NTRU kriptosustav

Pribanić, Valentina

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:891100

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Valentina Pribanić

NTRU KRIPTOSUSTAV

Diplomski rad

Voditelj rada:
Andrej Dujella, prof. dr. sc.

Zagreb, srpanj 2015.

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. _____

Sadržaj

Sadržaj **iii**

Uvod **1**

1 Osnovno o vektorskim prostorima **3**
1.1 Definicije i uvodne napomene 3
1.2 Prsteni polinoma 8
1.3 Kongruencije u prstenima 10

2 Rešetke **15**
2.1 Osnovni pojmovi o rešetkama i njihova svojstva 15
2.2 Problemi SVP-a i CVP-a 19
2.3 Teoremi Hermita i Minkowskog 20
2.4 Gaussova heuristika 23

3 Algoritmi redukcije rešetke **27**
3.1 Gaussova redukcija rešetke 27
3.2 LLL algoritam - opis 29
3.3 Rješavanje apprCVP pomoću LLL 33
3.4 BKZ-LLL algoritam 34

4 Osnovno o kriptografiji **37**
4.1 Kratka povijest kriptografije 37
4.2 Osnovni pojmovi u kriptografiji 38
4.3 Simetrični kriptosustavi 39
4.4 Asimetrični kriptosustavi 39
4.5 Usporedba simetričnog i asimetričnog kriptosustava 40
4.6 Vrste napada na kriptosustav 40

5 NTRU kriptosustav **41**
5.1 NTRUEncrypt - opis algoritma 41

5.1.1 Kreiranje ključeva 42
5.1.2 Šifriranje 42
5.1.3 Dešifriranje 43
5.1.4 Pogrešno dešifriranje 45
5.2 Sheme kriptiranja 46
5.2.1 NAEP 46
5.2.2 NAEP:SVES3 47

6 Sigurnost NTRU kriptosustava 49
6.1 Matematička pozadina NTRUEncrypt kriptosustava 49
6.2 Napadi bez korištenja strukture rešetke 49
6.2.1 Napad ”grubom silom” 49
6.2.2 Algoritam kolizije 50
6.3 NTRUEncrypt rešetka 51
6.4 Napadi koji koriste strukturu rešetke 53
6.4.1 LLL algoritam i NTRUEncrypt rešetka 53
6.4.2 Hibridni napad 54

7 Digitalni potpisi i NTRUSign 57
7.1 Digitalni potpisi pomoću rešetki 58
7.2 NTRUSign 59
7.2.1 Opis algoritma 60
7.3 Pronalazak komplementarne baze 61
7.4 Napad na NTRUSign 63

Bibliografija 65

Uvod

U ovom radu opisan je *NTRU* kriptosustav, jedan od novijih kriptosustava s javnim ključem. Unutar prva dva poglavlja, definiraju se osnovni pojmovi vezani uz vektorske prostore i teoriju rešetaka.
Treće poglavlje opisuje algoritme redukcije rešetki, među kojima posebno ističemo *LLL* algoritam i njegovo blokovno poboljšanje *BKZ-LLL* algoritam, kao do sada najbolje algoritme za napad na NTRU kriptosustav.
Četvrto poglavlje daje kratak pregled osnovnih pojmova u kriptografiji. Definiraju se simetrični i antisimetrični kriptosustavi, te se daje njihova kratka usporedba.
Posljednja tri poglavlja su temelj ovog rada. U petom poglavlju opisujemo *NTRUEncrypt*, inačicu NTRU kriptosustava koja se koristi za šifriranje. Opisan je sam algoritam, šifriranje, dešifriranje te posebne sheme kriptiranja koje pomažu u obrani od napada odabranim šifratom. Neki od mogućih napada na NTRUEncrypt opisani su u šestom poglavlju. Osnovna podjela napada je prema tome koriste li strukturu rešetke ili ne.
Rad završava opisom *NTRUSign*-a, inačice NTRU kriptosustava koja se koristi za digitalne potpise. Nakon kratkog uvoda u problem digitalnih potpisa, opisan je sam kriptosustav, te mogući napad.

Poglavlje 1

Osnovno o vektorskim prostorima

1.1 Definicije i uvodne napomene

Vektorski prostor promatramo kao skup na kojem je definirana binarna operacija zbrajanja i operacija množenja skalarom.

Napomena 1.1.1. *Binarne operacije zbrajanja* $+\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ *i množenja* $\cdot\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ *imaju sljedeća svojstva:*

(a) $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma,\ \forall\alpha,\beta,\gamma\in\mathbb{R}$.

(b) postoji $0\in\mathbb{R}$ *sa svojstvom* $\alpha+0=0+\alpha=\alpha,\ \forall\alpha\in\mathbb{R}$.

(c) za svaki $\alpha\in\mathbb{R}$ *postoji* $-\alpha\in\mathbb{R}$ *tako da je* $\alpha+(-\alpha)=-\alpha+\alpha=0$.

(d) $\alpha+\beta=\beta+\alpha,\ \forall\alpha,\beta\in\mathbb{R}$.

(e) $\alpha(\beta\gamma)=(\alpha\beta)\gamma,\ \forall\alpha,\beta,\gamma\in\mathbb{R}$.

(f) postoji $1\in\mathbb{R}$ *sa svojstvom* $\alpha\cdot 1=1\cdot\alpha=\alpha,\ \forall\alpha\in\mathbb{R}$.

(g) za svaki $\alpha\in\mathbb{R},\ \alpha\neq 0$, *postoji* $\alpha^{-1}\in\mathbb{R}$ *tako da je* $\alpha\alpha^{-1}=\alpha^{-1}\alpha=1$.

(h) $\alpha\beta=\beta\alpha,\ \forall\alpha,\beta\in\mathbb{R}$.

(i) $\alpha(\beta+\gamma)=\alpha\beta+\alpha\gamma,\ \forall\alpha,\beta,\gamma\in\mathbb{R}$.

Ako imamo skup $\mathbb{F}$, na kojem su zadane operacije $+\colon \mathbb{F}\times\mathbb{F}\to\mathbb{F}$ i $\cdot\colon \mathbb{F}\times\mathbb{F}\to\mathbb{F}$ koje zadovoljavaju svojstva iz (1.1.1), kažemo da je $\mathbb{F}$ polje. Neki od primjera polja su skupovi $\mathbb{Q},\mathbb{C}$ i $\mathbb{R}$. Polja su nam potrebna da bi mogli definirati operacije unutar vektorskih prostora.

Definicija 1.1.2. *Neka je V neprazan skup na kojem su zadane binarna operacija zbrajanja $+\colon V \times V \to V$ i operacija množenja skalarima iz polja $\mathbb{F}, \cdot\colon \mathbb{F} \times V \to V$. Uređena trojka $(V, +, \cdot)$ je* vektorski prostor *nad poljem $\mathbb{F}$ ako vrijedi:*

(a) $a + (b + c) = (a + b) + c, \ \forall a, b, c \in V.$

(b) postoji $0 \in V$ *sa svojstvom* $a + 0 = 0 + a = a, \ \forall a \in V.$

(c) za svaki $a \in V$ *postoji* $-a \in V$ *tako da je* $a + (-a) = -a + a = 0.$

(d) $a + b = b + a, \ \forall a, b \in V.$

(e) $\alpha(\beta a) = (\alpha\beta)a, \ \forall \alpha, \beta \in \mathbb{F}, \ \forall a \in V.$

(f) $(\alpha + \beta)a = \alpha a + \beta a, \ \forall \alpha, \beta \in \mathbb{F}, \ \forall a \in V.$

(g) $\alpha(a + b) = \alpha a + \alpha b, \ \forall \alpha \in \mathbb{F}, \ \forall a, b \in V.$

(h) $1 \cdot a = a \cdot 1 = a, \ \forall a \in V.$

Elementi vektorskog skupa nazivaju se vektori. Unutar ovog rada nećemo koristiti definiciju vektorskih prostora u njenoj punoj općenitosti. Promatrat ćemo skupove koji su sadržani u vektorskom prostoru $\mathbb{R}^n$, za neki $n \in \mathbb{N}$.

Definicija 1.1.3. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i $M \subset V, M \neq \emptyset$. Ako je i $(M, +, \cdot)$ vektorski prostor nad $\mathbb{F}$ uz iste operacije iz V, kažemo da je M* potprostor *od V.*

Za provjeru je li neki podskup doista potprostor dovoljno je provjeriti je li zatvoren na operacije zbrajanja i množenja skalarom.

Propozicija 1.1.4. *Neka je V vektorski prostor nad $\mathbb{F}$ i M neprazan podskup od V. Tada je M potprostor ako i samo ako vrijedi*

$$\alpha a + \beta b \in M, \ \forall \alpha, \beta \in \mathbb{F}, \ \forall a, b \in M.$$

Definicija 1.1.5. *Neka je V vektorski prostor nad $\mathbb{F}$. Izraz oblika*

$$\alpha_1 v_1 + \cdots + \alpha_k v_k, \quad v_1, \ldots, v_k \in V, \ \alpha_1, \ldots, \alpha_k \in \mathbb{F}, \ k \in \mathbb{N},$$

naziva se linearna kombinacija *vektora $v_1, \ldots, v_k$ s koeficijentima $\alpha_1, \ldots, \alpha_k$.*
Skup svih ovakvih linearnih kombinacija nazivamo linearna ljuska *vektora $\{v_1, \ldots, v_k\}$ i označavamo sa $[\{v_1, \ldots, v_k\}]$. Za skup $S \subset V$ za koji vrijedi $[S] = V$ kažemo da je* sustav izvodnica *za V.*

Definicija 1.1.6. *Neka je $k \in \mathbb{N}$. Skup vektora $\{v_1, \ldots, v_k\} \in V$ je* linearno nezavisan *ako za skalare $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$ vrijedi*

$$\alpha_1 v_1 + \cdots + \alpha_k v_k = 0 \Rightarrow \alpha_1 = 0, \ldots, \alpha_k = 0.$$

Ako je za neki $i \in \{1, \ldots, k\}$, $\alpha_i \neq 0$, skup je linearno zavisan.

Dakle, skup je linearno nezavisan ako je jedini način na koji neka linearna kombinacija unutar tog skupa može biti jednaka nuli, tako da su svi skalari jednaki nuli.

Definicija 1.1.7. *Konačan skup $\mathcal{B} = \{v_1, \ldots, v_n\}$, $n \in \mathbb{N}$, u vektorskom prostoru V je* baza *za V ako je linearno nezavisan i linearna ljuska ovog skupa je jednaka cijelom prostoru V.*

Definicija baze za vektorski prostor V je ekvivalentna tvrdnji da svaki vektor $w \in V$ možemo zapisati kao linearnu kombinaciju vektora baze

$$w = \alpha_1 v_1 + \cdots + \alpha_n v_n,$$

za jedinstveni izbor skalara $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$. Broj elemenata baze nazivamo dimenzija prostora V.

Neka od osnovnih svojstava baze su dana sljedećom propozicijom. Za polje skalara uzimamo skup realnih brojeva $\mathbb{R}$ i gledamo potprostore od $\mathbb{R}^n, n \in \mathbb{N}$.

Propozicija 1.1.8. *Neka je $V \subset \mathbb{R}^n$ vektorski prostor.*

(a) Postoji baza za V.

(b) Bilo koje dvije različite baze od V imaju isti broj elemenata.

(c) Neka je $v_1, v_2, \ldots, v_k$ baza za prostor V i $w_1, w_2, \ldots, w_k$ skup vektora iz V. Svaki od vektora w_i tada možemo zapisati kao linearnu kombinaciju vektora baze,

$$\begin{aligned} w_1 &= \alpha_{11} v_1 + \alpha_{12} v_2 + \cdots + \alpha_{1k} v_k, \\ w_2 &= \alpha_{21} v_1 + \alpha_{22} v_2 + \cdots + \alpha_{2k} v_k, \\ &\vdots \\ w_k &= \alpha_{k1} v_1 + \alpha_{k2} v_2 + \cdots + \alpha_{kk} v_k. \end{aligned}$$

Tada je skup $w_1, w_2, \ldots, w_k$ baza za V ako i samo ako je determinanta matrice

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1k} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{k1} & \alpha_{k2} & \cdots & \alpha_{kk} \end{pmatrix}$$

različita od nule.

Nadalje, opisujemo kako u $\mathbb{R}^n$ mjerimo duljinu vektora, normu vektora te ortogonalnost baze. Proizvoljan vektor $v \in \mathbb{R}^n$ možemo zapisati preko njegovih koordinata kao $v = (v_1, v_2, \ldots, v_n)$.

Definicija 1.1.9. *Neka su* $v = (v_1, v_2, \ldots, v_k)$, $w = (w_1, w_2, \ldots, w_k) \in V \subset \mathbb{R}^n$. Skalarni produkt *vektora definiramo kao*

$$v \cdot w = v_1 w_1 + v_2 w_2 + \cdots + v_k w_k.$$

Ako je $v \cdot w = 0$, *kažemo da su vektori* v *i* w ortogonalni.

Pomoću skalarnog produkta sada jednostavno definiramo Euklidsku normu vektora:

Definicija 1.1.10. Euklidska norma *vektora* $v = (v_1, v_2, \ldots, v_k) \in V \subset \mathbb{R}^n$, *u oznaci* $\|v\|$, *je*

$$\|v\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_k^2}.$$

Euklidska norma i skalarni produkt su povezani sljedećom relacijom:

$$v \cdot v = \|v\|^2.$$

Napomena 1.1.11. *Svojstva norme su:*

(a) $\|v\| \geq 0, \ \forall v \in V.$

(b) $\|v\| = 0 \Leftrightarrow \ v = 0.$

(c) $\|\alpha v\| = |\alpha| \, \|v\|, \forall \alpha \in \mathbb{F}, \forall v \in V.$

(d) $\|v + w\| \leq \|v\| + \|w\|, \forall v, w \in V.$

Svaka funkcija na vektorskom prostoru koja zadovoljava ove uvjete je norma. Dakle, norma ne mora nužno biti zadana preko skalarnog produkta.

Propozicija 1.1.12. *(Cauchy-Schwarz nejednakost) Neka su* $v, w \in V \subset \mathbb{R}^n$ *(na vektorskom prostoru* V *je definiran skalaran produkt). Vrijedi nejednakost*

$$|v \cdot w| \leq \|v\| \, \|w\|. \tag{1.1}$$

Jednakost vrijedi ako i samo ako su vektori linearno zavisni.

Dokaz. Ako je $v = 0$ ili $w = 0$ nemamo što dokazivati. Neka su v, w netrivijalni vektori i $\lambda \in \mathbb{R}$ skalar. Imamo

$$\begin{aligned} 0 \leq \|v - \lambda w\|^2 &= (v - \lambda w) \cdot (v - \lambda w) \\ &= v \cdot v - 2\lambda v \cdot w + \lambda^2 w \cdot w \\ &= \|v\|^2 - 2\lambda v \cdot w + \lambda^2 \|w\|^2. \end{aligned}$$

Uvrstimo $\lambda = \frac{v \cdot w}{\|w\|^2}$. Kako je $w \neq 0$, λ je dobro definiran. Dobivamo:

$$0 \leq \|v\|^2 - 2\frac{v \cdot w}{\|w\|^2} v \cdot w + \left(\frac{v \cdot w}{\|w\|^2}\right)^2 \|w\|^2 = \|v\|^2 - \frac{v \cdot w}{\|w\|^2} v \cdot w.$$

Množenjem ove nejednakosti sa $\|w\|^2$ imamo:

$$0 \leq \|v\|^2 \|w\|^2 - (v \cdot w)^2.$$

Preostaje izvaditi drugi korijen iz ove nejednakosti.
Ako je $w = \alpha v$, za neki skalar α, očito se dobije jednakost. Ako pak vrijedi jednakost, istim računom se pokaže $w = \lambda v$. □

Definicija 1.1.13. *Baza $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ prostora V je* ortogonalna *baza ako vrijedi,*

$$v_i \cdot v_j = 0 \quad \textit{za sve} \quad i \neq j.$$

Dodatno, ako je $\|v_i\| = 1, \forall i$, kažemo da je baza ortonormirana.

Algoritam koji bilo koju bazu vektorskog prostora prevodi u ortonormiranu bazu naziva se Gram-Schmidtov algoritam. Sljedeći teorem osigurava egzistenciju ortonormirane baze u konačnodimenzionalnom vektorskom prostoru.

Teorem 1.1.14. *(Gramm-Schmidtov postupak ortogonalizacije)*
Neka je V vektorski prostor na kojem je definiran skalarni produkt. Neka je dan linearno nezavisan skup $\{v_1, \ldots, v_k\}$, $k \in \mathbb{N}$ u V. Tada postoji ortonormiran skup $\{v_1^*, \ldots, v_k^*\}$ u V takav da je $[\{v_1, \ldots, v_j\}] = [\{v_1^*, \ldots, v_j^*\}], \forall j = 1, \ldots, k$.

Dokaz. Konstrukcija skupa $\{v_1^*, \ldots, v_k^*\}$ se provodi induktivno. Za bazu indukcije stavimo $v_1^* = \frac{1}{\|v_1\|} v_1$, što je dobro definirano jer je $v_1 \neq 0$. Očito su v_1 i v_1^* kolinearni pa razapinju isti potprostor. Pretpostavimo da je nađen ortonormiran skup $\{v_1^*, \ldots, v_j^*\}$ takav da vrijedi $[\{v_1, \ldots, v_j\}] = [\{v_1^*, \ldots, v_j^*\}]$ i konstruiramo v_{j+1}^*.
Uvedimo pomoćni vektor $f_{j+1} = v_{j+1} - \sum_{i=1}^{j} (v_{j+1} \cdot v_i^*) v_i^*$. Iz definicije vektora f_{j+1} vidimo da je okomit na vektore v_i^*, $\forall i = 1, \ldots, j$. Nadalje, vrijedi $[\{v_1^*, \ldots, v_j^*, f_{j+1}\}] = [\{v_1, \ldots, v_j, v_{j+1}\}]$. Trebamo pokazati da generatori s lijeve strane jednakosti pripadaju potprostoru s desne strane i obratno. Sada je $v_1^*, \ldots, v_j^* \in [\{v_1, \ldots, v_j, v_{j+1}\}]$ po pretpostavci

indukcije, a $f_{j+1} \in [\{v_1, \ldots, v_j, v_{j+1}\}]$ po definiciji vektora f_{j+1}. Obratno se dobije uz analogne argumente. Uočimo da je $v_{j+1} = f_{j+1} + \sum_{i=1}^{j}(v_{j+1} \cdot v_i^*)v_i^*$.
Skup $\{v_1^*, \ldots, v_j^*, f_{j+1}\}$ ima gotovo sva tražena svojstva. Preostaje provjeriti normu vektora f_{j+1}. No, za svaki skalar $\lambda \neq 0$ možemo uzeti vektor λf_{j+1} umjesto f_{j+1}. Neka je $\lambda = \|f_{j+1}\|^{-1}$ i definiramo $v_{j+1} = \frac{1}{\|f_{j+1}\|} f_{j+1}$. Uočimo da f_{j+1} ne može nikad biti nul-vektor, jer bi tada $v_{j+1} = \sum_{i=1}^{j}(v_{j+1} \cdot v_i^*)v_i^* \in [\{v_1^*, \ldots, v_j^*\}] = [\{v_1, \ldots, v_j\}]$, a to se kosi sa nezavisnošću skupa $\{v_1, \ldots, v_k\}$. □

1.2 Prsteni polinoma

Za potrebe NTRU kriptosustava potrebni su nam prsteni polinoma i osnovne operacije u njima. Neka je $\mathbb{F}$ polje. Izraz oblika $a(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n, n \in \mathbb{N}$, gdje su koeficijenti a_i iz $\mathbb{F}$ nazivamo polinom u varijabli x. Ako je $a_n \neq 0$, kažemo da je n stupanj polinoma $a(x)$ (oznaka $deg(a) = n$), a a_n njegov vodeći koeficijent.

Definicija 1.2.1. *Neprazan skup $R = R(+, \cdot)$ zovemo* prsten *ukoliko za operacije zbrajanja $+ \colon R \times R \to R$ i množenja $\cdot \colon R \times R \to R$ vrijedi:*

(a) Postoji element $0 \in R$ takav da vrijedi $a + 0 = 0 + a, \ \forall a \in R$.

(b) Za svaki $a \in R$ postoji $-a \in R$ takav da vrijedi $a + (-a) = -a + a = 0$.

(c) $a + (b + c) = (a + b) + c, \ \forall a, b, c \in R$.

(d) $a + b = b + a, \ \forall a, b \in R$.

(e) Postoji element $1 \in R$ takav da vrijedi $1 \cdot a = a \cdot 1 = a, \ \forall a \in R$.

(f) $a \cdot (b \cdot c) = (a \cdot b) \cdot c, \ \forall a, b, c \in R$.

(g) $a \cdot b = b \cdot a, \ \forall a, b \in R$.

(h) $a \cdot (b + c) = a \cdot b + a \cdot c, \ \forall a, b, c \in R$.

Koeficijente polinoma možemo uzimati iz proizvoljnog prstena. Skup svih takvih polinoma nazivamo prsten polinoma nad R, i označavamo s $R[x]$, tj.

$$R[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_nx^n : n \geq 0, \ a_0, a_1, \ldots, a_n \in R\}.$$

Posebno, za $\mathbb{F}$ polje, imamo prsten polinoma $\mathbb{F}[x]$.
Pojmovi djeljivosti, zajedničke mjere, Euklidova algoritma se mogu generalizirati na prstene polinoma.

Teorem 1.2.2. *(Teorem o dijeljenju s ostatkom) Neka je $\mathbb{F}$ polje te a i b polinomi iz $\mathbb{F}$, $b \neq 0$. Tada postoje polinomi k i r, gdje je $r = 0$ ili $deg(r) < deg(b)$, tako da vrijedi*

$$a = b \cdot k + r.$$

Dokaz. Započinjemo dokaz s bilo koja dva polinoma k i r za koje vrijedi

$$a = b \cdot k + r.$$

Npr. možemo uzeti $k = 0$ i $r = a$. Ako je $deg(r) < deg(b)$, gotovi smo. U suprotnom definiramo

$$b = b_0 + b_1 x + \cdots + b_d x^d \quad \text{i} \quad r = r_0 + r_1 x + \cdots + r_l x^l,$$

gdje je $b_d \neq 0, r_l \neq 0$ i $l \geq d$. Sada jednakost $a = b \cdot k + r$ zapišemo kao

$$a = b \cdot \left(k + \frac{r_l}{b_d} x^{l-d}\right) + \left(r - \frac{r_l}{b_d} x^{l-d} \cdot b\right) = b \cdot k' + r'.$$

Uočimo da je $deg(r') < deg(r)$. Ako je $deg(r') < deg(b)$, gotovi smo, u suprotnom ponovimo proces. Ovaj postupak možemo ponavljati sve dok je $deg(r) \geq deg(b)$. U svakom koraku smanjujemo stupanj od polinoma r, tako da ćemo doći do polinoma čiji je stupanj striktno manji od stupnja polinoma b. □

Za polinom $d \in \mathbb{F}[x]$ kažemo da je zajednički djelitelj dvaju polinoma $a, b \in \mathbb{F}[x]$ ako dijeli oba polinoma bez ostatka. Najveći među svim takvim polinomima nazivamo najveći zajednički djelitelj. Oznaka je $\gcd(a, b)$.

Napomena 1.2.3. *Najveći zajednički djelitelj tražimo preko Euklidovog algoritma za polinome:*

$$\begin{array}{lcll}
a & = & b \cdot k_1 + r_2 & 0 \leq deg(r_2) < deg(b) \\
b & = & r_2 \cdot k_2 + r_3 & 0 \leq deg(r_3) < deg(r_2) \\
r_2 & = & r_3 \cdot k_3 + r_4 & 0 \leq deg(r_4) < deg(r_3) \\
 & \vdots & & \vdots \\
r_{t-2} & = & r_{t-1} \cdot k_{t-1} + r_t & 0 \leq deg(r_t) < deg(r_{t-1}) \\
r_{t-1} & = & r_t \cdot k_t &
\end{array}$$

Tada je $d = r_t = \gcd(a, b)$.

Propozicija 1.2.4. *(Prošireni Euklidov algoritam za $\mathbb{F}[x]$) Neka je $\mathbb{F}$ polje te a i b polinomi iz $\mathbb{F}[x]$. Tada postoji najveći zajednički djelitelj d od a i b, i polinomi $u, v \in \mathbb{F}[x]$ za koje vrijedi*

$$a \cdot u + b \cdot v = d.$$

Dokaz. Najveći zajednički djelitelj polinoma a i b, $gcd(a, b)$ tražimo višestrukom primjenom teorema (1.2.2), uz pomoć algoritma opisanog sa (1.2.3). Nakon izračuna $gcd(a, b)$ supstitucijom unatrag u jednadžbama (1.2.3) dolazimo do u i v. □

1.3 Kongruencije u prstenima

Teoriju kongruencija za cijele brojeve uveo je Carl Friedrich Gauss 1801. godine u svom djelu *Disquisitiones Arithmeticae*. Također je uveo i oznaku za kongruenciju koju danas rabimo. Svojstva kongruencije koja vrijede za cijele brojeve se prenose i na prstene.

Definicija 1.3.1. *Neka je R prsten i $0 \neq m \in R$. Dva elementa $a, b \in R$ su* kongruentna modulo *m ako m dijeli njihovu razliku $a - b$ i pišemo $a \equiv b \pmod{m}$. U protivnom, kažemo da a nije konguentan b modulo m i pišemo $a \not\equiv b \pmod{m}$.*

Propozicija 1.3.2. *Neka je R prsten i $0 \neq m \in R$. Ako je*

$$a_1 \equiv b_1 \pmod{m} \quad i \quad a_2 \equiv b_2 \pmod{m}$$

onda je

$$a_1 \pm a_2 \equiv b_1 \pm b_2 \pmod{m} \quad i \quad a_1 a_2 \equiv b_1 b_2 \pmod{m}.$$

Pomoću kongruencija možemo definirati nove prstene iz već poznatih.

Definicija 1.3.3. *Neka je R prsten i $0 \neq m \in R$. Za svaki $a \in R$ možemo definirati skup $\overline{a} = \{a' \in R : a' \equiv a \pmod{m}\}$. Skup $\overline{a}$ nazivamo* klasa kongruencije *od a. Skup svih klasa kongruencije označavamo s $R/(m)$ ili R/mR.*

Klase kongruencije zbrajamo i množimo na uobičajeni način:

$$\overline{a} + \overline{b} = \overline{a + b} \quad \text{i} \quad \overline{a} \cdot \overline{b} = \overline{a \cdot b}.$$

Dakle, $R/(m) = R/mR = \{\bar{a} : a \in R\}$ i ovaj skup nazivamo kvocijentni prsten od R po m. Jedan od primjera ovakvog prstena je prsten ostataka modulo m u skupu $\mathbb{Z}$, tj. $\mathbb{Z}/m\mathbb{Z} = \{0, 1, 2, \ldots, m-1\}$. Primijetimo da nakon izvršene operacije u ovom prstenu uvijek moramo dodatno rezultat podijeliti sa m da bi dobili element skupa $\mathbb{Z}/m\mathbb{Z}$.
Osim standardnih operacija u prstenu, možemo tražiti i multiplikativni inverz elementa. Za nenul element $a \in R$ kažemo da ima multiplikativni inverz $b \in R$ ako vrijedi

$$ab \equiv 1 \pmod{m}.$$

Posebno, za prsten $\mathbb{Z}/m\mathbb{Z}$, ako je m prost broj, može se pokazati da svaki nenul element ima inverz.

Definicija 1.3.4. *Neka je N prirodan broj. Kvocijentni prsten*

$$R = \frac{\mathbb{Z}[x]}{(x^N - 1)}$$

nazivamo konvolucijski prsten polinoma (ranga N). Anologno, kvocijentni prsten

$$R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[x]}{(x^N - 1)}$$

nazivamo konvolucijski prsten polinoma (ranga N i modulo q).

Oblik konvolucijskih prstena R i R_q pojednostavljuje operacije. Naime, polinom $x^N - 1$ ima jednostavan oblik, tako da kada radimo operacije modulo $(x^N - 1)$, zapravo zahtijevamo da x^N bude jednak 1. Tako, npr. za x^k, zapišemo $k = iN + j, 0 \leq j < N$ te imamo

$$x^k = x^{iN+j} = (x^N)^i \cdot x^j = 1^i \cdot x^j = x^j.$$

Zapravo, dovoljno je s eksponentima napraviti operaciju modulo N.
Polinom $a(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ iz R (ili R_q) možemo također zapisati preko njegovog vektora koeficijenata $(a_0, a_1, a_2, \ldots, a_n) \in \mathbb{Z}^N$. Zbrajanje dva ovakva polinoma je standardno po koordinatama, odnosno zbroju $a(x) + b(x)$, $a, b \in R$ odgovara vektor koeficijenata $(a_0 + b_0, a_1 + b_1, a_2 + b_2, \ldots, a_{N-1} + b_{N-1})$. Za množenje koristimo sljedeću propoziciju.

Propozicija 1.3.5. *Umnožak dva polinoma $a(x)$, $b(x) \in R$ je dan formulom*

$$a(x) \circledast b(x) = c(x) \quad \textit{gdje je} \quad c_k = \sum_{i+j \equiv k \pmod{N}} a_i b_{k-i}, \tag{1.2}$$

gdje suma za c_k prolazi po svima i, j koji su između 0 i $N - 1$ te zadovoljavaju uvjet $i + j \equiv k \pmod{N}$. Produkt dva polinoma $a(x)$, $b(x) \in R_q$ dan je istom formulom, samo što dodatno treba reducirati vrijednost c_k modulo q.

Dokaz. Izračunamo produkt polinoma $a(x)$ i $b(x)$ te potom upotrijebimo da je $x^N = 1$.

$$\begin{aligned}
a(x) \circledast b(x) &= \left(\sum_{i=0}^{N-1} a_i x^i\right)\left(\sum_{j=0}^{N-1} b_j x^j\right) \\
&= \sum_{k=0}^{2N-2} \left(\sum_{i+j=k} a_i b_j\right) x^k \\
&= \sum_{k=0}^{N-1} \left(\sum_{i+j=k} a_i b_j\right) x^k + \sum_{k=N}^{2N-2} \left(\sum_{i+j=k} a_i b_j\right) x^{k-N} \\
&= \sum_{k=0}^{N-1} \left(\sum_{i+j=k} a_i b_j\right) x^k + \sum_{k=0}^{N-2} \left(\sum_{i+j=k+N} a_i b_j\right) x^k \\
&= \sum_{k=0}^{N-1} \left(\sum_{i+j \equiv k \pmod{N}} a_i b_j\right) x^k.
\end{aligned}$$

□

Napomena 1.3.6. *Za dva vektora* $a = (a_0, a_1, \ldots, a_n)$ *i* $b = (b_0, b_1, \ldots, b_n)$ *definiramo konvolucijsko množenje (ili cikličku konvoluciju) kao*

$$(a_0, a_1, \ldots, a_n) \circledast (b_0, b_1, \ldots, b_n) = (c_0, c_1, \ldots, c_n),$$

gdje su koeficijenti c_i *dani formulom (1.2).*

Polinome iz prstena R redukcijom njegovih koeficijenata modulo q, jednostavno prikažemo kao elemente prstena R_q. To preslikavanje iz R u R_q ima sljedeća svojstva:

$$\begin{aligned} (a(x) + b(x)) \bmod q &= (a(x) \bmod q) + (b(x) \bmod q)\,, \\ (a(x) \circledast b(x)) \bmod q &= (a(x) \bmod q) \circledast (b(x) \bmod q)\,. \end{aligned}$$

Za preslikavanje u drugom smjeru, tj. iz prstena R_q u prsten R koristimo centriranje koeficijenata.

Definicija 1.3.7. *Neka je* $a(x) \in R_q$. Centriranjem *polinoma* $a(x)$ *dobivamo polinom* $a'(x)$ *iz prstena* R *koji zadovoljava*

$$a'(x) \bmod q = a(x)$$

i čiji koeficijenti su izabrani iz intervala

$$-\frac{q}{2} < a'_i \leq \frac{q}{2}.$$

Mali broj polinoma iz R ima multiplikativni inverz, za razliku od prstena R_q. Multiplikativni inverz tražimo Euklidovim algoritmom.

Propozicija 1.3.8. *Neka je* q *prost broj. Tada* $a(x) \in R_q$ *ima multiplikativni inverz ako i samo ako*

$$\gcd(a(x), x^N - 1) = 1 \quad u \quad (\mathbb{Z}/q\mathbb{Z})[x]. \tag{1.3}$$

Ako vrijedi (1.3)*, onda inverz* $a(x)^{-1} \in R_q$ *možemo izračunati proširenim Euklidovim algoritmom. Prema propoziciji (1.2.4) možemo pronaći polinome* $u(x),\ v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$ *takve da vrijedi*

$$a(x)\, u(x) + (x^N - 1)\, v(x) = 1,$$

pa je $a(x)^{-1} = u(x) \in R_q$.

Dokaz. Po propoziciji (1.2.4) možemo pronaći polinome $u(x), v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$ takve da vrijedi

$$a(x)\, u(x) + (x^N - 1)\, v(x) = \gcd(a(x), x^N - 1).$$

Ako je $\gcd(a(x), x^N - 1) = 1$, onda reduciranjem modulo $x^N - 1$ dobivamo

$$a(x) \circledast u(x) = 1 \quad \text{u} \quad R_q.$$

Obratno, ako je $a(x)$ invertibilan u R_q, možemo pronaći polinom takav da je $a(x) \circledast u(x) = 1$ u R_q. Po definiciji prstena R_q to znači

$$a(x)\, u(x) \equiv 1 \pmod{(x^N - 1)},$$

pa po definiciji kongruencije postoji polinom $v(x)$ takav da vrijedi

$$a(x)\, u(x) - 1 = (x^N - 1)\, v(x) \quad \text{u} \quad (\mathbb{Z}/q\mathbb{Z})[x].$$

□

Poglavlje 2

Rešetke

2.1 Osnovni pojmovi o rešetkama i njihova svojstva

Ovo poglavlje započinjemo definicijom rešetki i navođenjem svojstava koja će se pokazati bitnima u određivanju najkraćeg vektora u rešetki.

Definicija 2.1.1. *Neka je $v_1, \ldots, v_d \in \mathbb{R}^n$ skup linearno nezavisnih vektora. Skup L generiran svim linearnim kombinacijama vektora $v_1, \ldots, v_d$ s koeficijentima iz $\mathbb{Z}$ nazivamo* rešetka*. Kraće zapisano:*

$$L = \{a_1 v_1 + \cdots + a_d v_d : a_1, \ldots, a_d \in \mathbb{Z}\}.$$

Alternativna definicija rešetke:

Definicija 2.1.2. Rešetka $L \subset \mathbb{R}^n$ *je diskretna aditivna podgrupa od $\mathbb{R}^n$. Odnosno, L je zatvorena na operacije zbrajanja i oduzimanja te također, postoji $\epsilon > 0$ takav da za svaki $v \in L$,*

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

Iz (2.1) vidimo da za svaki vektor $v \in L$, kugla oko v radijusa ϵ ne sadrži niti jedan drugi vektor iz rešetke osim samog v. Ove dvije definicije su ekvivalentne. Neke od diskretnih aditivnih grupa su nul-rešetka $\{0\}$ i cjelobrojna rešetka $(\mathbb{Z}^n, +)$. Štoviše, iz (2.1.2) slijedi da je svaka podgrupa rešetke opet rešetka, pa je tako i svaka podgrupa od $(\mathbb{Z}^n, +)$ rešetka.
Baza za rešetku L je svaki skup izvodnica koji generira L. Svaki vektor rešetke se može na jedinstven način prikazati preko vektora baze. Dimenzija rešetke je broj vektora baze. Može se pokazati da svaka rešetka ima barem jednu bazu, a ako ih ima više, sljedeći teorem, koji je analogan teoremu (1.1.8c) iskazuje njihovu vezu:

Teorem 2.1.3. *Neka je skup $\{v_1, \ldots, v_d\}$ baza za rešetku $L \subset \mathbb{R}^n$ i neka su vektori $w_1, \ldots, w_d$ iz L. Tada postoji jedinstvena matrica $A = (a_{i,j})_{1 \le i,j \le d}$ dimenzije $d \times d$ takva da je*

$w_i = \sum_{j=1}^{d} a_{i,j} v_j$ za svaki $1 \leq i \leq d$. Nadalje, skup $\{w_1, \ldots, w_d\}$ je baza za rešetku L ako i samo ako je $\det(A) = \pm 1$.

Dokaz. (skica dokaza) Svaki od vektora w_j možemo zapisati preko vektora baze:

$$\begin{aligned} w_1 &= \alpha_{11} v_1 + \alpha_{12} v_2 + \cdots + \alpha_{1d} v_d, \\ w_2 &= \alpha_{21} v_1 + \alpha_{22} v_2 + \cdots + \alpha_{2d} v_d, \\ &\vdots \\ w_d &= \alpha_{d1} v_1 + \alpha_{d2} v_2 + \cdots + \alpha_{dd} v_d. \end{aligned}$$

Kako su vektori elementi rešetke, svi koeficijenti u ovom raspisu su cjelobrojni. Da bismo sada izrazili vektore v_i preko vektora w_j potrebno je pronaći inverz matrice

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1d} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{d1} & \alpha_{d2} & \cdots & \alpha_{dd} \end{pmatrix}.$$

Koeficijenti u raspisu vektora v_i također moraju biti cjelobrojni, dakle, A^{-1} mora imati cjelobrojne elemente. Imamo

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

gdje su $\det(A),\ \det(A^{-1}) \in \mathbb{Z}$ pa je $\det(A) = \pm 1$.
Za obrat, definiramo adjunktu matrice A kao matricu B sa elementima

$$b_{ij} = (-1)^{i+j} \det(A_{ji}),$$

gdje je A_{ji} podmatrica matrice A dobivena izbacivanjem j-tog retka i i-tog stupca. Matrica B ima cjelobrojne elemente jer su oni jednaki determinantama podmatrica matrice A koja ima cjelobrojne elemente. Kako je

$$A^{-1} = \frac{1}{\det A} \cdot B,$$

i $\det(A) = \pm 1$ vidimo da i A^{-1} ima cjelobrojne elemente. □

Po prethodnom teoremu veza između dvije baze za rešetku L je matrica sa cjelobrojnim koeficijentima i determinantom koja iznosi ± 1.

Definicija 2.1.4. *Neka je L rešetka dimenzije n s bazom $v_1, v_2, \ldots, v_n$.* Fundamentalnu domenu *definiramo kao skup*

$$\mathcal{F}(v_1, v_2, \ldots, v_n) = \{t_1 v_1 + t_2 v_2 + \cdots + t_n v_n : 0 \leq t_i < 1\}.$$

Sljedeći rezultat pokazuje važnost fundamentalne domene:

Propozicija 2.1.5. *Neka je L rešetka dimenzije n i s fundamentalnom domenom $\mathcal{F}$. Tada svaki vektor $w \in \mathbb{R}^n$ možemo raspisati kao*

$$w = t + v, \quad \text{za jedinstvene} \quad t \in \mathcal{F} \quad i \quad v \in L.$$

Nadalje, unija svih translatiranih domena

$$\mathcal{F} + v = \{t + v : t \in \mathcal{F}\},$$

kada v prolazi kroz sve vektore u rešetki L, pokriva cijeli prostor $\mathbb{R}^n$.

Dokaz. Neka je $v_1, \ldots, v_n$ baza za rešetku L koja generira fundamentalnu domenu $\mathcal{F}$. Vektori $v_1, \ldots, v_n$ su linearno nezavisni u $\mathbb{R}^n$, pa su i baza za $\mathbb{R}^n$. Tada svaki vektor $w \in \mathbb{R}^n$ možemo zapisati kao

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \quad \text{za neke} \quad \alpha_1, \ldots, \alpha_n \in \mathbb{R}.$$

Svaki α_i zapišemo kao

$$\alpha_i = t_i + a_i \quad \text{gdje su} \quad 0 \leq t_i < 1 \quad \text{i} \quad a_i \in \mathbb{Z}.$$

Tada je

$$w = t_1 v_1 + t_2 v_2 + \cdots + t_n v_n + a_1 v_1 + a_2 v_2 + \cdots + a_n v_n,$$

gdje je $t_1 v_1 + t_2 v_2 + \cdots + t_n v_n \in \mathcal{F}$ i $a_1 v_1 + a_2 v_2 + \cdots + a_n v_n \in L$, odnosno w smo zapisali u željenom obliku.
Neka su sa $w = t + v = t' + v'$ zadane dvije reprezentacije vektora w kao sume vektora iz $\mathcal{F}$ i L. Imamo

$$(t_1 + a_1)v_1 + \cdots + (t_n + a_n)v_n = (t'_1 + a'_1)v_1 + \cdots + (t'_n + a'_n)v_n.$$

Kako su v_i linearno nezavisni slijedi:

$$t_i + a_i = t'_i + a'_i \quad \text{za svaki} \quad i = 1, 2, \ldots, n.$$

Dakle,

$$t_i - t'_i = a_i - a'_i \in \mathbb{Z}.$$

Kako su t_i i t'_i istovremeno $0 \leq t_i, t'_i < 1$, jedina mogućnost je da vrijedi $t_i - t'_i = 0$, odnosno $t_i = t'_i$. Dakle, $t = t'$, pa je i $v = v'$, čime smo pokazali i jedinstvenost. □

Nadalje, definiramo determinantu rešetke L.

Definicija 2.1.6. *Neka je L rešetka dimenzije n i $\mathcal{F}$ njena fundamentalna domena.* Determinanta *od rešetke L je $n-$ dimenzionalni volumen od $\mathcal{F}$. Oznaka je* $\det(L)$.

Vektore baze rešetke L možemo promatrati kao stranice fundamentalne domene $\mathcal{F}$. Tada je volumen najveći ako su vektori međusobno okomiti. Sljedeća propozicija nam daje gornju ogradu za $\det(L)$.

Propozicija 2.1.7. *(Hadamardova nejednakost) Neka je L rešetka dimenzije n, s bazom $v_1, \ldots, v_n$ i fundamentalnom domenom $\mathcal{F}$. Tada vrijedi*

$$\det(L) = Vol(\mathcal{F}) \leq \|v_1\| \, \|v_2\| \cdots \|v_n\|. \tag{2.2}$$

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 2.1.8. *Neka je $L \subset \mathbb{R}^n$ rešetka dimenzije n, vektori $v_1, \ldots, v_n$ baza za L i $\mathcal{F}$ fundametalna domena generirana tom bazom. Koordinate $i-$ tog vektora baze zapišemo kao*

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

te pomoću njih generiramo matricu

$$F = F(v_1, \ldots, v_n) = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}. \tag{2.3}$$

Tada je volumen od $\mathcal{F}$ dan s formulom

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

Dokaz. Volumen od $\mathcal{F}$ računamo kao integral konstantne funkcije 1 po $\mathcal{F}$,

$$Vol(\mathcal{F}) = \int_{\mathcal{F}} dx_1 dx_2 \cdots dx_n.$$

Napravimo zamjenu varijabli sa $x = (x_1, x_2, \ldots, x_n)$ u $t = (t_1, t_2, \ldots, t_n)$ prema formuli

$$(x_1, x_2, \ldots, x_n) = t_1 v_1 + t_2 v_2 + \cdots + t_n v_n.$$

U terminima matrice (2.3) promjena varijabli je dana sa $x = tF$. Jacobijan promjene je matrica F, a $\mathcal{F}$ dobijemo preko matrice F i jedinične kocke $C_n = [0,1]^n$. Dobivamo:

$$\begin{aligned}\int_{\mathcal{F}} dx_1 dx_2 \cdots dx_n &= \int_{FC_n} dx_1 dx_2 \cdots dx_n = \int_{C_n} |\det F|\, dt_1 dt_2 \cdots dt_n \\ &= |\det F|\, Vol(C_n) = |\det F|.\end{aligned}$$

□

Korolar 2.1.9. *Neka je $L \subset \mathbb{R}^n$ rešetka dimenzije n. Tada svaka fundamentalna domena od L ima isti volumen. Dakle,* $\det(L)$ *je nezavisna od odabira fundamentalne domene.*

Dokaz. Neka su $v_1, \ldots, v_n$ i $w_1, \ldots, w_n$ dvije baze za L pomoću kojih generiramo dvije fundamentalne domene. Neka su $F(v_1, \ldots, v_n)$ i $F(w_1, \ldots, w_n)$ matrice za odgovarajuće fundamentalne domene definirane sa (2.3). Po propoziciji (2.1.3) znamo da postoji cjelobrojna matrica A, sa $\det(A) = \pm 1$, takva da vrijedi

$$F(v_1, \ldots, v_n) = AF(w_1, \ldots, w_n). \tag{2.4}$$

Imamo niz jednakosti

$$\begin{aligned}Vol\Big(\mathcal{F}(v_1, \ldots, v_n)\Big) &= |\det(F(v_1, \ldots, v_n))| && \text{iz (2.1.8)} \\ &= |\det(AF(w_1, \ldots, w_n))| && \text{iz (2.4)} \\ &= |\det(A)|\,|\det(F(w_1, \ldots, w_n))| \\ &= |\det(F(w_1, \ldots, w_n))| && \text{jer je } \det A = \pm 1 \\ &= Vol\Big(\mathcal{F}(w_1, \ldots, w_n)\Big) && \text{iz (2.1.8)}\end{aligned}$$

□

2.2 Problemi SVP-a i CVP-a

Dva najbitnija problema vezana uz rešetke su pronalazak najkraćeg i najbližeg vektora unutar rešetke:

Problem najkraćeg vektora (The Shortest Vector Problem) - **SVP**:
Naći $v \in L$ takav da je norma $\|v\|$ minimalna.

Problem najbližeg vektora (The Closest Vector Problem) - **CVP**:
Za proizvoljan $w \in \mathbb{R}^n$, koji nije u L, naći $v \in L$ takav da je norma $\|w - v\|$ minimalna.

Rješenja ovih problema ne moraju biti jedinstvena. Oba problema se smatraju $\mathcal{NP}$ teškim problemima. Kriptosustavi koji se baziraju na $\mathcal{NP}$ problemima često uzimaju, da bi se postigla određena efikasnost, određene podklase problema. Naravno, ovakvim odabirom, uvijek postoji mogućnost da je odabrani problem lakše riješiti nego njegovu generalizaciju. Neke od varijanti SVP i CVP problema, koje se koriste u teoriji i praksi, su:

Problem najkraće baze (Shortest Basis Problem) - **SBP**:
Pronaći bazu $v_1, \ldots, v_n$ koja je "u nekom smislu" najkraća. Npr. možemo tražiti minimizaciju vektora baze po nekoj specifičnoj normi.

Aproksimativni problem najkraćeg vektora (Approximate Shortest Vector Problem) - **apprSVP**:
Neka je $\psi(n)$ funkcija koja ovisi o dimenziji n rešetke L. Ako je v' najkraći nenul unutar rešetke L, želimo pronaći nenul vektor $v \in L$ takav da vrijedi

$$\|v\| \leq \psi(n)\|v'\|.$$

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

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

U malim dimenzijama SVP problem se može riješiti egzaktno. Međutim, kako se dimenzija povećava $\mathcal{NP}$ teškoća problema dolazi do izražaja. Razlikujemo dvije vrste algoritama:

(a) Egzaktni algoritmi: Ovi algoritmi sigurno pronalaze najkraći vektor, međutim njihovo vrijeme izvršavanja je eksponencijalno u dimenziji rešetke. Problem se rješava pretragom po svim jako kratkim vektorima u rešetki. Najbolji deterministički algoritam je Kannanovo nabrajanje čije je vrijeme izvršavanja $n^{n/(2e)+O(n)}$, gdje je n dimenzija rešetke. Još jedan poznati algoritam je Ajtaiovo, Kumarov i Sivakumarovo sito čija je najveća kompleksnost $2^{O(n)}$.

(b) Aproksimativni algoritmi: Ovi algoritmi daju aproksimaciju za rješenje SVP problema, tako da je norma od pronađenog najkraćeg vektora ograđena odozgo. Algoritmi koji imaju polinomijalno vrijeme izvršavanje u svojoj ogradi imaju Hermitovu konstantu (2.3.1). Aproksimativni algoritmi su npr. LLL algoritam i blokovni Gama-Nguyen algoritam.

2.3 Teoremi Hermita i Minkowskog

Algoritmi za pronalazak najkraćeg i najbližeg vektora unutar rešetke su aproksimativni, te je dobro imati neki način ocjene koliko su ti vektori uistinu dobri za danu rešetku. Koliko je najkraći vektor zapravo velik ovisi ponajviše o dimenziji i determinanti rešetke.

Definicija 2.3.1. *Za danu rešetku L dimenzije n,* Hermitova konstanta γ_n *je najmanja vrijednost za koju vrijedi*

$$\|v\|^2 \leq \gamma_n \det(L)^{2/n},$$

gdje je $v \in L$ nenul vektor.

Točne vrijednosti Hermitove konstante su poznate za $1 \leq n \leq 8$ i za $n = 24$ te iznose

$$\gamma_2^2 = \frac{4}{3},\ \gamma_3^3 = 2,\ \gamma_4^4 = 4,\ \gamma_5^5 = 8,\ \gamma_6^6 = \frac{64}{3},\ \gamma_6^6 = 64,\ \gamma_7^7 = 64,\ \gamma_8^8 = 256 \text{ i } \gamma_{24} = 4.$$

Teorem 2.3.2. *(Hermitov teorem) Svaka rešetka L sadrži nenul vektor $v \in L$ takav da vrijedi*

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

Prethodni teorem zapravo kaže da vrijedi $\gamma_n \leq n$. Za velike dimenzije rešetke može se pokazati

$$\frac{n}{2\pi e} \leq \gamma_n \leq \frac{n}{\pi e}.$$

Postoje varijante Hermitovog teorema koje u obzir uzimaju više od jednog vektora. Za rešetku dimenzije n može se pokazati da vrijedi

$$\|v_1\|\,\|v_2\| \cdots \|v_n\| \leq n^{n/2}(\det L),$$

nejednakost koja je povezana sa (2.1.7). Znajući ove nejednakosti definiramo Hadamardov omjer:

Definicija 2.3.3. *Za bazu $\mathcal{B} = \{v_1, \ldots, v_n\}$ veličinu*

$$\mathcal{H}(\mathcal{B}) = \left(\frac{\det L}{\|v_1\|\,\|v_2\| \cdots \|v_n\|}\right)^{1/n}$$

zovemo Hadamardov omjer.

Vrijedi $0 < \mathcal{H}(\mathcal{B}) \leq 1$. Dakle, što je $\mathcal{H}(\mathcal{B})$ bliži jedinici baza je ortogonalnija.
U dokazu Hermitovog teorema koristimo teorem Minkowskog. Prije samog teorema navodimo potrebne definicije.

Definicija 2.3.4. *Za proizvoljan $a \in \mathbb{R}^n$ i $r > 0$ definiramo* zatvorenu kuglu *radijusa r kao skup*

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

Definicija 2.3.5. *Neka je S podskup od $\mathbb{R}^n$.*

(a) S *je* ograničen *ako su ograničene duljine svih vektora iz* S. *Ili,* S *je* ograničen *ako postoji radijus* r *takav da je* S *sadržan u kugli* $\mathbb{B}_r(0)$.

(b) S *je* simetričan *ako je za svaku točku* $a \in S$, *i točka* $-a$ *također iz* S.

(c) S *je* zatvoren *ako vrijedi: Ako je* $a \in \mathbb{R}^n$ *točka za koju svaka kugla* $\mathbb{B}_r(a)$ *sadrži točku iz* S, *onda je i* a *u* S.

(d) S *je* konveksan *ako je za svake dvije točke* $a, b \in S$, *i cijeli segment između* a *i* b *također iz* S.

Teorem 2.3.6. *(Teorem Minkowskog) Neka je* $L \subset \mathbb{R}^n$ *rešetka dimenzije* n *i neka je* $S \subset \mathbb{R}^n$ *simetričan, konveksni skup čiji volumen zadovoljava*

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

Tada S *sadrži nenul vektor rešetke. Nadalje, ako je* S *dodatno zatvoren, dovoljno je da vrijedi* $Vol(S) \geq 2^n \det(L)$.

Dokaz. Neka je $\mathcal{F}$ fundamentalna domena za L. Po propoziciji (2.1.5) znamo da se svaki vektor $a \in S$ može raspisati kao

$$a = v_a + w_a,$$

gdje je $v_a \in L$ i $w_a \in \mathcal{F}$. Promotrimo skup

$$\frac{1}{2}S = \left\{\frac{1}{2}a : a \in S\right\},$$

i preslikavanje

$$\frac{1}{2}S \longrightarrow \mathcal{F}, \quad \frac{1}{2}a \longrightarrow w_{\frac{1}{2}a}.$$

Volumen skupa $\frac{1}{2}S$ je smanjen za faktor 2^n, tako da vrijedi

$$Vol\left(\frac{1}{2}S\right) = \frac{1}{2^n}Vol(S) > \det L = Vol(\mathcal{F}).$$

Promatrano preslikavanje čuva volumen, pa kako je volumen domene veći od volumena kodomene, možemo pronaći dvije različite točke $\frac{1}{2}a_1$ i $\frac{1}{2}a_2$ koje imaju istu sliku. Dakle, u skupu S smo pronašli dvije različite točke za koje vrijedi

$$\frac{1}{2}a_1 = v_1 + w \quad \text{i} \quad \frac{1}{2}a_2 = v_2 + w, \quad v_1, v_2 \in L, w \in \mathcal{F}.$$

Oduzimanjem dobivamo nenul vektor

$$\frac{1}{2}a_1 - \frac{1}{2}a_2 = v_1 - v_2 \in L.$$

Vektor $\frac{1}{2}a_1 - \frac{1}{2}a_2$ je, radi simetričnosti i konveksnosti, u skupu S. Tada je nenul vektor $v_1 - v_2$ u presjeku skupa S i rešetke L, čime smo konstruirali nenul točku rešetke unutar skupa S.
Nadalje, pretpostavimo da je S zatvoren i da je $Vol(S) = 2^n \det(L)$. Za $k \geq 1$ proširimo S faktorom $1 + \frac{1}{k}$ i primijenimo prethodni rezultat da bi pronašli nenul vektor

$$0 \neq v_k \in \left(1 + \frac{1}{k}\right) S \cap L.$$

Svaki od vektora $v_1, v_2, \ldots$ je unutar ograničenog skupa $2S$, pa, jer je L diskretna, taj niz sadrži konačno mnogo različitih vektora. Izaberemo onaj vektor v koji se pojavljuje beskonačno mnogo puta u nizu. Dakle, imamo nenul vektor $v \in L$ koji je u presjeku

$$\bigcap_{k=1}^{\infty} \left(1 + \frac{1}{k}\right) S.$$

Kako je S zatvoren, ovaj presjek je upravo S, pa je $0 \neq v \in S \cap L$. □

Dokaz Hermitovog teorema sada slijedi kao posljedica.

Dokaz. (Teorema (2.3.2)) Neka je $L \subset \mathbb{R}^n$ rešetka i S n-dimenzionalna kocka sa središtem u nuli i stranicama duljine $2B$, tj.

$$S = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : -B \leq x_i \leq B \text{ za sve } 1 \leq i \leq n\}.$$

Skup S je simetričan, zatvoren i ograničen s volumenom $Vol(S) = (2B)^n$. Ako za duljinu stranice B stavimo $\det(L)^{1/n}$, volumen je jednak $2^n \det(L)$, te možemo primijeniti teorem Minkowskog da pronađemo vektor $0 \neq a \in S \cap L$. Za takav vektor $a = (a_1, \ldots, a_n)$ imamo

$$||a|| = \sqrt{a_1^2 + \cdots + a_n^2} \leq \sqrt{n}B = \sqrt{n}\det(L)^{1/n}.$$

□

2.4 Gaussova heuristika

Hermitovu konstantu možemo poboljšati ako koristimo hipersferu umjesto hiperkocke. Potreban nam je volumen kugle u $\mathbb{R}^n$.

Definicija 2.4.1. *Za $s > 0$ definiramo* gama funkciju $\Gamma(s)$ *s integralom*

$$\Gamma(s) = \int_0^\infty t^s e^{-t} \frac{dt}{t}. \tag{2.5}$$

Gama funkcija pojavljuje se u mnogim granama matematike. Navodimo nekoliko osnovnih svojstava ove funkcije.

Propozicija 2.4.2. *(a) Integral definiran sa* (2.5) *konvergira za $s > 0$.*

(b) $\Gamma(1) = 1$ *i* $\Gamma(s+1) = s\Gamma(s)$. *Ovime možemo proširiti gama funkciju za sve $s \in \mathbb{R}$ uz $s \neq 0, -1, -2 \dots$.*

(c) Za svaki $n \in \mathbb{N}$ imamo $\Gamma(n+1) = n!$.

(d) $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.

(e) Za velike vrijednosti od s vrijedi Stirlingova formula:

$$\Gamma(1+s)^{1/s} \approx \frac{s}{e}. \tag{2.6}$$

Formula za izračun volumena n-dimenzionalne kugle sadrži gama funkciju.

Teorem 2.4.3. *Neka je $\mathbb{B}_r(a)$ kugla radijusa r u $\mathbb{R}^n$. Tada je volumen od $\mathbb{B}_r(a)$ jednak*

$$Vol(\mathbb{B}_r(a)) = \frac{\pi^{n/2} r^n}{\Gamma(1+n/2)}. \tag{2.7}$$

Za velike vrijednosti n, volumen kugle $\mathbb{B}_r(a)$ je približno dan sa

$$Vol(\mathbb{B}_r(a))^{1/n} \approx \sqrt{\frac{2\pi e}{n}} r. \tag{2.8}$$

Za dokaz (2.8) možemo iskoristiti (2.7) i Stirlingovu formulu (2.6). Dobivamo

$$Vol(\mathbb{B}_r(a))^{1/n} = \frac{\pi^{1/2} r}{\Gamma(1+n/2)^{1/n}} \approx \frac{\pi^{1/2} r}{(n/2e)^{1/2}} = \sqrt{\frac{2\pi e}{r}}.$$

Pomoću teorema (2.4.3) možemo poboljšati ocjenu u Hermitovu teoremu (2.3.2) za velike vrijednosti n. Budući je kugla $\mathbb{B}_r(0)$ ograničena, zatvorena, konveksna i simetrična, po teoremu (2.3.6) ako odaberemo polumjer r takav da vrijedi

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

kugla $\mathbb{B}_r(0)$ će sadržavati točku rešetke koja nije nula. Ako je n velik, volumen kugle $\mathbb{B}_r(0)$ možemo aproksimirati sa (2.8), pa moramo odabrati r takav da vrijedi

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

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

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

Ovime smo poboljšali konstantu iz teorema (2.3.2) za faktor $\sqrt{2/\pi e} \approx 0.484$.

Nije poznato kako točno za veliki n ograničiti duljinu najkraćeg vektora unutar rešetke. Ipak, koristeći sljedeći princip možemo tu duljinu aproksimirati: Neka je $\mathbb{B}_r(0)$ velika kugla s centrom u 0. Tada je broj točaka rešetke u $\mathbb{B}_r(0)$ približno jednak omjeru volumena kugle $\mathbb{B}_r(0)$ i volumena fundamentalne domene $\mathcal{F}$. Broj elemenata u $\mathbb{B}_r(0) \cap L$ možemo promatrati kao broj kopija $\mathcal{F}$ koje stanu u kuglu $\mathbb{B}_r(0)$. Ovime dolazimo do Gaussove heuristike.

Definicija 2.4.4. *Neka je L rešetka dimenzije n. Očekivana* Gaussova najkraća duljina *je*

$$\sigma(L) = \sqrt{\frac{n}{2\pi e}} (\det(L))^{1/n}. \tag{2.9}$$

Gaussova heuristika *nam kaže da će najkraći nenul vektor v u proizvoljno odabranoj matrici zadovoljavati*

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

Preciznije, za odabrani $\epsilon > 0$, ako je n dovoljno velik, proizvoljno odabrana rešetka L dimenzije n će zadovoljavati

$$(1 - \epsilon)\, \sigma(L) \leq \|v\| \leq (1 + \epsilon)\, \sigma(L).$$

Napomena 2.4.5. *Za male vrijednosti od n bolje je koristiti točnu formulu* (2.7) *za volumen* $\mathbb{B}_r(0)$, *pa je očekivana Gaussova najkraća duljina za mali n jednaka*

$$\sigma(L) = \left(\Gamma(1 + n/2)\det(L)\right)^{1/n} / \sqrt{\pi}. \tag{2.10}$$

Npr. kada je $n = 5$, (2.9) *daje $\sigma(L) = 0.54106 \det(L)^{1/5}$, dok* (2.10) *daje $\sigma(L) = 0.717365 \det(L)^{1/5}$, što je značajna razlika. Kada je n veći, npr. $n = 200$ očekivane vrijednosti su*

$$\sigma(L) = 3.42198 \det(L)^{1/200} \quad i \quad \sigma(L) = 3.47756 \det(L)^{1/200},$$

što je puno manja razlika.

Gaussova heuristika će se pokazati korisnom kod pronalaženja kratkih vektora u rešetki. Naime, ako je najkraći vektor u rešetki L značajno kraći od $\sigma(L)$, algoritmi redukcije rešetke (poput LLL algoritma) jednostavnije pronalaze najkraći vektor. Analogno, Gaussova heuristika za CVP problem kaže da za rešetku $L \in \mathbb{R}^n$ dimenzije n i $w \in \mathbb{R}^n$ slučajno odabran, očekujemo da vektor $v \in (L)$ koji je najbliži w zadovoljava

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

Također, ako L sadrži točku koja je značajno bliža vektoru w nego $\sigma(L)$, algoritmi redukcije rešetke jednostavnije rješavaju CVP.

Poglavlje 3

Algoritmi redukcije rešetke

Cilj algoritama redukcije je iz zadane baze za rešetku dobiti onu u kojoj su vektori što ortogonalniji i što kraći. Sigurnost određenih kriptosustava ovisi o tome koliko se efikasno mogu riješiti problemi SVP-a i CVP-a (pa tako i apprSVP-a i apprCVP-a). *LLL* algoritam, opisan u ovom poglavlju rješava ove probleme do na faktor C^n, gdje je C mala konstanta, a n dimenzija rešetke vezane uz kriptosustav. Za rešetku dimenzije 2 koristimo Gaussovu redukciju.

3.1 Gaussova redukcija rešetke

Neka je $L \subset \mathbb{R}^2$ dvodimenzionalna rešetka s vektorima baze v_1 i v_2. Pretpostavimo da je $\|v_1\| < \|v_2\|$ (u suprotnom ih zamijenimo). Ideja algoritma je, dok god je to moguće, smanjivati v_2 za neki višekratnik drugog vektora baze v_1. Jedna mogućnost je da v_2 zamijenimo vektorom

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

koji je ortogonalan na v_1. Međutim, v_2^* nije nužno u rešetki 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 s $\lfloor x \rceil$ dan najbliži cijeli broj realnom broju x. Ako je v_2 i dalje veći, algoritam staje. Inače, zamijenimo v_1 i v_2 te ponovimo proces. Sljedeća propozicija pokazuje da algoritam staje u konačno mnogo koraka, te da je dobivena baza za L jako dobra.

Propozicija 3.1.1. *Neka je $L \subset \mathbb{R}^2$ dvodimenzionalna rešetka s vektorima baze 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 ||v_2|| < ||v_1|| then
        zamijenimo v_1 i v_2;
    end
    m = ⌊(v_1 · v_2)/||v_1||^2⌉;
    if m = 0 then
        vratimo trenutne vektore baze v_1 i v_2;
        break;
    else
        zamijenimo v_2 sa v_2 − m v_1;
    end
end
```

Kada algoritam završi, vektor v_1 je najkraći nenul vektor u rešetki L, tj. ovaj algoritam rješava SVP.

Dokaz. Pokazat ćemo da je v_1 najkraći nenul vektor u rešetki. Algoritam po završetku vraća vektore v_1 i v_2 za koje vrijedi:

$$\|v_2\| > \|v_1\|, \tag{3.1}$$

$$\frac{v_1 \cdot v_2}{\|v_1\|^2} \leq \frac{1}{2}. \tag{3.2}$$

Uzmimo proizvoljan nenul vektor $v \in L$. Za vektor v postoje jedinstveni $a_1, a_2 \in \mathbb{Z}$ takvi da je

$$v = a_1 v_1 + a_2 v_2.$$

Imamo

$$\begin{aligned}
\|v\|^2 &= \|a_1 v_1 + a_2 v_2\|^2 \\
&= a_1^2\|v_1\|^2 + 2a_1a_2(v_1 \cdot v_2) + a_2^2\|v_2\|^2 \\
&\geq a_1^2\|v_1\|^2 - 2|a_1a_2|\,|v_1 \cdot v_2| + a_2^2\|v_2\|^2 \\
&\geq a_1^2\|v_1\|^2 - |a_1a_2|\,\|v_1\|^2 + a_2^2\|v_2\|^2 \quad \text{iz (3.2)} \\
&\geq a_1^2\|v_1\|^2 - |a_1a_2|\,\|v_1\|^2 + a_2^2\|v_1\|^2 \quad \text{iz (3.1)} \\
&= (a_1^2 - |a_1|\,|a_2| + a_2^2)\|v_1\|^2.
\end{aligned}$$

Za prozvoljne realne brojeve t_1 i t_2 izraz

$$t_1^2 - t_1t_2 + t_2^2 = \left(t_1 - \frac{1}{2}t_2\right)^2 + \frac{3}{4}t_2^2 = \frac{3}{4}t_1^2 + \left(\frac{1}{2}t_1 - t_2\right)^2$$

je jednak nuli samo ako je $t_1 = t_2 = 0$. Kako su koeficijenti a_1, a_2 cijeli brojevi i nisu oba nula, imamo $\|v\|^2 \geq \|v_1\|^2$. □

3.2 LLL algoritam - opis

Problem SVP postaje teži kako se dimenzija povećava. Henrik Lenstra, Arjen Lenstra i Laszlo Lovasz 1982. godine su objavili LLL algoritam koji efikasno rješava probleme najkraćeg i najbližeg vektora u višim dimenzijama.
Pretpostavimo da rešetka L dimenzije n ima bazu $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$. Bazu $\mathcal{B}$ želimo transformirati u "bolju" bazu, tj. u bazu čiji su vektori što kraći, počevši s onim koji je najkraći. Također, želimo da su vektori što okomitiji, odnosno da je produkt $v_j \cdot v_i$ što bliži nuli za sve i, j. Da bi to postigli na bazu $\mathcal{B}$ primjenjujemo Gram-Schmidtov postupak ortogonalizacije (1.1.14). Za $i = 1$ je $v_1^* = v_1$, a za $i \geq 2$ imamo

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \quad \text{gdje je} \quad \mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \quad \text{za sve} \quad 1 \leq j \leq i-1. \tag{3.3}$$

Ovako dobivena baza $\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$ je ortogonalna baza za prostor razapet vektorima iz $\mathcal{B}$, međutim kako se unutar Gram-Schmidtovog postupka pojavljuju koeficijenti koji nisu cijeli brojevi, $\mathcal{B}^*$ nije baza za rešetku L razapetu sa $v_1, \ldots, v_n$. No, pokazat ćemo da te dvije baze imaju istu determinantu.

Propozicija 3.2.1. *Neka je $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ baza za rešetku L, a $\mathcal{B}^* = \{v_1^*, v_2^*, \ldots, v_n^*\}$ odgovarajuća baza dobivena Gram-Schmidtovim postupkom ortogonalizacije. Tada vrijedi:*

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

Dokaz. Neka je $F = F(v_1, v_2, \ldots, v_n)$ matrica generirana koordinatama vektora baze (opisano u (2.3)). Iz (2.1.8) znamo da je $\det(L) = |\det F|$. S vektorima baze $\mathcal{B}^*$ generiramo matricu $F^* = F(v_1^*, v_2^*, \ldots, v_n^*)$. Iz (3.3) vidimo da su matrice F i F^* povezane matričnom relacijom

$$MF^* = F,$$

gdje je M trokutasta matrica oblika

$$M = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ \mu_{2,1} & 1 & 0 & \dots & 0 & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \mu_{n-1,1} & \mu_{n-1,2} & \mu_{n-1,3} & \dots & 1 & 0 \\ \mu_{n,1} & \mu_{n,2} & \mu_{n,3} & \dots & \mu_{n-1,n} & 1 \end{pmatrix}.$$

Sada je

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

gdje smo koristili $\det M = 1$ i ortogonalnost u bazi $\mathcal{B}^*$. □

Prije samog LLL algoritma definiramo pojam LLL reducirane baze. U definiciji koristimo Gram-Schmidtovu bazu $\mathcal{B}^*$.

Definicija 3.2.2. *Neka je $\mathcal{B} = \{v_1, v_2, \dots, v_n\}$ baza za rešetku L, a $\mathcal{B}^* = \{v_1^*, v_2^*, \dots, v_n^*\}$ odgovarajuća baza dobivena Gram-Schmidtovim postupkom ortogonalizacije. Neka je $\delta \in [\frac{1}{4}, 1]$. Za bazu $\mathcal{B}$ kažemo da je* LLL reducirana *s faktorom δ ako zadovoljava:*

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

2. *Lovászov uvjet:*
$$\|v_i^*\|^2 \geq \left(\delta - \mu_{i,i-1}^2\right)\|v_{i-1}^*\|^2 \quad \textit{za sve} \quad 1 < i \leq n.$$

Lovászov uvjet možemo zapisati i kao

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

Ovaj uvjet kontrolira normu vektora $\|v_i^*\|$. Naime, kako Gram-Schmidtov algoritam ovisi o poretku vektora unutar baze, dobivena baza se može promijeniti ako v_i i v_{i-1} zamijene mjesta, tj. mogu se promijeniti v_i^* i v_{i-1}^*. Lovászov uvjet osigurava da se $\|v_i^*\|$ ne smanji previše. Najčešće se LLL redukcija radi sa $\delta = \frac{3}{4}$. LLL reducirana baza je skoro ortogonalna i vektori su poredani po rastućoj normi.
Sljedeći teorem nam kaže da je sa LLL reduciranom bazom moguće riješiti apprSVP.

Teorem 3.2.3. *Neka je L rešetka dimenzije n. Svaka LLL reducirana baza* $\{v_1, v_2, \ldots, v_n\}$ *zadovoljava sljedeća dva svojstva:*

$$\prod_{i=1}^{n} \|v_i\| \leq 2^{n(n-1)/4} \det L, \tag{3.4}$$

$$\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\| \quad \textit{za sve} \quad 1 \leq j \leq i \leq n. \tag{3.5}$$

Nadalje, početni vektor u LLL reduciranoj bazi zadovoljava

$$\|v_1\| \leq 2^{(n-1)/4} |\det L|^{1/n} \quad i \quad \|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in L} \|v\|. \tag{3.6}$$

Dakle, s LLL reduciranom bazom apprSVP se rješava s faktorom $2^{(n-1)/2}$.

Dokaz. Iz Lovászovog uvjeta i zato jer je $|\mu_{i,i-1}| \leq \frac{1}{2}$, slijedi

$$\|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|v_{i-1}^*\|^2 \geq \frac{1}{2} \|v_{i-1}^*\|^2. \tag{3.7}$$

Ako nejednakost (3.7) višestruko primijenimo na vektor v_j^* dobivamo ocjenu

$$\|v_j^*\|^2 \leq 2^{i-j} \|v_i^*\|^2. \tag{3.8}$$

Računamo

$$\begin{aligned}
\|v_i\|^2 &= \left\| v_i^* + \sum_{j=1}^{i-1} \mu_{i,j} v_j^* \right\|^2 && \text{iz} \quad (3.3) \\
&= \|v_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|v_j^*\|^2 && \text{jer su } v_1^*, v_2^*, \ldots, v_n^* \quad \text{ortogonalni} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \|v_j^*\|^2 && \text{jer je} \quad |\mu_{i,j}| \leq \frac{1}{2} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} 2^{i-j-2} \|v_i^*\|^2 && \text{iz} \quad (3.8) \\
&= \frac{1 + 2^{i-1}}{2} \|v_i^*\|^2 \\
&\leq 2^{i-1} \|v_i^*\|^2 && \text{jer je} \quad 1 \leq 2^{i-1} \quad \text{za svaki} \quad i \geq 1. \qquad (3.9)
\end{aligned}$$

Ako nejednakost (3.9) pomnožimo za svaki $1 \leq i \leq n$ dobivamo

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

gdje smo u zadnjem koraku koristili propoziciju (3.2.1). Dokaz tvrdnje (3.4) je gotov uzimanjem drugog korijena iz ove nejednakosti.
Nadalje, za svaki $j \leq i$ pomoću (3.9) (uz $i = j$) i (3.8) dobivamo ocjenu

$$\|v_j\|^2 \leq 2^{j-1}\|v_j^*\|^2 \leq 2^{j-1} \cdot 2^{i-j}\|v_i^*\|^2 = 2^{i-1}\|v_i^*\|^2.$$

Dokaz tvrdnje (3.5) je gotov uzimanjem drugog korijena iz ove nejednakosti.
U (3.5) uvrstimo $j = 1$, pomnožimo za svaki $1 \leq i \leq n$ i koristimo propoziciju (3.2.1) da dobijemo

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

Uzimanjem n-tog korijena dobivamo prvu ocjenu u (3.6).
Za dokaz druge ocjene u (3.6) uzmimo proizvoljan nenul vektor $v \in L$. Zapišimo v kao

$$v = \sum_{j=1}^{i} a_j v_j = \sum_{j=1}^{i} b_j v_j^*,$$

sa $a_i \neq 0$. Za razliku od a_i koji su cjelobrojni koeficijenti, b_i su realni. Posebno imamo $a_i \geq 1$. Po konstrukciji vektora v_i^* znamo da za proizvoljan k vektori $v_1^*, v_2^*, \ldots, v_k^*$ razapinju isti prostor kao i vektori $v_1, v_2, \ldots, v_k$. Dakle,

$$v \cdot v_i^* = a_i v_i^* \cdot v_i^* = b_i v_i^* \cdot v_i^* \quad \text{i} \quad v_i \cdot v_i^* = v_i^* \cdot v_i^*,$$

iz čega slijedi da je $a_i = b_i$, pa je i $|b_i| = |a_i| \geq 1$. U (3.5) uvrstimo $j = 1$ i dobivamo

$$\|v\|^2 = \left\| \sum_{j=1}^{i} b_j v_j^* \right\|^2 \leq b_i^2 \|v_i^*\|^2 \leq \|v_i^*\|^2 \leq 2^{-(i-1)}\|v_1\|^2 \leq 2^{-(n-1)}\|v_1\|^2.$$

Uzimanjem drugog korijena iz ove nejednakosti dobivamo i drugu ocjenu u (3.6). □

U svakom koraku vektori $v_1^*, v_2^*, \ldots, v_k^*$ su ortogonalni vektori dobiveni Gram-Schmidtovim algoritmom (1.1.14).

Teorem 3.2.4. *(LLL algoritam) Neka je $\{v_1, v_2, \ldots, v_n\}$ baza za rešetku L. Algoritam (3.1) u konačno mnogo koraka vraća LLL reduciranu bazu za rešetku L. Točnije, neka je $B = \max \|v_i\|$. Tada se koraci 3-9 algoritma (3.1) izvršavaju za $O(n^2 \log n + n^2 \log B)$, odnosno algoritam je polinomijalan.*

Input: Baza $v = \{v_1, v_2, \ldots, v_n\}$ za rešetku L dimenzije n
Output: LLL reducirana baza $v = \{v_1, v_2, \ldots, v_n\}$

```
k = 2;
v*_1 = v_1;
while k ≤ n do
    for j = 1, 2, ..., k − 1 do
        v_k = v_k − ⌊μ_{k,j}⌉ v*_j;
    end
    if ||v*_k||^2 ≥ (3/4 − μ^2_{k,k−1}) ||v*_{k−1}||^2 then
        k = k + 1;
    else
        zamijenimo v_{k−1} i v_k;
    end
end
```

Slika 3.1: LLL algoritam

3.3 Rješavanje apprCVP pomoću LLL

Ako rešetka L ima ortogonalnu bazu, problemi SVP-a i CVP-a se lako rješavaju. Uzmimo vektor $a \in L$ i raspišimo ga u ortogonalnoj bazi $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ rešetke L kao $a = a_1v_1 + a_2v_2 + \cdots + a_nv_n$. Tada je

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

Kako su koeficijenti $a_i \in \mathbb{Z}$, najkraći vektori u rešetki su najkraći vektori iz skupa $\{\pm v_1, \pm v_2, \ldots, \pm v_n\}$. Za CVP, pretpostavimo da želimo za zadani vektor $t \in \mathbb{R}^n$ pronaći vektor iz rešetke L koji mu je najbliži. Kako je $L \subset \mathbb{R}^n$ i L je dimenzije n, postoje koeficijenti $t_1, t_2, \ldots, t_n \in \mathbb{R}$ takvi da je

$$t = t_1v_1 + t_2v_2 + \cdots + t_nv_n.$$

Tada za vektor $a = a_1v_1 + a_2v_2 + \cdots + a_nv_n \in L$ imamo:

$$\|a - t\|^2 = (a_1 - t_1)^2\|v_1\|^2 + (a_2 - t_2)^2\|v_2\|^2 + \cdots + (a_n - t_n)^2\|v_n\|^2.$$

a_i su cjelobrojni koeficijenti tako da će $\|a - t\|$ biti minimizirana ako za a_i uzmemo onaj cijeli broj koji je najbliži koeficijentu t_i. Ovaj postupak neće dobro riješiti probleme SVP-a i CVP-a za one baze rešetke čiji su vektori jako neortogonalni. Uzmimo sada proizvoljnu bazu $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ za rešetku L. Preko baze $\mathcal{B}$ definiramo fundamentalnu domenu $\mathcal{F}$

(2.1.4). Također, iz (2.1.5) znamo da je svaki $w \in \mathbb{R}^n$ jedinstvena translacija oblika $\mathcal{F} + v$ za neki $v \in L$. Pretpostavit ćemo da je rješenje CVP onaj vrh paralelepipeda $L + v$ koji je najbliži vektoru w. Kako je

$$w = v + \epsilon_1 v_1 + \epsilon_2 v_2 + \cdots + \epsilon_n v_n \quad \text{za neke} \quad 0 \leq \epsilon_1, \epsilon_2, \ldots, \epsilon_n,$$

najbliži vrh nalazimo tako da ϵ_i, ako je manji od $\frac{1}{2}$, zamijenimo sa 0, a ako je ϵ_i veći ili jednak $\frac{1}{2}$ zamijenimo ga s 1. Opisani postupak uvelike ovisi o ortogonalnosti baze.

Teorem 3.3.1. *(Babaijev algoritam najbližeg vrha) Neka je $L \subset \mathbb{R}^n$ rešetka s bazom $\mathcal{B} = \{v_1, v_2, \ldots, v_n\}$ i neka je $w \in \mathbb{R}^n$ proizvoljan vektor. Ako su vektori baze $\mathcal{B}$ dovoljno ortogonalni, problem CVP možemo riješiti na sljedeći način:*

1 Raspišemo $w = t_1 v_1 + t_2 v_2 + \cdots + t_n v_n$, $t_1, t_2, \ldots, t_n \in \mathbb{R}$;
2 Stavimo $a_i = \lfloor t_i \rceil$ za $i = 1, 2, \ldots n$;
3 Rješenje problema je vektor $v = a_1 v_1 + a_2 v_2 + \cdots + a_n v_n$;

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

Sada u potpunosti možemo riješiti problem apprCVP:

Teorem 3.3.2. *(LLL apprCVP algoritam) Postoji konstanta C takva da za svaku rešetku L dimenzije n, zadanu bazom $v_1, v_2, \ldots, v_n$ algoritam:*

1 Iz baze $v_1, v_2, \ldots, v_n$ preko LLL algoritma dobijemo LLL reduciranu bazu;
2 Primijenimo Babaijev algoritam na LLL reduciranu bazu;

rješava problem apprCVP s faktorom C^n.

3.4 BKZ-LLL algoritam

BKZ-LLL (block Korkin-Zolotarev) algoritam je jedna od poboljšanih inačica 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 $v_1^*, v_2^*, \ldots,$ pripadne vektore dobivene Gram-Schmidtovim postupkom definirano preslikavanje

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

Dodatno, za $i = 0$ imamo identitetu $\pi_0(v) = v$.

Definicija 3.4.1. *Neka je L rešetka dimenzije n. Baza* $\{v_1, v_2, \ldots, v_n\}$ *rešetke L je* Korkin-Zoltarev (KZ) reducirana *ako vrijedi:*

(a) v_1 *je najkraći nenul vektor u L.*

(b) Za $i = 2, 3, \ldots, n$, *vektor* v_i *je izabran tako da* $\pi_{i-1}(v_i)$ *bude najkraći nenul 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 reducirana baza je općenito bolja nego LLL reducirana baza. Posebno, prvi vektor u KZ bazi je rješenje SVP problema. Zbog toga su i algoritmi koji traže KZ reduciranu 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 duljine β, npr.

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

i mijenjamo vektore sa KZ reduciranom bazom koja razapinje istu podrešetku.

Teorem 3.4.2. *Ako radimo BKZ-LLL algoritam na rešetki L dimenzije n i pritom koristimo blokove velličine* β, *algoritam će trebati ne više od* $O(\beta^{c\beta} n^d)$ *koraka, gdje su c i d male konstante. Dodatno, najkraći vektor* v_1 *će zadovoljavati*

$$\|v_1\| \leq \left(\frac{\beta}{\pi e}\right)^{\frac{n-1}{\beta-1}} \min_{0 \neq v \in L} \|v\|.$$

Napomena 3.4.3. *Po prethodnom teoremu BKZ-LLL algoritam rješava SVP za približni faktor od* $\beta^{n/\beta}$, *dok LLL to isto radi za faktor* $2^{n/2}$. *Kako povećavamo koeficijent* β *tako se povećava i točnost BKZ-LLL algoritma, ali i vrijeme izvršavanja.*

Poglavlje 4

Osnovno o kriptografiji

4.1 Kratka povijest kriptografije

Kriptografija je znanstvena disciplina koja se bavi proučavanjem matematičkih tehnika povezanih sa sigurnošću informacija, te metoda za slanje informacija u takvom obliku da ih može pročitati samo onaj kome su namijenjene.
Prvi oblici kriptografije pojavljuju se već kod starih Grka. Grci su koristil skital, spravu za šifriranje u obliku štapa oko kojeg se namotavala vrpca od pergamenta na koju se pisala poruka. Istu tu poruku bi mogla pročitati samo osoba koja je imala skital iste debljine.
U modernije vrijeme, razvojem računala i raznih komunikacijskih sustava, javlja se potreba za zaštitom informacija. Krajem sedamdesetih stručnjaci iz IBM-a razvijaju *DES* (*Data Encryption Standard*) najpoznatiji kriptografski mehanizam u povijesti. DES je bio standard za simetrične kriptosustave do 2001. godine kada ga je zamijenio *AES* (*Advances Encryption Standard*).
Novi pomaci u kriptografiji dogodili su se 1976. godine kada Diffie i Helman objavljuju *New Directions in Cryptography*. Ovaj rad je značajan jer se prvi puta uvodi ideja javnog ključa, te način razmjene ključeva koji se temelji na teškoći faktorizacije brojeva. Prvi kriptosustav koji je obuhvatio ove nove ideje bio je *RSA* kriptosustav, kojeg su 1978. godine objavili Rivest, Shamir i Adleman.
Početkom devedesetih razvija se kriptografija za digitalne potpise. Digitalni potpisi bitni su, između ostalog, za provjeru autentičnosti, kako podataka tako i osoba, identifikaciju itd. Godine 1991. prihvaćen je prvi međunarodni standard za digitalne potpise nazvan ISO/IEC 9796. Standard je baziran na kriptosustavima s javnim ključem, točnije na RSA kriptosustavu. Tri godine kasnije američka vlada prihvaća novi standard temeljen na ElGamalovom kriptosustavu, javnom kriptosustavu koji koristi problem računanja diskretnog logaritma u konačnim poljima.

4.2 Osnovni pojmovi u kriptografiji

Osnovna zadaća kriptografije je omogućiti pošiljatelju i primatelju nesmetanu komunikaciju preko komunikacijskog kanala, bez da neka treća osoba, koja nadzire njihovu komunikaciju, može razumjeti njihove poruke. Uobičajeno je da se ove tri osobe redom nazivaju Alice, Bob i Eva. Uvodimo nekoliko oznaka:

- $\mathcal{M}$ je skup svih mogućih poruka, tj. skup svih mogućih osnovnih elemenata otvorenog teksta.
- $\mathcal{C}$ je skup svih mogućih osnovnih elemenata šifrata. Element ovog skupa nazivamo šifrat.
- $\mathcal{K}$ je skup ključeva.
- Za svaki $k \in \mathcal{K}$ postoji jedinstvena funkcija šifriranja $E_k \colon \mathcal{M} \to \mathcal{C}$ i odgovarajuća funkcija dešifriranja $D_k \colon \mathcal{C} \to \mathcal{M}$.

Sustav koji zadovoljava prethodna svojstva nazivamo kriptosustav. Ako je $x \in \mathcal{M}$ otvoreni tekst, onda za funkcije šifriranja i dešifriranja treba vrijediti

$$D_k(E_k(x)) = x.$$

Proces u kojem primijenjujemo funkciju šifriranja na otvoreni tekst nazivamo šifriranje i, analogno, proces kada na šifrat primijenjujemo funkciju dešifriranja nazivamo dešifriranje. Sve poruke se šalju preko komunikacijskog kanala. Razlikujemo nekoliko vrsta:

- Fizički siguran kanal: Protivnik nema nikakav pristup kanalu kojim se šalju poruke
- Nesiguran kanal: Protivnik ima pristup kanalu i može dodavati, čitati, brisati ili prerasporediti informacije
- Siguran kanal: Protivnik ima pristup kanalu, ali nema mogućnost dodavanja, čitanja, brisanja ili preraspoređivanja informacija.

Osnovna pretpostavka u kriptografiji je da su skupovi $\mathcal{M}, \mathcal{C}$ i $\mathcal{K}$, kao i sve funkcije šifriranja i dešifriranja poznate. Jedine tajne informacije su ključevi za šifriranje i dešifriranje. Znanstvenu disciplinu koja se upravo bavi proučavanjem postupaka za čitanje skrivenih poruka bez poznavanja ključa nazivamo kriptoanaliza. Sigurnost kriptosustav se temelji na tajnosti ključa. Svakako, jedna od mogućnosti razbijanja kriptosustava je provjera svih mogućih ključeva. Zbog toga bi broj ključeva trebao biti dovoljno velik da bi ovaj pristup bio nepraktičan. Kriptosustav konstruiramo s ciljem da je pregled svih ključeva najbolja mogućnost za njegovo razbijanje.

4.3 Simetrični kriptosustavi

Za kriptosustav kažemo da je simetričan ako se ključ za šifriranje može jednostavno izračunati iz ključa za dešifriranje i obratno. U većini kriptosustava ti ključevi su jednaki, otkuda i ime simetričan. Najveći problem kod ovakvih kriptosustava je pronaći način za sigurnu razmjenu ključeva. Također, promjena ključa bi se trebala učestalo obavljati jer učestalo šifriranje istim ključem može biti pogodno za napad.
Prema shemi šifriranja simetrične kriptosustave možemo podijeliti na blokovne i protočne šifre. Blokovne šifre uzimaju blokove otvorenog teksta te ih šifriraju koristeći uvijek isti ključ, protočne šifre obrađuju svaki element otvorenog teksta koristeći niz ključeva. Također, prema načinu na koji šifriramo tekst razlikujemo supstitucijske i transpozicijske šifre. Kod supstitucijskih šifri svaki element otvorenog tekst se zamjenjuje nekim drugim elementom, dok se kod transpozicijskih šifri elementi permutiraju.

4.4 Asimetrični kriptosustavi

Neka su sa skupovima $\{E_k : k \in \mathcal{K}\}$ i $\{D_d : d \in \mathcal{K}\}$ dane sve funkcije šifriranja i dešifriranja, gdje je $\mathcal{K}$ skup svih ključeva. Promatramo par funkcija (E_k, D_d) i pretpostavimo da je za svaki ovakav par, ako znamo E_k, nepraktično uz dani šifrat $c \in C$ naći poruku $m \in \mathcal{M}$ takvu da vrijedi $E_k(m) = c$. Drugim riječima, uz poznavanje ključa k nije, u nekom razumnom vremenu, moguće pronaći ključ za dešifriranje d.
Funkcija E_k je u ovom slučaju osobna jednosmjerna funkcija. Naime, za funkciju f kažemo da je jednosmjerna ako se f lako, a njen inverz f^{-1} teško računa. Dodatno, ako znamo neki dodatni podatak (trapdoor - skriveni ulaz), onda je f osobna jednosmjerna funkcija. Unutar komunikacije Bob bi trebao posjedovati taj dodatni podatak da bi jednostavno izračunao inverz funkcije šifranja. Problemi faktorizacije brojeva su glavni izvori za osobne jednosmjerne funkcije.
Komunikaciju započinje kasniji primatelj poruke tako da pošalje svoj javni ključ pošiljatelju poruke. Pošiljatelj tada koristi primljeni javni ključ i šifrira poruku. Primatelj po primitku šifrata koristi svoj tajni ključ za dešifriranje. Komunikacija između više osoba se također pojednostavi, jer svi korisnici mogu svoje javne ključeve spremiti na jedno zajedničko mjesto iz kojeg pošiljatelj poruke odabire onaj javni ključ koji odgovara primatelju s kojim želi komunicirati. Dakle, kod ovakvog kriptosustava, osnovna pretpostavka je da poznavanje javnog ključa ne može pomoći pri izračunu tajnog ključa.
Neki od kriptosustava s javnim ključem su RSA, Rabinov, ElGamalov, Merkle-Hellmanov i kasnije opisani NTRU kriptosustav.

4.5 Usporedba simetričnog i asimetričnog kriptosustava

Očita razlika je tajnost ključeva. Kod simetričnog kriptosustava, iako su ključevi kraći nego kod asimetričnog, oba ključa moraju biti tajna. Ovo osobito može biti nedostatak u grupnoj komunikaciji, gdje je broj ključeva koji moraju biti tajni posebno velik. Svi sudionici grupne komunikacije imaju pristup javnim ključevima, poteškoća je jedino provjera autentičnosti tih javnih ključeva. Također, tajne ključeve kod simetričnih kriptosustava je potrebno učestalo mijenjati (čak i prilikom svake komunikacije), dok kod asimetričnog to nije potrebno. Jedan par ključeva može ostati isti kroz nekoliko godina. Još jedna bitna prednost asimetričnog kriptosustava je mogućnost digitalnog potpisa poruke.
Osnovni nedostatak zašto javni ključ ne koristimo za šifriranje, jest da su algoritmi s javnim ključem puno sporiji od simetričnih algoritama. Nastoji se uzeti najbolje od oba kriptosustava. Tako se kombinira dugotrajna mogućnost korištenja javnog/privatnog ključa kod asimetričnog kriptosustava s efikasnijom implementacijom simetričnih kriptosustava. Ako se komunikacija odvija preko simetričnog kriptosustava s ključem koji je razmijenjen preko kriptosustava s javnim ključem, dobivamo hibridni kriptosustav.

4.6 Vrste napada na kriptosustav

Cilj svakog napada na kriptosustav je iz šifrata otkriti otvoreni tekst, ili rjeđe, probati otkriti ključ kojim se šifrira. Razlikujemo nekoliko vrsta napada koje kriptoanalitičar može isprobati:

(a) **Samo šifrat**: Želimo samo preko šifrata doći do otvorenog teksta. Sheme kriptiranja koje su ranjive na ovaj napad smatraju se u potpunosti nesigurnima.

(b) **Poznat otvoreni tekst** Kriptoanalitičar posjeduje neki otvoreni tekst, ali i odgovarajući šifrat. Cilj je pronaći ključ koji se koristi.

(c) **Odabrani otvoreni tekst**: Postoji mogućnost odabira otvorenog teksta i njegovog šifrata. Ovaj napad je jači od prethodnog, ali je i manje realističan.

(d) **Odabrani šifrat**: Kriptoanalitičar odabire šifrat, te za njega može dobiti odgovarajući otvoreni tekst. Jedan način kako napadač nože doći do šifrata je ako dođe u posjed opreme koja se koristila za šifriranje.

(e) **Potkupljivanje, ucjena, krađa**: Napad koji ne pripada u kriptoanalizu, ali je u stvarnom svijetu svakako moguć. Često se kombinira sa prethodnim napadima.

Poglavlje 5

NTRU kriptosustav

NTRU je jedan od novijih kriptosustava s javnim ključem. Skraćenica NTRU dolazi od *N-th degree TRUncated polynomial ring*. Prvu verziju ovog algoritma razvili su Jeffrey Hoffstein, Jill Pipher i Joseph H. Silverman 1997. godine (ponekad se kratica NTRU tumači kao *Number Theory Research Unit*, zbog povezanosti navedenih znanstvenika s ovom grupom). Zajedno sa Danielom Liemanom ovi matematičari su osnovali *NTRU Cryptosystems, Inc.* i vlasnici su patenta za kriptosustav.
Šifriranje kod NTRU-a obavlja se pomoću prstena polinoma $R = \frac{\mathbb{Z}[x]}{x^N-1}$, u kojem je definirana operacija cikličke konvolucije. Također, polinomi iz R se reduciraju po dvama relativno prostim modulima p i q. NTRU encryption scheme ili skraćeno NTRUEncrypt se koristi za šifriranje, a NTRUSign se koristi za digitalne potpise.

5.1 NTRUEncrypt - opis algoritma

Neka je $N \geq 1$ cijeli broj koji predstavlja dimenziju prstena polinoma, te p i q dva relativno prosta modula. Promatramo kvocijentne prstene polinoma

$$R = \frac{\mathbb{Z}[x]}{x^N - 1}, \quad R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[x]}{x^N - 1}, \quad R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[x]}{x^N - 1}.$$

Polinom $a(x) \in R$ možemo promatrati kao element od R_p ili R_q nakon redukcije njegovih koeficijenata po p ili q. Također, polinom iz R_p ili R_q, centriranjem (1.3.7) njegovih koeficijenata možemo prikazati kao polinom iz prstena R. Nadalje, potrebni su nam binarni i ternarni polinomi:

Definicija 5.1.1. *Za prirodan broj d definiramo $B(d)$ kao skup svih polinoma koji imaju d koeficijenata jednakih* 1*, a preostali koeficijenti su jednaki* 0*. Svaki polinom iz $B(d)$ nazivamo* binarni *polinom.*

Definicija 5.1.2. *Za dva prirodna broja d_1 i d_2 definiramo*

$$T(d_1, d_2) = \left\{ a(x) \in R : \begin{array}{c} a(x) \text{ ima } d_1 \text{ koeficijenata jednakih } 1 \\ a(x) \text{ ima } d_2 \text{ koeficijenata jednakih } -1 \\ \text{svi preostali koeficijenti od } a(x) \text{ su } 0 \end{array} \right\}.$$

Svaki polinom iz $T(d_1, d_2)$ nazivamo ternarni *polinom.*

Sa $\mathcal{K}_f$ i $\mathcal{K}_g$ označeni su prostori polinoma iz kojih odabiremo privatne ključeve. Nadalje, $\mathcal{M}$ je prostor polinoma koji sadrži poruke koje želimo šifrirati (naravno trebamo imati metodu koja poruku koju želimo šifrirati prevodi u polinom iz ovog prostora), a $\mathcal{K}_r$ je skup trenutnih polinoma koji koristimo kod šifriranja. Ovi prostori polinoma su najčešće binarni ili ternarni polinomi. U ovom radu većina tvrdnji će biti iskazana za ternarne polinome. Sada možemo opisati sam postupak izračuna ključeva, šifriranja i dešifriranja.

5.1.1 Kreiranje ključeva

Za kreiranje NTRU ključeva Alice prvo izabire javne parametre (N, p, q). Za svoj privatni ključ Alice odabire na slučajan način dva polinoma

$$f(x) \in \mathcal{K}_f \quad \text{i} \quad g(x) \in \mathcal{K}_g.$$

Polinom $f(x)$ mora imati inverze modulo p i q. U suprotnom, trenutni $f(x)$ se odbacuje i izabire se novi slučajni polinom. Alice izračuna inverze

$$F_q(x) = f(x)^{-1} \quad \text{i} \quad F_p(x) = f(x)^{-1},$$

$F_q(x)$ i $F_p(x)$ su redom inverzi polinoma $f(x)$ u prstenima R_q i R_p. Pomoću $F_q(x)$ Alice računa polinom

$$h(x) = F_q(x) \circledast g(x) \ \in R_q. \tag{5.1}$$

Polinom $h(x)$ je javni ključ. Par $(f(x), F_p(x))$ je privatni ključ koji se koristi kod dešifriranja. Primijetimo da je dovoljno pohraniti $f(x)$, a $F_p(x)$ izračunati po potrebi.

5.1.2 Šifriranje

Pretpostavimo da Bob želi poslati Alice poruku $m(x) \in \mathcal{M}$. Bob izabire trenutni ključ, slučajan polinom $r(x) \in \mathcal{K}_r$ i uz dostupni javni ključ računa šifrat

$$e(x) \equiv ph(x) \circledast r(x) + m(x) \pmod{q}.$$

Šifrat $e(x)$ je element prstena R_q. Slučajan odabir polinoma $r(x)$ svrstava NTRUEncrypt u vjerojatnosne kriptosustave. Drugačiji odabir polinoma $r(x)$ dao bi drugačiji šifrat. Različite trenutne ključeve nije dobro koristiti za šifriranje iste poruke i različite poruke nije dobro šifrirati istim trenutnim ključem.

5.1.3 Dešifriranje

Po primitku šifrata Alice prvo računa

$$a(x) \equiv f(x) \circledast e(x) \pmod{q}.$$

Koeficijenti polinoma $a(x)$ trebaju biti unutar intervala $\left[-\frac{1}{2}q, \frac{1}{2}q\right]$. Da bi to postigli primjenimo operaciju centriranja (1.3.7) na polinom $a(x) \in R_q$. Sada je $a(x)$ element iz R. Sljedeći korak je redukcija polinoma $a(x)$ *modulo* p. Dobijemo

$$b(x) \equiv F_p(x) \circledast a(x) \pmod{p}.$$

Polinom $b(x)$ jednak je originalnoj poruci m.

Ako u NTRUEncrypt-u koristimo ternarne polinome, točnost dešifriranja možemo kontrolirati već samim odabirom parametara. Neka su (N, p, q) izabrani javni parametri i dodatno neka je d prirodan broj koji određuje broj koeficijenata koji su jednaki 1 i -1 u definiciji prostora ternarnih polinoma. Dodatno, neka su koeficijenti poruke $m(x)$ iz intervala $\left[-\frac{1}{2}p, \frac{1}{2}p\right]$. N, p, q, d i točno dešifriranje su povezani sljedećom propozicijom.

Propozicija 5.1.3. *Ako su parametri NTRU sustava izabrani tako da vrijedi*

$$q > (6d + 1)p, \tag{5.2}$$

onda je polinom $b(x)$ jednak poruci $m(x)$.

Dokaz.

$$\begin{aligned}
a(x) &\equiv f(x) \circledast e(x) \pmod{q} \\
&\equiv f(x) \circledast (ph(x) \circledast r(x) + m(x) \pmod{q}) \\
&\equiv f(x) \circledast ph(x) \circledast r(x) + f(x) \circledast m(x) \pmod{q} \\
&\equiv pf(x) \circledast F_q(x) \circledast g(x) \circledast r(x) + f(x) \circledast m(x) \pmod{q} \\
&\equiv pg(x) \circledast r(x) + f(x) \circledast m(x) \pmod{q}.
\end{aligned}$$

Promatramo polinom

$$pg(x) \circledast r(x) + f(x) \circledast m(x) \tag{5.3}$$

izračunat u R, a ne kao modulo q. Kako su polinomi $g(x)$ i $r(x)$ elementi istog skupa $\mathcal{K}_g = \mathcal{K}_r = T(d, d)$, najveći koeficijent koji se može pojaviti njihovoj konvoluciji je $2d$. Nadalje, $f(x)$ je element skupa $\mathcal{K}_f = T(d + 1, d)$, a koeficijenti polinoma $m(x)$ su između $-\frac{1}{2}p$ i $\frac{1}{2}p$, pa je najveći mogući koeficijent u njihovoj konvoluciji $(2d + 1) \cdot \frac{1}{2}p$. Najveći mogući koeficijent polinoma (5.3) tada je jednak

$$p \cdot 2d + (2d + 1) \cdot \frac{1}{2}p = \left(3d + \frac{1}{2}\right)p.$$

Radi pretpostavke (5.2) svaki koeficijent polinoma (5.3) je strogo manji od $\frac{1}{2}q$. Dakle, $a(x)$ nije samo izračunato modulo q, već točno izračunata vrijednost u R, tj.

$$a(x) = pg(x) \circledast r(x) + f(x) \circledast m(x).$$

Sada imamo:

$$\begin{aligned} b(x) &= F_p(x) \circledast a(x) \\ &= F_p(x) \circledast (pg(x) \circledast r(x) + f(x) \circledast m(x)) \\ &\equiv F_p(x) \circledast f(x) \circledast m(x) \pmod{p} \\ &\equiv m(x) \pmod{p}. \end{aligned}$$

Dakle, $b(x)$ i $m(x)$ su jednaki modulo p. □

Uvjet $q > (6d + 1)p$ osigurava da dešifriranje provedemo do kraja. No, kako je malo vjerojatno da će se baš svi koeficijenti polinoma $g(x)$ i $r(x)$ (pa tako i od $f(x)$ i $m(x)$) točno posložiti, dešifriranje će uspjeti i za manje vrijednosti parametra q. Tipičan izbor za parametre p i q je $p = 3$ i $q = 128$. Nadalje, u slučaju $\gcd(p, q) \neq 1$ i npr. neka tada $p|q$, dešifriranje bi bilo jako jednostavno. Naime, kako je $e(x) \equiv ph(x) \circledast r(x) + m(x) \pmod{q}$ i $p|q$ imamo da je $e(x) \equiv ph(x) \circledast r(x) + m(x) \pmod{p}$, iz čega jednostavno dobijemo tekst m.

Primjer 5.1.4. *Neka su parametri NTRUEncrypt sustava* $(N, p, q, d) = (7, 3, 128, 2)$. *Vidimo da za ovako odabrane parametre vrijedi*

$$128 = q > (6d + 1)p,$$

tako da će dešifriranje biti uspješno. Alice odabire proizvoljne polinome

$$f(x) = -1 + x + x^3 + x^4 - x^6 \in \mathcal{T}(3, 2) \quad i \quad g(x) = 1 + x^2 - x^3 - x^6 \in \mathcal{T}(2, 2).$$

Inverzi polinoma $f(x)$ modulo q i p su

$$\begin{aligned} F_q(x) &= 9 + 17x + 16x^2 + 126x^3 + 92x^4 + 60x^5 + 65x^6, \\ F_p(x) &= 1 - x - x^2 - x^4 + x^5 - x^6. \end{aligned}$$

Par polinoma $(f(x), F_p(x))$ je privatni ključ, a Alice objavi svoj javni ključ

$$h(x) = 88 + 6x + 90x^2 + 42x^3 + 31x^4 + 105x^5 + 22x^6.$$

Pretpostavimo da Bob želi poslati poruku

$$m(x) = -1 - x + x^3 + x^4$$

koristeći trenutni ključ

$$r(x) = -x + x^2 - x^3 + x^6.$$

Bob, pomoću javnog ključa, računa šifrat

$$e(x) = 45 + 12x + 50x^2 + 90x^3 + 58x^4 + 85x^5 + 44x^6.$$

Po primitku šifrata Alice jednostavno provodi dešifriranje. Prvo izračuna

$$f(x) \circledast e(x) \equiv 7 + 126x + x^2 + 119x^3 + 4x^4 + 119x^5 + 8x^6 \pmod{q},$$

što potom centrira modulo q da dobije polinom

$$a(x) = 7 - 2x + x^2 - 9x^3 + 4x^4 - 9x^5 + 8x^6.$$

Nadalje, $a(x)$ se reducira modulo p,

$$F_p(x) \circledast a(x) \equiv 2 + 2x + x^3 + x^4 \pmod{p}.$$

Centriranjem prethodnog polinoma modulo p dobiva se poruka $m(x)$.

5.1.4 Pogrešno dešifriranje

Iz propozicije (5.1.3) uočavamo vezu između polinom $a(x)$ i polinoma (5.3). Bez ternarnih polinoma i bez uvjeta (5.2) koeficijenti polinoma $pg(x) \circledast r(x) + f(x) \circledast m(x)$ nisu nužno unutar intervala $\left[-\frac{1}{2}q, \frac{1}{2}q\right]$. U tom slučaju dešifriranje neće biti uspješno. Ova pojava je prisutnija kod upotrebe binarnih polinoma. Da bi se ovo izbjeglo koeficijenti od $a(x)$ se trebaju smjestiti u interval $[A, A + q - 1]$. Za izračun A koristimo svojstvo konvolucije $(a \circledast b)(1) = a(1) \cdot b(1)$, gdje je sa $a(1)$ označena suma koeficijenata polinoma a. Vrijednosti $r(1)$ i $h(1)$ su poznate pa kod dešifriranja možemo izračunati

$$I = m(1) = e(1) - r(1) \cdot h(1) \pmod{q}.$$

Pretpostavimo dodatno, da je $m(1)$ unutar intervala $[N/2 - q/2, N/2 + q/2]$. Sada je

$$A = \left[\frac{p(1) \cdot r(1) \cdot g(1) + f(1) \cdot I}{N} - \frac{q}{2}\right].$$

Ako bilo koja vrijednost polinoma $pg(x) \circledast r(x) + f(x) \circledast m(x)$ nije unutar zadanog intervala, dešifriranje neće uspjeti. Ako je $width$[1]$(pg(x) \circledast r(x) + f(x) \circledast m(x)) < q$, dobivamo *wrap failure*. Dešifriranje u ovom slučaju neće uspjeti za početni A, ali može postojati neki drugi A' s kojim bi proveli postupak do kraja. Novi A' dobivamo redom isprobavajući vrijednosti $A \pm 1, A \pm 2, \ldots$ i reduciranjem $a(x)$ na interval $[A', A' + q - 1]$. Ako je $width(pg(x) \circledast r(x) + f(x) \circledast m(x)) \geq q$, dobivamo *gap failure*.

[1]Za polinom a stupnja n, $width(a) = max(a_0, \ldots, a_n) - min(a_0, \ldots, a_n)$.

5.2 Sheme kriptiranja

Da bi zaštitili NTRUEncrypt od napada odabranim šifratom uvodimo sheme kriptiranja. Za NTRU to su $NAEP$ i poboljšana inačica iste sheme, nazvana $SVES3$. Unutar algoritama pojavljuju se dvije operacije sa stringovima - *ekskluzivno-ili* ($\oplus$) i *konkatenacija* ($\|$).

Napomena 5.2.1. *Konkatenacija stringova je zapravo spajanje dva stringa u jedan. Npr. za stringove* 110 *i* 101 *operacijom konkatenacije dobivamo string*

$$(110 \| 101) = 110101.$$

Primjenom operacije ekskluzivno-ili (često se naziva i XOR) na prethodne stringove dobivamo

$$(110 \oplus 101) = 011,$$

gdje koristimo istinitosnu tablicu

P	Q	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

.

5.2.1 NAEP

Koristimo dvije *hash* funkcije:

$$G\colon \{0,1\}^{N-l} \times \{0,1\}^{l} \to \mathcal{K}_r \quad \text{i} \quad H\colon \{0,1\}^{N} \to \{0,1\}^{N}.$$

Funkcija G generira slučajan polinom $r \in \mathcal{K}_r$ te se stoga zove BPGM (Blinding Polynomial Generation Method). Funkcija H se naziva Mask Generation Function (MGF). Nadalje, za šifriranje poruke $M \in \{0,1\}^{N-l}$ potrebne su nam funkcije

$$compress(x) = (x \bmod q) \pmod 2,$$

$$B2P\colon \{0,1\}^{N} \to \mathcal{M} \cup \text{"}error\text{"} \quad \text{i} \quad P2B\colon \mathcal{M} \to \{0,1\}^{N}.$$

Funkcija $compress$ najprije koeficijente izraza $x \bmod q$ stavi unutar intervala $[0, q)$, a zatim ih reducira modulo 2. Funkciju $compress$ koristimo da bi reducirali veličinu ulaznih podataka. $B2P$ prevodi bitovni string u binarni polinom, a $P2B$ prevodi binarni polinom u bitovni string.
Odaberemo slučajan string $b \leftarrow \{0,1\}^{l}$, a zatim, da bi dobili trenutni polinom $r \in \mathcal{K}_r$,

primijenimo hash funkciju G, $r = G(M, b)$. Pomoću funkcija $compress$ i $B2P$ izračunamo novu poruku za šifriranje:

$$m = B2P((M \parallel b) \oplus H(compress(r \circledast h))).$$

U slučaju da funkcija $B2P$ vrati "error", naš početni b nije dobro odabran te ga moramo zamijeniti. Šifrat je sada

$$e = r \circledast h + m \in R_q.$$

Po primitku poruke e računamo $a = f \circledast e (\text{mod } q)$. Naravno, moramo paziti da su koeficijenti od a u odgovarajućem intervalu. Nadalje, $m = F_p^{-1} \circledast a (\text{mod } q)$ i $s = e - m$. Kod dešifriranje koristimo funkciju $P2B$ da dobijemo

$$M \parallel b = P2B(m) \oplus H(compress(P2B(s))).$$

Neka je $r = G(M, b)$. Tada, ako je $r \circledast h = s$ (mod q) i $m \in \mathcal{M}$, vratimo poruku M. Ako prethodno nije zadovoljeno, dešifriranje nije uspjelo.
Primijetimo, da smo, da bi poruka M bila dovoljno dugačka, uveli slučajan parameter b za popunu preostalih bitova.

5.2.2 NAEP:SVES3

Dvije su bitne promjene u $SVES3$ u odnosu na $NAEP$:

- Na poruku se dodaje jedan bajt koji označava duljinu poruke. Ako je potrebno poruka se nadopuni nulama.
- Hash funkcija G, uz M i b, za varijable još uzima i OID te h_{trunc}. Sa OID identificiramo koju shemu i koje parametre koristimo. h_{trunc} je jedan dio javnog ključa.

Korištenje h_{trunc} parametra kao varijable funkcije G ima za posljedicu ovisnost slučajno odabranog r o specifičnom javnom ključu. Međutim, kako SVES3, s trenutno preporučenim optimalnim parametrima, nema problema s pogreškom dešifriranja, h_{trunc} se zanemaruje.

Poglavlje 6

Sigurnost NTRU kriptosustava

U ovom poglavlju koristimo ternarne polinome. Dakle,

$$\mathcal{K}_f = \mathcal{T}(d+1, d),\ \mathcal{K}_g = \mathcal{T}(d, d),\ \mathcal{K}_r = \mathcal{T}(d, d).$$

6.1 Matematička pozadina NTRUEncrypt kriptosustava

Kako za javni ključ $h(x)$ prema (5.1) vrijedi $h(x) = F_q(x) \circledast g(x)$, imamo

$$f(x) \circledast h(x) = f \circledast F_q(x) \circledast g(x)$$

iz čega dobivamo vezu

$$f(x) \circledast h(x) \equiv g(x) \pmod{q}.$$

Dakle, kada znamo $h(x)$, problem je naći ternarne polinome $f(x)$ i $g(x)$ koji zadovoljavaju

$$f(x) \circledast h(x) \equiv g(x) \pmod{q}. \tag{6.1}$$

Primijetimo da rješenje ovog problema nije jedinstveno. Naime, ako su $f(x)$ i $g(x)$ rješenja, tada su i polinomi $x^k \circledast f(x)$ i $x^k \circledast g(x)$ također rješenja za svaki $0 \leq k < N$. Polinom $x^k \circledast f(x)$ nazivamo rotacija od $f(x)$. Štoviše, bilo koji par polinoma $f(x)$ i $g(x)$ s dovoljno malim koeficijentima i takvi da zadovoljavaju (6.1) mogu poslužiti kao ključ.

6.2 Napadi bez korištenja strukture rešetke

6.2.1 Napad "grubom silom"

Pretpostavimo da Eva želi pronaći privatni ključ "grubom silom", tj. pronaći i provjeriti sve moguće privatne ključeve. Ako je $f(x)$ privatni ključ, onda će polinom $f(x) \circledast h(x) \pmod{q}$

biti ternarni polinom. Polinomi koji zadovoljavaju ovaj uvjet će uglavnom biti rotacije polinoma $f(x)$. Zanima nas veličina skupa svih ternarnih polinoma oblika $\mathcal{T}(d_1, d_2)$. Prvo izaberimo d_1 koeficijenata koji su jednaki 1, a od preostalih $N - d_1$ koeficijenata, d_2 koji su jednaki -1. Dakle, ukupan broj polinoma je:

$$\#\mathcal{T}(d_1, d_2) = \binom{N}{d_1}\binom{N-d_1}{d_2} = \frac{N!}{d_1!d_2!(N-d_1-d_2)!}. \tag{6.2}$$

Ovaj broj je maksimalan ako su i d_1 i d_2 približno $N/3$. Za pronalazak ključa Eva treba provjeriti sve polinome iz $\mathcal{T}(d_1, d_2)$, no kako su i sve rotacije ključa $f(x)$ također ključevi Eva će zapravo trebati provjeriti $\mathcal{T}(d_1, d_2)/N$ polinoma.

Primjer 6.2.1. *Neka su parametri NTRUEncrypt sustava*

$$(N, p, q, d) = (251, 3, 257, 83).$$

Primijetimo da ovi parametri ne zadovoljavaju uvjet $q > (6d + 1)p$, tako da može doći do pogreške u dešifriranju. Eva mora provjeriti otprilike

$$\frac{\mathcal{T}(84, 83)}{251} = \frac{1}{251}\binom{251}{84}\binom{167}{83} \approx 2^{381.6}$$

polinoma da bi pronašla odgovarajući ključ.

6.2.2 Algoritam kolizije

Za pronalazak privatnog ključa Eva može koristiti algoritam kolizije. Od svih parova ternarnih polinoma

$$f_1(x) = \sum_{0 \le i < N/2} a_i x^i \quad \text{i} \quad f_2(x) = \sum_{N/2 \le i < N} a_i x^i$$

tražimo one koji zadovoljavaju $f_1(x) + f_2(x) \in \mathcal{T}(d + 1, d)$. Eva računa

$$f_1(x) \circledast h(x) \pmod q \quad \text{i} \quad -f_2(x) \circledast h(x) \pmod q$$

i raspoređuje ih u grupe prema njihovim koeficijentima. Grupe su napravljene tako da polinom

$$(f_1(x) + f_2(x)) \circledast h(x) \pmod q$$

ima male koeficijente. Znači, polinom $(f_1(x) + f_2(x))$ može biti potreban ključ. Broj polinoma koje treba provjeriti ovim algoritmom je otprilike drugi korijen iz (6.2). Ako želimo maksimizirati broj polinoma iz $\mathcal{T}(d+1, d)$ uzimamo $d \approx N/3$. Pomoću Stirlingove formule tada dobivamo

$$\#\mathcal{T}(d + 1, d) \approx \frac{N!}{((N/3)!)^3} \approx \left(\frac{N}{e}\right)^N \cdot \left(\left(\frac{N}{3e}\right)^{N/3}\right)^{-3} \approx 3^N.$$

Prema tome algoritam kolizije bi zahtijevao $O(3^{N/2}/\sqrt{N})$ koraka.

Ova dva pokušaja napada su neki od razloga zašto se problem pronalaska NTRU privatnih ključeva smatra teškim problemom. Naime, vidjeli smo da je broj ključeva koje treba provjeriti velik u oba algoritma. Nadalje, postoji veza između rješavanja SVP-a u određenim rešetkama i pronalaska privatnog ključa za NTRU. Algoritmi redukcije rešetki su trenutno najbolji za rješavanje ovih problema.

6.3 NTRUEncrypt rešetka

Problem pronalaska ključa u NTRU kriptosustavu se preformulira ili kao problem pronalaska najkraćeg vektora, ili kao problem pronalaska najbližeg vektora u posebnim vrstama rešetki. Pripadni NTRUEncrypt javni ključ $h(x)$ možemo pisati kao

$$h(x) = h_0 + h_1 x + \cdots + h_{N-1} x^{N-1}.$$

Javnom ključu $h(x)$ pridružimo rešetku L_h^{NTRU} dimenzije $2N$ razapetu stupcima matrice:

$$M_h^{NTRU} = \left(\begin{array}{cccc|cccc} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ \hline h_0 & h_{N-1} & \dots & h_1 & q & 0 & \dots & 0 \\ h_1 & h_0 & \dots & h_2 & 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \dots & h_0 & 0 & 0 & \dots & q \end{array}\right). \tag{6.3}$$

Vidimo da je matrica M_h^{NTRU} blok matrica koja se sastoji od jedinične matrice, cikličke permutacije javnog ključa $h(x)$, nul matrice i još jedne jedinične matrice, samo pomnožene s parametrom q. Zato matricu M_h^{NTRU} možemo skraćeno pisati kao

$$M_h^{NTRU} = \begin{pmatrix} I & 0 \\ \mathbf{h} & qI \end{pmatrix},$$

te ju promatramo kao matricu s koeficijentima u prstenu R. Svi parametri u ovoj matrici su javni, te se ona smatra lošom bazom za rešetku koju generira svojim stupcima.

Zanima nas veza između ovakvo definirane matrice i privatnih ključeva unutar NTRUEncrypt algoritma.

Propozicija 6.3.1. *Neka je* $f(x) \circledast h(x) \equiv g(x) \pmod q$ *i* $u(x) \in R$ *polinom za kojeg vrijedi*

$$f(x) \circledast h(x) = g(x) + qu(x). \tag{6.4}$$

Tada je

$$(f, -u)(M_h^{NTRU})^\top = (f, g), \tag{6.5}$$

odnosno vektor (f, g) *je u rešetki* L_h^{NTRU}.

Dokaz. Iz oblika matrice $(M_h^{NTRU})^\top$, jedinična matrica povrh nul-matrice, vidimo da je prvih N koordinata matrice umnoška (6.5) očito vektor f. Kod umnoška vektora $(f, -u)$ sa stupcima koji počinju sa h_k dobivamo:

$$f_0 h_k + f_1 h_{k-1} + \cdots + f_{N-1} h_{k+1} - qu_k,$$

što je k-ta koordinata vektora $f(x) \circledast h(x) - qu(x)$. Iz (6.4) slijedi da je prethodno također i k-ta koordinata vektora g. Dakle, drugih N koordinata umnoška (6.5) formiraju vektor g. Nadalje, iz jednakosti (6.5) vidimo da vektor (f, g) možemo dobiti kao linearnu kombinaciju redaka matrice $(M_h^{NTRU})^\top$, a kako je L_h^{NTRU} razapeta upravo recima te matrice, odnosno stupcima matrice M_h^{NTRU} dobivamo da je $(f, g) \in L_h^{NTRU}$. □

Uz skraćeni zapis matrice M_h^{NTRU} jednakost (6.5) možemo pisati kao

$$(f, -u)\begin{pmatrix} 1 & h \\ 0 & q \end{pmatrix} = (f, f \circledast h - qu) = (f, g).$$

Propozicija 6.3.2. *Neka su* (N, p, q, d) *parametri NTRUEncrypt kriptosustava. Pretpostavimo*

$$d \approx N/3 \quad i \quad q \approx 6d \approx 2N.$$

Neka je L_h^{NTRU} *rešetka s pripadajućim ključem* (f, g). *Vrijedi:*

a) $\det(L_h^{NTRU}) = q^N$.

b) $\|(f, g)\| \approx \sqrt{4d} \approx \sqrt{4N/3} \approx 1.155\sqrt{N}$.

c) Gaussova heuristika predviđa da je najkraći nenul vektor u NTRUEncrypt rešetki duljine

$$\sigma(L_h^{NTRU}) \approx \sqrt{Nq/\pi e} \approx 0.484N.$$

Dakle, ako je N velik, veća je vjerojatnost da su najkraći nenul vektori u rešetki L_h^{NTRU} *upravo* (f, g) *i njegove rotacije. Također,*

$$\frac{\|(f, g)\|}{\sigma(L)} \approx \frac{2.39}{\sqrt{N}},$$

pa je vektor (f, g) *reda* $O(1/\sqrt{N})$ *kraći nego što je predviđeno Gaussovom heuristikom.*

Dokaz. a) Iz propozicije (2.1.8) znamo da je $\det(L_h^{NTRU})$ jednaka determinanti matrice M_h^{NTRU}. Ova matrica je gornjetrokutasta pa joj je determinanta jednaka umnošku elemenata na dijagonali, tj. jednaka je q^N.

b) Polinomi f i g su, redom, iz prostora $\mathcal{T}(d+1, d)$ i $\mathcal{T}(d, d)$, tako da i jedan drugi imaju otprilike d koordinata jednakih 1 i d koordinata jednakih -1.

c) Iskoristimo tvrdnju (a) i činjenicu da je L_h^{NTRU} dimenzije $2N$, možemo procijeniti Gaussovu najkraću duljinu formulom (2.9),

$$\sigma(L_h^{NTRU}) = \sqrt{\frac{2N}{2\pi e}}(det(L))^{1/2N} = \sqrt{\frac{Nq}{\pi e}} \approx \sqrt{\frac{2N^2}{\pi e}} = N\sqrt{\frac{2}{\pi e}}.$$

□

6.4 Napadi koji koriste strukturu rešetke

6.4.1 LLL algoritam i NTRUEncrypt rešetka

Pokazat ćemo kako LLL algoritam radi na primjeru (5.1.4). Dakle, $N = 7$, $q = 128$ i javni ključ je

$$h(x) = 88 + 6x + 90x^2 + 42x^3 + 31x^4 + 105x^5 + 22x^6.$$

NTRU rešetka je generirana stupcima matrice

$$M_h^{NTRU} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
88 & 22 & 105 & 31 & 42 & 90 & 6 & 128 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 88 & 22 & 105 & 31 & 42 & 90 & 0 & 128 & 0 & 0 & 0 & 0 & 0 \\
90 & 6 & 88 & 22 & 105 & 31 & 42 & 0 & 0 & 128 & 0 & 0 & 0 & 0 \\
42 & 90 & 6 & 88 & 22 & 105 & 31 & 0 & 0 & 0 & 128 & 0 & 0 & 0 \\
31 & 42 & 90 & 6 & 88 & 22 & 105 & 0 & 0 & 0 & 0 & 128 & 0 & 0 \\
105 & 31 & 42 & 90 & 6 & 88 & 22 & 0 & 0 & 0 & 0 & 0 & 128 & 0 \\
22 & 105 & 31 & 42 & 90 & 6 & 88 & 0 & 0 & 0 & 0 & 0 & 0 & 128
\end{pmatrix}.$$

Na matricu M_h^{NTRU} primijenjen je algoritam LLL redukcije implementiran u programskom alatu *Matlab*. Na stupcima matrice napravi se Gram-Schmidtov postupak. Zatim se za

svaki stupac provjeri uvjet veličine i Lovászov uvjet. Korišten je parametar $\delta = \frac{3}{4}$. Po potrebi se ažuriraju koeficijenti u Gram-Schmidtovu postupku. Dodatno, ako Lovászov uvjet nije zadovoljen, mijenjamo poredak vektora baze. LLL algoritam uvelike ovisi o rasporedu vektora baze, tako da drugačiji početni raspored vektora baze može dovesti do drugačijeg oblika reducirane matrice M_{red}^{NTRU}. Algoritam vrati reduciranu matricu

$$M_{red}^{NTRU} = \begin{pmatrix} 1 & -1 & 0 & 1 & 0 & 1 & -1 & 6 & 5 & -8 & -25 & 1 & -5 & -24 \\ 1 & -1 & 1 & 0 & -1 & 1 & 1 & 1 & -18 & -4 & -2 & 7 & 1 & 24 \\ -1 & 0 & 1 & 1 & -1 & 1 & 0 & 24 & 19 & -10 & -6 & 1 & 6 & 2 \\ 0 & 1 & -1 & 1 & 0 & 1 & 1 & -24 & -6 & 15 & 0 & -6 & 1 & 5 \\ -1 & 1 & 0 & 0 & 1 & 1 & 1 & -2 & -19 & -3 & 6 & -1 & 24 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 0 & -5 & 0 & -4 & 2 & -23 & -24 & -6 \\ 0 & 0 & -1 & -1 & -1 & 1 & -1 & 1 & 18 & 14 & 25 & 24 & -2 & -1 \\ 1 & 1 & 0 & 0 & -1 & 0 & 1 & 6 & 11 & 13 & 12 & -15 & -25 & -15 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & -2 & 33 & 34 & 14 & 25 & -21 & -12 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 15 & 3 & 7 & -24 & 22 & 6 & -15 \\ -1 & 1 & 0 & 0 & 0 & 0 & -1 & 12 & -20 & 0 & -21 & -7 & -2 & 25 \\ 1 & -1 & -1 & 0 & 1 & 0 & 0 & 15 & 11 & 20 & 6 & 1 & 15 & 21 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & -25 & -5 & 32 & -1 & -14 & 12 & -6 \\ 0 & -1 & 0 & 1 & 0 & 0 & -1 & -21 & -33 & 22 & 14 & -12 & 15 & 2 \end{pmatrix}.$$

Koliko su baze stvarno ortogonalne možemo vidjeti u njihovim Hamardovim omjerima,

$$\mathcal{H}(M_h^{NTRU}) = 0.075911 \quad \text{i} \quad \mathcal{H}(M_{red}^{NTRU}) = 0.87530.$$

Najmanji vektor u reduciranoj bazi je prvi stupac matrice M_{red}^{NTRU},

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

Iz ovog vektora dobijemo polinome

$$f'(x) = 1 + x - x^2 - x^4 - x^5 \quad \text{i} \quad g'(x) = 1 - x - x^3 + x^4.$$

Vidimo da ovo nisu polazni polinomi iz primjera (5.1.4), međutim oni su rotacije tih polinoma, tj.

$$f'(x) = -x \circledast f(x) \quad \text{i} \quad g'(x) = -x \circledast g(x).$$

Dakle, Eva može koristiti polinome $f'(x)$ i $g'(x)$ za dešifriranje.

6.4.2 Hibridni napad

Hibridni napad kombinira napad rešetkom i napad algoritmom kolizije. Poznati su nam svi javni parametri iz NTRUEncrypt-a i želimo pronaći tajni ključ (f, g). Izaberemo $N_1 < N$

te iz rešetke uklonimo $2N_1 \times 2N_1$ podrešetku L_1 iz centra rešetke L. Matrica M_h^{NTRU} koja odgovara rešetki L sada ima oblik

$$M_h^{NTRU} = \begin{pmatrix} I_N & 0 \\ \mathbf{h} & qI_N \end{pmatrix} = \begin{pmatrix} I_{N-N_1} & 0 & 0 \\ * & L_1 & 0 \\ * & * & qI_{N-N_1} \end{pmatrix},$$

gdje L_1 ima oblik

$$M_h^{NTRU} = \begin{pmatrix} I_{N_1} & 0 \\ \mathbf{h}_1 & qI_{N_1} \end{pmatrix}.$$

U ovoj matrici h_1 je dio matrice koju smo dobili cikličkim ponavljanjem javnog ključa. Blokovi unutar L_1 ne moraju nužno biti iste veličine.
Pretpostavimo da je napadaču potrebno k_1 bitova da bi reducirao rešetku L_1 eliminiranjem svih N_1 q vektora. Ovom eliminacijom L_1 postaje donjetrokutasta matrica, s vrijednostima na dijagonali $\{q^{\alpha_1}, q^{\alpha_2}, \ldots, q^{\alpha_{2N_1}}\}$, gdje je $\alpha_1 + \alpha_2 + \cdots + \alpha_{2N_1} = N_1$ i koeficijenti α_i su skoro linearno padajući, tj. vrijedi

$$1 \approx \alpha_1 > \cdots > \alpha_{2N_1} \approx 0.$$

Ova redukcija će se pokazati i na originalnoj rešetki L, čija će odgovarajuća matrica također biti donje trokutasta s dijagonalom oblika

$$\{1, 1 \ldots, 1, q^{\alpha_1}, q^{\alpha_2}, \ldots, q^{\alpha_{2N_1}}, q, q, \ldots, q\}.$$

Dakle, potrebno je k_1 bitova da bi se postigla ova redukcija, sa $\alpha_{2N_1} \approx 0$. Ako uzimamo $k_2 > k_1$ bitova, možemo dobiti $\alpha_{2N_1} = \alpha > 0$. Kako se k_2 povećava, tako se α povećava.
Kod algoritma kolizije (meeet-in-the-middle attack) koeficijenti ključa f dijele se na dva bloka veličine N_1 i $N - N_1$. Napadač pokušava pogoditi raspodjelu koeficijenata od f na ta dva bloka i potom iskoristiti reduciranu bazu rešetke L da bi provjerio ispravnost raspodjele. Raspodjelu je dovoljno napraviti na otprilike pola koeficijenata bloka $N - N_1$, te ih usporediti s preostalim dijelom. Vjerojatnost da će raspodjela u blokove biti dobra dana je sa $p_s(\alpha)$, gdje je $p_s(0) = 0$ i $p_s(\alpha)$ se monotono povećava kako se povećava α. Za vrijednost $\alpha = 0.182$ odgovarajuća vjerojatnost je $p_s(0.182) = 2^{-13}$ i dobivena je eksperimentalno, uz k_2 manji od 60.3. Općenito, ako za je za dobivanje vrijednosti $p_s(\alpha)$, potrebno k_2 bita, onda se broj bitova potrebnih u algoritmu kolizije za pretragu bloka $N - N_1$ smanjuje, kako se vrijednost $N - N_1$ smanjuje i $p_s(\alpha)$ povećava. Optimalan odabir parametara k i N_1 je zahtijevan problem. Napadač želi ove parametre odabrati tako da, ako vjerojatnost $p_s(\alpha)$ odgovara N_1, k_2 odgovara jačini napada algoritma kolizije.

Poglavlje 7

Digitalni potpisi i NTRUSign

NTRUEncrypt, kao i svi ostali kriptosustavi, služi za prenošenje poruka preko nesigurnih komunikacijskih kanala. Digitalni potpisi imaju sličnu ulogu kao i vlastoručni potpisi na papirima. Točnije, pretpostavimo da posjedujemo neki oblik digitalnog dokumenta (npr. tekstualni dokument) D. Želimo tom dokumentu dodati dodatnu informaciju D^{sign} koja će potvrditi njegovu autentičnost. Digitalni potpisi, kao i asimetrični kriptosustavi, koriste javne i privatne ključeve te algoritme koji ih međusobno povezuju. Za stvaranje digitalnog potpisa trebamo:

K^{Pri}: privatni ključ za potpis.

K^{Pub}: javni ključ za provjeru potpisa.

Potpis: algoritam koji prima digitalni dokument D i privatni ključ K^{Pri}, te vraća potpis D^{sign} za D.

Provjeri: algoritam koji prima dokument D, potpis D^{sign} i javni ključ K^{Pub}, te provjerava da li potpis odgovara dokumentu. Ovaj algoritam *nema* pristup privatnom ključu K^{Pri}.

Osnovni uvjeti koje sheme za digitalne potpise trebaju zadovoljavati su:

- Napadač, ako zna K^{Pub}, ne može u razumnom vremenu naći K^{Pri}, niti bilo koji drugi privatni ključ koji napravi isti potpis kao K^{Pri}.
- Napadač, ako zna K^{Pub}, te posjeduje niz dokumenata $D_1, \ldots, D_n$, zajedno s njihovim potpisima $D_1^{sign}, \ldots, D_n^{sign}$, ne može u razumnom vremenu naći valjan potpis za bilo koji dokument D koji nije u nizu $D_1, \ldots, D_n$.

Svaki puta kada napravimo novi digitalni potpis, napadač dobije novu informaciju o paru dokument/potpis. Drugi uvjet zapravo kaže da napadač, iako zna novi par, ne može ga

iskoristiti u svom napadu.
Većina shema za digitalne potpise potpisuje malu količinu podataka, otprilike od 80 do 100 bitova. Jako je neučinkovito potpisati veliki digitalni dokument D, ponajviše jer bi potpis poprimio veličinu samog dokumenta kojeg potpisujemo. Da bi ovo izbjegli koristimo hash funkciju

$$Hash\colon \text{(dokument proizvoljne veličine)} \to \{0,1\}^k$$

kojoj je teško pronaći inverz. Uz hash funkciju više ne potpisujemo dokument D nego $Hash(D)$. Provjera potpisa se također vrši na $Hash(D)$.

7.1 Digitalni potpisi pomoću rešetki

Ako nam je poznata dobra baza $\mathcal{B}$ za rešetku L, možemo uz pomoć Babaijevog algoritma riješiti CVP (ili barem apprCVP) u rešetki L za zadani vektor $d \in \mathbb{R}^n$. Rješenje $s \in L$ je dovoljno blizu d, te je digitalni potpis za dokument d. Bilo tko, jer je javna baza poznata, može provjeriti da je $s \in L$ i da je s blizu d, ali bez poznavanja privatne (dobre) baze pronalazak točke rešetke s' koja je dovoljno blizu d i koja bi mogla poslužiti kao potpis je težak. Kada potpisujemo dokumente algoritmima baziranim na rešetkama, naš dokument je zapravo vektor iz $\mathbb{R}^n$. Na dokument koji potpisujemo primijenimo neku *hash* funkciju da bi dobili kratki dokument od nekoliko stotina bitova. Upotrebljavamo hash funkciju čija su kodomena vektori u $\mathbb{Z}^n$ s koordinatama iz zadanog intervala.
Ovaj opisani postupak je *GGH* (Goldreich–Goldwasser–Halevi) digitalni potpis. Dakle, imamo tri dijela algoritma:

1. **Izračun ključeva**: Odaberemo dobru $v_1, \ldots, v_n$ i lošu $w_1, \ldots, w_n$ bazu za L. Objavimo javni ključ $w_1, \ldots, w_n$.

2. **Potpisivanje**: Pomoću Babaijevog algoritma i dobre baze izračunamo $s \in L$ koji je blizu $d \in \mathbb{Z}^n$ kojeg želimo potpisati. Izrazimo $s = a_1 w_1 + \cdots + a_n w_n$. Objavimo potpis $(a_1, \ldots, a_n)$.

3. **Provjera**: Izračunamo $s = a_1 w_1 + \cdots + a_n w_n$. Provjerimo je li s dovoljno blizu d.

Potrebno je odrediti koliko s mora biti dovoljno blizu d da bi bili zadovljni rezultatom. Dakle, treba nam $\epsilon > 0$ takav da ako je

$$\|s - d\| < \epsilon,$$

možemo reći da je potpis valjan i suprotno ako ovo nije zadovoljeno. Jedna mogućnost je eksperimentima utvrditi koliko dobro Babaijev algoritam rješava apprCVP te odabrati ϵ, drugi pristup je teorijski i koristi Gaussovu heuristiku. Kod drugog pristupa pretpostavi se

da Babaijev algoritam, uz dobru bazu, rješava apprCVP za faktor $\sqrt{dim}$. Dobar odabir za ϵ u rešetki L dimenzije n je tada:

$$\epsilon = \sqrt{n}\sigma(L) \approx \frac{n(\det L)^{1/n}}{\sqrt{2\pi\epsilon}},$$

Svaki put kada potpišemo novi dokument, par (d, s) (dokument/potpis) otkriva novu informaciju o ključu v. Najmanje što znamo je da dokument d uz ključ v daje potpis s. Kod GGH sheme možemo reći i više. Naime, koristimo Babaijev algoritam za rješavanje apprCVP uz bazu $v_1, \ldots, v_n$ i vektor d. Tada vektor $d - s$ ima oblik

$$d - s = \sum_{i=1}^{n} \epsilon_i(d, s)v_i \quad \text{gdje je} \quad |\epsilon_i(d, s)| \leq \frac{1}{2}.$$

Pretpostavimo da je veliki broj dokumenata potpisan s istim ključem,

$$(d_1, s_1), (d_2, s_2), \ldots, (d_N, s_N).$$

Napadač sad ima pristup velikom broju točaka iz fundamentalne domene

$$\mathcal{F} = \{\epsilon_1 v_1 + \epsilon_2 v_2 + \cdots + \epsilon_n v_n : -\frac{1}{2} < \epsilon_1, \epsilon_2, \ldots, \epsilon_n \leq \frac{1}{2}\}$$

koje su razapete preko dobre baze $v_1, v_2, \ldots, v_n$. Ove točke se mogu upotrijebiti da bi se, barem aproksimativno, odredili vektori baze.

7.2 NTRUSign

NTRUSign algoritam za digitalne potpise napravljen je između 2001. i 2003. godine. Uz izumitelje NTRUEncrypt-a u izradi ovog algoritmu još su sudjelovali N. Howgrave-Graham i W. Whyte.
Algoritmi redukcije rešetke su učinkoviti kod napada na GGH digitalne potpise, posebno u nižim dimenzijama. U većim dimenzijama veličina ključa koji se koristi za GGH postaje prevelika (oko $O(n^2 \log n)$) za implemetaciju. Zato se koristi NTRU rešetka i njezina baza za javni ključ (veličine $O(n \log n)$).
NTRU rešetka L^{NTRU} dimenzije $2N$ sadrži kratki vektor (ključ) (f, g), kao i N cikličkih rotacija $(x^i \circledast f, x^i \circledast g)$, $0 \leq i \leq N$. Ovih N kratkih vektora tvori pola dobre baze. Da bi potpisali dokument D potrebna je dobra baza dimenzije $2N$, a ne samo ona koja se sastoji od N dobrih vektora. Općenito nije moguće pronaći "punu" bazu u kojoj su vektori duljine $O(\sqrt{N})$, tj. duljine od (f, g). Međutim, postoji komplementarna baza (F, G) takva da vrijedi:

$$f(x) \circledast G(x) - g(x) \circledast F(x) = q, \quad \|F\| = O(N) \quad \text{i} \quad \|G\| = O(N).$$

7.2.1 Opis algoritma

Pomoću parametara (N, q, d) i privatnih polinoma $f(x), g(x) \in T(d+1, d)$ izračunamo javni ključ

$$h(x) \equiv f(x)^{-1} \circledast g(x) \pmod{q}.$$

Da bi potpisali digitalni dokument $D = (D_1, D_2)$ potrebni su nam parovi polinoma (f, g) i (F, G). Izračunamo polinome

$$v_1(x) = \left\lfloor \frac{D_1(x) \circledast G(x) - D_2(x) \circledast F(x)}{q} \right\rceil,$$

$$v_2(x) = \left\lfloor \frac{-D_1(x) \circledast g(x) + D_2(x) \circledast f(x)}{q} \right\rceil.$$

Potpis za dokument D će biti polinom

$$s(x) = v_1 \circledast f(x) + v_2 \circledast F(x).$$

Za provjeru izračunamo polinom $t(x)$, čije koeficijente biramo što bliže koeficijentima $D_2(x)$,

$$t(x) \equiv h(x) \circledast s(x) \pmod{q}.$$

Preostaje provjeriti je li vektor (s, t) dovoljno blizu vektoru početnog dokumenta $D = (D_1, D_2)$.

Rešetka L_h^{NTRU} ima dvije baze: dobru $\left(\begin{smallmatrix} f & g \\ F & G \end{smallmatrix}\right)$ i lošu $\left(\begin{smallmatrix} 1 & h \\ 0 & q \end{smallmatrix}\right)$. Dokument D raspisujemo u dobroj bazi tako da sustav:

$$(D_1, D_2) = (u_1, u_2) \begin{pmatrix} f & g \\ F & G \end{pmatrix}$$

riješimo po (u_1, u_2). Dobivamo:

$$(u_1, u_2) = (D_1, D_2) \begin{pmatrix} f & g \\ F & G \end{pmatrix}^{-1} = (D_1, D_2) \begin{pmatrix} G/q & -g/q \\ -F/q & f/q \end{pmatrix}.$$

Kako koordinate od u_1 i u_2 nisu cjelobrojne, zaokružimo ih na najbliži cijeli broj,

$$v_1 = \lfloor u_1 \rceil \quad \text{i} \quad v_2 = \lfloor u_2 \rceil,$$

i izračunamo vektor

$$(s, t) = (v_1, v_2) \begin{pmatrix} f & g \\ F & G \end{pmatrix},$$

koji bi trebao biti dovoljno blizu D.

Kako je $(s, t) \in L_h^{NTRU}$, nepotrebno je da i s i t budu javni jer se t može izračunati preko s i javnog ključa h. No, ako su i s i t javni, svejedno treba provjeriti je li vektor (s, t) u L_h^{NTRU}, odnosno provjeriti je li $h \circledast s$ mod q jednako t.

7.3 Pronalazak komplementarne baze

U NTRUSign-u koristimo vektor F i G kao nadopunu dobre baze. Opisujemo algoritam koji pronalazi te vektore.

Definicija 7.3.1. *Neka su $a(x)$ i $b(x)$ polinomi s racionalnim koeficijentima. Ako je njihov najveći zajednički djelitelj* 1*, tada po Euklidovom algoritmu (Propozicija 1.2.4) znamo da postoje polinomi $A(x)$ i $B(x)$ takvi da vrijedi*

$$a(x)A(x) + b(x)B(x) = 1.$$

Koeficijenti od $A(x)$ i $B(x)$ su općenito racionalni brojevi. Najmanji prirodan broj za kojeg vrijedi

$$a(x)A(x) + b(x)B(x) = R \quad sa \quad A(x), B(x) \in \mathbb{Z}[x],$$

zove se rezultanta od $a(x)$ i $b(x)$. Oznaka je $Res(a(x), b(x))$.

Koraci algoritma su sljedeći:

- Pronaći polinome $f_1(x), f_2(x), g_1(x), g_2(x) \in \mathbb{Z}[x]$ i prirodne brojeve Res_f i Res_g takve da
$$f_1(x)f(x) + f_2(x)(x^N - 1) = Res_f,$$
$$g_1(x)g(x) + g_2(x)(x^N - 1) = Res_g.$$
- $\gcd(Res_f, Res_g) = 1$, inače ne možemo provesti algoritam.
Pronađimo cijele brojeve S_f i S_g takve da
$$S_f Res_f + S_g Res_g = 1.$$
- Neka je $A(x) = qS_f f_1(x)$ i $B(x) = -qS_g g_1(x)$. Uočimo
$$A(x) \circledast f(x) - B(x) \circledast g(x) = q \quad \text{u prstenu} \quad \mathbb{Z}[x]/(x^N - 1).$$
- Izračunamo inverze $f(x)^{-1}$ i $g(x)^{-1}$ u $\mathbb{R}[x]/(x^N - 1)$ i definiramo
$$C(x) = \left\lfloor \frac{1}{2}\Big(B(x) \circledast f(x)^{-1} + A(x) \circledast g(x)^{-1}\Big) \right\rceil.$$
- Traženi polinomi su:
$$F(x) = B(x) - C(x) \circledast f(x) \quad \text{i} \quad G(x) = A(x) - C(x) \circledast g(x).$$

U dokazu sljedeće propozicije potreban nam je pomoćni rezultat:

Lema 7.3.2. *Fiksirajmo vektor $a \in \mathbb{R}^N$, $t > 0$ i neka je $b \in \mathbb{R}^N$ vektor čiji su koeficijenti uniformno izabrani između $-t$ i t. Tada za većinu izbora vektora b vrijedi*

$$\|a \circledast b\| \approx \|a\| \, \|b\|. \tag{7.1}$$

Propozicija 7.3.3. *Fiksirajmo parametre (N, q, d), $q = O(n)$ i $d = O(n)$. Neka su $f(x)$ i $g(x)$ ternarni polinomi iz $T(d_1, d_2)$, gdje je $d_1 \approx d_2 \approx d$. Nadalje, neka su $f(x)$ i $g(x)$ relativno prosti s polinomom $x^N - 1$, i neka su njihove rezultante*

$$Res_f = Res(f(x), x^N - 1) \quad i \quad Res_g = Res(g(x), x^N - 1)$$

relativno prosti cijeli brojevi. Tada prethodno opisani algoritam pronalazi polinome $F(x), G(x) \in \mathbb{Z}[x]/(x^N - 1)$ koji zadovoljavaju

$$f(x) \circledast G(x) - g(x) \circledast F(x) = q, \tag{7.2}$$

i za njihove norme vrijedi:

$$\|F\| = O(N) \quad i \quad \|G\| = O(N). \tag{7.3}$$

Dokaz. Prethodno opisanim algoritmom dobiju se polinomi $F(x)$ i $G(x)$ koji zadovoljavaju

$$f \circledast G - g \circledast F = f \circledast (A - C \circledast g) - g \circledast (B - C \circledast f) = f \circledast A - g \circledast B = q,$$

čime smo pokazali (7.2).
Za dokaz (7.3) zapišemo

$$B \circledast f^{-1} = u_B + v_B,$$

gdje $u_B = \lfloor B \circledast f^{-1} \rceil$ ima cjelobrojne koeficijente, a koeficijenti od v_B su raspoređeni između $-\frac{1}{2}$ i $\frac{1}{2}$. Tada je

$$B - u_B \circledast f = B - (B \circledast f^{-1} - v_B) \circledast f = v_B \circledast f,$$

pa je po prethodnoj lemi

$$\|B - u_B \circledast f\| = \|v_B \circledast f\| \approx \|v_B\| \, \|f\| \approx \sqrt{N/12} \, \|f\| \approx \sqrt{Nd/6}.$$

Upotrijebili smo činjenicu da je norma vektora, čiji su koeficijenti raspoređeni između $-\frac{1}{2}$ i $\frac{1}{2}$, približno $\sqrt{N/12}$. Također, kako je $f \in \mathcal{T}(d_1, d_2)$, vrijedi $\|f\| = \sqrt{d_1 + d_2} \approx \sqrt{2d}$. Slično, za $u_A = \lfloor A \circledast g^{-1} \rceil$ vrijedi

$$\|A - u_A \circledast g\| \approx \sqrt{Nd/6}.$$

Iz algoritma imamo polinom C,

$$C = \left\lfloor \frac{1}{2}\left(B \circledast f^{-1} + A \circledast g^{-1}\right) \right\rceil.$$

Tvrdimo da je

$$B \circledast f^{-1} \approx A \circledast g^{-1}.$$

Uočimo da f i g zadovoljavaju uvjete prethodne leme,

$$1 = \|1\| = \|f \circledast f^{-1}\| \approx \|f\|\,\|f^{-1}\| \quad \text{i} \quad 1 = \|1\| = \|g \circledast g^{-1}\| \approx \|g\|\,\|g^{-1}\|.$$

Sada $A \circledast f - B \circledast g = q$ zapišemo kao

$$A \circledast g^{-1} - B \circledast f^{-1} = qf^{-1} \circledast g^{-1},$$

pa je

$$\|A \circledast g^{-1} - B \circledast f^{-1}\| = \|qf^{-1} \circledast g^{-1}\| \approx q\|f^{-1}\|\,\|g^{-1}\| \approx \frac{q}{\|f\|\,\|g\|} \approx \frac{q}{2d}.$$

Dakle,

$$A \circledast g^{-1} - B \circledast f^{-1} = w_C, \quad \text{sa} \quad \|w_C\| \approx q/2d.$$

Za polinom C vrijedi

$$C = u_B + w'_C = u_A + w''_C, \quad \text{sa} \quad \|w'_C\| \approx \|w''_C\| \approx q/2d.$$

Iz ovog dobivamo,

$$\|F\| = \|B - C \circledast f\| \lessapprox \sqrt{\frac{Nd}{6}} + \frac{q}{2d} \quad \text{i} \quad \|G\| = \|A - C \circledast g\| \lessapprox \sqrt{\frac{Nd}{6}} + \frac{q}{2d}.$$

Kako je izraz $\sqrt{Nd/6}$ puno veći od izraza $q/2d$, imamo $\|F\| = O(N)$ i $\|G\| = O(N)$. □

7.4 Napad na NTRUSign

Kao i kod *GGH* sheme, svakim novim potpisanim dokumentom otkrivamo nove informacije o ključu. Unutar vektora $a = [a_0, \ldots, a_{N-1}]$ promijenimo poredak koordinata $[a_0, a_{N-1}, \ldots, a_2, a_1]$ i to označimo sa $\overline{a}$. Za svaki dokument i njegov potpis računamo vrijednosti:

$$D_{1,k} - s_k = \epsilon_k \circledast f + \delta_k \circledast F \quad \text{i} \quad D_{2,k} - t_k = \epsilon_k \circledast g + \delta_k \circledast G,$$

gdje su koordinate vektora ϵ_k i δ_k raspoređene unutar intervala $-\frac{1}{2}$ i $\frac{1}{2}$. Sada se mogu računati različiti prosjeci. Npr. za (dovoljno velik) n različitih dokumenat/potpis parova može se pokazati da je

$$\frac{1}{n}\sum_{k=1}^{n}(D_{1,k} - s_k) \circledast (\overline{D_{1,k} - s_k}) \approx \frac{1}{12}(f \circledast \overline{f} + F \circledast \overline{F}).$$

Slično se mogu pronaći i vrijednosti:

$$f \circledast \overline{f} + F \circledast \overline{F}, \quad g \circledast \overline{f} + G \circledast \overline{F}, \quad f \circledast \overline{g} + F \circledast \overline{G} \quad \text{i} \quad g \circledast \overline{f} + G \circledast \overline{F}.$$

Ovi izračuni su ekvivalentni izračunu Grammove matrice za dobru bazu privatnog ključa. Nguyen-Regev algoritam za GGH je prilagođen za napad na NTRUSign i pronalazi ključ već uz nekoliko stotina parova dokumenat/potpis. NTRUSign se stoga smatra nesigurnim. Međutim, malim preturbacijama unutar svakog potpisa ovaj se problem zaobilazi.
Općenito, kod preturbacija, nakon što na dokument D, koji želimo potpisati, primijenimo neku hash funkciju, dobijemo novu poruku m. Prije nego što na poruci m upotrijebimo privatni ključ, odaberemo vektor pogreške e te zapravo potpisujemo $m + e$.

Bibliografija

[1] D. Bakić, *Predavanja iz kolegija Linearna algebra*, `http://web.math.pmf.unizg.hr/nastava/la/dodatno.html`.

[2] A. Dujella, *Uvod u teoriju brojeva*, skripta, `http://web.math.pmf.unizg.hr/~duje/utb/utblink.pdf`.

[3] A. Dujella i M. Maretić, *Kriptografija*, Udžbenici sveučilišta u Zagrebu, Element, 2007, ISBN 9789531975650.

[4] J. Hoffstein, J. Pipher i J.H. Silverman, *An Introduction to Mathematical Cryptography*, Undergraduate Texts in Mathematics, Springer, 2008, ISBN 9780387779942.

[5] N. Howgrave-Graham, *A hybrid lattice-reduction and meet-in-the-middle attack against NTRU*, Advances in cryptology—CRYPTO 2007, Lecture Notes in Comput. Sci., sv. 4622, Springer, Berlin, 2007, str. 150–169.

[6] A.J. Menezes, P.C. van Oorschot i S.A. Vanstone, *Handbook of applied cryptography*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1997, ISBN 0-8493-8523-7, With a foreword by Ronald L. Rivest.

[7] P.Q. Nguyen i B. Vallée, *The LLL Algorithm: Survey and Applications*, Information Security and Cryptography, Springer Berlin Heidelberg, 2009, ISBN 9783642022944.

[8] B. Širola, *Algebarske strukture*, skripta, `http://web.math.pmf.unizg.hr/nastava/alg/predavanja/ASpred.pdf`.

Sažetak

Većina današnjih kriptosustava s javnim ključem temelji se na problemima faktorizacije velikih brojeva i diskretnog logaritma. Pojavom kvantnih računala, ovi problemi bi postali rješivi. U ovom radu je opisan NTRU kriptosustav, kriptosustav koji bi ostao teško rješiv problem i za kvantna računala. Dijeli se na dva algoritma: NTRUEncrypt i NTRUSign.

Posebno, za šifriranje koristimo NTRUEncrypt. Ovaj kriptosustav temelji se na teoriji rešetaka, te problemu pronalaska najkraćeg vektora unutar rešetke. Sve operacije su unutar prstena polinoma s cjelobrojnim koeficijentima na kojima se dodatno definira konvolucijsko množenje. Najučinkovitiji napadi na NTRUEnrypt kriptosustav su oni koji koriste strukturu rešetki. Tu se posebno ističe LLL algoritam, te njegova blokovna verzija BKZ-LLL algoritam. LLL algoritam reducira bazu rešetke na skoro ortogonalnu bazu, te se u takvoj poboljšanoj bazi rješava problem najkraćeg vektora.

NTRUSign je kriptosustav s javnim ključem koji se koristi za digitalne potpise. Općenito, digitalni potpisi imaju sličnu ulogu kao vlastoručni potpisi na papiru. Veliki broj potpisanih dokumenata odaje puno informacija o privatnom ključu. Da bi se ovo spriječilo uvode se preturbacije unutar poruke prije samog potpisivanja.

Summary

Most of today's public key cryptosystems are based on two hard mathematical problems: factoring large integers and discrete logarithm problem. With possible advent of quantum computers, these problems can become solvable. This paper describes the NTRU cryptosystem, cryptosystem that is known to not be vulnerable to quantum computer based attacks. It consists of two algorithms: NTRUEncrypt, which is used for encryption, and NTRUSign, which is used for digital signatures.

NTRUEncrypt is lattice-based public key cryptosystem and is based on the shortest vector problems. Operations are based on objects in a polynomial ring with convolution multiplication and all polynomials in the ring have integer coefficients. Breaking the NTRUEncrypt cryptosystem is related with problem of lattice reduction. Some of lattice reduction algorithms used to break NTRUEncrypt are LLL algorithm and BKZ-LLL algorithm. LLL algoritham calculates an LLL reduced basis, i.e. short and nearly orthonogal basis.

NTRUSign is a public key cryptography digital signature algorithm. A digital signature is a scheme for verifying the authenticity of a digital document. Transcript of NTRUSign signatures leaks information about private key. To prevent leaks is recommended to use preturbation in message before the signature itself is calculated.

Životopis

Valentina Pribanić rođena je 15.09.1990. godine u Ogulinu. Osnovnu školu pohađa od 1997. do 2005. godine u Tounju. Po završetku osnovne škole upisuje se u Gimnaziju Bernardina Frankopana u Ogulinu, smjer Opća gimnazija. Srednju školu završava 2008. godine, oslobođena od polaganja mature i kao učenik generacije.
Iste godine upisuje preddiplomski studij matematike na Prirodoslovnom matematičkom fakultetu u Zagrebu. Po završetku preddiplomskog studija, 2012. godine, stječe titulu univ. bacc. math.
Diplomski studij Primijenjene matematike upisuje 2012. godine, također na Prirodoslovno matematičkom fakultetu u Zagrebu.