Klasični i kvantni napadi na problem diskretnog logaritma

Vlašić, Petar

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Petar Vlašić

KLASIČNI I KVANTNI NAPADI NA PROBLEM DISKRETNOG LOGARITMA

Diplomski rad

Voditelj rada:
prof. dr. sc. Andrej Dujella

Zagreb, Rujan 2018.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Zahvaljujem se svojoj materi, ćaći i ujaku koji su me trpili kroz moje neuspjehe i veselili se mojim uspjesima. Hvala i profesoru Dujelli, koji mi je bio od velike pomoći prilikom pisanja ovog rada, te čija su me zanimljiva predavanja potaknula da odaberem ovu temu.

Sadržaj

Uvod

Poznato je da postoje problemi koji su rješivi u teoriji ali nisu u praksi, takve probleme nazivamo neukrotivima i teškima. Neukrotivim problemima smatramo one za koje nije pronađeno rješenje polinomne vremenske složenosti na deterministočkom Turingovom stroju. Takav je i problem diskretnog logaritma. Pronađeni su kvantni algoritmi koji ga rješavaju u polinomnoj vremenskoj složenosti.

U ovom radu prvo ćemo objasniti razliku između klasične i kvantne izračunljivosti i uvesti ćemo pojmove koji će nam poslužiti poslije. Korištena literatura je Mladen Vuković. *Složenost algoritama*. Zagreb, PMF-Matematički odsjek, 2011, 36-52, 89-92 i Song Y. Yan. *Quantum Computational Number Theory* Springer, 2015, 33-56.

Nakon toga ćemo analizirati neke klasične napade poput Shanksovog Baby-Step Giant-Step algoritma, Silver-Pohlig-Hellman algoritma, ρ - metode za problem diskretnog logaritma, Index Calculus algoritam, Gordonov NFS algoritam. Korištena literatura je A. Godušová. *Number Field Sieve for Discrete Logarithm*. Charles University in Prague - Faculty of Mathematics and Physics, 2015, A. Joux and R. Lercier. *Number field sieve for the DLP. in: H.C.A van Tilborg, S. Jajodia* (Eds.): *Encyclopedia of Cryptography and security*. Springer, 2011, 867-873, Song Y. Yan. *Quantum Attacks on Public-Key Cryptosystems*. Springer, 2013, 93-109.

Nakon što opišemo klasične napade prvo ćemo opisati lakši slučaj kvantnog napada, zatim i općeniti slučaj kvantnog napada. Korištena literatura je F. X. Lin *Shor's Algorithm and the Quantum Fourier Transform*. McGill University, 2014, 5-11 i Song Y. Yan. *Quantum Attacks on Public-Key Cryptosystems*. Springer, 2013, 122-131. Posebno nas zanima vremenska i prostorna složenost tih napada.

Diplomski rad je napravljen u sklopu aktivnosti Projekta KK.01.1.1.01.0004 - Znanstveni centar izvrsnosti za kvantne i kompleksne sustave te reprezentacije Liejevih algebri.

Poglavlje 1

Klasična i kvantna izračunljivost

1.1 Klasična izračunljivost

Da bi pričali o izračunljivosti prvo moramo definirati Turingov stroj. Ideju i teoriju Turingovog stroja prvi je iznio i proučavao engleski matematičar Alan Turing, slijedi formalna definicija Turingovog stroja, Turing-prepoznatljivosti jezika i Turing-odlučivosti jezika.

Definicija 1.1.1. *Standardni višetračni Turingov stroj je uređena sedmorka* $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, q_D A, q_N E)$*, gdje je redom:*

- Q *konačan skup čije elemente nazivamo stanja*
- Σ *konačan skup, čije elemente nazivamo ulazni simboli. Pretpostavljamo da* Σ *ne sadrži "prazan" simbol kojeg označavamo sa* $\square$
- Γ *je konačan skup simbola koje nazivamo abeceda trake*
- δ *je funkcija prijelaza koju definiramo ovako:*
  - *ako je M deterministički Turingov stroj, onda*

    $$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k$$

  - *ako je M nedeterministički Turingov stroj, onda*

    $$\delta : Q \times \Gamma^k \to 2^{Q \times \Gamma^k \times \{L,R\}^k},$$

    gdje L i R specificiraju smjer u kojem se glava kreće. Kad je $k = 1$ *onda imamo klasični jednotračni Turingov stroj.*
- $\square \in \Gamma$ *je poseban znak kojeg nazivamo praznina*

- $q_0 \in Q$ *je početno stanje*
- $q_D A \in Q$ *je stanje prihvaćanja*
- $q_N E \in Q$ *je stanje odbijanja*

Definicija 1.1.2. *Kažemo da Turingov stroj* $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, q_{DA}, q_{NE})$ *prepoznaje neku riječ* $w \in \Gamma^*$ *ako postoji konačan niz parova* $(r_0, s_0), (r_1, s_1), ..., (r_m, s_m) \in Q \times \Gamma$*, te konačan niz simbola* $I_1, ..., I_m \in \{L, D\}$ *tako da vrijedi:*

1. $r_0 = q_0$ *i* s_0 *je prvi lijevi simbol riječi w;*
2. *za svaki* $j \in \{0, ..., m-1\}$ *imamo* $\delta(r_j, s_j) = (r_{j+1}, s_{j+1}, I_{j+1})$ *i* $r_j \notin \{q_D A, q_N E\}$
3. $r_m = q_{DA}$

Za proizvoljan Turingov stroj M sa L(M) označavamo skup svih riječi koje M prepoznaje. Kažemo da Turingov stroj M prepoznaje jezik L ako vrijedi $L = L(M)$.

Za jezik kažemo da je Turing-prepoznatljiv ako postoji Turingov stroj koji ga prepoznaje. Takve jezike također nazivamo rekurzivno prebrojivima.

Definicija 1.1.3. *Za neki jezik* $L \subseteq \Gamma^*$ *kažemo da je Turing-odlučiv ako postoji Turingov stroj M koji ga prepoznaje, te za svaku riječ* $w \in \Gamma^* \backslash L$ *stroj M s ulazom w staje u završnom stanju* q_{NE}*. Ako neki jezik nije odlučiv tada kažemo da je neodlučiv.*

Turingovi strojevi koji staju za svaki ulaz su dobri modeli za algoritam, dobro definirani niz koraka koji uvijek staje i daje odgovor. Ako algoritam za dani problem postoji, onda je problem odlučiv.

Sljedeće definicije formaliziraju klase složenosti bazirane na Turingovim strojevima.

Definicija 1.1.4. $\mathcal{P}$ *je klasa problema koje deterministički Turingov stroj rješava u polinomnom vremenu. Probleme koji spadaju u ovu klasu smatramo lakima za riješiti na računalu.*

Definicija 1.1.5. $\mathcal{NP}$ *je klasa problema koje nedeterministički Turingov stroj rješava u polinomnom vremenu. Probleme u ovoj klasi smatramo neukrotivima i teškima za riješiti na računalu.*

Kad govorimo u terminima jezika, $\mathcal{P}$ je klasa jezika koji se mogu odlučiti u polinomnom vremenu dok je $\mathcal{NP}$ klasa jezika koji se mogu prepoznati u polinomnom vremenu.

Definicija 1.1.6. $\mathcal{EXP}$ *je klasa problema koji su rješivi na determinističkom Turingovom stroju u vremenu ograničenom sa* 2^{n^i}.

Definicija 1.1.7. *Funkcija f je izračunljiva u polinomnom vremenu ako za svaki input w, $f(w)$ staje na Turingovom stroju u polinomnom vremenu. Jezik A je polinomno svediv na jezik B, što označavamo $A \leq_{\mathcal{P}} B$, ako postoji funkcija izračunljiva u polinomnom vremenu takva da za svaki input w vrijedi:*

$$w \in A \iff f(w) \in B.$$

Funkcija f se zove polinomna redukcija iz A u B.

Definicija 1.1.8. *Problem/jezik L je $\mathcal{NP}$-potpun ako zadovoljava sljedeća dva uvjeta:*

- $L \in \mathcal{NP}$
- $\forall A \in \mathcal{NP}, A \leq_{\mathcal{P}} L$.

Definicija 1.1.9. *Problem D je $\mathcal{NP}$-težak ako zadovoljava sljedeći uvjet:*

$$\forall A \in \mathcal{NP}, A \leq_{\mathcal{P}} D$$

gdje D može biti u $\mathcal{NP}$ ili ne mora biti. Dakle, $\mathcal{NP}$-težak znači da je težak kao neki $\mathcal{NP}$ problem ili teži.

Definicija 1.1.10. *$\mathcal{ZPP}$ je klasa problema rješiva u očekivajućem polinomnom vremenu, gdje je vjerojatnost za grešku nula, na vjerojatnosnom Turingovom stroju.*

Definicija 1.1.11. *$\mathcal{BPP}$ je klasa problema rješiva u očekivajućem polinomnom vremenu na vjerojatnosnom Turingovom stroju gdje je vjerojatnost za točan odgovor $\frac{1}{2} + \delta$ gdje je $\delta > 0$ neka fiksna vrijednost.*

Slično možemo definirati klase problema kao što su $\mathcal{P}$-Space, $\mathcal{P}$-Space potpun i $\mathcal{P}$-Space težak.

Definicija 1.1.12. *Neka je M neki Turingov stroj koji staje za svaki ulaz. Prostorna složenost Turingovog stroja M je funkcija $space_M : \mathbb{N} \to \mathbb{N}$, gdje je $space_T(n)$ maksimalan broj registara na traci po kojima glava stroja T čita, za svaki ulazni podatak duljine n.*

Definicija 1.1.13. *Za svaku funkciju $f : \mathbb{N} \to \mathbb{R}^+$ definiramo pripadne prostorne klase složenosti ovako:*

$\mathcal{D}$-Space$(f(n))$ = {$L : L$ je jezik odlučiv na nekom $\mathcal{O}(f(n))$-prostorno složenom determinističkom Turingovom stroju}

$\mathcal{N}$-Space$(f(n))$ = {$L : L$ je jezik odlučiv na nekom $\mathcal{O}(f(n))$-prostorno složenom nedeterminističkom Turingovom stroju}

Definicija 1.1.14. *Definiramo:*

$$\mathcal{P}\text{-Space} = \bigcup_{k\in\mathbb{N}} \mathcal{D}\text{-Space}(n^k)$$

$$\mathcal{NP}\text{-Space} = \bigcup_{k\in\mathbb{N}} \mathcal{N}\text{-Space}(n^k)$$

Definicija 1.1.15. *Problem/jezik L je $\mathcal{P}$-Space potpun ako zadovoljava sljedeća dva uvjeta:*

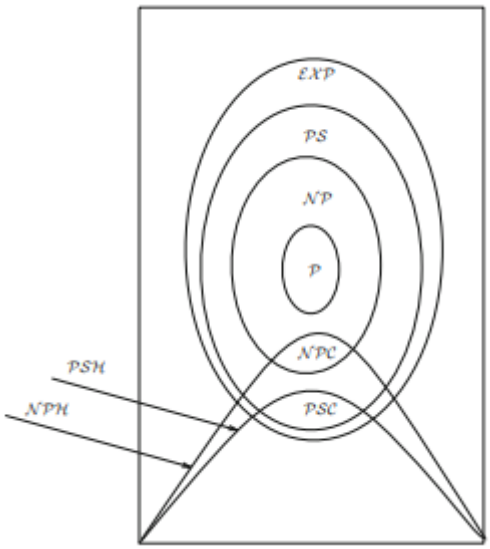
- $L \in \mathcal{P}$-*Space*
- $\forall A \in \mathcal{P}\text{-Space}, A \leq_{\mathcal{P}} L$.

Definicija 1.1.16. *Problem L je $\mathcal{P}$-Space težak ako zadovoljava sljedeći uvjet:*

$$\forall A \in \mathcal{P}\text{-Space}, A \leq_{\mathcal{P}} L$$

gdje L može biti u $\mathcal{P}$-Space ili ne mora biti.

Koristiti ćemo oznake $\mathcal{NP}$C za skup $\mathcal{NP}$-potpunih problema, $\mathcal{PS}$C za skup $\mathcal{P}$-Space potpunih problema, $\mathcal{NP}$H za skup $\mathcal{NP}$-teških problema i $\mathcal{PS}$H za skup $\mathcal{P}$-Space teških problema. Odnose između ovih klasa problema vidimo na slici 1.1.1.



Slika 1.1.1: Odnosi između navedenih klasa složenosti.

1.2 Kvantni sustavi

Kvantna računala su strojevi koji se oslanjaju na karakteristike kvantnih fenomena poput kvantnih interferencija i kvantnih sprezanja. Dok konvencionalna računala rade s *bitovima* koji isključivo imaju stanje 0 ili 1, kvantna računala rade s *kubitovima*. Stanje kvantnog računala opisuje se s baznim vektorom u Hilbertovom prostoru. Slijedi formalna definicija:

Definicija 1.2.1. *Kubit je kvantni sustav* $|\Psi\rangle$ *koje ima formu:*

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

gdje su $\alpha, \beta \in \mathbb{C}$*, takvi da* $\| \alpha \|^2 + \| \beta \|^2 = 1$*,* $|0\rangle$ *i* $|1\rangle$ *su bazni vektori Hilbertovog prostora.*

Kubit je kvantni sustav s dva stanja $|0\rangle$ i $|1\rangle$, čisto stanje kubita je linearna superpozicija tih dvaju stanja, ali kad mjerimo stanje kubita dobivamo jedno od ta dva stanja, takva diskretnost se naziva kvantizacija. Klasično se računalo može nalaziti u 2^N različitih stanja gdje je N broj bitova i njegovo stanje mjerenjem (očitavanjem) ostaje nepromijenjeno dok se kvantno može nalaziti u beskonačno mnogo različitih stanja, to su linearne superpozicije 2^N stanja računalne baze, mjerenjem dobivamo neko od 2^N stanja računalne baze, znači da mjerenjem mijenjamo stanje kvantnog računala.

Ako imamo kvantni sustav sa k stanja on može biti u $|c_1\rangle$, $|c_2\rangle$, ...,$|c_k\rangle$ ali također i u stanju superpozicije:

$$|\Psi\rangle = \sum_{i=0}^{2^k-1} \alpha_i |c_i\rangle,$$

gdje su $\alpha_i \in \mathbb{C}$, i $\sum_i \| \alpha_i \|^2 = 1$ i svaki $|c_i\rangle$ je bazni vektor Hilbertovog prostora.

Definicija 1.2.2. *Kvantni registar, ili općenitije kvantno računalo, je uređeni skup konačnog broja kubitova.*

Da bi fizički sustav izračunavao moramo moći promijeniti stanje sustava. Ovo se postiže nizom unitarnih transformacija vektora $|\Psi\rangle$ pomoću unitarnih matrica. To su matrice čiji inverz dobijemo konjugiranjem njihove transponirane matrice. Pretpostavimo da se izračunavanje odvija na kvantnom sustavu s jednim kubitom, tada je superpozicija dana sa:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

gdje su $\alpha, \beta \in \mathbb{C}$ i $\| \alpha \|^2 + \| \beta \|^2 = 1$, s dva moguća stanja $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ i $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Neka je unitarna matrica M:

$$M = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Tada kvantne operacije na kubitu možemo zapisati ovako:

$$M|0\rangle = |0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

$$M|1\rangle = |0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Ovo su, ustvari, kvantna vrata (analogno logičkim vratima). Logička vrata možemo smatrati logičkim operatorima. Tako bi operator NOT definirali kao matricu:

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Slično kvantna vrata za dva kubita definiramo ovako:

$$\begin{aligned} |00\rangle &\to |00\rangle \\ |01\rangle &\to |01\rangle \\ |10\rangle &\to \frac{1}{\sqrt{2}}|10\rangle + \frac{1}{\sqrt{2}}|11\rangle \\ |11\rangle &\to \frac{1}{\sqrt{2}}|10\rangle - \frac{1}{\sqrt{2}}|11\rangle \end{aligned}$$

Ustvari, definiramo unitarnu matricu M ovako:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

1.3 Kvantna izračunljivost i složenost

Kvantni Turingov stroj je generalizacija vjerojatnosnog Turingovog stroja, kod kojeg svaka ćelija na traci sadrži kubit. Neka je $\overline{\mathbb{C}}$ koji se sastoji od $\alpha \in \mathbb{C}$ takvih da Turingov stroj izračunava realni i imaginarni dio od α unutar 2^{-n} u vremenu polinomnom u n, tada se kvantni Turingov stroj može definirati kao:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F),$$

gdje

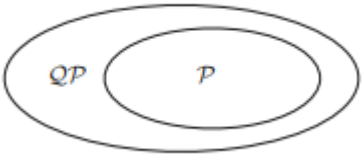
$$\delta : Q \times \Gamma \to \overline{\mathbb{C}}^{Q\times\Gamma\times\{L,R\}},$$

a ostatak je isti kao u vjerojatnosnom Turingovom stroju.

Definicija 1.3.1. *$\mathcal{QP}$ je klasa problema rješiva sa sigurnošću u polinomnom vremenu na kvantnom Turingovom stroju.*

Definicija 1.3.2. *$\mathcal{ZQP}$ je klasa problema rješiva na kvantnom Turingovom stroju u očekivajućem polinomnom vremenu gdje je vjerojatnost za grešku nula.*

Definicija 1.3.3. *$\mathcal{BQP}$ je klasa problema rješiva na kvantnom Turingovom stroju u polinomnom vremenu s ograničenom vjerojatnošću $\epsilon < \frac{1}{3}$ za grešku.*



Slika 1.3.1: Odnos između $\mathcal{P}$ i $\mathcal{QP}$.

Slika 1.3.2: Odnos između $\mathcal{ZQP}$ i $\mathcal{ZPP}$

.

Poznato je da vrijedi $\mathcal{P} \subseteq \mathcal{BPP} \subseteq \mathcal{BQP} \subseteq \mathcal{P}$-Space. Prema tome nije poznato je li kvantni Turingov stroj jači od vjerojatnosnog Turingovog stroja. Također nije poznata veza između $\mathcal{BQP}$ i $\mathcal{NP}$.

Poglavlje 2

Klasični napadi na DLP

2.1 Osnovni koncepti

Problem diskretnog algoritma ponekad ćemo označavati kao DLP (discrete logarithm problem).

Definicija 2.1.1. *DLP opisujemo ovako:*

$$\begin{cases} \text{Ulaz: } a, b, n \in \mathbb{N} \\ \text{Izlaz: } x \in \mathbb{N} \text{ takav da } a^x \equiv b (\text{mod } n), \text{ ako takav } x \text{ postoji} \end{cases} \tag{2.1.1}$$

gdje n može biti složen ili prost broj.

Unatoč što je ovaj problem dugo poznat, nije poznat efikasan algoritam koji ga rješava. Najbolji poznati algoritam je Gordonov NFS algoritam čija je vremenska složenost:

$$O\Big(\exp\Big(c(\log n)^{\frac{1}{3}} (\log\log n)^{\frac{2}{3}} \Big) \Big)$$

Postoje tri kategorije algoritama koje se koriste za rješavanje DLP-a:

1. Algoritmi koji rade za proizvoljne grupe, dakle oni koji ne iskorištavaju nikakva posebna svojstva grupa. To su Shanksova baby-step giant-step metoda, Pollardova ρ-metoda i λ-metoda.

2. Algoritmi koji rade dobro u konačnim grupama za koje red grupe nema velike proste djelitelje, tj. za grupe sa glatkim redom. Kažemo da je prirodan broj gladak ako nema velikih prostih djelitelja, a da je y-gladak ako nema prostih djelitelja većih od y. Silver-Pohlig-Hellman algoritam baziran na Kineskom teoremu o ostacima je u ovoj kategoriji.

3. Algoritmi koji iskorištavaju metode za reprezentiranje elemenata grupe kao produkata elemenata iz relativno malog skupa. U ovu grupu spadaju Index Caluculus algoritam i Gordonov NFS algoritam.

U sljedećim poglavljima ćemo se upoznati s nekim od ovih algoritama.

2.2 Shanksov Baby-Step Giant-Step algoritam

Neka je G konačna ciklička grupa reda n, a generator od G i $b \in G$. Očiti algoritam za izračunavanje potencija od a dok se b ne pronađe zahtjeva $O(n)$ operacija. Na primjer da bi izračunali $x = \log_2 15 \bmod 19$, moramo izračunavati $2^x \bmod 1$ za $x = 1, 2, ..., 19 - 1$ sve dok ne vrijedi $2^x \bmod 19 = 15$. Tako dobijemo da je rješenje 11. Jasno je da je ovo očito rješenje neefikasno za veliki n. Bolji algoritam od ovog očitog je Shanksov baby-step giant-step algoritam.

Neka je $m = \lfloor \sqrt{n} \rfloor$. Ovaj algoritam se temelji na činjenici da ako je $x = \log_a b$, onda možemo jedinstveno napisati $x = i + jm$, gdje $0 \leq i, j < m$. Na primjer, ako je $11 = \log_2 15 \bmod 19$, onda je $a = 2$, $b = 15$, $m = 4$, pa možemo pisati $11 = i + 4j$, gdje je očito $i = 1$ i $j = 2$. Slijedi opis algoritma:

Algoritam 2.2.1. *Ovaj algoritam računa diskretni logaritam x od y za bazu a modulo n takav da* $y = a^x \bmod n$*:*

1. (Inicijalizacija) Računamo $s = \lfloor \sqrt{n} \rfloor$.

2. (Računanje malog koraka) Izračunavamo prvi niz uređenih parova (ya^r, r), $r = 0, 1, 2, 3, ..., s - 1$, niz je označen sa S:

$$S = \{(y, 0), (ya, 1), (ya^2, 2), ..., (ya^{s-1}, s - 1) \bmod n\}$$

   i sortiramo S po ya^r, prvom elementu uređenih parova u S.

3. (Računanje velikog koraka) Izračunavamo drugi niz uređenih parova $(a^{ts}, ts), t = 1, 2, 3, ..., s$:

$$T = \{(a^s, 1), (a^{2s}, 2), ..., (a^{s^2}, s) \bmod n\}$$

   i sortiramo T po a^{ts}, prvom elementu parova u T.

4. (Traženje, uspoređivanje, izračunavanje) Pretražujemo oba niza S i T dok ne nađemo da je $ya^r = a^{ts}$ gdje je ya^r u S i a^{ts} u T i onda izračunamo $x = ts - r$ koji je tražena vrijednost.

Ovaj algoritam zahtjeva tablicu sa $O(m)$ ulaza gdje je $m = \lfloor \sqrt{n} \rfloor$. Koristeći algoritam za sortiranje možemo sortirati nizove S i T u $O(m \log m)$ operacija s Quicksort algoritmom. To znači da nam ovo daje algoritam za izračunavanje diskretnih logaritama koji koristi $O(\sqrt{n} \log n)$ vremena i $O(\sqrt{n})$ prostora. Ovaj algoritam radi za proizvoljne grupe, a ako je red grupe veći od 10^{40} onda postaje neprimjenjiv u praksi.

Primjer 2.2.1. *Pretpostavimo sad da želimo pronaći diskretan logaritam* $\log_{49} 70 \bmod 97$ *takav da* $70 = 49^x \bmod 97$. *Koristeći algoritam 2.2.1, imamo:*

1. $y = 70,\ a = 49,\ n = 97,\ s = \lfloor 97 \rfloor = 9$

2. Računamo mali korak:

$$\begin{aligned} S &= \{(y, 0), (ya, 1), ..., (ya^8, 8) \bmod 97\} \\ &= \{(70, 0), (35, 1), (66, 2), (33, 3), (65, 4), (81, 5), (89, 6), (93, 7), (95, 8)\} \\ &= \{(33, 3), (35, 1), (65, 4), (66, 2), (70, 0), (81, 5), (89, 6), (93, 7), (95, 8)\} \end{aligned}$$

3. Računamo veliki korak:

$$\begin{aligned} T &= \{(a^s, s), (a^{2s}, 2s), (a^{3s}, 3s), ..., (a^{10s}, 10s) \bmod 113\} \\ &= \{(18, 9), (33, 18), (12, 27), (22, 36), (8, 45), (47, 54), (70, 63), (96, 72), (79, 81)\} \\ &= \{(8, 45), (12, 27), (18, 9), (22, 36), (33, 18), (47, 54), (70, 63), (79, 81), (96, 72)\} \end{aligned}$$

4. Uspoređivanje i izračunavanje: Broj 70 je zajednička vrijednost prvog elementa u parovima od obe liste S i T sa $r = 0$ i $st = 63$, pa je $x = st - r = 63 - 0 = 63$. pa vrijedi $\log_{49} 70 \bmod 97 = 63$ ili ekvivalentno $49^{63} \bmod 97 = 70$.

2.3 Silver-Pohlig-Hellman algoritam

1978. godine Pohlig i Hellman su predložili važan algoritam, danas široko poznat kao Silver-Pohlig-Hellman algoritam za računanje DLP-a u polju $GF(q)$, gdje je $GF(q)$ polje koje se sastoji od cijelih brojeva modulo q, gdje je q prost broj. Algoritam radi $O(\sqrt{p})$ operacija i zauzima usporedivu količinu prostora, gdje je p najveći prosti faktor od $q - 1$. Pohlig i Hellman su pokazali da ako je

$$q - 1 = \prod_{i=1}^{k} {p_i}^{\alpha_i},$$

gdje su p_i različiti prosti brojevi i α_i prirodni brojevi, i ako su $r_1, ..., r_k$ bilo koji realni brojevi takvi da $0 \leq r_i \leq 1$, onda se logaritmi u $GF(q)$ mogu izračunati u

$$O\left(\sum_{i=1}^{k}\left(\log q + p_i^{1-r_i}(1+\log p_i^{r_i})\right)\right)$$

operacija na polju, koristeći

$$O\left(\log q \sum_{i=1}^{k}\left(1+p_i^{r_i}\right)\right)$$

bitova memorije, uz uvjet da se predizračunavanje koje zahtjeva

$$O\left(\sum_{i=1}^{k} p_i^{r_i}\log p_i^{r_i} + \log q\right)$$

operacija na polju obavi prvo. Ovaj algoritam je jako učinkovit ako je q "gladak" t.j. ako su svi prosti faktori od $q-1$ mali. Slijedi opis algoritma:

Algoritam 2.3.1. *Ovaj algoritam izračunava diskretni logaritam* $x = \log_a b \bmod q$*:*

1. $q-1$ rastavljamo na proste faktore:

$$q - 1 = \prod_{i=1}^{k} p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}.$$

2. Predizračunavamo tablicu $r_{p_i,j}$ za dano polje:

$$r_{p_i,j} = a^{j(q-1)/p_i} \bmod q,\ 0 \leq j < p_i.$$

3. Izračunavamo $x = \log_a b \bmod q$:

   a) Koristeći ideju sličnu kao baby-step giant-step algoritmu da pronađemo diskretne logaritme $x \bmod p_i^{\alpha_i}$. Da bi ovo izračunali promatramo sljedeću reprezentaciju ovog broja:

   $$x \bmod p_i^{\alpha_i} = x_0 + x_1 p_i + ... + x_{\alpha_i - 1} p_i^{\alpha_i - 1},$$

   gdje je $0 \leq x_n < p_i - 1$

      i. Da bi pronašli x_0, računamo $b^{(q-1)/p_i}$ što je jednako $r_{p_i,j}$ za neki j, i postavljamo $x_0 = j$ za koji

      $$b^{(q-1)/p_i} \bmod q = r_{p_i,j}.$$

      Ovo je moguće zato što vrijedi:

      $$b^{(q-1)/p_i} \equiv a^{x(q-1)/p} \equiv a^{x_0(q-1)/p} \bmod q = r_{p_i,x_0}.$$

ii. Da pronađemo x_1 računamo $b_1 = ba^{-x_0}$. Ako vrijedi

$$b_1^{(q-1)/p_i^2} \bmod q = r_{p_i,j},$$

onda postavljamo $x_1 = j$. Ovo je moguće zbog

$$b_1^{(q-1)/p_i^2} \equiv a^{(x-x_0)(q-1)/p_i^2} \equiv a^{(x_1+x_2p_i+\ldots)(q-1)/p_i} \equiv a^{x_1(q-1)/p} \bmod q = r_{p_i,x_1}.$$

iii. Da bi dobili x_2 promatramo broj $b_2 = ba^{-x_0-x_1p_i}$ i računamo

$$b_2^{(q-1)/p_i^3} \bmod q.$$

Postupak nastavljamo induktivno dok ne pronađemo sve $x_0, x_1, ..., x_{\alpha_i-1}$.

b) Koristimo Kineski teorem o ostatcima da bi pronašli jedinstvenu vrijednost od x iz kongruencija $x \bmod p_i^{\alpha_i}$.

Sada dajemo primjer kako ovaj algoritam funkcionira na konkretnom primjeru.

Primjer 2.3.1. *Pretpostavimo da treba izračunati diskretni logaritam* $x = \log_{15} 131 \bmod 337$. *Sad imamo* $a = 15$, $b = 131$, *i* $q = 337$ *(15 je generator od* $\mathbb{F}^*_{337}$*). Rješavamo ovaj problem po upravo opisanom algoritmu:*

1. Faktoriziramo $q - 1$:

$$336 = 2^4 \cdot 3 \cdot 7.$$

2. Računamo $r_{p_i,j}$ za dano polje $\mathbb{F}^*_{337}$ po formuli:

$$r_{p_i,j} = a^{j(q-1)/p_i} \bmod q,\ 0 \le j < p_i.$$

Ovo se samo jednom treba izračunati za ovo polje.

a) Računamo

$$\begin{aligned} r_{p_1,j} &= a^{j(q-1)/p_1} \bmod q = 15^{168j} \bmod 337 \text{ za } 0 \le j < p_1 = 2 \\ r_{2,0} &= 15^{168\cdot 0} \bmod 337 = 1 \\ r_{2,1} &= 15^{168\cdot 1} \bmod 337 = 336 \end{aligned} \tag{2.3.1}$$

b) Računamo

$$\begin{aligned} r_{p_2,j} &= a^{j(q-1)/p_2} \bmod q = 15^{112j} \bmod 337 \text{ za } 0 \le j < p_2 = 3 \\ r_{3,0} &= 15^{112\cdot 0} \bmod 337 = 1 \\ r_{3,1} &= 15^{112\cdot 1} \bmod 337 = 208 \\ r_{3,2} &= 15^{112\cdot 2} \bmod 337 = 128 \end{aligned} \tag{2.3.2}$$

c) Računamo

$$\begin{aligned} r_{p_3,j} &= a^{j(q-1)/p_3} \bmod q = 15^{48j} \bmod 337 \text{ za } 0 \le j < p_3 = 7 \\ r_{7,0} &= 15^{48\cdot 0} \bmod 337 = 1 \\ r_{7,1} &= 15^{48\cdot 1} \bmod 337 = 79 \\ r_{7,2} &= 15^{48\cdot 2} \bmod 337 = 175 \\ r_{7,3} &= 15^{48\cdot 3} \bmod 337 = 8 \\ r_{7,4} &= 15^{48\cdot 4} \bmod 337 = 295 \\ r_{7,5} &= 15^{48\cdot 5} \bmod 337 = 52 \\ r_{7,6} &= 15^{48\cdot 6} \bmod 337 = 64 \end{aligned} \tag{2.3.3}$$

Konstruiramo tablicu $r_{p_i,j}$ ovako:

p_i	j						
	0	1	2	3	4	5	6
2	1	336					
3	1	208	128				
7	1	79	175	8	295	52	64

Tablica je izvodljiva ako su svi p_i mali.

3. Računamo diskretni logaritam $x = \log_{15} 131 \bmod 337$. Vidimo da je $a = 15$ i $b = 131$.

a) Pronalazimo diskretne logaritme $x \bmod p_i^{\alpha_i}$ koristeći formulu

$$x \bmod p_i^{\alpha_i} = x_0 + x_1 p_i + ... + x_{\alpha_{i-1}} p_i^{\alpha_{i-1}},\ 0 \le x_n < p_i - 1$$

i. Pronalazimo diskretni logaritam $x \bmod p_1^{\alpha_1}$, tj. $x \bmod 2^4$:

$$x \bmod 337 \Longleftrightarrow x \bmod 2^4 = x_0 + 2x_1 + 4x_2 + 8x_3.$$

A. Da bi pronašli x_0 računamo:

$$b^{(q-1)/p_1} \bmod q = 131^{336/2} \bmod 337 = 1 = r_{p_1,j} = r_{2,0},$$

pa je prema tome $x_0 = 0$.

B. Da bi pronašli x_1 prvo računamo $b_1 = ba^{-x_0} = b = 131$, onda računamo:

$$b_1^{(q-1)/p_1^2} \bmod q = 131^{180/4} \bmod 181 = 1 = r_{p_1,j} = r_{2,0},$$

pa je prema tome $x_1 = 0$.

C. Da bi pronašli x_2 prvo računamo $b_2 = ba^{-x_0-x_1p_1} = b = 131$:

$$b_2^{(q-1)/p_1^3} \bmod q = 131^{336/8} \bmod 337 = 336 = r_{p_1,j} = r_{2,1},$$

pa je prema tome $x_2 = 1$

D. Da bi pronašli x_3 prvo računamo računamo $b_3 = ba^{-x_0-x_1p_1-x_2{p_1}^2} = 321$

$$b_3^{(q-1)/p_1^4} \bmod q = 321^{336/16} = 336 = r_{p_1,j} = r_{2,1},$$

pa je prema tome $x_3 = 1$

Sad smo dobili:

$$x \bmod 2^4 = x_0 + 2x_1 + 4x_2 + 8x_3 = 4 + 8 = 12 \Longrightarrow x \bmod 16 = 12$$

ii. Pronalazimo diskretni logaritam $x \bmod p_2^{\alpha_2}$, tj. $x \bmod 3$:

$$x \bmod 337 \Longleftrightarrow x \bmod 3 = x_0$$

Da bi pronašli x_0 računamo:

$$b^{(q-1)/p_2} \bmod q = 131^{336/3} \bmod 337 = 128 = r_{p_2,j} = r_{3,2},$$

pa je prema tome $x_0 = 2$. Pa smo dobili sljedeće:

$$x \bmod 3 = x_0 = 2 \Longrightarrow x \bmod 3 = 3$$

iii. Pronalazimo diskretni logaritam $x \bmod p_3^{\alpha_3}$, tj. $x \bmod 7$:

$$x \bmod 337 \Longleftrightarrow x \bmod 7 = x_0$$

Da bi pronašli x_0 računamo :

$$b^{(q-1)/p_3} \bmod q = 131^{336/7} \bmod 336 = 52 = r_{p_3,j} = r_{7,5},$$

pa je prema tome $x_0 = 5$. Pa imamo:

$$x \bmod 7 = x_0 = 5 \Longrightarrow x \bmod 7 = 5$$

b) Pronalazimo x u

$$x \bmod 181$$

takav da:

$$\begin{cases} x \bmod 16 = 8 \\ x \bmod 3 = 2 \\ x \bmod 7 = 5 \end{cases}$$

Da bi pronašli x koristimo Kineski teorem o ostatcima kako bi riješili sljedeći sustav kongruencija:

$$\begin{cases} x \equiv 8 \pmod{16} \\ x \equiv 2 \pmod{3} \\ x \equiv 5 \pmod{7} \end{cases}$$

Rješenje za ovaj sustav kongruencija je $x = 236$. Pa je prema tome $x = 236$ rješenje početnog problema.

2.4 ρ - metoda za problem diskretnog logaritma

Vremenska složenost ove metode slična je baby-step giant-step algoritmu ali ova metoda zahtjeva zanemarivu količinu memorije. Pretpostavimo da želimo pronaći x takav da

$$\alpha^x \equiv \beta \pmod{n}$$

Pretpostavljamo da je r red elementa α u multiplikativnoj grupi $\mathbb{Z}_n^*$. U ρ-metodi za DLP grupu $G = \mathbb{Z}_n^*$ particioniramo u tri skupa G_1, G_2, i G_3 podjednake veličine. Definiramo niz $x_0, x_1, x_2, ...$ na sljedeći način:

$$\begin{cases} x_0 = 1 \\ x_{i+1} = f(x_i) = \begin{cases} \beta \cdot x_i & \text{ako } x_i \in G_1 \\ x_i^2 & \text{ako } x_i \in G_2 \\ \alpha \cdot x_i & \text{ako } x_i \in G_3 \end{cases} \end{cases} \tag{2.4.1}$$

za $i \geq 0$. Ovaj niz definira dva niza cijelih brojeva $\{a_i\}$ i $\{b_i\}$ ovako:

$$\begin{cases} a_0 = 0 \\ a_{i+1} = \begin{cases} a_i & \text{ako } x_i \in G_1 \\ 2a_i & \text{ako } x_i \in G_2 \\ a_i + 1 & \text{ako } x_i \in G_3 \end{cases} \end{cases} \tag{2.4.2}$$

i

$$\begin{cases} b_0 = 0 \\ b_{i+1} = \begin{cases} b_i + 1 & \text{ako } x_i \in G_1 \\ 2b_i & \text{ako } x_i \in G_2 \\ b_i & \text{ako } x_i \in G_3 \end{cases} \end{cases} \tag{2.4.3}$$

Tražimo x_i i x_{2i} takve da $x_i = x_{2i}$. Stoga,

$$\alpha^i \beta^i = \alpha^{2i} \beta^{2i}.$$

Prema tome,

$$\beta^{b_i-2b_i} = \alpha^{2a_i-a_i}.$$

Ako na obje strane djelujemo s logaritmom koji ima bazu α, dobivamo

$$x = \log_\alpha \beta \equiv \frac{2a_i - a_i}{b_i - 2b_i} \pmod{r},$$

uz uvjet da $b_i \not\equiv 2b_i \pmod{n}$. Pripadni ρ algoritam možemo opisati na sljedeći način.

Algoritam 2.4.1. *Ovaj algoritam traži x takav da*

$$\alpha^x \equiv \beta \pmod{n}.$$

Inicijaliziramo $x_0 = 1, a_0 = 0, b_0 = 0$
Za $i = 1, 2, 3, \ldots$ radi
  Koristeći 2.4.1, 2.4.2 i 2.4.3 računamo (x_i, a_i, b_i) i (x_{2i}, a_{2i}, b_{2i})
    Ako $x_i = x_{2i}$, radi
    Postavi $r \leftarrow b_i - b_{2i} \bmod n$
    Ako je $r = 0$ završi algoritam s neuspjehom
      Inače izračunaj $x \equiv r^{-1}(a_{2i} - a_i) \pmod{n}$
      Izlaz je x

Slijedi primjer upravo opisanog algoritma.

Primjer 2.4.1. *Tražimo x takav da vrijedi*

$$87^x \equiv 362 \pmod{703}.$$

Zadajemo G_1, G_2 i G_3 ovako

$$G_1 = \{x \in \mathbb{Z}_{703} : x \equiv 1 \pmod 3\}$$
$$G_2 = \{x \in \mathbb{Z}_{703} : x \equiv 0 \pmod 3\}$$
$$G_3 = \{x \in \mathbb{Z}_{703} : x \equiv 2 \pmod 3\}$$

Za $i = 1, 2, 3, \ldots$ računamo (x_i, a_i, b_i) i (x_{2i}, a_{2i}, b_{2i}) dok ne dobijemo $x_i = x_{2i}$:

i	(x_i, a_i, b_i)	(x_{2i}, a_{2i}, b_{2i})
1	(362, 0, 1)	(562, 1, 1)
2	(562, 1, 1)	(448, 1, 3)
3	(277, 1, 2)	(691, 2, 8)
4	(448, 1, 3)	(83, 2, 10)
5	(486, 1, 4)	(448, 4, 10)
6	(691, 2, 8)	(691, 8, 22)

Za $i = 6$ pronalazimo podudaranje:

$$x_6 = x_{12} = 691.$$

Budući da je red od 87 u $\mathbb{Z}_{703}^*$ jednak 36 imamo

$$x \equiv \frac{a_{12} - a_6}{b_6 - b_{12}} \equiv \frac{8-2}{22-8} \equiv 15 \pmod{36}.$$

Dakle vrijedi,

$$87^{15} \equiv 362 \pmod{703}.$$

2.5 Index Calculus algoritam

Adleman je 1979. godine predložio algoritam, zvan index calculus, za računanje diskretnog logaritma čija je vremenska složenost:

$$O\left(\exp\left(c\sqrt{\log n \log\log n}\right)\right)$$

Index calculus je, u stvari, širok spektar metoda, uključujući CFRAC, QS i NFS za problem faktorizacije cijelih brojeva. U onome što slijedi mi proučavamo varijantu Adelmanovog index calculus algoritma za DLP u $(\mathbb{Z}/p\mathbb{Z})^*$.

Algoritam 2.5.1. *Ovaj algoritam traži $k \in \mathbb{N}$ takav da*

$$k \equiv \log_\beta \alpha \pmod{p} \quad \text{ili} \quad \alpha \equiv \beta^k \pmod{p}.$$

1. Predizračunavanje

   a) (Odabir baze faktora) Biramo bazu faktora koja se sastoji od prvih m prostih brojeva,
   $$\Gamma = \{p_1, p_2, p_3, ..., p_m\},$$
   gdje je $p_m \leq B$, B je granica baze faktora.

   b) (Računamo $\beta^e \bmod p$) Nasumično biramo skup eksponenata $e \leq p-2$, računamo $\beta^e \bmod p$ i rezultat faktoriziramo kao produkt potencija prostih brojeva.

   c) (Glatkoća) Uzimamo samo one relacije $\beta^e \bmod p$ koje su glatke u odnosu na B.
   $$\beta^e \bmod p = \prod_{i=1}^{m} p_i^{e_i},\ e_i \geq 0$$
   Kad takva relacija postoji,
   $$e \equiv \sum_{j=1}^{m} e_j \log_\beta p_j \pmod{p-1}.$$

d) (Ponavljanje) Ponavljamo korak c) dok ne pronađemo bar m takvih e da bi pronašli m takvih relacija i riješili $\log_\beta p_j$ za $j = 1, 2, ..., m$.

2. Računamo $k \equiv \log_\beta \alpha \pmod p$

a) Za svaki e iz 1.c) određujemo vrijednost $\log_\beta p_j$ za $j = 1, 2, ..., m$ rješavajući m modularnih linearnih jednadžbi s nepoznanicama $\log_\beta p_j$

b) (Računamo $\alpha\beta^r \bmod p$) Nasumično odabiremo eksponent $r \leq p-2$ i računamo $\alpha\beta^r \bmod p$

c) (Faktoriziramo $\alpha\beta^r \bmod p$ nad Γ)

$$\alpha\beta^r \bmod p = \prod_{j=1}^{m} p_j^{r_j},\ r_j \geq 0.$$

Ako je prethodni korak bio neuspješan vraćamo se na b). Ako je bio uspješan, onda je

$$\log_\beta \alpha \equiv -r + \sum_{j=1}^{m} r_j \log_\beta p_j.$$

Sad dajemo primjer ovog algoritma.

Primjer 2.5.1. *Tražimo x takav da*

$$x \equiv \log_{18} 192 \pmod{3259} \quad \text{ili} \quad 192 \equiv 18^x \pmod{3259}.$$

1. Predizračunavanje

a) (Odabir baze faktora) Biramo bazu faktora koja se sastoji od prva 4 prosta broja,

$$\Gamma = \{2, 3, 5, 7\}$$

gdje je $p_4 \leq 7$, 7 je granica baze faktora.

b) (Računamo $18^e \bmod 3259$) Nasumično biramo skup eksponenata $e \leq 3257$, računamo $18^e \bmod 3259$ i rezultat faktoriziramo kao produkt potencija prostih brojeva:

$$\begin{aligned} 18^2 &\equiv 2^2 \cdot 3^4 \pmod{3259} \\ 18^{78} &\equiv 3^3 \cdot 7 \pmod{3259} \\ 18^{449} &\equiv 5^2 \cdot 7 \pmod{3259} \\ 18^{1469} &\equiv 2^4 \cdot 5^2 \cdot 7 \pmod{3259} \end{aligned}$$

c) (Glatkoća) Gornje četiri relacije su glatke u odnosu na $B = 7$. Prema tome,

$$2 \equiv 2\log_{18} 2 + 4\log_{18} 3 \pmod{3258}$$
$$78 \equiv 3\log_{18} 3 + \log_{18} 7 \pmod{3258}$$
$$449 \equiv 2\log_{18} 5 + \log_{18} 7 \pmod{3258}$$
$$1469 \equiv 4\log_{18} 2 + 2\log_{18} 5 + \log_{18} 7 \pmod{3258}$$

2. Računamo $k \equiv \log_\beta \alpha \pmod{p}$

a) Nakon što riješimo sustav modularnih jednadžbi dobijemo:

$$\log_{18} 2 \pmod{3258} = 255$$
$$\log_{18} 3 \pmod{3258} = 3131$$
$$\log_{18} 5 \pmod{3258} = 1624$$
$$\log_{18} 7 \pmod{3258} = 459$$

b) (Računamo $192 \cdot 18^r \bmod p$) Nasumično biramo eksponent $r = 553 \leq 3258$ i računamo $192 \cdot 18^{553} \bmod 3359$.

c) (Faktoriziramo $192 \cdot 18^{553} \bmod 3259$ nad Γ)

$$192 \cdot 18^{553} \equiv 3 \cdot 5 \cdot 7 \pmod{3259}$$

Pa je onda,

$$\log_{18} 192 \equiv -553 + \log_{18} 3 + \log_{18} 5 + \log_{18} 7 = 1403$$

To znači da je,

$$18^{1403} \equiv 192 \pmod{3259}.$$

2.6 Gordonov NFS

Više od 10 godina otkad je izumljena, Adelmanova metoda i njene varijante bile su najbrži algoritmi za računanje diskretnih logaritama. No, to se promijenilo kad je Gordon 1993. predložio algoritam za računanje diskretnih logaritama nad $GF(p)$. Gordonov algoritam, baziran na situ polja brojeva (number field sieve NFS) za faktorizaciju cijelih brojeva, ima očekivano vrijeme trajanja

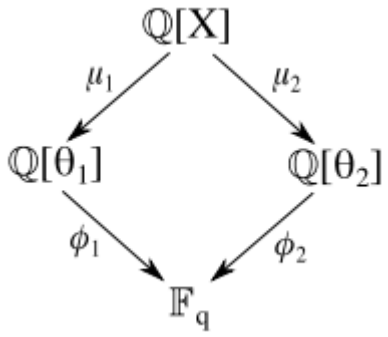
$$O\left(\exp\left(c(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}\right)\right)$$

Prvo dajemo opis algoritma, pa ćemo zatim na primjeru vidjeti kako algoritam računa problem diskretnog logaritma.

Algoritam 2.6.1. *Ovaj algoritam računa diskretni logaritam x tako da* $t^x \equiv u \pmod p$*, gdje su* t, u, p *ulazni podatci,* t *generator grupe* $\mathbb{F}_p^*$ *i* $p = q^n$*, gdje je* q *prost broj i* $n \in \mathbb{N}$*:*

1. Prvo pronalazimo dva ireducibilna polinoma f_1 if_2 takva da postoji $m \in \mathbb{Z}$ sa svojstvom:
$$f_i(m) \equiv 0 \pmod p$$

2. S $\theta_i \in \mathbb{C}$ označavamo korijen polinoma f_i i onda definiramo polja $K_i = \mathbb{Q}[\theta_i]$. Ova polja, tj. njihovi prstenovi cijelih brojeva O_{K_i} se koriste da se pronađu relacije između prostih ideala. Kad su njihovi prstenovi cijelih brojeva Dedekindove domene tada je osigurana jedinstvena faktorizacija ideala. Odnosi između polja definirani su homomorfizmima kao što vidimo na dijagramu 2.6.1 kao što je pokazano u [3]. Dakle imamo sljedeće homomorfizme
$$\mu_1 : X \to \theta_1,\ \mu_2 : X \to \theta_2,\ \phi_1 : \theta_1 \to m \ \text{ i } \ \phi_2 : \theta_2 \to m$$
Tražimo relacije između polja i kad ih pronađemo dovoljno sastavljamo matricu od tih relacija. Nakon toga koristimo preslikavanje ϕ_i da prebacimo rezultate u $\mathbb{F}_p$.



Slika 2.6.1: Odnosi polja.

3. Da pronađemo relacije prvo moramo odabrati glatkoću B_i, prosijati granice T_i i baze faktora $S_1 = \{P_1, P_2, ..., P_k\}$ i $S_2 = \{Q_1, Q_2, ..., Q_k\}$ sastavljene od prostih elemenata čija je norma manja od B_i. Onda tražimo parove (a, b) gdje su $a, b \in \mathbb{Z}$, $|a|, |b| \leq T_i$ i $\gcd(a, b) = 1$, takvi da se ideali $(a + b\theta_i)$ mogu faktorizirati elementima iz S_i. Da bi ovo postigli biramo nasumični par (a, b) računamo normu $N_{K\setminus Q}(a + b\theta_i)$ i testiramo je li norma B_i-glatka. Ako su svi ovi uvjeti zadovoljeni onda imamo:
$$(a + b\theta_1) = \prod_{P_j \in S_1} P_j^{e_j} \text{ i}$$

$$(a + b\theta_2) = \prod_{Q_j \in S_2} Q_j^{e_j}.$$

Ovo nam daje jednu potrebnu relaciju. Postupak ponavljamo dok ne pronađemo dovoljan broj relacija.

4. Kad imamo dovoljan broj relacija, onda kreiramo matricu, rješavamo je modulo $p-1$ i dobivamo diskretne logaritme elemenata iz baze faktora. Na kraju kad računamo diskretne logaritme elemenata konačno možemo primjeniti preslikavanja ϕ_i.

5. Jednom kad imamo diskretne logaritme faktora baze modulo $p-1$, koristimo ih da izračunamo traženi diskretni logaritam. Biramo nasumični $s \in \mathbb{Z}$ i računamo $t^s \cdot u \pmod{p}$. Biramo novi s sve dok ne dobijemo da se ovaj rezultat može faktorizirati tako da svi faktori pripadaju našoj bazi faktora. Kad to pronađemo imamo:

$$t^s \cdot u \equiv \prod_{i=1}^{r} c_i^{d_i} \pmod{p},$$

Djelujemo s logaritmom:

$$\log_t(t^s \cdot u) = \log_t(t^s \cdot t^x) \equiv \sum_{i=1}^{r} d_i \log_t(c_i) \pmod{\text{p}},$$

pa imamo

$$s + x \equiv \sum_{i=1}^{r} d_i \log_t(c_i) \pmod{q}.$$

Budući da već znamo sve diskretne logaritme $\log_t(c_i)$, iz ovoga lako računamo traženi x.

Slijedi primjer.

Primjer 2.6.1. *Pronalazimo x takav da* $2^x \equiv 9 \pmod{11}$.

1. Primjetimo da je 2 generator grupe $\mathbb{Z}_{11}^*$. Biramo dva ireducibilna polinoma, npr.

$$f_1 = x^2 + 7 \quad \text{i} \quad f_2 = x - 2.$$

Oba polinoma imaju isti korijen modulo 11, $m = 2$. Prvi polinom f_1 određuje polje $K_1 = \mathbb{Q}[\sqrt{7}i]$. Drugi polinom f_2 daje $K_2 = \mathbb{Q}$. Problem je što njegov prsten cijelih brojeva $\mathbb{Z}[\sqrt{7}i]$ nije integralno zatvoren. Prsten R je integralno zatvoren ako za svaki korijen x normiranog polinoma jedne varijable s koeficijentima u R koji je sadržan u kvocijentnom polju od R vrijedi da je x u R. Sada vidimo da $\mathbb{Z}[\sqrt{7}i]$

nije integralno zatvoren jer je $\frac{1+\sqrt{7}i}{2}$ koji je korijen normiranog polinoma $X^2 - X + 2$ sadržan u kvocijentnom polju $\mathbb{Q}(\sqrt{7}i)$ od $\mathbb{Z}[\sqrt{7}i]$, ali nije sadržan u $\mathbb{Z}[\sqrt{7}i]$.

Integralna zatvorenost nam je potrebna jer ako prsten cijelih brojeva nije integralno zatvoren, onda sigurno nije Dedekindova domena. Kad je prsten Dedekindova domena, imamo osiguranu jedinstvenu faktorizaciju ideala, te jedinstvenu faktorizaciju elemenata, barem lokalno. Zato trebamo pronaći proširenje koje je integralno zatvoreno. Npr., možemo uzeti $\mathbb{Z}[\frac{1+\sqrt{7}i}{2}]$.

2. Nadalje, homomorfizmi su definirani kao:

$$\phi_1 : a + b\sqrt{7}i \to a + 2b \pmod{11} \quad i$$
$$\phi_2 : a \to a \pmod{11},$$

pa imamo $\theta_1 = \sqrt{7}i$ i $\theta_2 = 2$.

3. Sljedeće što radimo je biranje baze faktorizacije za K_1 kao skup $\{-1, \sqrt{7}i, \frac{\sqrt{7}i+1}{2}, \frac{\sqrt{7}i-1}{2}\}$ sastavljen od generatora prostih ideala. Za K_2 skup $\{-1, 2, 3, 5\}$ sastavljen od prirodnih brojeva i -1. U fazi prosijavanja tražimo parove (a, b) gdje su $a, b \in \mathbb{Z}$ takvi da je $\gcd(a, b) = 1$ i ideal $(a+b\theta_i)$ faktorizira proste ideale s generatorima sadržanim u bazi.

Jedan takav par je $(-1, 1)$ jer se $(\theta_1 - 1)$ faktorizira kao $-1 \cdot (\frac{\sqrt{7}i+1}{2})^1 \cdot (\frac{\sqrt{7}i-1}{2})^2$ u K_1 i $(\theta_2 - 1)$ se faktorizira kao $-1 \cdot 2 \cdot 5$ u K_2.

Još takvih parova možemo naći u sljedećoj tablici gdje retci odgovaraju relacijama. Elementi koji predstavljaju stupce su generatori prostih ideala koje $(a + b\theta_i)$ faktorizira. Brojevi u tablici određuju eksponente faktorizacije u K_1 i K_2.

	-1	$\sqrt{7}i$	$\frac{\sqrt{7}i+1}{2}$	$\frac{\sqrt{7}i+1}{2}$	-1	2	3	5
$\theta_i + 1$	1	0	2	1	0	0	1	0
$\theta_i - 1$	1	0	1	2	0	0	0	0
$\theta_i + 3$	0	0	1	3	0	0	0	1
$\theta_i - 3$	1	0	3	1	1	0	0	0
$\theta_i + 5$	0	0	4	1	0	0	0	0
$\theta_i - 5$	0	0	1	4	1	0	1	0
$\theta_i + 7$	0	1	1	2	0	0	2	0
2	0	0	0	0	0	1	0	0

Da bi dobili nehomogenu matricu dodajemo još jednu relaciju $x^1 \equiv 2 \pmod{10}$, pa onda imamo jedinstveno rješenje sustava.

4. Dakle, matricu sastavljamo od gornjih relacija.

$$\begin{bmatrix} 1 & 0 & 2 & 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 & 0 & -1 \\ 1 & 0 & 3 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

s vektorom $v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top$ na desnoj strani. Kao rješenje modulo 10 dobivamo $\begin{bmatrix} -5 & 1 & -3 & -1 & -5 & 1 & -2 & 4 \end{bmatrix}^\top$, pa je

$$\log_2(\phi_1(-1)) = \log_2(10) = 5 \equiv -5 \pmod{10}$$
$$\log_2(\phi_1(\sqrt{7}i)) = \log_2(2) \equiv 1 \pmod{10}$$
$$\log_2(\phi_1(\tfrac{\sqrt{7}i+1}{2})) = \log_2(\tfrac{1}{2}+1) = \log_2(7) \equiv -3 \pmod{10}$$
$$\log_2(\phi_1(\tfrac{\sqrt{7}i-1}{2})) = \log_2(-\tfrac{1}{2}+1) = \log_2(-5) = \log_2(6) = 9 \equiv -1 \pmod{10}$$
$$\log_2(-1) = \log_2(10) = 5 \equiv -5 \pmod{10}$$
$$\log_2(2) \equiv 1 \pmod{10}$$
$$\log_2(3) = 8 \equiv -2 \pmod{10}$$
$$\log_2(4) \equiv 4 \pmod{10}$$

Sad imamo izračunate logaritme modulo 10:
$\log_2(-1) = 5, \quad \log_2(3) = 8, \quad \log_2(6) = 9, \quad \log_2(7) = 7, \quad \log_2(2) = 1$
$\log_2(5) = 4, \quad \log_2(10) = 5.$

5. Koristeći istu metodu kao u Index-Calculus algoritmu biramo neku potenciju broja 2 i pomnožimo je s 9, faktoriziramo rezultat i potvrdimo da su svi faktori već izračunati diskretni logaritmi.
Ako je eksponent 1 onda imamo $9 \cdot 2^1 \equiv 7 \pmod{11}$. Znamo da je $\log_2 7 \equiv 7 \pmod{10}$,pa je onda

$$\log_2(9 \cdot 2^1) \pmod{10} = \log_2 2^{x+1} \pmod{10} = \log_2 7 \pmod{10}.$$

Što povlači da je

$$x + 1 \equiv 7 \pmod{10}$$

$$x \equiv 6 \pmod{10}.$$

Dakle,

$$2^6 \equiv 9 \pmod{10}.$$

Poglavlje 3

Kvantni napadi na DLP

3.1 Odnosi između DLP-a i kriptografije bazirane na DLP-u

Kao što smo vidjeli zasad ne postoji klasični napad na DLP koji bi bio primjenjiv u praksi. Ovo povlači da je kriptografija bazirana na DLP-u sigurna i neprobojna u polinomnom vremenu. Prema tome, kad bi uspjeli riješiti DLP problem to bi onda bilo ekvivalentno razbijanju kriptografije bazirane na DLP-u.

Iako zasad nije poznat efikasan klasični algoritam za razbijanje kriptografije bazirane na DLP-u, Shor je pokazao da se DLP može riješiti u $\mathcal{BQP}$ vremena, gdje je $\mathcal{BQP}$ klasa problema koja je efikasno rješiva na kvantnom Turingovom stroju.

Prema tome, kriptografija bazirana na DLP-u se može razbiti u polinomnom vremenu na kvantnom računalu.

3.2 Razine težine za različite DLP

Postoje tri osnovna tipa DLP-a gledajući po razini težine da ih se riješi, sada ih navodimo.

1. DLP u aditivnoj grupi $G = \mathbb{Z}_n$ je lak za riješiti. Promotrimo aditivnu cikličku grupu $\mathbb{Z}_{102}$ reda 102. Rješavamo

$$n \equiv \log_5 16 \pmod{102}$$

$$5n \equiv 16 \pmod{102}$$

Ovaj tip DLP-a se može riješiti u polinomnom vremenu koristeći Euklidov algoritam.

$$\begin{aligned} n &\equiv 5^{-1} \cdot 16 \pmod{102} \\ &\equiv 41 \cdot 16 \pmod{102} \\ &= 44. \end{aligned}$$

2. DLP u multiplikativnoj grupi $G = \mathbb{Z}_p^*$ je težak za riješiti. Promotrimo mulitplikativnu grupu $G = \mathbb{Z}_{107}^*$ reda 106. Rješavamo

$$n \equiv \log_5 22 \pmod{107}$$

$$n^5 \equiv 22 \pmod{107}$$

Ovaj tip DLP-a je težak i zasad ne postoji polinomni algoritam koji ga rješava. Naravno, u ovom primjeru su mali brojevi pa se lako riješi koristeći "grubu silu".

$$\log_5 22 \equiv 5 \pmod{107}$$

3. DLP u grupi eliptičkih krivulja $G = E(\mathbb{F}_p)$ je također težak za izračunati. Pogledajmo primjer eliptičke krivuljen ad konačnim poljem:

$$E/\mathbb{F}_7 : y^2 \equiv x^3 + x + 1 \pmod{7}$$

gdje je $\{P(2,2), Q(0,6)\} \in E(\mathbb{F}_7)$.Točka P je reda 5 u grupi $E(\mathbb{F}_7)$, pa želimo pronaći $r \equiv \log_P Q \pmod{5}$.

Ovaj tip DLP-a je težak i zasad ne postoji polinomni algoritam koji ga rješava. Naravno, u ovom primjeru su opet mali brojevi pa se također lako riješi koristeći "grubu silu".

$$\log_P Q \equiv 3 \pmod{5}$$

Budući da se za posljednja dva tipa DLP-a nikad nije pronašao efikasan algoritam došla je potreba za razvijanjem kvantnih algoritama. Znači da su upravo ovakvi problemi bili motivacija za kvantne algoritme i kvantna računala.

3.3 Ideja kvantnog napada na DLP

Prisjetimo se da prilikom rješavanja DLP-a želimo pronaći k u

$$g^r \equiv x \pmod{p},$$

gdje je g generator u multiplikativnoj grupi $\mathbb{Z}_p^*$. Pretpostavljamo da je red od g u $\mathbb{Z}_p^*$ jednak k, tj.

$$g^k \equiv 1 \pmod{p}.$$

Primjetimo da u kvantnom algoritmu za faktoriziranje pokušavamo pronaći

$$g^r \equiv 1 \pmod{p},$$

gdje je r red od g u $\mathbb{P}_{p-1}$. U kvantnom algoritmu za diskretni logaritam pokušavamo pronaći

$$g^r \equiv x \pmod{p},$$

gdje je r diskretni logaritam baze g u $\mathbb{P}_{p-1}$, tj.

$$r \equiv \log_g x \pmod{p-1},$$

Definicije od r u dva kvantna algoritma su različite. No, budući da

$$g^r \equiv x \pmod{p},$$

možemo definirati funkciju dviju varijabli,

$$f(a,b) = g^a x^{-b} \equiv 1 \pmod{p}$$

takvu da

$$a - br \equiv k \pmod{p}.$$

Ovo možemo zbog

$$\begin{aligned} g^a x^{-b} &\equiv g^a (g^r)^{-b} \\ &\equiv g^a g^{-br} \\ &\equiv g^{a-br} \\ &\equiv g^k \pmod{p}. \end{aligned}$$

Prema tome, u kvantnom algoritmu za diskretni logaritam moramo dobiti r u

$$r \equiv (a-k)b^{-1} \pmod{p-1}.$$

Ovo je, u stvari, samo inverzni problem. Shor je pokazao da kvantni algoritmi mogu pronaći r u polinomnom vremenu. Pronalazak reda grupe u polinomnom vremenu omogućila je kvantna Fourierova transformacija koja je konstruirana iz brze Fourierove transformacije. Ona je njezin ekvivalent, ali na kvantnom sklopu. Dajemo njenu definiciju.

Definicija 3.3.1. *Neka je* $\{|0\rangle, |1\rangle, ..., |N-1\rangle\}$ *ortonormirana baza kvantnog sustava i neka je* $|\phi\rangle = \sum_{j=0}^{N-1} |j\rangle$ *kvantno stanje. Onda je kvantna Fourierova transformacija* A_N *preslikavanje definirano s:*

$$|\phi\rangle = \sum_{j=0}^{N-1} |j\rangle \rightarrow \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(\frac{2\pi i}{n}\right)^{-jk} |k\rangle.$$

Bazno se stanje transformira ovako:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(\frac{2\pi i}{n}\right)^{-jk} |k\rangle.$$

Kvantnu Fourierovu transformaciju možemo konstruirati u polinomnom vremenu samo kada je N, iz definicije, gladak. Vidjet ćemo kako smo u općenitom slučaju postigli da je N gladak prilikom korištenja Fourierove transformacije A_N. U lakšem slučaju o tome nismo morali razmišljati jer je uvjet bio da $p-1$ bude gladak.

Naravno, ako je $p-1$ gladak, tj. $p-1$ ima male proste faktore, onda se DLP u $\mathbb{Z}_p^*$ može riješiti u polinomnom vremenu pomoću Pohlig-Hellmanovog algoritma. Za općeniti p još uvijek nema klasičnog polinomnog algoritma. U onome što slijedi, prvo raspravljamo o lakšem slučaju, pa zatim o općenitom slučaju kvantnog napada na DLP.

3.4 Lakši slučaj kvantnog napada

Lakši slučaj kvantnog napada na DLP je, u stvari, kvantna verzija Pohlig-Hellmanove metode za DLP. Prisjetimo se da tražimo diskretni logaritam r u

$$g^r \equiv x \pmod{p},$$

gdje je g generator multiplikativne grupe $\mathbb{Z}^*$ i p je prost broj gdje je $p-1$ gladak. Pohlig-Hellmanov algoritam ovaj problem rješava u polinomnom vremenu na klasičnom računalu. Nema neke prednosti koristiti kvantno računalo za rješavanje ovog lakšeg slučaja ali to radimo zato da pokažemo da kvantna računala mogu riješiti ovaj problem jednako dobro kao i klasična računala.

Algoritam 3.4.1. *Za dane* $g, x \in \mathbb{N}$ *i* p *prost broj ovaj algoritam pronalazi cijeli broj* r *takav da* $g^r \equiv x \pmod{p}$ *ako* r *postoji. Koristi tri kvantna registra.*

1. Započinjemo s inicijalnim stanjem

$$|\Psi_0\rangle = |0\rangle|0\rangle$$

izabiremo brojeve a i b mod $p-1$ uniformno i radimo Fourierovu transformaciju modulo $p-1$, označenu s A_{p-1}. Pa je stanje stroja nakon ovog koraka sljedeće

$$\begin{aligned}|\Psi_1\rangle &= \frac{1}{\sqrt{p-1}}\sum_{a=0}^{p-2}|a\rangle \cdot \frac{1}{\sqrt{p-1}}\sum_{b=0}^{p-2}|b\rangle \\ &= \frac{1}{p-1}\sum_{a=0}^{p-2}\sum_{b=0}^{p-2}|a,b\rangle.\end{aligned}$$

2. Računamo $g^a x^{-b} \pmod p$, vrijednosti od a i b se moraju čuvati na traci. Ovo dovodi kvantno računalo u stanje $|\Psi_2\rangle$:

$$|\Psi_2\rangle = \frac{1}{p-1}\sum_{a=0}^{p-2}\sum_{b=0}^{p-2}|a,b,g^a x^{-b} \pmod p\rangle.$$

3. Koristimo Fourierovu transformaciju A_{p-1} da bi preslikali $|a\rangle \to |c\rangle$ s vjerojatnosnom amplitudom

$$\sqrt{\frac{1}{p-1}}\exp\left(\frac{2\pi i a c}{p-1}\right)$$

i $|b\rangle \to |d\rangle$ s vjerojatnosnom amplitudom

$$\sqrt{\frac{1}{p-1}}\exp\left(\frac{2\pi i b d}{p-1}\right).$$

Prema tome, stanje $|a,b\rangle$ će se promijeniti u stanje

$$\frac{1}{p-1}\sum_{c=0}^{p-2}\sum_{d=0}^{p-2}\exp\left(\frac{2\pi i}{p-1}(ac+bd)\right)|c,d\rangle.$$

Ovo ostavlja stroj u stanju $|\Psi_3\rangle$:

$$|\Psi_3\rangle = \frac{1}{(p-1)^2}\sum_{a,b,c,d=0}^{p-2}\exp\left(\frac{2\pi i}{p-1}(ac+bd)\right)|c,d,g^a x^{-b} \pmod p\rangle.$$

4. Promatramo stanje kvantnog računala i vadimo potrebne informacije. Vjerojatnost stanja koje promatramo $|c,d,g^k \pmod p\rangle$ je

$$\text{Prob}(c,d,g^k) = \left|\frac{1}{(p-1)^2}\sum_{\substack{a,b \\ a-rb\equiv k \pmod{p-1}}}\exp\left(\frac{2\pi i}{p-1}(ac+bd)\right)\right|^2$$

gdje je sumiramo po svim (a, b) takvim da

$$a - rb \equiv k \pmod{p-1}.$$

5. Radimo supstituciju

$$a \equiv k + rb \pmod{p-1}$$

dobivamo

$$\text{Prob}(c, d, g^k) = \left| \frac{1}{(p-1)^2} \sum_{b}^{p-2} \exp\left(\frac{2\pi i}{p-1}(kc + b(d + rc))\right)\right|^2.$$

Primjetimo da ako $d + rc \not\equiv 0 \pmod{p-1}$, onda je vjerojatnost 0. Prema tome, vjerojatnost je različita od 0 ako i samo ako $d + rc \equiv 0 \pmod{p-1}$, tj.

$$r \equiv -dc^{-1} \pmod{p-1}.$$

6. Kako je naše izračunavanje proizvelo a, nasumični c i odgovarajući $d \equiv -rc \pmod{p-1}$ i ako je $\gcd(c, p-1) = 1$, tada možemo pronaći r pronalazeći multiplikativni inverz od c koristeći Euklidov algoritam. No što je još važnije, vjerojatnost da je $\gcd(c, p-1) = 1$ je

$$\frac{\phi(p-1)}{p-1} > \frac{1}{\log p},$$

u stvari,

$$\liminf \frac{\phi(p-1)}{p-1} \approx \frac{e^{-\gamma}}{\log\log p}.$$

To znači da samo trebamo broj eksperimenata koji je polinoman u $\log p$ da bi dobili r s velikom vjerojatnošću.

3.5 Općeniti slučaj kvantnog napada

Upravo smo pokazali da kvantna računala mogu rješiti poseban slučaj DLP-a jednako dobro kao i klasična računala. No, kvantna računala mogu riješiti i općeniti slučaj DLP-a efikasno u polinomnom vremenu za razliku od klasičnog računala.

Prisjetimo se da se posebni slučaj DLP-a oslanja na činjenicu da je $p - 1$ gladak. U općenitom slučaju, uklanjamo ovu restrikciju odabiranjem nasumično glatkog q takvog da $p \leq q \leq 2p$. Takav q se može pronaći u polinomnom vremenu tako da nijedna prosta potencija, veća od $c \log q$, ne dijeli q, za neku konstantu c neovisnu o p.

Algoritam 3.5.1. *Neka je g generator od $\mathbb{Z}_p^*, x \in \mathbb{Z}_p$. Ovaj algoritam pronalazi cijeli broj r takav da $g^r \equiv x \pmod{p}$.*

1. Izabrati nasumični glatki broj q takav da $p \leq q \leq 2p$. Primjetimo da ne zahtjevamo da je $p-1$ gladak.

2. Isto kao u posebnom slučaju, odabiremo brojeve a i b modulo $p-1$ uniformno i radimo Fourierovu transformaciju modulo $p-1$. Ovo ostavlja kvantno računalo u stanju $|\Psi_1\rangle$:
$$|\Psi_1\rangle = \frac{1}{p-1}\sum_{a=0}^{p-2}\sum_{b=0}^{p-2}|a, b \pmod{p}\rangle.$$

3. Računamo $g^a x^{-b}$ mod p i čuvamo a i b na traci. Ovo ostavlja kvantno računalo u stanju $|\Psi_2\rangle$:
$$|\Psi_2\rangle = \frac{1}{p-1}\sum_{a=0}^{p-2}\sum_{b=0}^{p-2}|a, b, g^a x^{-b} \pmod{p}\rangle.$$

4. Koristimo Fourierovu transformaciju A_q da bi preslikali $|a\rangle \to |c\rangle$ s vjerojatnosnom amplitudom
$$\frac{1}{\sqrt{q}}\exp\left(\frac{2\pi i a c}{q}\right)$$
i $|b\rangle \to |d\rangle$ s vjerojatnosnom amplitudom
$$\frac{1}{\sqrt{q}}\exp\left(\frac{2\pi i b d}{q}\right).$$
Prema tome, stanje $|a, b\rangle$ će se promijeniti u stanje
$$\frac{1}{p-1}\sum_{c=0}^{p-2}\sum_{d=0}^{p-2}\exp\left(\frac{2\pi i}{q}(ac+bd)\right)|c, d\rangle.$$
Ovo ostavlja stroj u stanju $|\Psi_3\rangle$:
$$|\Psi_3\rangle = \frac{1}{(p-1)q}\sum_{a,b=0}^{p-2}\sum_{c,d=0}^{q-1}\exp\left(\frac{2\pi i}{q}(ac+bd)\right)|c, d, g^a x^{-b} \pmod{p}\rangle.$$

5. Promotrimo stanje kvantnog računala i izvlačimo potrebne informacije. Vjerojatnost stanja koje promatramo je gotovo ista kao u specijalnom slučaju:
$$\text{Prob}(c, d, g^k) = \left|\frac{1}{(p-1)q}\sum_{\substack{a,b \\ a-rb\equiv k \pmod{p-1}}}\exp\left(\frac{2\pi i}{q}(ac+bd)\right)\right|^2.$$

6. Koristimo relaciju

$$a \equiv k + br - (p-1)\left\lfloor \frac{br+k}{p-1} \right\rfloor$$

i radimo supstituciju da bi dobili amplitudu:

$$\frac{1}{(p-1)q} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{q}\left(brc + kc + bd + -c(p-1)\left\lfloor \frac{br+k}{p-1} \right\rfloor\right)\right),$$

tako da suma iz 5. postaje:

$$\text{Prob}(c, d, g^k) = \left| \frac{1}{(p-1)q} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{q}\left(brc + kc + bd + -c(p-1)\left\lfloor \frac{br+k}{p-1} \right\rfloor\right)\right)\right|^2.$$

7. Može se pokazati da se neki parovi vrijednosti c i d pojavljuju s visokom vjerojatnošću i zadovoljavaju ogradu:

$$\left| rc + d - \frac{r}{p-1}(c(p-1) \bmod q \right| \leq \frac{1}{2}.$$

Jednom kad se takav par c, d može pronaći, r se može izračunati, kako je r jedina nepoznanica u

$$\left| d + \frac{r(c(p-1) - c(p-1) \bmod q)}{p-1} \right| \leq \frac{1}{2}.$$

Primjetimo također da

$$q \quad | \quad (c(p-1) - c(p-1) \bmod q),$$

onda kad podijelimo obje strane s q, dobijemo

$$\left| \frac{d}{q} - \frac{rl}{p-1} \right| \leq \frac{1}{2q}.$$

Da se pronađe r, samo zaokružimo $\frac{d}{q}$ na najbliži višekratnik od $\frac{l}{p-1}$,označen s $\frac{m}{p-1}$ i onda računamo r iz

$$\frac{m}{p-1} = \frac{rl}{p-1}.$$

Pa je,

$$r = \frac{m}{l}.$$

Bibliografija

[1] A. Godušová. *Number Field Sieve for Discrete Logarithm*. Charles University in Prague - Faculty of Mathematics and Physics, 2015

[2] A. Joux and R. Lercier. *Number field sieve for the DLP. in: H.C.A van Tilborg, S. Jajodia* (Eds.): *Encyclopedia of Cryptography and security*. Springer, 2011

[3] F. X. Lin *Shor's Algorithm and the Quantum Fourier Transform*. McGill University, 2014

[4] M. Vuković. *Složenost algoritama*. Zagreb, PMF-Matematički odsjek, 2011

[5] S. Y. Yan. *Quantum Attacks on Public-Key Cryptosystems*. Springer, 2013

[6] S. Y. Yan. *Quantum Computational Number Theory*. Springer, 2015

Sažetak

Ovaj rad proučava problem diskretnog logaritma (DLP) koji je zanimljiv jer je težak u općenitom slučaju. U prvom poglavlju uvode se pojmovi prostorne i vremenske složenosti na klasičnom računalu i navode se odnosi između osnovnih klasa složenosti. Zatim smo pričali o kvantnim računalima koja su zasnovana na karakteristikama kvantnih fenomena, poput kvantnih interferencija i kvantnih sprezanja. Slično kao za klasična računala definirali smo klase vremenske i prostorne složenosti za kvantna računala.

U drugom poglavlju prvo je dana definicija problema diskretnog logaritma. U radu se nakon toga obrađuju klasični napadi na DLP. Klasične napade možemo podijeliti u tri skupine. Algoritmi koji rade na proizvoljnim grupama, dakle ne iskorištavaju nikakva posebna svojstva algoritama. To su Shanksova baby-step giant-step metoda i Pollardova ρ-metoda. Onda imamo algoritme koji rade dobro u konačnim grupama za koje je red grupe gladak. U ovoj kategoriji se nalazio Silver-Pohlig-Hellman algoritam. Konačno, postoje algoritmi koji iskorištavaju metode za reprezentiranje grupe elemenata iz relativno malog skupa. U ovu grupu spadaju Adelmanov index calculus algoritam i Gordonov NFS algoritam.

No ipak, nijedan od ovih algoritama nije primjenjiv u praksi što povlači da je kriptografija bazirana na DLP-u sigurna. Kad bi uspjeli efikasno riješiti DLP onda bi i razbili kriptografiju baziranu na DLP-u.

Shor je pokazao da se DLP može riješiti u $\mathcal{BQP}$ vremenskoj složenosti, gdje je $\mathcal{BQP}$ klasa problema koji su efikasno rješivi na kvantnom Turingovom stroju. Prema tome, kvantna računala mogu razbiti kriptografiju baziranu na DLP-u u polinomnom vremenu. S ovim se bavi treće poglavlje. Dajemo ideju kvantnog napada na DLP. Zatim radimo lakši slučaj kvantnog napada koji je, u stvari, kvantna verzija Pohlig-Hellmanovog algoritma. Iako nema stvarne potrebe za kvantnim algoritmom za lakši slučaj DLP-a, to smo napravili da pokažemo da kvantna računala jednako dobro rade za lakši slučaj kao i klasična. Na kraju pokazujemo da kvantna računala mogu riješiti i općeniti slučaj DLP- u polinomnom vremenu.

Summary

This thesis studies discrete logarithm problem (DLP) which is interesting because it is hard in general case. In the first chapter we introduce concepts of time and space complexity on a classical computer and we discuss relations between complexity classes. Than we discuss about quantum computers which rely on quantum phenomena, such as quantum interference and quantum entanglement. Than we define time and space complexity classes, in a similar way as for conventional computers.

In the second chapter first we introduce definition of the discrete logarithm problem. Next, in the thesis we describe classical attacks on the DLP. There are three different categories of classical attacks. Algorithms that work for arbitrary groups, that is, those that do not exploit any specific properties of groups. Algorithms in this category are Shanks baby-step giant-step method and ρ-method. Then there are algorithms that work well in finite groups that have smooth order. Silver-Pohlig-Hellman is in this category. At last, we have algorithms that exploit methods for representing group elements as products of elements form relatively small set. In this category are Adleman's index calculus algorithm and Gordon's NFS algortihm.

However, none of these methods is not effective in practice so this would imply that the DLP-based cryptography is secure. Solving DLP is equivalent to breaking DLP-based cryptography.

Shor showed that DLP can be solved in $\mathcal{BQP}$, where $\mathcal{BQP}$ is the class of problem that is efficiently solvable in polynomial time on a quantum Turing machine. Hence, all DlP-based cryptography systems can be broken in polynomial time on a quantum computer. This is discussed in third chapter. We give idea for quantum attack on DLP. Then we do easy case of DLP, we did that to show that quantum computers compute easy case equally good as conventional computers. At last, we show that quantum computers can solve general case of DLP in polynomial time.

Životopis

Rođen sam 24. listopada 1993. u Splitu. Završio sam Osnovnu školu ”Stjepan Radić” u Imotskom, a zatim Matematičku gimnaziju u Srednjoj školi ”Dr. Mate Ujević” u Imotskom. 2012. godine sam upisao Preddiplomski sveučilišnni studij Matematika na Matematičkom odsjeku PMF-a u Zagrebu. Preddiplomski sveučilišni studij sam završio 2016. te stekao titulu univ.bacc.math. U listopadu 2016. godine upisujem Diplomski sveučilišni studij Računarstvo i matematika na PMF-u u Zagrebu, kojeg trenutno završavam.