UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Jovana Dubak

# Problem ranca

Specijalistički rad

Podgorica, 2016.

UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

# Problem ranca

Specijalistički rad

Kriptografija

Mentor: Prof. dr Vladimir Božović

Jovana Dubak

Matematika i računarske nauke

Podgorica, septembar 2016.

## Apstrakt

Zadatak ovog rada je bio definisanje problema ranca i problema zbira podskupa, kao njegovog posebnog slučaja. Predstavljen je značaj rešetke u rješavanju navedenog problema, a samim tim i opisan način za nalaženje najkraćeg i najbližeg vektora rešetke. Algoritmi za nalaženje kratkih vektora u rešetki se nazivaju algoritmi redukcije rešetke, a jedan od najpoznatijih je LLL algoritam. Osim toga, navedena je primjena LLL algoritma u problemu nalaženja najkraćeg, odnosno najbližeg vektora. Takođe, navedeni su kripto-sistemi zasnovani na složenosti rešetke i dat poseban osvrt na primjenu LLL algoritma u kriptoanalizi.

## Abstract

A task of this paper was to define knapsack problem and subset sum problem as a special case of knapsack problem. The importance of lattice is presented in solving mentioned problem and therefore described a way for finding shortest and closest vector in lattice. Algorithms that find short vectors in lattices are called lattice reduction algorithms and the most famous of these is LLL algorithm. Further, application of LLL algorithm is specified in finding shortest vector problem, respectively closest vector problem. Cryptosystems based on hard lattice problems are also presented and it is given a special review at application of LLL to cryptanalysis.

# Sadržaj

Glava 1

# Uvod

Problem ranca (engl. knapsack problem) je kombinatorni problem optimizacije.

**Definicija 1.1** (Problem ranca)**.** *Iz zadatog skupa od n elemenata, gdje svaki element ima pridruženu težinu $w_j$ i vrijednost $p_j$, $1 \leq j \leq n$, cilj je sakupiti određeni broj elemenata, tako da vrijednost ranca bude maksimalna, ali da težina ne pređe zadati kapacitet W.*

Problem se može preformulisati na sljedeći način: Recimo da postoji 10 komada hrane od kojih svaki komad ima određenu nutritivnu vrijednost i svoju težinu. Problem ranca je odabrati podskup hrane takav da se ne pređe zadato ograničenje u ukupnoj težini, a da je nutritivna vrijednost što veća. Kao poseban slučaj problema ranca može da se posmatra problem zbira podskupa.
On se zasniva na sljedećem: dat je skup cijelih brojeva i neki drugi cijeli broj $s$, da li zbir elemenata bilo kojeg neprazanog podskupa tog skupa daje $s$?
U cilju lakšeg razumijevanja navešćemo jednostavan primjer: Dat je skup $\{-7, -3, -2, 5, 8\}$ . Da li postoji neprazan podskup čiji je zbir nula? Odgovor je *da*, jer $\{-3, -2, 5\}$ iznosi nula.
Jedan od najranijih kripto-sistema javnog ključa je knapsack kripto-sistem, prvi put opisan od strane Ralph Merkle-a i Martin Hellman-a 1978. godine. Ovaj kripto-sistem je baziran na problemu zbira podskupa. Postoje verzije problema zbira podskupa koje su rješive, međutim, pokazalo se da ovaj problem može biti i nerješiv. Osnovna ideja Merkle-Hellman-ove šeme je transformacija teškog ili nerješivog problema zbira podskupa u problem zbira podskupa koji je lako riješiti.
U računarstvu, problem ranca ima veliki značaj u kriptografiji i teoriji složenosti algoritama. Najpoznatiji algoritam za rješavanje problema zbira podskupa je *kolizioni algoritam*. Međutim, ovaj algoritam se ne može iskoristiti za kreiranje kripto-sistema, pa se koriste drugi efikasniji algoritmi, o kojima će kasnije biti riječi.

## 1.1 Matematička priprema

U ovom dijelu ćemo uvesti osnovne definicije i pojmove, iz oblasti algebre i teorije brojeva, koji se koriste u nastavku rada. Za cijele brojeve $m$ i $n$, $m > 0$, jednoznačno su određeni cijeli brojevi $q$ i $r$ (količnik i najmanji pozitivni

ostatak), takvi da je $n = qm + r$ , $0 \leq r < m$. Svi cijeli brojevi mogu se razbiti u $m$ klasa, prema ostatku koji daju pri dijeljenju sa $m$.

**Definicija 1.2.** *Neka je $m \geq 1$ cio broj. Kažemo da su cijeli brojevi $a$ i $b$ kongruentni po modulu $m$ ako je njihova razlika $a - b$ djeljiva sa $m$, odnosno ako $m \mid a - b$. Tada pišemo*

$$a \equiv b \pmod{m}.$$

Kongruentnost po modulu $m$ je relacija ekvivalencije u skupu cijelih brojeva. Klase ekvivalencije te relacije nazivaju se klase ostataka po modulu $m$. Skup svih cijelih brojeva kongruentnih datom broju po modulu $m$ sačinjava jednu takvu klasu koju obilježavamo na sljedeći način:

$$[a] = \{x \in Z \mid x \equiv a \pmod{m}\}.$$

Svaki cijeli broj pripada tačno jednoj klasi ostataka po datom modulu $m$ i svaka klasa ostataka sadrži tačno jedan od brojeva $0, 1, 2, ..., m - 1$, a broj različitih klasa je $m$. Skup $\mathbf{Z}/m\mathbf{Z} = \{[a] \mid a \in \mathbf{Z}\}$ je skup klasa ostataka modula $m$. Unutar ovog skupa posmatramo podskup multiplikativno invertibilnih elemenata $(\mathbf{Z}/m\mathbf{Z})^*$,

$$(\mathbf{Z}/m\mathbf{Z})^* = \{a \in \mathbf{Z}/m\mathbf{Z} \mid nzd(a, m) = 1\}$$

Sljedeća lema daje potreban i dovoljan uslov da bi element $\mathbf{Z}/m\mathbf{Z}$ bio multipilikativni inverz, odnosno multiplikativno invertibilan.

**Lema 1.1.** *Neka je $a$ cio broj. Tada je*

$$a \cdot b \equiv 1 \pmod{m} \text{ za neko } b \in \mathbf{Z} \text{ akko } nzd(a, m) = 1.$$

*Broj $b$ nazivamo multiplikativnim inverzom broja $a$ po modulu $m$.*

Funkciju $\varphi(m)$ nazivamo Ojlerovom funkcijom i definišemo kao

$$\varphi(m) = \mid (\mathbf{Z}/m\mathbf{Z})^* \mid .$$

Postupak za određivanje najvećeg zajednikog djelioca prirodnih brojeva $a$ i $b$ u oznaci $nzd(a, b)$ je *Euklidov algoritam*:
Neka su $a, b \in \mathbf{Z}, a > b > 0$. Najveći zajednički djelilac $nzd(a, b)$ može se naći iterativnim postupkom deljenja sa

ostatkom. Neka je $r_0 = a, r_1 = b$ i

$$r_0 = q_1 r_1 + r_2, 0 < r_2 < r_1,$$
$$r_1 = q_2 r_2 + r_3, 0 < r_3 < r_2,$$
$$\dots$$
$$r_{k-1} = q_k r_k + r_{k+1}, 0 < r_{k+1} < r_k$$
$$\dots$$
$$r_{n-2} = q_{n-1} r_{n-1} + r_n, 0 < r_n < r_{n-1}$$
$$r_{n-1} = q_n r_n + r_{n+1}, 0 = r_{n+1}.$$

Po konstrukciji važi $r_1 > r_2 > ...$ i ostatak postaje 0 poslije konačnog broja koraka. Posljednji ostatak različit od 0 je najveći zajednički djelilac.

Bitno je napomenuti da se $nzd(a, b)$ može predstaviti kao linerana kombinacija brojeva a i b. O tome govori sljedeća lema.

**Lema 1.2.** *Ako je $d = nzd(a, b)$ za dva prirodna broja a i b onda postoje brojevi u, v tako da*

$$d = au + bv$$

*Specijalno, od svih prirodnih brojeva oblika $ax + by$, gdje su $x, y \in \mathbf{Z}$, broj $au + bv$ je najmanji.*

*Prošireni Euklidov algoritam*, pored nalaženja najvećeg zajedničkog djelioca cijelih brojeva a i b, nalazi cijele brojeve u i v koji zadovoljavaju

$$au + bv = nzd(a, b).$$

Sljedeće na šta ćemo se bazirati su algebarske strukture.

**Definicija 1.3.** *Grupa $(G, *)$ je skup G sa binarnom operacijom $*$ koja zadovoljava sljedeća svojstva:*

1. *Zatvorenost: Za svako a, b iz G, $a * b$ je takođe iz G*
2. *Asocijativnost: Za svako a, b i c iz G, $(a * b) * c = a * (b * c)$.*
3. *Neutralni element: Postoji element e iz G takav da za svaki a iz G, $e * a = a * e = a$.*
4. *Inverzni element ili inverz: Za svako a iz G, postoji element b, takođe iz G takav da $a * b = b * a = e$, gdje je e neutralni element.*

*Strukture kod kojih je zadovoljen prvi uslov nazivamo grupoidima. Za one kod kojih važi 1. i 2. kažemo da su polugrupe. Ako su zadovoljeni uslovi 1., 2. i 3. strukturu nazivamo monoid.*

**Definicija 1.4.** *Skup H je podgrupa grupe G ako je podskup od G i grupa u odnosu na operaciju definisanu na G. $H \subset G$ je podgrupa grupe G ako i samo ako je zatvorena u odnosu na * i na inverzni element.*

**Definicija 1.5.** *Za grupu $(G, \cdot)$ kažemo da je komutativna ili Abelova grupa ako za sve $a, b \in G$ važi $ab = ba$.*

Grupu $(G, \cdot)$ nazivamo "multiplikativna" grupa, a binarnu operaciju $\cdot$ "množenje". U Abelovoj grupi binarnu operaciju zapisujemo aditivno, tj. ako grupu zadamo sa $(G, +)$, onda je nazivamo "aditivna" grupa i podrazumijevamo da je Abelova. Neutralni element aditivne grupe nazivamo nula (i označavamo sa 0), a inverzni element od a označavamo sa -a i nazivamo suprotni element.

**Definicija 1.6.** *Prsten je bilo koji neprazan skup R zajedno sa dvjema binarnim operacijama + (sabiranje elemenata prstena) i $\cdot$ (množenje elemenata prstena) tako da važi:*

1. *$(R, +)$ je Abelova grupa*
2. *$(R, \cdot)$ je polugrupa, tj. množenje na R je asocijativno*
3. *množenje je distributivno u odnosu na sabiranje, tj.*

$$\forall a, b, c \in R \text{ važi}$$
$$a(b + c) = ab + bc \text{ i } (a + b)c = ac + bc$$

## 1.2 Osnovne definicije i pojmovi u kriptografiji

Tajnost poruke je osnov kriptografije. Kriptoanaliza je nauka koja se bavi razbijanjem šifri, odnosno otkrivanjem sadržaja običnog teksta na osnovu kriptograma, a bez poznavanja ključa. Kriptografija i kriptoanaliza zajedno čine nauku zvanu kriptologija.

Osnovni zadatak kriptografije je omogućiti dvjema osobama (nazivaćemo ih pošiljalac i primalac - u kriptografskoj literaturi su za njih rezervisana imena Alisa i Bob) slanje poruka na način da treća osoba (njihov protivnik - u literaturi se najčešće zove Eva ili Oskar) ne može razumjeti njihove poruke. Poruku koju pošiljalac želi poslati primaocu nazivaćemo običan tekst (engl. plaintext). To može biti tekst na njihovom jeziku, numerički podaci i drugo. Pošiljalac modifikuje običan tekst koristeći unaprijed dogovoreni ključ. Taj postupak se naziva kriptovanje, a dobijeni rezultat kriptogram (engl. ciphertext). Kod kriptovanja razlikujemo dvije transformacije: enkripciju i dekripciju. Postupkom enkripcije ili šifrovanja otvoreni tekst se transformiše u kriptovani tekst (kriptovanu poruku). Kriptovana poruka se postupkom dekripcije ili dešifrovanja vraća u običan tekst. Ključ nam omogućava da prilikom enkripcije i dekripcije vršimo transformaciju običnog teksta u kriptovani tekst i obratno.

Glavna podjela kripto-sistema je na simetrične i asimetrične kripto-sisteme. Kod simetričnih kripto-sistema, ključ za dekripciju se može izračunati poznavajući ključ za enkripciju i obratno. U stvari, najčešće su ovi ključevi identični. Sigurnost ovih kripto-sistema leži u tajnosti ključa. Zato se oni zovu i kripto-sistemi sa tajnim ključem. Kod kripto-sistema sa javnim ključem ili asimetričnih kripto-sistema, ključ za dekripciju se ne može izračunati iz ključa za enkripciju. Ovdje je ključ za enkripciju javni ključ. Naime, bilo ko može enkriptovati poruku pomoću njega, ali

samo osoba koja ima odgovarajući ključ za dekripciju (privatni ili tajni ključ) može dekriptovati tu poruku. Ideju javnog ključa prvi su javno iznijeli Whitfield Diffie i Martin Hellman 1976. godine, kada su dali predlog rješenja problema razmjenjivanja ključeva za simetrične kripto-sisteme putem nesigurnih kanala komunikacije.

Napomenućemo da je jedan od značajnijih simetričnih kripto-sistema DES (Data Encryption Standard), dok u asimetrične kripto-sisteme ubrajamo El Gamal, RSA (Rivest Shamir Adler Algoritam) i ECC (Elliptic curve cryptography).

Glava 2

# Rešetke: Osnovne definicije i svojstva

Ovo poglavlje započinjemo definicijom rešetke i navođenjem svojstava koja će se pokazati bitnim za određivanje najkraćeg vektora u rešetki.

**Definicija 2.1.** *Neka je $v_1, v_2, \ldots, v_n \in \mathbb{R}^m$ skup linearno nezavisnih vektora. Rešetka L generisana sa $v_1, v_2, \ldots, v_n$ je skup svih cjelobrojnih linearnih kombinacija vektora $v_1, v_2, \ldots, v_n$,*

$$L = \{a_1v_1 + a_2v_2 + \ldots + a_nv_n : a_1, a_2, \ldots, a_n \in \mathbb{Z}\}.$$

*Baza rešetke L je bilo koji skup linearno nezavisnih vektora koji generišu L. Svake dvije baze rešetke L imaju isti broj elemenata. Taj se broj podudara sa dimenzijom prostora.*

Pretpostavimo da je $v_1, v_2, \ldots, v_n$ baza rešetke $L$ i da je $w_1, w_2, \ldots, w_n \in L$ drugi skup vektora iz $L$. Slično kao i za vektorski prostor možemo svako $w_j$ predstaviti kao linearnu kombinaciju baznih vektora:

$$\begin{aligned} w_1 &= a_{11}v_1 + a_{12}v_2 + \ldots + a_{1n}v_n, \\ w_2 &= a_{21}v_1 + a_{22}v_2 + \ldots + a_{2n}v_n, \\ &\vdots \qquad\qquad\qquad \vdots \\ w_n &= a_{n1}v_1 + a_{n2}v_2 + \ldots + a_{nn}v_n, \end{aligned}$$

s tim što su svi $a_{ij}$ cijeli brojevi.

Ako pokušamo da izrazimo $v_i$ preko $w_j$, to će podrazumijevati proces traženja inverzne matrice matrici

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}.$$

Budući da je $v_i$ cjelobrojna linearna kombinacija vektora $w_j$, potrebno je da elementi matrice $A^{-1}$ budu cijeli

brojevi. Dakle,

$$1 = det(I) = det(AA^{-1}) = det(A)det(A^{-1}),$$

gdje su $det(A)$ i $det(A^{-1})$ cijeli brojevi, pa mora važiti $det(A) = \pm 1$. Obrnuto, ako je $det(A) = \pm 1$, iz teorije matrica imamo da $A^{-1}$ mora imati cjelobrojne elemente.

**Propozicija 2.1.** *Bilo koje dvije baze rešetke $L$ su povezane matricom, koja sadrži cjelobrojne koeficijente i determinantu, čija je vrijednost $\pm 1$.*

**Definicija 2.2.** *Integralna (ili cjelobrojna) rešetka je rešetka čiji svi vektori imaju cjelobrojne koordinate. Ekvivalentno, integralna rešetka je aditivna podgrupa polja $\mathbb{Z}^m$, za neko $m \geq 1$.*

**Definicija 2.3.** *Podskup $L$ prostora $\mathbb{R}^m$ je aditivna podgrupa ako je zatvoren u odnosu na sabiranje i oduzimanje. Naziva se diskretnom aditivnom podgrupom ako postoji pozitivna konstanta $\varepsilon > 0$ sa sljedećim svojstvom: za svako $v \in L$,*

$$L \cap \{w \in \mathbb{R}^m : \|v - w\| < \varepsilon\} = \{v\}. \tag{2.1}$$

Iz (2.1 ) vidimo da za svaki vektor $v \in L$, lopta oko $v$ poluprečnika $r$ ne sadrži niti jedan drugi vektor iz rešetke osim $v$. Definicije 2.1 i 2.3 su ekvivalentne.

**Teorema 2.1.** *Podskup prostora $\mathbb{R}^m$ je rešetka ako i samo ako je diskretna aditivna podgrupa.*

**Definicija 2.4.** *Neka je $L$ rešetka dimenzije $n$ i neka je $v_1, v_2, \ldots, v_n$ baza u $L$. Fundamentalni domen (ili fundamentalni paralelopiped) za $L$ koji odgovara ovoj bazi je skup $\mathcal{F}(v_1, v_2, \ldots, v_n) = \{t_1v_1 + \ldots + t_nv_n : 0 \leq t_i \leq 1\}$.*

Nadalje, definišemo determinantu rešetke $L$.

**Definicija 2.5.** *Neka je $L$ rešetka dimenzije $n$ i neka je $\mathcal{F}$ fundamentalni domen za $L$. Tada $n$-dimenzionalnu zapreminu domena $\mathcal{F}$ nazivamo determinantom rešetke $L$. Obilježava se sa $det(L)$.*

Vektore baze rešetke $L$ možemo posmatrati kao stranice fundamentalnog domena $\mathcal{F}$. Tada je zapremina najveća ako su vektori međusobno ortogonalni. Sljedeća propozicija nam daje gornje ograničenje za $det(L)$.

**Propozicija 2.2** (Hadamarova nejednakost)**.** *Neka je $L$ rešetka, a $v_1, v_2, \ldots, v_n$ bilo koja baza za $L$ i neka je $\mathcal{F}$ fundamentalni domen za $L$. Tada važi*

$$detL = Vol(\mathcal{F}) \leq \|v_1\|\|v_2\| \ldots \|v_n\|.$$

Drugim riječima, što je Hadamardova nejednakost bliža jednakosti, baza je ortogonalnija. Ako znamo izračunati $det(L)$ možemo provjeriti koliko je baza blizu ortogonalnoj. Determinantu je jednostavno izračunati ako je rešetka $L$ dimenzije $n$ i sadržana je u prostoru $\mathbb{R}^n$.

**Propozicija 2.3.** *Neka je $L \subset \mathbb{R}^n$, vektori $v_1, v_2, \ldots, v_n$ baza za $L$ i $\mathcal{F}$ fundametalni domen generisan tom bazom. Koordinate i-tog vektora baze zapisujemo kao*

$$v_i = (r_{i1}, \ldots, r_{in}),$$

*i pomoću njih formiramo matricu*

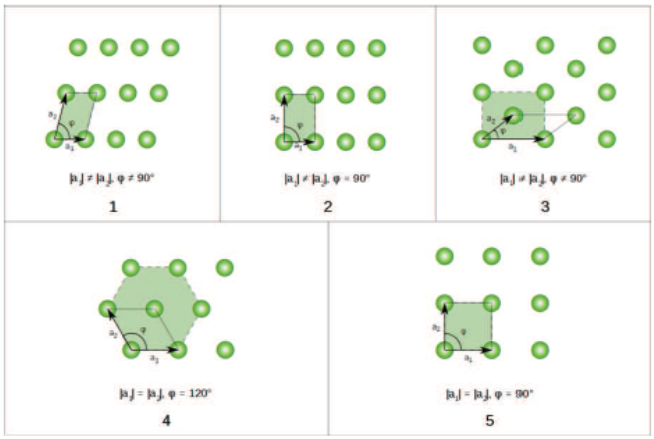
$$F = F(v_1, v_2, \ldots, v_n) = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ r_{21} & r_{22} & \ldots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \ldots & r_{nn} \end{pmatrix}.$$

*Tada je zapremina domena $\mathcal{F}$ data formulom*

$$Vol(\mathcal{F}(v_1, v_2, \ldots, v_n)) = |det(\mathcal{F}(v_1, v_2, \ldots, v_n)|.$$

**Posledica 2.1.** *Neka je $L \subset \mathbb{R}^n$ rešetka dimenzije $n$. Tada svaki fundamentalni domen od $L$ ima istu zapreminu. Dakle, $det(L)$ je nezavisna od odabira fundamentalnog domena.*

Radi ilustracije sljedećih 5 tipova rešetki se može definisati u Euklidskoj ravni $\mathbb{R}^2$, sa bazom $(a_1, a_2)$:




Slika 2.1: Pet osnovnih tipova rešetki u Euklidskoj ravni

Glava 3

# Kripto-sistemi bazirani na kongruencijama

U ovom poglavlju opisaćemo jednostavan model kripto-sistema sa javnim ključem. Ova verzija prikazuje neočekivanu vezu sa rešetkama dimenzije dva, a samim tim i fatalnu ranjivost kada je dimenzija niža.

Alisa bira veliki pozitivni cijeli broj $q$, koji je javni parametar i dva druga cijela broja $f$ i $g$, koji zadovoljavaju

$$f < \sqrt{\tfrac{q}{2}},\ \sqrt{\tfrac{q}{4}} < g < \sqrt{\tfrac{q}{2}} \text{ i } nzd(f,g) = 1$$

i računa

$$h \equiv f^{-1}g \pmod{q},$$

gdje je $0 < h < q$. Primijetimo da su $f$ i $g$ mali brojevi u poređenju sa $q$.

Alisin privatni ključ je par brojeva $(f,g)$, dok je javni ključ broj $h$. Bob bira običan tekst $m$ i slučajan broj $r$ koji zadovoljavaju

$$0 < m < \sqrt{\tfrac{q}{4}} \text{ i } 0 < r < \sqrt{\tfrac{q}{2}},$$

a zatim računa kriptogram

$$e \equiv rh + m \pmod{q}, \text{ gdje je } 0 < e < q,$$

koji šalje Alisi.

Alisa vrši dekripciju poruke računanjem prvo

$$a \equiv fe \pmod{q}, \text{ gdje je } 0 < a < q,$$

a onda i računanjem

$$b \equiv f^{-1}a \pmod{g},$$

gdje je $0 < b < q$.

Kao rezultat dobijamo $b = m$, što znači da je Alisa zaista primila Bobovu poruku $m$. Prvo što možemo da zapazimo je da $a$ zadovoljava

$$a \equiv fe \equiv f(rh + m) \equiv frf^{-1}g + fm \equiv rg + fm \pmod{q}.$$

Ograničenje veličina na $f$, $g$, $r$ i $m$ povlači da je cio broj $rg + fm$ mali,

$$rg + fm < \sqrt{\frac{q}{2}}\sqrt{\frac{q}{2}} + \sqrt{\frac{q}{2}}\sqrt{\frac{q}{4}} < q.$$

Dakle, kad Alisa izračuna $a \equiv fe \pmod{q}$, gdje je $0 < a < q$ ona dobija tačnu vrijednost

$$a = rg + fm. \tag{3.1}$$

Ovo je ključna stvar: (3.1 ) je jednakost cijelih brojeva, a ne samo brojeva kongruentnih modulo $q$. Na kraju, Alisa računa

$$b \equiv f^{-1}a \equiv f^{-1}(rg + fm) \equiv f^{-1}fm \equiv m \pmod{g},$$

gdje je $0 < b < g$. Kako je $m < \sqrt{\frac{q}{4}} < g$, slijedi da je $b = m$.

Kako Eva može da napadne ovaj sistem? Jedan način je primjenom proste sile (brute-force), da traži sve moguće privatne ključeve ili sve moguće poruke, ali ova metoda zahtijeva previše operacija. Razmotrićemo Evin zadatak, koji se odnosi na nalaženje privatnog ključa $(f, g)$ iz poznatog javnog ključa $(q, h)$.

Nije teško uočiti da Eva može da nađe bilo koji par pozitivnih cijelih brojeva $F$ i $G$ koji zadovoljavaju

$$Fh \equiv G \pmod{q}, F = \mathcal{O}(\sqrt{q}), G = \mathcal{O}(\sqrt{q}).$$

Tada $(F, G)$ može poslužiti kao ključ za dekripciju.

Prepisujući kongruenciju u obliku $Fh = G + qR$, preformulisaćemo Evin zadatak kao uporedo traženje para malih cijelih brojeva $(F, G)$ sa osobinom da je $F(1, h) - R(0, q) = (F, G)$. Jasno, Eva zna koje su vrijednosti vektora $v_1 = (1, h)$ i $v_2 = (0, q)$, od kojih je svaki dužine $\mathcal{O}(q)$ i želi da nađe linearnu kombinaciju $w = a_1v_1 + a_2v_2$ tako da $w$ bude dužine $\mathcal{O}(\sqrt{q})$.

Znači, Eva treba da nađe kratki nenulti vektor u skupu vektora $L = \{a_1v_1 + a_2v_2 : a_1, a_2 \in Z\}$, pod uslovom da su koeficijenti $a_1$ i $a_2$ cijeli brojevi.

Nesrećom za Alisu i Boba postoji ekstremno brz metod za nalaženje kratkog vektora u 2-dimenzionalnoj rešetki.

Glava 4

# Problem zbira podskupa i knapsack kripto-sistemi

Pretpostavimo da je data lista pozitivnih cijelih brojeva $(M_1, M_2, ..., M_n)$ i neki drugi cijeli broj $S$. Zadatak je naći podskup elemenata u listi čiji je zbir jednak $S$. (Možemo pretpostaviti da postoji bar jedan takav podskup.)
Postoji i drugi način da opišemo problem zbira podskupa. Lista $M = (M_1, M_2, ..., M_n)$ pozitivnih cijelih brojeva je javni podatak. Bob bira binarni vektor $x = (x_1, x_2, ..., x_n)$, koji je tajni podatak, tj. za svako $x_i$ važi da uzima vrijednost 0 ili 1. Bob računa zbir $S$ i šalje Alisi. Problem zbira podskupa zahtijeva od Alise da nađe prvobitni vektor $x$ ili drugi binarni vektor, koji daje isti zbir. Primijetimo da vektor $x$ daje Alisi informaciju koje $M_i$ treba uključiti u $S$. Ukoliko je $x_i = 1$, onda $M_i$ ulazi u zbir, međutim ako je $x_i = 0$ onda $M_i$ ne ulazi u zbir. Dakle, nalaženje vektora $x$ je isto kao i nalaženje podskupa od $M$. Lako se uočava da Alisa može da nađe $x$ razmatranjem svih $2^n$ binarnih vektora dužine $n$.

**Propozicija 4.1.** *Neka je $M = (M_1, M_2, ..., M_n)$ i neka je $(M, S)$ problem zbira podskupa. Za sve skupove cijelih brojeva $I$ i $J$, koji zadovoljavaju:*

$$I \subset \{i : 1 \leq i \leq \frac{1}{2}n\} \text{ i } J \subset \{j : \frac{1}{2}n < j \leq n\},$$

*računa se i pravi lista vrijednosti*

$$A_I = \sum_{i \in I} M_I \text{ i } B_J = S - \sum_{j \in J} M_J.$$

*Tada ove liste uključuju par skupova $I_0$ i $J_0$, koji zadovoljavaju $A_{I_0} = B_{J_0}$, a skupovi $I_0$ i $J_0$ daju rješenje problema zbira podskupa*

$$S = \sum_{i \in I_0} M_i + \sum_{j \in J_0} M_j.$$

*Broj elemenata u svakoj listi je najviše $2^{\frac{n}{3}}$, pa je trenutno vrijeme algoritma $\mathcal{O}(2^{\frac{n}{2+\varepsilon}})$, gdje je $\varepsilon$ neka mala vrijednost, koja se računa za sortiranje i upoređivanje liste.*

Ako je $n$ veliki broj, onda je uglavnom teško riješiti neki slučajni primjer problema zbira podskupa. Pretpostavimo

da Alisa posjeduje neko skriveno znanje ili potrebnu informaciju o $M$, koja joj garantuje jedinstvenost rješenja $x$ i omogućava joj njegovo lako nalaženje. Tada Alisa može da koristi problem zbira podskupa kao kripto-sistem sa javnim ključem. Bobova poruka je vektor $x$, a njegova enkriptovana poruka je suma $S = \sum x_i M_i$ i samo Alisa može lako da otkrije $x$, znajući $S$.

Koji trik Alisa koristi da osigura da samo ona može da riješi određeni problem zbira podskupa, a da niko drugi ne može? Jedna mogućnost je da koristi problem zbira podskupa koji je jednostavno riješiti, ali da nekako zamaskira to rješenje za druge.

**Definicija 4.1.** *Super-rastući cjelobrojni niz je lista pozitivnih cijelih brojeva $r = (r_1, r_2, ..., r_n)$ sa svojstvom da $r_{i+1} \geq 2r_i$, za svako $1 \leq i \leq n-1$.*

**Lema 4.1.** *Neka je $r = (r_1, r_2, ..., r_k)$ super-rastući niz. Tada je $r_k > r_{k-1} + ... + r_2 + r_1$, za svako $2 \leq k \leq n$.*

**Propozicija 4.2.** *Neka je $(M, S)$ problem zbira podskupa u kojem cijeli brojevi iz M formiraju super-rastući niz. Pretpostavimo da postoji rješenje x, da je jedinstveno i može biti izračunato sljedećim brzim algoritmom:*

| |
|---|
| for $i = n, n-1, \ldots 1$; |
| ako je $S \geq M_i$, postavi $x_i = 1$ i oduzmi $M_i$ od S |
| Inače postavi $x_i = 0$ |
| Kraj petlje |

**Primjer 4.1.** *Neka je niz $M = (3, 11, 24, 50, 115)$ super-rastući. Želimo da zapišemo $S = 142$ kao zbir elemenata skupa $\boldsymbol{M}$ sljedećim algoritmom. Prvo, $S \geq 115$, pa je $x_5 = 1$ i zamijenimo S sa $S - 115 = 27$. Dalje, $27 < 50$ i onda je $x_4 = 0$. Nastavljajući proces, $27 \geq 24$, pa je $x_3 = 1$ i S postaje $27 - 24 = 3$. Zatim, $3 < 11$, slijedi $x_2 = 0$ i konačno $3 \geq 3$, znači $x_1 = 1$. Primijetimo da se S gubi sa $3 - 3 = 0$, što znači da je vektor $x = (1, 0, 1, 0, 1)$ rješenje. Provjerom dobijamo tačan rezultat*

$$1 \cdot 3 + 0 \cdot 11 + 1 \cdot 24 + 0 \cdot 50 + 1 \cdot 115 = 142.$$

Prvi pokušaj da se napravi kripto-sistem zasnovan na *NP*-kompletnom[1] problemu je bio od strane Merkle-a i Hellman-a u kasnim 70-im (75-im). Merkle i Hellman su predložili kripto-sistem sa javnim ključem zasnovan na super-rastućem problemu zamaskiranog zbira podskupa upotrebom kongruencija.

S ciljem da kreira par javni/privatni ključ Alisa počinje sa super-rastućim nizom $r = (r_1, r_2, ..., r_n)$. Ona takođe bira dva velika cijela broja $A$ i $B$, koji su tajni i zadovoljavaju $B > 2r_n$ i $nzd(A, B) = 1$.

Alisa formira novi niz $M$, koji nije super-rastući postavljajući

$$M_i \equiv Ar_i \pmod{B}, \text{ gdje je } 0 \leq M_i \leq B.$$

[1] Problem pripada klasi NP, i nazivamo ga problemom nedeterminističke polinomske složenosti, ako se rješenje datog problema može verifikovati algoritmom polinomske složenosti. Preciznije, za unaprijed dato rješenje se utvrđuje se da li su ispunjeni svi uslovi problema. Pri tome je potrebno da on bude zadat kao problem odlučivanja, da bi odgovor bio i formalno (matematički) korektan.

Niz $M$ je Alisin javni ključ. Da bi izvršio enkripciju poruke, Bob bira običan tekst $x$, koji je binarni vektor i šalje Alisi kriptogram

$$S = \sum_{i=1}^{n} x_i M_i.$$

Alisa dekriptuje $S$ računajući prvo

$$S' \equiv A^{-1}S \pmod{B}, \text{ gdje je } 0 \leq S' < B,$$

a potom rješava problem zbira podskupa za $S'$ koristeći super-rastući niz $r$ i brzi algoritam, opisan u Propoziciji 4.2 . Ispravnost dekripcije leži u činjenici da je $S'$ kongruentan sa

$$S' \equiv A^{-1}S \equiv A^{-1}\sum_{i=1}^{n} x_i M_i \equiv A^{-1}\sum_{i=1}^{n} x_i A r_i \equiv \sum_{i=1}^{n} x_i r_i \pmod{B}.$$

Pretpostavka da je $B > 2r$ i Lema 4.1 govore Alisi da je

$$\sum_{i=1}^{n} x_i r_i \leq \sum_{i=1}^{n} r_i < 2r_n < B,$$

pa biranjem $S'$ u opsegu od 0 do $B-1$, ona obezbjeđuje dobijanje tačne jednakosti $S = \sum_{i=1}^{n} x_i M_i$, a ne samo kongruencije.

**Primjer 4.2.** *Neka je $r = (3, 11, 24, 50, 115)$ super-rastući niz koji je Alisin tajni parametar. Pretpostavimo da je ona odabrala brojeve $A = 113$ i $B = 250$. Tada je njen zamaskirani niz oblika*

$$M \equiv (3 \cdot 113, 11 \cdot 113, 24 \cdot 113, 50 \cdot 113, 115 \cdot 113) \pmod{250} = (89, 243, 212, 150, 245).$$

*Bob treba da pošalje Alisi tajnu poruku $x = (1, 0, 1, 0, 1)$. On enkriptuje $\boldsymbol{x}$ računajući*

$$S = x \cdot M = 1 \cdot 89 + 0 \cdot 243 + 1 \cdot 212 + 0 \cdot 150 + 1 \cdot 245 = 546.$$

*Kad dobije poruku S, Alisa je pomnoži sa 177, multiplikativnim inverzom broja 113 po modulu 250 i dobija*

$$S' \equiv 177 \cdot 546 = 142 \pmod{250}.$$

*Zatim, Alisa koristi algoritam iz Propozicije 4.2 da riješi $S' = x \cdot r$, za super-rastući niz $\boldsymbol{r}$ i na taj način dolazi do originalne poruke $\boldsymbol{x}$ (Primjer 4.1).*

| Alisa | Bob |
|---|---|
| Generisanje ključa | |
| Bira super rastući niz $r = (r_1, r_2, ..., r_k)$.<br>Bira brojeve A i B, pri čemu je $B > 2r_n$ i $nzd(A, B) = 1$.<br>Računa $M_i \equiv Ar_i \pmod{B}$, gdje je $1 \leq i \leq n$.<br>Objavljuje javni ključ $M = (M_1, M_2, ..., M_n)$. | |
| Enkripcija | |
| | Bira običan tekst **x**.<br>Koristi Alisin javni ključ M<br>da izračuna $S = x \cdot M$.<br>Šalje Alisi kriptogram S. |
| Dekripcija | |
| Računa $S' \equiv A^{-1}S \pmod{B}$.<br>Rješava problem zbira podskupa $S'$,<br>koristeći super-rastući niz **r**.<br>Običan tekst **x** zadovoljava $x \cdot r = S$. | |

Kripto-sistemi zasnovani na problemu zamaskiranog zbira podskupa poznati su kao *kripto-sistemi zbira podskupa* ili *knapsack* kripto-sistemi. Osnovna ideja je početi sa super-rastućim nizom, koji je tajni podatak, zamaskirati ga, koristeći prilagodljive linearne operacije i objaviti zamaskirani niz kao javni ključ.

*Napomena:* Važno pitanje koje treba uzeti u obzir, a tiče se knapsack kripto-sistema, je kolika treba da bude veličina različitih parametara kako bi se postigao željeni nivo sigurnosti. Postoji $2^n$ binarnih vektora $x = (x_1, x_2, ..., x_n)$, a kao što je već viđeno u Propoziciji 4.2, postoji kolizioni algoritam, pa je moguće prekinuti knapsack kripto-sistem u $\mathcal{O}(2^{\frac{n}{2}})$ operacija. Dakle, s obzirom na to da dobijamo sigurnost reda $2^k$, potrebno je uzeti $n > 2k$. Ali iako ovo obezbjeđuje sigurnost protiv kolizionog napada, ne znači da sprečava postojanje drugog, efikasnijeg napada.

*Napomena:* Pretpostavljajući da smo odabrali vrijednost za $n$, pitamo se koliko velike druge parametre moramo da uzmemo? Poželjno je da $r_1$ ne bude previše malo, jer je onda lakše izvesti napad, pa uzimamo da je $r_1 > 2^n$. Priroda super-rastućeg niza povlači da je

$$r_n > 2r_{n-1} > 4r_{n-1} > ... > 2^n r_1 > 2^{2n}.$$

Tada je $B > 2r_n = 2^{2n+1}$, pa nalazimo da elementi javnog ključa $M_i$ i kriptogram $S$ zadovoljavaju

$$M_i = \mathcal{O}(2^{2n}) \text{ i } S = \mathcal{O}(2^{2n}).$$

Dakle, javni ključ $M$ je lista od $n$ cijelih brojeva, gdje je svaki broj dužine približno $2n$ bita, običan tekst $x$ se sastoji od informacije od $n$ bita, dok je kriptogram približno $2n$ bita dužine.

Sada ćemo ukratko opisati kako Eva može da preformuliše problem zbira podskupa upotrebljavajući vektore. Pretpostavimo da ona želi da zapiše $S$ kao zbir podskupa iz skupa $M = (m_1, ..., m_n)$. Njen prvi korak je formiranje matrice

$$\begin{pmatrix} 2 & 0 & 0 & \dots & 0 & m_1 \\ 0 & 2 & 0 & \dots & 0 & m_2 \\ 0 & 0 & 2 & \dots & 0 & m_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & m_n \\ 1 & 1 & 1 & \dots & 1 & S \end{pmatrix}. \tag{4.1}$$

Vrste matrice ( 4.1 ) su vektori, koje ćemo obilježavati sa

$$\begin{aligned} v_1 &= (2, 0, 0, \dots, 0, m_1), \\ v_2 &= (0, 2, 0, \dots, 0, m_2), \\ &\vdots \qquad\qquad \vdots \\ v_n &= (0, 0, 0, \dots, 2, m_n), \\ v_{n+1} &= (1, 1, 1, \dots, 1, S). \end{aligned}$$

Baš kao i u 2-dimenzionalnom primjeru opisanom u poglavlju 3, Eva posmatra skup svih cjelobrojnih linearnih kombinacija vektora $v_1, v_2, \dots, v_{n+1}$,

$$L = \{a_1 v_1 + a_2 v_2 + \dots + a_n v_n + a_{n+1} v_{n+1} : a_1, a_2, \dots, a_{n+1} \in \mathbb{Z}\}.$$

Sada pretpostavimo da je $x = (x_1, x_2, \dots, x_n)$ rješenje datog problema zbira podskupa. Tada rešetka $L$ sadrži vektor

$$t = \sum_{i=1}^{n} x_i v_i - v_{n+1} = (2x_1 - 1, 2x_2 - 1, \dots, 2x_n - 1, 0),$$

čija je posljednja kordinata jednaka nuli, jer je $S = x_1 m_1 + \dots + x_n m_n$.

Dolazimo do suštine materije. Pošto $x_i$ uzima vrijednosti 0 ili 1, slijedi da će vrijednosti za $2x_i - 1$ biti $\pm 1$, pa je vektor $t$ prilično kratak, $\|t\| = \sqrt{n}$. S druge strane, vidjeli smo da je $m_i = \mathcal{O}(2^{2n})$ i $S = \mathcal{O}(2^{2n})$, pa su svi vektori koji generišu $L$ dužine $\|v\| = \mathcal{O}(2^{2n})$. Zaključujemo da je mala vjerovatnoća da rešetka $L$ sadrži bilo koji nenulti vektor dužine $\sqrt{n}$ koji nije $t$. Ako pretpostavimo da Eva zna algoritam za nalaženje malog nenultog vektora u rešetki,onda ona može otkriti običan tekst $x$.

Algoritmi za nalaženje kratkih vektora u rešetkama se nazivaju *algoritmi redukcije rešetke*. Najpoznatiji od tih je LLL algoritam i njegove varijante, kao što je LLL-BKZ.

Glava 5

# Kratki vektori u rešetkama

Sve rešetke u $\mathbb{R}^n$ imaju beskonačno mnogo elemenata, dok se u kriptografiji razmatraju rešetke nad konačnim poljem. Dva matematička problema koja se koriste u kripto-sistemima na bazi rešetki su problem najkraćeg vektora (engl. Shortest Vector Problem - SVP) i problem nalaženja najbližeg vektora (engl. Closest Vector Problem - CVP). SVP problem se sastoji iz pronalaženja najkraćeg vektora u nekoj rešetki, za datu bazu rešetke. CVP problem zahtijeva da se, za datu bazu rešetke i neki vektor $v$ koji nije dio rešetke, pronađe najkraći vektor koji pripada toj rešetki, a koji je najmanje udaljen od navedenog vektora $v$.[4]

## 5.1 Problem najkraćeg i najbližeg vektora

Neke od varijanti SVP i CVP problema, koje se koriste u teoriji i praksi, su:

**Problem najkraće baze** (SBP): Nalazimo bazu $v_1, \ldots, v_n$, koja je u nekom smislu najkraća za rešetku. Na primjer, zahtijevamo da

$$\max_{1 \leq i \leq n} \|v_i\| \quad ili \quad \sum_{i=1}^{n} \|v_i\|$$

bude minimizirano. Postoje različite verzije SBP-a, zavisno od načina mjerenja "veličine" baze.

**Aproksimativni problem najkraćeg vektora** (apprSVP): Neka je $\psi(n)$ funkcija koja zavisi od $n$. U rešetki $L$ dimenzije $n$, naći nenulti vektor čija je dužina najviše $\psi(n)$ puta veća od dužine najkraćeg vektora rešetke. Drugim riječima, ako je $v_{shortest}$ najkraći nenulti vektor u $L$ naći vektor $v \in L$ koji zadovoljava

$$\|v\| \leq \psi(n)\|v_{shortest}\|.$$

Rješenje problema se mijenja izborom funkcije $\psi(n)$.

**Aproksimativni problem najbližeg vektora** (apprCVP): Analogno apprSVP-u, samo što tražimo aproksimativno rješenje za CVP.

### 5.1.1 Problem najkraćeg vektora

Neka je data baza vektorskog prostora $V$ i norma $N$ (euklidska norma $l^2$) za rešetku $L$. SVP problem se sastoji u pronalaženju najkraćeg nenultog vektora u $V$, izmjerenog normom $N$ u rešetki $L$, odnosno $\lambda(L)$, gdje je $\lambda(L) = min\{\|v\|_N : v \in L, v \neq 0\}$.

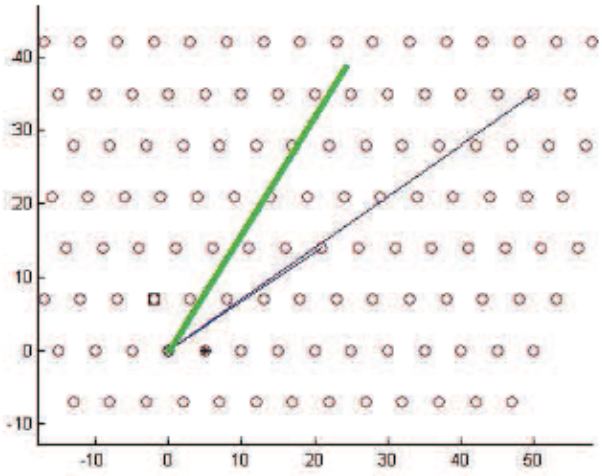
LLL algoritam za redukciju baze omogućava pronalaženje kratkog vektora u polinomijalnom vremenu, ali ne rješava problem u potpunosti. Pošto za sada ne postoji efikasan algoritam koji će da riješi SVP (CVP) u bilo kojoj izabranoj većoj dimenziji, uobičajeno je da se ovi problemi definišu kao aproksimacija početnog problema.[4] Dati problem je naveden kao apprSVP.

Posmatrajmo sljedeći primjer rešetke koja je generisana vektorima $i = (50, 35)$ i $j = (21, 14)$ na $\mathbb{R}^2$. Najkraći nenulti vektor je $a = (5, 0)$. Sljedeći najkraći je $b = (-2, 7)$. U opštem slučaju rešetka je generisana na $\mathbb{R}^n$, što čini ovaj problem NP-teškim.

### 5.1.2 Problem najbližeg vektora

U CVP-u je data baza vektorskog prostora $V$ i norma $N$ (euklidska norma $l^2$) za rešetku $L$ kao i za vektor $v$ tog prostora $V$, koji ne pripada rešetki $L$. Problem se sastoji u pronalaženju vektora koji pripada rešetki $L$ koji je najbliži datom vektoru $v$. Kao i za SVP, postoji takođe i aproksimacija CVP problema.[4]. Odgovarajući problem smo nazvali apprCVP.

Posmatraćemo primjer sa rešetkom $L$ koja je generisana kao u primjeru SVP problema i slučajni vektor $v$ (deblja linija), koji ne pripada rešetki $L$. Potrebno je pronaći vektor koji pripada rešetki $L$, a koji je najbliži vektoru $v$.




Slika 5.1: Primjer CVP problema na rešetki L generisanoj na $R^2$

## 5.2 Teorema Hermita i teorema Minkovskog

Algoritmi za nalaženje najkraćeg i najbližeg vektora unutar rešetke su aproksimativni, pa je dobro imati neki način ocjene koliko su ti vektori zaista dobri za datu rešetku. Kolika je dužina najkraćeg vektora zapravo zavisi ponajviše od dimenzije i determinante rešetke.

**Teorema 5.1** (Hermit)**.** *Svaka rešetka $L$ dimenzije $n$ sadrži nenulti vektor $v \in L$ koji zadovoljava*

$$\|v\| \leq \sqrt{n} det(L)^{\frac{1}{n}}.$$

Prvo ćemo se upoznati sa osnovnim definicijama, a zatim formulisati teoremu Minkovskog.

**Definicija 5.1.** *Za svako $a \in \mathbb{R}^n$ i bilo koje $R > 0$, (zatvorena) lopta poluprečnika $R$, sa centrom u $a$ je skup*

$$\mathbb{B}_R(a) = \{x \in \mathbb{R}^n : \|x - a\| \leq R\}.$$

**Definicija 5.2.** *Neka je $S$ podskup od $\mathbb{R}^n$.*

*(a) $S$ je ograničen ako su dužine vektora iz $S$ ograničene. Ekvivalentno, $S$ je ograničen, ako postoji lopta poluprečnika $R$, tako da je $S$ sadržan u lopti.*

*(b) $S$ je simetričan ako za svaku tačku $a \in S$, $-a$ je takođe iz $S$.*

*(c) $S$ je konveksan ako za bilo koje dvije tačke $a$ i $b$ iz $S$ poligonalna linija koja povezuje tačke $a$ i $b$ leži u $S$.*

*(d) $S$ je zatvoren ako zadovoljava sljedeće svojstvo: Ako je tačka $a \in \mathbb{R}^n$ takva da svaka lopta $\mathbb{B}_R(a)$ sadrži tačku iz $S$, onda je i $a \in S$.*

**Teorema 5.2** (Minkovski)**.** *Neka je $L \subset \mathbb{R}^n$ rešetka dimenzije $n$ i neka je $S \subset \mathbb{R}^n$ simetričan konveksan skup, za čiju zapreminu važi*

$$Vol(S) > 2^n det(L).$$

*Tada $S$ sadrži nenulti vektor rešetke. Ako je $S$ takođe zatvoren, onda je dovoljno uzeti $Vol(S) \geq 2^n det(L)$.*

## 5.3 Gausova heuristika

Najkraći vektor rešetke nikada nije jedinstven, postoji uvijek više od jednog vektora dužine $l$ ( to su $v$ i $-v$, ako je $v$ najkraći ). U praksi dužina $l$ nije uvijek poznata. U ovom slučaju moguće je približno odrediti dužinu ovog vektora heuristički, koristeći Gausovu heuristiku. Gausova heuristika predviđa da će broj tačaka rešetke unutar datog skupa $S$ biti približno jednak količniku zapremine skupa i zapremine paralelopipeda rešetke (determinanta rešetke).[5]

**Teorema 5.3.** *Neka je $\mathbb{B}_R(a)$ lopta poluprečnika $R$ u $\mathbb{R}^n$. Tada je zapremina lopte $\mathbb{B}_R(a)$ jednaka*

$$Vol(\mathbb{B}_R(a)) = R^n \cdot \frac{\sqrt{\pi^n}}{\Gamma(\frac{n}{2}+1)}$$

*gdje je $\Gamma(x)$ gama funkcija[1]. Za velike vrijednosti $n$ zapremina lopte $\mathbb{B}_R(a)$ je približno data sa*

$$Vol(\mathbb{B}_R(a))^{\frac{1}{n}} \approx \sqrt{\frac{2\pi e}{n}} R. \tag{5.1}$$

Pomoću navedene teoreme možemo poboljšati ocjenu u Hermitovoj teoremi za velike vrijednosti $n$. Budući da je lopta $\mathbb{B}_R(0)$ ograničena, zatvorena, konveksna i simetrična, po Teoremi 5.1 ako odaberemo poluprečnik $R$ tako da važi

$$Vol(\mathbb{B}_R(0) \geq 2^n det(L)$$

lopta $\mathbb{B}_R(0)$ će sadržati tačku rešetke koja nije nula. Ako je $n$ veliko, zapreminu lopte $\mathbb{B}_R(0)$ možemo aproksimirati sa ( 5.1 ), pa moramo odabrati $R$ takav da važi

$$\sqrt{\frac{2\pi e}{n}} R \gtrapprox 2det(L)^{\frac{1}{n}}.$$

Za velike vrijednosti dimenzije $n$ postoji nenulti vektor $v \in L$ koji zadovoljava

$$\|v\| \lessapprox \sqrt{\frac{2n}{\pi e}} \cdot (det(L))^{\frac{1}{n}}.$$

Ovim smo poboljšali $\sqrt{n}$ iz Teoreme 5.1 za konstantu $\sqrt{\frac{2}{\pi e}} \approx 0.484$. Nije poznato kako tačno za veliko $n$ ograničiti dužinu najkraćeg vektora unutar rešetke. Ipak, koristeći sljedeći princip možemo tu dužinu aproksimirati:

> Neka je $\mathbb{B}_R(0)$ lopta sa centrom u 0. Tada je broj tačaka rešetke u $\mathbb{B}_R(0)$ približno jednak količniku zapremine lopte $\mathbb{B}_R(0)$ i zapremine fundamentalnog domena $\mathcal{F}$. Broj elemenata u $\mathbb{B}_R(0) \cap L$ možemo posmatrati kao broj kopija $\mathcal{F}$ koje su smještene u lopti $\mathbb{B}_R(0)$. Na ovaj način dolazimo do Gausove heuristike.

**Definicija 5.3.** *Neka je $L$ rešetka dimenzije $n$. Očekivana Gausova najkraća dužina je*

$$\lambda(L) = \sqrt{\frac{n}{2\pi e}} (det(L))^{\frac{1}{n}}.$$

[1]Gama funkcija je proširenje faktorijelske funkcije u realne i kompleksne brojeve. Za kompleksan broj z sa pozitivnim realnim dijelom, gama funkcija je definisana sa $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$.

*Gausova heuristika nam kaže da će najkraći nenulti vektor* $\boldsymbol{v}$ *u proizvoljno odabranoj matrici zadovoljavati*

$$\|v\| \approx \lambda(L).$$

*Preciznije, za odabrano* $\varepsilon > 0$*, ako je* $n$ *dovoljno veliko, proizvoljno odabrana rešetka* $L$ *dimenzije* $n$ *će zadovoljavati*

$$(1-\varepsilon)\lambda(L) \leq \|v\| \leq (1+\varepsilon)\lambda(L).$$

*Napomena:* Za male vrijednosti $n$ bolje je koristiti tačnu formulu za zapreminu $\mathbb{B}_R(0)$, pa je očekivana Gausova najkraća dužina za malo $n$ jednaka

$$\lambda(L) = \frac{(\Gamma(1+\frac{n}{2})det(L))^{\frac{1}{n}}}{\sqrt{\pi}}.$$

Gausova heuristika se pokazala korisnom kod pronalaženja kratkih vektora u rešetki. Naime, ako je najkraći vektor u rešetki $L$ značajno kraći od $\lambda(L)$ algoritmi redukcije rešetke (poput LLL algoritma) jednostavnije pronalaze najkraći vektor. Analogno, Gausova heuristika za CVP problem se svodi na to da za rešetku $L \in \mathbb{R}^n$ dimenzije $n$ i slučajno odabrani vektor $w \in \mathbb{R}^n$, vektor $v \in L$ koji je najbliži $w$ zadovoljava

$$\|v - w\| \approx \lambda(L).$$

Takođe, ako $L$ sadrži tačku koja je značajno bliža vektoru $w$ od $\lambda(L)$, algoritmi redukcije rešetke jednostavnije rješavaju CVP.

**Primjer 5.1.** *Neka je* $(m_1, m_2, \ldots, m_n, S)$ *knapsack problem. Rešetka* $L_{M,S}$ *je generisana vektorima, koji su vrste matrice ( 4.1 ) . Matrica* $L_{M,S}$ *je dimenzije* $n+1$ *i determinanta* $detL_{M,S} = 2^n S$*. Ranije smo pomenuli da broj* $S$ *zadovoljava* $S = \mathcal{O}(2^{2n})$*, pa je* $S^{\frac{1}{n}} \approx 4$*. Ovo nam dozvoljava da aproksimiramo Gausovu najkraću dužinu kao*

$$\lambda(L) = \sqrt{\frac{n+1}{2\pi e}}(det(L_{M,S}))^{\frac{1}{n+1}} = \sqrt{\frac{n+1}{2\pi e}}(2^n S)^{\frac{1}{n+1}} \approx \sqrt{\frac{n}{2\pi e}} \cdot 2S^{\frac{1}{n}} \approx \sqrt{\frac{n}{2\pi e}} \cdot 8 \approx 1.936\sqrt{n}.$$

*S druge strane, kao što je opisano u poglavlju 4, rešetka* $L_{M,S}$ *sadrži vektor* $\boldsymbol{t}$ *dužine* $\sqrt{n}$*, pa znajući* $\boldsymbol{t}$ *možemo riješiti problem zbira podskupa. Zato je i rješavanje SVP za rešetku* $L_{M,S}$ *slično kao rješavanje problema zbira podskupa.*

Glava 6

# Babai algoritam i korišćenje "dobre" baze za rješavanje apprSVP

Ako rešetka $L \subset \mathbb{R}^n$ ima bazu $v_1, \ldots, v_n$, koja sadrži vektore koji su u parovima ortogonalni, tj.

$$v_i \cdot v_j = 0, \text{ za svako } i \neq j,$$

problemi SVP i CVP se lako rješavaju. Dakle, za računanje SVP razmatramo dužinu bilo kojeg vektora u $L$, datog formulom

$$\|a_1v_1 + a_2v_2 + \ldots + a_nv_n\|^2 = a_1^2\|v_1\|^2 + a_2^2\|v_2\|^2 + \ldots + a_n^2\|v_n\|^2.$$

Kako su koeficijenti $a_1, \ldots, a_n \in \mathbb{Z}$, vidimo da su najkraći nenulti vektori u $L$ zapravo najkraći vektori iz skupa $\{\pm v_1, \ldots, \pm v_n\}$. Slično, pretpostavimo da želimo da nađemo vektor u $L$ koji je najbliži datom vektoru w $\in \mathbb{R}^n$. Kako je $L \subset \mathbb{R}^n$ i $L$ je dimenzije $n$, postoje koeficijenti $t_1, \ldots, t_n \in \mathbb{R}$ takvi da je

$$w = t_1v_1 + t_2v_2 + \ldots + t_nv_n.$$

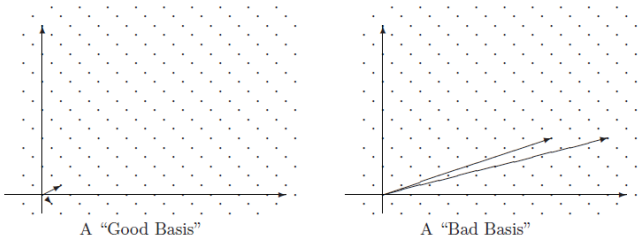
Tada za vektor $v = a_1v_1 + a_2v_2 + \ldots + a_nv_n \in L$ imamo:

$$\|v - w\|^2 = (a_1 - t_1)^2\|v_1\|^2 + (a_2 - t_2)^2\|v_2\|^2 + \ldots + (a_n - t_n)^2\|v_n\|^2. \tag{6.1}$$

Koeficijenti $a_i$ su cijeli brojevi, pa je jednakost ( 6.1 ) minimizirana ako za svako $a_i$ uzmemo onaj cijeli broj koji je najbliži odgovarajućem $t_i$.

Ovaj postupak neće dobro riješiti probleme SVP-a i CVP-a za one baze rešetke čiji vektori nijesu ortogonalni.

Na slici 6.1 su prikazane dvije baze za istu rešetku. Prva baza je "dobra", u smislu da su vektori stvarno ortogonalni, a druga baza je "loša" jer je ugao između baznih vektora prilično mali.




Slika 6.1: Dvije različite baze za jednu istu rešetku

**Teorema 6.1** (Babai Algoritam najbližeg čvora)**.** *Neka je $L \subset \mathbb{R}^n$ rešetka sa bazom $v_1, \ldots, v_n$ i neka je $w \in \mathbb{R}^n$ proizvoljan vektor. Ako su vektori baze dovoljno ortogonalni problem CVP možemo riješiti na sljedeći način:*

*Pišemo $w = t_1v_1 + t_2v_2 + \ldots + t_nv_n$, gdje su $t_1, \ldots, t_n \in \mathbb{R}$.*

*Postavimo $a_i = \lceil t_i \rfloor$ za $i = 1, 2, \ldots, n$.*

*Rješenje problema je vektor $v = a_1v_1 + a_2v_2 + \ldots + a_nv_n$.*

*Vektor $v$ će biti dobro rješenje za apprCVP ako je baza dovoljno "blizu" ortogonalnoj. U suprotnom vektor $v$ neće biti dobro rješenje.*

Glava 7

# Kripto-sistemi zasnovani na složenosti rešetke

Sredinom 90-ih su bili uvedeni neki kripto-sistemi čiji su osnovni problemi bili SVP i/ili CVP u rešetki $L$ dimenzije $n$. Najvažniji od njih su Ajtai-Dwork kriptosistem, GGH kriptosistem Goldreich-a, Goldwasser-a i Halvei-a i NTRU kriptosistem predložen od strane Hoffstein-a, Pipher-a i Silvermana.
Motivacija za uvod ovih kripto-sistema je bila dvostruka. Prvo, to je određenost interesa da imamo kripto-sisteme zasnovane na različitosti teških matematičkih problema. Drugo, kripto-sistemi zasnovani na rešetki su mnogo brži od faktorizacije ili problema diskretnog logaritma kao što su El Gamal, RSA i ECC. Grubo govoreći,u cilju postizanja $k$ bita sigurnosti enkripcija i dekripcija za El Gamal, RSA i ECC zahtijeva $\mathcal{O}(k^3)$ operacija, dok enkripcija i dekripcija za sisteme bazirane na rešetki zahtijeva $\mathcal{O}(k^2)$ operacija. Dalje, jednostavne operacije iz linearne algebre jednostavno je implementirati softverski i hardverski. Moramo napomenuti da analiza sigurnosti kripto-sistema zasnovanih na teoriji brojeva i diskretnom logaritmu nije ni blizu objašnjena kao što je to slučaj sa sistemima zasnovanim na rešetki. Ipak jako je malo takvih sistema u poređenju sa kripto-sistemima kao što je RSA.

Ataji-Dwork sistem je interesantan zbog činjenice da su Ataji i Dwork dokazali potpunu sigurnost njihovog sistema. Sistem ne važi jedino ako najgori slučaj problema rešetke može biti riješen u polinomijalnom vremenu. Ovaj važan teorijski rezultat je neutralizovan zbog ograničenja da će veličina ključa biti $\mathcal{O}(n^4)$, što dovodi do ogromnih ključeva.

GGH kripto-sistem je jedan od najzapaženijih kriptosistema zasnovanih na složenosti rešetke. GGH, kao i drugi sistemi na bazi rešetke koristi problem najbližeg vektora (CVP). Osnovna ideja je za bilo koju bazu rešetke jednostavno generisati vektor koji je blizu neke tačke koja pripada rešetki, na primjer uzimanjem neke tačke u rešetki i dodavanjem nekog malog vektora greške.[2] S druge strane, da bi iz ovog novog vektora dobili prvobitni vektor koji pripada rešetki potrebna je posebna baza.
Privatni ključ u GGH je baza $B$ rešetke $L$ koja posjeduje "povoljna" svojstva – kratki, gotovo ortogonalni vektori i unimodalna matrica U. To je po definiciji kvadratna matrica sa cjelobrojnim elementima i determinantom jednakom $+1$ ili $-1$. Javni ključ je druga baza rešetke oblika $B' = UB$.
Za neku poruku $M$, prostor poruke se sastoji od vektora $\{\lambda_1, \ldots, \lambda_n\}$ u intervalu $-M < \lambda_i < M$.[4]

Naravno, GGH kripto-sistem nije siguran za tri dimenzije jer čak i kada uzimamo brojeve koji su dovoljno veliki da pretragu učine beskorisnom, puno je brzih algoritama koji pronalaze "povoljnu" bazu u nižim dimenzijama. Algoritam za traženje "povoljne" baze u 2-dimenzionalnom prostoru datira još od Gaussa.

NTRU kripto-sistem je sistem sa javnim ključem. Akronim NTRU je nastao od naziva algoritma "Nth Degree Truncated Polynomial Ring Units". Sastoji se iz dva algoritma, NTRUEncrypt koji se koristi za enkripciju i NTRUSign koji se koristi za digitalni potpis. Navedeni kripto-sistem se zasniva na težini faktorisanja određenih polinoma sa veoma malim koeficijentima. Napad na ovakav sistem je usko povezan sa rješavanjem problema već pomenutog najbližeg vektora (CVP). Najpoznatiji je napad koji se zasniva na redukciji rešetke. Najčešće korišćeni algoritam za redukciju rešetke je LLL algoritam. Rezultat uspješnog napada redukcijom rešetke je potpun pad sistema jer se dovodi u pitanje privatni ključ. Ukoliko je dimenzija rešetke veća i najkraći vektor duži nije jednostavno izvršiti napad, pa je ova vrsta napada otežana ako se izaberu dovoljno sigurni početni parametri.

Glava 8

# Rješavanje SVP i CVP problema primjenom LLL algoritma

Za mnoge probleme zasnovane na rešetki se pretpostavlja ili je već dokazano da su generalno teški i kao takvi su pogodni za osnove pojedinih kripto-sistema. Nadalje, težina najgoreg slučaja određenih problema zasnovanih na rešetkama se koristi kao osnova za kripto-sisteme.

Algoritmi zasnovani na redukciji baze rešetke imaju za cilj da se na osnovu početne baze nekom transformacijom dođe do nove baze iste rešetke, koja se sastoji iz relativno kratkih i ortogonalnih vektora. LLL algoritam je jedan od ranih primjera efikasnih algoritama sa redukcijom baze u polinomijalnom vremenu. Algoritam je uz dodatna poboljšanja korišćen za razbijanje nekoliko kripto-sistema i tako dobio status ozbiljnog alata u kriptoanalizi. LLL algoritam su 1982.godine pronašli Ardžen Lenstra, Henrik Lenstra i Laslo Lovaš.

LLL algoritam rješava probleme SVP, CVP, apprSVP, apprCVP do na konstantu $C^n$, gdje je $C$ mala konstanta, a $n$ dimenzija rešetke. Za rešetku dimenzije 2 koristimo Gausovu redukciju.

## 8.1 Gausova redukcija rešetke

Neka je $L \subset \mathbb{R}^2$ dvodimenzionalna rešetka sa baznim vektorima $v_1$ i $v_2$. Pretpostavimo da je $\|v_1\| < \|v_2\|$. Ideja algoritma je, dok god je moguće smanjivati $v_2$ za neki sadržalac drugog vektora baze $v_1$. Jedna mogućnost je da $v_2$ zamijenimo sa vektorom

$$v_2^* = v_2 - \frac{v_1 \cdot v_2}{\|v_1\|^2} v_1,$$

koji je ortogonalan na $v_1$. Međutim, nije neophodno da je $v_2^*$ iz rešetke $L$. Zato, vektor $v_2$ zamijenimo vektorom

$$v_2 - \left\lfloor \frac{v_1 \cdot v_2}{\|v_1\|^2} \right\rceil v_1,$$

gdje je sa $\lfloor x \rceil$ dat najbliži cijeli broj realnom broju $x$. Ako je $v_2$ i dalje veći, algoritam se zaustavlja. Inače zamijenimo $v_1$ i $v_2$ i ponovimo proces. Sljedeća propozicija pokazuje da se algoritam zaustavlja za konačno mnogo

koraka, pa je dobijena baza iz $L$ jako dobra.

**Propozicija 8.1.** *Neka je $L \subset \mathbb{R}^2$ dvodimenzionalna rešetka sa baznim vektorima $v_1$ i $v_2$. Sljedeći algoritam je konačan i daje dobru bazu za $L$:*

**Input**: Vektori baze $v_1, v_2$ za rešetku $L$ dimenzije 2
**Output**: Dobra baza za $L$

```
while do
    if ||v2|| < ||v1|| then
        zamijenimo v1 i v2;
    end
    m = ⌊(v1·v2)/||v1||²⌉;
    if m = 0 then
        vratimo trenutne vektore baze v1 i v2;
        break;
    else
        zamijenimo v2 sa v2 − mv1;
    end
end
```

*Po završetku algoritma $v_1$ je najkraći nenulti vektor u rešetki $L$, tj. ovaj algoritam rješava SVP.*

## 8.2 LLL algoritam - opis

Problem SVP postaje teži kako se dimenzija povećava. LLL algoritam uspješno rješava problem najkraćeg i najbližeg vektora u većim dimenzijama. Pretpostavimo da rešetka $L$ dimenzije $n$ ima bazu $B = \{v_1, v_2, \ldots, v_n\}$. Bazu $B$ želimo transformisati u "bolju" bazu, tj. u bazu čiji su vektori što kraći, počevši s onim koji je najkraći. Takođe, želimo da su vektori što ortogonalniji, odnosno da je proizvod $v_i \cdot v_j$ što bliži nuli za sve $i, j$. Da bi to postigli na bazu $B$ primjenjujemo Gram-Šmitov postupak ortogonalizacije. Za $i = 1$ je $v_1^* = v_1$, a za $i \geq 2$ imamo

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \text{ gdje je } \mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \text{ za sve } 1 \leq j \leq i-1.$$

Ovako dobijena baza $B^* = \{v_1^*, v_2^*, \ldots v_n^*\}$ je ortogonalna baza za prostor generisan vektorima iz $B$, ali kako se unutar Gram-Šmitovog postupka pojavljuju koeficijenti koji nisu cijeli brojevi, $B^*$ nije baza za rešetku $L$ generisanu sa vektorima $v_1, v_2, \ldots, v_n$. Međutim, pokazaćemo da te dvije baze imaju istu determinantu.

**Propozicija 8.2.** *Neka je $B = \{v_1, v_2, \ldots, v_n\}$ baza za rešetku $L$, a $B^* = \{v_1^*, v_2^*, \ldots v_n^*\}$ odgovarajuća baza dobijena Gram-Šmitovim postupkom ortogonalizacije. Tada važi:*

$$det(L) = \prod_{i=1}^{n} \|v_i^*\|$$

Prije samog LLL algoritma definišimo pojam LLL redukovane baze. U definiciji koristimo Gram-Šmitovu bazu $B^*$.

**Definicija 8.1.** *Neka je $B = \{v_1, v_2, \ldots, v_n\}$ baza za rešetku $L$, a $B^* = \{v_1^*, v_2^*, \ldots v_n^*\}$ odgovarajuća baza dobijena Gram-Šmitovim postupkom ortogonalizacije. Neka je $\delta \in \left[\frac{1}{4}, 1\right]$. Za bazu $B$ kažemo da je LLL redukovana sa konstantom $\delta$ ako zadovoljava:*

1. *Uslov veličine*

$$|\mu_{i,j}| = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \leq \frac{1}{2} \text{ za sve } 1 \leq j < i \leq n.$$

2. *Lovašov uslov*

$$\|v_i^*\|^2 \geq (\delta - \mu_{i,j-1}^2)\|v_{i-1}^*\|^2 \text{ za sve } 1 < i \leq n.$$

*Lovašov uslov možemo zapisati i kao*

$$\|v_i^* + \mu_{i,j-1}^2 v_{i-1}^*\|^2 \geq \delta\|v_{i-1}^*\|^2 \text{ za sve } 1 < i \leq n.$$

Ovaj uslov kontroliše normu vektora $\|v_i^*\|$. Naime, kako Gram-Šmitov algoritam zavisi od poretka vektora unutar baze, dobijena baza se može promijeniti ako $v_i$ i $v_{i-1}$ zamijene mjesta, tj. mogu se promijeniti $v_i^*$ i i $v_{i-1}^*$. Lovašov uslov osigurava da se $\|v_i^*\|$ ne smanji previše. Najčešće se LLL redukcija radi sa $\delta = \frac{3}{4}$. LLL redukovana baza je skoro ortogonalna i vektori su poređani po rastućoj normi. Sljedeća teorema govori da je sa LLL redukovanom bazom moguće riješiti apprSVP.

**Teorema 8.1.** *Neka je L rešetka dimenzije n. Svaka LLL redukovana baza $\{v_1, v_2, \dots v_n\}$ ima sljedeća dva svojstava:*

$$\prod_{i=1}^{n} \|v_i\| \leq 2^{\frac{n(n-1)}{4}} detL,$$

$$\|v_j\| \leq 2^{\frac{i-1}{2}} \|v_i^*\| \text{ za sve } 1 \leq j \leq i \leq n.$$

*Nadalje, početni vektor u LLL redukovanoj bazi zadovoljava*

$$\|v_1\| \leq 2^{\frac{n-1}{4}} |detL|^{\frac{1}{n}} \text{ i } \|v_1\| \leq 2^{\frac{n-1}{2}} \min_{0 \neq v \in L} \|v\|.$$

*Dakle, sa LLL redukovanom bazom apprSVP se rješava sa konstantom $2^{\frac{n-1}{2}}$.*

**Teorema 8.2** (LLL algoritam)**.** *Neka je $\{v_1, v_2, \dots v_n\}$ baza za rešetku L. Algoritam (Slika 8.1 ) vraća LLL redukovanu bazu za rešetku L za konačno mnogo koraka. Tačnije, neka je $B = \max \|v_i\|$. Tada se koraci 3-9 algoritma (Slika 8.1 ) izvršavaju za $\mathcal{O}(n^2 \log n + n^2 \log B)$, odnosno algoritam je polinomijalan.*

**Input**: Baza $v = \{v_1, v_2, \dots, v_n\}$ za rešetku $L$ dimenzije $n$
**Output**: *LLL* reducirana baza $v = \{v_1, v_2, \dots, v_n\}$
1 $k = 2$;
2 $v_1^* = v_1$;
3 **while** $k \leq n$ **do**
4  **for** $j = 1, 2, \dots, k-1$ **do**
5  $v_k = v_k - \lfloor \mu_{k,j} \rceil v_j^*$;
6  **end**
7  **if** $\|v_k^*\|^2 \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right)\|v_{k-1}^*\|^2$ **then**
8  $k = k + 1$;
9  **else**
10  zamijenimo $v_{k-1}$ i $v_k$;
11  **end**
12 **end**

Slika 8.1: LLL algoritam

U poglavlju 6 smo vidjeli da ako rešetka $L$ ima ortogonalnu bazu onda je lako riješiti SVP i CVP probleme. LLL algoritmom ne dobijamo ortogonalnu bazu, već bazu u kojoj su vektori *kvazi-ortogonalni.* Stoga, kombinujemo LLL algoritam (Slika 8.1 ) sa Babai algoritmom (Teorema 6.1 ) i dobijamo algoritam koji u potpunosti rješava apprCVP.

**Teorema 8.3** (LLL apprCVP algoritam)**.** *Postoji konstanta $C$ takva da za svaku rešetku $L$ dimenzije $n$, generisanu bazom $\{v_1, v_2, \ldots, v_n\}$ algoritam:*

| |
|---|
| *Iz baze $v_1, v_2, \ldots, v_n$ preko LLL algoritma dobijemo LLL redukovanu bazu.* |
| *Primijenimo Babaijev algoritam na LLL redukovanu bazu.* |

*rješava problem apprCVP sa konstantom $C^n$.*

## 8.3 BKZ - LLL algoritam

BKZ-LLL (block Korkin-Zolotarev) algoritam je jedna od poboljšanih verzija LLL algoritma. Algoritam ima duže vrijeme izvršavanja, ali zato je konačni rezultat bolji. Neka je za proizvoljan niz vektora $v_1, v_2, \ldots$ za $i \geq 1$ i za $v_1^*, v_2^*, \ldots$ vektore dobijene Gram-Šmitovim postupkom definisano preslikavanje

$$\pi : L \to \mathbb{R}^n \text{ i } \pi_i(v) = v - \sum_{j=1}^{i} \frac{v \cdot v_j^*}{\|v_j^*\|^2} v_j^*.$$

Takođe, za $i = 0$ postoji identitet $\pi_0(v) = v$.

**Definicija 8.2.** *Neka je $L$ rešetka dimenzije $n$. Baza $\{v_1, v_2, \ldots, v_n\}$ rešetke $L$ je Korkin - Zoltarev (KZ) redukovana ako važi:*

*(a) $v_1$ je najkraći nenulti vektor u $L$.*

*(b) Za $i = 2, 3, \ldots, n$ vektor $v_i$ je izabran tako da $\pi_{i-1}(v_i)$ bude najkraći nenulti vektor u $\pi_{i-1}(L)$.*

*(c) Za sve $1 \leq i < j \leq n$ imamo $|\pi_{i-1}(v_i) \cdot \pi_{i-1}(v_j)| \leq \frac{1}{2}\|\pi_{i-1}(v_i)\|^2$.*

KZ redukovana baza je generalno bolja nego LLL redukovana baza. Specijalno, prvi vektor u KZ bazi je rješenje SVP problema. Zbog toga su i algoritmi koji traže KZ redukovanu bazu eksponencijalnog trajanja. BKZ varijanta LLL algoritma tamo gdje obični LLL algoritam zamijeni samo dva vektora, vrši zamjenu na cijelom bloku vektora. Dakle, radimo s blokom vektora dužine $\beta$ npr.

$$v_k, v_{k+1}, \ldots, v_{k+\beta-1}$$

i mijenjamo vektore sa KZ redukovanom bazom koja generiše istu podrešetku.

Glava 9

# Primjena LLL algoritma u kriptoanalizi

U zadnje vrijeme metoda redukcije baze rešetke se često koristi za rješavanje mnogih problema u kriptoanalizi i teoriji kodiranja, uključujući i kompromitovanje nekoliko varijanti RSA i DSA (Digital Signature Algoritam), nalaženja malih rješenja modularnih jednačina i sličnih problema. Posljednja klasa problema je kreiranje kripto-sistema na osnovu težine rješavanja SVP. Također, LLL i njegove poboljšane verzije imaju širok spektar primjene u teorijskoj i primijenjenoj matematici, ali mi ćemo se ipak zadržati samo na njegovom značaju u kriptoanalizi.

## 9.1 LLL i kripto-sistemi bazirani na kongruencijama

Podsjetimo se kripto-sistema opisanog u poglavlju 3 . Alisa bira modulo $q$ i tajne parametre $f$ i $g$, njen javni ključ je cijeli broj $h \equiv f^{-1}g \pmod{q}$. Evi je dostupna informacija o vrijednostima $q$ i $h$ i ona želi da nađe privatni ključ $f$ i $g$. Način na koji Eva nalazi privatni ključ je posmatrajući male vektore rešetke $L$ generisane sa

$$v_1 = (1, h) \text{ i } v_2 = (0, q).$$

Vektor $(f, g)$ je iz $L$ i data ograničenja veličina na $f$ i $g$ ukazuju na to da je vektor $(f, g)$ najkraći vektor u $L$.

**Primjer 9.1.** *Neka je dato* $q = 122430513841$ *i* $h = 39245579300$. *Primjenjujemo Gausovu redukciju rešetke za rešetku generisanu sa*

$$(1, 39245579300) \text{ } i \text{ } (0, 122430513841).$$

*Nakon 11 iteracija algoritma nalazimo najkraći bazni vektor*

$$(-231231, -195698) \text{ } i \text{ } (-368222, 217835).$$

*Koristeći invarijantnost u odnosu na promjenu znaka slijedi da je Alisin privatni ključ* $f = 231231$ *i* $g = 195698$.

## 9.2 LLL i problem ranca

Iz poglavlja 3 uočavamo kako možemo jednostavno preformulisati problem ranca (problem zbira podskupa) koji je definisan sa $M = (m_1, \ldots, m_n)$ i $S$ kao problem rešetke koristeći rešetku $L_{M,S}$ čiji su bazni vektori vrste matrice

( 4.1 ). U primjeru ( 5.1 ) smo objasnili zašto je vektor $t \in L_{M,S}$ dužine $\|t\| = \sqrt{n}$ obično upola kraći nego ostali nenulti vektori u $L_{M,S}$.

U sljedećem primjeru koristimo LLL algoritam kako bi riješili problem ranca.

**Primjer 9.2.** *Neka je* $M = (89, 243, 212, 150, 245)$ *i* $S = 546$ *problem zbira podskupa iz primjera ( 4.1 ). Primijenjujemo LLL na rešetku generisanu vrstama matrice*

$$A_{M,S} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 89 \\ 0 & 2 & 0 & 0 & 0 & 243 \\ 0 & 0 & 2 & 0 & 0 & 212 \\ 0 & 0 & 0 & 2 & 0 & 150 \\ 0 & 0 & 0 & 0 & 2 & 245 \\ 1 & 1 & 1 & 1 & 1 & 546 \end{pmatrix}.$$

*Nakon 21 koraka LLL vraća redukovanu bazu*

$$\begin{pmatrix} -1 & 1 & -1 & 1 & -1 & 0 \\ 1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 2 \\ 1 & -1 & -1 & -1 & -1 & 2 \\ -2 & -2 & 4 & 0 & -2 & 0 \\ -6 & -4 & -6 & -6 & 0 & -3 \end{pmatrix}.$$

*Zapisujemo najkraći vektor*

$$(-1, 1, -1, 1, -1, 0)$$

*kao linearnu kombinaciju baznih vektora koji su vrste matrice* $A_{M,S}$

$$(-1, 1, -1, 1, -1, 0) = (-1, 0, -1, 0, -1, 1)A_{M,S}.$$

*Vektor* $(-1, 0, -1, 0, -1, 1)$ *daje rješenje problema ranca*

$$-89 - 212 - 245 + 546 = 0.$$

Glava 10

# Zaključak

Cilj rada je bio definisanje knapsack problema, definisanje problema zbira podskupa kao njegovog specijalnog slučaja, upoznavanje sa terminima rešetke i opisivanja algoritama za nalaženje najkraćeg (najbližeg) vektora u rešetki.
Vidjeli smo, da ako je broj $n$, koji predstavlja dužinu vektora, veliki uglavnom je teško riješiti problem zbira podskupa. Da bi se obezbijedila nemogućnost lakog nalaženja poruke $x$ koristi se zamaskirani problem zbira podskupa. Kripto-sistemi zasnovani na tom problemu su knapsack kripto-sistemi.
Otkrivanje poruke $x$ je moguće poznavanjem algoritama, koji nalaze kratke nenulte vektore u rešetki, od kojih je najpoznatiji LLL.
Pokazalo se da problem često nastaje u raspodjeli resursa, gdje postoje finansijska ograničenja i izučava se u oblastima kao što su kombinatorika, kompjuterske nauke, teorija složenosti algoritama, kriptografija i primijenjena matematika.

# Bibliografija


[1] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, An Introduction to Mathematical Cryptography, Springer Science & Business Media, LLC, 2008.

[2] Paeng, Seong-Hun, Bae Eun Jung, Kil-Chan Ha, A lattice based public key cryptosystem using polynomial representations, Springer Berlin Heidelberg, 2003.

[3] Bernstein, Daniel J., Johannes Buchmann, and Erik Dahmen, eds. Post-quantum cryptography, Springer Science & Business Media, 2009.

[4] Primena kvantne mehanike u kriptografiji, kvantno računarstvo i post-kvantni šifarski sistemi, Slaven Ijačić, Univerzitet Singidunum, Beograd, 2014.

[5] Schneider, Michael, Computing Shortest Lattice Vectors on Special Hardware, 2011.