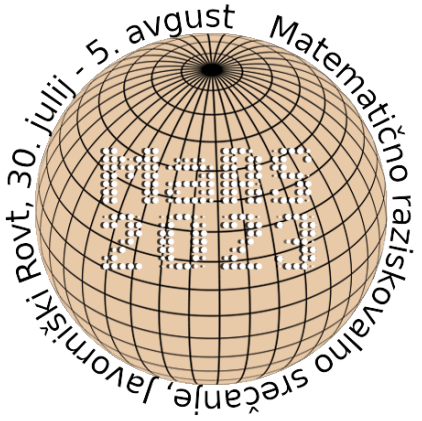
Hitra faktorizacija velikih števil

Lenart Dolinar, Kaja Rajter, Jakob Žorž
Mentor: Nino Cajnkar



Povzetek

Problem hitre faktorizacije velikih števil je eden od najpomembnejših problemov v sodobni kriptografiji, saj je temelj za reševanje problema diskretnega logaritma in sorodne oblike kodiranja. V nalogi smo obravnavali enega od hitrejših znanih algoritmov za faktorizacijo in sicer metodo eliptičnih krivulj (ECM) ter analizirali njegovo časovno zahtevnost.

1 Grupe

Definicija 1 ***Grupa*** *$(G,\circ)$ je par neprazne množice G in binarne operacije $\circ : G\times G\to G$, za katero veljajo sledeči aksiomi grup.*

- *Za vse $x, y, z \in G$ velja* asociativnost*: $x \circ (y \circ z) = (x \circ y) \circ z$.*
- *Obstaja tak element $e \in G$, da za vsak element $x \in G$ velja $x \circ e = e \circ x = x$. Element e imenujemo* enota *grupe.*

- *Za vsak element $x \in G$ obstaja tak x^{-1}, da je $x \circ x^{-1} = x^{-1} \circ x = 1$. Element $x^{-1} \in G$ imenujemo* inverz *elementa x.*

Definicija 2 *Če je operacija $\circ$ v grupi $(G, \circ)$ dodatno* komutativna*, torej za vse pare $x, y \in G$ velja $x \circ y = y \circ x$, je G* Abelova grupa.

Definicija 3 *Grupa je* ciklična*, če je generirana z enim samim elementom.* Red elementa *$x \in G$ je najmanjše naravno število n, da je $x^n = e$. Torej je ciklična grupa G oblike $G = \{e, x, x^2, \ldots, x^{n-1}\}$.*

Primer 1 *Naj bo množica G množica vseh celoštevilskih ostankov pri deljenju s praštevilom p in operacija $*$ množenje po modulu p. Potem rečemo, da je $(G, *)$* multiplikativna grupa *in jo označimo $\mathbb{Z}_p^\star$.*

Multiplikativna grupa je primer Abelove ciklične grupe, saj predstavlja množenje po modulu p in je generirana z enim samim elementom, t.j. p. Grupa $\mathbb{Z}_p^\star$ je torej oblike $\mathbb{Z}_p^\star = \{e, p, p^2, \ldots, p^{n-1}\}$.

1.1 Kolobarji

Definicija 4 *Naj bo množica K opremljena z binarnima operacijama seštevanja $(x, y) \to x + y$ in množenja $(x, y) \to xy$. Tako strukturo imenujemo* ***kolobar****, če velja*

- *$(K, +)$ je Abelova grupa,*
- *Za vse $x, y, z \in G$ velja asociativnost $x(yz) = (xy)z$ in obstaja tak element $e \in K$, da za vsak $x \in K$ velja $xe = ex = x$. Imenujemo ga enota kolobarja K,*
- *Velja* distributivnost*: za vse $x, y, z \in K$ velja $(x + y)z = xz + yz$ in $z(x + y) = zx + zy$.*

Kolobar, v katerem je operacija množenja komutativna, imenujemo ***komutativen kolobar****.*

1.2 Polja

Definicija 5 Polje *je komutativen kolobar, v katerem je vsak neničelen element obrnljiv.*

Polje je torej kolobar, z dodatnima pogojema, da je množenje komutativno in ima vsak element inverz.

Definicija 6 *Naj bo K polje. Polje L je* podpolje *polja K, če je L polje in velja $L \subseteq K$. K imenujemo* razširitev *polja L.*

Naj bo $a \in K$ in K razširitev polja L. Pravimo, da je a *algebraičen* nad L, če obstaja nekonstanten polinom $f(x) \in L[x]$, da je $f(a) = 0$. Razširitev K polja L je *algebraična*, če je vsak element $a \in K$ algebraičen nad L.
Polje K je *algebraično zaprto*, če je vsak nekonstanten polinom stopnje $n \geq 1$ nad K razcepen na linearne faktorje.

Izrek 1 *Naj bo I podgrupa za seštevanje kolobarja K z operacijo seštevanja $+$. Potem postane množica vseh odsekov $K/I = \{a + I \mid a \in K\}$ aditivna grupa z operacijo $(a + I) + (b + I) = (a + b) + I$.*

Če še dodatno velja $a + I = a' + I$ in $b + I = b' + I$, potem $ab + I = a'b' + I$, kar je ekvivalentno, da za vsaka $a, b \in K$ velja, da

$$ab \in I \implies a \in I \vee b \in I.$$

Definicija 7 *Če je K/I kolobar, potem I imenujemo* ideal. *Če je I ideal in K/I polje, potem I imenujemo* praideal.

2 Gladkost

Definicija 8 *Naj bosta B in n naravni števili, pri čemer ima n praštevilski razcep $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$. Število n je* B-gladko *natanko tedaj, ko za vsak $i \in \{1, \dots, k\}$ velja: $p_i < B$.*

S $\Psi(x, y)$ označimo število praštevil, ki so manjša od x in so y-gladka. Verjetnost, da je poljubno izbrano naravno število y-gladko, je

$$P_{gladko}(x, y) = \frac{\Psi(x, y)}{x}.$$

Izrek 2 *Naj bo $\pi(x) : \mathbb{R}^+ \to \mathbb{N}$ funkcija, ki pozitivnemu realnemu številu x priredi število vseh praštevil, manjših ali enakih x.* Izrek o praštevilih *nam pove, da se funkcija π asimptotsko obnaša enako kot $x \to \frac{x}{\log x}$, oziroma*

$$\lim_{x \to \infty} \frac{\pi(x)}{\frac{x}{\log x}} = 1.$$

Izrek 3 *Naj bo $\varepsilon > 0$ in $3 \leq u \leq (1 - \varepsilon)\frac{\log x}{\log \log x}$ ter x kot v prejšnjem izreku. Potem velja*

$$\Psi(x, x^{\frac{1}{u}}) = x \cdot u^{-u + o(u)},$$

pri čemer je $o(u)$ funkcija x in u, ki enakomerno konvergira proti 0, ko gre x proti neskončno.

Posledica 1 *Naj bo* $y \leq x_1, x_2 \leq x$ *in* $u = \frac{\log x}{\log y}$, *pri čemer u zadošča pogojem prejšnjega izreka. Potem velja, da je*

$$P_G(x_1, y) \cdot P_G(x_2, y) = P_G(x_1 \cdot x_2, y)^{1+o(u)}.$$

Dokaz: Naj bo $u_1 = \frac{\log x_1}{\log y}$ in $u_2 = \frac{\log x_2}{\log y}$. Brez škode za splošnost lahko predpostavimo, da je $u_2 \geq u_1$. Ker sta u_1 in u_2 večja ali enaka 1, vemo, da sta $\log u_1$ in $\log u_2$ večja od 0. Sledi, da je $\log(u_1 + u_2)$ večje od 0. Naj bo $L(u_1, u_2) = u_1 \cdot \log u_1 + u_2 \cdot \log u_2$ in $R(u_1, u_2) = (u_1 + u_2)\log(u_1 + u_2)$.

Ker je $\log(u_1 + u_2)$ večje od $\log u_1$ in od $\log u_2$, velja $L(u_1, u_2) \leq R(u_1, u_2)$. Če je $u_1 \geq \frac{u_2}{\log u_2}$, velja tudi $\log u_1 \geq \log u_2 - \log\log u_2$.
Ker je $u_1 = \frac{u_2}{\log u_2}$, dobimo:

$$\begin{aligned} \log u_2 + \log 2 &\geq \log(u_1 + u_2), \\ \log(2u_2) &\geq \log(u_1 + u_2), \\ L(u_1, u_2) &\geq (u_1, u_2) \cdot (\log(u_1 + u_2) - \log 2 - \log\log u_2. \end{aligned}$$

Ko gre u_2 proti neskončno, velja $L(u_1, u_2) \geq (1 - \varepsilon)R(u_1, u_2)$.

Če je $u_1 \leq \frac{u_2}{\log u_2}$, velja tudi $L(u_1, u_2) \geq u_2 \log u_2$ in dobimo

$$u_2 \cdot \log u_2 \cdot (1 + \frac{1}{\log u_2}) \cdot (1 + \log(1 + \frac{1}{\log u_2})) \geq (u_1 + u_2)\log(u_1 + u_2).$$

Ponovno, ko gre u_2 proti neskončno velja

$$R(u_1, u_2) \leq (1 + \varepsilon)u_2 \log u_2 \leq (1 + \varepsilon)L(u_1, u_2).$$

Sledi

$$P_G(x_1, y) \cdot P_G(x_2, y) = P_G(x_1 \cdot x_2, y)^{1+o(u)},$$

kar je točno to, kar smo želeli dokazati. □

3 L-notacija

Z algoritmom želimo faktorizirati velika števila v razumnem času.

Definicija 9 ***Časovna zahtevnost*** *je podatek o tem, koliko bitnih operacij bo program naredil pri danih vhodnih podatkih, preden bo vrnil rešitev.*

Časovno zahtevnost označimo z O notacijo, ki označuje red rasti problema.

Notacija	Zahtevnost
$O(1)$	konstantna
$O(\log n)$	logaritemska
$O(n)$	linearna
$O(n \log n)$	vmesna
$O(n^2)$	kvadratna
$O(n^c); c > 1$	polinomska
$O(c^n)$	eksponenta

Definicija 10 *Algoritmi, ki imajo večjo časovno zahtevnost od polinomske in manjšo od eksponentne glede na bitni zapis vnosa, so* subeksponentni.

Časovno zahtevnost algoritma lahko predstavimo z L-notacijo

$$L_x(\alpha, c) = exp\{c(\log x)^{\alpha}(\log \log x)^{1-\alpha}\}.$$

Izrek 4 *Naj bodo a, b, c in d pozitivna realna števila ter naj velja $a > c$. Potem velja*

$$P_G(L_x(a,b), L_x(c,d) = L_x\left(a - c, (a-c)\,\frac{b}{d}\right)^{-1+o(u)}.$$

L-notacije želimo tudi seštevati in množiti, kar lahko storimo s pomočjo formul iz naslednjega izreka.

Izrek 5 *Naj bosta (a, b) in (c, d) para pozitivnih realnih števil. Potem velja*

$$L_{\{L_x(a,b)\}} = L_x\left(ac, db^c a^{(1-c)}\right)^{(1+o(u))}$$

in

$$L_x(a,b) \cdot L_x(c,d) = \begin{cases} L_x(a,b)^{1+o(u)}; a > c \\ L_x(a, b+d); a = c \end{cases}.$$

Dodatno lahko predpostavimo, da je (a, b) leksikografsko večje od (c, d), torej velja, ko je $a > c$ ali $a = c$ in $b > d$. Potem velja

$$L_x(a,b) + L_x(c,d) = L_x(a,b)^{1+o(u)}.$$

4 Pollardova $p-1$ metoda

Pollardova $p-1$ metoda je algoritem za faktorizacijo velikih števil. Deluje na principu, da poišče take praštevilske faktorje, da velja, da je $p-1$ B_0-gladko. Koraki v algoritmu so sledeči.

1. Izberemo poljubno število $x_0 \in [2, N-2]$, tako da je $gcd(x_0, N) = 1$.
2. Izračunamo $M = \prod_{q \in \mathbb{P}, q \leq B_0} q^{\lfloor log_q N \rfloor}$.
3. Izberemo poljubno število tuje N.
4. Izračunamo $g = gcd(x_0^M - 1, N)$.
5. Če je $1 < g < N$, vrnemo g.
6. Če je $g = 1$, potem izberemo večjo mejo gladkosti B_0 in se vrnemo nazaj na 2. korak.
7. Drugače izberemo drugačen x_0 in ponovimo korake od začetka.

Primer 2 *Poskusili bomo faktorizirati število $N = 299$.*

1. *Za B_0 izberemo število 5.*
2. *Izračunamo $M = 2^2 \times 3^1 \times 5^1 = 60$.*
3. *Izberemo $x_0 = 2$.*
4. *$g = gcd(2^{60} - 1, 299) = 13$.*
5. *Ker je $1 < 13 < 299$, vrnemo 13.*
6. *Velja $299/13 = 23$, ker je praštevilo, torej dobimo popolno faktorizacijo; $299 = 13 \times 23$.*

5 Algoritem za hitro potenciranje

Želimo izračunati a^b za $a \in \mathbb{Z}, b \in \mathbb{N}$. Naivno se da b-krat pomnožiti a. Množenje porabi $O(1)$ korakov, torej cel algoritem porabi $O(b)$ časa. Izkaže

se, da obstaja veliko hitrejši algoritem, ki deluje za poljubne grupe. Oglejmo si primer izračuna a^{32}. To lahko naredimo na naslednji način:

$$\begin{aligned} a^2 &= a \cdot a \\ a^4 &= a^2 \cdot a^2 \\ a^8 &= a^4 \cdot a^4 \\ a^{16} &= a^8 \cdot a^8 \\ a^{32} &= a^{16} \cdot a^{16} \end{aligned}$$

Uporabili smo torej le 5 množenj, namesto 31, kot bi jih, če bi računali naiven način. Izkaže se, da je to mogoče za vse potence.

Izrek 6 *Naj bo $(G, \cdot)$ grupa za množenje in $a \in G$. Potem lahko a^n izračunamo v $O(\log n)$ časa.*

Dokaz: Naj bo $n = \sum_{i=0}^{k} b_i 2^i$ binarna razčlenitev števila n, torej je za $\forall i : b_i \in \{0, 1\}$ in k mora biti dovolj velik. Velja tudi $k \leq \log_2 n$, ker je 2^k največji člen v razčlenitvi števila n. Potem je $a^n = \prod_{i=0}^{k} a^{b_i 2^i}$. Torej je a^n produkt največ $k + 1$ potenc a^{2^i}. Potence a^{2^i} lahko izračunamo tako, da začnemo z $a^1 = a$ in potem $a^{2^i} = (a^{2^{i-1}})^2$. Torej lahko vse potence a^{2^i} izračunamo v $O(\log n)$ časa in posledično tudi a^n. □

6 Razširjen evklidov algoritem

Bezoutova identiteta pravi, da za vsaki števili $a, b \in \mathbb{Z}$ obstajata taki števili $x, y \in \mathbb{Z}$, da velja $ax + by = d$, kjer je $d = \gcd(a, b)$. V tem razdelku bomo spoznali algoritem, ki nam bo omogočil, da bomo našli x in y. Zaradi preglednosti bomo iskanje najmanjšega skupnega delitelja s pomočjo razširjenega Evklidovega algoritma označili s $gcdx(x, y)$.

Algoritem 1 $gcdx(a, b) \to (d, x, y)$

- *Če je $b = 0$, potem vrnemo $(a, 1, 0)$.*
- *Izračunamo $(d, x', y') = gcdx(b, a \bmod b)$.*
- *Vrnemo $(d, y', x' - \lfloor a/b \rfloor y')$.*

Dokaz: Če je $b = 0$, je $\gcd(a, b) = a$ in $ax + by = a$ za $x = 1$ in $y = 0$. Predpostavimo, da $gcdx(b, a \bmod b)$ vrne pravilen rezultat. Potem velja

$$bx' + (a \bmod b)y' = d.$$

Velja tudi zveza $a \pmod b = a - \lfloor a/b \rfloor b$, zato jo lahko vstavimo v zgornjo enačbo in dobimo

$$\begin{aligned} bx' + (a - \lfloor a/b \rfloor b)y' &= d, \\ bx' + ay' - \lfloor a/b \rfloor by' &= d, \\ ay' + b(x' - \lfloor a/b \rfloor y') &= d, \end{aligned}$$

Torej vrne tudi naš algoritem pravilni rezultat. □

Velja $gcdx(a, b) = gcdx(b, a)$, zaradi česar lahko brez škode za splošnost rečemo, da je $a \geq b$. Časovna zahtevnost je $O(\log a)$, ker se v vsakem koraku a zmanjša za vsaj polovico.
Za velika števila, ki so večja od 2^{64}, so operacije z njimi $O(\log n)$, zato je časovna zahtevnost algoritma $O(\log^2 a)$.

7 Algoritem Metode eliptičnih krivulj (ECM)

7.1 Eliptične krivulje

Definicija 11 Projektivna ravnina $\mathbb{P}^2$ *nad poljem* $\mathbb{F}$ *je kvocientni prostor* $\mathbb{F}^3 - \{0\}/\sim$, *kjer je ekvivalenčna relacija podana s predpisom* $(a, b, c) \sim (\alpha a, \alpha b, \alpha c)$ *za vsak* $\alpha \in \mathbb{F} - \{0\}$. *Točke v* $\mathbb{P}^2$ *so torej podane s homogenimi koordinatami* $[a, b, c] = [\alpha a, \alpha b, \alpha c]$ *za vse* $\alpha \neq 0$.

Definicija 12 *Polinom P je* homogen*, če velja*

$$P(\lambda x, \lambda y, \lambda z) = \lambda^d (x, y, z)$$

za vse $\lambda \in \mathbb{F}$.

Definicija 13 Algebraična krivulja*, podana s homogenim polinomom P, je množica točk* $C_p = \{A \in \mathbb{P}^2, P(A) = 0\}$. *Algebraična krivulja je* gladka*, če nima nobenih samopresečišč ali singularnosti.*

Definicija 14 Gladko kubično krivuljo *nad algebraično zaprtim poljem lahko zapišemo v kratki Weierstrassovi obliki:* $y^2 z = x^3 + axz^2 + bz^3$.

7.2 Algoritem

Algoritem metode eliptičnih krivulj ima dve fazi in sicer glavno fazo ter 1. fazo.

Algoritem 2 *Glavna Faza:*
Vhod: *eliptična krivulja $E_{W,A,B}$ nad $\mathbb{Q}$, točka P_0 neskončnega reda in meja gladkosti B_1.*

- *Izberemo poljubno eliptično krivuljo in neko poljubno netrivialno točko na njej. Imamo mejo gladkosti B_1. Število elementov v grupi točk na eliptični krivulji mora biti manjše od B_1.*
- *Izračunamo* $M = \prod_{q \in \mathbb{P} \wedge q \leq B_1} q^{\lfloor \log_q(N) \rfloor}$.
- *Seštejemo točke na eliptični krivulji* $P_0^M \pmod N$.
- *Izračunaj $gcdx(z, N)$.*
- *Če je $gcdx(z, N) \neq 1$, potem je $gcdx(z, N)$ faktor števila N.*

Faza 1:

- $B_1 = L_B\left(\frac{1}{2}, \frac{\sqrt{2}}{2}\right)$ *je dobra meja za gladkost.*
- *Ponavljaj, dokler ne najdeš prafaktorja:*
- $S = \{B_1$*-gladki elementi iz* $(p - \sqrt{p}, p + \sqrt{p})\}; u = |S|$.
  - *Izvedi glavno fazo na N, za mejo gladkosti B_1 in za eliptično krivuljo E.*

Dokaz: Pravilnost algoritma:

Obstaja praštevilo p, ki deli število N. Ker je red $E(\mathbb{F}_p)$ B_1-gladek, so vsi prafaktorji od $E(\mathbb{F}_p)$ B_1 gladki. Ker je B_1 majhen glede na N, so $\lfloor \log_2 N \rfloor$ za vsak q iz M višji ali enaki potencam praštevil iz razcepa $\mathbb{F}_p$. Posledično $|\mathbb{F}_p|$ deli M. Opazimo, da veljata enakosti $Z \equiv 0 \pmod p$ in $g \equiv 0 \pmod p$, kar pomeni, da je p hkrati faktor od Z in g, torej je res g praštevilski faktor od N. □

8 Časovna zahtevnost

Pokazali bomo, da je časovna kompleksnost celega algoritma enaka

$$T(ECM) = \log^3(N) L_B\left(\frac{1}{2}, \sqrt{2}\right).$$

Najprej pogledamo glavno fazo. Tukaj imamo tri korake, ki nezanemarljivo prispevajo h končni časovni zahtevnosti algoritma:

- $M = \prod_{q \in \mathbb{P} \wedge q \leq B_1} q^{\lfloor \log_q(N) \rfloor}$,
- $P_0^M \pmod N$,
- $xgcd(z, N)$.

Vzamemo definicijo M in logaritmiramo obe strani:

$$\begin{aligned}
\log M &= \sum_{q \in \mathbb{P} \wedge q \leq B_1} \lfloor \log_q N \rfloor \log q \\
&\leq \log(N) \sum_{q \in \mathbb{P} \wedge q \leq B_1} \log q \\
&\leq \log(N) \sum_{q \in \mathbb{P} \wedge q \leq B_1} \log B_1 \\
&= \log(N) \frac{B_1}{\log B_1} \log B_1 \\
&= B_1 \log(N).
\end{aligned}$$

Torej je časovna zahtevnost druge operacije $O(B_1 \log N)$. Časovna zahtevnost tretje operacije je $O(\log^2 N)$, ker je to časovna zahtevnost potenciranja za velike številke. Zaradi implementacije se časovni zahtevnosti zmnožita, torej je časovna zahtevnost glavne faze $O(B_1 \log^3 N)$.

Naj bo $P_B(B_1)$ verjetnost, da bo algoritem našel praštevilo p v fazi 1, če je p B_1-gladko.

Izrek 7 *Obstaja pozitivna izračunljiva konsatnta c, da velja:*
Naj bodo $n, v, w \in \mathbb{Z}_{a>1}$. Števila n, v, w so taki, da ima n vsaj dva različna praštevilska delitelja, ki sta večja od 3. Za manjši praštevilski delitelj od n, ki je večji od 3, velja $p \leq v$. Označimo

$$u = |\{s \in \mathbb{Z} : |s - (p+1)| \leq \sqrt{p} \text{ in vsak praštevilski delitelj od } s \text{ je } \leq w\}|.$$

Z N označimo število trojic $(a, x, y) \in (\mathbb{Z}_n)^3$ za katere je ECM uspešen. Potem velja

$$\frac{N}{n^3} > \frac{c(u-2)}{(\log p)(2\sqrt{p}+1))}.$$

Potem je $P_B(B_1) \geq \frac{c(u-2)}{(\log p)(2\sqrt{p}+1))}$, kar se upošteva pri O notaciji. Časovna zahtevnost celega algoritma je torej

$$T(ECM) = B_1 \log^3(N) P_G^{-1}(B, B_1).$$

Nadalje izberemo mejo gladkosti B_1 tako, da bo časovna zahtevnost minimalna. To se zgodi natanko tedaj, ko velja $B_1 = L_B(\alpha, c)$. Sedaj izračunamo verjetnost, da je poljubno izbrano število manjše od B, B_1-gladko

$$\begin{aligned} P_G(B, B_1) &= (L_B(1,1), L_B(\alpha, c)) = \\ &= L_B(1-\alpha, \frac{1-\alpha}{c})^{1+o(1)} \end{aligned}$$

Najboljša vrednost α je $\frac{1}{2}$, torej je časovna zahtevnost

$$T(ECM) = \log^3(N) L_B\left(\frac{1}{2}, \sqrt{2}\right).$$

Faktor $\log^3(N)$ je zanemarljiv, zato je

$$T(ECM) = L_B\left(\frac{1}{2}, \sqrt{2}\right).$$