subseteq \mathbf{co\text{-}NP}$. Opće prihvaćena slutnja je da vrijedi $\mathbf{P} \neq \mathbf{NP}$. To se smatra jednim od najvažijih nerješenih matematičkih problema i spada među sedam tzv. *Millenium Prize Problems*.

Definicija 2.5. 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 2.2. Neka je L_1 = "Za polinom $p(x)$ s cjelobrojnim koeficijentima, postoji li interval na kojem $p(x)$ pada?", L_2 = "Za polinom $q(x)$ s cjelobrojnim koeficijentima, postoji li 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 $q(x)$ primjeniti potpogram za L_2.

Primjer 2.3. 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 $L_2 \leq_P L_1$. Primjenimo algoritam za L_1 na broj N. Tako dobijemo faktor M. Potom ponovo primjenimo 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 primjenimo 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 primjenimo 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 bi odredili sljedeći bit, primijenimo L_2 za $2^{n-2} + 2^{n-3} - 1$ ako je odgovor bio NE, odnosno $2^{n-3} - 1$ 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.

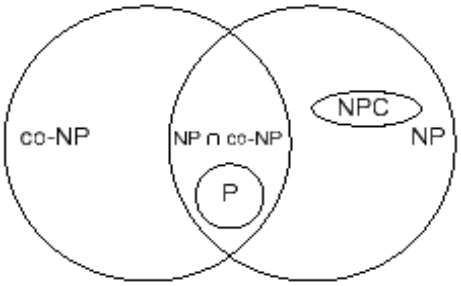
Ilustrirajmo gornji dokaz na konkretnom primjeru $N = 91$. U pozivima algoritma za L_2, pitamo se redom: Ima li 91 faktor između 2 i 63 (DA), između 2 i 31 (DA), između 2 i 15 (DA), između 2 i 7 (DA), između 2 i 3 (NE), između 2 i 5 (NE), između 2 i 6 (NE). Zaključujemo da je broj 91 ima faktor čiji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

Definicija 2.6. Problem odluke L je **NP**-*potpun* ako je $L \in \mathbf{NP}$ i $L_1 \leq_P L$ za svaki $L_1 \in \mathbf{NP}$. 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 koji smo već ranije susreli. Drugi primjer je problem trgovačkog putnika koji treba naći najkraću rutu koja prolazi kroz sve gradove na njegovoj karti. Primjer **NP**-potpunog problema je također i bojanje geografske karte s tri boje.

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

Hipotetski, odnos promatranih klasa izgleda ovako:



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

Definicija 2.7. Kažemo da je problem odluke L rješiv u *vjerojatnosno (randomizirano) polinomijanom 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 odogovor 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 koga je odgovor NE točan s vjerojatnošću većom od $1 - 2^{-k}$.

2.2 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 za zbrajanje:

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

Algoritmi za "školsko" (ili "naivno") množenje i dijeljenje imaju složenost $O(\ln^2 N)$. Podsjetimo se tih algoritama. Neka je $x = (x_n, \ldots, x_0)_b$, $y = (y_t, \ldots, y_0)_b$. Želimo izračunati $x \cdot y = (w_{n+t+1}, \ldots, w_0)_b$.

Algoritam za "školsko" množenje:

for $(0 \leq i \leq n + t + 1)$ $w_i = 0$
for $(0 \leq i \leq t)$ {
    $c = 0$
    for $(0 \leq i \leq t)$ {
        $(uv)_b = w_{i+j} + x_j \cdot y_i + c$; $w_{i+j} = v$; $c = u$ }
    $w_{n+t+1} = u$ }

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 za dijeljenje s ostatkom:

for $(0 \leq i \leq n - t)$ $q_j = 0$
while $(x \geq yb^{n-t})$ $q_{n-t} = q_{n-t} + 1$; $x = x - yb^{n-t}$
for $(n \geq i \geq t - 1)$ {
    if $(x_i = y_t)$ then $q_{i-t-1} = b - 1$
        else $q_{i-t-1} = \lfloor (x_i b + x_{i-1})/y_t \rfloor$
    while $(q_{i-t-1}(y_t b + y_{t-1}) > x_i b^2 + x_{i-1} b + x_{i-2})$
        $q_{i-t-1} = q_{i-t-1} - 1$
    $x = x - q_{i-t-1} y b^{i-t-1}$
    if $(x < 0)$ $x = x + yb^{i-t-1}$; $q_{i-t-1} = q_{i-t-1} - 1$ }
$r = x$

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 zbrajanje imaju složenost

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

(Schönhage-Strassenov algoritam iz 1971. godine). 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 tzv. *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

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

Za sada 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

$$T(2n) \leq T(n) + cn \tag{2.1}$$

za neku konstantu c. Iz (2.1) slijedi da, uz dovoljno veliku konstantu C, vrijedi

$$T(2^k) \leq C(3^k - 2^k), \tag{2.2}$$

za $k \geq 1$. Zaista, neka je konstanta $C \geq c$ odabrana tako da (2.2) vrijedi $k = 1$, te pretpostavimo da (2.2) 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

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

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 $(\ln N)^{1.585}$.

2.3 Modularno množenje i potenciranje

U većini kriptosustava s javnim ključem, šifriranje i dešifriranje je opisano pomoću operacija u prstenu $\mathbb{Z}_m$, 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 n, 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 tzv. *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 $R > m$, $(m, R) = 1$ 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 jednostavni pomak za n znamenaka.

Lema 2.1. *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 $TR^{-1} \bmod m$ zove 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 množenje se može provesti bez modularne redukcije modulo m. Naravno, modularnu redukciju trebamo da bi 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:

$q_0 = \lfloor x/b^n \rfloor$; $r_0 = x - q_0 b^n$; $r = r_0$; $i = 0$
while $(q_i > 0)$ {
    $q_{i+1} = \lfloor q_i a/b^n \rfloor$; $r_{i+1} = q_i a - q_{i+1} b^n$;
    $i = i + 1$; $r = r + r_i$ }
while $(r \geq p)$ $r = r - p$

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 bi x^n izračunali kao $x \cdot x \cdots x$ pomoću $n - 1$ množenja vrlo je neefikasan.

Najjednostavnija i najstarija među efikasnim metodama je tzv. 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 $(1\,1\,0\,1)_2$. Sada x^{13} možemo izračunati kao

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

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

Binarna metoda (s desna na lijevo):

$z = 1$; $y = x$
for $(0 \le i \le d-1)$ {
    if $(n_i = 1)$ then $z = z \cdot y$
    $y = y^2$ }
$z = z \cdot y$

Binarna metoda (s lijeva na desno):

$z = x$
for $(d-1 \ge i \ge 0)$ {
    $z = z^2$
    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 $\le d+1$, a u prosječnom slučaju je oko $d/2$) Dakle, u slučaju modularnog potenciranja složnost 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, jedna ideja za moguće poboljšanje je korištenje tzv. "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

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

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 do sada 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 Montromeryjeve 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).

Montgomeryjevo potenciranje

$y = \text{Mont}(x, R^2 \bmod m)$; $z = R \bmod m$
for $(d \geq i \geq 0)$ {
    $z = \text{Mont}(z, z)$
    if $(n_i = 1)$ then $z = \text{Mont}(z, y)$ }
$z = \text{Mont}(z, 1)$

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

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

Za primjene u kriptografiji, pored grupe $\mathbb{Z}_m^*$, jedna od najvažnijih je grupa točaka na eliptičkoj krivulji nad konačnim poljem, uz operaciju zbrajanja točaka. Jedna od specifičnosti ove grupe je da u njoj inverzna operacija (oduzimanje) nije nimalo kompliciranija od originalne grupovne operacije (zbrajanja). Ova činjenica se može iskoristiti za efikasnije potenciranje (kod aditivnog zapisa ćemo govoriti – multipliciranje). Glavna ideja je zamjena binarnog zapisa sa zapisom u kojem su dopuštene znamenke -1, 0, 1. Prikaz broja n u obliku $n = \sum_{i_0}^{d} s_i 2^i$, $s_i \in \{-1, 0, 1\}$, zovemo *SD (signed digit)*

prikaz od n. Jasno je da SD prikaz nije jedinstven. Naime, imamo 3^{d+1} kombinacija, a samo $2^{d+1}-1$ brojeva koji se mogu prikazati s $d+1$ znamenkom. Npr. 3 = (0 1 1) = (1 0 -1). Ova višeznačnost nam sugerira da pokušamo izabrati prikaz koji će imati što više nula, a to će rezultirati efikasnijim multipliciranjem.

Reći ćemo da je SD prikaz *rijedak* ili *nesusjedan* (non-adjacent form, kraće: NAF prikaz) ako nema susjednih znamenaka različitih od 0, tj. ako je $s_i s_{i+1} = 0$ za svaki i. Može se pokazati da svaki prirodan broj n ima jedinstveni NAF prikaz. Nadalje, NAF ima najmanju težinu (broj znamenki različitih od 0) među svim SD prikazima od n, a najviše za jednu znamenku je dulji od najkraćeg SD prikaza od n.

Očekivana (prosječna) težina NAF prikaza je $d/3$, za razliku od binarnog prikaza kod kojeg je očekivana težina $d/2$.

Sljedeći algoritam iz poznatog binarnog zapisa $(n_{d-1}, \ldots, n_0)_2$ broja n računa njegov NAF prikaz $(s_d, \ldots, s_0)$.

Algoritam za NAF prikaz

$c_0 = 0$
for $(0 \le i \le d)$ {
    $c_{i+1} = \lfloor (n_i + n_{i+1} + c_i)/2 \rfloor$
    $s_i = n_i + c_i - 2c_{i+1}$ }

Umjesto formula iz ovog algoritma, možemo koristiti sljedeću tablicu koja za sve moguće vrijednosti ulaznih podataka u i-tom koraku (n_i, c_i, n_{i+1}) daje odgovarajuće vrijednosti izlaznih podataka (c_{i+1}, s_i).

n_i	0	0	0	0	1	1	1	1
c_i	0	0	1	1	0	0	1	1
n_{i+1}	0	1	0	1	0	1	0	1
c_{i+1}	0	0	0	1	0	1	1	1
s_i	0	0	1	-1	1	-1	0	0

Sve metode za potenciranje zasnovane na binarnom prikazu, mogu se jednostavno modificirati za NAF prikaz. Prikažimo to za binarnu metodu (s lijeva na desno).

Binarna metoda s predznakom (aditivna verzija):

$z = x$
for $(d-1 \ge i \ge 0)$ {
    $z = z + z$
    if $(n_i = 1)$ then $z = z + x$
    if $(n_i = -1)$ then $z = z - x$ }

2.4 Euklidov algoritam

Razmotrit ćemo sada problem nalaženja najvećeg zajedničkog djelitelja dva cijela broja a i b, u oznaci (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

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

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 smo već susreli kod RSA kriptosustava, gdje su eksponenti d i e povezani relacijom $de \equiv 1 \pmod{\varphi(n)}$.) Veza između ova dva problema dolazi preko sljedećeg teorema.

Teorem 2.2. *Postoje cijeli brojevi x, y takvi da je $ax + by = (a, b)$.*

Dokaz: Neka je g najmanji prirodan broj oblika $ax + by$, $x, y \in \mathbb{Z}$. Tvrdimo da je $g = (a, b)$. Jasno je da svaki zajednički djelitelj od a i b dijeli $ax + by = g$. Stoga $(a, b) | g$. Pretpostavimo da $g \nmid a$. Tada je $a = qg + r$, $0 < r < g$. No, $r = (1 - qx)a - qyb$, pa smo dobili kontradikciju s minimalnošću od g. Dakle, $g | a$ i sasvim isto zaključujemo da $g | b$. Stoga je $g \leq (a, b)$, pa zaključujemo da je $g = (a, b)$. □

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 $(a, b) = (b, a \bmod b)$.

Kako je $(a, b) = (|a|, |b|)$, možemo pretpostaviti da $a > b \geq 0$.

Euklidov algoritam:

while $(b > 0)$ $(a, b) = (b, a \bmod b)$

return a

Da bi 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 Fibonaccijevi 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 $\lfloor F_{k+1}/F_k \rfloor$ i $F_{k+1} \bmod F_k = F_{k-1}$, to su svi kvocijenti u Euklidovom 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

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

slijedi

Teorem 2.3. *Neka su $a, b \leq N$. Tada je broj koraka u Euklidovom algoritmu za računanje (a, b) manji ili jednak*

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

Može se pokazati da je prosječan broj koraka u Euklidovom 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 Euklidovog algoritma zahtjeva jedno dijeljenje brojeva $\leq N$, dobivamo da je složenost Euklidovog algoritma $O(\log^3 N)$. 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 Euklidovom algoritmu).

Euklidov algoritam se može iskoristiti i za nalaženje cijelih brojeva x, y takvih da je $ax + by = (a, b)$. Dakle, možemo ga koristiti za rješavanje linearnih Diofantskih jednadžbi. Kao što smo već vidjeli, u slučaju da je $(a, b) = 1$ na taj se način može dobiti modularni inverz.

Prošireni Euklidov algoritam:

$(x, y, g, u, v, w) = (1, 0, a, 0, 1, b)$;
while$(w > 0)$ {
    $q = \lfloor g/w \rfloor$;
    $(x, y, g, u, v, w) = (u, v, w, x - qu, y - qv, g - qw)$ }
return (x, y, g)

Prikazat ćemo još jedan algoritam za računanje najvećeg zajedničkog djelitelja, tzv. "binarni gcd algoritam". Kod njega se umjesto dijeljenja koriste samo operacije oduzimanja i pomaka (dijeljenja sa 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 Euklidovom 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 Euklidovom algoritmu.

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

Binarni gcd algoritam:

$\beta = \min\{v_2(a), v_2(b)\}$
$a = a/2^{v_2(a)}$; $b = b/2^{v_2(b)}$
while$(a \neq b)$
    $(a, b) = (\min\{a, b\}, |b - a|/2^{v_2(b-a)})$
return $2^\beta a$

2.5 Kineski teorem o ostatcima

Kineski teorem o ostatcima govori o rješenju sustava linearnih kongruencija. Ime mu se vezuje uz kineskog matematičara iz prvog stoljeća Sun-Tsua. Smatra se da je taj teorem korišen već u to vrijeme u kineskoj vojsci za prebrojavanje vojnika. Recimo 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 2.4. *Neka su $m_1, \ldots, m_k$ u parovima relativno prosti prirodni brojevi, tj. $(m_i, m_j) = 1$ 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 $(M_i, m_i) = 1$, to pomoću Euklidovog 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 to da on omogućava da se računanje po jednom velikom modulu zamjeni 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 situaciji dobro je 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).

Garnerov algoritam za CRT:

for $(1 \leq i \leq k-1)$ {
  $\mu_i = \prod j = 1^i m_j$;
  $c_i = \mu_i^{-1} \bmod m_{i+1}$ }
$M = \mu_{k-1} m_k$

$x = x_1$
for $(1 \leq i \leq k-1)$ {
  $y = ((x_{i+1} - x)c_i) \bmod m_{i+1}$;
  $x = x + y\mu_i$ }
$x = x \mod M$

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 Euklidovog 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.

Induktivni algoritam za CRT:

$m = m_1$; $x = x_1$
for $(2 \leq i \leq k)$ {
  nađi u, v takve da je $um + vm_i = 1$;
  $x = umx_i + vm_ix$;
  $m = mm_i$;
  $x = x \bmod m$ }

Možda je zanimljivo za spomenuti da je "obrnuti" problem od onog koji se razmatra u Kineskom teoremu o ostatcima puno teži od originalnog problema. Preciznije, može se pokazati da je problem odluke "Za danih k parova $(x_1, m_1), \ldots, (x_k, m_k)$, odrediti postoji li x takav da $x \not\equiv x_i \pmod{m_i}$ za $i = 1, \ldots, k$." NP-potpun.

2.6 Verižni razlomci

U ovom poglavlju vidjet ćemo još jednu primjenu Euklidovog algoritma. Iz prvog koraka u Euklidovom 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 a/b u jednostavni verižni razlomak*. Općenito, za $\alpha \in \mathbb{R}$ se prikaz broja α u obliku

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

gdje je $a_0 \in \mathbb{Z}$, te $a_1, a_2, \ldots \in \mathbb{N}$, zove *razvoj broja α u jednostavni verižni (ili neprekidni) razlomak*. Verižni razlomak kraće zapisujemo kao $[a_0; a_1, a_2, \ldots]$. Brojevi $a_0, a_1, a_2, \ldots$ se zovu *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 izračunati pomoću Euklidovog algoritma. Kod iracionalnog broja, mi najčešće znamo samo njegovu racionalnu aproksimaciju (obično binarni ili decimalni zapis s fiksnim brojeva točnih znamenaka). To znači da će i u njegovom razvoju biti točni samo neki početni a_i-ovi.

Primjer 2.4. Izračunati razvoj u jednostavni verižni razlomak broja $\sqrt{101}$, ali tako da u svim računima koristimo samo 10 najznačajnijih decimalnih znamenaka. Dobije se

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

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

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

Željeli bi 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.

Razvoj u verižni razlomak

$i = 0$
$q = \lfloor a/b \rfloor$; $r = a - bq$; $r' = a' - b'q$

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

U slučaju kada je α kvadratna iracionalnost, tj. iracionalan broj koji je rješenje neke kvadratne jednadžbe s racionalnih 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 $d, s_0, t_0 \in \mathbb{Z}$, $t_0 \ne 0$, 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 računamo rekurzivno na sljedeći način:

$$a_i = \lfloor \alpha_i \rfloor, \quad s_{i+1} = a_i t_i - s_i, \quad t_{i+1} = \frac{d - s_{i+1}^2}{t_i}, \quad \alpha_{i+1} = \frac{s_{i+1} + a_0}{t_{i+1}}.$$

Uočimo da iako je α iracionalan broj, ovaj algoritam radi samo s cijelim brojevima. 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 reći i preciznije 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 palidromni, tj. $a_i = a_{r-i}$ za $i = 1, 2, \ldots, r-1$. Na primjer,

$$\sqrt{101} = [10; \overline{20}], \quad \sqrt{13} = [3; \overline{1, 1, 1, 1, 6}], \quad \sqrt{113} = [10; \overline{1, 1, 1, 2, 2, 1, 1, 1, 20}].$$

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

Racionalne brojeve

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

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

$$\begin{aligned} p_0 = a_0, \quad p_1 = a_0a_1 + 1, \quad p_{k+2} &= a_{k+2}p_{k+1} + p_k, \\ q_0 = 1, \quad q_1 = a_1, \quad q_{k+2} &= a_{k+2}q_{k+1} + q_k. \end{aligned}$$

Indukcijom se lako dokazuje sljedeća važna relacija:

$$q_k p_{k-1} - p_k q_{k-1} = (-1)^k. \tag{2.3}$$

Relacija (2.3) 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. 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.

Konvergente se javljaju i kod rješavanja nekih diofantskih jednadžbi. Posebno je važna njihova uloga u rješavanju *Pellovih jednadžbi*. To su jednadžbe oblika $x^2 - dy^2 = 1$, gdje je d prirodan broj koji nije potpun kvadrat. Često se uz ovu jednadžbu promatra i jednadžba $x^2 - dy^2 = -1$. Ako sa $\frac{p_k}{q_k}$ označimo konvergente u razvoju u verižni razlomak broja $\sqrt{d}$, onda vrijedi

$$p_k^2 - dq_k^2 = (-1)^{k+1} t_{k+1}\,,$$

gdje je niz (t_k) definiran u algoritmu za razvoj od $\sqrt{d}$. Odavde se lako dobije da jednadžba $x^2 - dy^2 = 1$ uvijek ima (beskonačno) rješenja u prirodnim brojevima, dok jednadžba $x^2 - dy^2 = -1$ ima rješenja ako i samo ako je duljina perioda r u razvoju od $\sqrt{d}$ neparna. Ako je (X, Y) najmanje rješenje u prirodnim brojevima jednadžbe $x^2 - dy^2 = 1$, onda je $(X, Y) = (p_{r-1}, q_{r-1})$ ili (p_{2r-1}, q_{2r-1}) u ovisnosti o tome je li duljina perioda r parna ili neparna.

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 prosti broj p oblika $4k+1$ naći cijele brojeve x, y takve da je $x^2 + y^2 = p$. Pokazat ćemo dvije konstrukcije koje obje koriste verižne razlomke.

1. konstrukcija (Hermite)

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 $\frac{p_{n+1}}{q_{n+1}}$, to je

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

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 $(zq_n - pp_n)^2 + q_n^2 = p$.

2. konstrukcija (Legendre)

Promotrimo Pellovu jednadžbu $x^2 - py^2 = 1$. Kao što smo već rekli, ona ima beskonačno mnogo rješenja u prirodnim brojevima. Neka je (X, Y) najmanje takvo rješenje. Iz $(X+1)(X-1) = pY^2$ slijedi da je $X+1 = ab^2p$, $X-1 = ac^2$ ili $X+1 = ab^2$, $X-1 = ac^2p$, gdje je $a = (X+1, X-1) = 1$ ili 2, a b, c su neki prirodni brojevi. Odavde je $c^2 - pb^2 = -\frac{2}{a}$ ili $b^2 - pc^2 = \frac{2}{a}$. Zbog minimalnosti od (X, Y), otpada mogućnost $a = 2$ u drugoj jednadžbi. Dakle, imamo

$$c^2 - pb^2 = -1, \quad c^2 - pb^2 = -2 \quad \text{ili} \quad b^2 - pc^2 = 2.$$

Ako je $p \equiv 1 \pmod 4$, onda je $c^2 - pb^2 \equiv 0, 1$ ili $3 \pmod 4$, pa stoga mora vrijediti $c^2 - pb^2 = -1$ jer su druga i treća jednadžba nemoguće modulo 4. No, nužan i dovoljan uvjet da bi jednadžba $c^2 - pb^2 = -1$ imala rješenja je da period u razvoju u verižni razlomak broja $\sqrt{p}$ bude neparan. Dakle, imamo

$$\sqrt{p} = [a_0; \overline{a_1, \ldots, a_n, a_n \ldots, a_1, 2a_0}].$$

Broj $\alpha_{n+1} = \frac{s_{n+1}+\sqrt{p}}{t_{n+1}}$ je čisto periodičan (nema pretperioda) i period mu je palidroman. Neka je $\alpha'_{n+1} = \frac{s_{n+1}-\sqrt{p}}{t_{n+1}}$ njegov konjugat. Nije teško za vidjeti da je razvoj od $-\frac{1}{\alpha'_{n+1}}$ također čisto periodičan, s time da se parcijalni kvocijenti unutar perioda pojavljuju u obrnutom redosljedu od onih kod α_{n+1}. Kako je period od α_{n+1} palidroman, zaključujemo da je $-\frac{1}{\alpha'_{n+1}} = \alpha_{n+1}$. Stoga je

$$\alpha'_{n+1}\alpha_{n+1} = \frac{s_{n+1}^2 - p}{t_{n+1}^2} = -1,$$

tj. $p = s_{n+1}^2 + t_{n+1}^2$.

2.7 Kvadratne kongruencije

Definicija 2.8. 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.

S kongruencijom $x^2 \equiv a \pmod{m}$ i pitanjem ima li ona rješenje i ako ima, kako ih naći, susreli smo se ranije kod Rabinovog kriptosustava.

Pretpostavimo sada da je modul komgruencije 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 podsjetili definicije primitivnog korijena, recimo nešto o tome kako se on nalazi. Možemo krenuti redom i testirati da li je $g = 2$, $g = 3$, ... 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. Samo testiranje da li je g primitivni korijen se zasniva na sljedećoj očitoj činjenici: g je primitivan 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 primitivni 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 $\prod_p \left(1 - \frac{1}{p(p-1)}\right) \approx 0.3739558$. Poznato je (Hooley, 1967) da tzv. generalizirana Riemannova

slutnja (GRH) povlači Artinovu slutnju. GRH također povlači da je najmanji primitivni korijen modulo p reda veličine $O(\ln^6 p)$.

Vratimo se sada kvadratnim ostatcima modulo p.

Definicija 2.9. 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 tzv. *Eulerovog 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 Euklidovom algoritmu. Taj algoritam je zasnovan na tzv. *Gaussovom 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 sa $(\frac{q}{p})$, 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 Legendreovog simbola da jedan od parametara (donji) bude prost. To nas vodi do potrebe uvođenja poopćenja Legendreovog simbola kod kojeg parametri neće morati biti prosti.

Definicija 2.10. Neka je m neparan prirodan broj i $m = \prod p_i^{\alpha_i}$ njegov rastav na proste faktore, te neka je a proizvoljan cijeli broj. *Jacobijev simbol* $(\frac{a}{m})$ se definira sa

$$\left(\frac{a}{m}\right) = \prod \left(\frac{a}{p_i}\right)^{\alpha_i},$$

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

Jasno je da ako je m prost, onda se Jacobijev i Legendreov simbol podudaraju. Ako je $(a, m) > 1$, 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 jedanki 1. Spomenimo da postoji i općenitiji pojam, tzv. Kroneckerov simbol $(\frac{a}{b})$, koji se definira za proizvoljne cijele brojeve a i b (b može biti paran i negativan).

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

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

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 za računanje Jacobijevog simbola $(\frac{a}{m})$

$a = a \bmod m$
$t = 1$
while $(a \neq 0)$ {
    while (a paran) {
        $a = a/2$;
        if $(m \equiv 3, 5 \pmod 8)$ then $t = -t$ }
    $(a, m) = (m, a)$
    if $(a \equiv m \equiv 3 \pmod 4)$ then $t = -t$
    $a = a \bmod m$ }
if $(m = 1)$ then return t
    else return 0

Vidimo da je ovaj algoritam vrlo sličan Euklidovom algoritmu. Jedina bitna razlika je u posebnom tretiranju faktora 2, kojeg moramo izlučiti prije nego što zamjenimo gornji i donji parametar.

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

$$x^2 \equiv a \pmod p \,. \tag{2.4}$$

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. Zapravo, mogli bi reći da taj oblik i nije jako specijalan, budući da pola prostih brojeva ima takav oblik.

Propozicija 2.5. *Ako je $p \equiv 3 \pmod 4$, onda je $x = a^{(p+1)/4}$ rješenje kongruencije (2.4).*

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

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

□

Prethodnu propoziciju je 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 ta činjenica se koristi u sljedećoj propoziciji.

Propozicija 2.6. *Ako je $p \equiv 5 \pmod 8$, onda je jedan brojeva $a^{(p+3)/8}$ i $2^{(p-1)/4}a^{(p+3)/8}$ rješenje kongruencije (2.4).*

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 (2.4).

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

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

□

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 na nalaženje kvadratnog neostatka. Pretpostavimo li da vrijedi tzv. 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 Eulerovom 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 (2.4).

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

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 $(0 \leq i \leq s-1)$ {
    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 pokazali dvije konstrukcije za rješavanje problema zbroja kvadrata. Sada ćemo navesti jednu modifikaciju prve konstrukcije, tzv. 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 tzv. broj klasa $h(-4d)$ jednak 1, tj. kada su sve binarne kvadratne forme s distriminantom $-4d$ međusobno ekvivalentne). Neka je dakle z cijeli broj takav da je

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

i neka je $\frac{p}{2} < z < p$. Primijenjujemo 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.

Cornacchia - Smithov algoritam

if $((\frac{-d}{p}) = -1)$ then return *nema rješenja*
$z = \sqrt{-d} \bmod p$;
if $(2z < p)$ then $z = p - z$
$(a, b) = (p, z)$
$c = \lfloor \sqrt{p} \rfloor$
$t = p - b^2$
if $(t \not\equiv 0 \pmod{d})$ then return *nema rješenja*
if $(t/d$ nije potpun kvadrat$)$ then return *nema rješenja*
return $(b, \sqrt{t/d})$

2.8 Kvadrati i kvadratni korijeni

U ovom poglavlju razmatramo pitanje kako za dani prirodni broj n što efikasnije odrediti da li je 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-Smithovom algoritmu.

Algoritam za računanje $\lfloor \sqrt{n} \rfloor$ je zapravo varijanta poznate Newtonove iterativne metode za približno računaje korijena jednadžbe. Podsjetimo se na se u Newtovoj metodi aproksimacije rješenja jednadžbe $f(x) = 0$ računanju 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 za $\lfloor \sqrt{n} \rfloor$

$x = n$
$y = \lfloor (x + \lfloor n/x \rfloor)/2 \rfloor$
while $(y < x)$ {
    $x = y$; $y = \lfloor (x + \lfloor n/x \rfloor)/2 \rfloor$ }
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 da li je n potpun kvadrat ili nije. Jedna mogućnost je izračunati $\lfloor \sqrt{n} \rfloor$ 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 relativno prost s n. Dakle, ako je n kvadratni neostatak po nekom modulu, onda n sigurno nije kvadrat. Ova ideja se 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:

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

Jedna preporučena kombinacije 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 bi 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 za detekciju kvadrata

$t = n \bmod 64$
if $(q64[t] = 0)$ then return n *nije kvadrat*
$r = n \mod 45045$
if $(q63[r \bmod 63] = 0)$ then return n *nije kvadrat*
if $(q65[r \bmod 65] = 0)$ then return n *nije kvadrat*
if $(q11[r \bmod 11] = 0)$ then return n *nije kvadrat*
$q = \lfloor \sqrt{n} \rfloor$ if $(n \neq q^2)$ them return n *nije kvadrat*
else return n *je kvadrat;* $\sqrt{n} = q$

Poglavlje 3

Eliptičke krivulje

3.1 Grupovni zakon

Kada smo govorili o kriptosustavima zasnovanima na problemu diskretnog logaritma, spomenili smo da je grupa točaka na eliptičkoj krivulji nad konačnim poljem jedna od najvažnijih grupa koje se koriste u kriptografiji. Pored toga, eliptičke krivulje igraju važnu ulogu kod faktorizacije, te dokazivanja prostosti, što su također važne teme u kriptografiji.

Neka je K polje. Općenito, eliptička krivulja nad K je nesingularna kubna krivulja. Dakle, ona ima jednadžbu oblika

$$F(x,y) = ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + ix + j = 0,$$

gdje su koeficijenti $a, b, c, \ldots, j \in K$, a nesingularnost znači da je u svakoj točki na krivulji $F(x,y) = 0$, promatranoj nad algebarskim zatvorenjem od K, barem jedna parcijalna derivacija funkcije F različita od 0. Svaka takva jednadžba može se biracionalnim transformacijama (racionalnim transformacijama čiji je inverz također racionalna transformacija) svesti na oblik

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

koji zovemo *Weierstrassova forma*. Nadalje, ako je karakteristika polja K različita od 2 i 3 (tj. $1+1 \neq 0$ i $1+1+1 \neq 0$), onda se ova jednadžba može transformirati u oblik

$$y^2 = x^3 + ax + b,$$

koji zovemo *kratka Weierstrassova forma*. Uvjet nesingularnosti je sada da kubni polinom $f(x) = x^3 + ax + b$ nema višestrukih nultočaka (u $\overline{K}$), a to je pak ekvivalentno uvjetu da je *diskriminanata* $D = -4a^3 - 27b^2$ različita od 0.

U daljnjem će za nas *eliptička krivulja* nad pojem K (karakteristike različite od 2 i 3) biti skup svih točaka $(x,y) \in K \times K$ koji zadovoljavaju jednadžbu

$$E: \quad y^2 = x^3 + ax + b,$$

gdje su $a, b \in K$ i $4a^3 + 27b^2 \neq 0$, zajedno s "točkom u beskonačnosti" $\mathcal{O}$. Taj skup ćemo označavati s $E(K)$.

Točka u beskonačnosti se pojavljuje prirodno ukoliko eliptičku krivulju prikažemo u projektivnoj ravnini. *Projektivnu ravninu* $\mathbb{P}^2(K)$ dobijemo tako da na skupu $K^3 \setminus \{(0,0,0)\}$ uvedemo relaciju ekvivalencije $(X, Y, Z) \sim (kX, kY, kZ)$, $k \in K$, $k \neq 0$. Ako u (afinoj) jednadžbi eliptičke krivulje uvedemo supstituciju $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$, dobivamo projektivnu jednadžbu

$$Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Ako je $Z \neq 0$, onda klasa ekvivalencije od (X, Y, Z) ima reprezentant $(x, y, 1)$, pa tu klasu možemo identificirati sa (x, y). Međutim, postoji i jedna klasa ekvivalencije koja sadrži točke za koje je $Z = 0$. Ona ima reprezentant $(0, 1, 0)$ i tu klasu identificiramo s točkom u beskonačnosti $\mathcal{O}$.

Kao što smo već rekli u poglavlju 1.3, operacija (zbrajanje) na skupu $E(\mathbb{R})$ se uvodi "geometrijski", tako da su tri točke na krivulji E kolinearne ako i samo ako im je suma jednaka neutralnom elementu $\mathcal{O}$. Naravno da se ovaj geometrijski zakon može opisati i eksplicitnim formulama za koordinate zbroja točaka. Tako dobivene formule onda mogu poslužiti za definiciju zbrajanja točaka na eliptičkoj krivulji nad proizvoljnim poljem (uz malu modifikaciju ako je karakteristika polja 2 ili 3). Navedimo sada te formule:

Neka je $P = (x_1, y_1)$, $Q = (x_2, y_2)$. Tada je

1) $-\mathcal{O} = \mathcal{O}$
2) $-P = (x_1, -y_1)$
3) $\mathcal{O} + P = P$
4) ako je $Q = -P$, onda je $P + Q = \mathcal{O}$
5) ako je $Q \neq -P$, onda je $P + Q = (x_3, y_3)$, gdje je

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = -y_1 + \lambda(x_1 - x_3),$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{ako je } x_2 \neq x_1, \\ \frac{3x_1^2 + a}{2y_1}, & \text{ako je } x_2 = x_1. \end{cases}$$

Broj λ je "koeficijent smjera" pravca kroz P i Q, odnosno tangente u točki P u slučaju $P = Q$.

Pokazuje se da je $(E(K), +)$ abelova grupa. Sva svojstva abelove grupe su evidentna, osim asocijativnosti čiji je dokaz nešto kompliciraniji.

Za primjene u kriptografiji, najvažniji je slučaj kada je K konačno polje $\mathbb{F}_q$. Posebno su važni slučajevi $q = p$ (prost broj) i $q = 2^k$. S druge strane, u teoriji brojeva najvažniju ulogu imaju eliptičke krivulje nad poljem racionalnih brojeva $\mathbb{Q}$.

3.2 Eliptičke krivulje nad $\mathbb{Q}$

Najvažija činjenica o eliptičkim krivuljama nad $\mathbb{Q}$ jest Mordell-Weilov teorem.

Teorem 3.1 (Mordell-Weil). *$E(\mathbb{Q})$ je konačno generirana abelova grupa.*

Mordell-Weilov teorem nam, drugim riječima, kaže da postoji konačan skup racionalnih točaka $P_1, \ldots, P_k$ na E iz kojih se sve ostale racionalne točke na E mogu dobiti povlačeći sekante i tangente. Kako je svaka konačno generirana abelova grupa izomorfna produktu cikličkih grupa, dobivamo sljedeću neposrednu posljedicu Mordell-Weilovog teorema.

Korolar 3.2.

$$E(\mathbb{Q}) \cong E(\mathbb{Q})_{tors} \times \mathbb{Z}^r$$

Podgrupa $E(\mathbb{Q})_{\text{tors}}$ od $E(\mathbb{Q})$ koja se sastoji od svih točaka konačnog reda naziva se *torzijska grupa* od E, a nenegativni cijeli broj r se naziva *rang* od E i označava se s $\text{rank}(E)$. Korolar nam kaže da postoji r racionalnih točaka $P_1, \ldots, P_r$ na krivulji E sa svojstvom da se svaka racionalna točka P na E može prikazati u obliku

$$P = T + m_1 P_1 + \cdots + m_r P_r,$$

gdje je T neka točka konačnog reda, a $m_1, \ldots, m_r$ cijeli brojevi. Ovdje $m_1 P_1$ označava sumu $P_1 + \cdots + P_1$ od m_1 pribrojnika, koja se često označava i sa $[m_1]P_1$.

Postavlja se pitanje koje sve vrijednosti mogu poprimiti $E(\mathbb{Q})_{\text{tors}}$ i $\text{rank}(E)$. Nadalje, pitanje je kako ih izračunati za konkretnu krivulju E. Pokazuje se da je puno lakše dati odgovore na ova pitanja za torzijsku grupu, nego za rang.

Mazur je 1978. godine dokazao da postoji točno 15 mogućih torzijskih grupa. To su grupe:

$$\begin{aligned} &\mathbb{Z}_n, \quad \text{za } n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 \\ &\mathbb{Z}_2 \times \mathbb{Z}_n, \quad \text{za } n = 2, 4, 6, 8. \end{aligned}$$

Točke reda 2 su upravo točke s y-koordinatom jednakom 0. Možemo imati 0, 1 ili 3 takve točke, što ovisi o broju racionalnih nultočaka polinoma $x^3 + ax + b$. Te točke, zajedno s točkom $\mathcal{O}$, čine podgrupu od $E(\mathbb{Q})_{\text{tors}}$ koje je ili trivijalna ili jednaka $\mathbb{Z}_2$ ili jednaka $\mathbb{Z}_2 \times \mathbb{Z}_2$. Ostale točke konačnog reda možemo naći pomoću sljedećeg teorema.

Teorem 3.3 (Lutz-Nagell). *Neka je eliptička krivulja E zadana jednadžbom*

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{Z}.$$

Ako je $P = (x, y) \in E(\mathbb{Q})_{tors}$, onda su x, y cijeli brojevi, te vrijedi da je ili $y = 0$ ili y^2 dijeli diskriminantu D.

Kao što smo već napomenuli, pitanja koja se tiču ranga su puno teža, a zadovoljavajući odgovori još uvijek nisu poznati. Vjeruje se da rang može biti proizvoljno velik, tj. da za svaki $M \in \mathbb{N}$ postoji eliptička krivulja nad $\mathbb{Q}$ takva da je $\text{rank}(E) \geq M$. No, danas se tek zna da postoji eliptička krivulja ranga ≥ 24. Tu su krivulju pronašli Martin i McMillen 2000. godine.

Pitanje koliko velik može biti rang je od interesa i za primjene eliptičkih krivulja u kriptografiji. Naime, za problem diskretnog logaritma u grupi $\mathbb{Z}_p^*$ postoji subeksponencijalni algoritam, tzv. Index Calculus metoda. U toj metodi se koristi ulaganje od $\mathbb{Z}_p^*$ u prsten $\mathbb{Z}$, te činjenica da prostih brojeva (generatora od $\mathbb{Z}$) ima "mnogo". Kad bi mogli naći eliptičke krivulje jako velikog ranga, onda bi na problem diskretnog logaritma u grupi $E(\mathbb{F}_p)$ mogli pokušati primijeniti analogon Index Calculus metode tako što bi tu krivulje uložili u krivulju nad $\mathbb{Q}$ velikog ranga, čiji bi generatori igrali ulogu prostih brojeva u originalnoj Index Calculus metodi. Procjenjuje se da bi za proste brojeve p koji su danas u uporabi u kriptografiji ($\approx 2^{160}$) trebali koristiti krivulje ranga većeg od 180.

Skoro sve ideje za konstrukciju eliptičkih krivulja (relativno) velikog ranga zasnivaju se na idejama koje je uveo Jean-Francois Mestre. Prikazat ćemo jednu njegovu konstrukciju kojom je 1991. godine dobio beskonačno mnogo eliptičkih krivulja ranga ≥ 11. Ta konstrukcija se obično naziva *Mestreova polinomijalna metoda*. Polazište u konstrukciji je sljedeća činjenica.

Lema 3.4. *Neka je $p(x) \in \mathbb{Q}[x]$ normiran polinom i $\deg p = 2n$. Tada postoje jedinstveni polinomi $q(x), r(x) \in \mathbb{Q}[x]$ takvi da je $p = q^2 - r$ i $\deg r \leq n - 1$.*

Polinom q možemo naći sukcesivnim računanjem koeficijenata ili iz asimptotskog razvoja od $\sqrt{p}$.

Pretpostavimo sada da je $p(x) = \prod_{i=1}^{2n}(x - a_i)$, gdje su $a_1, \ldots, a_{2n}$ različiti racionalni brojevi. Tada na krivulji

$$C: \quad y^2 = r(x)$$

leže točke $(a_i, \pm q(a_i))$, $i = 1, \ldots, 2n$. Ako je $\deg r = 3$ ili 4, te $r(x)$ nema višestrukih korijena, onda C predstavlja eliptičku krivulju. Za $\deg r = 3$ to je sasvim jasno. Ako je $\deg r = 4$, onda izaberemo jednu racionalnu točku na C (npr. $(a_1, q(a_1))$) za točku u beskonačnosti i transformiramo C u eliptičku krivulju. Pokažimo to na primjeru krivulje oblika $y^2 = ax^4 + bx^3 + cx^2 + dx$ koja ima racionalnu točku $(0, 0)$. Uz supstituciju $x = 1/t$, $v = t^2 y$, dobivamo $v^2 = dt^3 + ct^2 + bt + a$. Množenjem sa d^2, dobivamo eliptičku krivulju u Weierstrassovoj formi $V^2 = T^3 + cT^2 + bdT + ad^2$.

Za $n = 5$ skoro svi izbori a_i-ova daju $\deg r = 4$. Tada C ima 10 racionalnih točaka oblika $(a_i, q(a_i))$ i možemo očekivati da ćemo dobiti eliptičku krivulju ranga ≥ 9. Meste je konstruirao familiju eliptičkih krivulja (tj. eliptičku krivulju nad poljem racionalnih funkcija $\mathbb{Q}(t)$) ranga ≥ 11, tako

da je uzeo $n = 6$ i $a_i = b_i + t$, $i = 1, \ldots 6$; $a_i = b_{i-6} - t$, $i = 7, \ldots, 12$. Sada polinom $r(x)$ općenito ima stupanj 5. Zato možemo pokušati izabrati brojeve $b_1, \ldots, b_6$ tako da koeficijent uz x^5 bude jednak 0. U prvom Mestreovom primjeru iz 1991. godine bilo je $b_1 = -17$, $b_2 = -16$, $b_3 = 10$, $b_4 = 11$, $b_5 = 14$, $b_6 = 17$.

Kasnije su Mestre, Nagao i Kihara, koristeći slične konstrukcije, poboljšali ovaj rezultat, tako da je danas rekord za rang nad $\mathbb{Q}(t)$ jednak 14. Sve ove krivulje imaju trivijalnu torzijsku grupu. Fermigier i Kulesz su modificirali Mestreovu metodu, te dobili familije krivulja s (relativno) velikim rangom i netrivijalnom torzijskom grupom.

3.3 Eliptičke krivulje nad konačnim poljima

Za primjene u kriptogtafiji najvažije su eliptičke krivulje nad konačnim poljima. Konačno polje s q elemenata označava se s $\mathbb{F}_q$. Važni primjeri konačnih polja su polja $\mathbb{F}_p$, gdje je p prost broj. To su upravo polja $\mathbb{Z}_p$ ostataka modulo p. Neka je karakteristika polja $\mathbb{F}_q$ jednaka p. Tada $\mathbb{F}_q$ sadrži prosto potpolje $\mathbb{F}_p$, i $\mathbb{F}_q$ je vektorski prostor nad poljem $\mathbb{F}_p$. Ako s k označimo dimenziju od $\mathbb{F}_q$ kao vektorskog prostora nad $\mathbb{F}_p$, onda je $q = p^k$. Pokazuje se da za svaku potenciju prostog broja $q = p^k$ postoji jedinstveno (do na izomorfizam) polje s q elemenata. Jedna realizacija tog polja je $\mathbb{Z}_p[x]/(f(x))$, gdje je $f(x)$ neki ireducibilni polinom stupnja k nad $\mathbb{Z}_p$. Elementi ovog polja su polinomi nad $\mathbb{Z}_p$ stupnja $\leq k - 1$, dok su pripadne operacije zbrajanje i množenje polinoma u $\mathbb{Z}_p[x]$, s time da se nakon množenja računa ostatak pri dijeljenju s polinomom $f(x)$.

Na ovom mjestu se možemo pitati kako naći ireducibilni polinom stupnja k nad $\mathbb{Z}_p$. Pokazuje se da normiranih ireducibilnih polinoma stupnja k nad $\mathbb{Z}_p$ ima približno p^k/k, tj. otprilike svaki k-ti normirani polinom stupnja k nad $\mathbb{Z}_p$ je ireducibilan. Testiranje da li je konkretni polinom ireducibilan zasniva se na činjenici da je polinom $f(x)$ stupnja k nad $\mathbb{Z}_p$ ireducibilan ako i samo ako je $\gcd(f(x), x^{p^j} - x) = 1$ za $j = 1, 2, \ldots \lfloor k/2 \rfloor$. Posljednji uvjet se provjerava Euklidovim algoritmom za polinome. Da bi operacije u polju $\mathbb{F}_q$ bile što efikasnije, obično se polinom $f(x)$ bira tako da ima što manju težinu W (broj koeficijenata različitih od 0). U slučaju $q = 2^k$, koje je najzanimljiviji za primjene u kriptografiji, čini se da je uvijek moguće postići da je $W = 3$ ili $W = 5$.

Neka je E eliptička krivulja nad konačnim poljem $\mathbb{F}_q$, $q = p^k$. Kao što smo već rekli, ako je $p > 3$, onda E ima jednadžbu oblika

$$y^2 = x^3 + ax + b.$$

Ako je $p = 3$, onda E ima jednadžbu oblika

$$y^2 = x^3 + ax^2 + bx + c,$$

a ako je $p = 2$, onda se E može transformirati u jedan od slijedeća dva oblika

$$y^2 + cy = x^3 + ax + b \quad \text{ili} \quad y^2 + xy = x^3 + ax^2 + b.$$

Postavlja se pitanje, što se može reći općenito o grupi $E(\mathbb{F}_q)$, tj. o njezinom redu $\#E(\mathbb{F}_q)$ i strukturi. Lako je zaključiti da je $\#E(\mathbb{F}_q) \in [1, 2q + 1]$. Naime, na E imamo točku $\mathcal{O}$, a pored toga svakom od q mogućih x-eva odgovaraju najviše dva y-na. No, samo pola elemenata od $\mathbb{F}_q$ imaju kvadratni korijen (to su elementi oblika g^{2n}, gdje je g generator (cikličke) grupe $\mathbb{F}_q^*$), pa možemo očekivati da je $\#E(\mathbb{F}_q) \approx q + 1$. Preciznu informaciju o redu grupe $E(\mathbb{F}_q)$ daje poznati Hasseov teorem.

Teorem 3.5 (Hasse).

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}$$

Veličina $t = q+1-\#E(\mathbb{F}_q)$ naziva se *Frobeniusov trag*. Prema Hasseovom teoremu je $|t| \leq 2\sqrt{q}$. Hasseov teorem nećemo dokazivati, ali ćemo pokušati dati heurističko objašnjenje za pojavu $\sqrt{q}$ u nejednakostima. Pretpostavimo da je $q = p$. Tada je

$$\#E(\mathbb{F}_p) = p + 1 + \sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p} \right). \tag{3.1}$$

Zamislimo da Legendreov simbol $(\frac{x^3+ax+b}{p})$ poprima vrijednosti 1 i -1 na slučajan način. Iz teorije vjerojatnosti (tzv. zakon ponovljenog logaritma) poznato je da ako je X_k niz nezavisnih slučajnih varijabli koje poprimaju vrijednosti ± 1, onda za njihovu sumu $t(n) = \sum_{k \leq n} X_k$ vrijedi (s vjerojatnošću 1) da je

$$\limsup_{n \to \infty} \frac{t(n)}{\sqrt{\frac{n}{2} \ln \ln n}} = 1.$$

Dakle, gornja ograda za $t(n)$ je "ugrubo" proporcionalna s $\sqrt{n}$, što se može usporediti s Hasseovim teoremom koji za gornju ogradu od t daje $2\sqrt{q}$.

Vrijedi i svojevrstan obrat Hasseovog teorema (Dueringov teorem) koji kaže da za svaki cijeli broj

$$m \in \langle p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p} \rangle$$

postoji eliptička krivulja nad $\mathbb{F}_p$ takva da je $\#E(\mathbb{F}_p) = m$.

U primjenama eliptičkih krivulja, često biramo eliptičke krivulje čiji red ima neko zadano aritmetičko svojstvo (prost je, ima samo male proste faktore, i sl.). Pritom je jako važna činjenica, koju je dokazao Lenstra, a koja kaže da su redovi $\#E(\mathbb{F}_p)$ za $(a, b) \in \mathbb{F}_p \times \mathbb{F}_p$ skoro uniformno distribuirani unutar intervala $\langle p - \sqrt{p}, p + \sqrt{p} \rangle$ (centralne polovice Hasseovog intervala). To znači da će red slučajno odabrane eliptičke krivulje nad $\mathbb{F}_p$ imati zadano

svojstvo s približno istom vjerojatnošću kao i slučajno odabran prirodan broj reda veličine kao p.

O mogućim redovima grupe $E(\mathbb{F}_q)$ u općem slučaju $q = p^k$, govori sljedeći teorem.

Teorem 3.6. *Neka je $q = p^k$. Tada postoji eliptička krivulja E nad $\mathbb{F}_q$ takva da je $\#E(\mathbb{F}_q) = q + 1 - t$ ako i samo ako je $|t| \leq 2\sqrt{q}$ i t zadovoljava jedan od uvjeta:*

1) $(t, p) = 1$

2) k *je paran i* $t = \pm 2\sqrt{q}$ *ili* $(t = \pm\sqrt{q}$ *i* $p \not\equiv 1 \pmod 3)$ *ili* $(t = 0$ *i* $p \not\equiv 1 \pmod 4)$

3) k *je neparan i* $t = 0$ *ili* $(t = \pm\sqrt{2q}$ *i* $p = 2)$ *ili* $(t = \pm\sqrt{3q}$ *i* $p = 3)$.

O strukturi grupa $E(\mathbb{F}_q)$ govori sljedeći teorem.

Teorem 3.7. *Neka je E eliptička krivulja nad $\mathbb{F}_q$. Tada je*

$$E(\mathbb{F}_q) \cong \mathbb{Z}_{n_1} \times \mathbb{Z}n_2,$$

gdje su n_1 i n_2 prirodni brojevi i vrijedi $n_1 | n_2$ i $n_1 | q - 1$.

Ako je $n_1 = 1$, onda je grupa $E(\mathbb{F}_q)$ ciklička. Iz uvjeta da $n_1 | (n_2, q - 1)$, zaključujemo da se može očekivati da će općenito n_1 biti mali prirodan broj, a grupa $E(\mathbb{F}_q)$ "skoro ciklička".

Recimo nešto o implimentaciji grupovne operacije na $E(\mathbb{F}_q)$. Zadržimo se na slučaju $q = p > 3$. Ako zboj dvije točke $P + Q$ računamo po formuli za zbrajanje točaka na krivulji zadanoj afinom jednadžbom, vidimo da trebamo jednom računati inverz, te imamo još četiri množenja u polju. Kod računanja $P + P$, ponovo imamo jedan inverz, te još četiri množenja (od kojih su dva kvadriranja). Znamo da se inverz može izračunati pomoću Euklidovog algoritma. Iako je njegova složenost teoretski ista kao složenost množenja, u praksi je množenje ipak znatno brže od računja inverza. Računanje inverza se može izbjeći korištenjem tzv. *težinskih projektivnih koordinata* u kojima projektivnoj točki (X, Y, Z) odgovara afina točka $(\frac{X}{Z^2}, \frac{Y}{Z^3})$ (prvoj koordinati smo dali težinu 2, a drugoj 3). Tada jednadžba eliptičke krivulje postaje

$$Y^2 = X^3 + aXZ^4 + bZ^6.$$

U ovim novim koordinatama se kod računanja zbroja točaka uopće ne pojavljuje dijeljenje. Zbroj $P + Q$ se može izračunati uz 16 množenja, a zbroj $P + P$ uz 10 množenja. Neka je $P = (X_1, Y_1, Z_1)$. Dat ćemo formule za računanje koordinata (X_2, Y_2, Z_2) točke $P + P$:

$$\lambda_1 = 3X_1^2 + aZ_1^4, \quad \lambda_2 = 4X_1Y_1^2, \quad \lambda_3 = 8Y_1^4,$$
$$X_2 = \lambda_1 - 2\lambda_2, \quad Y_2 = \lambda_1(\lambda_2 - x_2) - \lambda_3, \quad Z_2 = 2Y_1Z_1.$$

U primjenama eliptičkih krivulja često je potrebno izračunati višekratnik neke točke P, tj. točku

$$[m]P = \underbrace{P + P + \cdots + P}_{m \text{ pribrojnika}}.$$

To se može efikasno napraviti pomoću algoritama za potenciranje u abelovim grupama koje smo obradili u poglavlju 2.3. Posebno je pritom koristan SD prikaz broja m, budući da je grupa eliptičke krivulje primjer grupe u kojoj je invertiranje (računanje točke $-P$) vrlo jednostavno. Broj operacija za računanje $[m]P$ za eliptičku krivulju nad poljem $\mathbb{F}_q$ je $O(\ln m \ln^2 q)$.

3.4 Računanje reda grupe $E(\mathbb{F}_q)$

Hoće li konkretna eliptička krivulja biti prikladna za primjene u kriptografiji, ovisi prvenstveno o redu grupe $E(\mathbb{F}_q)$. Da bi problem diskretnog logaritma u toj grupi bio dovoljno težak, $\#E(\mathbb{F}_q)$ bi trebao imati barem jedan prosti faktor veći od 2^{160}. Obično se koriste krivulje kod kojih je $\#E(\mathbb{F}_q)$ oblika $h \cdot r$, gdje je r prost broj, a h mali prirodan broj.

Za krivulje specijalnog oblika poznati su efikasni algoritmi za problem diskretnog logaritma. To su tzv. *anomalne krivulje* kod kojih je $t = 1$, tj. $\#E(\mathbb{F}_q) = q$, te *supersingularne krivulje* kod kojih $p|t$, što za $p > 3$ znači da je $\#E(\mathbb{F}_p) = p + 1$. Stoga takve krivulje nisu prikladne za primjene u kriptografiji.

Reći ćemo sada nešto o metodama za određivanje reda $\#E(\mathbb{F}_q)$.

Prva metoda koju ćemo spomenuti koristi Legendreov simbol (odnosno njegovo poopćeneje za $\mathbb{F}_q$), tj. formulu

$$\#E(\mathbb{F}_q) = p + 1 + \sum_{x \in \mathbb{F}_q} \Big(\frac{x^3 + ax + b}{p}\Big).$$

Složenost ovog algoritma je $O(p \ln^2 p)$, što možemo pisati i kao $O(p^{1+\varepsilon})$, gdje je ε proizvoljno mala pozitivna konstanta. Ovaj algoritam je efikasan samo za vrlo male p-ove, a praktičiki je neprimjenjiv za $p > 10000$.

Prikat ćemo sada *Schanks-Mestreovu metodu* čija je složenost $O(p^{1/4+\varepsilon})$ i koja je u praksi primjenjiva za $p < 10^{30}$.

Iz Hasseovog teorema znamo da je $\#E(\mathbb{F}_p) = p+1-t$, $|t| \leq 2\sqrt{p}$. Izaberimo slučajnu točku $P \in E(\mathbb{F}_p)$. Želimo naći broj $N \in \langle p+1-2\sqrt{p}, p+1+2\sqrt{p}\rangle$ takav da je $[N]P = \mathcal{O}$. Takav broj N sigurno postoji jer, po Lagrengeovom teoremu, red od P dijeli $\#E(\mathbb{F}_p)$. Ako je red od P veći od $4\sqrt{p}$, onda je takav N jedinstven i jednak je $\#E(\mathbb{F}_p)$. Naivan način za pronalaženje broja N bio bi da ispitamo svih $\lfloor 4\sqrt{p} \rfloor$ mogućih brojeva. Bolji način se zasniva na tzv. *Shanksovoj "baby step - giant step"* (BSGS) metodi. Neka je

$Q = [p + 1 + \lfloor 2\sqrt{p} \rfloor]P$. Tada za broj $n = p + 1 + \lfloor 2\sqrt{p} \rfloor - N$ vrijedi da je $0 \leq n \leq 4\sqrt{p}$ i

$$[n]P = [p + 1 + \lfloor 2\sqrt{p} \rfloor - N]P = Q.$$

Dakle, zapravo trebamo riješiti problem diskretnog logaritma. Iako za taj problem nemamo jako efikasan algoritam, ipak ga BSGS metodom možemo riješiti efikasnije nego da redom uvrštavamo sve moguće n-ove. Neka je $m = \lceil 2p^{1/4} \rceil$. Tada je $n < m^2$, pa n možemo prikazati u obliku

$$n = im + j, \quad 0 \leq i \leq m - 1, \quad 0 \leq j \leq m - 1.$$

"Mali koraci" (baby steps) se sastoje u računanju točaka $[j]P$, $0 \leq j \leq m-1$ (nova točka dobiva se iz stare dodavanjem P - mali korak). "Veliki koraci" (giant steps) se sastoje u računanju točaka $Q - [i]([m]P)$, $0 \leq i \leq m - 1$ (nova točka dobiva se iz stare oduzimanjem $[m]P$ - veliki korak). Za svaki i testiramo postoji li j takav da je

$$Q - [i]([m]P) = [j]P.$$

Kada takve i, j pronađemo, tada je traženi n jednak $im + j$. Dakle, imamo sljedeći algoritam:

Shanks-Mestreova metoda:

$m = \lceil 2p^{1/4} \rceil$
$P \in E(\mathbb{F}_p)$, $|P| > 4\sqrt{p}$
$Q = [p + 1 + \lfloor 2\sqrt{p} \rfloor]P$
for $(0 \leq j \leq m - 1)$
    izračunaj i spremi $[j]P$
for $(0 \leq i \leq m - 1)$ {
    if $(Q - [i]([m]P) = [j]P$ za neki $0 \leq j \leq m - 1)$ then
        $t = im + j - \lfloor 2\sqrt{p} \rfloor$ }
return t

Primjer 3.1. *Zadana je krivulja*

$$E : y^2 = x^3 + 3x + 5$$

nad poljem $\mathbb{F}_{163}$. *Odrediti red grupe* $E(\mathbb{F}_{163})$.

Ovdje je $m = 8$. Uzmimo $P = (1, 3)$. Tada je $Q = [163 + 1 + 25]P = (106, 61)$. U sljedećoj tablici su prikazani "mali koraci":

j	0	1	2	3	4	5	6	7
$[j]P$	$\mathcal{O}$	$(1, 3)$	$(162, 162)$	$(4, 154)$	$(11, 37)$	$(143, 101)$	$(77, 80)$	$\mathbf{(118, 5)}$

Izračunamo $R = [8]P = (97, 150)$. "Veliki koraci" su prikazani u sljedećoj tablici:

i	0	1	2	3	4	5	6	7
$Q-[i]R$	$(106,61)$	$(79,83)$	$(145,65)$	$\mathbf{(118,5)}$	$(1,160)$	$(142,61)$	$(7,83)$	$(124,8)$

Dakle, $n = 3 \cdot 8 + 7 = 31$, $t = 31 - 25 = 6$ i konačno $\#E(\mathbb{F}_{163}) = 163 + 1 - 6 = 158$.

Ako je red točke P manji od $4\sqrt{p}$, onda nam će nam ovaj algoritam dati više mogućih kandidata za red grupe $\#E(\mathbb{F}_p)$. Dakle, postavlja se pitanje postoji li točka $P \in E(\mathbb{F}_p)$ čiji je red P veći od $4\sqrt{p}$. Potvrdan odgovor na ovo pitanje dao je Mestre. Da bi formulirali njegov rezultat, treba nam pojam "twista". Za eliptičku krivulju E na poljem K danu jednadžbom $y^2 = x^3 + ax + b$ i $g \in \mathbb{F}_p^*$, *(kvadratni) twist* od E s g je eliptička krivulja čija je jednadžba $gy^2 = x^3 + ax + b$, odnosno, iz supstituciju $X = gx$, $Y = g^2y$, $Y^2 = X^3 + g^2aX + g^3b$. U slučaju kada je $K = \mathbb{F}_p$, onda svi twistovi od E čine dvije klase izomorfnih krivulja. One kod kojih je g kvadratni ostatak modulo p izomorfne su s E, dok su sve one kod kojih je g kvadratni neostatak modulo p izomorfne jednoj drugoj krivulji koju ćemo označiti s E'. Iz formule za prikaz $\#E(\mathbb{F}_p)$ pomoću Legendreovih simbola, direktno slijedi da je

$$\#E(\mathbb{F}_p) + \#E'(\mathbb{F}_p) = 2p + 2.$$

To znači da ako znamo red od $\#E(\mathbb{F}_p)$, onda znamo i red od $\#E'(\mathbb{F}_p)$, i obrnuto. Sada možemo navesti gore najavljeni Mestreov rezultat koji kaže da ako je $p > 457$, onda postoji točka reda većeg od $4\sqrt{p}$ na barem jednoj od krivulja E i E'. Štoviše, takvih točaka ima relativno mnogo (ima ih više od $c \ln p / \ln \ln p$ za neku konstantu c).

Prvi polinomijalni algoritam za računanje reda grupe $E(\mathbb{F}_q)$ dao je Schoof 1995. godine. Taj algoritam imao je složenost $O(\ln^8 q)$. Kasnije su Atkin i Elkies poboljšali Schoofov algoritam do složenosti $O(\ln^6 q)$, pa je danas moguće izračunati red grupe $E(\mathbb{F}_p)$ za proste brojeve $p < 10^{500}$. Vrlo kratko ćemo spomenuti neke od ideja koje se koriste u Schoofovom algoritmu. Polazna ideja je računanje broja t tako da se izračuna $t \bmod l$ za male proste brojeve l. Ako je l_{max} najmanji prosti broj takav da je

$$\prod_{\substack{l \text{ prost} \\ l \le l_{max}}} l > 4\sqrt{q},$$

onda iz poznavanja $t \bmod l$ za $2 \le l \le l_{max}$, pomoću Kineskog teorema o ostatcima možemo izračunati t. Broj l_{max} je reda veličine $O(\ln q)$, pa je broj kongruencija u pripadnom sustavu $O(\frac{\ln q}{\ln \ln q})$. U određivanju $t \bmod l$ koristi se tzv. *Frobeniousov endomorfizam*. To je preslikavanje $\varphi : E(\mathbb{F}_q) \to E(\mathbb{F}_q)$ zadano sa $\varphi(x, y) = (x^q, y^q)$, $\varphi(\mathcal{O}) = \mathcal{O}$. Frobeniusov endomorfizam φ i Frobeniusov trag t povezani su relacijom

$$\varphi^2 - [t]\varphi + [q] = [0],$$

tj. za svaku točku $P = (x, y) \in E(\mathbb{F}_q)$ vrijedi

$$(x^{q^2}, y^{q^2}) - [t](x^q, y^q) + [q](x, y) = \mathcal{O}.$$

Neka je točka $P \in E(\mathbb{F}_q)$ takva da je $[l]P = \mathcal{O}$, te neka je $q_l = q \bmod l$. Ako za $\tau \in 0, 1, \ldots, l-1$ vrijedi $\varphi^2(P) + [q_l]P = [\tau]\varphi(P)$, onda je $t \bmod l = \tau$.

Poglavlje 4

Testiranje i dokazivanje prostosti

4.1 Distribucija prostih brojeva

Definicija 4.1. 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 $p_1, p_2, \ldots, p_m$ bili svi 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* (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 tzv. *logaritamsko-integralna funkcija*

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

Po L'Hopitalovom pravilu neposredno dobivamo da je

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

Stoga je PNT ekvivalentan sa $\pi(x) \sim \mathrm{li}(x)$.

Slijedeći važan rezultat kojeg ćemo spomenuti je *Dirichletov teorem o prostim brojevima u aritmetičkom nizu*. On kaže da 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

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

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 sa

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

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* (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 PNT, dok je Riemannova slutnja ekvivalentna sa sljedećom ocjenom

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

Preciznije, ova ocjena za konkretni ε je ekvivalentna s tvrdnjom da $\zeta(s)$ nema nultočaka u poluravnini $\mathrm{Re}(s) > \frac{1}{2} + \varepsilon$.

Kod ocjene veličine najmanjeg kvadratnog ostatka modulo p, spominjali smo *proširenu Riemannovu slutnju* (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 $(n, d) > 1$.

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 $(n, d) = 1$, 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 $d = p$ prost broj. Neka je g primitivni korijen modulo p. Tada je, zbog multiplikativnosti, χ u potpunosti određen sa $\chi(g)$. 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 sa

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

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

4.2 Pseudoprosti brojevi

Vidjeli smo da se u konstrukciji većine kriptosustava s javnim ključem kreće od jednog ili više velikih prostih brojeva. Stoga se postavlja 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 egzakztno 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

$$b^p \equiv b \pmod{p}, \tag{4.1}$$

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 (4.1).

Definicija 4.2. Kažemo da je složen broj n *pseudoprost* u bazi b (kraće: n je psp(b)) ako je

$$b^n \equiv b \pmod{n}.$$

Npr. $341 = 11 \cdot 31$ je psp(2), $91 = 7 \cdot 13$ je psp(3).

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 "rijeđi" od prostih brojeva. Pa ipak i njih ima beskonačno mnogo.

Teorem 4.1. *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

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

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 $b^2 - 1$, to 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 $b^{2p}-1$, to 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 bi 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 zovu *Carmichaelovi brojevi*. Ukoliko je poznata faktorizacija od n, onda je lako ustanoviti da li je 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, to on mora biti produkt različitih prostih brojeva. Ostaje za vidjeti zašto n ne može biti produkt od dva prosta broja. 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 Fermatovog teorema. Sada ćemo pokazati kako se malim modificiranjem testa taj nedostatak može ukloniti.

Neka je n neparan prirodan broj, $(b,n) = 1$, 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 $b^{(n-1)/2^{i-1}} \equiv 1 \pmod{n}$, a jedina rješenja kongruencije $x^2 \equiv 1 \pmod{n}$ ako je n prost su $x \equiv \pm 1 \pmod{n}$. Dakle, kombinirajući mali Fermatov teorem sa svojstvom kongruencije $x^2 \equiv 1 \pmod{p}$ dobivamo jači zahtjev od onog iz definicije pseudoprostih brojeva.

Definicija 4.3. 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

$$b^t \equiv 1 \pmod{n} \text{ ili postoji } r,\ 0 \leq r < s \text{ takav da je } b^{2^r \cdot t} \equiv -1 \pmod{n}, \quad (4.2)$$

onda kažemo da je n *jaki pseudoprosti broj u bazi* b (ili da je n spsp(b)).

Ako uvjet (4.2) nije ispunjen za neki b, $0 < b < n$, tada je broj n složen. U tom slučaju broj b zovemo *svjedok složenosti od* n.

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 $2^{85} \equiv 32 \pmod{341}$ i $2^{170} \equiv 1 \pmod{341}$. Kao primjer jakog pseudoprostog broja navedimo npr. da je 91 spsp(10) jer je $10^{45} \equiv -1 \pmod{91}$. 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.

Teorem 4.2. *Neka je n neparan složen broj. Tada je n jaki pseudoprosti broj u bazi b za najviše $(n-1)/4$ baza b, $0 < b < n$.*

Teorem 4.2 nam pokazuje da u slučaju jakih pseudoprostih brojeva ne postoji analogon Carmichaelovih brojeva. Dakle, nemoguće da složen broj bude jaki pseudoprosti broj u svakoj bazi.

Prije dokaza teorema 4.2 dokažimo lemu koja će nam dati potrebne informacije o kongruencijama oblika $x^m \equiv \pm 1 \pmod{n}$. Sa $\nu_m(t)$ ćemo označavati najveći cijeli broj k takav da m^k dijeli t.

Lema 4.3. *Neka je* $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ *neparan broj, te neka je*

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

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 postoji j takav da je $g_i^j \equiv x \pmod{p_i^{\alpha_i}}$. Označimo taj j s $\operatorname{ind}_{g_i} x$. Sada je kongruencija $x^m \equiv c \pmod{n}$ ekvivalentna sustavu kongruencija

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

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

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

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

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

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

Sada iz svojstava linearnih kongruencija zaključujemo da ako ovaj sustav ima rješenja, onda ih ima s, a nužan i dovoljan uvjet za postojanje rješenja je da za svaki i broj $(m, \varphi(p_i^{\alpha_i}))$ dijeli $\frac{\varphi(p_i^{\alpha_i})}{2}$. Odavde imamo da $(2^{\nu_2(m)} m_1, p_i^{\alpha_i - 1} \cdot 2^{\nu_2(p-1)} \cdot q_i)$ dijeli $p_i^{\alpha_i - 1} \cdot 2^{\nu_2(p-1)-1} \cdot q_i$, gdje su m_1 i q_i neparni brojevi. No, ovo je očito ekvivalentno s $\nu_2(m) < \nu_2(p_i - 1)$ za $i = 1, \ldots, r$, tj. s $\nu_2(m) < \nu$. □

Dokaz teorema 4.2:

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 4.3, ova kongruencija ima $d = (p(p-1), n-1)$ 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)

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

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 $u \leq w$. Pretpostavimo da je $b^t \equiv 1 \pmod{n}$. Po lemi 4.3, takvih baza ima $(t,v) \cdot (t,z) \leq vz$. Pretpostavimo sada da je $b^{2^r t} \equiv -1 \pmod{n}$ za neki r, $0 \leq r < s$. Po lemi 4.3, ova kongruencija ima rješenja ako i samo ako je $r < u$, a broj rješenja je $2^r(t,v) \cdot 2^r(t,z) \leq 4^r vz$. 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 $(t,v) \leq v$, $(t,z) \leq z$ mora biti stroga, jer bi 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 bi dobili da $z | v$, što bi značilo da je $v = z$ i $p = q$, što je kontradikcija.

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

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

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):

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

□

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 mod 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 znamo da je sigurno n složen. Ako n prođe test za k b-ova, onda je vjerojatnost da je n složen $\leq \frac{1}{4^k}$.

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". 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 4.2 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-Rabinovog 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 (4.2). 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 4.4 (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 Ankenyjevog teorema na 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 dvije. Neparan složen broj n je *Eulerov pseudoprost broj u bazi* b (n je epsp(b)) ako zadovoljava Eulerov kriterij:

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

(ovdje $(\frac{b}{n})$ označava Jacobijev simbol). Može se pokazati da je svaki spsp(b) ujedno i epsp(b), pa je stoga tzv. Soloway-Strassenov test, koji je zasnovan na Eulerovim pseudoprostim brojevima, manje efikasan od Miller-Rabinovog testa.

Sada ćemo definirati Lucasove pseudoproste brojeve. Neka su α i β korijeni polinoma $x^2 - ax + b = 0$, $a, b \in \mathbb{Z} \setminus \{0\}$. Definiramo nizove $U_k(a,b) = \frac{\alpha^k - \beta^k}{\alpha - \beta}$, $V_k(a,b) = \alpha^k + \beta^k$. Oni nizovi se zovu *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

$$U_{\delta(p)} \equiv 0 \pmod{p}, \tag{4.3}$$

gdje je $\delta(p) = p - (\frac{D}{p})$. 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 $\mathrm{lpsp}(a, b)$). 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.

4.3 Dokazivanje prostosti pomoću eliptičkih krivulja

Ukoliko 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. Sada ćemo reći nešto o metodama kojima se može dokazati da je dani broj prost.

Teorem 4.5 (Pocklington). *Neka je s djelitelj od $n-1$ koji je veći od $\sqrt{n}$. Pretpostavimo da postoji prirodan broj a takav da vrijedi*

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

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

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 Fermatovog teorema $b^{p-1} \equiv 1 \pmod{p}$, to zaključujemo da s dijeli $p - 1$. No, to je nemoguće budući da je $s > \sqrt{n}$, a $p \leq \sqrt{n}$. □

Primjer 4.1. *Dokažimo da je broj $n = 153533$ prost.*

Imamo $n - 1 = 2^2 \cdot 131 \cdot 293$, pa možemo uzeti $s = 4 \cdot 131$. Prosti djelitelji od s su 2 i 131. Možemo uzeti $a = 2$ jer je $2^{n-1} \equiv 1 \pmod{n}$, $(2^{(n-1)/2} - 1, n) = 1$, $(2^{(n-1)/131} - 1, n) = 1$. Stoga Pocklingtonom teorem povlači da je n prost. Ovdje smo implicitno koristili da je 131 prost. Da bi dokazali prostost

od 131, možemo postupiti na isti način. Imamo $131 - 1 = 130 = 2 \cdot 5 \cdot 13$, pa uzmimo $s = 13$. Tada iz $2^{130} \equiv 1 \pmod{131}$ i $(2^{10} - 1, 131) = 1$ slijedi da je 131 prost (uz pretpostavku da je broj 13 prost).

U prethodnom primjeru smo vidjeli da primjenom Pocklingtonovog teorema pitanje o prostosti jednog broja svodimo da isto pitanje za jedan ili više manjih brojeva, i taj postupak nastavljamo sve dok brojevi ne postanu dovoljno mali.

Da bi dokazali prostost broja n pomoću Pocklingtovog teorema, moramo poznavati barem djelomičnu faktorizaciju broja $n - 1$. No, kao što smo već više puta napomenuli, faktorizacija velikih brojeva je općenito težak problem. Ipak, ova metoda je vrlo prikladna u slučaju brojeva specijalnog oblika, kod kojih je poznata faktorizacija dovoljno velikog faktora od $n - 1$.

Teorem 4.6 (Proth). *Neka je $l \geq 2$, $k \geq 1$, $k \not\equiv 0 \pmod 3$ i $k \leq 2^l + 1$. Tada je broj $n = k \cdot 2^l + 1$ prost ako i samo ako je $3^{k \cdot 2^{l-1}} \equiv -1 \pmod n$.*

Dokaz: Pretpostavimo da je $3^{k \cdot 2^{l-1}} \equiv -1 \pmod n$. Stavimo $s = 2^l$, $a = 3$, $n = k \cdot 2^l + 1$. Tada je $a^{n-1} = 3^{k \cdot 2^l} \equiv (-1)^2 \equiv 1 \pmod n$ i $a^{(n-1)/2} \equiv -1 \pmod n$. Budući da n dijeli $a^{(n-1)/2}+1$, to je on relativno prost s $a^{(n-1)/2}-1$. Po Pocklingtonovom teoremu zaključujemo da je broj n prost.

Dokažimo sada obrat. Neka je n prost. Tada je, zato što 3 ne dijeli k, $n \equiv 2 \pmod 3$, pa imamo

$$3^{k \cdot 2^{l-1}} = 3^{(n-1)/2} \equiv (\frac{3}{n}) \equiv (\frac{n}{3}) \equiv (\frac{2}{3}) \equiv -1 \pmod n.$$

□

Postoje metode za dokazivanje prostosti koje se zasnivaju na faktorizaciji od $n+1$, umjesto od $n-1$. Spomenimo samo tzv. *Lucas-Lehmerovu metodu* za dokazivanje prostosti Mersennovih brojeva.

Teorem 4.7 (Lucas-Lehmer). *Neke 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}.

Kao što smo već napomenuli, problem s primjenom Pocklingtonovog 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 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.

Dakle, promatrat ćemo eliptičke krivulje nad prstenom $\mathbb{Z}_n$. Budući da n ne mora biti prost, može se dogoditi da neke točke na $E(\mathbb{Z}_n)$ nećemo moći zbrojiti jer će se u formuli za zbrajanje točaka u nazivniku pojaviti broj koji nije invertibilan modulo n. No, to nam neće biti problem jer će to značiti da je n složen. Štoviše, moći ćemo mu naći netrivijalni faktor tako da izračunamo najveći zajednički djelitelj tog nazivnika i broja n.

Teorem 4.8. *Neka je E eliptička krivulja nad $\mathbb{Z}_n$, gdje je $(6, n) = 1$ i $n > 1$, dana jednadžbom $y^2 = x^3 + ax + b$. Neka je m prirodan broj koji ima prosti faktor $q > (n^{1/4} + 1)^2$. Ako postoji točka $P \in E(\mathbb{Z})$ takva da je*

$$[m]P = \mathcal{O} \quad i \quad [m/q]P \neq \mathcal{O},$$

onda je broj n prost.

Dokaz: Ako je n složen, onda ima prosti faktor $p \leq \sqrt{n}$. Promotrimo eliptičku krivulju E' nad $\mathbb{Z}_p$ danu istom jednadžbom kao i E. Neka je m' red grupe $E'(\mathbb{Z}_p)$. Po Hasseovom teoremu je

$$m' \leq p + 1 + \sqrt{p} = (\sqrt{p} + 1)^2 \leq (n^{1/4} + 1)^2 < q.$$

Stoga je $(m', q) = 1$, pa postoji $u \in \mathbb{Z}$ takav da je $uq \equiv 1 \pmod{m'}$. Neka je $P' \in E'(\mathbb{Z}_p)$ točka dobivena iz P redukcijom koordinata modulo p. Tada je po uvjetu teorema $[m/q]P' \neq \mathcal{O}$. No, s druge strane imamo

$$[m/q]P' = [uq \cdot \frac{m}{q}]P' = [um]P' = [u]([m]P') = \mathcal{O}.$$

□

Primjer 4.2. *Dokažimo da je broj $n = 907$ prost.*

Neka je E eliptička krivulja zadana jednadžbom $y^2 = x^3 + 10x - 2$ nad $\mathbb{Z}_n$. Neka je $P = (819, 784)$ i $q = 71$. Tada je $[71]P = \mathcal{O}$. Budući je $71 > (907^{1/4} + 1)^2$, odavde slijedi da je broj 907 prost (ako je poznato da je broj 71 prost).

U praksi je kod velikih brojeva n najproblematičiji dio algoritma pronalaženje eliptičke krivulje za koju će red grupe $E(\mathbb{Z}_n)$, a to će biti broj m iz teorema, imati dovoljno veliki prost faktor. Jedna je mogućnost biranje krivulja na slučajan način, pa računanje njihovih redova sa Schoofovim algoritmom. Da bi ocijenili kolika je vjerojatnost uspjeha pronalaženja odgovarajuće krivulje, trebali bi znati nešto o distribuciji prostih brojeva u intevalu oblika $[x + 1 - 2\sqrt{x}, x + 1 - 2\sqrt{x}]$. Nažalost, o tome postoje samo (nedokazane) hipoteze. Ako bi vrijedilo

$$\pi(x + 1 - 2\sqrt{x}) - \pi(x + 1 - 2\sqrt{x}) > A\frac{\sqrt{x}}{\ln x},$$

za neku konstantu A (što je hipoteza motivirana teoremom o prostim brojevima), onda bi očekivani broj operacija u Goldwasser-Killianovom algoritmu bio $O(\ln^{10} n)$. Mogli bi reći da je interval iz Hasseovog teorema dovoljno velik za praksu, ali ne i za trenutno stanje teorije. Adleman i Huang su 1992. predložili algoritam koji umjesto eliptičkih krivulja koristi Jacobijane hipereliptičkih krivulja, a za koji se korištenjem poznatih rezultata o distribuciji prostih brojeva u intervalu oblika $[x, x + x^{3/4}]$ može dokazati da mu je očekivani broj operacija polinomijan.

Atkin i Morain su 1993. predložili jednu varijantu dokazivanja prostosti pomoću eliptičkih krivulja, za koju se danas smatra da je najefikasnija u praksi. Pomoću te metode se danas može efikasno dokazati prostost brojeva s oko 1000 znamenaka. Metoda koristi tzv. eliptičke krivulje s *kompleksnim množenjem* s pripadnim imaginarnim kvadratnim poljem $\mathbb{Q}(\sqrt{-d})$. Za takve krivulje E vrijedi da ako je $4p = x^2 + dy^2$ (podsjetimo se da se ovakav prikaz može dobiti pomoću Cornacchia-Smithovog algoritma), onda su mogući redovi od E nad $\mathbb{Z}_p$ brojevi $p + 1 \pm x$. Dakle, ove brojeve možemo efikasno izračunati, te vidjeti imaju li dovoljno veliki prosti faktor. Kad pronađemo red koji nas zadovoljava, onda samu krivulju konstruiramo koristeći teoriju kompleksnog množenja, posebno tzv. *j-invarijante*.

4.4 Polinomijalni AKS algoritam za dokazivanje prostosti

Godine 2002. Agrawal, Kayal i Saxena objavili su prvi polinomijalni algoritam za dokazivanje prostosti. 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 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 Fermatovom teoremu. Točnije, polazište mu je sljedeća lema.

Lema 4.9. *Neka je $a \in \mathbb{Z}$, $n \in \mathbb{Z}$, $n \geq 2$ i $(a, n) = 1$. Tada je broj n prost ako i samo ako vrijedi*

$$(X + a)^n \equiv X^n + a \pmod{n}, \tag{4.4}$$

tj. akko su odgovarajući koeficijenti polinoma na lijevoj i desnoj strani kongruencije (4.4) 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 (4.4).

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 $(q, a^{n-q}) = 1$. Zaključujemo da koeficijent od X^q nije djeljiv sa q^k, pa stoga nije kongruentan 0 modulo n. □

Doslovna primjena prethodne leme neće dati efikasan algoritam za dokazivanje prostosti jer bi trebali izračunati n koeficijenata na lijevoj strani od (4.4). Jednostavan način za reduciranje broja koeficijenata koje treba izračunati jest da se obje strane kongruencije (4.4) reduciraju modulo $X^r - 1$, za prikladno odabrani mali broj r. Dakle, provjerava se vrijedi li

$$(X + a)^n \equiv X^n + a \pmod{X^r - 1,\, n}. \tag{4.5}$$

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 4.9 je jasno da ako je n prost, onda (4.5) vrijedi za sve a i r. Pokazuje se da vrijedi i djelomični obrat, tj. da za prikladno odabran r vrijedi da ako je (4.5) zadovoljeno za nekoliko a-ova, onda n mora biti potencija 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.

AKS algoritam:

1. if (n je potencija prirodnog broja) then return n *je složen*
2. Nađi r takav da je $o_r(n) > 4\ln^2 n$
3. if ($1 < (a, n) < n$ za neki $a \le r$) then return n *je složen*
4. if ($n \le r$) then return n *je prost*
5. for ($1 \le a \le \lfloor 2\sqrt{\varphi(r)}\ln n \rfloor$) {
   if ($(X + a)^n \not\equiv X^n + a \pmod{X^r - 1,\, n}$) then
   return n *je složen* }
6. return n *je prost*

Za dokazati korektnost 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.

Recimo nešto o složenosti AKS algoritma. Najprije trebamo testirati da li je 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 $k = 2$. 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 n vrijedi $o_r(n) >$

$4\ln^2 n$. Najprije dokažimo jedan pomoćni rezultat. Sa $[a_1, \ldots, a_m]$ označavat ćemo najmanji zajednički višekratnik brojeva $a_1, \ldots, a_m$.

Lema 4.10. *Neka je* $d_n = [1, 2, \ldots, n]$. *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 integracijom se 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 $n(2n+1)\binom{2n}{n}$ 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 4.11. *Postoji* $r \leq \lceil 16\ln^5 n \rceil$ *takav da je* $o_r(n) > 4\ln^2 n$.

Dokaz: Neka su $r_1, r_2, \ldots, r_t$ svi brojevi za koje je $o_r(n) \leq 4\ln^2 n$. Svaki od ovih brojeva dijeli produkt $\prod_{i=1}^{\lfloor 4\ln^2 n\rfloor}(n^i - 1)$. No,

$$\prod_{i=1}^{\lfloor 4\ln^2 n\rfloor}(n^i - 1) < n^{\sum_{i=1}^{\lfloor 4\ln^2 n\rfloor} i} < n^{10\ln^4 n} < 2^{16\ln^5 n}.$$

Prema Lemi 4.10 je

$$[1, 2, \ldots, \lceil 16\ln^5 n\rceil] \leq 2^{\lceil 16\ln^5 n\rceil}.$$

Kako je najmanji zajednički višekratnik brojeva $r_1, r_2, \ldots, r_t$ manji od $2^{\lceil 16\ln^5 n\rceil}$, zaključujemo da postoji prirodan broj $r \leq \lceil 16\ln^5 n\rceil$ takav da je $o_r(n) > 4\ln^2 n$. □

Koristeći Lemu 4.11 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, to 2. korak algoritma ima složenost $O(\ln^{7+\varepsilon} n)$.

U 3. koraku, r puta računamo najveći zajednički djelitelj dvaju brojeva. Složenst 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 ocjena se može poboljšati preciznijom analizom gornje ograde za r, koja se može dobiti korištenjem preciznijih rezultata o distribuciji prostih brojeva (umjesto Leme 4.10). Ako se još k tome, umjesto "školskog množenja", koriste algoritmi za brzo množenje, dobiva se ocjena $O(\ln^{7.5+\varepsilon} n)$.

Poglavlje 5

Metode faktorizacije

5.1 Pollardova ρ metoda

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 teškoći su 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 ocjene smo pretpostavili da nam je dostupna tablica svih prostih brojeva $\leq \sqrt{n}$. U protivnom, dijelili bi 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.

Jedna od najjednostavnijih metoda faktorizacije čija je složenost bolja od $O(\sqrt{n})$ je tzv. *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 $(x_i - x_j, n)$.

Niz (x_i) se konstruira pomoću preslikavanja $f : \mathbb{Z}_n \to \mathbb{Z}_n$. Od tog preslikavanja se 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:

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

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 ilustrira pretperiod, a "okrugli dio" od ρ ilustrira čisto periodični dio niza.

Postavlja se pitanje kada možemo očekivati prvo ponavljaanje 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 dva broja koja su međusobno kongruentna modulo p, s vjerojatnošću većom od $1/2$. Vjerojatnost da je k slučajno izabranih brojeva 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 $1/2$, postoje dvije osobe koje imaju rođendan istog dana.

Ako bismo faktor od n tražili tako da računamo $(x_i - x_j, n)$ za sve i, j, to bi bilo vrlo neefikasno. Puno efikasnije je računati samo $(x_{2i} - x_i, n)$. 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 5.1. *Faktorizirajmo broj* $n = 1387$.

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

$$\begin{aligned} &x_1 = 2^2 - 1 = 3, \quad y_1 = 3^2 - 1 = 8, \quad (y_1 - x_1, n) = (-5, 1387) = 1;\\ &x_2 = 8, \quad y_2 = f(63) = 1194, \quad (y_2 - x_2, n) = (1186, 1387) = 1;\\ &x_3 = 63, \quad y_3 = f(1194^2 - 1 \bmod n) = f(1186^2 - 1 \bmod n) = 177,\\ &\qquad (y_3 - x_3, n) = (144, 1387) = 19.\end{aligned}$$

Dakle, 19 je djelitelj od 1387. Zaista, $1387 = 19 \cdot 73$.

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 diskretnoj logaritma u općoj konačnoj abelovoj grupi.

5.2 Pollardova $p-1$ metoda

Pollardova $p-1$ metoda iz 1974. godine spada u specijalne metode faktorizacije. Njezino polazište je ponovo 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 $(a,p) = 1$. Štoviše, vrijedi $a^m \equiv 1 \pmod{p}$ za svaki višekratnik od $p-1$. Ako nađemo m, onda nam $(a^m - 1, n)$ 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 $\leq B$. Ako je broj $p-1$ B-gladak, onda za m možemo uzeti najmanji zajednički višekratnik brojeva $1, 2, \ldots, B$. Za ovako odabrani m, broj operacija za računanje $a^m \pmod{n}$ 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 5.2. Neka je $n = 540143$. Izaberimo $B = 8$ i $a = 2$. Tada je $m = 2^3 \cdot 3 \cdot 5 \cdot 7 = 840$. Imamo da je $2840 \bmod n = 53047$ i $(53046, n) = 421$. Zaista, $n = 421 \cdot 1283$.

Pomoću $p-1$ metode je Baillie 1980. godine našao 25-znamenkasti faktor Mersennovog 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 $E(\mathbb{F}_p)$, 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 $\mathbb{F}_p$ dovoljno glatkog reda.

5.3 Faktorizacija pomoću eliptičkih krivulja

Godine 1987. H. W. Lenstra je predložio modifikaciju Pollardove $p-1$ metode koja koristi eliptičke krivulje. Kao rezultat dobio je subeksponencijalni algoritam koji i danas predstavlja jedan od najefikasnijih algoritama za faktorizaciju.

Slično kao kod metode dokazivanja prostosti pomoću eliptičkih krivulja, i ovdje ćemo raditi s eliptičkim krivuljama nad prstenom $\mathbb{Z}_n$. Dok je kod dokazivanja prostosti postojala (mala) mogućnosti da je n složen (pa da $\mathbb{Z}_n$ nije polje), ovdje ćemo od početka biti sigurni da je n složen. Pretpostavit ćemo da je $(n, 6) = 1$, te ćemo promatrati eliptičke krivulje oblika

$$E_{a,b}: \quad y^2 = x^3 + ab + b,$$

gdje je $(4a^3+27b^2, n) = 1$. Kada je n prost, onda na eliptičkoj krivulji postoji samo jedna projektivna točka koja ne odgovara nekoj afinoj točki (točka u beskonačnosti). U slučaju kada je n složen, takvih točaka može biti više.

Opišimo sada osnovne korake u *Lenstrinom algoritmu za faktorizaciju* (ECM – Elliptic Curve Method).

1. Izbor eliptičke krivulje.

   Postoji više načina za izbor odgovarajuće eliptičke krivulje. Na primjer, možemo izabrati slučajno elemente $a, x, y \in \mathbb{Z}_n$, pa izračunati $b = (y^2 - x^3 - ax) \bmod n$. Neka je $g = (4a^3 + 27b^2, n)$. Ako je $1 < g < n$, onda smo našli netrivijalni faktor od n. Ako je $g = n$, onda biramo nove a, x, y. Ako je $g = 1$, onda smo našli eliptičku krivulju $E_{a,b}$ nad $\mathbb{Z}_n$ i točku $P = (x, y)$ na njoj.

2. Neka je k najmanji zajednički višekratnik brojeva $1, 2, \ldots, B$, za prikladno odabranu granicu B. U praksi se obično uzima najprije $B = 10000$, a potom se granica po potrebi povećava.

3. Računamo $[k]P \in E_{a,b}(\mathbb{Z}_n)$ koristeći formule za zbrajanje točaka:

   $$(x_3, y_3) = (\lambda^2 - x_1 - x_2 \bmod n,\ \lambda(x_1 - x_3) - y_1 \bmod n),$$

   gdje je $\lambda = (3x_1^2 + a) \cdot (2y_1)^{-1} \bmod n$ ako su točke jednake, a $\lambda = (y_1 - y_2)(x_1 - x_2)^{-1} \bmod n$, inače.

4. Ako se u računanju $[k]P$ dogodi da neki zbroj točaka ne možemo izračunati zato što ne možemo izračunati d^{-1} jer d nema inverz modulo n, onda izračunamo $g = (d, n)$. Ako je $g \neq n$, onda smo našli netrivijalni faktor od n.

5. U slučaju neuspjeha, možemo izabrati novu eliptičku krivulju ili poveáti granicu B.

Primjer 5.3. *Faktorizirati broj* $n = 187$.

Neka je $B = 3$, pa je $k = [1, 2, 3] = 6$. Izaberimo eliptičku krivulju $y^2 = x^3 + x + 25$ i točku na njoj $P = (0, 5)$. Računamo $[6]P = [2](P + [2]P)$. Najprije računamo $[2]P$. Pripadni λ je $10^{-1} = 131 \bmod 187$, pa dobivamo $[2]P = (144, 18)$. Zatim računamo $[3]P = P + [2]P$. Pripadni λ je $13 \cdot 144^{-1} = 178 \bmod 187$, pa je $[3]P = (124, 176)$. Konačno, računamo $[6]P = [2]([3]P)$. Pripadni λ je $127 \cdot 165^{-1}$. Kod računanja inverza od 165 modulo 187, dobivamo da taj inverz ne postoji jer je $(165, 187) = 11$. Odavde zaključujemo da je 11 faktor od 187. Zaista, $187 = 11 \cdot 17$.

O čemu ovisi uspjeh ovog algoritma? Slično kao kod $p-1$ metode, i ovdje bi k trebao biti višekratnik reda pripadne grupe. U ovom slučaju k bi trebao biti višekratnik od $\#E(\mathbb{Z}_p)$, gdje je p neki prosti faktor od n. Zaista, u tom slučaju će kod računanja $[k]P$ pripadni nazivnik biti djeljiv s p, pa neće biti invertibilan modulo n. Naime, u $E(\mathbb{Z}_p)$ će vrijediti da je $[k]P = \mathcal{O}$.

Kod ocjene složenosti ovog algoritma ključno je pitanje kako optimalno odabrati granicu B. Uvedimo oznaku

$$\psi(x, y) = \#\{1 \leq n \leq x \,:\, n \text{ je } y\text{-gladak}\}.$$

Koristeći činjenicu da su redovi $\#E(\mathbb{Z}_p)$ skoro uniformno distribuirani unutar Hasseovog intervala, dolazimo do sljedeće ocjene za vjerojatnost uspjeha algoritma:

$$prob(B) > c \cdot \frac{\psi(p + 1 + 2\sqrt{p}, B) - \psi(p + 1 - 2\sqrt{p}, B)}{\sqrt{p} \ln p}.$$

Kako je, s druge strane, broj operacija potrebnih za pokušaj faktorizacije pomoću jedne krivulje proporcionalan s B, željeli bi minimizirati vrijednost $B/prob(B)$. Pokazuje se da se, koristeći gore navedenu ocjenu za $prob(B)$, minimum postiže za

$$B = e^{(\sqrt{2}/2 + o(1))\sqrt{\ln p \ln \ln p}},$$

dok je složenost algoritma

$$e^{(\sqrt{2} + o(1))\sqrt{\ln p \ln \ln p}}.$$

U najlošijem slučaju (kada je $p = O(\sqrt{n})$), složenost metode faktorizacije pomoću eliptičkih krivulja je $e^{O(\sqrt{\ln n \ln \ln n})}$. Dakle, to je subeksponencijalni algoritam.

Iako postoje algoritmi bolje složenosti (algoritam sita polja brojeva), važno svojstvo ECM je da njezina složenost ovisi o najmanjem prostom

faktoru od n. Zato ona nije najprikladnija za faktorizaciju RSA modula, tj. brojeva oblika $n = pq$, gdje su p i q bliski prosti brojevi. Međutim, kod faktorizacije "slučajnih" brojeva, ECM često daje bolje rezultate od ostalih metoda, jer takvi brojevi obično imaju neki prosti faktor koji je znatno manji od $\sqrt{n}$. Čak i kod primjene asimptotski boljih metoda, unutar tih algoritma potrebno je faktorizirati neke pomoćne brojeve, za koje možemo očekivati da se ponašaju kao slučajni brojevi, pa se tu ECM može koristiti kao pomoćna metoda.

Među faktorizacijama dobivenim pomoću ECM, spomenimo nalaženje 33-znamenkastog faktora Fermatovog broja $2^{2^{15}}+1$ (Crandall, van Halewyn, 1997.), te nalaženje 49-znamenkastog faktora Mersenneovog broja $2^{2071}-1$ (Zimmermann, 1998.).

5.4 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 tzv. *Fermatova faktorizacija*, koja je primjenjiva na brojeve n koji su produkti dva bliska broja. Takav n je tada razlika kvadrata dva prirodna broja, 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.

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 $(s+t, n)$ i $(t-s, n)$ netrivijalni faktori od n.

Prva takva metoda koju ćemo upoznati 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,

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

Verižni razlomak se može izračunati pomoću sljedećeg algoritma:

$$a_0 = \lfloor\sqrt{n}\rfloor,\ s_0 = 0,\ t_0 = 1,$$

$$s_{i+1} = a_i t_i - s_i,\ t_{i+1} = \frac{n - s_{i+1}^2}{t_i},\ a_i = \left\lfloor \frac{a_0 + s_i}{t_i} \right\rfloor \text{ za } i \geq 0.$$

Neka je $\frac{p_i}{q_i} = [a_0; a_i, \ldots, a_i]$. Tada vrijedi

$$p_i^2 - nq_i^2 = (-1)^{i+1} \cdot t_{i+1} \text{ i } 0 < t_i < 2\sqrt{n}.$$

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 $(p_{k_1} \cdots p_{k_m} + w, n)$ dati netrivijalni faktor od n.

Primjer 5.4. *Faktorizirati broj* $n = 9073$.

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	95	286	381	1119	2619	16833

Očito je $w_0 w_4 = (-1)^1 t_1 (-1)^5 t_5 = 36^2$. Stoga je

$$p_0^2 p_4^2 = (95 \cdot 2619)^2 \equiv 3834^2 \equiv 36^2 \pmod{9073}.$$

Izračunamo: $(3834 + 36, 9073) = 43$. Zaista, $9073 = 43 \cdot 211$.

Da bi metoda verižnog razlomka postala stvarno relativno efikasna (subeksponencijalna) metoda, gore opisanu ideju treba kombinirati s korištenjem tzv. *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. 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 pola prostih brojeva se 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, i ovdje se pokazuje 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 $N'_{11} = N_{11}/1307$, 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.

5.5 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.

$$S = \{t^2 - n : \lfloor\sqrt{n}\rfloor + 1 \le t \le \lfloor\sqrt{n}\rfloor + A\},$$

za neki prikladno odabrani A.

Glavna ideja ove metode je da umjesto da za svaki $s \in S$ djeljeći ga s prostim brojevima $p \in \mathcal{B}$ provjeravamo da li je $\mathcal{B}$-broj, mi 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.

Algoritam kvadratnog sita: Neka je n neparan složen broj.

1. Odaberimo brojeve B i A, oba reda veličine $e^{\sqrt{\ln n \ln\ln n}}$. Obično se uzima da je $P < A < P^2$.

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 takav da je $(\frac{n}{p}) = 1$, 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 iz S).

5. Za p iz točke **4.**, pogledamo listu iz točke **2.** U stupcu ispod p stavimo 1 kod svih vrijednosti od $t^2 - n$ kod kojih $p | t - t_1$; promjenimo 1 u 2 kod onih kod kojih $p^2 | t - t_1$; promjenimo 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 točki **5.** stavimo 1 ili promjenimo 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 završimo sa svim prostim brojevima $p \leq B$, odbacimo sve $t^2 - n$, osim onih koji su postali jednaki 1 nakon dijeljenja sa svim potencijama prostih brojeva $p \leq B$. Dobit ćemo tako tablicu, u kojoj će stupac koji odgovara b_i imati vrijednosti elemenata $t^2 - n$ iz S koji su $\mathcal{B}$-brojevi, a ostali stupci će odgovarati vrijednostima $p \in \mathcal{B}$ za koje je $(\frac{n}{p}) = 1$.

9. Ostatak postupka je isti kao kod opće metode faktorske baze.

Primjer 5.5. Neka je $n = 1042387$. Uzmimo $B = 50$ i $A = 500$. Ovdje je $\lfloor\sqrt{n}\rfloor = 1020$. Naša faktorska baza se sastoji od 8 prostih brojeva $\{2, 3, 11, 17, 19, 23, 43, 47\}$. 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 rješenje $t_1 = t_{1,0} + t_{1,1} \cdot 3 + t_{1,2} \cdot 3^2 + \cdots + t_{1,\beta-1} \cdot 3^{\beta-1}$ kongruencije $t_1^2 \equiv 1042387 \pmod{3^\beta}$, $t_{1,j} \in \{0, 1, 2\}$. Možemo uzeti $t_{1,0} = 1$. Modulo 9 imamo: $(1 + 3t_{1,1})^2 \equiv 7 \pmod 9$, odakle je $t_{1,1} = 1$. Modulo 27 imamo: $(1+4+9t_{1,2})^2 \equiv 25 \pmod{27}$, 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 $t_2 = 3^6 - 589 = 140 \quad (t_2 \equiv 1112 \pmod{3^5})$.

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 (u stvari, za 729 nemamo skokova, već samo promjenimo 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 put se zaustavljamo kod skoka za 243.

Nakon što ovaj postupak primijenimo na preostalih 6 brojeva u faktorskoj bazi, dobit ćemo tablicu 500×8 u kojoj redci odgovaraju t-ovima između 1021 i 1520. Izbacimo li sve redke 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
1021	54	1	3						
1027	12342	1	1	2	1				
1030	18513		2	2	1				
1061	83334	1	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	
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. 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 peti i zadnji redak, dobivamo

$$(1112 \cdot 1520)^2 \equiv (3^3 \cdot 17 \cdot 23 \cdot 47)^2 \pmod{1042387},$$

$$647853^2 \equiv 496179^2 \pmod{1042387},$$

odakle dobivamo netrivijalni faktor $(647853 - 496179, 1042387) = 1487$. Drugi faktor je 701.

Očekivani broj operacija metodom kvadratnog sita je

$$O\Big(e^{\sqrt{\log n \, \log\log n}}\Big),$$

dakle, skoro isto kao kod faktorizacije pomoću eliptičkih krivulja. Spomenimo da je 1996. 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* (number field sieve) koja kombinira ideje iz metode kvadratnog sita i algebarsku teoriju brojeva. Kod ove metode je 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 broja Fermatovog broja $2^{2^9} + 1$. Ovom su metodom faktorizirani brojevi RSA-130 (1996. godine), te RSA-140 i RSA-155 (1999. godine).

Broj RSA-155 je u kolovozu 1999. godine faktorizirala grupa istraživača pod vodstvom Hermana te Riellea. Evo tog broja i njegove faktorizacije:

$$109417386415705274218097073220403576120037329454492059909138421314763499842889\backslash$$
$$347847179972578912673324976257528997818337970765372440271467435315933543338 97 =$$
$$102639592829741105772054196573991675900716567808038066803341933521790711307779 \cdot$$
$$\cdot 106603488380168454820927220360012878679207958575989291522270608237193062808643.$$