Hilat ja kryptografia

Pro gradu -tutkielma
Oskar Olaussen
2308124
Matemaattisten tieteiden laitos
Oulun yliopisto
Syksy 2017

Sisältö

Johdanto

Hilan käsite on helpointa ymmärtää analogian kautta. Samoin kuin vektoriavaruudella hilalla on kantavektorit, mutta toisin kuin vektoriavaruuden tapauksessa, joka virittyy, kun kantavektoreita kerrotaan reaaliluvuilla, hila virittyy kantavektoreita kokonaisluvuilla kertomalla.
Tässä tutkielmassa käsitellään hiloja julkisen avaimen salausmenetelmien pohjana. Julkisen avaimen salausmenetelmän periaatteena on, että viestin lähettäjän ja vastaanottajan ei tarvitse etukäteen vaihtaa salausavainta, vaan sopia käytettävä julkisen avaimen salausmenetelmä. Tämän jälkeen lähettäjä salaa viestin julkisella avaimella ja vastaanottaja purkaa salauksen salaisella avaimellaan. Jotta tämä on mahdollista, on julkisen ja salaisen avaimen välillä oltava jokin matemaattinen yhteys. Menetelmien turvallisuus perustuu siihen, että salaisen avaimen ratkaiseminen julkisesta avaimesta on vaikeaa.
Tutuimpien julkisen avaimen salausmenetelmien turvallisuus perustuu kokonaislukujen alkutekijöihin jakoon, diskreetin logaritmin ongelmaan tai elliptisiin käyriin. Erityisesti 1990-luvulta eteenpäin on kuitenkin kehitetty useita salausmenetelmiä, joiden turvallisuus perustuu hilaongelmiin. Tällaisia ongelmia ovat esimerkiksi hilan lyhyimmän vektorin tai tiettyä vektoria lähinnä olevan vektorin löytäminen. Hilaongelmiin perustuvista salausmenetelmistä tässä tutkielmassa käsitellään GGH- ja NTRU-salausmenetelmiä. Hilaongelmiin perustuvien menetelmien kehittämistä motivoi se, että niiden vaatimat laskutoimitukset ovat usein nopeampia kuin muiden menetelmien. Lisäksi on hyvä, että on olemassa useisiin eri ongelmiin perustuvia salausmenetelmiä, jolloin yhden ongelman ratkeaminen ei tee kaikista menetelmistä turvattomia.
Tutkielmassa käsitellään myös salausmenetelmiä vastaan kehitettyjä hyökkäyksiä. Useat näistä hyökkäyksistä perustuvat LLL-algoritmin käyttöön. LLL-algoritmi on tehokas työkalu lyhyiden vektorien löytämiseen hilasta. Lisäksi sillä on muita ominaisuuksia, jotka ovat hyökkäyksien kannalta hyödyllisiä. Tutkielmassa osoitetaan LLL-algoritmin oikeellisuus ja päättyminen.
Kaikki tutkielman esimerkit ovat tutkielman tekijän muotoilemia. Tutkielmassa on käytetty pääasiallisena lähteenä teosta *An Introduction to Mathematical Cryptography* [7].

1 Lineaarialgebra ja hila

Ennen hilan määrittelyä tutustutaan lineaarialgebran perusteisiin. Lukijan oletetaan tuntevan peruskäsitteistö siinä laajuudessa, missä ne on esitetty Oulun yliopiston kursseilla Lineaarialgebra I ja II. Nämä asiat löytyvät kurssien luentomonisteista [9] ja [16].

1.1 Lineaarialgebraa

Kerrataan aluksi joitain tuttuja peruskäsitteitä.

Määritelmä 1.1. Joukon $\mathbb{R}^n$ osajoukko V on *vektoriavaruus*, jos kaikilla $v_1, v_2 \in V$ ja $\alpha_1, \alpha_2 \in \mathbb{R}$ pätee

$$\alpha_1 v_1 + \alpha_2 v_2 \in V.$$

Vektoriavaruus V on siis suljettu yhteenlaskun ja joukon $\mathbb{R}$ alkioilla kertomisen suhteen.

Määritelmä 1.2. Olkoot $v_1, v_2, \ldots, v_m \in V$. Vektori

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_m v_m$$

on vektoreiden $v_1, v_2, \ldots, v_m$ *lineaarikombinaatio*. Kaikkien lineaarikombinaatioiden joukko on joukon $\{v_1, v_2, \ldots, v_m\}$ virittämä *aliavaruus*.

Määritelmä 1.3. Vektorit $v_1, v_2, \ldots, v_m$ ovat *lineaarisesti riippumattomat*, jos yhtäsuuruus

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_m v_m = 0$$

pätee vain, kun

$$\alpha_1 = \alpha_2 = \ldots = \alpha_m = 0.$$

Mitään lineaarisesti riippumattomista vektoreista ei siis voida esittää muiden vektoreiden summana.

Määritelmä 1.4. Vektoriavaruuden V *kanta* koostuu lineaarisesti riippumattomista vektoreista, joiden virittämä aliavaruus on V.

Vektoriavaruuden V mikä tahansa vektori w voidaan esittää kantavektoreiden lineaarikombinaationa.

Esimerkki 1.5. Osoitetaan, että vektorit $v_1 = (-1, 2)$ ja $v_2 = (0, -1)$ muodostavat kannan avaruudelle $\mathbb{R}^2$. Tarkistetaan ensin, että vektorit ovat lineaarisesti riippumattomat. Nyt

$$\alpha_1(-1,2) + \alpha_2(0,-1) = (0,0) \implies \begin{cases} -\alpha_1 & = 0 \\ 2\alpha_1 & = \alpha_2 \end{cases} \implies \begin{cases} \alpha_1 & = 0 \\ \alpha_2 & = 0 \end{cases}.$$

Vektorit ovat siis lineaarisesti riippumattomat. Lisäksi vektorit v_1 ja v_2 virittävät avaruuden $\mathbb{R}^2$, sillä jos mielivaltainen $w = (x, y) \in \mathbb{R}^2$, niin

$$\alpha_1(-1,2)+\alpha_2(0,-1) = (x,y) \implies \begin{cases} -\alpha_1 & = x \\ 2\alpha_1 - \alpha_2 & = y \end{cases} \implies \begin{cases} \alpha_1 & = -x \\ \alpha_2 & = -y - 2x \end{cases}.$$

Jokainen avaruuden $\mathbb{R}^2$ vektori voidaan siis esittää lineaarisesti riippumattomien vektoreiden v_1 ja v_2 lineaarikombinaationa. Näin ollen nämä vektorit ovat avaruuden $\mathbb{R}^2$ eräs kanta.

Voidaan osoittaa, että kaikille vektoriavaruuksille on olemassa kanta, ja että vektoriavaruuden eri kannoissa on sama määrä kantavektoreita. Tätä määrää kutsutaan vektoriavaruuden *dimensioksi*. Todistukset kuitenkin eksyvät kauas tämän tutkielman aiheesta, joten ne sivuutetaan.

Määritelmä 1.6. Olkoot v,$w \in \mathbb{R}^n$ ja olkoon

$$v = (x_1, x_2, \ldots, x_n) \text{ ja } w = (y_1, y_2, \ldots, y_n).$$

Vektoreiden *sisätulo* on

$$v \cdot w = x_1y_1 + x_2y_2 + \ldots + x_ny_n.$$

Vektorin *pituus* tai *euklidinen normi* on

$$\|v\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}.$$

Vektorin L_1*-normi* on

$$L_1(v) = \sum_{i=1}^{n} |x_i|.$$

Selvästi huomataan, että

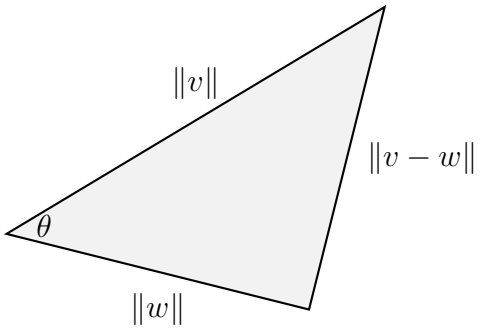
$$v \cdot v = \|v\|^2.$$

Lause 1.7. *Olkoot vektorit $v, w \in \mathbb{R}^n$ ja θ vektoreiden välinen kulma. Tällöin*

$$v \cdot w = \|v\|\|w\| \cos(\theta).$$

Todistus. Todistetaan lause tilanteessa, jossa vektoria ei saada toisesta skalaarilla kertomalla. Koulugeometriasta tutun kosinilauseen mukaan

$$\|v - w\|^2 = \|v\|^2 + \|w\|^2 - 2\|v\|\|w\| \cos(\theta).$$



Toisaalta pätee

$$\|v-w\|^2 = (v-w)\cdot(v-w) = v\cdot v - 2(v\cdot w) + w\cdot w = \|v\|^2 + \|w\|^2 - 2(v\cdot w).$$

Yhdistämällä nämä tulokset saadaan

$$\|v\|^2 + \|w\|^2 - 2\|v\|\|w\| \cos(\theta) = \|v\|^2 + \|w\|^2 - 2(v \cdot w).$$

Yhtälöä sieventämällä saadaan väite

$$v \cdot w = \|v\|\|w\| \cos(\theta).$$

□

Määritelmä 1.8. Kanta $v_1, v_2, \ldots, v_n$ on *ortogonaalinen*, jos

$$v_i \cdot v_j = 0 \ \forall i \neq j.$$

Jos lisäksi

$$\|v_i\| = 1 \ \forall \ i,$$

niin kanta on *ortonormaali.*

Myöhemmin esiteltävissä sovelluksissa kannan ortogonaalisuus on tärkeässä roolissa. Esitelläänkin Gram-Schmidt algoritmi, jolla ortogonaalinen kanta voidaan luoda, kun jokin kanta on tiedossa.

Lause 1.9 (Gram-Schmidt algoritmi)**.** *Olkoon $v_1, v_2, \ldots, v_n$ vektoriavaruuden $V \subset \mathbb{R}^m$ kanta. Tällöin seuraava algoritmi antaa ortogonaalisen kannan $v_1^*, v_2^*, \ldots, v_n^*$ vektoriavaruudelle V.*

Asetetaan $v_1 = v_1^*$ ja $i = 2$.
while $i \leq n$ **do**
  Lasketaan $\mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2}$ kaikille $1 \leq j < i$.
  Asetetaan $v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*$.
  $i = i + 1$.
end while
return $v_1^*, v_2^*, \ldots, v_n^*$.

Kantojen $v_1, v_2, \ldots, v_i$ ja $v_1^*, v_2^*, \ldots, v_i^*$ kaikkien lineaarikombinaatioiden joukot ovat samat.

Todistus. Todistetaan ortogonaalisuus induktiolla. Oletetaan, että vektorit $v_1^*, \ldots, v_{i-1}^*$ ovat keskenään ortogonaaliset ja osoitetaan vektorin v_i^* ortogonaalisuus niiden kanssa. Olkoon $k < i$. Tällöin

$$v_i^* \cdot v_k^* = \big(v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*\big) \cdot v_k^* = v_i \cdot v_k^* - \mu_{i,k} \|v_k^*\|^2 = 0.$$

Tässä toinen yhtäsuuruus seuraa siitä, että $v_k^* \cdot v_j^* = 0$ kaikilla $j \neq k$, ja kolmas yhtäsuuruus luvun $\mu_{i,k}$ määritelmästä.
Todistetaan lopuksi väite lineaarikombinaatioiden joukoista. Vektorin v_i^* määritelmästä nähdään, että vektori v_i kuuluu vektoreiden $v_1^*, v_2^*, \ldots, v_i^*$ lineaarikombinaatioiden joukkoon, sillä

$$v_i = \sum_{j=1}^{i-1} \mu_{i,j} v_j^* - v_i^*.$$

Osoitetaan sisältyvyys toiseen suuntaan induktiolla. Oletetaan, että vektorit $v_1^*, \ldots, v_{i-j}^*$ kuuluvat vektoreiden $v_1, \ldots, v_{i-1}$ lineaarikombinaatioiden joukkoon, ja osoitetaan, että v_i^* kuuluu vektoreiden $v_1, \ldots, v_i$ lineaarikombinaatioiden joukkoon. Määritelmästä

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*,$$

joten v_i^* kuuluu vektoreiden $v_1^*, \ldots, v_{i-1}^*, v_i$ lineaarikombinaatioiden joukkoon. Näin ollen väite seuraa induktio-oletuksesta. □

Algoritmien analyysissä tuloksen oikeellisuuden lisäksi on tärkeää tutkia algoritmin *aikakompleksisuutta*. Sillä tarkoitetaan algoritmissa suoritettavien operaatioiden määrää. Tätä varten otetaan käyttöön niin sanottu iso $\mathcal{O}$ -notaatio.

Määritelmä 1.10. Merkintä $g(x) = \mathcal{O}(f(x))$ tarkoittaa, että on olemassa reaaliluvut $M > 0$ ja $x_0 > 0$, siten että $g(x) \leq Mf(x)$ kaikilla $x \geq x_0$.

Funktio $f(x)$ siis asettaa ylärajan funktion $g(x)$ kasvunopeudelle. Algoritmien analyysissä $f(x)$ on suoritettavien operaatioiden määrä.

Lause 1.11. *Olkoon polynomi $g(x)$ astetta n. Tällöin $g(x) = \mathcal{O}(x^n)$.*

Todistus. Polynomi $g(x)$ on astetta n, eli

$$g(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0, \text{ missä } a_i \in \mathbb{R} \text{ kaikilla } i = 1, 2, \ldots, n.$$

Kun $x \geq 1$, niin

$$\begin{aligned} g(x) &= a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0 \\ &\leq |a_n|x^n + |a_{n-1}|x^{n-1} + \cdots + |a_0| \\ &\leq |a_n|x^n + |a_{n-1}|x^n + \cdots + |a_0| \\ &= x^n(|a_n| + |a_{n-1}| + \cdots + |a_0|). \end{aligned}$$

Valitsemalla $M = |a_n| + |a_{n-1}| + \cdots + |a_0|$ ja $x_0 = 1$ väite on todistettu. □

Gram-Schmidt algoritmin aikakompleksisuusluokka on $\mathcal{O}(mn^2)$. Algoritmin rivillä kolme suoritetaan $i-1$ kertaa kaksi pistetuloa ja yksi jakolasku. Pistetulossa on $m-1$ yhteenlaskua ja m kertolaskua. Näin ollen rivillä kolme suoritetaan $(i-1)(2(2m-1)+1)$ operaatiota. Rivillä neljä suoritetaan $i-1$ kertolaskua, $i-2$ yhteenlaskua ja yksi vähennyslaskua. Vektorin ortogonalisointiin menee siis kokonaisuudessaan

$$(i-1)(2(2m-1)+1) + i - 1 + i - 2 + 1 = 4mi - 4m + i - 1$$

operaatiota. Pahimmillaan (kun $i = n$) yhden vektorin ortogonalisointiin menee siis $4mn+4m+n-1$ operaatiota. Koska ortogonalisoitavia vektoreita on n, niin algoritmissa suoritettavien operaatioiden määrä on ylhäältä rajoitettu luvulla

$$n(4mn + 4m + n - 1) = 4mn^2 + 4mn + n^2 - n.$$

Algoritmin aikakompleksisuusluokka on siis $\mathcal{O}(4mn^2 + 4mn + n^2 - n) = \mathcal{O}(mn^2)$, sillä Lauseen 1.11 perusteella luokka määräytyy korkeimman asteen termin mukaan.

Määritelmä 1.12. Olkoon $A \in \mathbb{R}^{n\times n}$ matriisi. Matriisin A *determinantti* voidaan kehittää rivin tai sarakkeen mukaan. Jos determinantti kehitetään rivin i mukaan, niin

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij}),$$

missä $\det(A_{ij})$ on alkion a_{ij} *alideterminantti*, joka saadaan poistamalla rivi i ja sarake j. Reaaliluvun a determinantti $\det(a) = a$.

Todetaan seuraavat tulokset ilman todistusta.

- $\det(AB) = \det(A)\det(B)$.
- $\det(A^{-1}) = \frac{1}{\det(A)}$
- Olkoon $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$. Jos $B = \begin{pmatrix} m_1a_{11} & m_1a_{12} & \cdots & m_1a_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ m_na_{n1} & m_na_{n2} & \cdots & m_na_{nn} \end{pmatrix}$, niin $\det(B) = m_1m_2\cdots m_n \det(A)$.
- $\det(A) \neq 0$ jos ja vain jos on olemassa käänteismatriisi A^{-1}.

Määritelmä 1.13. Kokonaislukumatriisi A on *unimodulaarinen*, jos $\det(A) = \pm 1$.

1.2 Hila

Määritelmä 1.14. Olkoon vektorit $v_1, \ldots, v_n \in \mathbb{R}^m$ lineaarisesti riippumattomia. Tällöin vektoreiden $v_1, \ldots, v_n$ virittämä *hila* on joukko

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

Hila on selvästi analoginen vektoriavaruuden kanssa, kuitenkin sillä erolla, että hilaan kuuluvat ne lineaarikombinaatiot, joissa vektorien kertoimet ovat kokonaislukuja.
Samoin kuin vektoriavaruuden tapauksessa, kutsutaan hilan virittäviä vektoreita kutsutaan *kannaksi*, kaikissa hilan kannoissa on sama määrä vektoreita ja tätä määrää kutsutaan hilan *asteeksi*.

Lause 1.15. *Olkoon $v_1, \ldots, v_n$ hilan L kanta. Vektorit $w_1, \ldots, w_n \in L$ ovat hilan L kanta jos ja vain jos on olemassa unimodulaarinen matriisi A, siten että $AV = W$, missä matriisin V riveinä ovat vektorit v_i ja vastaavasti matriisin W riveinä ovat vektorit w_i.*

Lauseen 1.15 todistus on esitetty päälähteessä [7] luonnosmaisena. Seuraava todistus on tutkielman tekijän sen pohjalta itse muotoilema.

Todistus. $\Longrightarrow$
Vektorit $v_1, \ldots, v_n$ ovat hilan L kanta, joten vektorit $w_1, \ldots, w_n$ voidaan esittää kannan lineaarikombinaatioina:

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

Tämä on yhtäpitävää matriisiyhtälön $AV = W$ kanssa, missä

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

Vektorit $w_1, \ldots, w_n$ kuuluvat hilaan, joten kertoimet a_{ij} ovat kokonaislukuja. Jotta $w_1, \ldots, w_n$ voi olla hilan kanta, niin vektorit $v_1, \ldots, v_n$ on voitava esittää niiden lineaarikombinaationa. Matriisin A on täten oltava kääntyvä, sillä $V = A^{-1}W$. Lisäksi matriisin A^{-1} alkioiden tulee olla kokonaislukuja. Determinantin määritelmästä seuraa, että tällöin $\det(A)$ ja $\det(A^{-1})$ ovat kokonaislukuja. Näin ollen

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

joten $\det(A) = \pm 1$. Täten väite on osoitettu vasemmalta oikealle.
$\Longleftarrow$
Matriisiteoriasta tiedetään tulos

$$A^{-1} = \frac{\mathrm{Adj}(A)}{\det(A)},$$

missä $\mathrm{Adj}(A)$ on matriisin A *liittomatriisi*, joka muodostetaan vaihtamalla matriisin A alkiot niiden alideterminanteilla ja vaihtamalla niistä joka toisen merkkiä ja transponoimalla näin saatua matriisia. Matriisin A kaikki alideterminantit ovat kokonaislukuja, joten $\mathrm{Adj}(A)$ on kokonaislukumatriisi ja täten myös $A^{-1} = \pm\mathrm{Adj}(A)$ on kokonaislukumatriisi. Kantavektorit $v_1, v_2 \ldots, v_n$ voidaan esittää vektoreiden $w_1, w_2, \ldots, w_n$ avulla, sillä $V = A^{-1}W$. Näin ollen mielivaltainen hilan L vektori t voidaan esittää vektoreiden $w_1, w_2, \ldots, w_n$ avulla seuraavasti

$$t = t_1v_1 + t_2v_2 + \cdots + t_nv_n = t_1(a_{11}^{-1}w_1 + \cdots + a_{1n}^{-1}w_n) + \cdots + t_n(a_{n1}^{-1}w_1 + \cdots + a_{nn}^{-1}w_n)$$

$$= \sum_{i=1}^{n} (\sum_{j=1}^{n} t_j a_{ji}^{-1}) w_i,$$

missä $t_i, a_{ji}^{-1} \in \mathbb{Z}$. Vektorit $w_1, w_2, \ldots, w_n$ siis virittävät hilan L. Hilan aste on n ja virittäviä vektoreita on n kappaletta, joten $w_1, w_2, \ldots, w_n$ on hilan L kanta. □

Määritelmä 1.16. Olkoon hilan L kanta $v_1, v_2, \ldots, v_n$. Joukko

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

on hilan L *perusalue*.

Lause 1.17. *Olkoon hilan $L \subset \mathbb{R}^n$ aste n ja olkoon $\mathcal{F}$ hilan L perusalue. Tällöin jokainen $w \in \mathbb{R}^n$ voidaan esittää muodossa*

$$w = t + v$$

yksikäsitteisillä $t \in \mathcal{F}$ ja $v \in L$.

Todistus. Olkoon $v_1, v_2, \ldots, v_n$ hilan L kanta, joka antaa perusalueen $\mathcal{F}$. Tällöin vektorit $v_1, v_2, \ldots, v_n$ ovat lineaarisesti riippumattomia avaruudessa $\mathbb{R}^n$, joten ne ovat myös avaruuden kanta. Tällöin mielivaltainen $w \in \mathbb{R}^n$ voidaan kirjoittaa muodossa

$$w = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n,$$

missä $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. Nyt jokainen α_i voidaan kirjoittaa muodossa

$$\alpha_i = t_i + a_i,$$

missä $0 \leq t_i < 1$ ja $a_i \in \mathbb{Z}$. Näin ollen

$$w = \overbrace{t_1 v_1 + t_2 v_2 + \cdots + t_n v_n}^{\in \mathcal{F}} + \overbrace{a_1 v_1 + a_2 v_2 + \cdots + a_n v_n}^{\in L}.$$

Vektori w voidaan siis esittää halutussa muodossa. Osoitetaan vielä, että esitys on yksikäsitteinen. Oletetaan, että $w = t + v = t^* + v^*$, missä $t, t^* \in \mathcal{F}$ ja $v, v^* \in L$. Tällöin

$$(t_1+a_1)v_1+(t_2+a_2)v_2+\cdots+(t_n+a_n)v_n = (t_1^*+a_1^*)v_1+(t_2^*+a_2^*)v_2+\cdots+(t_n^*+a_n^*)v_n.$$

Tästä seuraa, että

$$[(t_1 + a_1) - (t_1^* + a_1^*)]v_1 + \cdots + [(t_n + a_n) - (t_n^* + a_n^*)]v_n = 0.$$

Koska vektorit $v_1, \ldots, v_n$ ovat lineaarisesti riippumattomia, niin on oltava

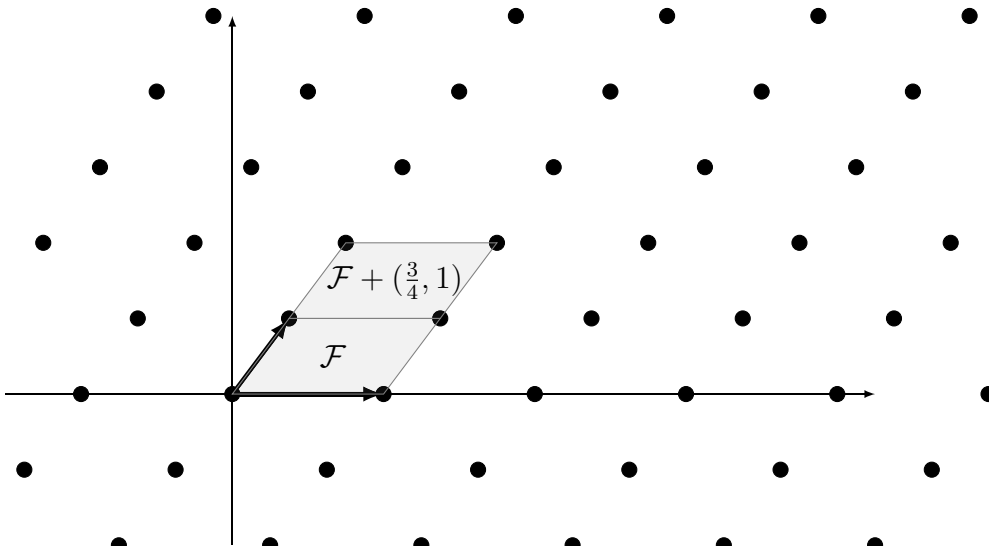
$$t_i + a_i = t_i^* + a_i^* \forall i = 1, 2, \ldots, n.$$

Täten

$$t_i - t_i^* = a_i^* - a_i.$$

On siis oltava niin, että luku $t_i - t_i^*$ kuuluu kokonaislukuihin. Mutta koska $0 \leq t_i, t_i^* < 1$, niin tästä seuraa, että $t_i = t_i^*$, mistä seuraa, että $a_i^* = a_i$. Täten yksikäsitteisyys on osoitettu. □

Perusaluetta voidaan siis "siirtää"kantavektoreiden avulla ja nämä translaatiot täyttävät koko avaruuden $\mathbb{R}^n$. Tätä on havainnollistettu Kuvassa 1.



Kuva 1: Vektoreiden $(\frac{3}{4}, 1)$ ja $(2, 0)$ virittämän hilan pisteet, perusalue $\mathcal{F}$ ja esimerkki perusalueen translaatiosta.

Määritelmä 1.18. Olkoon L n-ulotteinen hila ja $\mathcal{F}$ sen perusalue. Tällöin perusalueen $\mathcal{F}$ tilavuutta kutsutaan *hilan L determinantiksi*. Tälle käytetään merkintää $\det(L)$.

Lause 1.19. *Olkoon L hila, $v_1, v_2, \ldots, v_n$ sen eräs kanta ja $\mathcal{F}$ sen perusalue. Olkoon*

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

vektorin v_i koordinaattiesitys. Muodostetaan kantavektoreista matriisi F seuraavasti:

$$F = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}.$$

Tällöin perusalueen tilavuudelle pätee

$$\mathrm{Vol}(\mathcal{F}) = |\det(F)|.$$

Todistus. Todistuksessa käytetään moniulotteista analyysia, jonka perusteisiin ei ole mahdollisuutta syventyä tässä tutkielmassa. Perusalueen $\mathcal{F}$ tilavuus voidaan laskea integroimalla vakiofunktiota 1 perusalueen yli. Tilavuudelle pätee

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

Perusalue

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

joten muuttuja $x = (x_1, x_2, \ldots, x_n)$ voidaan vaihtaa muuttujaan $t = (t_1, t_2, \ldots, t_n)$ kaavaa

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

noudattaen. Muuttujanvaihto voidaan kirjoittaa yhtälönä $x = tF$. Vaihdon Jacobin matriisi on F ja perusalue $\mathcal{F}$ on n-ulotteisen yksikkökuution $\mathcal{C}_n$ kuva matriisilla F kerrottaessa. Näin ollen

$$\int_{\mathcal{F}} dx_1\, dx_2 \cdots dx_n = \int_{F\mathcal{C}_n} dx_1\, dx_2 \cdots dx_n = \int_{\mathcal{C}_n} dt_1\, dt_2 \ldots dt_n\, |\det(F)| = \mathrm{Vol}(\mathcal{C}_n)|\det(F)|$$

$$= |\det(F)|.$$

□

Huomautus 1.20. Nähdään, että $\det(L)$ ei riipu siitä, mitä hilan kantaa käytetään, sillä

$$\det(L) = |\det(F)|.$$

Lauseesta 1.15 seuraa, että kahden saman hilan kannan $v_1, v_2, \ldots, v_n$ ja $w_1, w_2, \ldots, w_n$ välillä on yhteys

$$AV = W,$$

missä V ja W ovat matriiseja, joiden riveinä ovat vastaavat kantavektorit ja $\det(A) = \pm 1$. Näin ollen

$$|\det(W)| = |\det(AV)| = |\det(A)\det(V)|$$

$$= |\pm \det(V)| = |\det(V)| = |\det(F)| = \det(L).$$

Lause 1.21 (Hadamardin epäyhtälö)**.** *Olkoon L hila, $v_1, v_2, \ldots, v_n$ sen eräs kanta, $\mathcal{F}$ sen perusalue ja F matriisi, jonka riveinä ovat kantavektorit. Tällöin*

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

Todistus. Muodostetaan matriisi M jakamalla jokainen matriisin F rivi i sen pituudella $\|v_i\|$. Matriisin M rivien pituus on siis 1, jolloin tiedetään, että $|\det(M)| \leq 1$, sillä determinantin itseisarvo on matriisin M rivivektoreiden virittämän suuntaissärmiön tilavuus. Yhtäsuuruus pätee jos ja vain jos rivit ovat ortogonaaliset. Tästä saadaan matriisin laskusääntöjä käyttämällä

$$|\det(F)| = \|v_1\|\|v_2\| \cdots \|v_n\||\det(M)| \leq \|v_1\|\|v_2\| \cdots \|v_n\|.$$

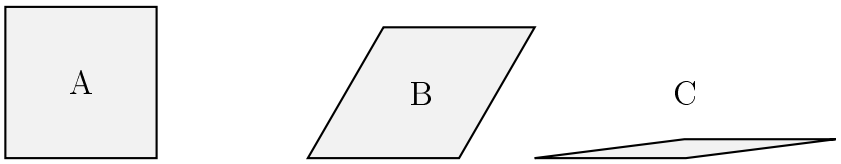
Epäyhtälön yhtäsuuruus pätee vain, jos kantavektorit ovat ortogonaaliset. □

Hadamardin epäyhtälön tulos on hyvin luonnollinen geometrisessa mielessä, jota havainnollistetaan seuraavassa esimerkissä. Epäyhtälöä voidaan käyttää apuna kannan ortogonaalisuuden mittaamiseen. *Hadamardin suhde kannalle* $\mathcal{B} = v_1, \ldots, v_n$ on suure

$$\mathcal{H}(\mathcal{B}) = \Big(\frac{\det(L)}{\|v_1\|\|v_2\| \cdots \|v_n\|}\Big)^{\frac{1}{n}}.$$

Osamäärä on Hadamardin lauseen perusteella selvästi välillä $]0, 1]$ ja lisäksi sitä lähempänä lukua 1, mitä ortogonaalisempi kanta on.

Esimerkki 1.22. Perusaluetta voidaan ajatella n-ulotteisena suuntaissärmiönä (tähän viittaa myös yksi sen englanninkielisestä nimistä *fundamental parallelpiped for* L), jonka sivujen pituudet ovat kantavektoreiden pituudet. Näin ajatellen on helppo huomata, että suuntaissärmiön tilavuus on sitä suurempi, mitä kohtisuoremmat kantavektorit ovat.



Kaikkien kantavektoreiden pituus on kaksi yksikköä.

Perusalue	Kantavektorit	Kantavektoreiden kulma	Tilavuus	Hadamardin suhde
A	$(2,0);(0,2)$	$90°$	4	1
B	$(2,0);(1,\sqrt{3})$	$60°$	$2\sqrt{3}$	$\sqrt{\frac{\sqrt{3}}{2}} \approx 0,93$
C	$(2,0);(\frac{3\sqrt{7}}{4},\frac{1}{4})$	$7.18°$	$\frac{1}{2}$	$\frac{1}{2\sqrt{2}} \approx 0,35$

2 Hilaongelmat

Kuten johdannossa mainittiin, niin hiloihin perustuvien salausmenetelmien turvallisuus perustuu lyhyimmän vektorin (englanniksi *shortest vector problem*, SVP) tai lähimmän vektorin (*closest vector problem*, CVP) ongelmaan. Tässä luvussa tutustutaan näihin ongelmiin.

2.1 Merkle-Hellman salausmenetelmä

Ralph Merkle ja Martin Hellman kehittivät selkäreppuongelmaan perustuvan julkisen avaimen salausmenetelmän vuonna 1978. Sen turvallisuus ei perustu SVP tai CVP ongelmiin, mutta sen murtamiseen voidaan käyttää kyseisiin ongelmiin perustuvia ideoita, joten salausmenetelmä esitellään motivaationa ongelmiin perehtymiseen.
Julkisen avaimen salauksessa käyttäjällä on kaksi avainta; yksityinen ja julkinen. Julkinen avain on kaikkien tiedossa ja sitä käytetään käyttäjälle lähetettävien viestien salaamiseen. Idea on siinä, että salauksen voi purkaa vain vastaanottajan yksityisellä avaimella, joka on vain vastaanottajan tiedossa. Julkisen ja yksityisen avaimen välillä on jokin matemaattinen yhteys, joten salausmenetelmän turvallisuus perustuu siihen, kuinka laskennallisesti vaikeaa yksityisen avaimen päätteleminen on julkisesta avaimesta.

Määritelmä 2.1. Olkoon $M = \{M_1, M_2, \ldots, M_n\}$ joukko positiivisia kokonaislukuja ja S jokin positiivinen kokonaisluku. *Selkäreppuongelmaksi* kutsutaan tehtävää, jossa on löydettävä sellainen joukon M osajoukko, jonka summa on S.

Jotta ongelman käsittely olisi mielekästä, niin jatkossa oletetaan, että on olemassa vähintään yksi ratkaisu S.

Esimerkki 2.2. Olkoon $M = \{2, 5, 8, 15, 22\}$. Jos $S = 7$, niin selvästi ainoa ratkaisu on joukko $\{2, 5\}$. Jos $S = 22$, niin saadaan kaksi ratkaisua, jotka ovat $\{2, 5, 15\}$ ja $\{22\}$.

Ratkaisu voidaan esittää muodossa

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

missä vektorin $x = (x_1, x_2, \ldots, x_n)$ jokainen koordinaatti on joko 1 tai 0, eli koordinaatti kertoo tuleeko vastaava joukon alkio summaan vai ei. Tällöin tehtävänä on siis löytää jokin vektori x, joka tuottaa halutun tuloksen.
Luonnollisesti tehtävä voidaan ratkaista laskemalla kaikkien osajoukkojen

summat. Niiden lukumäärä kuitenkin on 2^n, joten eksponentin n kasvaessa laskenta hidastuu todella nopeasti. Seuraavasta lauseesta nähdään, että eksponentti voidaan puolittaa.

Lause 2.3. *Olkoon $M = \{M_1, M_2, \ldots, M_n\}$ ja (M, S) selkäreppuongelma. Kaikille kokonaislukujoukoille I ja J, joille pätee*

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

lasketaan

$$A_I = \sum_{i \in I} M_i \text{ ja } B_J = S - \sum_{j \in J} M_j$$

ja listataan nämä joukot ja arvot. Tällöin listat sisältävät joukot I_0 ja J_0, joille pätee $A_{I_0} = B_{J_0}$. Tällöin

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

ja selkäreppuongelma on ratkaistu. Listojen alkioiden määrä on enimmillään $2^{n/2}$, joten algoritmin aikakompleksisuusluokka on $\mathcal{O}(2^{n/2+\epsilon})$, missä ϵ on pieni luku, joka ottaa huomioon listojen vertailun.

Todistus. Jos x on vektori, joka ratkaisee ongelman, niin ratkaisu voidaan kirjoittaa muodossa

$$\sum_{1 \leq i \leq \frac{1}{2}n} x_i M_i = S - \sum_{\frac{1}{2}n < i \leq 1} x_i M_i.$$

Tunnetusti joukkojen I ja J osajoukkojen lukumäärä on yhteensä $2 \cdot 2^{n/2}$, sillä joukkojen I ja J alkioiden lukumäärät ovat $n/2$. □

Kryptografiassa on kaksi perusperiaatetta. Salaamisen pitää olla nopeaa lähettäjälle, mutta sen purkaminen pitää olla vaikeaa ilman jotain lisätietoa, jota kutsutaan *salaoveksi*. Tämän lisäksi salatun tekstin purkamisen tuloksen pitää olla yksikäsitteinen. Jotta selkäreppuongelman pohjalta voidaan rakentaa salausmenetelmä, niin on ratkaistava kaksi haastetta; luvun n kasvaessa ongelman ratkaiseminen tulee hyvin hitaaksi ja ratkaisu ei ole yksikäsitteinen. Seuraavaksi käymme näiden haasteiden kimppuun

Määritelmä 2.4. Kokonaislukujono $r = (r_1, r_2, \ldots, r_n)$ on *superkasvava*, jos

$$r_{i+1} \geq 2r_i \text{ kaikilla } 1 \leq i \leq n-1.$$

Lause 2.5. *Olkoon $r = (r_1, r_2, \ldots, r_n)$ superkasvava jono. Tällöin*

$$r_k > r_{k-1} + \cdots + r_2 + r_1 \text{ kaikilla } 2 \leq k \leq n.$$

Todistus. Todistetaan induktiolla. Jos $k = 2$, niin määritelmän perusteella $r_2 \geq 2r_1 > r_1$, joten väite pätee. Seuraavaksi tehdään induktio-oletus, että väite pätee jollain kokonaisluvulla k. Tällöin induktio-oletuksen nojalla

$$r_{k+1} \geq 2r_k = r_k + r_k > r_k + (r_{k-1} + \cdots + r_2 + r_1).$$

□

Jos joukon M alkiot muodostavat superkasvavan jonon, niin selkäreppuongelma ratkeaa helposti seuraavan lauseen algoritmilla. Lisäksi ratkaisu on yksikäsitteinen

Lause 2.6. *Olkoon (M, S) selkäreppuongelma, jossa joukon M alkiot muodostavat superkasvavan jonon. Jos ratkaisu x on olemassa, niin se on yksikäsitteinen ja saadaan seuraavalla algoritmilla:*

Asetetaan $i = n$ ja $S_n = S$.
while $i \geq 1$ **do**
  if $S_n \geq M_i$ **then**
    Asetetaan $x_i = 1$ ja $S_{i-1} = S_i - M_i$.
  else
    Asetetaan $x_i = 0$ ja $S_{i-1} = S_i$.
  end if
  $i = i - 1$
end while
return $x = (x_1, x_2, \ldots, x_n)$.

Todistus. Koska oletetaan, että ratkaisu on olemassa niin selvyyden vuoksi kutsutaan ratkaisua vektoriksi y ja algoritmin tulosta vektoriksi x. Näin ollen $\sum_{i=1}^{n} y_i M_i = S$ ja on osoitettava, että $y = x$.
Todistetaan induktiolla. Oletetaan, että $x_i = y_i$ kaikilla $k < i \leq n$. On osoitettava, että tästä seuraa $x_k = y_k$. Jos $k = n$, niin väite pätee selvästi, joten induktion perusaskel on suoritettu. Hypoteesi tarkoittaa, että algoritmia on suoritettu "alaspäin"luvusta $i = n$ lukuun $i = k + 1$ ja jokaisessa vaiheessa $x_i = y_i$. Ennen kuin algoritmi suoritetaan vaiheessa $i = k$, niin

$$S_k = S - \sum_{i=k+1}^{n} x_i M_i = \sum_{i=1}^{n} y_i M_i - \sum_{i=k+1}^{n} x_i M_i = \sum_{i=1}^{k} y_i M_i.$$

Suoritettaessa algoritmia vaiheessa $i = k$ on kaksi vaihtoehtoa:

1. $y_k = 1$, jolloin $S_k \geq M_k$, sillä tällöin M_k on mukana summassa, joka muodostaa luvun S_k. Tällöin pätee $x_k = 1$.

2. $y_k = 0$ jolloin$M_k > M_1 + M_2 + \cdots + M_{k-1} \geq S_{k-1} = S_k$. Tällöin pätee $x_k = 0$.

Molemmissa tilanteissa $y_k = x_k$, joten ollaan todistettu väite $x = y$. Myös yksikäsitteisyys on todistettu, sillä osoitettiin, että mielivaltainen ratkaisu on yhtenevä deterministisen algoritmin antaman ratkaisun kanssa, joka luonnollisesti on aina sama. □

Algoritmi suorittaa luupin n kertaa, ja jokaisella kierroksella suoritetaan vain muutamia peruslaskutoimituksia. Näin ollen algoritmi kuuluu aikakompleksisuusluokkaan $\mathcal{O}(n)$.

Esimerkki 2.7. Olkoon $M = \{2, 6, 20, 57, 201\}$, joka on selvästi superkasvava. Olkoon $S = 260$. Ratkaistaan selkäreppuongelma edellä esitetyllä algoritmilla.

1. $i = 5, S_5 = 260$
   $S_5 \geq M_5 = 201 \implies x_5 = 1$ ja $S_4 = 260 - 201 = 59$.

2. $i = 4, S_4 = 59$
   $S_4 \geq M_4 = 57 \implies x_4 = 1$ ja $S_3 = 59 - 57 = 2$.

3. $i = 3, S_3 = 2$
   $S_3 < M_3 = 20 \implies x_3 = 0$ ja $S_2 = 2$.

4. $i = 2, S_2 = 2$
   $S_2 < M_2 = 6 \implies x_2 = 0$ ja $S_1 = 2$.

5. $i = 1, S_1 = 2$
   $S_1 \geq M_1 = 2 \implies x_1 = 1$.

6. $i = 0$, joten luupin ehto ei enää toteudu. Palautetaan vektori $x = (1, 0, 0, 1, 1)$.

Koossa on nyt työkalut selkäreppuongelmaan perustuvan salausmenetelmän luomiseen.

Alice[1] luo superkasvavan jonon $r = (r_1, \ldots, r_n)$. Hän myös valitsee kaksi

[1]Alice ja Bob ovat salausmenetelmien analyysissa käytettäviä hahmoja, joista toinen on viestin lähettäjä ja salaaja, toinen vastaanottaja ja salauksen purkaja. Eve on salakuuntelija, joka yrittää murtaa salausta.

salaista kokonaislukua A ja B, joille pätee

$$B > 2r_n \text{ ja } \text{syt}(A, B) = 1.$$

Alice luo uuden jonon M seuraavasti:

$$M_i \equiv Ar_i \pmod{B} \text{ missä } 0 \leq M_i < B.$$

Muunnoksen tarkoituksena on naamioida alkuperäinen jono. Ilman muunnosta salauksen purkaminen olisi triviaalia Lauseen 2.6 algoritmilla. Jono M on Alicen julkinen avain ja kolmikko (A, B, r) yksityinen avain.
Lähettääkseen Alicelle viestin x (binäärivektorin) Bob salaa viestin seuraavasti:

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

S on siis Bobin Alicelle lähettämä salattu viesti.
Alice purkaa salauksen laskemalla ensin luvun

$$S^* \equiv A^{-1}S \pmod{B} \text{ missä } 0 \leq S^* < B.$$

Tässä A^{-1} on luvun A käänteisalkio modulo B. Seuraavaksi Alice ratkaisee selkäreppuongelman (r, S^*) käyttämällä Lauseen 2.6 algoritmia. Salauksen purkaminen onnistuu, sillä

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

Koska $B > 2r_n$, niin

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

joten valitsemalla luvun S^* väliltä $[0, B-1]$ varmistutaan siitä, että pätee $S^* = \sum_{i=1}^{n} x_i r_i$, eikä pelkästään kongruenssi modulo B. Ehdon $\text{syt}(A, B) = 1$ on oltava voimassa, sillä muuten käänteisalkiota A^{-1} ei ole olemassa.

Esimerkki 2.8. Olkoon $r = (2, 6, 20, 57, 201)$ Alicen superkasvava jono, ja olkoon $A = 77$ ja $B = 410$. Täten

$$M = (2 \cdot 77, 6 \cdot 77, 20 \cdot 77, 57 \cdot 77, 201 \cdot 77) \pmod{B} = (154, 52, 310, 289, 307).$$

Bob haluaa lähettää Alicelle viestin $x = (1, 0, 0, 1, 1)$. Hän laskee salatun viestin

$$S = \sum_{i=1}^{n} x_i M_i = 154 + 289 + 307 = 750.$$

Alice laskee käänteisalkion $A^{-1} = 213$ ja laskee luvun

$$S^* = 213 \cdot 750 \equiv 260 \pmod{410}.$$

Sitten Alice ratkaisee selkäreppuongelman (r, S^*). Tulos on $(1, 0, 0, 1, 1)$, kuten nähtiin Esimerkissä 2.7. Salaus on siis purettu onnistuneesti.

Esimerkissä käytetyt arvot ovat luonnollisesti käytännöllisyyden vuoksi pieniä. Käytännössä on oltava $r_1 > 2^n$, sillä salausmenetelmää vastaan on olemassa helppoja hyökkäyksiä, jos $r_1 < 2^n$. Seuraava hyökkäyksen kuvaus seuraa Galbraithin esitystä [3].
Olkoon r_1 huomattavasti pienempi kuin 2^n. Tällöin on luultavaa, että myös r_2 on samaa kokoluokkaa. Lisäksi tiedetään, että

$$Ar_1r_2 \equiv Ar_2r_1 \pmod{B} \implies M_1r_2 \equiv M_2r_1 \pmod{B}.$$

Näin ollen $M_1r_2 - M_2r_1$ on jokin luvun B (pieni) monikerta. Nyt kokeilemalla saamme ehdokkaita luvun B arvoksi. Ehdokkaiden määrää rajaa entisestään se, että

$$\max(M_i) < B \text{ ja hyvin todennäköisesti } B < 2\max(M_i).$$

Jokaiselle ehdokkaalle B^* voidaan laskea

$$A^* \equiv M_1 r_1^{-1} \pmod{B^*}$$

ja sitten testata, mikäli jono $A^{*^{-1}}M_i$ näyttää superkasvavalta.

Luvun r_1 on siis oltava suurempi kuin 2^n. Superkasvavuudesta seuraa, että

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

Näin ollen $B > 2r_n > 2^{2n+1}$, joten

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

Miten salakuuntelija Eve voisi hyökätä menetelmää vastaan yleisemmässä tapauksessa? Oletetaan, että Evellä on tiedossa Alicen julkinen avain M ja Bobin lähettämä salattu viesti S. Aluksi hän muodostaa matriisin

$$\begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & M_1 \\ 0 & 2 & 0 & \cdots & 0 & M_2 \\ 0 & 0 & 2 & \cdots & 0 & M_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & M_n \\ 1 & 1 & 1 & \cdots & 1 & S \end{pmatrix}.$$

Merkitään matriisin rivivektoreita v_i, missä $i = 1, 2, \ldots, n+1$, ja tutkitaan hilaa

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

Olkoon binäärivektori $x = (x_1, \ldots, x_n)$ Bobin selkokielinen viesti. Tällöin hilaan L kuuluu vektori

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

Viimeinen koordinaatti on 0, sillä $S = \sum_{i=1}^{n} x_i M_i$. Koska kaikki koordinaatit x_i ovat joko 0 tai 1, niin vektorin t koordinaatit $t_i = 2x_i - 1 = \pm 1, i = 1, \ldots, n$. Vektori t on siis lyhyt, sillä $\|t\| = \sqrt{n}$. Hilan virittävät vektorit ovat pituudeltaan $\|v_i\| = \mathcal{O}(2^{2n})$, sillä $m_i = \mathcal{O}(2^{2n})$ ja $S = \mathcal{O}(2^{2n})$, joten on epätodennäköistä, että hilassa on muita yhtä lyhyitä nollasta eroavia vektoreita kuin t. Even tehtävänä on siis löytää hilan L lyhyin vektori eli ratkaista lyhyimmän vektorin ongelma.

Huomautus 2.9. Tehokkaan hyökkäyksen Merkle-Hellman salausta vastaan esitteli Shamir vuonna 1982 julkaisussaan *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem* [14]. Hänen menetelmänsä ei kuitenkaan perustunut hilojen teoriaan. Vuonna 1982 julkaistiin myöhemmin tarkemmin käsiteltävä LLL-algoritmi [11], jonka avulla voidaan löytää lyhyitä vektoreita hilasta. Algoritmi asetti monet selkäreppuongelmaan perustuvat salausmenetelmät vaaraan. Tästä syystä niitä ei enää juurikaan käytetä.

2.2 Lyhyimmän ja lähimmän vektorin ongelmat

Määritelmä 2.10. Hilan L nollasta eroava vektori v, joka on lyhin euklidisen normin $\|v\|$ mielessä on *lyhyimmän vektorin ongelman* (SVP) ratkaisu.

Määritelmä 2.11. Olkoon $w \in \mathbb{R}^m \setminus L$. Hilan L vektori v, joka minimoi euklidisen normin $\|w - v\|$ on *lähimmän vektorin ongelman* (CVP) ratkaisu vektorin w suhteen.

Kummankaan ongelman ratkaisut eivät ole selvästikään yksikäsitteisiä, ja molemmat ongelmat ovat laskennallisesti vaikeita; CVP on NP-täydellinen[2], kuten myös SVP tietyillä ehdoilla.

Ongelmista on olemassa tärkeitä muunnelmia, jotka esitellään seuraavaksi.

[2]NP-täydellisyys tarkoittaa, että ongelmaa ei voida ratkaista deterministisellä Turingin koneella polynomiajassa. Katso lisää esimerkiksi Goldreichin teoksesta *Foundations of Cryptography* [4].

Määritelmä 2.12. Olkoon L n-asteinen hila ja $\psi(n)$ kuvaus luonnollisilta luvuilta reaaliluvuille. Vektori v, jonka pituus on enintään $\psi(n)$-kertaa hilan lyhyimmän nollasta eroavan vektorin pituus, on *likimääräisen lyhyimmän vektorin ongelman* ratkaisu (englanniksi *approximate shortest vector problem, apprSVP*). Tehtävänä on siis löytää vektori, joka toteuttaa ehdon

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

Ratkaisu riippuu luonnollisesti funktion $\psi(n)$ valinnasta. Myös lähimmän vektorin ongelmasta on olemassa likimääräinen versio apprCVP, jonka määritelmä on vastaava.

Seuraavaksi tutustutaan lauseisiin, jotka antavat ylärajan hilan lyhyimmän nollasta eroavan vektorin pituudelle, sekä valaisevat mistä tekijöistä tämä yläraja riippuu.

Määritelmä 2.13. Olkoon $a \in \mathbb{R}^n$ ja $R > 0$. Joukko

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

on *R-säteinen a-keskinen pallo.*

Määritelmä 2.14. Olkoon $S \subset \mathbb{R}^n$.

1. S on *rajoitettu*, jos on olemassa $M \in \mathbb{Z}$ siten, että

   $$\max_{v \in S} \|v\| < M.$$

   Tällöin selvästi $S \subset \mathbf{B}_M(0)$.

2. S on *symmetrinen*, jos kaikille $a \in S$ pätee $-a \in S$.

3. S on *konveksi*, jos kaikilla $a, b \in S$ myös jana pisteestä a pisteeseen b kuuluu joukkoon S.

4. S on *suljettu*, jos siitä, että kaikilla R

   $$\mathbf{B}_R(a) \cap S \neq \emptyset$$

   seuraa, että $a \in S$.

Lause 2.15 (Minkowskin lause)**.** *Olkoon $L \subset \mathbb{R}^n$ n-ulotteinen hila ja $S \subset \mathbb{R}$ rajoitettu, symmetrinen ja konveksi joukko, jolle pätee*

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

Tällöin S sisältää nollasta eroavan vektorin, joka kuuluu hilaan L. Jos S on suljettu, niin ehdoksi riittää $\mathrm{Vol}(S) \geq 2^n \det(L)$.

Todistus. Olkoon $\mathcal{F}$ hilan L perusalue. Lauseesta 1.17 seuraa, että jokainen vektori $a \in \mathbb{R}^n$ voidaan kirjoittaa yksikäsitteisesti muodossa

$$a = v_a + w_a, \text{ missä } v_a \in L \text{ ja } w_a \in \mathcal{F}.$$

Muodostetaan joukko

$$\frac{1}{2}S = \{\frac{1}{2}a \ : a \in S\}.$$

Koska jokaista joukon alkiota pienennetään kertoimella $\frac{1}{2}$, niin joukon tilavuus pienenee kertoimella $\frac{1}{2^n}$. Näin ollen pätee

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

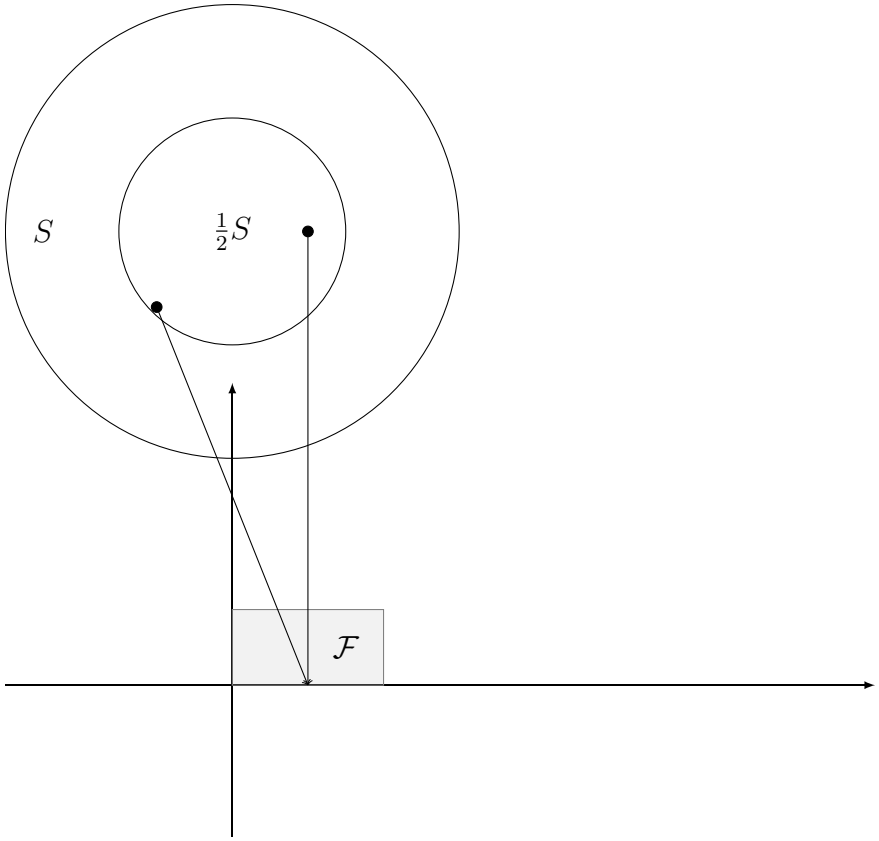
Määritellään kuvaus

$$f : \ \frac{1}{2}S \to \mathcal{F}, \ \frac{1}{2}a \to w_{\frac{1}{2}a}.$$

Osoitetaan, että f ei ole injektio. Tehdään vastaoletus f on injektio. Injektiivisyydestä ja siitä, että f on kokoelma yksittäisten pisteiden translaatioita ja täten lokaalisti tilavuuden säilyttävä, seuraa, että kuvaus f säilyttää tilavuuden. Näin ollen

$$\text{Vol}\Big(\frac{1}{2}S\Big) = \text{Vol}(\mathcal{F}).$$

Tämä on ristiriita ja f ei ole injektio. Esimerkki Kuvassa 2.



Kuva 2: Hilan kantavektorit ovat (0,2) ja (1,0). $S = \mathbf{B}_3(0, 6)$. Kuvaus f vie pisteet $(1, 6)$ ja $(-1, 5)$ samaan pisteeseen $(1, 0)$.

Kahdella joukon $\frac{1}{2}S$ erillisellä vektorilla $\frac{1}{2}a_1$ ja $\frac{1}{2}a_2$ on siis sama kuva. Vektoreille pätee

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

Vähentämällä vektorit saadaan nollasta eroava vektori:

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

Toisaalta tämä vektori on pisteestä a_1 pisteeseen $-a_2$ kulkevan janan keskipiste. Symmetrisyydestä seuraa, että $-a_2 \in S$, joten konveksisuudesta seuraa, että vektori $v_1 - v_2$ kuuluu joukkoon S. Olemme siis löytäneet nollasta eroavan hilaan kuuluvan vektorin, joka sisältyy joukkoon S. Osoitetaan vielä, että joukon S ollessa suljettu ehto $\mathrm{Vol}(S) \geq 2^n \det(L)$ on riittävä. Olkoon $k \in \mathbb{Z}_+$. Jokaisella k kasvatetaan joukkoa S kertoimella $1 + \frac{1}{k}$. Juuri todistetun Minkowskin lauseen version perusteella on olemassa nollasta eroava

vektori

$$v_k \in \Big(1 + \frac{1}{k}\Big) S \cap L.$$

Jokainen näistä vektoreista $v_1, v_k \ldots$ kuuluu rajoitettuun joukkoon $2S$, joten toisistaan eroavia vektoreita on jonossa vain äärellinen määrä. On siis olemassa jokin vektori v, joka esiintyy jonossa äärettömän monesti, joten tämä nollasta eroava hilavektori kuuluu leikkaukseen

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

Koska S on suljettu, niin tämä leikkaus on yhtäsuuri joukon S kanssa. □

Minkowskin lauseen avulla on helppoa todistaa Hermiten lause, joka antaa ylärajan hilan lyhyimmälle nollasta eroavalle vektorille.

Lause 2.16 (Hermiten lause)**.** *Jokainen n-ulotteinen hila L sisältää nollasta eroavan vektorin v, jolle pätee*

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

Todistus. Käytetään Minkowskin lausetta. Olkoon S n-ulotteinen hyperkuutio, jolle pätee

$$S = \{(x_1 \ldots, x_n) \ : -B \leq x_1 \leq B \text{ kaikilla } i = 1, \ldots, n\}.$$

Joukko S on symmetrinen, suljettu, rajoitettu ja

$$\mathrm{Vol}(S) = (2B)^n.$$

Asetetaan $B = \det(L)^{1/n}$, jolloin $\mathrm{Vol}(S) = 2^n \det(L)$. Minkowskin lauseesta seuraa, että on olemassa nollasta eroava hilavektori $a \in S \cap L$. Joukon S määritelmästä seuraa

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

□

Hermiten lauseen tulosta voidaan parantaa käyttämällä hyperkuution sijaan hyperpalloa.

Lause 2.17. *Olkoon $\boldsymbol{B}_R(a)$ hyperpallo avaruudessa R^n. Kun n on suuri, niin pallon tilavuudelle pätee*

$$\mathit{Vol}(\boldsymbol{B}_R(a))^{1/n} \approx \sqrt{\frac{2\pi e}{n}} R.$$

Todistus. Todistus ohitetaan. Katso esimerkiksi [7, s. 401]. □

Pallo $\mathbf{B}_R(0)$ on selvästi rajoitettu, suljettu, konveksi ja symmetrinen, joten Minkowskin lauseen perusteella pätee, että jos R valitaan siten, että

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

niin pallo $\mathbf{B}_R(0)$ sisältää nollasta eroavan hilavektorin. Oletetaan, että n on suuri. Tällöin R tulee valita niin, että

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

Koska nollasta eroava hilavektori v sisältyy palloon, niin pätee

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

Arviota ollaan nyt parannettu. Jatketaan kuitenkin pohdintaa. Palloon sisältyvien hilavektoreiden lukumäärää voidaan arvioida todennäköisyyslaskennan avulla. Nyt

$$\#\{v \in L : \|v\| \leq R\} \approx \frac{\mathrm{Vol}(\mathbf{B}_R(0))}{\mathrm{Vol}(\mathcal{F})}.$$

Kun valitaan R niin, että palloon kuuluu vain yksi hilavektori, niin tämä hilavektori on luonnollisesti lyhyin. Ratkaistaan R käyttämällä Lauseen 2.17 arviota.

$$\begin{aligned}
\frac{\mathrm{Vol}(\mathbf{B}_R(0))}{\mathrm{Vol}(\mathcal{F})} &\approx 1 \\
(\frac{2\pi e}{n})^{n/2} R^n &\approx \det(L) \\
R &\approx \sqrt{\frac{n}{2\pi e}} (\det(L))^{1/n}.
\end{aligned}$$

Tähän perustuu *Gaussin heuristiikka*. Sen mukaan satunnaisesti valitun hilan lyhyimmälle vektorille $v_{\mathbf{lyhyin}}$ pätee

$$\|v_{\mathrm{lyhyin}}\| \approx \sqrt{\frac{n}{2\pi e}} (\det(L))^{1/n}.$$

3 Hilan redusointialgoritmit

Hilan redusointialgoritmilla tarkoitetaan algoritmia, joka ottaa syötteenä hilan mielivaltaisen kannan, ja tulostaa syötettä ortogonaalisemman ja lyhyemmistä vektoreista koostuvan kannan samalle hilalle. Redusoinnin motivaationa on se, että tällaisella kannalla on helpompi ratkaista (vähintäänkin likimääräisiä) hilaongelmia.

3.1 Gaussin redusointialgoritmi

Olkoon hila L astetta kaksi ja sen kantavektorit v_1 ja v_2. Voidaan olettaa, että $\|v_1\| < \|v_2\|$. Korvaamalla vektori v_2 vektorilla v_2^*, jolle pätee

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

saadaan vektorit v_1 ja v_2^*, jotka ovat ortogonaaliset. Vektori v_2^* ei kuitenkaan välttämättä kuulu hilaan L, sillä luku $\frac{v_1 \cdot v_2}{\|v_1\|^2}$ ei välttämättä ole kokonaisluku. Korvataankin v_2 vektorilla v_2', jolle pätee

$$v_2' = v_2 - m v_1, \text{ missä } m = \left\lceil \frac{v_1 \cdot v_2}{\|v_1\|^2} \right\rfloor.$$

Merkinnällä $\lceil a \rfloor$ tarkoitetaan luvun a pyöristämistä lähimpään kokonaislukuun.
Osoitetaan, että kannat v_1, v_2 ja v_1, v_2' virittävät saman hilan. Seuraava päättely on tutkielman tekijän omaa. Olkoon V matriisi, jonka rivit ovat vektorit v_1, v_2 ja V' matriisi, jonka rivit ovat vektorit v_1, v_2'. Tällöin

$$V' = AV, \text{ missä } A = \begin{pmatrix} 1 & 0 \\ -m & 1 \end{pmatrix}.$$

A on selvästi unimodulaarinen, jolloin Lauseen 1.15 perusteella kannat virittävät saman hilan.
Jos $\|v_1\| < \|v_2'\|$, niin prosessi lopetetaan. Muussa tapauksessa samat vaiheet toistetaan, mutta niin, että vektorista v_1 vähennetään vektorin v_2' monikertoja. Seuraavassa algoritmin tarkka kuvaus.

Lause 3.1 (Gaussin redusointialgoritmi)**.** *Olkoon $L \subset \mathbb{R}^2$ toista astetta oleva hila, jonka kantavektorit ovat v_1 ja v_2. Tällöin seuraava algoritmi antaa kannan, jolle pätee, että v_1' on hilan lyhyin nollasta eroava vektori, joten algoritmi ratkaisee lyhyimmän vektorin ongelman. Lisäksi kantavektorien väliselle kulmalle θ pätee* $60° < \theta < 120°$.

while $m \neq 0$ **do**
  if $\|v_1\| > \|v_2\|$ **then**
    $v_1 := v_2$ ja $v_2 := v_1$.
  end if
  $m = \lceil \frac{v_1 \cdot v_2}{\|v_1\|^2} \rfloor$.
  if $m = 0$ **then**
    return Uudet kantavektorit $v_1' = v_1$ ja $v_2' = v_2$.
  else
    $v_2 = v_2 - mv_1$.
  end if
end while

Todistus. Todistetaan ensin, että v_1' on hilan lyhyin nollasta eroava vektori. Koska algoritmin suoritus on päättynyt, niin tiedetään, että $\|v_2'\| > \|v_1'\|$. Lisäksi

$$\frac{|v_1' \cdot v_2'|}{\|v_1'\|^2} < \frac{1}{2}, \tag{1}$$

sillä $m = 0$, joten $\frac{v_1' \cdot v_2'}{\|v_1'\|^2}$ on oltava välillä $]-\frac{1}{2}, \frac{1}{2}[$. Olkoon $w \in L$ jokin nollasta eroava hilavektori. Tällöin

$$w = a_1 v_1' + a_2 v_2' \text{ joillain } a_1, a_2 \in \mathbb{Z}.$$

Nyt nähdään, että

$$\begin{aligned}
\|w\|^2 &= \|a_1 v_1' + a_2 v_2'\|^2 = a_1^2\|v_1'\|^2 + 2a_1a_2(v_1' \cdot v_2') + a_2^2\|v_2'\|^2 \\
&\geq a_1^2\|v_1'\|^2 - 2|a_1a_2||v_1' \cdot v_2'| + a_2^2\|v_2'\|^2 \\
&> a_1^2\|v_1'\|^2 - |a_1a_2|\|v_1\|^2 + a_2^2\|v_2'\|^2 \qquad (2) \\
&\geq a_1^2\|v_1'\| - |a_1a_2|\|v_1'\| + a_2^2\|v_1'\|^2 \qquad (3) \\
&= (a_1^2 - |a_1||a_2| + a_2^2)\|v_1'\|^2.
\end{aligned}$$

Epäyhtälö (2) seuraa epäyhtälöstä (1) ja epäyhtälö (3) tiedosta $\|v_2'\| > \|v_1'\|$.

Kaikilla reaaliluvuilla t_1 ja t_2 pätee, että

$$t_1^2 - t_1t_2 + t_2^2 = (t_1 - \frac{1}{2}t_2)^2 + \frac{3}{4}t_2^2 = \frac{3}{4}t_1^2 + (\frac{1}{2}t_1 - t_2)^2 = 0$$

jos ja vain jos $t_1 = t_2 = 0$. Koska vektori w eroaa nollavektorista, niin kokonaisluvut a_1 ja a_2 eivät voi olla yhtä aikaa nolla. Tällöin

$$(a_1^2 - |a_1||a_2| + a_2^2) \geq 1,$$

joten $\|w\|^2 \geq \|v_1'\|^2$, josta seuraa, että $\|w\| \geq \|v_1'\|$. Mielivaltainen vektori w on siis lyhimmillään saman mittainen kantavektorin v_1' kanssa, joten väite on todistettu.
Todistetaan sitten, että $60° < \theta < 120°$. Seuraava päättely eroaa päälähteestä [7], ja on tutkielman tekijän omaa päättelyä. Lauseesta 1.7 seuraa, että

$$\cos(\theta) = \frac{v_1' \cdot v_2'}{\|v_1'\|\|v_2'\|}.$$

Koska $\|v_2'\| \geq \|v_1'\|$, niin

$$|\cos(\theta)| \leq \frac{|v_1' \cdot v_2'|}{\|v_1'\|^2} < \frac{1}{2}.$$

Näin ollen $-\frac{1}{2} < \cos(\theta) < \frac{1}{2}$, joten $60° < \theta < 120°$.

□

Esimerkki 3.2. Käytetään "huonoa"kantaa $v_1 = (\frac{15}{2}, 4)$ ja $v_2 = (17, 8)$. Kanta on hyvin kaukana ortogonaalisesta, sillä vektorien välinen kulma on noin 2.87°. Seuraavassa käydään algoritmin toiminta läpi iteraatio kerrallaan.

1. $v_1 = (\frac{15}{2}, 4)$ ja $v_2 = (17, 8) \implies \|v_1\| = \frac{17}{2}$ ja $\|v_2\| \approx 18,78$.

   $$m = \left\lceil \frac{v_1 \cdot v_2}{\|v_1\|^2} \right\rfloor = \left\lceil \frac{159,5}{72,25} \right\rfloor = 2 \implies v_2 = (17, 8) - 2(\frac{15}{2}, 4) = (2, 0)$$

2. $\|v_1\| = \frac{17}{2} > 2 = \|v_2\| \implies$ Vaihdetaan vektorit keskenään, joten $v_1 = (2, 0)$ ja $v_2 = (\frac{15}{2}, 4)$.
   $m = \lceil \frac{15}{4} \rfloor = 4 \implies v_2 = (\frac{15}{2}, 4) - 4(2, 0) = (-\frac{1}{2}, 4)$

3. $\|v_1\| = 2$ ja $\|v_2\| \approx 4,03$
   $m = \lceil -\frac{1}{4} \rfloor = 0 \implies$ Palautetaan uudet kantavektorit $v_1' = (2, 0)$ ja $v_2' = (-\frac{1}{2}, 4)$.

Algoritmilla saatiin siis vain kolmella iteraatiolla huomattavasti ortogonaalisemmat kantavektorit, joiden välinen kulma on noin 97.13°. Sitäkin tärkeämpää on kuitenkin lyhyimmän vektorin ongelman ratkaisu, joka on $v_1 = (2, 0)$.

Gaussin redusointialgoritmin kompleksisuusluokka on $\mathcal{O}(\log(X)^3)$, missä X on reaaliluku, jolle pätee

$$\|v_i\| \leq X, \ i = 1, 2.$$

Tässä v_1 ja v_2 ovat algoritmiin syötettävät alkuperäiset kantavektorit. Todistus sivuutetaan; katso esimerkiksi [2, s. 350].

3.2 LLL-algoritmi

Kuten aikaisemmin mainittiin, LLL-algoritmin julkaisivat A.K. Lenstra, H.W. Lenstra ja L. Lovász vuonna 1982. Algoritmia voidaan pitää Gaussin redusointialgoritmin yleistyksenä mielivaltaista astetta olevalle hilalle, sillä sen tavoitteena on luoda mahdollisimman lyhyistä ja ortogonaalisista vektoreista koostuva kanta.
Olkoon $v_1, \ldots, v_n$ hilan L kanta. Käytetään Gram-Schmidt algoritmia ja luodaan ortogonaalinen kanta $v_1^*, \ldots, v_n^*$. On tärkeää huomata, että Gram-Schmidt algoritmilla saatu kanta virittää saman vektoriavaruuden kuin $v_1, \ldots, v_n$, muttei samaa hilaa. Tämä johtuu siitä, että

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \text{ missä } \mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2} \text{ kaikilla } 1 \leq j \leq i-1.$$

Kertoimet μ_{ij} eivät selvästikään ole välttämättä kokonaislukuja. Kannalla on kuitenkin seuraava yhteys hilan determinanttiin.

Lause 3.3. *Olkoon $v_1, \ldots, v_n$ hilan L kanta ja $v_1^*, \ldots, v_n^*$ Gram-Schmidt algoritmilla saatu ortogonaalinen kanta vastaavalle vektoriavaruudelle. Tällöin*

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

Todistus. Olkoon F matriisi, jonka riveinä ovat vektorit $v_1, \ldots, v_n$. Lauseesta 1.19 tiedetään, että $\det(L) = |\det(F)|$. Olkoon F^* matriisi, jonka riveinä ovat vektorit $v_1^*, \ldots, v_1^*$ ja

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

Nyt $F = MF^*$ ja lisäksi tiedetään, että $\det M = 1$, sillä M on yksikköalakolmiomatriisi. Näin ollen

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

Viimeinen yhtäsuuruus seuraa siitä, että matriisin F^* rivit ovat ortogonaaliset. □

Määritelmä 3.4. Olkoon $v_1, \ldots, v_n$ hilan L kanta ja $v_1^*, \ldots, v_n^*$ Gram-Schmidt algoritmilla saatu vastaava ortogonaalinen kanta. Kantaa $v_1, \ldots, v_n$ kutsutaan *LLL-redusoiduksi*, jos se toteuttaa seuraavat kaksi ehtoa:

Suuruusehto

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

Lovászin ehto

$$\|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right)\|v_{i-1}^*\|^2 \text{ kaikilla } 1 < i \leq n.$$

Osoitetaan seuraavaksi, että LLL-redusoidulla kannalla on niitä ominaisuuksia, mitä "hyvällä"kannalla halutaan olevan.

Lause 3.5. *Olkoon L n-asteinen hila. Mielivaltaiselle hilan LLL-redusoidulle kannalle pätee seuraavat tulokset:*

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

$$\|v_j\| \leq 2^{(i-1)/2}\|v_i^*\| \textit{ kaikilla } 1 \leq j < i \leq n.$$

Lisäksi vektorille v_1 pätee

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

Vektori v_1 on siis likimääräisesti lyhyimmän vektorin ongelman ratkaisu, kun kuvaukseksi $\psi(n)$ valitaan $2^{(n-1)/2}$.

Todistus. Todistetaan ensin, että

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

Lovászin ehdosta ja siitä, että suuruusehdon perusteella $|\mu_{i,i-1}| \leq \frac{1}{2}$ seuraa, että

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

Käyttämällä epäyhtälöä (4) toistuvasti saadaan tulos

$$\|v_j^*\|^2 \leq 2^{i-j}\|v_i^*\|^2. \tag{5}$$

Gram-Schmidt algoritmin määritelmästä saadaan

$$\begin{aligned}
\|v_i\|^2 &= \|v_i^* + \sum_{j=1}^{i-1} \mu_{i,j} v_j^*\|^2 \\
&= \|v_i^*\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \|v_j^*\|^2, \text{ sillä vektorit } v_1^*, \ldots, v_n^* \text{ ovat ortogonaalisia.} \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \|v_j^*\|^2, \text{ sillä } |\mu_{i,j}| \leq \frac{1}{2}. \\
&\leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} 2^{i-j-2} \|v_i^*\|^2 \text{ (seuraa epäyhtälöstä (5).)} \\
&= \frac{1 + 2^{i-1}}{2} \|v_i^*|^2 \\
&\leq 2^{i-1} \|v_i^*\|^2.
\end{aligned}$$

Nyt siis

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

missä viimeinen yhtäsuuruus seuraa Lauseesta 3.3. Väite saadaan ottamalla neliöjuuret puolittain.
Edeltävän todistuksen perusteella kaikilla $j \leq i$ pätee

$$\|v_j\|^2 \leq 2^{j-1} \|v_j^*\|^2.$$

Epäyhtälöstä (5) seuraa, että

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

Ottamalla neliöjuuret puolittain saadaan väite

$$\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\| \text{ kaikilla } 1 \leq j < i \leq n.$$

Asettamalla tässä tuloksessa $j = 1$ ja käyttämällä Lausetta 3.3 saadaan

$$\|v_1\|^n \leq \prod_{i=1}^{n} 2^{(i-1)/2} \|v_i^*\| = 2^{n(n-1)/4} \prod_{i=1}^{n} \|v_i^*\| = 2^{n(n-1)/4} \det(L).$$

Otetaan n:s juuri puolittain, jolloin saadaan väite

$$\|v_1\| \leq 2^{(n-1)/4} |\det(L)|^{1/n}.$$

Todistetaan lopuksi tulos

$$\|v_1\| \leq 2^{(n-1)/2} \min_{0 \neq v \in L} \|v\|.$$

Olkoon $v \in L$ mielivaltainen nollasta eroava hilavektori. Vektorille v pätee

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

missä $i \leq n$ ja $a_i \neq 0$. Toinen yhtäsuuruus pätee, sillä Lauseen 1.9 perusteella vektorit $v_1, \ldots, v_i$ ja $v_1^*, \ldots, v_i^*$ virittävät saman avaruuden. Luonnollisesti luvut $a_1, \ldots, a_i$ ovat kokonaislukuja ja luvut $b_1, \ldots, b_i$ reaalilukuja. Tiedetään, että

$$v_i^* \cdot v_i^* = v_i^* \cdot v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_i^* \cdot v_j^* = v_i^* \cdot v_i,$$

sillä ortogonaalisuuden perusteella $v_i^* \cdot v_j^* = 0$, kun $j \neq i$. Lisäksi pätee

$$v \cdot v_i^* = \sum_{j=1}^{i} a_j v_j \cdot v_i^* = a_i v_i \cdot v_i^* + \sum_{j=1}^{i-1} a_j v_j \cdot v_i^*.$$

Vektori $\sum_{j=1}^{i-1} a_j v_j$ kuuluu Lauseen 1.9 perusteella kannan $v_1^*, \ldots, v_{i-1}^*$ virittämään avaruuteen, joten

$$v \cdot v_i^* = a_i v_i \cdot v_i^*.$$

Toisaalta pätee

$$v \cdot v_i^* = b_i v_i^* \cdot v_i^* = a_i v_i \cdot v_i^* \text{ ja lisäksi } v_i \cdot v_i^* = v_i^* \cdot v_i^*,$$

joten $a_i = b_i$. Täten $|a_i| = |b_i| \geq 1$. Käytetään tulosta

$$\|v_j\| \leq 2^{(i-1)/2} \|v_i^*\| \text{ kaikilla } 1 \leq j < i \leq n,$$

ja asetetaan $j = 1$. Nyt

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

Ottamalla neliöjuuret puolittain saadaan

$$2^{(n-1)/2} \|v\| \geq \|v_1\|.$$

Tämä pätee mielivaltaiselle nollasta eroavalle hilavektorille v, joten erityisesti se pätee lyhyimmälle nollasta eroavalle hilavektorille ja näin väite on todistettu. □

Seuraava tulos on tutkielman tekijän itse havaitsema ja todistama.

Seuraus 3.6. *LLL-redusoidun kannan Hadamardin suhteelle pätee*

$$\Big(\frac{\det L}{\|v_1\| \cdots \|v_n\|}\Big)^{1/n} \geq \frac{1}{2^{(n-1)/4}}.$$

Todistus. Edellisestä lauseesta tiedetään, että

$$\|v_1\| \cdots \|v_n\| \leq 2^{n(n-1)/4} \det(L).$$

Näin ollen

$$\Big(\frac{\det(L)}{\|v_1\| \cdots \|v_n\|}\Big) \geq \frac{\det(L)}{2^{n(n-1)/4} \det(L)} = \frac{1}{2^{n(n-1)/4}}.$$

Väite seuraa, kun otetaan n:s juuri puolittain. □

Seuraavaksi esitellään LLL-algoritmi ja osoitetaan, että se tulostaa LLL-redusoidun kannan hilalle L, kun syötteenä on jokin hilan L kanta. Käytännöllisyyden vuoksi käsitellään hiloja, joiden kantavektorit ovat kokonaislukuvektoreita.

Lause 3.7 (LLL-algoritmi)**.** *Olkoon $v_1, v_2, \ldots, v_n$ hilan $L \in \mathbb{Z}^n$ kanta. Seuraava algoritmi tulostaa LLL-redusoidun kannan hilalle L.*

1: Annetaan hilan L kanta $v_1, v_2, \ldots, v_n$ syötteenä.
2: Asetetaan $k = 2$.
3: Asetetaan $v_1^* = v_1$.
4: **while** $k \leq n$ **do**
5: $\quad j = k - 1$.
6: $\quad$ **while** $j \geq 1$ **do**
7: $\quad\quad$ Asetetaan $v_k = v_k - \lceil \mu_{k,j} \rfloor v_j$.
8: $\quad\quad$ Asetetaan $j = j - 1$.
9: $\quad$ **end while**
10: $\quad$ **if** $\|v_k^*\|^2 \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right)\|v_{k-1}^*\|^2$ **then**
11: $\quad\quad$ $k = k + 1$
12: $\quad$ **else**
13: $\quad\quad$ Vaihdetaan $v_{k-1} := v_k$ ja $v_k := v_{k-1}$.
14: $\quad\quad$ Asetetaan $k = \max(k - 1, 2)$.
15: $\quad$ **end if**
16: **end while**
17: **return** LLL-redusoitu kanta $v_1, v_2, \ldots, v_n$.

Tässä $\mu_{i,j} = (v_i \cdot v_j^*)/\|v_j^*\|$, missä vektorit $v_1^*, \ldots, v_k^*$ on saatu Gram-Schmidt algoritmilla sen hetkisillä vektoreiden $v_1, \ldots, v_k$ arvoilla.

Todistus. Selvästi algoritmin päättyessä tulostettava kanta on LLL-redusoitu, sillä riveillä $6 - 7$ suoritettava luuppi varmistaa, että suuruusehto toteutuu ja jokaisen vektorin tulee läpäistä Lovászin ehto rivillä 10. Onkin osoitettava, että algoritmin suoritus päättyy. Tämä ei ole itsestään selvää, sillä pääluupin päättyminen riippuu indeksistä k, joka jokaisella iteraatiolla joko kasvaa rivillä 11 tai pienentyy rivillä 14. Osoitetaan, että rivi 14 suoritetaan äärellisen monta kertaa. Tällöin k kasvaa jossain vaiheessa suuremmaksi kuin n, sillä jokaisella iteraatiolla suoritetaan joko rivi 11 tai 14.
Olkoon $v_1, \ldots, v_n$ hilan L kanta ja $v_1^*, \ldots, v_n^*$ Gram-Schmidt algoritmilla saatu kanta. Kaikilla $\ell = 1, 2, \ldots, n$ *alihila* L_ℓ on vektoreiden $v_1, \ldots, v_\ell$ virittämä hila. Määritellään lisäksi suureet

$$d_\ell = \prod_{i=1}^{\ell} \|v_i^*\|^2 \text{ ja } D = \prod_{\ell=1}^{n} d_\ell = \prod_{i=1}^{n} \|v_i^*\|^{2(n+1-i)}.$$

LLL-algoritmin suorituksen aikana luvun D arvo muuttuu vain rivillä 13 suoritettavan vektoreiden vaihdon johdosta. Tällöin d_{k-1} on ainut tekijä, joka muuttuu, sillä jos $\ell < k-1$, niin vaihdettavat vektorit v_{k-1}^* ja v_k^* eivät vaikuta arvoon, ja jos $\ell \geq k$, niin sekä v_{k-1}^* että v_k^* kuuluvat tuloon, jolloin niiden paikan vaihtaminen ei luonnollisestikaan vaikuta tuloon. Arvioidaan tekijän d_{k-1} muutosta. Algoritmissa suoritetaan vektoreiden vaihto vain jos Lovászin

ehto ei ole voimassa, joten

$$\|v_k^*\|^2 < \left(\frac{3}{4} - \mu_{k,k-1}\right)\|v_{k-1}^*\|^2 \leq \frac{3}{4}\|v_{k-1}^*\|^2.$$

Näin ollen

$$d_{k-1}^{\text{uusi}} = \|v_1^*\|^2\|v_2^*\|^2 \cdots \|v_{k-2}^*\|^2\|v_k^*\|^2$$

$$= \|v_1^*\|^2\|v_2^*\|^2 \cdots \|v_{k-2}^*\|^2\|v_{k-1}^*\|^2 \frac{\|v_k^*\|^2}{\|v_{k-1}^*\|^2} = d_{k-1}^{\text{vanha}} \frac{\|v_k^*\|^2}{\|v_{k-1}^*\|^2} \leq \frac{3}{4} d_{k-1}^{\text{vanha}}.$$

Jos vektoreiden vaihto suoritetaan N kertaa, niin D siis pienenee vähintään kertoimella $(3/4)^N$. Käyttäen samaa päättelyä kuin Lauseessa 3.3 voidaan osoittaa, että

$$\det(L_\ell)^2 = \prod_{i=1}^{\ell} \|v_i\|^2 = d_\ell.$$

Koska alihilan L_ℓ virittävät kantavektorit ovat kokonaislukuvektoreita, niin myös alihilan alideterminantti on kokonaisluku. Näin ollen suure

$$d_\ell = \prod_{i=1}^{\ell} \|v_i^*\|^2$$

on positiivinen kokonaisluku ja

$$D = \prod_{\ell=1}^{n} d_\ell \geq 1.$$

Lukua D voidaan siis pienentää kertoimella $3/4$ vain äärellisen monesti, joten myös rivi 14 suoritetaan äärellisen monta kertaa. □

On osoitettu, että tehokkaasti toteutettuna algoritmin aikakompleksisuus on $\mathcal{O}(n^6(\log B)^2)$, missä $B = \max \|v_i\|$ [11].

Esimerkki 3.8. Syötetään LLL-algoritmiin[3] kanta

$$V = \begin{pmatrix} -2 & -1 & -1 & 0 \\ 72 & -72 & -6 & 31 \\ 668 & -668 & -73 & 291 \\ -271 & 271 & 14 & -115 \end{pmatrix}.$$

[3]Tässä on käytetty Wolfram-kielen funktiota LatticeReduce, joka käyttää Storjohannin kehittämää algoritmin implementaatiota [15].

Kannan Hadamardin suhde on

$$\sqrt[4]{\frac{9}{3\sqrt{6}\sqrt{11365}\sqrt{109162}\sqrt{160303}}} \approx 0,02.$$

LLL-algoritmin tulosteena saadaan LLL-redusoitu kanta

$$V^{\mathrm{LLL}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ -1 & 1 & -1 & 0 \\ -1 & -2 & 0 & 0 \\ 1 & -1 & -2 & 0 \end{pmatrix}.$$

Redusoidun kannan Hadamardin suhde on

$$\sqrt[4]{\frac{9}{\sqrt{3}\sqrt{5}\sqrt{6}}} \approx 0,98.$$

Redusoitu kanta on siis selvästi ortogonaalisempi ja lyhyempi, kuten haluttiin.

3.3 LLL-algoritmin käyttö hilaongelmien ratkaisussa

Lauseesta 3.5 nähtiin, että LLL-algoritmin tulosteena saatavan kannan ensimmäinen kantavektori ratkaisee likimääräisesti lyhyimmän vektorin ongelman, kun kuvaukseksi $\psi(n)$ valitaan $2^{(n-1)/2}$. Algoritmia voidaan käyttää myös likimääräisesti lähimmän vektorin ongelman ratkaisemiseen.
Ilmiselvin lähestymistapa on käyttää LLL-algoritmia - ja saada näin LLL-redusoitu kanta - ja käyttää näin saatua kantaa alaluvussa 4.1 esiteltävässä Babain algoritmissa. Babain algoritmi toimii paremmin mitä ortogonaalisempi hilan kanta on.
Ravi Kannan julkaisi vuonna 1987 heuristisen tavan lähimmän vektorin ongelman ratkaisuun [10]. Tämä *upotustekniikkana* tästä edespäin kutsuttava menetelmä pyrkii muuttamaan lähimmän vektorin ongelman lyhyimmän vektorin ongelmaksi. Olkoon $v_1, \ldots, v_n$ n-asteisen hilan $L \subset \mathbb{Z}^n$ kanta ja $w \in \mathbb{Z}^n$ mielivaltainen kokonaislukuvektori, jonka suhteen lähimmän vektorin ongelma halutaan ratkaista. Lisäksi olkoon t lähimmän vektorin ongelman ratkaisu ja $e = t - w$. Muodostetaan nyt kanta

$$V' = \begin{pmatrix} w & 1 \\ v_1 & 0 \\ \vdots & \vdots \\ v_n & 0 \end{pmatrix}.$$

Kanta V' virittää hilan L'. Määritellään vektorit t', w' ja e' seuraavasti

$$t' = (t, 0),\ w' = (w, 1) \text{ ja } e' = (e, 1).$$

Tällä merkinnällä tarkoitetaan, että vektorit vastaavat vektoreita t, w ja e lukuunottamatta loppuun liitettyä uutta koordinaattia. Selvästi vektorit t' ja w' kuuluvat hilaan L'. Näin ollen vektoreiden erotus

$$e' = (w, 1) - (t, 0)$$

kuuluu myös hilaan L' ja on lisäksi lyhyt verrattuna muihin hilan vektoreihin. Tekniikan ideana on soveltaa LLL-algoritmia kantaan V' ja näin pyritään saamaan hilan L' lyhyin vektori. Upotustekniikan toimivuus riippuu siitä, kuinka pitkä vektori e' on verrattuna muihin hilan lyhyisiin vektoreihin.

Esimerkki 3.9. Olkoon

$$V = \begin{pmatrix} 300 & 4 & -7 \\ 0 & 14 & 3 \\ 8 & -10 & 13 \end{pmatrix}$$

ja $w = (-10, -10, -10)$. Nyt siis

$$V' = \begin{pmatrix} -10 & -10 & -10 & 1 \\ 300 & 4 & -7 & 0 \\ 0 & 14 & 3 & 0 \\ 8 & -10 & 13 & 0 \end{pmatrix}.$$

Syötetään kanta V' LLL-algoritmiin, jolloin saadaan tuloste

$$\begin{pmatrix} -2 & -6 & 6 & 1 \\ -12 & -2 & -1 & 2 \\ 2 & -8 & -9 & -1 \\ 8 & 0 & -7 & 50 \end{pmatrix}.$$

Upotustekniikalla saadaan siis tulokseksi $e = (-2, -6, 6)$, joten $t = (-8, -4, -16)$, joka on hilavektori, sillä $t = -v_2 - v_3$.

4 GGH-salausmenetelmä

Goldreich, Goldwasser ja Halevi julkaisivat lähimmän vektorin ongelmaan perustuvan salausmenetelmänsä vuonna 1997 [5]. Tässä luvussa tutustutaan sen toimintaperiaatteeseen ja esitellään hyökkäyksiä menetelmää vastaan.

4.1 Babain algoritmi

Babai julkaisi algoritminsa, joka ratkaisee likimääräisen lähimmän vektorin ongelman tietyillä ehdoilla, vuonna 1986 [1]. Motivoidaan ensin algoritmin toimintaa.

Jos hilan L kanta on ortogonaalinen, niin hilaongelmien ratkaisu on helppoa. Hilavektorin w pituuden neliö on muotoa

$$\begin{aligned}\|w\|^2 = w \cdot w &= (a_1 v_1 + a_2 v_2 + \cdots + a_n v_n) \cdot (a_1 v_1 + a_2 v_2 + \cdots + a_n v_n) \\ &= a_1^2 \|v_1\|^2 + a_2^2 \|v_2\|^2 + \cdots + a_n^2 \|v_n\|^2.\end{aligned}$$

Lyhyin hilavektori on siis lyhyin kantavektoreista, sillä sen kertoimeksi a_i voidaan asettaa 1 ja muut nollaksi.

Entä kuinka löytyy mielivaltaista vektoria t lähimpänä oleva hilavektori, kun kanta on ortogonaalinen? Kirjoitetaan t ensin muodossa

$$t = b_1 v_1 + b_2 v_2 + \cdots + b_n v_n, \text{ missä } b_1, \ldots, b_n \in \mathbb{R}.$$

Tämä luonnollisesti onnistuu, sillä vektorit $v_1, \ldots, v_n$ ovat hilan lisäksi myös avaruuden R^n kanta. Olkoon mielivaltainen hilavektori v muotoa

$$v = a_1 v_1 + a_2 v_2 + \cdots + a_n v_n.$$

Tällöin samalla päättelyllä kuin lyhyimmän vektorin tapauksessa vektoreiden v ja w etäisyyden neliö on

$$\|v - w\|^2 = (a_1 - b_1)^2 \|v_1\|^2 + \cdots + (a_n - b_n)^2 \|v_n\|^2.$$

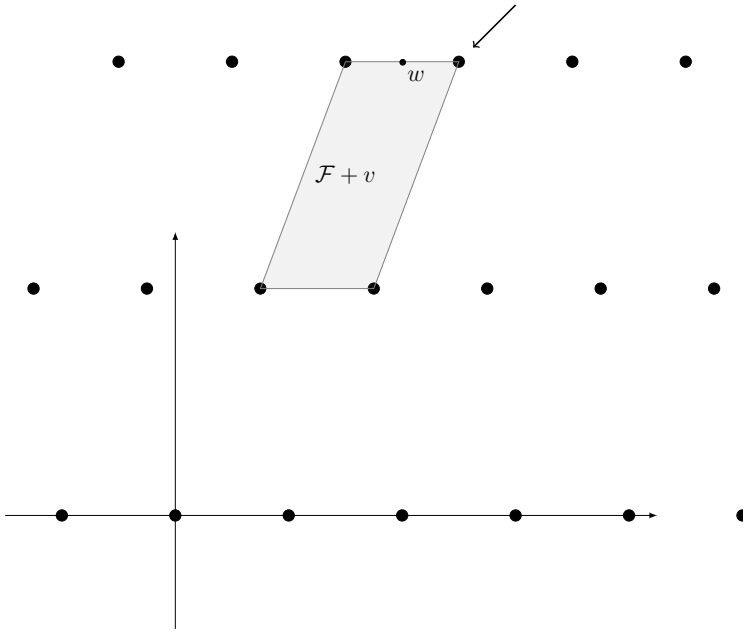
Etäisyys minimoituu, kun kertoimet $(a_i - b_i)$ ovat pienimmillään. Kertoimien a_i tulee olla kokonaislukuja, joten valitaan luvut a_i siten, että ne ovat vastaavaa lukua b_i lähimpänä oleva kokonaisluku. Valitaan siis $a_i = \lceil b_i \rfloor$.

Miten toimia yleisemmässä tapauksessa? Olkoon $v_1, v_2, \ldots, v_n$ hilan L kanta ja $\mathcal{F}$ hilan perusalue. Lauseen 1.17 perusteella perusalueen translaatiot täyttävät koko avaruuden, joten jokainen $w \in \mathbb{R}^n$ kuuluu yksikäsitteisesti johonkin perusalueen translaatioon $\mathcal{F} + v$, missä $v \in L$. Otamme lähimmän

vektorin ongelman ratkaisuksi perusalueen translaation vektoria w lähimpänä olevan kärjen. Menetelmää havainnollistetaan Kuvassa 3. Formaalisti

$$w = v + e_1 v_1 + e_2 v_2 + \cdots + e_n v_n, \text{ joillain } 0 \leq e_1, e_2, \ldots, e_n < 1.$$

Lähin kärki löydetään korvaamalla kerroin e_i luvulla 0, jos $e_i < \frac{1}{2}$ ja luvulla 1, jos $e_i \geq \frac{1}{2}$.



Kuva 3: Menetelmä antaa perusalueen translaation kärjen lähimmän vektorin ongelman ratkaisuksi.

Muotoa voidaan yksinkertaistaa yhdistämällä hilavektorin v kertoimet lukuihin e_i. Näin saadaan algoritmin lopullinen muoto.

Babain algoritmi:

Kirjoitetaan w muodossa $w = t_1v_1+t_2v_2+\cdots+t_nv_n$ missä $t_1, t_2, \ldots, t_n \in \mathbb{R}$.
Asetetaan $i = 1$.
while $i \leq n$ **do**
    Asetetaan $a_i = \lceil t_i \rfloor$.
    $i = i + 1$.
end while
return $u = a_1v_1 + a_2v_2 + \cdots + a_nv_n$.

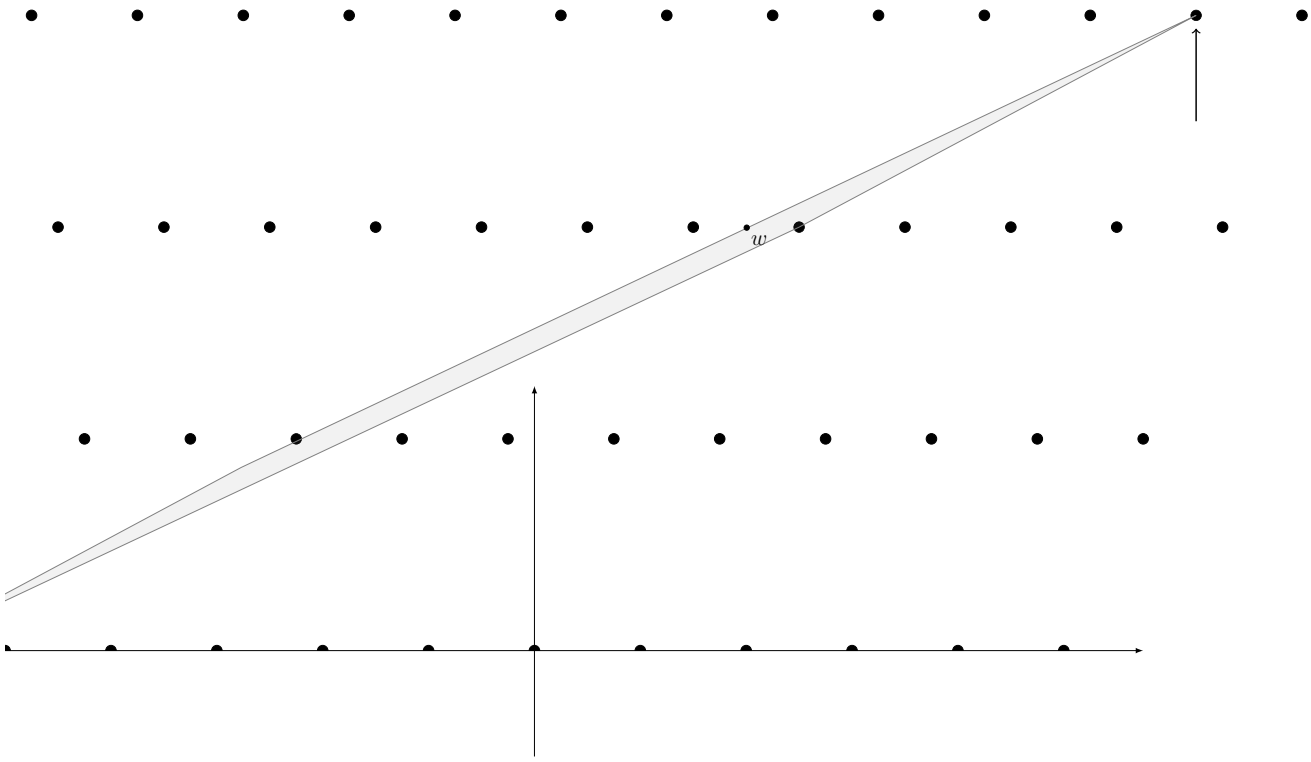
Vektorin w esittäminen kannan $v_1, v_2, \ldots, v_n$ avulla vaatii matriisiyhtälön $W = TV$ ratkaisemista. Yhtälö ratkaistaan muodostamalla matriisin V käänteismatriisi. Käänteismatriisin ratkaiseminen on aikakompleksisuusluokaltaan (käytännöllisillä algoritmeilla) $\mathcal{O}(n^3)$ [13], joten tämä on myös Babain algoritmin aikakompleksisuusluokka, sillä algoritmi ei sisällä muita operaatioita pyöristyksien lisäksi, jotka voidaan suorittaa merkityksettömässä ajassa.

Algoritmi on hyödyllistä esittää joskus matriisimuodossa

$$u = \lceil wV^{-1} \rfloor V,$$

jossa pyöristysmerkintä tarkoittaa, että jokainen vektorin alkio pyöristetään lähimpään kokonaislukuun.

Nyt ollaan siis löydetty algoritmi, joka näyttää ratkaisevan lähimmän vektorin ongelman. Algoritmi toimii kuitenkin vain, kun hilan kanta on riittävän ortogonaalinen. Jos näin ei ole, niin algoritmin palauttama vektori voi olla todella kaukana oikeasta ratkaisusta.



Kuva 4: Kuvassa on sama hila ja piste w kuin Kuvassa 3. Vaikka oikea vastaus on itseasiassa perusalueen translaation kärki, niin Babain algoritmi antaa väärän tuloksen. Katso Esimerkki 4.1.

Esimerkki 4.1. Käytetään kantaa $v_1 = (\frac{3}{2}, 4), v_2 = (2, 0)$ kuten Kuvassa 3 ja olkoon $w = (\frac{401}{100}, \frac{799}{100})$. Esitetään w kantavektoreiden avulla. Toisin sanoen on ratkaistava yhtälöpari

$$\begin{cases} t_1 \frac{3}{2} + t_2 2 & = \frac{401}{100} \\ t_1 4 & = \frac{799}{100} \end{cases} \implies \begin{cases} t_1 & = \frac{799}{400} \\ t_2 & = \frac{811}{1600} \end{cases}.$$

Seuraavaksi pyöristetään luvut t_1 ja t_2 lähimpiin kokonaislukuihin: $\lfloor t_1 \rceil = 2$ ja $\lfloor t_2 \rceil = 1$. Algoritmi antaa vastaukseksi

$$2(\frac{3}{2}, 4) + (2, 0) = (5, 8),$$

mikä onkin oikea tulos.

Käytetään sitten kantaa $v_1 = (\frac{15}{2}, 4)$ ja $v_2 = (17, 8)$ kuten Kuvassa 4. Nyt ratkaistava yhtälöpari on

$$\begin{cases} t_1\frac{15}{2} + t_2 17 & = \frac{401}{100} \\ t_1 4 + t_2 8 & = \frac{799}{100} \end{cases} \Longrightarrow \begin{cases} t_1 & = \frac{415}{32} \\ t_2 & = \frac{-8777}{1600} \end{cases}.$$

Pyöristyksien tulokset ovat $\lfloor t_1 \rceil = 13$ ja $\lfloor t_2 \rceil = -5$. Näin ollen algoritmi antaa tuloksen

$$13(\frac{15}{2}, 4) - 5(17, 8) = (\frac{25}{2}, 12),$$

joka on kaukana oikeasta tuloksesta.

4.2 Toimintaperiaate

Alicen yksityinen avain koostuu vektoreista $v_1, v_2, \ldots, v_n \in \mathbb{Z}^n$, jotka ovat lineaarisesti riippumattomat ja riittävän ortogonaaliset. Ortogonaalisuus voidaan tarkistaa Hadamardin suhteella. Nämä vektorit ovat kanta hilalle L. Seuraavaksi Alice luo "huonon"kannan (ortogonaalisuuden mielessä) kertomalla matriisia V - jossa vektorit $v_1, v_2, \ldots, v_n$ ovat riveinä - oikealta unimodulaarisella matriisilla A, jolloin

$$W = AV.$$

Matriisin W rivit ovat hilan L kanta Lauseen 1.15 perusteella. Kanta $w_1, w_2, \ldots, w_n$ on Alicen julkinen avain.

Bobin viestin tulee olla vektori m, jonka alkiot ovat kokonaislukuja. Lähettääkseen viestin Bob valitsee lyhyen virhevektorin r ja laskee vektorin

$$c = mW + r,$$

joka on salattu viesti. Virhevektori valitaan niin, että $r \in \{\pm\sigma\}^n$, missä julkinen parametri σ on positiivinen kokonaisluku.

Vektori c on siis lähellä hilavektoria mW. Näin ollen käyttämällä Babain algoritmia yksityisellä "hyvällä"kannallaan Alice saa ratkaistua hilavektorin mW. Viestin purku on yksinkertaista, sillä hilavektoria kerrotaan vain oikealta käänteismatriisilla W^{-1}.

Esimerkki 4.2. Olkoon Alicen salainen avain

$$V = \begin{pmatrix} 6 & 3 & 0 \\ 3 & -3 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$

Kanta on "hyvä", sillä Hadamardin suhde on

$$\sqrt[3]{\frac{3}{\sqrt{5}\sqrt{2}\cdot 1}} \approx 0,98.$$

Olkoon matriisi

$$A = \begin{pmatrix} 72 & 26 & 5 \\ 668 & 247 & 44 \\ -271 & -95 & -20 \end{pmatrix}, \ \det(A) = 1.$$

Alice laskee uuden kannan

$$W = AV = \begin{pmatrix} 510 & 138 & 15 \\ 4749 & 1263 & 132 \\ -1911 & -528 & -60 \end{pmatrix}.$$

Kannan Hadamardin suhde on

$$\sqrt[3]{\frac{1}{\sqrt{3449}\sqrt{22685066}\sqrt{437145}}} \approx 0,0025.$$

Tästä voidaan nähdä kannan "huonontamisen"periaate. Valitsemalla matriisi A siten, että sen alkiot ovat suuria tulee kannan W vektoreista todennäköisesti pitkiä. Huomautuksesta 1.20 tiedetään, että hilan determinantti ei riipu kannasta, joten tällöin Hadamardin suhde pienenee. Haluttu A on helppo muodostaa kertomalla keskenään useita unimodulaarisia matriiseja.

Bob haluaa lähettää viestin $m = (3, 1, 4)$ ja valitsee virhevektorin $r = (1, -1, 1)$. Salattu viesti c on siis

$$c = mW + r = (1729, 16037, -6500).$$

Alice käyttää Babain algoritmia "hyvällä"kannalla V ja ratkaisee hilavektorin mW.

$$\begin{cases} 6t_1 + 3t_2 &= 1729 \\ 3t_1 - 3t_2 &= 16037 \\ 3t_3 &= -6500 \end{cases} \implies \begin{cases} t_1 &= 1974 \\ t_2 &= -\frac{10115}{3} \\ t_3 &= -\frac{6500}{3} \end{cases}.$$

Pyöristyksien jälkeen Alice saa tuloksen

$$Wm = 1974v_1 - 3372v_2 - 2167v_3.$$

Käänteismatriisilla W^{-1} kertomalla hän tulee lopputulokseen $m = (3, 1, 4)$, joten salaus on purettu onnistuneesti.

Jos Eve yrittää purkaa salausta käyttämällä Babain algoritmia julkisella avaimella W, niin hän saa ratkaistavaksi yhtälöryhmän

$$\begin{cases} 510t_1 + 4749t_2 - 1911t_3 &= 1729 \\ 138t_1 + 1263t_2 - 528t_3 &= 16037 \\ 15t_1 + 132t_2 - 60t_3 &= -6500 \end{cases} \implies \begin{cases} t_1 &= \frac{41626360}{3} \\ t_2 &= 821755 \\ t_3 &= -1660894 \end{cases}.$$

Pyöristyksien jälkeen Eve hilavektorin

$$13875453w_1 + 821755w_2 - 16608954w_3.$$

Eve saa siis viestiksi $(13875453, 821755, -216608954)$. Salauksen murtaminen siis epäonnistui todella pahasti.

GGH:lla salaaminen ja salauksen purku on hyvin yksinkertaista; molempiin sisältyy vain matriisituloja ja/tai matriisin kääntämisiä. Näiden operaatioiden tiedetään olevan aikakompleksisuusluokkaa $\mathcal{O}(n^3)$ [13]. Yksinkertaisuus onkin yksi syy, miksi hilaongelmiin perustuvat salausmenetelmät ovat yleisemminkin kiinnostavia. Perinteisemmät menetelmät kuten RSA käyttävät useita huomattavasti vaativampia operaatioita kuten modulaarisia kertolaskuja.

Yksi järjestelmään liittyvä ongelma on se, kuinka σ tulee valita. Jos virhevektori r on liian pitkä, niin salattu viesti c voi olla lähempänä jotain toista hilavektoria kuin vektoria mW. Tällöin luonnollisesti tapahtuu purkamisvirhe, eli Alice saa purkamisen tuloksena jotain muuta kuin vektorin m.

Seuraavan lemman todistus perustuu Goldreichin, Goldwasserin ja Halevin [5] ja siitä seuraavan Lauseen todistus Hartmannin [6, s. 66-67] esitykseen.

Lemma 4.3. *Viestin purkamisessa tapahtuu virhe jos ja vain jos* $\lceil rV^{-1} \rfloor \neq 0$.

Todistus. Olkoon $T = VW^{-1}$. T on unimodulaarinen, sillä

$$W = AV \implies A = WV^{-1} \implies A^{-1} = VW^{-1},$$

ja kuten Lauseen 1.15 todistuksesta muistetaan, niin unimodulaarisen matriisin käänteismatriisi on unimodulaarinen. Käytetään salauksen purkamiselle matriisinotaatiota.

$$u = \lceil cV^{-1} \rfloor T.$$

Purkaminen onnistuu, jos $u = m$. Tutkitaan nyt, millä ehdolla tämä toteutuu.

$$u = \lceil cV^{-1} \rfloor T = \lceil (mW + r)V^{-1} \rfloor T = \lceil mWV^{-1} + rV^{-1} \rfloor T = \lceil mT^{-1} + rV^{-1} \rfloor T.$$

Matriisi T^{-1} on unimodulaarinen ja m on kokonaislukuvektori, joten mT^{-1} on kokonaislukuvektori ja voidaan ottaa pyöristyksen ulkopuolelle. Nyt

$$u = \lceil rV^{-1} \rfloor T + mT^{-1}T = \lceil rV^{-1} \rfloor T + m,$$

joten

$$u = m.$$

Tästä seuraa, että

$$\lceil rV^{-1} \rfloor T = 0,$$

mikä on yhtäpitävää sen kanssa, että

$$\lceil rV^{-1} \rfloor = 0.$$

□

Lause 4.4. *Olkoon V yksityinen avain ja ρ matriisin V^{-1} sarakkeiden L_1-normien maksimi. Jos $\sigma < \frac{1}{2\rho}$, niin purkuvirheitä ei tapahdu.*

Todistus. Osoitetaan, että jos $\sigma < \frac{1}{2\rho}$, niin $\lceil rV^{-1} \rfloor = 0$. Tällöin lause on todistettu edellisen lemman perusteella. Käytetään matriisin V^{-1} j:stä sarakkeesta merkintää V_j^{-1} ja sarakkeen alkioista merkintää v_i.

$$\lceil rV^{-1} \rfloor = 0 \leftrightarrow \lceil rV_j^{-1} \rfloor = 0 \text{ kaikilla } j = 1, 2, \ldots, n.$$

Mielivaltaisella j pätee

$$rV_j^{-1} = \sum_{i=1}^{n} r_i v_i \leq |r_1||v_i| = \sum_{i=1}^{n} \sigma |v_i| = \sigma \sum_{i=1}^{n} |v_i| = \sigma L_1(V_j^{-1}) \leq \sigma\rho.$$

On siis oltava

$$\sigma\rho < \frac{1}{2},$$

jotta $\lceil rV_j^{-1} \rfloor = 0$. Näin ollen

$$\sigma < \frac{1}{2\rho}.$$

□

4.3 Hyökkäyksiä GGH-salausmenetelmää vastaan

Alaluvussa 3.3 käsiteltiin menetelmiä, joilla lähimmän vektorin ongelmaa voidaan pyrkiä ratkaisemaan LLL-algoritmin avulla. Nämä menetelmät luonnollisesti soveltuvat myös GGH-salausmenetelmää vastaan hyökkäämiseen, sillä sen turvallisuus perustuu lähimmän vektorin ongelmaan.
Intuitiivisimpiä hyökkäyksiä on käyttää LLL-algoritmia julkiseen "huonoon"kantaan W, jolloin tulosteena saadaan ortogonaalisempi kanta, jota voidaan käyttää Babain algoritmissa lähimmän vektorin ongelman ratkaisuun. Hartmann havaitsi kokeellisesti, että tämän hyökkäyksen onnistumistodennäköisyys laskee huomattavasti, jos kantojen V ja W välillä oleva transformaatiomatriisi A on muodostettu kertomalla useita unimodulaarisia matriiseja keskenään. Jos kerrottavia matriiseja on 150, niin onnistumisprosentti on n. 20. Kerrottavien matriisien määrän ollessa 350 päästään jo hyvin lähelle nollaa. Hartmann totesi myös kokeellisesti, että Kannanin upotustekniikka on tehokas hyökkäys GGH:ta vastaan, kunhan käytettävän avaimen aste on pienempi kuin 200 [6, s.68-71].
Kiinnostavimman ja tehokkaimman hyökkäyksen esitteli Nguyen vuonna 1999 [12]. Muistetaan, että salattu viesti $c = mW + r$, missä $r \in \{\pm\sigma\}^n$. Täten

$$c \equiv mW \mod \sigma \text{ ja } c + (\sigma, \sigma, \ldots, \sigma) \equiv mW \mod 2\sigma.$$

Jos W on kääntyvä modulo 2σ (tarkoittaen matriisia, joka saadaan kun jokainen matriisin W alkio lasketaan modulo 2σ), niin saadaan ratkaistua m mod 2σ. Kääntyvyydestä ei kuitenkaan ole taetta. Tällöin $m \mod 2\sigma$ voidaan ratkaista modulaarisesta yhtälöryhmästä kiinalaisella jäännöslauseellla, sillä c ja W ovat tunnettuja.
Merkitään $m_{2\sigma} = m \mod 2\sigma$. Nyt tiedetään, että

$$m - m_{2\sigma} = 2\sigma m',$$

missä $m' \in \mathbb{Z}^n$. Nyt

$$c = mW + r \implies c = (m_{2\sigma} + 2\sigma m')W + r \implies \frac{c - m_{2\sigma}W}{2\sigma} = m'W - \frac{r}{2\sigma}.$$

Nyt siis ollaan saatu helpompi lähimmän vektorin ongelma, sillä virhevektorin pituus on $\sqrt{\frac{n}{4}}$, kun alkuperäisessä ongelmassa se oli $\sigma\sqrt{n}$. Seuraavassa lyhyt kuvaus hyökkäyksen toimintaperiaatteesta.

1. Ratkaistaan kaikki mahdolliset vektorit $m_{2\sigma}$.
2. Ratkaistaan vastaavat m' helpotetusta lähimmän vektorin ongelmasta upotustekniikalla.

3. Ratkaistaan $m = m_{2\sigma} + 2\sigma m'$.

4. Tarkistetaan vastauksen mielekkyys.

Tällä tekniikalla Nguyen onnistui murtamaan kaikki GGH:n julkistuksen yhteydessä haasteena annetut salatut viestit, joissa salausavaimen aste oli alle 400. Jos salausavaimen aste kasvaa 400, niin sen koko on niin iso, ettei sen käyttö ole käytännöllistä. Salausmenetelmä oli siis käytännössä murrettu.

5 NTRU-salausmenetelmä

NTRU-salausmenetelmän kehittivät Jeffrey Hoffstein, Jill Pipher ja Joseph H. Silverman vuonna 1996. Se on harvoja salausmenetelmiä, jota vastaan ei ole tunnettua kvanttitietokonehyökkäystä. Tämän vuoksi se on hyvin kiinnostava tutkimuksen kohde.

5.1 Konvoluutiopolynomirenkaat

Määritelmä 5.1. Olkoon N positiivinen kokonaisluku. *Astetta N konvoluutiopolynomirengas* on tekijärengas

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

Vastaavasti *konvoluutiopolynomirengas modulo q astetta N* on tekijärengas

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

Tässä tutkielmassa ei syvennytä renkaiden teorian perusteisiin. Edellä määriteltyjen rakenteiden ymmärtämiseksi on tiedettävä, että niiden alkiot ovat muotoa

$$a_{N-1}x^{N-1} + \cdots + a_1 x + a_0,$$

missä kertoimet a_i kuuluvat tekijärenkaan R tapauksessa kokonaislukuihin ja tekijärenkaan R_q tapauksessa kokonaislukuihin modulo q. Lisäksi alkioilla laskettaessa eksponentit esitetään modulo N, sillä lasketaan modulo $x^N - 1$, jolloin $x^N = 1$.
Alkio

$$a(x) = a_0 + a_1 x + \cdots + a_{N-1}x^{N-1} \in R$$

voidaan esittää vektorimuodossa

$$(a_0, a_1, \cdots, a_{N-1}) \in \mathbb{Z}^N.$$

Luonnollisesti esitys voidaan tehdä vastaavasti tekijärenkaan R_q alkioille. Tekijärenkaissa yhteenlasku toimii kuten vektoreilla, eli

$$a(x) + b(x) \leftrightarrow (a_0 + b_0, a_1 + b_1, \cdots, a_{N-1} + b_{N-1}).$$

Alkioiden tulo on määritelty seuraavassa lauseessa.

Lause 5.2. *Kahden tekijärenkaan R alkion tulo saadaan kaavasta*

$$(a_0, a_1, \ldots, a_{N-1}) * (b_0, b_1, \ldots, b_{N-1}) = (c_0, c_1, \ldots, c_{N-1}), \text{ missä } c_k = \sum_{i+j \equiv k \mod N} a_i b_j.$$

Tässä indeksit i ja j käyvät läpi joukon $\{0, 1, \ldots, N-1\}$ ja niiden on toteutettava ehto $i+j \equiv k \mod N$. Tekijärenkaalle R_q tulo toimii samalla tavalla, mutta kertoimet c_k ovat modulo q.

Todistus. Lasketaan ensin alkioille tavallinen tulo ja käytetään sitten tietoa $x^N = 1$.

$$a(x) * b(x) = (\sum_{i=0}^{N-1} a_i x^i) * (\sum_{j=0}^{N-1} b_j x^j) = \sum_{k=0}^{2N-2} (\sum_{i+j=k} a_i b_j) x^k =$$

$$\sum_{k=0}^{N-1} (\sum_{i+j=k} a_i b_j) x^k + \sum_{k=N}^{2N-2} (\sum_{i+j=k} a_i b_j) x^{k-N} = \sum_{k=0}^{N-1} (\sum_{i+j=k} a_i b_j) x^k + \sum_{k=0}^{N-2} (\sum_{i+j=k+N} a_i b_j) x^k$$

$$= \sum_{k=0}^{N-1} (\sum_{i+j \equiv k \mod N} a_i b_j) x^k.$$

□

Monimutkaisen näköisen kaavan varsinainen idea on siinä, että alkioille lasketaan tavallinen polynomien tulo ja sitten otetaan eksponentit modulo N.

Esimerkki 5.3. Olkoon $R = \frac{\mathbb{Z}[x]}{x^4-1}$,

$$a(x) = 3 - 5x + 2x^3 \text{ ja } b(x) = 1 + 4x + 3x^2 + 6x^3.$$

Tällöin

$$a(x)*b(x) = 3+12x+9x^2+18x^3-5x-20x^2-15x^3-30x^4+2x^3+8x^4+6x^5+12x^6$$

$$= 3+12x+9x^2+18x^3-5x-20x^2-15x^3-30+2x^3+8+6x+12x^2 = -19+13x+x^2+5x^3.$$

Jos toimitaan tekijärenkaassa R_7, niin

$$a(x) * b(x) = 2 + 6x + x^2 + 5x^5.$$

Määritelmä 5.4. Olkoon $a(x) \in R_q$. Tällöin alkion $a(x)$ *keskusnosto tekijärenkaalle* R on yksikäsitteinen alkio $a'(x) \in R$, jolle pätee

$$a'(x) \mod q = a(x),$$

missä alkion $a'(x)$ kertoimet a'_i ovat välillä $]-\frac{q}{2}, \frac{q}{2}]$.

Esimerkki 5.5. Olkoon $N = 5,\ q = 5$ ja

$$a(x) = -3 + 2x + x^3 + 4x^4 \in R_5.$$

Keskusnoston kertoimet kuuluvat joukkoon $\{-2, -1, 0, 1, 2\}$, joten keskusnosto

$$a'(x) = 2 + 2x + x^3 - x^4.$$

Käydään seuraavaksi läpi joitain tuloksia liittyen käänteisalkioihin.

Lause 5.6. *Olkoon $a(x)$ ja $b(x)$ polynomeja polynomirenkaassa $(\mathbb{Z}/q\mathbb{Z})[x]$, missä q on alkuluku. Tällöin pätee*

$$a(x) = b(x)k(x)+r(x),\ \textit{missä}\ k(x), r(x) \in (\mathbb{Z}/q\mathbb{Z})[x]\ \textit{ja}\ r(x) = 0\ \textit{tai}\ \deg r(x) < \deg b(x).$$

Todistus. Todistus ohitetaan. Todistus löytyy esimerkiksi lähteestä [7] sivuilta 99-100. □

Lause 5.7 (Laajennettu Eukleideen algoritmi)**.** *Olkoon $a(x)$ ja $b(x) \neq 0$ polynomeja polynomirenkaassa $(\mathbb{Z}/q\mathbb{Z})[x]$, missä q on alkuluku. Tällöin seuraavalla algoritmilla saadaan polynomien* suurin yhteinen tekijä $d(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$.

$$\begin{aligned}
a(x) &= b(x)k_1(x) + r_2(x),\ \textit{missä}\ 0 \leq \deg r_2(x) < \deg b(x),\\
b(x) &= r_2(x)k_2(x) + r_3(x),\ \textit{missä}\ 0 \leq \deg r_3(x) < \deg r_2(x),\\
r_2(x) &= r_3(x)k_3(x) + r_4(x),\ \textit{missä}\ 0 \leq \deg r_4(x) < \deg r_3(x),\\
&\vdots\ \vdots\\
r_{t-2}(x) &= r_{t-1}(x)k_{t-1}(x) + r_t(x),\ \textit{missä}\ 0 \leq \deg r_t(x) < \deg r_{t-1}(x),\\
r_{t-1}(x) &= r_t(x)k_t(x),\\
r_t(x) &= d(x) = \mathrm{syt}((a(x), b(x)).
\end{aligned}$$

Lisäksi pätee, että on olemassa polynomit $u(x), v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$ siten, että

$$a(x)u(x) + b(x)v(x) = d(x).$$

Todistus. Väite seuraa, kun Lausetta 5.6 toistetaan. Algoritmi päättyy selvästi, sillä jakojäännösten $r_i(x)$ asteet muodostavat aidosti vähenevän kokonaislukujonon, joka on alhaalta rajoitettu. Polynomit $u(x)$ ja $v(x)$ saadaan käyttämällä algoritmia lopputuloksesta takaisin päin. □

Esimerkki 5.8. Olkoon $q = 7$,

$$a(x) = 4x^3 + x^2 + 4x + 5 \text{ ja } b(x) = x^3 + x + 2.$$

Käytetään laajennettua Eukleideen algoritmia. Muistetaan, että laskeminen tapahtuu modulo 7.

$$\begin{aligned} 4x^3 + x^2 + 4x + 5 &= (x^3 + x + 2)4 + x^2 + 4 \\ x^3 + x + 2 &= (x^2 + 4)x + 4x + 2 \\ x^2 + 4 &= (4x + 2)2x + 3x + 4 \\ 4x + 2 &= (3x + 4)6 + 6 \\ 3x + 4 &= (4x + 3)6 \end{aligned}$$

Nähdään, että $\text{syt}(a(x), b(x)) = 6$. Algoritmista saadaan myös tulos

$$6 = (6x^2 + 2x + 4)(x^3 + x + 2) + (2x^2 + 6x + 1)(4x^3 + x^2 + 4x + 5).$$

Lause 5.9. *Olkoon q alkuluku. Tällöin alkiolla $a(x) \in R_q$ on käänteisalkio kertolaskun suhteen jos ja vain jos*

$$\text{syt}(a(x), x^N - 1) = 1 \text{ } \textit{polynomirenkaassa } (\mathbb{Z}/q\mathbb{Z})[x].$$

Käänteisalkio saadaan Eukleideen algoritmilla laskemalla polynomit $u(x), v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$, joille pätee

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

Tällöin käänteisalkio $a^{-1}(x) = u(x)$ polynomirenkaassa R_q.

Todistus. Edeltävistä lauseista tiedetään, että on olemassa polynomit $u(x), v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$, joille pätee

$$a(x)u(x) + (x^N - 1)v(x) = \text{syt}(a(x), x^N - 1).$$

Jos suurin yhteinen tekijä on 1, niin laskettaessa modulo $x^N - 1$ tästä seuraa suoraan, että $a(x) * u(x) = 1$ tekijärenkaassa R_q. Toisaalta, jos on olemassa käänteisalkio $u(x)$, siten että $a(x) * u(x) = 1$ tekijärenkaassa R_q, niin tämä tarkoittaa, että

$$a(x)u(x) \equiv 1 \mod x^N - 1.$$

Kongruenssin määritelmästä seuraa, että

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

polynomirenkaassa $(\mathbb{Z}/q\mathbb{Z})[x]$. □

5.2 Toimintaperiaate

Nyt on koottu työkalut NTRU-salausmenetelmän kuvaamiseen. Olkoot p ja q alkulukuja. Alkion $a(x) \in R$ ajatellaan kuuluvan tilanteesta riippuen konvoluutiopolynomirenkaisiin R_q tai R_p. Vastaavasti konvoluutiopolynomirenkaan R_p tai R_q alkio voidaan keskusnostaa renkaan R alkioksi. Oletetaan lisäksi, että N on alkuluku, $\text{syt}(N, q) = \text{syt}(p, q) = 1$ ja $q > (6d + 1)p$.

Määritelmä 5.10. Positiivisilla kokonaisluvuilla d_1 ja d_2 määritellään joukko

$$\mathcal{T}(d_1, d_2) = \left\{ a(x) \in R : \begin{array}{c} \text{polynomilla } a(x) \text{ on } d_1 \text{ kerrointa, jotka ovat 1.} \\ \text{polynomilla } a(x) \text{ on } d_2 \text{ kerrointa, jotka ovat -1.} \\ \text{polynomin } a(x) \text{ muut kertoimet ovat 0.} \end{array} \right\}.$$

Alice valitsee julkiset parametrit (N, p, q, d). Hänen yksityinen avaimensa koostuu kahdesta satunnaisesta polynomista

$$f(x) \in \mathcal{T}(d + 1, d) \text{ ja } g(x) \in \mathcal{T}(d, d).$$

Tämän jälkeen hän laskee käänteisalkiot

$$F_q(x) = f(x)^{-1} \in R_q \text{ ja } F_p(x) = f(x)^{-1} \in R_p.$$

Jos käänteisalkioita ei ole olemassa, niin Alice valitsee uuden polynomin $f(x)$. Seuraavaksi Alice laskee julkisen avaimen

$$h(x) = F_q(x) * g(x) \ \in R_q.$$

Bobin viesti on polynomi $m(x) \in R$, joka on konvoluutiopolynomirenkaan R_p jonkin alkion keskusnosto, eli pätee $-\frac{1}{2}p < m_i \leq \frac{1}{2}p$. Bob valitsee satunnaisen alkion joukosta $r(x) \in \mathcal{T}(d, d)$ ja laskee alkion

$$e(x) \equiv ph(x) * r(x) + m(x) \mod q.$$

Salattu viesti on $e(x) \in R_q$.
Alice purkaa viestin laskemalla ensin alkion

$$a(x) \equiv f(x) * e(x) \mod q.$$

Tämän jälkeen hän keskusnostaa alkion $a(x)$ renkaalle R ja laskee alkion

$$b(x) \equiv F_p(x) * a(x) \mod p.$$

Osoitetaan seuraavaksi, että purkaminen on onnistunut, eli $b(x) = m(x)$.

Lause 5.11. *Jos parametrit* (N, p, q, d) *on valittu oikein, erityisesti niin, että*

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

niin $b(x) = m(x)$ *ja salaus on purettu onnistuneesti.*

Todistus. Tutkitaan tarkemmin alkiota $a(x)$.

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

Tutkitaan nyt polynomia

$$pg(x) * r(x) + f(x) * m(x)$$

renkaassa R. Polynomit $g(x)$ ja $r(x)$ kuuluvat joukkoon $\mathcal{T}(d, d)$, joten jos kaikki kertoimet, jotka ovat 1 ja kaikki kertoimet, jotka ovat -1 osuvat kohdakkain, niin tulon $g(x) * r(x)$ suurin kerroin on $2d$. Vastaavasti $f(x)$ kuuluu joukkoon $\mathcal{T}(d+1, d)$ ja alkion $m(x)$ kertoimet kuuluvat välille $]-\frac{1}{2}p, \frac{1}{2}p]$, joten tulon $f(x) * m(x)$ suurin mahdollinen kerroin on $f(x) * m(x)$ on $(2d+1)\frac{1}{2}p$. Näin ollen polynomin $pg(x) * r(x) + f(x) * m(x)$ suurin mahdollinen kerroin on

$$2dp + (2d+1)\frac{1}{2}p = (3d + \frac{1}{2})p,$$

joten kun ehto $q > (6d+1)p$ on voimassa, niin kaikki kertoimet ovat aidosti pienempiä kuin $\frac{1}{2}q$. Tästä seuraa, että alkio $a(x)$ modulo q pysyy muuttumattomana, kun Alice keskusnostaa sen renkaalle R.
Muistetaan lopuksi, että $F_p(x) = f(x)^{-1}$ modulo p. Näin ollen

$$\begin{aligned} b(x) &\equiv F_p(x) * a(x) \mod p \\ &\equiv F_p(x) * (pg(x) * r(x) + f(x) * m(x)) \mod p \\ &\equiv F_p(x) * f(x) * m(x) \mod p \\ &\equiv m(x) \mod p \end{aligned}$$

□

5.3 NTRU-salausmenetelmä hilojen teorian näkökulmasta

Edellä esitetyllä ei ensisilmäyksellä näytä olevan mitään tekemistä hilojen kanssa. Seuraavaksi kuitenkin näytetään, että salaisten polynomien $f(x)$ ja

$g(x)$ ratkaiseminen palautuu lyhyimmän vektorin ongelman ratkaisemiseksi tietynlaisessa hilassa.
Huomataan ensin, että polynomien välillä pätee seuraavanlainen yhteys:

$$f(x) * h(x) \equiv g(x) \mod q,$$

sillä $F_q(x) * g(x) = h(x)$ konvoluutiopolynomirenkaassa R_q. Huomataan lisäksi, että jos pari $(f(x), g(x))$ ratkaisee kongruenssin, niin myös $(x^k * f(x), x^k * g(x)$ on ratkaisu kaikilla $k = 0, 1, \ldots, N-1$. Polynomia $x^k * f(x)$ kutsutaan polynomin $f(x)$ *rotaatioksi*. Jos rotaatiota käytetään salauksen purkuun, niin saadaan viestin rotaatio $x^k * m(x)$.

Määritelmä 5.12. Olkoon

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

NTRU-salausmenetelmän julkinen avain. Tällöin *NTRU-hila* L_h^{NTRU} on $2N$-asteinen hila, jonka virittää kanta

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

Kantavektorit ovat totutusti matriisin M_h^{NTRU} rivit. Matriisi koostuu neljästä $N \times N$ lohkosta, joten voidaan merkitä

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

Olkoon

$$a(x) = a_0 + a_1 x + \cdots + a_{N-1} x^{N-1} \text{ ja } b(x) = b_0 + b_1 x + \cdots + b_{N-1} x^{N-1}$$

renkaan R polynomeja. Merkitään

$$(a, b) = (a_0, a_1, \ldots, a_{N-1}, b_0, b_1, \ldots, b_{N-1}) \ \in \mathbb{Z}^{2N}.$$

Lause 5.13. *Olkoon* $f(x) * h(x) \equiv g(x) \mod q$ *ja* $u(x) \in R$ *polynomi, jolle pätee*

$$f(x) * h(x) = g(x) + qu(x).$$

Tällöin

$$(f, -u)M_h^{NTRU} = (f, g),$$

eli vektori (f, g) *on kantavektoreiden lineaarikombinaatio, jossa kertoimet ovat kokonaislukuja. Vektori* (f, g) *siis kuuluu hilaan* L_h^{NTRU}.

Todistus. Matriisiteorian perusteella voidaan käyttää matriisin M_h^{NTRU} lohkoesitystä. Tällöin

$$(f, -u)M_h^{\mathrm{NTRU}} = (f, -u)\begin{pmatrix} 1 & h \\ 0 & q \end{pmatrix} = (f, f * h - qu) = (f, g).$$

□

Lause 5.14. *Olkoot* (N, p, q, d) *NTRU-salausmenetelmän parametrit. Oletetaan yksinkertaisuuden vuoksi*

$$p = 3,\ d \approx N/3 \text{ ja } q \approx 6pd \approx 2pN.$$

Tällöin pätee

1. $\det(L_h^{NTRU}) = q^N$,
2. $\|(f, g)\| \approx \sqrt{4d} \approx \sqrt{4N/3} \approx 1,155\sqrt{N}$,
3. *Jos* N *on suuri, niin suurella todennäköisyydellä hilan lyhyimmät nollasta eroavat vektorit ovat* (f, g) *ja sen rotaatiot.*

Todistus.

1. Tunnetusti $\det(L_h^{\mathrm{NTRU}}) = |\det(M_h^{\mathrm{NTRU}})|$. Matriisi on yläkolmiomatriisi, joten sen determinantti on diagonaalialkioiden tulo, mikä on q^N.

2. Muistetaan, että polynomeissa $f(x)$ ja $g(x)$ on molemmissa arviolta d kerrointa, jotka ovat 1 ja d, jotka ovat -1. Muut ovat nollia. Väite seuraa euklidisen normin määritelmästä.

3. Gaussin heuristiikka ennustaa, että hilan lyhyin vektori on pituudeltaan noin
$$\sqrt{\frac{2N}{2\pi e}} = (\det(L_h^{\mathrm{NTRU}}))^{1/2N}\sqrt{Nq/\pi e} \approx 0,838N.$$

Kun N on suuri, niin

$$\frac{\|(f,g)\|}{\sigma(L_h^{\mathrm{NTRU}})} \approx \frac{1,155\sqrt{N}}{0.838N} \approx \frac{1,38}{\sqrt{N}}.$$

Vektorin (f,g) pituus on siis arviolta $1.38/\sqrt{N}$ kertaa Gaussin heuristiikan ennustaman lyhyimmän nollasta eroavan vektorin pituus.

□

Edeltävän lauseen perusteella siis salaisten polynomien $f(x)$ ja $g(x)$ ratkaiseminen palautuu lyhyimmän vektorin ongelmaksi hilassa L_h^{NTRU}. Tästä näkökulmasta luonnollinen hyökkäys on käyttää LLL-algoritmia kantaan M_h^{NTRU}. Ongelmana on, että Lauseesta 3.5 tiedetään, että LLL-algoritmi ratkaisee likimääräisen lyhyimmän vektorin ongelman, kun kuvaukseksi $\psi(n)$ valitaan $2^{(n-1)/2}$, eli tässä tapauksessa $2^{N-1/2}$. Luvun N kasvaessa suureksi kerroin kasvaa isoksi ja vektorin (f,g) löytyminen LLL-algoritmilla muuttuu epätodennäköiseksi.

Huomautus 5.15. LLL-algoritmista on kehitetty yleistys, jota kutsutaan BKZ-LLL-algoritmiksi. Tämä algoritmi pystyy ratkaisemaan likimääräisesti lyhyimmän vektorin ongelman, kun kuvaukseksi $\psi(n)$ valitaan $\beta^{2N/\beta}$. Algoritmin aikakompleksisuus on kuitenkin eksponentiaalinen parametrin β suhteen. Kokeellisesti on todettu, että NTRU-salausmenetelmän turvallisuus on samaa luokkaa RSA ja Elgamal salausmenetelmien kanssa, kun N kuuluu välille $[250, 1000]$ [8].

Lähdeluettelo

[1] L. Babai: *On Lova'sz' lattice reduction and the nearest lattice point problem.* Combinatorica 6(1):1–13, 1986.

[2] S. Galbraith: *Mathematics of Public Key Cryptography.* University Press Cambridge, 2012.

[3] S. Galbraith: *Mathematics of Public Key Cryptography.* https://www.math.auckland.ac.nz/ sgal018/crypto-book/ch19a.pdf, viitattu 12.7.2017.

[4] O. Goldreich: *Foundations of Cryptography.* Cambridge University Press, 2001.

[5] O. Goldreich, S. Goldwasser ja S. Halevi: *Public-key cryptosystems from lattice reduction problems. In CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology.* 112–131, Springer-Verlag, 1997.

[6] M. Hartmann: *The Ajtai-Dwork Cryptosystem and Other Cryptosystems Based on Lattices*(Master's Thesis). http://www.math.uzh.ch/index.php?file&key1=35477, viitattu 27.7.2017.

[7] J. Hoffstein, J. Pipher ja J. H. Silverman: *An Introduction to Mathematical Cryptography.* Springer-Verlag New York, 2008.

[8] J. Hoffstein, J. H. Silverman ja William Whyte: *Estimated Breaking Times for NTRU Lattices.* NTRU Cryptosystems, Inc., 2003.

[9] E. Järvenpää: *Lineaarialgebra I.* cc.oulu.fi/ tvedenju/linalg1/files/linalg1.pdf, viitattu 7.6.2017.

[10] R. Kannan: Minkowski's convex body theorem and integer programming. Math. Oper. Res., 12(3):415–440, 1987.

[11] A.K. Lenstra, H.W. Lenstra ja L. Lovász: *Factoring polynomials with rational coefficients*, Mathematische Annalen 261, 515-534, Springer-Verlag, 1982.

[12] P. Q. Nguyen: *Cryptanalysis of the Goldreich–Goldwasser–Halevi Cryptosystem from Crypto '97.* In CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, s. 288–304, 1999. Springer-Verlag.

[13] M. Petkovic ja P. Stanimirovic: *Generalized matrix inversion is not harder than matrix multiplication*, Journal of Computational and Applied Mathematics Volume 230, 270-282, 2009.

[14] A. Shamir: *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem*, Foundations of Computer Science, 1982. SFCS '08. 23rd Annual Symposium on, Chicago, IL, USA, 3.-5.11.1982. IEEE

[15] A. Storjohann: *Faster Algorithms for Integer Lattice Basis Reduction.* Technical Report 249. Zurich, Sveitsi: Department Informatik, ETH. 30.7.1996.

[16] Tekijä tuntematon: *Lineaarialgebra II.* noppa.oulu.fi/noppa/kurssi/802119p/materiaali/802119P_laii.pdf, viitattu 7.6.2017.