On the Lattice Isomorphism Problem

Ishay Haviv* Oded Regev†

Abstract

We study the *Lattice Isomorphism Problem* (LIP), in which given two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ the goal is to decide whether there exists an orthogonal linear transformation mapping $\mathcal{L}_1$ to $\mathcal{L}_2$. Our main result is an algorithm for this problem running in time $n^{O(n)}$ times a polynomial in the input size, where n is the rank of the input lattices. A crucial component is a new generalized *isolation lemma*, which can isolate n linearly independent vectors in a given subset of $\mathbb{Z}^n$ and might be useful elsewhere. We also prove that LIP lies in the complexity class SZK.

1 Introduction

An m-dimensional *lattice* $\mathcal{L}$ of rank n is defined as the set of all integer combinations of n linearly independent vectors $b_1, \ldots, b_n \in \mathbb{R}^m$, which form a *basis* of the lattice. This mathematical object, despite its simplicity, hides a rich geometrical structure, which was extensively studied in the last decades by the theoretical computer science community. This was initiated by the discovery of the famous LLL algorithm in 1982 [27] and was further motivated by Ajtai's cryptographic application of lattices in 1996 [3]. To date, lattices have numerous applications in several areas of computer science including algorithms, computational complexity and cryptography.

One of the most fundamental lattice problems is the Shortest Vector Problem (SVP), where given a lattice basis the goal is to find a shortest nonzero vector in the lattice. This problem is known to be NP-hard (under randomized reductions) for approximation factors which are almost polynomial in the lattice rank n [23, 20, 30] and to be solved in its exact version by algorithms of running time exponential in n [22, 32]. However, SVP with approximation factors of $\sqrt{n/\log n}$ and $\sqrt{n}$ is known to be in coAM and in coNP respectively [18, 1], hence is not NP-hard for these factors unless the polynomial time hierarchy collapses. A major challenge in the area is to understand how hard SVP and related lattice problems are for polynomial approximation factors, as this is what lattice-based cryptography relies on.

This paper is concerned with the Lattice Isomorphism Problem (LIP). Two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are *isomorphic* if there exists an orthogonal linear transformation mapping $\mathcal{L}_1$ to $\mathcal{L}_2$. In LIP one wishes to decide whether two given lattices are isomorphic or not. The problem was studied by Plesken and Souvignier [38] (using ideas from the earlier work [37]) who suggested algorithms

*School of Computer Science, The Academic College of Tel Aviv-Yaffo, Tel Aviv 61083, Israel.

†Courant Institute of Mathematical Sciences, New York University. This material is based upon work supported by the National Science Foundation under Grant No. CCF-1320188. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

that can solve the problem in low dimensions for specific lattices of interest. The asymptotic complexity of the problem was later considered by Dutour Sikirić, Schürmann, and Vallentin [15], and it also showed up in cryptographic applications of lattices [41]. Recently, Lenstra, Schoof, and Silverberg presented an efficient algorithm that can decide if a given lattice is isomorphic to $\mathbb{Z}^n$, assuming some information about its symmetries is provided as a hint [28].

Deciding whether two given combinatorial or algebraic structures are isomorphic is a notorious question in the theory of computing. A well-known special case of this problem is the Graph Isomorphism Problem (GIP), in which given two graphs G_1 and G_2 one has to decide whether there exists an edge-preserving bijection from the vertex set of G_1 to that of G_2. The best known worst-case running time of an algorithm for GIP is $2^{\widetilde{O}(\sqrt{n})}$, where n stands for the number of vertices [7]. It was shown in [19] that GIP lies in the complexity class coAM. This implies that, unless the polynomial time hierarchy collapses, GIP is not NP-hard, and it is a long-standing open question whether there exists a polynomial time algorithm solving it (see, e.g., [5]). Interestingly, it was shown in [15] that the isomorphism problem on lattices is at least as hard as that on graphs.

Another isomorphism problem of interest is the Code Equivalence Problem, in which given two n-dimensional linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ over some field $\mathbb{F}$ the goal is to decide whether there exists a permutation on the coordinates mapping $\mathcal{C}_1$ to $\mathcal{C}_2$. This problem was studied by Petrank and Roth [36], who showed that it lies in coAM and is at least as hard as GIP. Recently, Babai showed an algorithm solving it in time $(2 + o(1))^n$ (see [6, Appendix 7.1]).

1.1 Our Results

Our main result is an algorithm that given two lattices computes *all* orthogonal linear transformations mapping one lattice to another and, in particular, decides LIP.

Theorem 1.1. *There exists an algorithm that given two bases of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n, outputs all orthogonal linear transformations $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ for which $\mathcal{L}_2 = O(\mathcal{L}_1)$ in running time $n^{O(n)} \cdot s^{O(1)}$ and in polynomial space, where s denotes the input size. In addition, the number of these transformations is at most $n^{O(n)}$.*

We note that the bound in Theorem 1.1 on the number of orthogonal linear transformations mapping one lattice to another is tight up to the constant in the exponent. To see this, observe that the isomorphisms from the lattice $\mathbb{Z}^n$ to itself are precisely all the $2^n \cdot n! = n^{\Omega(n)}$ sign permutations. This implies that the running time of the algorithm from Theorem 1.1 is optimal, up to the constant in the exponent, given that it outputs all isomorphisms between the two input lattices. However, the challenge of finding a more efficient algorithm which only decides LIP is left open.

The algorithm from Theorem 1.1 is crucially based on a new version of the celebrated *isolation lemma* of Valiant and Vazirani [43]. A standard version of the lemma says that for every set $C \subseteq \mathbb{Z}^n$ of short vectors (in ℓ_∞ norm), most short integer vectors z have a *single* vector in C that minimizes the inner product with z over all vectors in C. The isolation lemma has appeared in the literature in several variations for various applications, ranging from the design of randomized algorithms, e.g., [33, 34, 12, 24], to results in computational complexity, e.g., [42, 44, 40, 4] (for a survey see [21]). Whereas the lemma is usually used to isolate one vector, for our application we need to isolate n linearly independent vectors in C. The lemma below guarantees the existence of a vector z and

a sequence of n linearly independent vectors in C, each of which uniquely minimizes the inner product with z over all vectors in C which are not in the linear span of the previous ones.

Lemma 1.2. *Let $C \subseteq \mathbb{Z}^n$ be a set of vectors satisfying $\|c\|_\infty \leq K$ for every $c \in C$ and $\mathrm{span}(C) = \mathbb{R}^n$. Let $z = (z_1, \ldots, z_n)$ be a random vector such that each z_i is independently chosen from the uniform distribution over $\{1, \ldots, R\}$ for $R = K(2K+1)n^3/\varepsilon$. Then, with probability at least $1 - \varepsilon$, there are n linearly independent vectors $x_1, \ldots, x_n \in C$ such that for every $1 \leq j \leq n$, the minimum inner product of z with vectors in $C \setminus \mathrm{span}(x_1, \ldots, x_{j-1})$ is uniquely achieved by x_j.*

We actually prove this in a more general setting, in which span can be replaced by any function satisfying some condition. This more general statement includes as special cases some of the previously known variations of the isolation lemma, and might be useful elsewhere. See Section 3 for details.

Finally, we prove that LIP, which naturally lies in NP, has a statistical zero-knowledge proof system and hence belongs to the complexity class SZK. This result was independently observed by Greg Kuperberg [25].

Theorem 1.3. LIP *is in* SZK.

It is well known that $\mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [16, 2]. As a result, just like many other lattice problems (e.g., the problem of approximating the length of a shortest nonzero vector to within polynomial factors, which is central in lattice-based cryptography), LIP is unlikely to be NP-hard. We note, though, that the reduction from the Graph Isomorphism Problem (GIP) [15] gives some evidence that LIP is a hard problem, evidence that is lacking for other lattice problems.

1.2 Overview of Proofs and Techniques

1.2.1 The Algorithm for LIP

The input of LIP consists of two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n, and the goal is to decide if there exists an orthogonal linear transformation O satisfying $\mathcal{L}_2 = O(\mathcal{L}_1)$. In order to find such an O it suffices to find n linearly independent vectors in $\mathcal{L}_1$ and their image in $\mathcal{L}_2$ according to O. Since O preserves lengths, a possible approach is to compute n linearly independent short vectors of $\mathcal{L}_1$ and try to map them to all n-tuples of short vectors of $\mathcal{L}_2$.

Consider the case where the lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ have only one shortest nonzero vector (up to sign). In this case, there are only two possible choices for how an isomorphism from $\mathcal{L}_1$ to $\mathcal{L}_2$ can act on these vectors. Hence, one can recursively solve the problem by considering the lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ projected to the spaces orthogonal to their shortest vectors. This demonstrates that the hard instances of the problem are those where the lattices have n linearly independent shortest vectors, so in the rest of this discussion let us assume that we are in this case.

Given the lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ it is possible to compute the sets A_1 and A_2 of all shortest nonzero vectors in $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Indeed, by the algorithm of [32], the running time needed to compute A_1 and A_2 is $2^{O(n)}$ (or $n^{O(n)}$, if we insist on polynomial space complexity [22]). Now, consider the algorithm that for certain n linearly independent vectors in A_1 tries all the linear transformations that map them to n linearly independent vectors in A_2 and checks if at least one of them is orthogonal and maps $\mathcal{L}_1$ to $\mathcal{L}_2$. Notice that the running time of this algorithm crucially

depends on the number of shortest nonzero vectors in the lattices, which is usually referred to as their *kissing number*. It is easy to see that the kissing number of a lattice of rank n is at most 2^{n+1}.[1] This implies that the suggested algorithm has running time whose dependence on n is bounded by $2^{O(n^2)}$. However, the true worst-case running time of this algorithm is a function of the maximum possible kissing number of a lattice of rank n, whose value is an open question. The best currently known lower bound is $n^{\Omega(\log n)}$ [9] (see also [14, Page 151]), hence even if this lower bound were tight (which does not seem particularly likely), the algorithm would run in time $n^{\Omega(n \log n)}$, which is still asymptotically slower than our algorithm.

We improve on the above naive algorithm by showing a way to isolate n linearly independent vectors in the sets A_1 and A_2. In Theorem 4.2 we prove, using our isolation lemma (Lemma 1.2), that for a lattice $\mathcal{L}_1$ as above there exists a relatively short vector v in the dual lattice $\mathcal{L}_1^*$ that *uniquely* defines n linearly independent vectors $x_1, \ldots, x_n$ in A_1. These vectors are defined as follows: for every $1 \leq j \leq n$, the minimum inner product of v with vectors in $A_1 \setminus \text{span}(x_1, \ldots, x_{j-1})$ is uniquely achieved by x_j. Given such a v (which can be found by enumerating all short vectors in $\mathcal{L}_1^*$), we try all vectors of norm $\|v\|$ in $\mathcal{L}_2^*$, of which there are at most $n^{O(n)}$. Once we find the image of v under O, we use it to apply the same process as above with A_2 obtaining n linearly independent vectors in A_2. Since O preserves inner products, these vectors must be the images of $x_1, \ldots, x_n$ under O, which allows us to find O.

1.2.2 LIP is in SZK

We turn to discuss the proof of Theorem 1.3 which says that LIP lies in the complexity class SZK. Since SZK is known to be closed under complement [35], it suffices to show a statistical zero-knowledge proof system that enables an efficient verifier to verify that two given lattices are *not* isomorphic. The high level idea is similar to known proof systems of the complement of other isomorphism problems, e.g., Graph Isomorphism and Code Equivalence [43, 36]. In these proof systems the verifier picks uniformly at random one of the two objects given as input and sends to the prover a random representative of its isomorphism class. The verifier accepts if and only if the prover identifies which of the two objects was chosen. A crucial observation is that if the two objects are isomorphic then the prover gets a sample which is independent of the chosen object, hence she is not able to identify it with probability higher than $1/2$. On the other hand, if the objects are not isomorphic, then the object sent by the verifier is isomorphic to exactly one of the two, hence a computationally unbounded prover is able to answer correctly. Moreover, the correct answer is known in advance to the verifier, who therefore does not learn anything new from the prover's answer.

In the lattice analogue of the above proof system, in addition to choosing a lattice that forms a representative of the isomorphism class, the verifier also has to choose a basis that generates the lattice. Observe that the basis should be chosen in a way that does not provide any useful information for the prover, and in particular, must not depend on the input bases. To deal with this difficulty we use known efficient algorithms to sample lattice vectors from the discrete Gaussian distribution [17] (see also [10]), and prove that polynomially many samples suffice to obtain a generating set for the lattice (Lemma 5.4). We can then send a random rotation of this set of

[1] Indeed, if there are more than 2^{n+1} shortest nonzero lattice vectors then at least two of them belong to the same coset of $2\mathcal{L}$ and have nonzero average, hence their average is a shorter lattice vector, in contradiction.

vectors. In fact, to avoid issues of accuracy, we instead send the matrix of all pairwise inner products of these vectors (the Gram matrix).

1.3 Outline

The rest of the paper is organized as follows. In Section 2 we gather basic definitions and results we shall later use. In Section 3 we prove our new generalized isolation lemma and derive Lemma 1.2. In Section 4 we present our algorithm for LIP and prove Theorem 1.1. This is done in two steps, where in the first we assume that the input lattices contain n linearly independent shortest vectors (i.e., $\lambda_1 = \lambda_n$), and in the second we extend the algorithm to the general case. Finally, in Section 5 we prove Theorem 1.3.

2 Preliminaries

2.1 General

An *orthogonal* linear transformation (or *isometry*) $O : V_1 \to V_2$ is a linear transformation that preserves inner products, that is, $\langle x, y\rangle = \langle O(x), O(y)\rangle$ for every $x, y \in V_1$. For a set $A \subseteq V_1$ we use the notation $O(A) = \{O(x) \mid x \in A\}$. For a matrix B we denote its ith column by b_i, and $O(B)$ stands for the matrix whose ith column is $O(b_i)$. The *Gram matrix* of a matrix B is defined to be the matrix $G = B^T \cdot B$, or equivalently, $G_{i,j} = \langle b_i, b_j\rangle$ for every i and j. The Gram matrix of a matrix B determines its columns up to an orthogonal linear transformation, as stated below. Note that $\mathrm{span}(B)$ stands for the subspace spanned by the columns of B.

Fact 2.1. *Let B and D be two matrices satisfying $B^T \cdot B = D^T \cdot D$. Then there exists an orthogonal linear transformation $O : \mathrm{span}(B) \to \mathrm{span}(D)$ for which $D = O(B)$.*

2.2 Lattices

An m-dimensional *lattice* $\mathcal{L} \subseteq \mathbb{R}^m$ is the set of all integer combinations of a set of linearly independent vectors $\{b_1, \ldots, b_n\} \subseteq \mathbb{R}^m$, i.e., $\mathcal{L} = \{\sum_{i=1}^n a_i b_i \mid \forall i.\ a_i \in \mathbb{Z}\}$. The set $\{b_1, \ldots, b_n\}$ is called a *basis* of $\mathcal{L}$ and n, the number of vectors in it, is the *rank* of $\mathcal{L}$. Let B be the m by n matrix whose ith column is b_i. We identify the matrix and the basis that it represents and denote by $\mathcal{L}(B)$ the lattice that B generates. The norm of a basis B is defined by $\|B\| = \max_i \|b_i\|$. A basis of a lattice is not unique. It is well known that two bases B_1 and B_2 generate the same lattice of rank n if and only if $B_1 = B_2 \cdot U$ for a *unimodular* matrix $U \in \mathbb{Z}^{n\times n}$, i.e., an integer matrix satisfying $|\det(U)| = 1$. The determinant of a lattice $\mathcal{L}$ is defined by $\det(\mathcal{L}) = \sqrt{\det(B^T B)}$, where B is a basis that generates $\mathcal{L}$. It is not difficult to verify that $\det(\mathcal{L})$ is independent of the choice of the basis. A set of (not necessarily linearly independent) vectors that generate a lattice is called a *generating set* of the lattice. A lattice $\mathcal{M}$ is a *sublattice* of a lattice $\mathcal{L}$ if $\mathcal{M} \subseteq \mathcal{L}$, and it is a *strict sublattice* if $\mathcal{M} \subsetneq \mathcal{L}$. If a lattice $\mathcal{L}$ and its sublattice $\mathcal{M}$ span the same subspace, then the *index* of $\mathcal{M}$ in $\mathcal{L}$ is defined by $|\mathcal{L} : \mathcal{M}| = \det(\mathcal{M})/\det(\mathcal{L})$. It is easy to see that if $\mathcal{M}$ is a sublattice of $\mathcal{L}$ such that $|\mathcal{L} : \mathcal{M}| = 1$ then $\mathcal{M} = \mathcal{L}$.

The length of a shortest nonzero vector in $\mathcal{L}$ is denoted by $\lambda_1(\mathcal{L}) = \min\{\|u\| \mid u \in \mathcal{L} \setminus \{0\}\}$. The following simple and standard fact provides an upper bound on the number of short vectors in a lattice of rank n (see, e.g., [32]).

Fact 2.2. *For every lattice $\mathcal{L}$ of rank n and for every $t \geq 0$, the number of vectors in $\mathcal{L}$ of norm at most $t \cdot \lambda_1(\mathcal{L})$ is at most $(2t+1)^n$.*

Proof. Consider all the (open) balls of radius $\lambda_1(\mathcal{L})/2$ centered at the lattice points of distance at most $t \cdot \lambda_1(\mathcal{L})$ from the origin. These balls are pairwise disjoint and are all contained in the ball centered at the origin whose radius is $(t + 1/2) \cdot \lambda_1(\mathcal{L})$. This implies that their number is at most

$$\Big(\frac{(t+1/2)\cdot\lambda_1(\mathcal{L})}{\lambda_1(\mathcal{L})/2}\Big)^n = (2t+1)^n. \qquad \square$$

The definition of λ_1 is naturally extended to the *successive minima* $\lambda_1, \ldots, \lambda_n$ defined as follows:

$$\lambda_i(\mathcal{L}) = \inf\{r > 0 \mid \text{rank}(\text{span}(\mathcal{L} \cap (r \cdot \mathcal{B}))) \geq i\},$$

where $\mathcal{B}$ denotes the ball of radius 1 centered at the origin. A somewhat related lattice parameter, denoted $bl(\mathcal{L})$, is defined as the minimum norm of a basis that generates $\mathcal{L}$. It is known that bl is related to the nth successive minimum by $\lambda_n(\mathcal{L}) \leq bl(\mathcal{L}) \leq \frac{\sqrt{n}}{2} \cdot \lambda_n(\mathcal{L})$ (see, e.g., [11]).

As mentioned before, two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are *isomorphic* if there exists an orthogonal linear transformation $O : \text{span}(\mathcal{L}_1) \to \text{span}(\mathcal{L}_2)$ for which $\mathcal{L}_2 = O(\mathcal{L}_1)$. In this paper we study the computational problem, called the *Lattice Isomorphism Problem* (LIP), defined as follows. The input consists of two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ and we are asked to decide if they are isomorphic or not. One subtle issue is how to specify the input to the problem. One obvious way is to follow what is commonly done with other lattice problems, namely, the lattices are given as a set of basis vectors whose entries are given as rational numbers. This however leads to what we feel is an unnecessarily restricted definition: orthogonal matrices typically involve irrational entries, hence bases of two isomorphic lattices will typically also include irrational entries. Such bases, however, cannot be specified exactly as an input. Instead, we follow a much more natural definition (which is clearly as hard as the previous one, making our results stronger) in which the input bases are specified in terms of their *Gram matrices*. Notice that a Gram matrix specifies a basis only up to rotation, but this is clearly inconsequential for LIP.

Definition 2.3. *In the Lattice Isomorphism Problem (LIP) the input consists of two Gram matrices G_1 and G_2, and the goal is to decide if there exists a unimodular matrix U for which $G_1 = U^T \cdot G_2 \cdot U$.*

For clarity, in our algorithms we assume that the input is given as a basis, and we ignore issues of precision. This is justified because (1) an arbitrarily good approximation of a basis can be extracted from a Gram matrix using the Cholesky decomposition; and (2) given a good enough approximation of a purported orthogonal transformation it is possible to check if it corresponds to a true lattice isomorphism by extracting the corresponding (integer-valued) unimodular matrix U that converts between the bases and checking the equality $G_1 = U^T G_2 U$, which only involves exact arithmetic. We note that an alternative, possibly more disciplined, solution is to avoid working with lattice vectors directly and instead work with their integer coefficients in terms of a lattice basis, and use the Gram matrix to compute norms and inner products (see, e.g., [13, Page 80]).

2.3 Dual Lattices

The *dual lattice* of a lattice $\mathcal{L}$, denoted by $\mathcal{L}^*$, is defined as the set of all vectors in $\text{span}(\mathcal{L})$ that have integer inner product with all the lattice vectors of $\mathcal{L}$, that is,

$$\mathcal{L}^* = \{u \in \text{span}(\mathcal{L}) \mid \forall v \in \mathcal{L}.\ \langle u, v\rangle \in \mathbb{Z}\}.$$

The *dual basis* of a lattice basis B is denoted by B^* and is defined as the one which satisfies $B^T \cdot B^* = I$ and $\text{span}(B) = \text{span}(B^*)$, that is, $B^* = B(B^T B)^{-1}$. It is well known that the dual basis generates the dual lattice, i.e., $\mathcal{L}(B)^* = \mathcal{L}(B^*)$.

In [8] Banaszczyk proved relations between parameters of lattices and parameters of their dual. Such results are known as *transference theorems*. One of his results, which is known to be tight up to a multiplicative constant, is the following.

Theorem 2.4 ([8]). *For every lattice $\mathcal{L}$ of rank n, $1 \leq \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \leq n$.*

2.4 Korkine-Zolotarev Bases

Before defining Korkine-Zolotarev bases we need to define the *Gram-Schmidt orthogonalization process*. For a sequence of vectors $b_1, \ldots, b_n$ define the corresponding Gram-Schmidt orthogonalized vectors $\tilde{b}_1, \ldots, \tilde{b}_n$ by

$$\tilde{b}_i = b_i - \sum_{j=1}^{i-1} \mu_{i,j}\tilde{b}_j, \quad \mu_{i,j} = \frac{\langle b_i, \tilde{b}_j\rangle}{\langle \tilde{b}_j, \tilde{b}_j\rangle}.$$

In words, $\tilde{b}_i$ is the component of b_i orthogonal to $b_1, \ldots, b_{i-1}$. A Korkine-Zolotarev basis is defined as follows.

Definition 2.5. *Let B be a basis of a lattice $\mathcal{L}$ of rank n and let $\widetilde{B}$ be the corresponding Gram-Schmidt orthogonalized basis. For $1 \leq i \leq n$ define the projection function $\pi_i^{(B)}(x) = \sum_{j=i}^{n} \langle x, \tilde{b}_j\rangle \cdot \tilde{b}_j / \|\tilde{b}_j\|^2$ that maps x to its projection on* $\text{span}(\tilde{b}_i, \ldots, \tilde{b}_n)$*. A basis B is a* Korkine-Zolotarev basis *if for all $1 \leq i \leq n$,*

- *$\tilde{b}_i$ is a shortest nonzero vector in $\pi_i^{(B)}(\mathcal{L}) = \{\pi_i^{(B)}(u) \mid u \in \mathcal{L}\}$,*
- *and for all $j < i$, the Gram-Schmidt coefficients $\mu_{i,j}$ of B satisfy $|\mu_{i,j}| \leq \frac{1}{2}$.*

A basis B is a dual Korkine-Zolotarev basis *if its dual B^* is a Korkine-Zolotarev basis.*

Lagarias, Lenstra and Schnorr [26] related the norms of the vectors in a Korkine-Zolotarev basis to the successive minima of the lattice, as stated below.

Theorem 2.6 ([26]). *If B is a Korkine-Zolotarev basis of a lattice $\mathcal{L}$ of rank n, then for all $1 \leq i \leq n$,*

$$\|b_i\| \leq \sqrt{i} \cdot \lambda_i(\mathcal{L}).$$

The following lemma provides an upper bound on the coefficients of short lattice vectors in terms of a dual Korkine-Zolotarev basis.

Lemma 2.7. *Let B be a dual Korkine-Zolotarev basis of a lattice $\mathcal{L}$ of rank n and let $a_1, \ldots, a_n$ be integer coefficients of a vector $v = \sum_{i=1}^{n} a_i \cdot b_i$ of $\mathcal{L}$ satisfying $\|v\| \leq t \cdot \lambda_1(\mathcal{L})$. Then, for every $1 \leq i \leq n$, $|a_i| \leq t \cdot n^{3/2}$.*

Proof. Since the dual basis B^* satisfies $B^T \cdot B^* = I$, it follows that $a_i = \langle v, b_i^* \rangle$ for every $1 \leq i \leq n$. By the Cauchy-Schwarz inequality, we obtain that

$$|a_i| = |\langle v, b_i^* \rangle| \leq \|v\| \cdot \|b_i^*\| \leq t \cdot \lambda_1(\mathcal{L}) \cdot \sqrt{n} \cdot \lambda_n(\mathcal{L}^*) \leq t \cdot n^{3/2},$$

where the second inequality follows from Theorem 2.6, and the third one from Theorem 2.4. □

Lemma 2.8. *Let B be a dual Korkine-Zolotarev basis of a lattice $\mathcal{L}$ of rank n satisfying $\lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$, and let B^* be its dual. Then, for every integer coefficients $a_1, \ldots, a_n$ satisfying $|a_i| \leq K$ for every $1 \leq i \leq n$, the vector $v = \sum_{i=1}^{n} a_i \cdot b_i^* \in \mathcal{L}^*$ satisfies $\|v\| \leq n^{5/2} \cdot K \cdot \lambda_1(\mathcal{L}^*)$.*

Proof. First, use Theorem 2.4 twice to obtain

$$\lambda_n(\mathcal{L}^*) \leq \frac{n}{\lambda_1(\mathcal{L})} = \frac{n}{\lambda_n(\mathcal{L})} \leq n \cdot \lambda_1(\mathcal{L}^*).$$

Now, by the triangle inequality and Theorem 2.6 applied to the Korkine-Zolotarev basis B^*, it follows that

$$\|v\| \leq \sum_{i=1}^{n} |a_i| \cdot \|b_i^*\| \leq n \cdot K \cdot \sqrt{n} \cdot \lambda_n(\mathcal{L}^*) \leq n^{5/2} \cdot K \cdot \lambda_1(\mathcal{L}^*).$$ □

2.5 Gaussian Measures on Lattices

For $n \in \mathbb{N}$ and $s > 0$ let $\rho_s : \mathbb{R}^n \to (0, 1]$ be the *Gaussian function* centered at the origin scaled by a factor of s defined by

$$\forall x \in \mathbb{R}^n.\ \rho_s(x) = e^{-\pi \|x/s\|^2}.$$

We define the *discrete Gaussian distribution* with parameter s on a lattice $\mathcal{L}$ of rank n by its probability function

$$\forall x \in \mathcal{L}.\ D_{\mathcal{L},s}(x) = \frac{\rho_s(x)}{\rho_s(\mathcal{L})},$$

where for a set A we denote $\rho_s(A) = \sum_{x \in A} \rho_s(x)$. Notice that the sum $\rho_s(\mathcal{L})$ over all lattice vectors is finite, as follows from the fact that $\int_{\mathbb{R}^n} \rho_s(x) \mathrm{d}x = s^n$. It can be shown that $D_{\mathcal{L},s}$ has expectation zero and expected squared norm close to $s^2 n / 2\pi$ if s is large enough. We need the following concentration result of Banaszczyk [8].

Lemma 2.9 ([8], Lemma 1.5(i))**.** *Let $\mathcal{L}$ be a lattice of rank n, and let u be a vector chosen from $D_{\mathcal{L},s}$. Then, the probability that $\|u\| \geq s \cdot \sqrt{n}$ is $2^{-\Omega(n)}$.*

We also need the following simple claim, which follows from techniques in [8].

Claim 2.10 ([8])**.** *For every n-dimensional lattice $\mathcal{L}$, a real $s > 0$ and a vector $w \in \mathbb{R}^n$,*

$$\rho_s(w + \mathcal{L}) \geq \rho_s(w) \cdot \rho_s(\mathcal{L}).$$

Proof. The claim follows from the following calculation.

$$\begin{aligned}\rho_s(w+\mathcal{L}) &= \sum_{x\in\mathcal{L}} e^{-\pi\|w+x\|^2/s^2} = \frac{1}{2}\cdot\sum_{x\in\mathcal{L}}\left(e^{-\pi\|x+w\|^2/s^2}+e^{-\pi\|x-w\|^2/s^2}\right)\\ &= e^{-\pi\|w\|^2/s^2}\cdot\sum_{x\in\mathcal{L}}\left(e^{-\pi\|x\|^2/s^2}\cdot\cosh(2\pi\langle x,w\rangle/s^2)\right) \geq \rho_s(w)\cdot\rho_s(\mathcal{L}),\end{aligned}$$

where the inequality holds since $\cosh(\alpha)\geq 1$ for every α. □

The problem of efficient sampling from the discrete Gaussian distribution was studied by Gentry, Peikert and Vaikuntanathan [17]. They showed a sampling algorithm whose output distribution is statistically close to the discrete Gaussian distribution on a given lattice, assuming that the parameter s is sufficiently large. For convenience, we state below a recent result of Brakerski et al. [10] providing an *exact* sampling algorithm from the discrete Gaussian distribution.

Lemma 2.11 ([10], Lemma 2.3)**.** *There exists a probabilistic polynomial time algorithm* SampleD *that given a basis B of a lattice $\mathcal{L}$ of rank n and $s\geq \max_i\|\tilde{b}_i\|\cdot\sqrt{\ln(2n+4)/\pi}$ outputs a sample distributed according to $D_{\mathcal{L},s}$.*

2.6 Lattice Algorithms

The following two lemmas provide efficient algorithms for computing lattice bases. In the first, the lattice is given by a generating set, and in the second it is given as an intersection of a lattice and a subspace. Both algorithms are based on what is known as matrices of Hermite normal form (see, e.g., [31, Chapter 8]).

Lemma 2.12. *There is a polynomial time algorithm that given a set of vectors computes a basis for the lattice that they generate.*

Lemma 2.13 ([29], Lemma 1)**.** *There is a polynomial time algorithm that given a basis of an m-dimensional lattice $\mathcal{L}$ and a subspace S of $\mathbb{R}^m$ computes a basis of the lattice $\mathcal{L}\cap S$.*

The following theorem of Kannan [22] provides an algorithm for the Shortest Vector Problem with running time $n^{O(n)}$ and polynomial space complexity. We note that a faster algorithm with running time $2^{O(n)}$ was obtained by Micciancio and Voulgaris in [32], however its space complexity is exponential in n.

Theorem 2.14 ([22])**.** *There exists an algorithm that given a basis of a lattice $\mathcal{L}$ of rank n computes a shortest nonzero vector of $\mathcal{L}$ in running time $n^{O(n)}\cdot s^{O(1)}$ and in polynomial space, where s denotes the input size.*

The definition of Korkine-Zolotarev bases (Definition 2.5) immediately implies that a Korkine-Zolotarev basis generating a given lattice of rank n can be efficiently computed using n calls to an algorithm that finds a shortest nonzero vector in a lattice. This gives us the following corollary.

Corollary 2.15. *There exists an algorithm that given a basis of a lattice $\mathcal{L}$ of rank n computes a Korkine-Zolotarev basis generating $\mathcal{L}$ in running time $n^{O(n)}\cdot s^{O(1)}$ and in polynomial space, where s denotes the input size.*

Another corollary of Theorem 2.14 is the following.

Corollary 2.16. *There exists an algorithm that given a basis of a lattice $\mathcal{L}$ of rank n and a number $t \geq 1$, outputs all the lattice vectors $v \in \mathcal{L}$ satisfying $\|v\| \leq t \cdot \lambda_1(\mathcal{L})$ in running time $(t \cdot n)^{O(n)} \cdot s^{O(1)}$ and in polynomial space, where s denotes the input size.*

Proof. Given a lattice $\mathcal{L}$ of rank n it is possible to compute $\lambda_1(\mathcal{L})$ using Theorem 2.14 and a dual Korkine-Zolotarev basis B generating $\mathcal{L}$ using Corollary 2.15. Now, consider the algorithm that goes over all the linear integer combinations of the vectors in B with all coefficients of absolute value at most $t \cdot n^{3/2}$ and outputs the ones that have norm at most $t \cdot \lambda_1(\mathcal{L})$. The correctness of the algorithm follows from Lemma 2.7.

By Theorem 2.14 and Corollary 2.15, the space complexity needed to compute $\lambda_1(\mathcal{L})$ and B is polynomial in the input size s, and the running time is $n^{O(n)} \cdot s^{O(1)}$. The number of iterations in the algorithm above is $(2t \cdot n^{3/2} + 1)^n = (t \cdot n)^{O(n)}$. It follows that the algorithm has space complexity polynomial in s and running time $(t \cdot n)^{O(n)} \cdot s^{O(1)}$, as required. □

3 A Generalized Isolation Lemma

In this section we prove a new generalized version of the isolation lemma of [43]. The situation under study is the following. Let C be a set of vectors in $\mathbb{Z}^n$ with bounded entries, and let $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ be some function from the power set of $\mathbb{Z}^n$ to itself, which we refer to as an *elimination function*. It might be useful to think of E as the linear span function restricted to $\mathbb{Z}^n$, as for this function we will obtain Lemma 1.2.

Our goal is to show that a random integer n-dimensional vector z with bounded entries with high probability *uniquely* defines a sequence of vectors $x_1, \ldots, x_d$ in C as follows. The vector x_1 is the unique vector in $C \setminus E(\emptyset)$ that achieves the minimum inner product of z with vectors in $C \setminus E(\emptyset)$. Once x_1 is chosen, it cannot be chosen anymore, and, moreover, a certain subset of C, denoted $E(\{x_1\})$, is eliminated from C so that its elements cannot be chosen in the next steps. Similarly, x_2 is the unique vector in $C \setminus E(\{x_1\})$ that achieves the minimum inner product of z with vectors in $C \setminus E(\{x_1\})$, and, as before, the elements in the set $E(\{x_1, x_2\})$ cannot be chosen from now on. This process proceeds until we obtain d vectors $x_1, \ldots, x_d$ which eliminate the whole C, that is, $C \subseteq E(\{x_1, \ldots, x_d\})$, and satisfy $x_j \in C \setminus E(\{x_1, \ldots, x_{j-1}\})$ for every $1 \leq j \leq d$.

The above process is a generalization of several known cases of the isolation lemma. For example, if the function E is defined to output the empty set on itself and $\mathbb{Z}^n$ on every other set, then the process above will give us a single vector $x_1 \in C$ that uniquely minimizes the inner product with z, just like the standard isolation lemma. As another example, consider the function E which is defined to act like the identity function on sets of size smaller than d and to output $\mathbb{Z}^n$ on every other set. With this E we will obtain d vectors which uniquely achieve the minimum d inner products of vectors in C with z. Another example for a function E, which is the one used for Lemma 1.2, is defined by $E(A) = \text{span}(A) \cap \mathbb{Z}^n$. Using this elimination function we obtain d linearly independent vectors $x_1, \ldots, x_d$ in C, such that x_j uniquely achieves the minimum inner product of z with vectors in $C \setminus \text{span}(\{x_1, \ldots, x_{j-1}\})$ for every $1 \leq j \leq d$ where $d = \text{rank}(\text{span}(C))$.

We turn to define the type of elimination functions considered in our isolation lemma.

Definition 3.1. *For a set family $\mathcal{F} \subseteq P(\mathbb{Z}^n)$, which is closed under intersection and satisfies $\mathbb{Z}^n \in \mathcal{F}$, define its* elimination function $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ *by* $E(A) = \bigcap\{X \in \mathcal{F} \mid A \subseteq X\} \in \mathcal{F}$.

We note that all the elimination functions considered in the examples above can be defined as in Definition 3.1. For the standard isolation lemma take $\mathcal{F} = \{\emptyset, \mathbb{Z}^n\}$, for the d uniquely achieved minimum inner products take $\mathcal{F} = \{X \subseteq \mathbb{Z}^n \mid |X| < d\} \cup \{\mathbb{Z}^n\}$, and for the span elimination function take $\mathcal{F}$ to be the family of all sets $S \cap \mathbb{Z}^n$ where S is a linear subspace of $\mathbb{R}^n$. It is easy to see that all these set families are closed under intersection and include $\mathbb{Z}^n$.

Claim 3.2. *Let $\mathcal{F} \subseteq P(\mathbb{R}^n)$ be a set family as in Definition 3.1, and let $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ be its elimination function. Then, for every $A, B \in P(\mathbb{Z}^n)$,*

1. *$A \subseteq E(A)$,*
2. *$A \subseteq E(B)$ implies $E(A) \subseteq E(B)$, and*
3. *$A \subseteq B$ implies $E(A) \subseteq E(B)$.*

Proof. Item 1 is immediate from the definition of E. For Item 2, assume $A \subseteq E(B)$. By the definition of E, $E(A)$ is contained in every set of $\mathcal{F}$ which contains A, hence, in particular, it is contained in $E(B)$. For Item 3, assume $A \subseteq B$. This implies that every set of $\mathcal{F}$ which contains B contains A as well, therefore $E(A) \subseteq E(B)$. □

Remark 3.3. *It can be shown that for every function $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ which satisfies Items 1 and 2 in Claim 3.2 there exists a set family $\mathcal{F}$ which is closed under intersection and induces E as in Definition 3.1.*

The following definition will be used in the statement of our isolation lemma.

Definition 3.4. *For an elimination function $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ as in Definition 3.1 and a set $C \subseteq \mathbb{Z}^n$, a* chain *of length d in C is a sequence of d vectors $x_1, \ldots, x_d \in C$ such that $x_j \notin E(\{x_1, \ldots, x_{j-1}\})$ for every $1 \leq j \leq d$. If, in addition, $C \subseteq E(\{x_1, \ldots, x_d\})$ we say that the chain is* maximal. *We say that a vector $z \in \mathbb{Z}^n$* uniquely defines a chain *$x_1, \ldots, x_d$ in C if for every $1 \leq j \leq d$, the minimum inner product of z with vectors in $C \setminus E(\{x_1, \ldots, x_{j-1}\})$ is uniquely achieved by x_j.*

Lemma 3.5 (A Generalized Isolation Lemma)**.** *Let $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ be an elimination function as in Definition 3.1. Let $C \subseteq \mathbb{Z}^n$ be a set of vectors satisfying $\|c\|_\infty \leq K$ for every $c \in C$, such that every chain in C has length at most m. Let $z = (z_1, \ldots, z_n)$ be a random vector such that each z_i is independently chosen from the uniform distribution over $\{1, \ldots, R\}$ for $R = K(2K+1)m^2n/\varepsilon$. Then, with probability at least $1 - \varepsilon$, z uniquely defines a maximal chain in C.*

We need the following additional notations to be used in the proof.

Definition 3.6. *For a set $C \subseteq \mathbb{Z}^n$ and a vector $z \in \mathbb{Z}^n$, we let $C_z[r]$ denote the set of all vectors in C whose inner product with z is r, that is, $C_z[r] = \{c \in C \mid \langle z, c\rangle = r\}$. For an elimination function $E : P(\mathbb{Z}^n) \to P(\mathbb{Z}^n)$ as in Definition 3.1, we say that a set $C_z[r]$ is* contributing *to C if it is not contained in the set obtained by applying E to the set of vectors in C whose inner product with z is smaller than r, equivalently, $C_z[r] \nsubseteq E(\cup_{r':r'<r} C_z[r'])$.*

Proof of Lemma 3.5. For a vector $z = (z_1, \ldots, z_n)$, we say that an index $1 \leq i \leq n$ is *singular* if (1) z uniquely defines a chain $x_1, \ldots, x_j$ in C for some $j \geq 0$, but (2) there are at least two vectors in $C \setminus E(\{x_1, \ldots, x_j\})$ that differ in the ith coordinate and achieve the minimum inner product with z among the vectors in $C \setminus E(\{x_1, \ldots, x_j\})$. We prove below that for every $1 \leq i \leq n$, the probability that i is singular is at most ε/n. By the union bound, with probability at least $1 - \varepsilon$ none of the indices is singular, thus the lemma follows.

From now on fix an arbitrary index $1 \leq i \leq n$ and the values of $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n$. For every $-K \leq t \leq K$ denote

$$C^{(t)} = \{c \in C \mid c_i = t\}.$$

Partition every $C^{(t)}$ into the sets $C_z^{(t)}[r]$, and note that every $c, c' \in C^{(t)}$ are in the same set if and only if $\sum_{j:j\neq i} z_j c_j = \sum_{j:j\neq i} z_j c'_j$, independently of the value of z_i. Similarly, the order of the sets $C_z^{(t)}[r]$ in a non-decreasing value of r is independent of z_i. Finally, observe that for every $t \neq t'$ and every two sets in the partitions of $C^{(t)}$ and $C^{(t')}$, there is at most one value of z_i for which the inner products of z with the vectors in the two sets are equal.

For every t we denote by $A_z^{(t)}$ the set of all integers r for which $C_z^{(t)}[r]$ is contributing to $C^{(t)}$. Using Item 3 of Claim 3.2, one can choose one vector from every set $C_z^{(t)}[r]$ for $r \in A_z^{(t)}$ to obtain a chain of length $|A_z^{(t)}|$. Hence, our assumption on C implies that $|A_z^{(t)}| \leq m$. This gives us $2K+1$ sets $A_z^{(t)}$, each of which is of size at most m. Hence, there are at most $m^2 \cdot \binom{2K+1}{2} = m^2(2K+1)K$ possible values of z_i for which two distinct sets $A_z^{(t)}$ intersect. Since z_i is uniformly chosen from $\{1, \ldots, R\}$, the probability that two distinct sets $A_z^{(t)}$ intersect is at most $m^2(2K+1)K/R = \varepsilon/n$.

To complete the proof, it suffices to show that if i is singular for a vector z, then there exist two distinct intersecting sets $A_z^{(t)}$. Assume that i is singular for a vector z. This implies that z uniquely defines a chain $x_1, \ldots, x_j$ in C for some $j \geq 0$, but there are two vectors b, c satisfying $b_i \neq c_i$ that achieve the minimum inner product of z with vectors in $C \setminus E(\{x_1, \ldots, x_j\})$. Partition C into the sets $C_z[r]$, and let A_z be the set of all integers r for which $C_z[r]$ is contributing to C. Using Item 2 of Claim 3.2, it follows that there exists some $r \in A_z$ for which the contributing set $C_z[r]$ contains b and c which both do not belong to $E(\cup_{r':r'<r} C_z[r'])$. In particular, since $C_z^{(t)}[r] \subseteq C_z[r]$ for every t and r, Item 3 of Claim 3.2 implies that $C_z^{(b_i)}[r]$ is contributing to $C^{(b_i)}$ and that $C_z^{(c_i)}[r]$ is contributing to $C^{(c_i)}$. Hence, r belongs to both $A_z^{(b_i)}$ and $A_z^{(c_i)}$, as required. □

Now, we turn to derive the special case of the previous lemma, which is used in the next section (Lemma 1.2). To state it, we use the following definition which is analogous to Definition 3.4 for the span elimination function.

Definition 3.7. *For a set $A \subseteq \mathbb{R}^m$ and a vector $v \in \mathbb{R}^m$, we say that v* uniquely defines a linearly independent chain *of length n in A if there are n vectors $x_1, \ldots, x_n \in A$ such that for every $1 \leq j \leq n$, the minimum inner product of v with vectors in $A \setminus \mathrm{span}(x_1, \ldots, x_{j-1})$ is uniquely achieved by x_j.*

Corollary 3.8. *Let $C \subseteq \mathbb{Z}^n$ be a set of vectors satisfying $\|c\|_\infty \leq K$ for every $c \in C$ and $\mathrm{span}(C) = \mathbb{R}^n$. Let $z = (z_1, \ldots, z_n)$ be a random vector such that each z_i is independently chosen from the uniform distribution over $\{1, \ldots, R\}$ for $R = K(2K+1)n^3/\varepsilon$. Then, with probability at least $1 - \varepsilon$, z uniquely defines a linearly independent chain of length n in C.*

Proof. Consider the set family $\mathcal{F} = \{S \cap \mathbb{Z}^n \mid S \text{ is a subspace of } \mathbb{R}^n\}$. The family $\mathcal{F}$ includes $\mathbb{Z}^n$ and is closed under intersection since subspaces of $\mathbb{R}^n$ are. The elimination function E that $\mathcal{F}$ induces is defined by $E(A) = \text{span}(A) \cap \mathbb{Z}^n$. Observe that the vectors of every chain in C (with respect to this E) are linearly independent, thus its length is at most n. Apply Lemma 3.5 with $m = n$ to obtain that the random vector z, with probability $1 - \varepsilon$, uniquely defines a maximal linearly independent chain in C. Finally, the assumption $\text{span}(C) = \mathbb{R}^n$ implies that the length of every maximal linearly independent chain in C is n. □

4 The Algorithm

In this section we present our algorithm for LIP proving Theorem 1.1.

4.1 The Case $\lambda_1 = \lambda_n$

We start with the special case of lattices of rank n that satisfy $\lambda_1 = \lambda_n$ (i.e., contain n linearly independent shortest vectors), and prove the following.

Theorem 4.1. *There exists an algorithm that given two bases of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n satisfying $\lambda_1 = \lambda_n$, outputs all orthogonal linear transformations $O : \text{span}(\mathcal{L}_1) \to \text{span}(\mathcal{L}_2)$ for which $\mathcal{L}_2 = O(\mathcal{L}_1)$ in running time $n^{O(n)} \cdot s^{O(1)}$ and in polynomial space, where s denotes the input size. In addition, the number of these transformations is at most $n^{O(n)}$.*

The algorithm that implies Theorem 4.1 relies on the following theorem (recall Definition 3.7).

Theorem 4.2. *Let $\mathcal{L}$ be a lattice of rank n satisfying $\lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$, and let A denote the set of all shortest nonzero vectors of $\mathcal{L}$. Then there exists a vector $v \in \mathcal{L}^*$ that uniquely defines a linearly independent chain of length n in A and satisfies $\|v\| \leq 5n^{17/2} \cdot \lambda_1(\mathcal{L}^*)$.*

Proof. Let B be a dual Korkine-Zolotarev basis generating the lattice $\mathcal{L}$, and let C be the set of coefficients of shortest nonzero vectors of $\mathcal{L}$ in terms of the basis B, that is,

$$C = \{x \in \mathbb{Z}^n \mid Bx \in A\}.$$

Observe that Lemma 2.7 applied with $t = 1$ implies that all the entries of the integer vectors in C have absolute value at most $n^{3/2}$. Since $\lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$, A contains n linearly independent vectors, hence their coefficient vectors in C are linearly independent as well, so $\text{span}(C) = \mathbb{R}^n$. We apply the isolation lemma (Corollary 3.8) with $K = n^{3/2}$ and, say, $\varepsilon = 1/2$. We obtain that for $R = 2K(2K+1)n^3 \leq 5n^6$, there exists a vector $z \in \{1, \ldots, R\}^n$ that uniquely defines a linearly independent chain of length n in C. Since $\langle x, y \rangle = \langle Bx, B^*y \rangle$ for every $x, y \in \mathbb{R}^n$, it follows that the vector

$$v = B^*z = \sum_{i=1}^{n} z_i \cdot b_i^* \in \mathcal{L}^*$$

uniquely defines a linearly independent chain of length n in A. Finally, since B is a dual Korkine-Zolotarev basis generating $\mathcal{L}$ and $\lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$, Lemma 2.8 implies that

$$\|v\| \leq n^{5/2} \cdot 5n^6 \cdot \lambda_1(\mathcal{L}^*) = 5n^{17/2} \cdot \lambda_1(\mathcal{L}^*).$$ □

Proof of Theorem 4.1. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the lattices generated by the input bases B_1 and B_2. Consider the algorithm that acts as follows (see Algorithm 1). For $i \in \{1,2\}$, the algorithm computes the set A_i of all shortest nonzero vectors of $\mathcal{L}_i$ and the set W_i of all vectors in the dual lattice $\mathcal{L}_i^*$ of norm at most $5n^{17/2} \cdot \lambda_1(\mathcal{L}_i^*)$. These sets can be computed using the algorithm from Corollary 2.16. Given these sets, the algorithm finds a $w_1 \in W_1$ that uniquely defines a linearly independent chain of length n in A_1 and the corresponding chain $x_1, \ldots, x_n \in A_1$. The existence of w_1 is guaranteed by Theorem 4.2. Then, the algorithm goes over all vectors $w_2 \in W_2$ and for every w_2 which uniquely defines a linearly independent chain $y_1, \ldots, y_n \in A_2$ it checks if the linear transformation $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$, defined by $O(x_i) = y_i$ for every $1 \leq i \leq n$, is orthogonal and maps $\mathcal{L}_1$ to $\mathcal{L}_2$. If this is the case, then O is inserted to the output set.

Algorithm 1 Lattice Isomorphism – Special Case

Input: Two bases of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n satisfying $\lambda_1(\mathcal{L}_1) = \lambda_n(\mathcal{L}_1)$ and $\lambda_1(\mathcal{L}_2) = \lambda_n(\mathcal{L}_2)$.
Output: The set $Output$ of all orthogonal linear transformations $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ for which $\mathcal{L}_2 = O(\mathcal{L}_1)$.

1: **for all** $i = 1, 2$ **do**
2: $\quad A_i \leftarrow \{x \in \mathcal{L}_i \mid \|x\| = \lambda_1(\mathcal{L}_i)\}$ ▷ Corollary 2.16
3: $\quad W_i \leftarrow \{x \in \mathcal{L}_i^* \mid \|x\| \leq 5n^{17/2} \cdot \lambda_1(\mathcal{L}_i^*)\}$ ▷ Corollary 2.16
4: **end for**
5: **for all** $w_1 \in W_1$ **do**
6: $\quad$ **if** w_1 uniquely defines a linearly independent chain of length n in A_1 **then**
7: $\quad\quad (x_1, \ldots, x_n) \leftarrow$ the maximal linearly independent chain that w_1 uniquely defines in A_1
8: $\quad\quad$ **goto** line 11
9: $\quad$ **end if**
10: **end for**
11: **for all** $w_2 \in W_2$ **do**
12: $\quad$ **if** w_2 uniquely defines a linearly independent chain of length n in A_2 **then**
13: $\quad\quad (y_1, \ldots, y_n) \leftarrow$ the maximal linearly independent chain that w_2 uniquely defines in A_2
14: $\quad\quad O \leftarrow$ the linear transformation that maps x_i to y_i for every $1 \leq i \leq n$
15: $\quad\quad$ **if** O is orthogonal and satisfies $\mathcal{L}_2 = O(\mathcal{L}_1)$ **then**
16: $\quad\quad\quad Output \leftarrow Output \cup \{O\}$
17: $\quad\quad$ **end if**
18: $\quad$ **end if**
19: **end for**

We turn to prove the correctness of the algorithm. It is clear from the algorithm that any linear transformation in the output is orthogonal and maps $\mathcal{L}_1$ to $\mathcal{L}_2$. We claim that every orthogonal linear transformation that maps $\mathcal{L}_1$ to $\mathcal{L}_2$ is in the output. To see this, let $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ be such a transformation. Consider the vector $u = O(w_1)$ where $w_1 \in \mathcal{L}_1^*$ is the vector which is computed by the algorithm and uniquely defines a linearly independent chain $x_1, \ldots, x_n$ in A_1. Since O preserves inner products, it follows that $u \in \mathcal{L}_2^*$ and that

$$\|u\| = \|w_1\| \leq 5n^{17/2} \cdot \lambda_1(\mathcal{L}_1^*) = 5n^{17/2} \cdot \lambda_1(\mathcal{L}_2^*).$$

Therefore, u belongs to W_2. Since $A_2 = O(A_1)$, it follows that u uniquely defines a linearly inde-

pendent chain of length n in A_2, and that this chain is $O(x_1), \ldots, O(x_n)$. Thus, the chain $y_1, \ldots, y_n$, which is computed by the algorithm for u, satisfies $O(x_i) = y_i$ for every $1 \leq i \leq n$, so the algorithm includes O in its output.

Now we analyze the running time and the space complexity of Algorithm 1. We start with the running time, and focus on its dependence on the rank n, ignoring terms which are polynomial in the input size s. By Corollary 2.16, the running time needed to compute the sets A_i is $n^{O(n)}$ and to compute the sets W_i is $(5n^{17/2} \cdot n)^{O(n)} = n^{O(n)}$. By Fact 2.2, we have $|A_i| = 2^{O(n)}$ and $|W_i| = n^{O(n)}$. Given a vector w and a set A, it is possible to check in time polynomial in $|A|$ and in the input size if w uniquely defines a linearly independent chain of length n in A, and if so to compute the chain. Hence the total running time is $n^{O(n)}$. For the space complexity of the algorithm recall that the algorithm from Corollary 2.16 requires only polynomial space. In order to have only polynomial space complexity in Algorithm 1, it should be implemented in a way that the sets A_i and W_i are not stored at any step of the algorithm. Instead, whenever the algorithm checks if a vector uniquely defines a linearly independent chain of length n in a set A_i, this set should be recomputed. Since the number of calls to this procedure is $n^{O(n)}$, the running time remains $n^{O(n)}$. Similarly, the sets W_i should not be stored, as it suffices to enumerate their elements in order to implement the algorithm. Therefore, the algorithm can be implemented in a way that requires only polynomial space complexity and the stated running time. Finally, observe that the number of returned orthogonal linear transformations is bounded from above by $|W_2|$, hence is at most $n^{O(n)}$. □

4.2 The General Case

Now, we turn to deal with the general case, where the successive minima of the input lattices are not necessarily all equal. We start with the following simple lemma.

Lemma 4.3. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two lattices, and let $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ be an orthogonal linear transformation satisfying $\mathcal{L}_2 = O(\mathcal{L}_1)$. For $i \in \{1, 2\}$, let V_i be the linear subspace spanned by all shortest nonzero vectors of $\mathcal{L}_i$, and let π_i denote the projection of $\mathrm{span}(\mathcal{L}_i)$ to the orthogonal complement to V_i. Then,*

1. *The restriction $O|_{V_1}$ of O to V_1 is an orthogonal linear transformation mapping the lattice $\mathcal{L}_1 \cap V_1$ to the lattice $\mathcal{L}_2 \cap V_2$. In addition, the lattices $\mathcal{L}_1 \cap V_1$ and $\mathcal{L}_2 \cap V_2$ have the same rank k and they both satisfy $\lambda_1 = \lambda_k$.*
2. *The restriction $O|_{\pi_1(\mathrm{span}(\mathcal{L}_1))}$ of O to $\pi_1(\mathrm{span}(\mathcal{L}_1))$ is an orthogonal linear transformation mapping the lattice $\pi_1(\mathcal{L}_1)$ to the lattice $\pi_2(\mathcal{L}_2)$.*

Proof. Since O preserves lengths, v is a shortest nonzero vector of $\mathcal{L}_1$ if and only if $O(v)$ is a shortest nonzero vector of $\mathcal{L}_2$, thus $O(V_1) = V_2$. This implies that O satisfies $O(\mathcal{L}_1 \cap V_1) = \mathcal{L}_2 \cap V_2$, so does its restriction $O|_{V_1}$. Therefore, the lattices $\mathcal{L}_1 \cap V_1$ and $\mathcal{L}_2 \cap V_2$ are isomorphic and, in particular, have the same rank k. Since these lattices contain k linearly independent shortest nonzero vectors, it follows that they both satisfy $\lambda_1 = \lambda_k$. Now, observe that every $x \in \mathrm{span}(\mathcal{L}_1)$ satisfies $O(\pi_1(x)) = \pi_2(O(x))$. Hence,

$$O(\pi_1(\mathcal{L}_1)) = \pi_2(O(\mathcal{L}_1)) = \pi_2(\mathcal{L}_2),$$

so $O|_{\pi_1(\mathrm{span}(\mathcal{L}_1))}$ is an orthogonal linear transformation mapping $\pi_1(\mathcal{L}_1)$ to $\pi_2(\mathcal{L}_2)$. □

Equipped with Lemma 4.3, Theorem 1.1 follows quite easily.

Proof of Theorem 1.1. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the lattices generated by the input bases B_1 and B_2. Consider the algorithm that acts as follows (see Algorithm 2). For $i \in \{1,2\}$, the algorithm computes the linear subspace V_i spanned by all shortest nonzero vectors of $\mathcal{L}_i$ and the projection π_i of $\mathrm{span}(\mathcal{L}_i)$ to the orthogonal complement to V_i. This can be done using the algorithm from Corollary 2.16 for computing shortest nonzero vectors of a given lattice. If the lattices $\mathcal{L}_1 \cap V_1$ and $\mathcal{L}_2 \cap V_2$ do not have the same rank, then the algorithm outputs that the lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are not isomorphic. Otherwise, the algorithm computes, using the algorithm from Theorem 4.1, all orthogonal linear transformations O_1 that map $\mathcal{L}_1 \cap V_1$ to $\mathcal{L}_2 \cap V_2$, and recursively computes all orthogonal linear transformations O_2 that map $\pi_1(\mathcal{L}_1)$ to $\pi_2(\mathcal{L}_2)$. Finally, the algorithm checks for every such pair (O_1, O_2) if the transformation O, defined on $\mathrm{span}(\mathcal{L}_1)$ by $O|_{V_1} = O_1$ and $O|_{\pi_1(\mathrm{span}(\mathcal{L}_1))} = O_2$, maps $\mathcal{L}_1$ to $\mathcal{L}_2$, and if so, inserts it to the output set.

Algorithm 2 Lattice Isomorphism – General Case

Input: Two bases of lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ of rank n.

Output: The set $Output$ of all orthogonal linear transformations $O : \mathrm{span}(\mathcal{L}_1) \to \mathrm{span}(\mathcal{L}_2)$ for which $\mathcal{L}_2 = O(\mathcal{L}_1)$.

for all $i = 1, 2$ **do**
  $V_i \leftarrow \mathrm{span}(\{x \in \mathcal{L}_i \mid \|x\| = \lambda_1(\mathcal{L}_i)\})$ ▷ Corollary 2.16
  $\pi_i \leftarrow$ the projection of $\mathrm{span}(\mathcal{L}_i)$ to the orthogonal complement to V_i
end for
if $\mathrm{rank}(\mathcal{L}_1 \cap V_1) \neq \mathrm{rank}(\mathcal{L}_2 \cap V_2)$ **then**
  return $\emptyset$
end if
$Output_1 \leftarrow$ Lattice Isomorphism Special Case$(\mathcal{L}_1 \cap V_1, \mathcal{L}_2 \cap V_2)$ ▷ Algorithm 1 (Theorem 4.1)
$Output_2 \leftarrow$ Lattice Isomorphism General Case$(\pi_1(\mathcal{L}_1), \pi_2(\mathcal{L}_2))$ ▷ A recursive call
$Output \leftarrow \emptyset$
for all $O_1 \in Output_1, O_2 \in Output_2$ **do**
  $O \leftarrow$ the linear transformation defined on $\mathrm{span}(\mathcal{L}_1)$ by $O|_{V_1} = O_1$ and $O|_{\pi_1(\mathrm{span}(\mathcal{L}_1))} = O_2$
  if O satisfies $\mathcal{L}_2 = O(\mathcal{L}_1)$ **then**
    $Output \leftarrow Output \cup \{O\}$
  end if
end for
return $Output$

It is easy to see that the rank of the input lattices decreases in every recursive call of the algorithm. Therefore, the algorithm terminates, and its correctness follows from Lemma 4.3.

We turn to show that the running time of Algorithm 2 on lattices of rank n is $n^{O(n)} \cdot s^{O(1)}$, where s denotes the input size. As before, we ignore in the analysis terms which are polynomial in s. Denote by $r \leq n$ the number of recursive calls, let n_i denote the rank of the input lattices of

the ith recursive call, and observe that $\sum_{i=1}^{r} n_i = n$. We analyze the total running time of every step of Algorithm 2 in all the r recursive calls together.

Using Corollary 2.16, it can be shown that the running time of computing the subspaces V_1 and V_2 and the projections π_1 and π_2 in the ith recursive call is $n_i^{O(n_i)}$. Hence, the total running time of the loop in line 1 is $n^{O(n)}$. Given V_i and π_i, by Lemmas 2.13 and 2.12, it is possible to compute in polynomial time bases for the lattices $\mathcal{L}_i \cap V_i$ and $\pi_i(\mathcal{L}_i)$. By Theorem 4.1, the output of Algorithm 1 (line 8) in the ith recursive call has size $n_i^{O(n_i)}$, and its computation requires running time $n_i^{O(n_i)}$. This implies that the total running time of the calls to Algorithm 1 is $n^{O(n)}$, and that the total running time of the loop in line 11 is at most

$$\prod_{i=1}^{r} n_i^{O(n_i)} \leq \prod_{i=1}^{r} n^{O(n_i)} = n^{O(n)},$$

so the total running time of Algorithm 2 is bounded by $n^{O(n)} \cdot s^{O(1)}$, as required. In addition, it follows that the number of linear transformations that the algorithm outputs is at most $n^{O(n)}$.

We finally note that it is not difficult to see that Algorithm 2 can be implemented in polynomial space and in running time as before. To do so, for computing V_i (line 2) one has to enumerate the shortest nonzero vectors of $\mathcal{L}_i$ and to store only the ones which are linearly independent of the previously stored ones. Similarly, in the loop of line 11, the elements of the $Output_i$'s should not be stored but should be recursively enumerated in parallel. Since the depth of the recursion is at most n, all the linear transformations which together define a purported O can be simultaneously stored in space complexity polynomial in the input size. □

5 The Lattice Isomorphism Problem is in SZK

In this section we present an SZK proof system for the complement of LIP implying Theorem 1.3. To do so, we need some properties of the discrete Gaussian distribution on lattices, proven in the following section.

5.1 Gaussian-Distributed Generating Sets

In Lemma 5.4 below we bound the number of samples from the discrete Gaussian distribution $D_{\mathcal{L},s}$ needed in order to get a generating set of $\mathcal{L}$ with high probability. We start with the following lemma.

Lemma 5.1. *For every lattice $\mathcal{L}$ of rank n and a strict sublattice $\mathcal{M} \subsetneq \mathcal{L}$,*

1. *If* $\mathrm{span}(\mathcal{M}) \subsetneq \mathrm{span}(\mathcal{L})$ *and* $s \geq c \cdot \lambda_n(\mathcal{L})$ *then* $\Pr_{x \in D_{\mathcal{L},s}}[x \in \mathcal{M}] \leq \frac{1}{1+e^{-\pi c^{-2}}}$.
2. *If* $s \geq c \cdot bl(\mathcal{L})$ *then* $\Pr_{x \in D_{\mathcal{L},s}}[x \in \mathcal{M}] \leq \frac{1}{1+e^{-\pi c^{-2}}}$.

Proof. For a lattice $\mathcal{L}$ and a strict sublattice $\mathcal{M}$, let w be a vector in $\mathcal{L} \setminus \mathcal{M}$. Then, the lattice $\mathcal{M}$ and its coset $w + \mathcal{M}$ are disjoint and are both contained in $\mathcal{L}$. Using Claim 2.10, we obtain that

$$\rho_s(\mathcal{L}) \geq \rho_s(\mathcal{M}) + \rho_s(w + \mathcal{M}) \geq (1 + \rho_s(w)) \cdot \rho_s(\mathcal{M}),$$

which implies that

$$\Pr_{x \in D_{\mathcal{L},s}}[x \in \mathcal{M}] = \frac{\rho_s(\mathcal{M})}{\rho_s(\mathcal{L})} \leq \frac{1}{1+\rho_s(w)}.$$

For Item 1, observe that if $\mathrm{span}(\mathcal{M}) \subsetneq \mathrm{span}(\mathcal{L})$ then there exists a vector $w \in \mathcal{L} \setminus \mathcal{M}$ such that $\|w\| \leq \lambda_n(\mathcal{L})$. Applying the above argument with this w for $s \geq c \cdot \lambda_n(\mathcal{L})$ completes the proof. For Item 2, recall that the lattice $\mathcal{L}$ is generated by a basis all of whose vectors are of norm at most $bl(\mathcal{L})$. Since $\mathcal{M}$ is a strict sublattice of $\mathcal{L}$ at least one of these vectors does not belong to $\mathcal{M}$, so there exists a vector $w \in \mathcal{L} \setminus \mathcal{M}$ such that $\|w\| \leq bl(\mathcal{L})$. Apply again the above argument for $s \geq c \cdot bl(\mathcal{L})$, and we are done. □

Remark 5.2. *Note that the bounds given in Lemma 5.1 converge to* $1/2$ *as the parameter* s *increases.*

The following corollary resembles Corollary 3.16 in [39].

Corollary 5.3. *For every lattice* $\mathcal{L}$ *of rank* n *and* $s \geq \lambda_n(\mathcal{L})$*, the probability that a set of* n^2 *vectors chosen independently from* $D_{\mathcal{L},s}$ *contains no* n *linearly independent vectors is* $2^{-\Omega(n)}$*.*

Proof. Let $u_1, \ldots, u_{n^2}$ denote n^2 samples from $D_{\mathcal{L},s}$. For every $1 \leq i \leq n$ let A_i be the event that

$$\mathrm{rank}(\mathrm{span}(u_1, \ldots, u_{(i-1)n})) = \mathrm{rank}(\mathrm{span}(u_1, \ldots, u_{in})) < n.$$

Fix some i and condition on a fixed choice of $u_1, \ldots, u_{(i-1)n}$ that span a subspace of rank smaller than n. Observe that, by Item 1 of Lemma 5.1, the probability that

$$u_{(i-1)n+1}, \ldots, u_{in} \in \mathcal{L} \cap \mathrm{span}(u_1, \ldots, u_{(i-1)n})$$

is at most $(1+e^{-\pi})^{-n} = 2^{-\Omega(n)}$. This implies that the event A_i happens with probability $2^{-\Omega(n)}$. Therefore, except with probability $2^{-\Omega(n)}$, none of the A_i's happens, thus $u_1, \ldots, u_{n^2}$ contain n linearly independent vectors. □

Lemma 5.4. *For every lattice* $\mathcal{L}$ *of rank* n *satisfying* $\det(\mathcal{L}) \geq 1$ *and every* $s \geq bl(\mathcal{L})$*, the probability that a set of*

$$n^2 + n\log(s\sqrt{n})(n + 20\log\log(s\sqrt{n}))$$

vectors chosen independently from $D_{\mathcal{L},s}$ *does not generate* $\mathcal{L}$ *is* $2^{-\Omega(n)}$*.*

Proof. Let $u_1, \ldots, u_{n^2}$ denote the first n^2 samples from $D_{\mathcal{L},s}$. Since $s \geq bl(\mathcal{L}) \geq \lambda_n(\mathcal{L})$, Corollary 5.3 implies that, except with probability $2^{-\Omega(n)}$, they contain n linearly independent vectors. By Lemma 2.9 and the union bound, with a similar probability, each of these vectors has norm at most $s \cdot \sqrt{n}$. Denote by $\mathcal{M}_0 \subseteq \mathcal{L}$ the sublattice generated by $u_1, \ldots, u_{n^2}$. By the assumption $\det(\mathcal{L}) \geq 1$, we obtain that the index of $\mathcal{M}_0$ in $\mathcal{L}$ satisfies

$$|\mathcal{L} : \mathcal{M}_0| = \frac{\det(\mathcal{M}_0)}{\det(\mathcal{L})} \leq (s \cdot \sqrt{n})^n.$$

Now, define $h = n\log(s\sqrt{n})$ and $\ell = n + 20\log\log(s\sqrt{n})$, and let $v_1, \ldots, v_{h\cdot\ell}$ be the remaining $h \cdot \ell$ vectors chosen from $D_{\mathcal{L},s}$. For every $1 \leq i \leq h$ define the sublattice

$$\mathcal{M}_i = \mathcal{L}(u_1, \ldots, u_{n^2}, v_1, \ldots, v_{i\cdot\ell}) \subseteq \mathcal{L},$$

and let A_i be the event that $\mathcal{M}_{i-1} = \mathcal{M}_i \subsetneq \mathcal{L}$. If none of the A_i's happens, then $M_i = \mathcal{L}$ for some i or $|\mathcal{M}_i : \mathcal{M}_{i-1}| \geq 2$ for every i. In the latter case it follows that

$$|\mathcal{L} : \mathcal{M}_h| \leq (s\sqrt{n})^n \cdot 2^{-h} = 1.$$

Therefore, in both cases the lattice $\mathcal{M}_h$, which is generated by the $n^2 + h \cdot \ell$ samples from $D_{\mathcal{L},s}$, equals the lattice $\mathcal{L}$.

In order to complete the proof it remains to show that the probability of every event A_i is at most $2^{-\Omega(n)-\log\log(s\sqrt{n})}$, as this implies by the union bound that the probability that at least one A_i happens is at most

$$h \cdot 2^{-\Omega(n)-\log\log(s\sqrt{n})} \leq 2^{-\Omega(n)}.$$

So fix some i, condition on a fixed choice of $v_1, \ldots, v_{(i-1)\ell}$ which do not generate $\mathcal{L}$, and apply Item 2 of Lemma 5.1 to obtain that the probability that all the vectors $v_{(i-1)\cdot\ell+1}, \ldots, v_{i\cdot\ell}$ belong to $\mathcal{M}_{i-1}$ is at most $(1 + e^{-\pi})^{-\ell} \leq 2^{-\Omega(n)-\log\log(s\sqrt{n})}$. Thus, this is an upper bound on the probability that the event A_i happens, so we are done. □

5.2 LIP is in SZK

Theorem 5.5. LIP *is in* SZK.

Proof. It is sufficient to prove that the *complement* of LIP has a statistical zero-knowledge proof system with respect to a honest verifier (HVSZK) [35]. This follows from the HVSZK proof system given in Algorithm 3. Let B_1 and B_2 be two bases of lattices of rank n, and define

$$s = \max(\|B_1\|, \|B_2\|) \cdot \sqrt{\ln(2n+4)/\pi},$$

where $\|B\|$ denotes the norm of a longest vector in a basis B. It can be assumed without loss of generality that the lattices have determinant 1, since if their determinants are distinct then they are clearly not isomorphic, and otherwise they can be scaled to have determinant 1.

In the proof system, the verifier chooses uniformly at random an $i \in \{1, 2\}$ and sends to the prover the Gram matrix $G = W^T \cdot W$, where the columns of W are

$$N = n^2 + n\log(s\sqrt{n})(n + 20\log\log(s\sqrt{n}))$$

lattice vectors of $\mathcal{L}(B_i)$ independently chosen from $D_{\mathcal{L}(B_i),s}$ using the algorithm SampleD from Lemma 2.11. Finally, the verifier accepts if and only if the prover correctly guesses i.

By Lemma 5.4, except with probability $2^{-\Omega(n)}$, the set W generates the lattice $\mathcal{L}(B_i)$, so for simplicity we assume from now on that this is the case. The running time needed by the verifier in the protocol is clearly polynomial in the input size. So we turn to prove correctness, that is, that for every prover's strategy the verifier rejects YES instances with some non-negligible probability, whereas NO instances are accepted for some prover's strategy.

Assume that (B_1, B_2) is a YES instance. This means that the lattices $\mathcal{L}(B_1)$ and $\mathcal{L}(B_2)$ are isomorphic, so there exists an orthogonal linear transformation $O : \text{span}(B_1) \to \text{span}(B_2)$ mapping $\mathcal{L}(B_1)$ to $\mathcal{L}(B_2)$. Recall that the matrix W is chosen either from $D^N_{\mathcal{L}(B_1),s}$ or from $D^N_{O(\mathcal{L}(B_1)),s}$. Since O preserves inner products, the Gram matrix of W equals the Gram matrix of $O(W)$. Therefore, the

Algorithm 3 An HVSZK Proof System for the complement of LIP

Input: Two bases B_1 and B_2 of lattices of rank n with determinant 1.

1: $s \leftarrow \max(\|B_1\|, \|B_2\|) \cdot \sqrt{\ln(2n+4)/\pi}$ ▷ Lemma 2.11
2: Verifier chooses uniformly at random $i \in \{1,2\}$
3: $N \leftarrow n^2 + n\log(s\sqrt{n})(n + 20\log\log(s\sqrt{n}))$ ▷ Lemma 5.4
4: **for all** $1 \leq j \leq N$ **do**
5: $\quad w_j \leftarrow \mathsf{SampleD}(B_i, s)$ ▷ Lemma 2.11
6: **end for**
7: Verifier sends $G = W^T \cdot W$ to the prover
8: Prover returns $i' \in \{1,2\}$
9: Verifier accepts if and only if $i = i'$

distribution of the matrix G, which is sent to the prover, is independent of i. Hence, the probability that the verifier accepts is at most $1/2$.

Now, assume that (B_1, B_2) is a NO instance, and consider the following strategy for the computationally unbounded prover: Given G, the prover returns the i' for which there exists a vector set W of size N that generates $\mathcal{L}(B_{i'})$ and satisfies $G = W^T \cdot W$. To complete the proof it remains to show that this i' is unique whenever $\mathcal{L}(B_1)$ and $\mathcal{L}(B_2)$ are not isomorphic. Indeed, if $W_1^T \cdot W_1 = W_2^T \cdot W_2$ then by Fact 2.1 there exists an orthogonal linear transformation O mapping W_1 to W_2, and this implies that the lattices generated by the sets W_1 and W_2 are isomorphic.

To complete the proof, it remains to observe that the presented proof system is statistical zero-knowledge, that is, the honest verifier "learns nothing" from the interaction with the prover other than the fact that the lattices are not isomorphic. To see this, consider the probabilistic polynomial-time simulator that runs the proof system playing the roles of both the honest verifier and the prover, where as prover it returns i' which equals the i chosen by the simulated verifier. Observe that the distribution of the transcript obtained from this simulation is statistically close to the one obtained from a run by a honest verifier and a prover. □

Remark 5.6. *We remark that by fixing the answer of the prover in the unlikely event that the set W does not generate the lattice, it follows that the complement of* LIP *has a honest verifier* perfect *zero-knowledge proof system.*

Acknowledgement

We would like to thank Frank Vallentin for useful comments.

References

[1] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. *Journal of the ACM*, 52(5):749–765, 2005. Preliminary version in FOCS'04.

[2] W. Aiello and J. Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991. Preliminary version in FOCS'87.

[3] M. Ajtai. Generating hard instances of lattice problems. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 1–32. Dept. Math., Seconda Univ. Napoli, Caserta, 2004. Preliminary version in STOC'96.

[4] V. Arvind and P. Mukhopadhyay. Derandomizing the isolation lemma and lower bounds for circuit size. In *APPROX-RANDOM*, pages 276–289, 2008.

[5] L. Babai. Automorphism groups, isomorphism, reconstruction. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, chapter 27, pages 1447–1540. North-Holland, Amsterdam, 1996.

[6] L. Babai, P. Codenotti, J. A. Grochow, and Y. Qiao. Code equivalence and group isomorphism. In *SODA*, pages 1395–1408, 2011.

[7] L. Babai and E. M. Luks. Canonical labeling of graphs. In *STOC*, pages 171–183, 1983.

[8] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[9] E. S. Barnes and N. J. A. Sloane. New lattice packings of spheres. *Canadian J. Math.*, 35(1):117–130, 1983.

[10] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.

[11] J.-Y. Cai and A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *FOCS*, pages 468–477, 1997.

[12] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM J. Comput.*, 24(5):1036–1050, 1995. Preliminary version in STOC'93.

[13] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer-Verlag, 1993.

[14] J. H. Conway and N. J. Sloane. *Sphere packings, lattices and groups*. Springer Verlag, 3rd edition, 1998.

[15] M. Dutour Sikirić, A. Schürmann, and F. Vallentin. Complexity and algorithms for computing Voronoi cells of lattices. *Math. Comput.*, 78(267):1713–1731, 2009.

[16] L. Fortnow. The complexity of perfect zero-knowledge. In S. Micali, editor, *Advances in Computing Research*, volume 5, pages 327–343. JAC Press, Inc., 1989. Preliminary versions in SCTC'87 and STOC'87.

[17] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[18] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. System Sci.*, 60(3):540–563, 2000. Preliminary version in STOC'98.

[19] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991. Preliminary version in FOCS'86.

[20] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory of Computing*, 8(23):513–531, 2012. Preliminary version in STOC'07.

[21] L. A. Hemaspaandra and M. Ogihara. The isolation technique. In *The Complexity Theory Companion*, chapter 4, pages 67–89. Springer-Verlag, 2002.

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

[23] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, Sept. 2005. Preliminary version in FOCS'04.

[24] A. Klivans and D. A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *STOC*, pages 216–223, 2001.

[25] G. Kuperberg. Personal communication, 2013.

[26] J. C. Lagarias, H. W. Lenstra, Jr., and C.-P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.

[27] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[28] H. Lenstra, R. Schoof, and A. Silverberg. Lattices with symmetry. 2013. Unpublished.

[29] D. Micciancio. Efficient reductions among lattice problems. In *SODA*, pages 84–93, 2008.

[30] D. Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(22):487–512, 2012.

[31] 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, MA, 2002.

[32] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. *SIAM J. Comput.*, 42(3):1364–1391, 2013. Preliminary version in STOC'10.

[33] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. Preliminary version in STOC'87.

[34] H. Narayanan, H. Saran, and V. V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arboresences and edge-disjoint spanning trees. *SIAM J. Comput.*, 23(2):387–397, 1994. Preliminary version in SODA'92.

[35] T. Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000. Preliminary version in STOC'96.

[36] E. Petrank and R. M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997.

[37] W. Plesken and M. Pohst. Constructing integral lattices with prescribed minimum. I. *Mathematics of Computation*, 45(171):209–221, 1985.

[38] W. Plesken and B. Souvignier. Computing isometries of lattices. *J. Symb. Comput.*, 24(3-4):327–334, 1997.

[39] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34, 2009. Preliminary version in STOC'05.

[40] K. Reinhardt and E. Allender. Making nondeterminism unambiguous. *SIAM J. Comput.*, 29(4):1118–1131, 2000. Preliminary version in FOCS'97.

[41] M. Szydlo. Hypercubic lattice reduction and analysis of GGH and NTRU signatures. In *EUROCRYPT*, pages 433–448. Springer-Verlag, 2003.

[42] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. Preliminary version in FOCS'89.

[43] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986. Preliminary version in STOC'85.

[44] A. Wigderson. NL/poly $\subseteq \oplus$ L/poly. In *Structure in Complexity Theory Conference (SCTC)*, pages 59–62, 1994.