Rácsok és algoritmusok a kriptográfiában

Fazekas Péter László

Témavezető: Kutas Péter

1.1. Összefoglalás

Ezen évben témavezetőm Kutas Péter segítségével beleástam magam a rácsokon adott algoritmikus problémák kriptográfiai vonatkozásaiba. Ezek mellett megismerkedtem a duális rács geometriájával, amely elmélet nagyon erősnek bizonyul a különböző algoritmikus problémák megoldásában, megismerkedtem számomra új felsorolós algoritmusokkal, és befejeztem egy általam már korábban elkezdett projektet, ami lényege, hogy a felsorolós algoritmusok futásidejét egy-egy a rácsok geometriáját leíró konstanssal kössem össze.

1.2. Rácsok

Azt mondjuk, hogy $\mathbb{R}^d$ egy D részhalmaza diszkrét, ha létezik egy c pozitív valós szám, hogy tetszőleges $\mathbf{x},\mathbf{y} \in D$ esetén $\|\mathbf{x}-\mathbf{y}\| \geq c$, ahol $\|.\|$ most és a továbbiakban a 2-es normát jelöli. Az euklideszi vektorterünkön az ezt indukáló skalárszorzást $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$ esetén $\mathbf{x}\cdot\mathbf{y}$ módon jelöljük.

Az $\mathbb{R}^d$ egy diszkrét, additív részcsoportját *rácsnak* hívjuk. Minden $L \subset \mathbb{R}^d$ rácshoz léteznek $\mathbf{b}_1,\ldots,\mathbf{b}_n \in \mathbb{R}^d$ lineárisan független vektorok, hogy

$$L = \{s_1\mathbf{b}_1 + \ldots + s_n\mathbf{b}_n \mid s_i \in \mathbb{Z}\}.$$

Ezen vektorokat az L rács *bázisának*, az L rácsot pedig a $\mathbf{b}_1,\ldots,\mathbf{b}_n$ vektorok által *generált* rácsnak hívjuk. Könnyen meggondolható, hogy tetszőleges $\mathbf{b}_1,\ldots,\mathbf{b}_n$ lineárisan független vektorok egész együtthatós lineáris kombinációi egy rácsot alkotnak. Ezen rácsra az $L(\mathbf{b}_1,\ldots,\mathbf{b}_n)$ jelölést is használjuk. Az n pozitív egész számot a rács *dimenziójának* hívjuk.

Egy rács egy bázisára mátrixként is gondolhatunk, ahol bázis vektorai alkotják a mátrix oszlopait. Ha $B \in \mathbb{R}^{d\times n}$ az L rács bázisa, akkor a rács detereminánsa $\det L = \sqrt{\det B^T B}$. Egy legalább 2 dimenziós rácsnak végtelen sok bázisa van. A $B, B' \in \mathbb{R}^{d\times n}$ bázisok pontosan akkor generálják ugyanazt a rácsot, ha létezik $U \in \mathrm{GL}_n(\mathbb{Z})$, hogy $B' = BU$. Ezek szerint egy rács determinánsa jól definiált, a fenti determináns értéke a rács tetszőleges bázisa esetén ugyanaz marad.

Legyen $\mathbf{b}_1,\ldots,\mathbf{b}_n$ az $L \subset \mathbb{R}^d$ rács bázisa. A bázis Gram-Schmidt ortogonalizáltját általában $\mathbf{b}_1^*,\ldots,\mathbf{b}_n^*$ jelöli. Ekkor a rácshoz három alapvető tartomány tartozik:

1. A bázishoz tartozó fundamentális paralelepipedon:

$$P(\mathbf{b}_1,\ldots,\mathbf{b}_n) = P = \{x_1\mathbf{b}_1 + \ldots + x_n\mathbf{b}_n \mid -1/2 < x_i \leq 1/2\}$$

2. A bázishoz tartozó fundamentális téglatest:

$$R(\mathbf{b}_1,\ldots,\mathbf{b}_n) = R = \{x_1\mathbf{b}_1^* + \ldots + x_n\mathbf{b}_n^* \mid -1/2 < x_i \leq 1/2\}$$

3. A rács Voronoi-cellája [8]:

$$V(L) = V = \{\mathbf{x} \in \mathbb{R}^n \mid \forall \mathbf{y} \in {}^{L}\!/_{-\mathbf{z} \sim \mathbf{z}} \colon \|\mathbf{x}+\mathbf{y}\| < \|\mathbf{x}\| \leq \|\mathbf{x}-\mathbf{y}\|\}$$

Ezen három test rácspontokba való eltoltjai a rács egy-egy parkettázását adják, továbbá

$$\operatorname{vol} P = \operatorname{vol} R = \operatorname{vol} V = \det L.$$

Az L rács *fedési sugara* a

$$\mu(L) = \sup_{\mathbf{x} \in V(L)} \|\mathbf{x}\|$$

konstans, ami a rács Voronoi-cellája átmérőjének fele.

1.3. Alapvető algoritmikus problémák és bázisredukció

Nem mindegy, hogy egy rácsot melyik bázisával reprezentáljuk. Általában azt a bázist szeretjük jobban, amely vektorai közelebb vannak a merőlegeshez. Itt az algoritmikus problémák jobban kezelhetők. Ezt remekül illusztrálja a CVP (Closest Vector Problem) algoritmikus probléma megoldása olyan rács esetén, amelynek adott egy egymásra merőleges vektorokból álló bázisa.

Itt a feladat a következő: adott egy $B \in \mathbb{Q}^{d \times n}$ bázis és egy $\mathbf{t} \in \mathbb{Q}^d$ vektor. Találjuk meg azt az $\mathbf{x} \in L(B)$ rácsvektort, amelyre $\|\mathbf{x}-\mathbf{t}\|$ minimális (mivel $L(B)$ egy diszkrét halmaz, így ilyen létezik). Ezen probléma az általános felállásban NP-nehéz [5]. Mi a helyzet akkor, ha a rácsunknak egy olyan bázisa adott, amely vektorai merőlegesek egymásra? Ha $L(B) = \mathbb{Z}^n$, a feladatunk nagyon egyszerű: a $\mathbf{t}$ vektort koordinátánként a legközelebbi egészre kell kerekíteni, és meg is találtuk a legközelebbi rácsvektort. A feladat hasonlóan megoldható tetszőleges egymásra merőleges bázisvektorokkal adott rács esetén is.

Egy másik alapvető algoritmikus probléma rácsokon az SVP (Shortest Vektor Problem). Itt a feladat a következő: Adott egy $B \in \mathbb{Q}^{d \times n}$ bázis. Találjuk meg azon $\mathbf{x} \in L(B) \setminus \{\mathbf{0}\}$ rácsvektort, amelyre $\|\mathbf{x}\|$ minimális. (Azaz találjuk meg a B bázis által generált rács legrövidebb nem-$\mathbf{0}$ vektorát.) Ha a bázisunk egymásra merőleges vektorokból áll, akkor ezen probléma megoldása is nagyon egyszerű: válasszuk ki a legrövidebb bázisvektort.

Egy L rács legrövidebb nem-$\mathbf{0}$ vektorának hosszát $\lambda_1(L)$ jelöli. A legrövidebb vektor hosszára Minkowski felső becslése a következő (lásd például [13] 12. oldal 1.24):

$$\lambda_1(L) \leq \sqrt{n}(\det L)^{1/n}.$$

Immár több évtizedes nyitott kérdés, hogy SVP NP-nehéz probléma-e a klasszikus értelemben. Jelenleg annyi ismert, hogy SVP NP-nehéz véletlenszerű visszavezetés mellett, azaz ha SVP-re létezne egy polinomiális algoritmus abból NP = RP következne. [5]

Ezen problémák megoldásában alapvető eszköz, a bázisredukció [17], azaz a rács egy olyan bázisának találása, amely vektorai közel merőlegesek egymásra. Most két alapvető ilyen algoritmust emelnék ki. Az egyik a HKZ (Hermit-Kortain-Zoltarev) algoritmus, a másik a híres LLL (Lenstra-Lenstra-Lovász) algoritmus, amely polinomiális időben kellően merőleges bázist talál.

1.4. Dualitás

A rácsok duálisának geometriájáról egy hosszabb bevezető található a [1] és [2] helyeken. Erről adok egy rövid összefoglalót itt.

Legyen $L \subset \mathbb{R}^d$ egy n dimenziós rács. Ekkor az

$$\hat{L} \stackrel{\text{def}}{=} \{\mathbf{y} \in \text{Span}(L) \mid \forall \mathbf{x} \in L\colon \mathbf{x} \cdot \mathbf{y} \in \mathbb{Z}\}$$

rácsot az L rács *duális rácsának* hívjuk. Könnyen meggondolható, hogy ez tényleg egy rácsot definiál. Ha adott az L rács egy $B \in \mathbb{R}^{d \times n}$ bázisa, akkor ezen bázishoz tartozik a duális rács egy $D \in \mathbb{R}^{d \times n}$ bázisa, amelyet egyértelműen definiál a

$$D^T B = I$$

egyenlet, ahol I a megfelelő méretű identitásmátrix.

Ha adott egy $H \subset \mathbb{R}^d$ altér, akkor jelölje π_H az ezen altérre merőleges vetítést. Ha adott egy L rács egy $\mathbf{b}_1, \ldots, \mathbf{b}_n$ bázisa, akkor $i > 1$ esetén legyen $\pi_i = \pi_{\text{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})}$, továbbá π_1 jelölje az identitást.

Legyen $H_i = \text{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n)$, és $\tilde{\mathbf{b}}_i = \pi_{H_i}(\mathbf{b}_i)$. Ekkor a $\mathbf{b}_1, \ldots, \mathbf{b}_n$ bázishoz tartozó duális bázist

$$\mathbf{d}_i = \frac{\mathbf{b}_i}{\|\tilde{\mathbf{b}}_i\|^2}$$

módon kapjuk meg, ami egy geometriai képet ad a duális bázisról: A $\mathbf{d}_i$ duális bázisvektort úgy kapjuk meg, hogy a $\mathbf{b}_i$ vektort merőlegesen vetítjük a többi bázisvektorra, majd azt a kapott vektorral azonos irányba álló vektort vesszük, amely hossza a kapott vektor hosszának reciproka.

Egy fontos összefüggés egy L rács és az $\hat{L}$ duálisa között a következő:

$$1 \leq 2\mu(\hat{L})\lambda_1(L) \leq n, \tag{1.1}$$

Az ilyen összefüggéseket átviteli tételeknek hívják (transference theorems). Az itt leírt átviteli tétel egy alkalmazását bemutatom a következő alfejezetben.

1.5. Felsorolós algoritmusok, és merőlegesség mérése

Az első felsorolós algoritmust Kannan kezdte el [12], és azóta ezen algoritmusok számos javításon vannak túl [11], [15]. Ezektől valamiben eltér egy a duális rács redukcióján alapuló felsorolós algoritmus [7], [19]. Ezen fejezetben összekötöm a felsorolós algoritmusok futásidejét kettő a rácsok bázisának geometriáját leíró konstanssal, és leírok egy lehetséges alkalmazást.

Az egyszerűség kedvéért a továbbiakban tegyük fel, hogy a rácsunk teljes dimenziós, azaz $n = d$.

Legyen adott egy L rács, és annak egy $\mathbf{b}_1, \ldots, \mathbf{b}_n$ bázisa. Legyen $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ a bázis Gram-Schmidt ortogonalizáltja.

Először vizsgáljuk SVP megoldását L-ben. Legyen $\mathbf{b} \in L$ amelyre $\|\mathbf{b}\| = \lambda_1(L)$. Tegyük fel, hogy $\mathbf{b} = x_1\mathbf{b}_1^* + \ldots + x_n\mathbf{b}_n^*$, és adott egy $\|x_i\mathbf{b}_i^*\| \leq r_i$ felső becslés. Ekkor a legrövidebb nem-$\mathbf{0}$ vektor megtalálásához elég átvizsgálni a

$$\{x_1\mathbf{b}_1^* + \ldots + x_n\mathbf{b}_n^* \mid |x_i| \leq r_i/\|\mathbf{b}_i^*\|\}$$

halmazban található rácspontokat.

Most vizsgáljuk a CVP megoldását L-ben egy $\mathbf{t}$ célvektorral. Legyen $\mathbf{b} \in L$ olyan, hogy $\|\mathbf{t}-\mathbf{b}\|$ minimális. Tegyük fel, hogy $\mathbf{t} = t_1\mathbf{b}_1^* + \ldots + t_n\mathbf{b}_n^*$ és $\mathbf{b} = x_1\mathbf{b}_1 + \ldots + x_n\mathbf{b}_n$, és legyen adott egy $|x_i - t_i| \cdot \|\mathbf{b}_i^*\| \leq r_i$ felső becslés. Ekkor a $\mathbf{t}$ vektorhoz legközelebb eső rácsvektort megtaláljuk a

$$\{x_1\mathbf{b}_1^* + \ldots + x_n\mathbf{b}_n^* \mid |x_i - t_i| \leq r_i/\|\mathbf{b}_i^*\|\}$$

halmazban található rácspontokat átvizsgálva.

Az ilyen rácspontok száma egyik esetben sem több mint

$$\prod_{i=1}^{n}\left(\left\lfloor \frac{2r_i}{\|\mathbf{b}_i^*\|}\right\rfloor + 1\right).$$

Egy L rács egy $\mathbf{b}_1, \ldots, \mathbf{b}_n$ bázisának *merőlegességi* defektusa a

$$\delta(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \frac{\|\mathbf{b}_1\|}{\|\mathbf{b}_1^*\|} \cdot \ldots \cdot \frac{\|\mathbf{b}_n\|}{\|\mathbf{b}_n^*\|}$$

konstans, ahol $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ a bázis Gram-Schmidt ortogonalizáltja. Ez a konstans leírja, hogy a bázis vektorai mennyire térnek el a merőlegestől. Egy ehhez hasonló konstans a következő:

$$\mu(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \frac{2\mu(L_1)}{\|\mathbf{b}_1^*\|} \cdot \frac{2\mu(L_2)}{\|\mathbf{b}_2^*\|} \cdot \ldots \cdot \frac{2\mu(L_n)}{\|\mathbf{b}_n^*\|} = 2^n \cdot \frac{\mu(L_1) \cdot \ldots \cdot \mu(L_n)}{\det L}$$

ahol $L_i = L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_i)$. Nem nehéz meggondolni, hogy SVP esetén $r_i = \|\mathbf{b}_i\|$ CVP esetén pedig $r_i = \mu(L_i)$ megfelelő felső becslések. Ezt felhasználva SVP esetén a

$$3^n\delta(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

CVP esetén pedig a

$$2^n\mu(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

felső becsléseket kapjuk a felsorolandó vektorok számára. Adódik a kérdés, hogy mennyire jó bázisokat tudunk találni? Ha $\mathbf{b}_1, \ldots, \mathbf{b}_n$ HKZ-redukált bázis akkor

$$\delta(\mathbf{b}_1, \ldots, \mathbf{b}_n) \leq n^{n/2}$$

ha pedig $\mathbf{b}_1, \ldots, \mathbf{b}_n$-hez tartozó $\mathbf{d}_1, \ldots, \mathbf{d}_n$ duális bázis HKZ redukált (olyan értelemben, hogy a vektorokat fordított sorrendben vesszük) akkor 1.1 és a HKZ redukció definíciója alapján

$$\frac{2\mu(L_i)}{\|\mathbf{b}_i^*\|} = 2\mu(L_i)\lambda_1(\hat{L}_i) \leq i.$$

és ezek szerint

$$\mu(\mathbf{b}_1, \ldots, \mathbf{b}_n) \leq n!.$$

Itt fontos megjegyezni, hogy a HKZ redukció magába foglalja a legrövidebb vektorok találását, így ilyen redukált bázis találása nehezebb mint SVP megoldása. Ezt a **??** a branch and bound módszert használva egy rekurzív algoritmussal küszöbölik ki. A HKZ redukáltságnál egy valamivel gyengébb redukciót alkalmaznak. Itt még sok apró technikai részlet van, viszont végül a kapott algoritmusok futásideje nagyságrendileg az itt leírt konstansokat adja vissza.

Ha a CVP megoldásához a duális redukció során az SVP valamilyen approximációs változatát használjuk, akkor még mindig jó minőségű bázist kapunk ilyen értelemben. Az SVP approximációs változatát SVP_γ módon jelöljük ahol γ: $\mathbb{N} \to \mathbb{N}$. Itt a feladat egy olyan $\mathbf{b} \in L \setminus \{\mathbf{0}\}$ vektor találása, ahol $\|\mathbf{b}\| \leq \gamma(n) \cdot \lambda_1(L)$. Ekkor a következő felső becslést kapjuk:

$$\mu(\mathbf{b}_1, \ldots, \mathbf{b}_n) \leq n! \cdot \gamma(n) \cdot \gamma(n-1) \cdot \ldots \cdot \gamma(1).$$

Ez sok lehetőséget nyit fel: Az előfeldolgozást jelentősen gyorsabban elvégezhetjük, cserébe végül valamivel lassabban végezhetjük el a felsorolást. Érdekes lenne kísérleteket végezni ilyen szempontból, hogy lássuk a gyakorlatban mennyire működik az így kapott algoritmus.

1.6. További algoritmusok

Az előző fejezetben leírt felsorolós algoritmusok $n^{O(n)}$ időben és polinomiális tárban működnek. Ennél sokkal jobb futásidő elérhető, ha az algoritmus exponenciális tárat használ. Erre első példa [3], majd ezt sok javítás követi [4], [18]. Ezen algoritmusok véletlent használnak, viszont véletlen használata nélkül is adható algoritmus egy rács Voronoi-celláját használva [14].

Ezen témakör egy nagy átfogó nyitott kérdése a következő: Található-e algoritmus amely polinomiális tárban és $2^{O(n)}$ időben fut.

1.7. Rácsok a kriptográfiában

A kriptográfiában fontos, hogy olyan algoritmikus problémákat találjunk, amelyek nem csak bizonyos esetekben, hanem átlagos esetben is nehezen megoldhatóak. Ilyen problémákra lehet alapozni adott kriptográfiai sémák biztonságát. Jelenleg az összes ilyen kriptográfiai rendszer rácsokon adott algoritmikus problémák nehézségén alapszik. Ezen fejezetben leírom ezeket az algoritmikus problémákat [16] alapján.

gapSVP_γ: Legyen L egy olyan rács, hogy vagy $\lambda_1(L) \leq 1$ vagy $\lambda_1(L) > \gamma(n)$. Legyen adott az L rács egy B bázisa. Döntsük el, hogy melyik esetben vagyunk.

SIVP_γ (Shortest Independent Vectors Problem): Legyen adott egy n dimenziós L rács egy B bázisa. Kerressünk olyan $\mathbf{s}_1, \ldots, \mathbf{s}_n$ lineárisan független vektorokat, amelyekre $\|\mathbf{s}_i\| \leq \gamma(n)\lambda_i(L)$.

Itt $\lambda_i(L)$ az L rács i. szukcesszív minimumát jelöli:

$$\lambda_i(L) = \inf\{r \mid \exists \mathbf{s}_1, \ldots, \mathbf{s}_i \in L \text{ lineárisan független vektorok, hogy } \|\mathbf{s}_1\| \leq r, \ldots, \|\mathbf{s}_i\| \leq r\}$$

A modern kriptográfia a *Short Integer Solution* (SIS) és *Learning With Error* (LWE) problémákon alapszik. A SIS probléma megoldása a rendes értelemben, míg LWE kvantumosan visszavezethető az SIVP_γ és gapSVP_γ problémákra megfelelő γ választással. A SIS és LWE problémák jelentősége abban rejlik, hogy ezek egy-egy véletlenszerűen választott problémacsaládra

vannak definiálva, viszont az SIVP_γ és gapSVP_γ problémák pedig a klasszikus bonyulultságelméleti értelemben nehezek. Így kapunk problémákat amik megoldása az átlagos esetben olyan nehéz mint a bemutatott rácsproblémák a legrosszabb esetben.

1.8. Rács izomorfizmus probléma

Egy jelenleg felkapott algoritmikus probléma az úgynevezett LIP (Lattice Isomorphism Problem) [10]. Itt a feladat a következő: Legyen adott két izomorf rács $L, L' \subset \mathbb{Q}^n$ és ezeknek egy-egy bázisa $B, B' \in \mathbb{Q}^{n\times n}$. Találjunk olyan $O \in O_n(\mathbb{Q})$ ortogonális és $U \in GL_n(\mathbb{Z})$ unimoduláris mátrixot, hogy $B = UB'O$.

Ez a probléma még abban az esetben is nehéznek bizonyul, ha $L = \mathbb{Z}^n$. A félév során ezt az esetet vizsgáltam meg közelebbről. Itt szemléletesen jól látszik, hogy mi adja a probléma nehézségét. Mivel itt $\mathbb{Z}^n$-ről van szó, így feltehetjük, hogy B a szokásos bázis (Mindegy, hogy $\mathbb{Z}^n$ melyik bázisa van megadva, a szokásos bázist akkor is ismerjük). Most $L' = O\mathbb{Z}^n$, viszont B' nem feltétlen a szokásos bázis. A nehézség onnan adódik, hogy mivel nem ismerjük az O transzformációt, így nem látjuk a szokásos bázist. Ha látnánk, akkor rögtön meg is tudnánk határozni az (egyik) O transzformációt, ami L-et L'-be viszi. Számunkra a B' bázis lényegében ugyanúgy néz ki, mintha egy tetszőleges csúnya rács egy nem túl merőleges bázisára néznénk rá.

Jelenlegi tudásunk szerint itt az a legjobb megoldás, hogy megtaláljuk L'-ben a szokásos bázist (Azaz azt az U transzformációt, ami a B' bázist a B bázisba viszi), és akkor meg tudjuk határozni az O transzformációt.

Egy ilyen bázis megtalálásának nehézsége $O\mathbb{Z}^n$-ben nyitott kérdés. A [6], [9] cikkekben bemutatott módszerek egy-egy exponenciális algoritmust adnak. Itt gondolkodtam valamennyit a [9] cikkben bemutatott algoritmus egy általánosításán, amit a blokkok számának növelésével kapunk, viszont ez az általánosítás egyelőre nem tűnik bizonyíthatóan működőképesnek.

Hivatkozások

[1] The dual lattice. `https://cseweb.ucsd.edu/classes/sp14/cse206A-a/lec3.pdf`. Accessed: 2025-05-17.

[2] Dual lattices. `https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/Du` Accessed: 2025-05-17.

[3] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 601–610, New York, NY, USA, 2001. Association for Computing Machinery.

[4] M. Ajtai, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 53–57, 2002.

[5] H. Bennett. The complexity of the shortest vector problem. *SIGACT News*, 54(1):37–61, Mar. 2023.

[6] H. Bennett, A. Ganju, P. Peetathawatchai, and N. Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? algorithms and cryptography with the simplest lattice. Cryptology ePrint Archive, Paper 2021/1548, 2021.

[7] J. Blömer. Closest vectors, successive minima, and dual hkz-bases of lattices. In U. Montanari, J. D. P. Rolim, and E. Welzl, editors, *Automata, Languages and Programming*, pages 248–259, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[8] J. Conway and N. Sloane. Low-dimensional lattices vi: Voronoi reduction of three-dimensional lattices. *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, 436, 03 2001.

[9] L. Ducas. Provable lattice reduction of $\mathcal{F}^n$ with blocksize $n/2$. Cryptology ePrint Archive, Paper 2023/447, 2023.

[10] L. Ducas and W. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. Cryptology ePrint Archive, Paper 2021/1332, 2021.

[11] G. Hanrot and D. Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In A. Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 170–186, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[12] R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, page 193–206, New York, NY, USA, 1983. Association for Computing Machinery.

[13] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[14] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013.

[15] D. Micciancio and M. Walter. Fast lattice point enumeration with minimal overhead. Cryptology ePrint Archive, Paper 2014/569, 2014.

[16] C. Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Paper 2015/939, 2015.

[17] C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987.

[18] P. Voulgaris and D. Micciancio. Faster exponential time algorithms for the shortest vector problem. In *ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[19] U. Wagner and G. Maze. Improvements in closest point search based on dual hkz-bases. *Theoretical Computer Science*, 520:62–74, 2014.