Zero-Knowledge Interactive Proof Systems for New Lattice Problems

Claude Crépeau* and Raza Ali Kazmi*

McGiLL University
crepeau@cs.mcgill.ca, raza-ali.kazmi@mail.mcgill.ca

Abstract. In this work we introduce a new hard problem in lattices called **Isometric Lattice Problem** (**ILP**) and reduce **Linear Code Equivalence** over prime fields and **Graph Isomorphism** to this problem. We also show that this problem has an (efficient prover) perfect zero-knowledge interactive proof; this is the only hard problem in lattices that is known to have this property (with respect to malicious verifiers). Under the assumption that the polynomial hierarchy does not collapse, we also show that **ILP** cannot be **NP-complete**. We finally introduce a variant of **ILP** over the rationals radicands and provide similar results for this new problem.

1 Introduction

Zero-Knowledge interactive proof systems **ZKIP** [6] have numerous applications in cryptography such as Identification Schemes, Authentication Schemes, Multiparty Computations, etc. Appart from cryptographic applications these proof systems play an important part in the study of complexity theory. The first **IP** for lattice problems (**coGapCVP**$_\gamma$, **coGapSVP**$_\gamma$) was presented by Goldreich and Goldwasser [13]. However, these proofs are only *honest-verifier* Perfect Zero-Knowledge and known to have *inefficient provers*. Micciancio and Vadhan [10] presented Interactive Proofs for **GapCVP**$_\gamma$ and **GapSVP**$_\gamma$. These proofs are Statistical Zero-Knowledge and have efficient provers[1] as well. In this paper we introduce a new hard problem called **ISOMETRIC LATTICE PROBLEM** (**ILP**). We present **IP** systems for the **ILP**. These proof systems are *Perfect* Zero-Knowledge and have *efficient* provers. We show that a variant of **ILP** over the integers is at least as hard as **Graph Isomorphism** (**GI**) [4, 5] and **Linear Code Equivalence** (**LCE**) [7, 5]. This is the only hard problem problem known in lattices that have a (*malicious*-verifier) Perfect Zero-Knowledge **IP** system with an *efficient* prover. We also show that **ILP** is unlikely to be **NP-complete**. Finally we also introduce another variant of **ILP** over the rational-radicands and provide similar results for this problem.

* Supported in part by Québec's FRQNT, Canada's NSERC and CIFAR.

[1] The Prover runs in probabilistic polynomial time given a certificate for the input string.

2 Notations

For any matrix $\mathbf{A}$, we denote its transpose by $\mathbf{A^t}$. Let $O(n,\mathbb{R}) = \{Q \in \mathbb{R}^{n\times n} : Q \cdot Q^{\mathbf{t}} = \mathbf{I}\}$ denote the group of $n \times n$ orthogonal matrices over $\mathbb{R}$. Let $GL_k(\mathbb{Z})$ denote the group of $k \times k$ invertible (unimodular) matrices over $\mathbb{Z}$. Let $GL_k(\mathbb{F}_q)$ denote the set of $k \times k$ invertible matrices over the finite field $\mathbb{F}_q$. Let $\mathcal{P}_n$ denote the set of $n \times n$ permutation matrices. Let σ_n be the set of all permutations of $\{1,\ldots,n\}$. For $\pi \in \sigma_n$, we denote P_π the corresponding $n \times n$ permutation matrix. $\mathcal{P}(n,\mathbb{F}_q)$ denotes the set of $n \times n$ monomial matrices (there is exactly one nonzero entry in each row and each column) over $\mathbb{F}_q$. $\mathcal{D}_{\epsilon_n}$ is the set of diagonal matrices $D_\epsilon = diag(\epsilon_1,\ldots,\epsilon_n)$, $\epsilon_i = \pm 1$ for $i = 1,\ldots,n$. For a real vector $\mathbf{v} = (v_1,\ldots,v_n)$ we denote its Euclidean norm by $\|\mathbf{v}\| = \sqrt{v_1^2 + \cdots + v_n^2}$ and max-norm $\|\mathbf{v}\|_\infty = max_{i=1}^n |v_i|$ and for any matrix $\mathbf{B} = [\mathbf{b}_1|\mathbf{b}_2|\ldots|\mathbf{b}_k] \in \mathbb{R}^{n\times k}$ we define its norm by $\|\mathbf{B}\| = max_{i=1}^n \|\mathbf{b}_i\|$. For any ordered set of linearly independent vectors $\{\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_k\}$, we denote $\{\tilde{\mathbf{b}}_1,\tilde{\mathbf{b}}_2,\ldots,\tilde{\mathbf{b}}_k\}$, its Gram-Schmidt orthogonalization.

2.1 Lattices

Let $\mathbb{R}^n$ be an n-dimensional Euclidean space and let $\mathbf{B} \in \mathbb{R}^{n\times k}$ be a matrix of rank k. A lattice $\mathcal{L}(\mathbf{B})$ is the set of all vectors

$$\mathcal{L}(\mathbf{B}) = \left\{ \mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^k \right\}.$$

The integer n and k are called the dimension and rank of $\mathcal{L}(\mathbf{B})$. A lattice is called full dimensional if $k = n$. Two lattices $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are equivalent if and only if there exists a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{k\times k}$ such that $\mathbf{B}_1 = \mathbf{UB}_2$.

2.2 q-ary Lattices

A lattice $\mathcal{L}$ is called q-ary, if it satisfies $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$ for a positive integer q. In other words, the membership of a vector $\mathbf{v} \in \mathcal{L}$ is given by $\mathbf{v} \bmod q$. Let $\mathbf{G} = [\mathbf{g}_1|\ldots|\mathbf{g}_k] \in \mathbb{Z}_q^{n\times k}$ be a $n \times k$ matrix of rank k over $\mathbb{Z}_q^{n\times k}$. We define below two important families of q-ary lattices used in cryptography

$$\Lambda_q(\mathbf{G}) = \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{y} \equiv \mathbf{G} \cdot \mathbf{s} \pmod q, \text{for some vector } \mathbf{s} \in \mathbb{Z}^k\}$$

$$\Lambda_q^\top(\mathbf{G}) = \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{y} \cdot \mathbf{G} \equiv 0 \pmod q\}.$$

A basis $\mathbf{B}$ of $\Lambda_q(\mathbf{G})$ is

$$\mathbf{B} = [\mathbf{g}_1|\ldots|\mathbf{g}_k|\mathbf{b}_{k+1}|\ldots|\mathbf{b}_n] \in \mathbb{Z}^{n\times n}$$

where $\mathbf{b}_j = (0,...,q,...,0) \in \mathbb{Z}^n$ is a vector with its j-th coordinate equal to q and all other coordinates are 0, $k+1 \le j \le n$. A basis of $\Lambda_q^\top$ is given by $q \cdot (\mathbf{B}^{-1})^{\mathbf{t}}$.

2.3 Discrete Gaussian distribution on Lattices

For any $s > 0$, $\mathbf{c} \in \mathbb{R}^n$, we define a Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter s.

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{s,\mathbf{c}}(\mathbf{x}) = e^{\frac{-\pi\|\mathbf{x}-\mathbf{c}\|}{s^2}}.$$

Let $\mathcal{L}$ be any n dimensional lattice and $\rho_{s,\mathbf{c}}(\mathcal{L}) = \sum_{y\in\mathcal{L}} \rho_{s,\mathbf{c}}(y)$. We define a *Discrete Gaussian* distribution on $\mathcal{L}$

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

Theorem 1 *Given a basis $\mathbf{B} = [\mathbf{b}_1|\dots|\mathbf{b}_k] \in \mathbb{R}^{n\times k}$ of an n-dimensional lattice $\mathcal{L}$, a parameter $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$ and a center $\mathbf{c} \in \mathbb{R}^n$, the algorithm* SampleD ([2], section 4.2 page 14) *outputs a sample from a distribution that is statistically close to $D_{s,\mathbf{c},\mathcal{L}}$.*

Theorem 2 *There is a deterministic polynomial-time algorithm that, given an arbitrary basis $\{\mathbf{b}_1,\dots,\mathbf{b}_k\}$ of an n-dimensional lattice $\mathcal{L}$ and a set of linearly independent lattice vectors $\mathbf{S} = [\mathbf{s}_1|\mathbf{s}_2\dots|\mathbf{s}_k] \in \mathcal{L}$ with ordering $\|\mathbf{s}_1\| \leq \|\mathbf{s}_2\| \leq \dots \leq \|\mathbf{s}_k\|$, outputs a basis $\{\mathbf{r}_1\dots\mathbf{r}_k\}$ of $\mathcal{L}$ such that $\|\tilde{\mathbf{r}}_i\| \leq \|\tilde{\mathbf{s}}_i\|$ for $1 \leq i \leq k$.*

2.4 Orthogonal Matrices and Givens Rotations

A Givens rotation is an orthogonal $n \times n$ matrix of the form

$$G_{(i,j,\theta)} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}, i \neq j$$

The non-zero elements of a Givens matrix $G_{(i,j,\theta)}$ are given by

$$g_{k,k} = 1 \text{ for } k \neq i,\ j \text{ and } g_{i,i} = g_{j,j} = c$$

$$g_{i,j} = s = -g_{j,i} \text{ for } i < j$$

where $c = \cos(\theta)$ and $s = \sin(\theta)$.

The product $G_{(i,j,\theta)} \cdot \mathbf{v}$ represents a counter-clockwise rotation of the vector $\mathbf{v}$ in the (i,j) plane by angle θ. Moreover, only the *i-th* and *j-th* entries of $\mathbf{v}$ are affected and the rest remains unchanged. Any orthogonal matrix $Q \in \mathbb{R}^{n\times n}$

can be written as a product of $\frac{n(n-1)}{2}$ Givens matrices and a diagonal matrix $D_\epsilon \in \mathcal{D}_{\epsilon_n}$

$$Q = D_\epsilon \left(G_{(1,2,\theta_{1,2})} \cdot \cdots G_{(1,n,\theta_{1,n})}\right) \cdot \left(G_{(2,3,\theta_{2,3})} \cdots G_{(2,n,\theta_{2,n})}\right) \cdots \left(G_{(n-1,n,\theta_{n-1,n})}\right).$$

The angles $\theta_{i,j} \in [0, 2\pi]$, $1 \leq i < j \leq n$ are called angles of rotation.

2.5 Properties of Givens Matrices

1. *Additivity*: For angles $\theta, \phi \in [0, 2\pi]$ and any vector $\mathbf{v} \in \mathbb{R}^n$
$$G_{(i,j,\phi)} \cdot G_{(i,j,\theta)}\mathbf{v} = G_{(i,j,\phi+\theta)}\mathbf{v}.$$
2. *Commutativity*: For angles $\theta_{i,j}$, $\theta_{j,i}, \theta_{y,z} \in [0, 2\pi]$ and $\{i, j\} \cap \{y, z\} = \emptyset$ or $\{i, j\} = \{y, z\}$.
$$G_{(i,j,\theta_{i,j})} \cdot G_{(j,i,\theta_{j,i})}\mathbf{v} = G_{(j,i,\theta_{j,i})} \cdot G_{(i,j,\theta_{i,j})}\mathbf{v}$$
$$G_{(i,j,\theta_{i,j})} \cdot G_{(y,z,\theta_{y,z})}\mathbf{v} = G_{(y,z,\theta_{y,z})} \cdot G_{(i,j,\theta_{i,j})}\mathbf{v}.$$
3. *Linearity*: For any Givens matrix $G_{(i,j,\theta_{i,j})}$, any vector $\mathbf{v} \in \mathbb{R}^n$ and any permutation $\pi \in \sigma_n$
$$G_{\left(\pi(i),\pi(j),\theta_{\pi(i),\pi(j)}\right)} P_\pi \cdot \mathbf{v} = P_\pi G_{(i,j,\theta_{i,j})} \cdot \mathbf{v}$$
P_π is the corresponding permutation matrix of π.

2.6 The Set $\mathcal{R}$

Computationally it is not possible to work over arbitrary real numbers as they require infinite precision. However, there are reals that can be represented finitely and one can add and multiply them without losing any precision. For example we can represent numbers $\sqrt{7}$ and $\sqrt[4]{5}$ as $< 2, 7 >$ and $< 4, 5 >$. In, general, a real number r that has the following form

$$r = a_1 \sqrt[n_{11}]{x_{11} + \sqrt[n_{21}]{x_{21} + \cdots + \sqrt[n_{k1}]{x_{k1}}}} + a_2 \sqrt[n_{12}]{x_{12} + \sqrt[n_{22}]{x_{22} + \cdots + \sqrt[n_{k2}]{x_{k2}}}} + \cdots + a_l \sqrt[n_{1l}]{x_{1l} + \sqrt[n_{2l}]{x_{2l} + \cdots + \sqrt[n_{kl}]{x_{kl}}}}.$$

where a_j's, n_{ij}'s $\in \mathbb{Q}$, x_{i_j}'s $\in \mathbb{Q}^+ \cup \{0\}$ and $l, k_1 \cdots k_l \in \mathbb{N}$; can be represented as

$$\begin{aligned} r = a_1 < n_{11}, x_{11} + < n_{21}, x_{21} + \cdots + < n_{k1}, x_{k1} >> \cdots > + \\ a_2 < n_{12}, x_{12} + < n_{22}, x_{22} + \cdots + < n_{k2}, x_{k2} >> \cdots > + \\ \cdots + a_l < n_{1l}, x_{1l} + < n_{2l}, x_{2l} + \cdots + < n_{kl}, x_{kl} >> \cdots > . \end{aligned}$$

We call such numbers *rational radicands* and denote the set of all rational radicands $\mathcal{R}$. [2]

[2] In this notation any rational number x can be represented as $\pm < 1, x >$.

2.7 The Set $\overline{O(n,\mathcal{R})}$

Let $O(n,\mathcal{R})$ denote a set of $n \times n$ orthogonal matrices over $\mathcal{R}$. In this sub-section we will define a subset $\overline{O(n,\mathcal{R})} \subset O(n,\mathcal{R})$ that has the following properties:

- Any orthogonal matrix $Q \in \overline{O(n,\mathcal{R})}$ has finite representation.
- If $Q \in \overline{O(n,\mathcal{R})}$, then $Q^{\mathbf{t}} \in \overline{O(n,\mathcal{R})}$.
- $\overline{O(n,\mathcal{R})}$ is a finite set.

Let $\mathcal{P}$ be any desired publicly known positive polynomial in the size of the input bases $\mathbf{B}_1, \mathbf{B}_2 \in O(n,\mathcal{R})$ and $\delta = \frac{\pi}{2^{\mathcal{P}}}$. We denote the set of angles $C = \{0, \delta, 2\delta, \ldots, \theta, \ldots, 2\pi - \delta\}$. We denote $\overline{O(n,\mathcal{R})}$ to be the set of $n \times n$ orthogonal matrices corresponding to C that can be written as a product of commuting Givens rotations. More, precisely

$$\overline{O(n,\mathcal{R})} = \{G_{(1,2,\theta_1)} \cdot G_{(3,4,\theta_2)} \cdots G_{(x-1,x,\theta_{x/2})} : \theta_i \in C, 1 \leq i \leq x\}.$$

where $x = n$ if n is even, otherwise $x = n-1$. Clearly $\overline{O(n,\mathcal{R})}$ is a finite set, since C is a finite set. Furthermore for any integer $\mathcal{P} \geq 2$,

$$\sin\left(\tfrac{\pi}{2^{\mathcal{P}}}\right) = \tfrac{1}{2}\underbrace{< 2, 2- < 2, 2 + \cdots + < 2, 2 >> \cdots >}_{\mathcal{P}-1}$$
$$\cos\left(\tfrac{\pi}{2^{\mathcal{P}}}\right) = \tfrac{1}{2}\underbrace{< 2, 2+ < 2, 2 + \cdots + < 2, 2 >> \cdots >}_{\mathcal{P}-1}.$$

For any integer $0 \leq N \leq 2^{\mathcal{P}+1}$ $\sin(\frac{N\pi}{2^{\mathcal{P}}})$ and $\cos(\frac{N\pi}{2^{\mathcal{P}}})$ can be computed in $O(\mathcal{P})$ time (see appendix A). Let $Q \in \overline{O(n,\mathcal{R})}$,

$$Q = G_{(1,2,\theta_1)} \cdot G_{(3,4,\theta_2)} \cdots G_{(x-1,x,\theta_{x/2})} \text{ for some } \theta_1, ..., \theta_i, ..., \theta_{x/2} \in C.$$

We will show that $Q^{\mathbf{t}} \in \overline{O(n,\mathcal{R})}$. Let

$$Q' = G_{(1,2,2\pi-\theta_1)} \cdot G_{(3,4,2\pi-\theta_2)} \cdots G_{(x-1,x,2\pi-\theta_{x/2})}.$$

Clearly if $\theta_i \in C$, then $2\pi - \theta_i \in C$. Therefore, it follows that $Q' \in \overline{O(n,\mathcal{R})}$.

$$\begin{aligned} Q \cdot Q' &= \left(G_{(1,2,\theta_1)} G_{(1,2,2\pi-\theta_1)}\right) \cdot \left(G_{(3,4,\theta_2)} G_{(3,4,2\pi-\theta_2)}\right) \cdots \left(G_{(x-1,x,\theta_{x/2})} G_{(x-1,x,2\pi-\theta_{x/2})}\right) \\ &= G_{(1,2,\theta_1+2\pi-\theta_1)} \cdot G_{(3,4,\theta_2+2\pi-\theta_2)} \cdots G_{(x-1,x,\theta_{x/2}+2\pi-\theta_{x/2})} \\ &= G_{(1,2,2\pi)} \cdot G_{(3,4,2\pi)} \cdots G_{(x-1,x,2\pi)} \\ &\qquad \text{but } G_{(i,j,2\pi)} = \mathbf{I}, \text{ therefore } G_{(1,2,2\pi)} \cdot G_{(3,4,2\pi)} \cdots G_{(x-1,x,2\pi)} = \mathbf{I}. \end{aligned}$$

3 Isometric Lattices

Definition 1 *Let $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{n\times k}$ be two bases of rank k. We say that two lattices $\mathcal{L}(\mathbf{B}_1) \cong \mathcal{L}(\mathbf{B}_2)$ are isometric if there exists a matrix $U \in GL_k(\mathbb{Z})$ and a matrix $Q \in O(n, \mathbb{R})$ such that $\mathbf{B}_2 = Q\mathbf{B}_1 U$.*

Decision Problem ILP: Given two matrices $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{n\times k}$, decide whether $\mathcal{L}(\mathbf{B}_1) \cong \mathcal{L}(\mathbf{B}_2)$.

3.1 Variants of ILP

Let $\mathbf{S}_{(\mathbf{B}_1,\mathbf{B}_2)} = \{\mathbf{B} \in \mathbb{R}^{n\times k} : \mathcal{L}(\mathbf{B}) \cong \mathcal{L}(\mathbf{B}_1) \cong \mathcal{L}(\mathbf{B}_2)\}$ be the set of bases that are isometric to $\mathbf{B}_1$ and $\mathbf{B}_2$. The **ILP** seems to be very similar to **LCE** [7, 5]. Therefore, it is natural to ask if one can obtain a **PZKIP** for **ILP** by mimicking the **LCE** proof system.[3] However, if we try to mimic the proof system for **LCE** we are faced with following problems. Recall that a proof system is zero-knowledge if there exists a probabilistic polynomial time simulator that can forge transcripts that are distributed identically (or statistically close to) real transcripts.

- In the **LCE** proof system the prover picks uniformly and independently invertible matrices from $\mathbb{F}_q^{k\times k}$. In comparison the corresponding set $(GL_k(\mathbb{Z}))$ in **ILP** is countably infinite. Therefore there exists no uniform distribution on $GL_k(\mathbb{Z})$.
- Computationally it is not possible to work over reals as they required infinite precision and almost all elements in $O(n, \mathbb{R})$, have infinite representation. Whereas in **LCE** every element in the corresponding set $\mathcal{P}(n, \mathbb{F}_q)$ can be represented with $O(n^2 \log q)$ bits. Note that in theory the uniform distribution exists on $O(n, \mathbb{R})$ [14–16], but computationally it is not possible to pick uniformly from $O(n, \mathbb{R})$ as this would require infinite computational power.

A natural solution would be to define some finite subsets $\overline{GL_k(\mathbb{Z})}$, $\overline{O(n, \mathbb{R})}$ of $GL_k(\mathbb{Z})$, $O(n, \mathbb{R})$ and pick uniformly from $\overline{GL_k(\mathbb{Z})}$ and $\overline{O(n, \mathbb{R})}$. However, this solution may not preserve the zero-knowledge property of the proof system. To see this let $\mathbf{B}_2 = \overline{Q}\mathbf{B}_1\overline{U}$, be two isometric bases that can be represented finitely, where $\overline{Q} \in \overline{O(n, \mathbb{R})}$ and $\overline{U} \in \overline{GL_k(\mathbb{Z})}$.

$$[\mathbf{B}_1] = \left\{\overline{Q}'\mathbf{B}_1\overline{U}' : \overline{Q}' \in \overline{O(n, \mathbb{R})} \text{ and } \overline{U}' \in \overline{GL_k(\mathbb{Z})}\right\}$$
$$[\mathbf{B}_2] = \left\{\overline{Q}'\mathbf{B}_2\overline{U}' : \overline{Q}' \in \overline{O(n, \mathbb{R})} \text{ and } \overline{U}' \in \overline{GL_k(\mathbb{Z})}\right\}.$$

1. The prover picks uniformly $i \in \{1, 2\}$.
2. The prover picks uniformly $\mathbf{B} \in [\mathbf{B}_i]$ and sends $\mathbf{B}$ to the receiver.
3. The verifier uniformly picks $j \in \{1, 2\}$ and sends j to the prover.

[3] The **IP** for **LCE** is **PZKIP** with an efficient prover see [5].

Note that the zero-knowledge property requires that from $\mathbf{B}$ the verifier should not be able to learn i except with probability $\frac{1}{2}$ (for perfect zero-knowledge) or $\frac{1}{2} + negl$ (for statistical zero-knowledge). This implies that $[\mathbf{B}_1] = [\mathbf{B}_2]$ (for perfect zero-knowledge) or $|[\mathbf{B}_1] \cup [\mathbf{B}_2]| - |[\mathbf{B}_1] \cap [\mathbf{B}_2]| = negl$ (for statistical zero-knowledge). Note that any $\mathbf{B} \in [\mathbf{B}_1]$ can only be in $[\mathbf{B}_2]$ if and only if $\overline{Q}' \cdot \overline{Q}^{\mathbf{t}} \in \overline{O(n, \mathbb{R})}$ and $\overline{U}^{-1} \cdot \overline{U}' \in \overline{GL_k(\mathbb{Z})}$. Similarly, any $\mathbf{B} \in [\mathbf{B}_2]$ can only be in $[\mathbf{B}_1]$ if and only if $\overline{Q}' \cdot \overline{Q} \in \overline{O(n, \mathbb{R})}$ and $\overline{U} \cdot \overline{U}^{-1} \in \overline{GL_k(\mathbb{Z})}$. Therefore sets $\overline{O(n, \mathbb{R})}$ and $\overline{GL_k(\mathbb{Z})}$ must be a group under multiplication. But this seems unlikely to happen in general. To see this lets try to construct a finite subgroup $\overline{O(n, \mathbb{Q})} \leq O(n, \mathbb{Q})$.

- Let $Q \in O(n, \mathbb{Q})$. We add Q in $\overline{O(n, \mathbb{Q})}$, therefore $\overline{O(n, \mathbb{Q})} \leftarrow \overline{O(n, \mathbb{Q})} \cup \{Q\}$.
- Since $\overline{O(n, \mathbb{Q})}$ has to be a multiplicative group, we must add $Q \cdot Q$ and $Q^{\mathbf{t}}$ to it. Hence $\overline{O(n, \mathbb{Q})} \leftarrow \overline{O(n, \mathbb{Q})} \cup \{Q \cdot Q\} \cup \{Q^{\mathbf{t}}\}$.
- By the same argument $Q \cdot Q \cdot Q$ and $Q^{\mathbf{t}} \cdot Q^{\mathbf{t}}$ must also be added to $\overline{O(n, \mathbb{Q})}$. Hence, this process may never end and $\overline{O(n, \mathbb{Q})}$ will become an infinite set. Similarly if we try to construct a finite subgroup $\overline{GL_k(\mathbb{Z})} \leq GL_k(\mathbb{Z})$ we will face the same problem.

In order to deal with these issues we will present two variants of isometric lattice problems. We will show that one of the variant are at least hard as **GI** and **LCE**. We further show that both variants are unlikely to be **NP-complete** unless the polynomial hierarchy collapses [18, 19].

3.2 Isometric Lattices over $\mathbb{Z}$

Definition 2 *Let $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{n \times k}$ be two bases of rank k. We say that two lattices $\mathcal{L}(\mathbf{B}_1) \cong_{\mathbb{Z}} \mathcal{L}(\mathbf{B}_2)$ are isometric over integers if there exists a matrix $U \in GL_k(\mathbb{Z})$ and a matrix $Q \in O(n, \mathbb{Z})$ such that $\mathbf{B}_2 = Q\mathbf{B}_1U$.*

Decision Problem ILP$_{\mathbb{Z}}$: Given two matrices $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{n \times k}$, decide whether $\mathcal{L}(\mathbf{B}_1) \cong_{\mathbb{Z}} \mathcal{L}(\mathbf{B}_2)$.

3.3 Isometric Lattices over $\mathcal{R} \subset \mathbb{R}$

Definition 3 *Let $\mathbf{B}_1, \mathbf{B}_2 \in \mathcal{R}^{n \times k}$ be two bases of rank k. We say that two lattices $\mathcal{L}(\mathbf{B}_1) \cong_{\mathcal{R}} \mathcal{L}(\mathbf{B}_2)$ are isometric over $\mathcal{R}$ if there exists a matrix $U \in GL_k(\mathbb{Z})$ and a matrix $Q \in \overline{O(n, \mathcal{R})}$ such that $\mathbf{B}_2 = Q\mathbf{B}_1U$.*

Decision Problem ILP$_{\mathcal{R}}$: Given two matrices $\mathbf{B}_1, \mathbf{B}_2 \in \mathcal{R}^{n \times k}$, decide whether $\mathcal{L}(\mathbf{B}_1) \cong_{\mathcal{R}} \mathcal{L}(\mathbf{B}_2)$.

4 Interactive Proof System for $\mathbf{ILP}_{\mathbb{Z}}$

The set of $n \times n$ orthogonal matrices over integers $O(n, \mathbb{Z})$ is finite and of cardinality $2^n \cdot n!$. In fact the set $O(n, \mathbb{Z})$ is exactly equal to the set of $n \times n$ signed permutation matrices. Therefore, any element $Q \in O(n, \mathbb{Z})$ can be written as a product $Q = D \cdot P$ for some $D \in \mathcal{D}_{\epsilon_n}$ and $P \in \mathcal{P}_n$. Furthermore, for any matrix $\mathbf{B} \in \mathbb{Z}^{k \times n}$ the Hermite normal form $\mathbf{HNF}(\mathbf{B})$ only depends on the lattice $\mathcal{L}(\mathbf{B})$ generated by $\mathbf{B}$ and not on a particular lattice basis. Moreover, one can compute $\mathbf{HNF}(\mathbf{B}')$ from any basis $\mathbf{B}'$ of $\mathcal{L}$ in polynomial time [17]. Since $\mathbf{HNF}(\mathbf{B}) = \mathbf{HNF}(\mathbf{B}')$, the Hermite normal form does not give any information about the input basis. This will completely bypass the need for picking random elements from the set $GL_k(\mathbb{Z})$.

An Interactive Proof for $\mathbf{ILP}_{\mathbb{Z}}$

- Input $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{n \times k}$.
  1. Repeat for $l := poly(\|\mathbf{B}_1\| + \|\mathbf{B}_2\|)$ rounds.
     (a) Prover picks uniformly an orthogonal matrix $Q' \in O(n, \mathbb{Z})$.
     (b) Prover computes $\mathbf{H} \leftarrow \mathbf{HNF}(Q'\mathbf{B}_1)$ and sends it to the verifier.
     (c) Verifier randomly picks $c \in \{1, 2\}$ and sends it to the prover.
     (d) Prover sends the verifier an orthogonal matrix $P \in O(n, \mathbb{Z})$.
        i. if $c = 1$ then $P = Q'$.
        ii. if $c = 2$ then $P = Q'Q^{\mathbf{t}}$.
  2. Verifier will accept the proof if for all l rounds $\mathbf{H} = \mathbf{HNF}(P\mathbf{B}_c)$.

Theorem 3 *The proof system for* $\mathbf{ILP}_{\mathbb{Z}}$ *is a malicious verifier perfect-zero knowledge interactive proof with an efficient prover.*

Proof:
Completeness: Clearly, if $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are isometric lattices over the integers, then the prover will never fail convincing the verifier.
Soundness: If $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are not isometric over integers, then the only way for the prover to cheat is to guess c correctly in each round. Since, c is chosen uniformly and independently from $\{1, 2\}$, the probability of prover guessing c in all round is 2^{-l}. Note that verifier's computations are done in polynomial time.
Efficient Prover: The steps $1a$ and $1d$ can be done efficiently. The Hermite normal forms can be computed in polynomial time using the algorithm presented in [17]. Therefore the expected running time of the prover is polynomial.
Zero-Knowledge: Let V^* be any probabilistic polynomial time (possiblly malicious) verifier. Let $\mathcal{T}(V^*)$ denote the set of all possible transcripts that could be produced as a result of the prover P and V^* carrying out the interactive proof with a yes instance $(\mathbf{B}_1, \mathbf{B}_2)$ of $\mathbf{ILP}_{\mathbb{Z}}$. Let S denote the simulator, which will produce the possible set of forged transcripts $\mathcal{T}(S)$. We denote $\mathbf{Pr}_{V^*}(\mathcal{T})$ the probability distribution on $\mathcal{T}(V^*)$ and we denote $\mathbf{Pr}_S(\mathcal{T})$ the probability distribution on $\mathcal{T}(S)$.

We will show that:

1. The expected running time of S is polynomial.
2. $\mathbf{Pr}_{V^*}(\mathcal{T}) = \mathbf{Pr}_S(\mathcal{T})$ i.e. the two distributions are identical.

Input: $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{n\times k}$ such that $\mathcal{L}(\mathbf{B}_1) \cong_{\mathbb{Z}} \mathcal{L}(\mathbf{B}_2)$.

1. $T = (\mathbf{B}_1, \mathbf{B}_2)$.
2. **for** $j = 1$ **to** $l = poly(\|\mathbf{B}_1\| + \|\mathbf{B}_2\|)$ **do**
   (a) old state $\leftarrow$ state(V^*)
   (b) repeat
      i. Pick uniformly $i \in \{1, 2\}$.
      ii. Pick uniformly Q'_j from $O(n, \mathbb{Z})$.
      iii. Compute $\mathbf{H}'_j \leftarrow \mathbf{HNF}(Q'_j\mathbf{B}_i)$.
      iv. Call V^* with input $\mathbf{H}'_j$ and obtain c'.
      v. **if** $i = c'$ **then**
         – Concatenate $\left(\mathbf{H}'_j, i, Q'_j\right)$ to the end of T.
         else
         – Set state(V^*) $\leftarrow$ old state.
      vi. until $i = c'$

Simulator S for $\mathbf{ILP}_{\mathbb{Z}}$.

Since V^* runs in polynomial time and that the probability $i = c'$ is $1/2$, on average S will generate two triples $\left(\mathbf{H}'_j, i, Q'_j\right)$ for every triple it concatenates to the transcript T and hence, the average running time of S is polynomial .

Using induction we will show that $\mathbf{Pr}_{V^*}(\mathcal{T}) = \mathbf{Pr}_S(\mathcal{T})$. Let $\mathbf{Pr}_{V^*}(\mathcal{T}_j)$ and $\mathbf{Pr}_S(\mathcal{T}_j)$ denote the probability distributions on the partial set of transcripts that could occur at the end of the j-th round.

Base case: If $j = 0$, then in both case $T = (\mathbf{H}_1, \mathbf{H}_2)$, hence both probabilities are identical.

Inductive Step: Suppose both distributions $\mathbf{Pr}_{V^*}(\mathcal{T}_{j-1})$ and $\mathbf{Pr}_S(\mathcal{T}_{j-1})$ are identical for some $j \geq 1$.

Now let's go back and see what happens at the j-th round of our interactive proof for $\mathbf{ILP}_{\mathbb{Z}}$. The probability that at this round V^* picks $c = 1$ is some number $0 \leq p \leq 1$ and the probability that $c = 2$ is $1 - p$. Moreover, the prover picks an orthogonal matrix Q' with probability $\frac{1}{2^n n!}$. This probability is independent of how the verifier picks $c \in \{1, 2\}$. Therefore the probability that at the j-th round $\left(\mathbf{H}'_j, i, Q'_j\right)$ is on the transcript of the **IP** if $c = 1$ is $\frac{p}{2^n n!}$ and if $c = 2$ is $\frac{1-p}{2^n n!}$

The simulator S in any round will pick an orthogonal matrix Q'_j with probability $\frac{1}{2^n n!}$. The probability that $i = 1$ and $c' = 1$ is $\frac{p}{2}$

and the probability $i = 2$ and $c' = 2$ is $\frac{1-p}{2}$.

In both cases the corresponding triple $\left(\mathbf{H}'_j, i, Q'_j\right)$ will be written to the transcript. Note with probability $1/2$ nothing is added to the transcript. The probability that $\left(\mathbf{H}'_j, 1, Q'_j\right)$ is written on the transcript in j-th round during the m-th iteration of the **repeat** loop is $\frac{p}{2^m \times (2^n n!)}$. Therefore the total probability that

$\left(\mathbf{H}'_j, 1, Q'_j\right)$ is written on the transcript in the j-th round is

$$\frac{p}{2 \times (2^n n!)} + \frac{p}{2^2 \times (2^n n!)} + ... + \frac{p}{2^m \times (2^n n!)} + ...$$
$$= \frac{p}{2 \times (2^n n!)} \left(1 + \frac{1}{2} + \frac{1}{4} + ... + \frac{1}{2^{m-1}} + ...\right) = \frac{p}{2^n n!}.$$

Similarly the total probability that $\left(\mathbf{H}'_j, 2, Q'_j\right)$ is written on the transcript in the j-th round is $\frac{1-p}{2^n n!}$. Hence, by induction, the two probability distributions are identical $\mathbf{Pr}_{V^*}(\mathcal{T}) = \mathbf{Pr}_S(\mathcal{T})$.

5 Sampling a Lattice Basis in Zero-Knowledge and ILP$_{\mathcal{R}}$

Suppose $\mathbf{B} \in \mathbb{R}^{n \times k}$ is a basis of some lattice $\mathcal{L}(\mathbf{B})$. Recall that $\mathbf{B}'$ is a basis of $\mathcal{L}(\mathbf{B})$ if and only if $\mathbf{B}' \in \{\mathbf{B}U : U \in GL_k(\mathbb{Z})\}$ and that the algorithm SampleD [2] takes an input basis $\mathbf{B} = [\mathbf{b}_1|\mathbf{b}_2|\ldots|\mathbf{b}_k] \in \mathbb{R}^{n \times k}$, an appropriate parameters $s \in \mathbb{R}$ and $\mathbf{c} \in \mathbb{R}^n$ and outputs a lattices point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ that is distributed according to the discrete Gaussian distribution $D_{s,\mathbf{c},\mathcal{L}}$ [2]. SampleD is zero-knowledge in a sense that the output point $\mathbf{v}$ leaks almost no information about the input basis $\mathbf{B}$ except the bound s with overwhelming probability [2]. Furthermore, for an n dimensional $\mathcal{L}$ if we pick $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n^2}\}$ lattice points independently according to $D_{s,\mathcal{L}}$, then $\mathbf{V}$ contain a subset of k linearly independent vectors, except with $negl(n)$ probability ([12], Corollary 3.16).

Let $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_k\}$ be a basis of a lattice $\mathcal{L}$ and suppose $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_k\}$ is a set of linearly independent vectors that belong to $\mathcal{L}$. There exists a deterministic polynomial time algorithm that will output a basis $\mathbf{T} = \{\mathbf{t}_1, \cdots, \mathbf{t}_k\}$ of $\mathcal{L}$ such that $||\mathbf{t}_i||_2 \leq ||\mathbf{s}_i||_2$ for $1 \leq i \leq k$ ([1], page 129).

Using the above two algorithms we will present a probabilistic polynomial time algorithm Sample$\mathcal{L}$ that will take an input basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ of some lattice $\mathcal{L}$, $\mathbf{c} \in \mathbb{R}^n$, a parameter $s \geq \omega(\sqrt{\log n}) \cdot ||\widetilde{\mathbf{B}}||$ and outputs a basis $\mathbf{T}$, such that $\mathbf{T}$ leaks no information about the basis $\mathbf{B}$, except s (the bound on the norm of $\mathbf{B}$) with overwhelming probability.

Protocol 1 Sample$\mathcal{L}$

Input $\left(\mathbf{B} \in \mathbb{R}_{\mathcal{R}}^{n \times k}, k, n, s\right)$

1. Sample $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n^2}\}$ points independently using the algorithm Sample$D(\mathbf{B}, 0, s)$).
2. Pick $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\} \subset \mathbf{V}$, such that $\mathbf{S}$ is a set of linearly independent vectors.
3. Using the deterministic algorithm output the basis $\mathbf{T}$, such that $\mathcal{L}(\mathbf{T}) = \mathcal{L}(\mathbf{B})$.

It is easy to see that if $\mathbf{B} \in \mathbb{R}_{\mathcal{R}}^{n \times k}$ then so $\mathbf{T} \in \mathbb{R}_{\mathcal{R}}^{n \times k}$. Since $\mathbf{T}$ and $\mathbf{B}$ are bases of the same lattice, there exists a $U \in GL_k(\mathbb{Z})$ such that

$$\mathbf{T} = \mathbf{B}U.$$

6 An Interactive Proof for $\mathbf{ILP}_{\mathcal{R}}$

– Input $\mathbf{B}_1, \mathbf{B}_2 \in \mathcal{R}^{n\times k}$ such that $\mathcal{L}(\mathbf{B}_1) \cong_{\mathcal{R}} \mathcal{L}(\mathbf{B}_2)$.
1. Prover set $s = \log n \cdot max\{\|\widetilde{\mathbf{B}}_1\|, \|\widetilde{\mathbf{B}}_2\|\}$.
2. for $i = 1$ to $l = poly(\|\mathbf{B}_1\| + \|\mathbf{B}_2\|)$ rounds do.
   (a) Prover picks uniformly an orthogonal matrix $Q'_j \leftarrow \overline{O(n, \mathcal{R})}$.
   (b) Prover picks $\mathbf{B}'_j \leftarrow \text{Sample}\mathcal{L}\left(Q'_j\mathbf{B}_1, k, n, s\right)$.
   (c) Prover sends the basis $\mathbf{B}'_j$ to the verifier.
   (d) Verifier randomly picks $c_j \in \{1, 2\}$ and sends it to the prover.
   (e) Prover sends the verifier an orthogonal matrix $P_j \in \overline{O(n, \mathcal{R})}$.
      i. if $c_j = 1$, then $P_j = Q'_j$.
      ii. if $c_j = 2$ then $P_j = Q'_jQ^{\mathbf{t}}$, where $Q \in \overline{O(n, \mathcal{R})}$ is such that $\mathcal{L}(\mathbf{B}_2) = \mathcal{L}(Q\mathbf{B}_1)$.
3. Verifier will accept the proof if for all l rounds $\mathcal{L}(\mathbf{B}) = \mathcal{L}(P_j\mathbf{B}_{c_j})$.

Theorem 4 *The proof system for* $\mathbf{ILP}_{\mathcal{R}}$ *is a statistical zero-knowledge interactive proof with an efficient prover.*

Proof:
Completeness: If $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are isometric lattices, then $\mathbf{B}_2 = Q\mathbf{B}_1U$ for some $Q \in \overline{O(n, \mathcal{R})}$ and $U \in GL_k(\mathbb{Z})$. Clearly,

$$\mathcal{L}(Q'_j\mathbf{B}_1) = \mathcal{L}(\mathbf{B}) = \mathcal{L}(Q'_jQ^{\mathbf{t}}\mathbf{B}_2)$$

since $\mathbf{B}'_j = Q'_j\mathbf{B}_1U'_j$ and $\mathbf{B}'_j = Q'_jQ^{\mathbf{t}}\mathbf{B}_2UU'_j$ for some $U'_j \in GL_k(\mathbb{Z})$. Therefore, the prover will always be able to convince the verifier.
Soundness: If $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are not isometric over $\mathcal{R}$, then the only way for the prover to deceive the verifier is for him to guess correctly c_j in each round. Since c_j is chosen uniformly from $\{1, 2\}$, the probability of the prover guessing c_j in all rounds is 2^{-l}. Hence, the protocol is sound.
Efficient Prover: Clearly the prover can perform steps 1, $2a$, $2c$ and $2e$ in expected polynomial-time. In step $2b$ the prover picks a lattice basis using Sample$\mathcal{L}$, which runs in expected polynomial time. Hence the total expected running time of the prover is polynomial.
Zero-Knowledge: Let V^* be any probabilistic polynomial time (possibly malicious) verifier. Let $\mathcal{T}(V^*)$ denote the set of all possible transcripts that could be produced as a result of P and V^* carrying out the interactive proof on a **yes** instance $(\mathbf{B}_1, \mathbf{B}_2)$ of $\mathbf{ILP}_{\mathcal{R}}$. Let $S_{\mathcal{R}}$ denote the simulator, which will produce the possible set of forged transcripts $\mathcal{T}(S_{\mathcal{R}})$. We denote $\mathbf{Pr}_{V^*}(\mathcal{T})$ the probability distribution on $\mathcal{T}(V^*)$ and we denote $\mathbf{Pr}_{S_{\mathcal{R}}}(\mathcal{T})$ the probability distribution on $\mathcal{T}(S_{\mathcal{R}})$. We will prove that:

1. $S_{\mathcal{R}}$ is polynomial.
2. $\mathbf{Pr}_{V^*}(\mathcal{T}) \sim \mathbf{Pr}_{S_{\mathcal{R}}}(\mathcal{T})$ i.e the two distributions are statistically close.

Input: $\mathbf{B}_1, \mathbf{B}_2 \in \mathcal{R}^{n \times k}$ such that $\mathcal{L}(\mathbf{B}_1) \cong_{\mathcal{R}} \mathcal{L}(\mathbf{B}_2)$.

1. Set $s = \log n \cdot max\{\|\widetilde{\mathbf{B}}_1\|, \|\widetilde{\mathbf{B}}_2\|\}$.
2. $T = (\mathbf{B}_1, \mathbf{B}_2)$.
3. **for** $j = 1$ **to** $l = poly(\|\mathbf{B}_1\| + \|\mathbf{B}_2\|)$ **do**
   (a) old state $\leftarrow$ state(V^*)
   (b) repeat
      i. Pick uniformly $i_j \in \{1, 2\}$.
      ii. Pick uniformly Q'_j from $\in \overline{O(n, \mathcal{R})}$.
      iii. Compute $\mathbf{H}'_j \leftarrow \text{Sample}\mathcal{L}\left(Q'_j \mathbf{B}_{i_j}, k, n, s\right)$.
      iv. Call V^* with $\mathbf{H}'_j$ and obtain i'.
      v. **if** $i_j = i'$ **then**
         – Concatenate $\left(\mathbf{H}'_j, i_j, Q'_j\right)$ to the end of T.
      else
         – Set state(V^*) $\leftarrow$ old state.
      vi. until $i_j = i'$.

Simulator $S_{\mathcal{R}}$ for $\mathbf{ILP}_{\mathcal{R}}$.

Running time of the simulator : What is the probability that $i_j = i'$? In other words, on average how many triples $\left(\mathbf{H}'_j, i_j, Q'_j\right)$ will the simulator $S_{\mathcal{R}}$ generate for every triple it concatenates to T? We note that $Q'Q^{\mathbf{t}}$ and Q' are uniformly distributed over $\overline{O(n, \mathcal{R})}$, and $\mathcal{L}(Q'\mathbf{B}_1) = \mathcal{L}(Q'Q^{\mathbf{t}}\mathbf{B}_2)$ therefore the probability that the lattice $\mathcal{L}(\mathbf{H}'_j)$ is obtained by rotating the lattice $\mathcal{L}(\mathbf{B}_1)$ is equal to the probability that it is obtain by rotating $\mathcal{L}(\mathbf{B}_2)$. Furthermore the algorithm Sample$\mathcal{L}$ ensures that as far as the parameters are chosen appropriately, $\mathbf{H}'_j$ will leak almost no information (apart from the bound s) about the input basis except with negligible probability. Hence, on the average the simulator will generate roughly 2 triples for every triple it adds to T. Therefore the expected running time of $S_{\mathcal{R}}$ is roughly twice the running time of V^*. By definition V^* runs in probabilistic polynomial time. Hence the running time of $S_{\mathcal{R}}$ is also expected polynomial time.

We will prove that the two probability distributions $\mathbf{Pr}_{V^*}(\mathcal{T})$ and $\mathbf{Pr}_{S_{\mathcal{R}}}(\mathcal{T})$ are statistically close as follows. We first prove that the two distributions are statistically close for one round ($l = 1$). Then we will invoke the sequential composition lemma 4.3.11 on page 216 of [9], which implies that an interactive proof which is zero-knowledge for one round remains zero-knowledge for polynomially many rounds.

Case $l = 1$: Let $(\mathbf{B}'_1, c_1, P'_1)$ denote a transcript produced as a result of an interactive proof and $(\mathbf{H}'_1, i_1, Q'_1)$ denote a transcript produced by the simulator. In the interactive proof P picks uniformly P'_1 over $\overline{O(n, \mathcal{R})}$ and $S_{\mathcal{R}}$ also picks Q'_1 uniformly over $\overline{O(n, \mathcal{R})}$. Hence both P'_1 and Q'_1 are identically distributed. Also $\mathbf{B}'_1$ and $\mathbf{H}'_1$ are computed by Sample$\mathcal{L}$. Therefore they are almost identically distributed to $D_{s,\mathbf{c},\mathcal{L}}$ and thus to each other.

Let p be the probability that V^* picks $c_1 = 1$ and $1 - p$ be the probability that it picks $c_1 = 2$ in the interactive proof. The probability may depend on the state of V^*. The simulator picks $i_1 \in \{1, 2\}$ uniformly and independent of

how V^* picks i'. Also given $\mathbf{H}'_1$, the probability that V^* can guess the index i_1 is at most $\frac{1}{2} + negl$. Therefore probability that V^* picks $i' = 1$ is nearly p and $i' = 2$ is nearly $1 - p$ respectively. This means that i_1 and c_1 have nearly the same distributions.

Therefore, it follows that $(\mathbf{B}'_1, c_1, P'_1)$ and $(\mathbf{H}'_1, i_1, Q'_1)$ are statistically close. Hence for one round the two distributions are statistically close. Hence, by lemma 4.3.11 for any polynomially many rounds we have $\mathbf{Pr}_{V^*}(\mathcal{T}) \sim \mathbf{Pr}_{S_{\mathcal{R}}}(\mathcal{T})$.

7 Isometric Lattice Problem is not Easy

In this section we will show that $\mathbf{ILP}_{\mathbb{Z}}$ is at least as hard as Linear Code Equivalence problem over prime fields $\mathbb{F}_p$ and Graph Isomorphism.

Theorem 5 $\mathbf{ILP}_{\mathbb{Z}}$ *is at least as hard as* **LCE** (*Linear Code Equivalence problem*) *over prime fields* $\mathbb{F}_p$.

Proof Let $\mathbf{G} = [\mathbf{g}_1|\dots|\mathbf{g}_k] \in \mathbb{F}_p^{n\times k}$ be a basis of some $[k, n]$ linear code C

$$\psi : C \longrightarrow \Lambda_2(\mathbf{G}); \quad \mathbf{G} \longrightarrow \mathbf{B}$$

where $\Lambda_2(\mathbf{G})$ be the corresponding p-ary lattice. Recall from section 2 that $\mathbf{B} = [\mathbf{g}_1|\dots|\mathbf{g}_k|\mathbf{b}_{k+1}|\dots|\mathbf{b}_n] \in \mathbb{Z}^{n\times n}$ is a basis of $\Lambda_p(\mathbf{G})$. Where $\mathbf{b}_j = (0, ..., p, ..., 0) \in \mathbb{Z}^n$ and the j-th coordinate is equal to p, for $k + 1 \le j \le n$. Clearly the map ψ can be computed in polynomial time. Let $\mathbf{G}_1 = [\mathbf{g}_{11}|\dots|\mathbf{g}_{1k}] \in \mathbb{F}_p^{n\times k}$ and $\mathbf{G}_2 = [\mathbf{g}_{21}|\dots|\mathbf{g}_{2k}] \in \mathbb{F}_p^{n\times k}$ be two code generators.

$\Longrightarrow$ Suppose $\mathbf{G}_1$ and $\mathbf{G}_2$ generate linearly equivalent codes i.e $\mathbf{G}_2 = P\mathbf{G}_1M$ for $M \in GL_k(\mathbb{F}_p)$ and monomial matrix $P' \in \mathcal{P}(n, \mathbb{F}_q)$. Note that we can write P' as a product of a permutation matrix $P \in \mathcal{P}_n$ and an invertible diagonal matrix $D \in \mathbb{F}_p^{n\times k}$. Write $\mathbf{G}_2 = P\mathbf{G}'_1M$, where $\mathbf{G}'_1 = D\mathbf{G}_1$ and let $\Lambda_p(\mathbf{G}'_1)$ and $\Lambda_p(\mathbf{G}_2)$ be corresponding lattices.

$$\begin{gathered}\text{For any } \mathbf{v} \in \Lambda_p(\mathbf{G}_2) \Longleftrightarrow \mathbf{v} \equiv \mathbf{G}_2 \cdot \mathbf{s} \pmod p, \text{ for some } \mathbf{s} \in \mathbb{Z}^k \\ \Longrightarrow \mathbf{v} \equiv P\mathbf{G}'_1M \cdot \mathbf{s} \pmod p \equiv P\mathbf{G}'_1 \cdot \mathbf{s}' \pmod p,\ \mathbf{s}' = M\mathbf{s} \in \mathbb{Z}^k \\ \Longrightarrow \mathbf{v} \in \Lambda_p(P\mathbf{G}'_1)\end{gathered}$$

Hence, $\Lambda_p(\mathbf{G}_2) \subseteq \Lambda_p(P\mathbf{G}'_1)$. Since, $P\mathbf{G}'_1 = \mathbf{G}_2M^{-1}$, by the same argument $\Lambda_p(P\mathbf{G}'_1) \subseteq \Lambda_p(\mathbf{G}_2)$, we have $\Lambda_p(P\mathbf{G}'_1) = \Lambda_p(\mathbf{G}_2)$. Therefore, there exists a $U \in GL_k(\mathbb{Z})$ such that

$$\psi(\mathbf{G}_2) = \psi(P\mathbf{G}'_1)U = P\psi(\mathbf{G}'_1)U$$

$\Longleftarrow$ Now suppose $\mathbf{G}_1$ and $\mathbf{G}_2$ are not linearly equivalent and suppose $\psi(\mathbf{G}_2) = Q\psi(\mathbf{G}_1)U$ for $Q \in O(n, \mathbb{Z})$ and $U \in GL_k(\mathbb{Z})$. Note we can write any $Q \in O(n, \mathbb{Z})$ as $Q = PD_\epsilon$, for some $D_\epsilon \in \mathcal{D}_{\epsilon_n}$ and $P \in \mathcal{P}_n$. But $P' = PD_\epsilon \pmod p$ is a monomial matrix. Further U is also non-singular over $\mathbb{F}_p$. Therefore, $\psi(\mathbf{G}_2) = Q\psi(\mathbf{G}_1)U$, which implies

$$\mathbf{G}_2 = P'(\mathbf{G}_1)M \mod p \text{ for some } M \in GL_k(\mathbb{F}_p) \text{ and } M \equiv U \pmod p$$

This contradicts the assumption that $\mathbf{G}_1$ and $\mathbf{G}_2$ are not linearly equivalent. Therefore $\mathbf{ILP}_{\mathbb{Z}}$ is at least as hard as **LCE**.

Theorem 6 $\mathbf{ILP}_{\mathbb{Z}}$ *is at least as hard as the* **GI** (*Graph Isomorphism*) *problem.*

Proof Petrank and Roth [20] reduced **GI** to **PCE** (Permutation Code Equivalence). More precisely they provided a polynomial time mapping ϕ from the set of all graphs to the set of generator matrices over $\mathbb{F}_2$ such that two graphs G_1 and G_2 are isomorphic if and only if $\phi(G_1)$ and $\phi(G_2)$ are permutation equivalent codes. We will prove that **ILP** is at least as hard as **GI**, by reducing the **PCE** over $\mathbb{F}_2$ to **ILP**. Let $\mathbf{G} = [\mathbf{g}_1|\ldots|\mathbf{g}_k] \in \mathbb{F}_2^{n\times k}$ be a basis of some $[k, n]$ linear code C

$$\psi : C \longrightarrow \Lambda_2(\mathbf{G}); \quad \mathbf{G} \longrightarrow \mathbf{B}$$

where $\Lambda_2(\mathbf{G})$ is the corresponding 2-ary lattice. Recall from section 2 that $\mathbf{B} = [\mathbf{g}_1|\ldots|\mathbf{g}_k|\mathbf{b}_{k+1}|\ldots|\mathbf{b}_n] \in \mathbb{Z}^{n\times n}$ is a basis of $\Lambda_2(\mathbf{G})$. Where $\mathbf{b}_j = (0, ..., 2, ..., 0) \in \mathbb{Z}^n$ and the j-th coordinate is equal to 2, for $k + 1 \leq j \leq n$. Clearly the map ψ can be computed in polynomial time. Let $\mathbf{G}_1 = [\mathbf{g}_{11}|\ldots|\mathbf{g}_{1k}] \in \mathbb{F}_2^{n\times k}$ and $\mathbf{G}_2 = [\mathbf{g}_{21}|\ldots|\mathbf{g}_{2k}] \in \mathbb{F}_2^{n\times k}$ be two code generators and $\Lambda_2(\mathbf{G}_1)$ and $\Lambda_2(\mathbf{G}_2)$ be corresponding lattices.

$\Longrightarrow$) Suppose $\mathbf{G}_1$ and $\mathbf{G}_2$ are permutation equivalent i.e. $\mathbf{G}_2 = P\mathbf{G}_1M$ for $M \in GL_k(\mathbb{F}_2)$ and $P \in \mathcal{P}_n$. Let $\mathbf{G}'_1 = P\mathbf{G}_1$. Therefore we can write $\mathbf{G}_2 = \mathbf{G}'_1M$. By definition for any $\mathbf{v} \in \Lambda_2(\mathbf{G}_2)$, there exists an $\mathbf{s} \in \mathbb{Z}^k$ such that

$$\mathbf{v} \equiv \mathbf{G}_2 \cdot \mathbf{s} \equiv \mathbf{G}'_1M \cdot \mathbf{s} \pmod 2.$$
$$\Longrightarrow \mathbf{v} \equiv P\mathbf{G}_1 \cdot \mathbf{s}' \pmod 2, \text{ where } \mathbf{s}' = M \cdot \mathbf{s} \in \mathbb{Z}^k \Longrightarrow \mathbf{v} \in \Lambda_2(P\mathbf{G}_1).$$

Hence, $\Lambda_2(\mathbf{G}_2) \subseteq \Lambda_2(P\mathbf{G}_1)$. Since, $P^{\mathbf{t}}\mathbf{G}_2M^{-1} = \mathbf{G}_1$ by the same argument $\Lambda_2(P\mathbf{G}_1) \subseteq \Lambda_2(\mathbf{G}_2)$. Hence, there exist a $U \in GL_k(\mathbb{Z})$ such that

$$\psi(\mathbf{G}_2) = \psi(P\mathbf{G}_1)U \Longrightarrow \mathbf{B}_2 = P\mathbf{B}_1U$$

$\Longleftarrow$) Now suppose $\mathbf{G}_1$ and $\mathbf{G}_2$ are not permutation equivalent and suppose $\psi(\mathbf{G}_2) = Q\psi(\mathbf{G}_1)U$ for $Q \in O(n, \mathbb{Z})$ and $U \in GL_k(\mathbb{Z})$. Note that $Q \equiv P \pmod 2$, for some $P \in \mathcal{P}_n$. For every $\mathbf{v} \in \Lambda_2(\mathbf{G}_2)$ we have

$$\mathbf{v} \equiv \mathbf{G}_2\mathbf{u} \pmod 2 \text{ for some } \mathbf{u} \in \mathbb{Z}^k.$$

Since, $\Lambda_2(\mathrm{Q}\mathbf{G}_1) = \Lambda_2(\mathbf{G}_2)$, we also have $\mathbf{v} \equiv (Q\mathbf{G}_1)\mathbf{u} \equiv (P\mathbf{G}_1)\mathbf{u} \pmod 2$ for some $\mathbf{u} \in \mathbb{Z}^k$. This means that $P\mathbf{G}_1$ and $\mathbf{G}_2$ have the same span over $\mathbb{F}_2$. This contradicts the assumption that $\mathbf{G}_1$ and $\mathbf{G}_2$ are not permutation equivalent. This proves that **ILP** is at least as hard as **GI**.

7.1 ILP is unlikely to be NP-complete

In this sub-section we show that $\mathbf{ILP}_{\mathbb{S}}$ is unlikely to be NP-complete (where $\mathbb{S} = \mathbb{Z}$ or $\mathbb{S} = \mathcal{R}$ see section 3.1). We do this by constructing a constant round interactive proof for the **Non-Isometric Lattice problem** (**co-ILP**$_{\mathbb{S}}$), i.e. the complementary problem of $\mathbf{ILP}_{\mathbb{S}}$. Then we invoke results from the field of complexity theory, implying that if the complement of a problem Π has a constant round interactive proof and Π is NP-complete then the polynomial hierarchy collapses [18, 19]. It is widely believed that the polynomial hierarchy does not collapse, therefore we end up with the conclusion that **ILP** is unlikely to be NP-complete.

Constant Round IP for co-ILP$_{\mathbb{S}}$

– Input $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{S}^{n \times k}$ bases such that $\mathcal{L}(\mathbf{B}_1) \ncong_{\mathbb{S}} \mathcal{L}(\mathbf{B}_2)$.

1. Verifier sets $l = poly(|\mathbf{B}_1| + |\mathbf{B}_2|)$.
2. Verifier picks uniformly $j_1, \ldots, j_l \in \{1, 2\}$.
3. If $\mathbb{S} = \mathbb{Z}$ then the verifier picks independent random orthogonal matrices
$$Q_1, \ldots, Q_l \in O(n, \mathbb{Z}).$$
Else verifier picks independently random orthogonal matrices
$$Q_1, \ldots, Q_l \in \overline{O(n, \mathcal{R})}.$$
4. For $1 \leq i \leq l$, verifier computes a basis $\mathbf{H}'_i$ for the lattice $\mathcal{L}(Q_i \mathbf{B}_{j_i})$. If $\mathbb{S} = \mathbb{Z}$, then $\mathbf{H}'_i \leftarrow \mathbf{HNF}(Q_i \mathbf{B}_{j_i})$, otherwise $\mathbf{H}'_i$ is computed using algorithm Sample$\mathcal{L}$ from section 5.
5. For $1 \leq i \leq l$, the all-powerful prover computes and sends j'_i such that $\mathbf{H}'_i$ and $\mathbf{B}_{j'_i}$ are isometric.
6. Verifier accepts the proof if $j_i = j'_i$ for all $1 \leq i \leq l$.

Completeness: Clearly, if $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are non-isometric lattices then the prover will never fail convincing the verifier.
Soundness: Suppose $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$ are isometric lattices. The probability that prover can guess $(i_1, ..., i_l)$ given $(\mathbf{H}'_1, ..., \mathbf{H}'_l)$ is 2^{-l} if $\mathbb{S} = \mathbb{Z}$ and $2^{-l} + negl$ if $\mathbb{S} = \mathcal{R}$.

8 Conclusion and Acknowledgement

We conclude with an open problem related to our work. Construct a Malicious verifier statistical zero-knowledge proof system with an efficient prover for the Isometric Lattice Problem over rationals $\mathbf{ILP}_{\mathbb{Q}}$. We would also like to thank Professor Chris Peikert, for his help and patience, who always took time out of his busy schedule to answer our questions.

References

1. Daniele Micciancio and Shafi Goldwasser *Complexity of Lattice Problems: A Cryptographic Perspective.* Springer International Series in Engineering and Computer Science, volume 671, March 2002.

2. Craig Gentry, Chris Peikert and Vinod Vaikuntanathan. *How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions.* In STOC, pages 197 – 206 (2008).
3. David Cash, Dennis Hofheinz, Eike Kiltz and Chris Peikert. *Bonsai trees, or how to delegate a lattice basis.* In EUROCRYPT, pages 523 – 552 (2010).
4. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness* . W. H. Freeman & Company, 1990.
5. Raza Ali Kazmi *Cryptography from Post-Quantum Assumptions.* PhD Thesis, School of Computer Science, McGill University, 2015. Supervised by Claude Crépeau. https://eprint.iacr.org/2015/376.
6. Shafi Goldwasser, Silvio Micali and Charles Rackoff. *The knowledge Complexity of Interactive Proof Systems.* SIAM Journal on Computing, volume 18, issue 1, pages 186 – 208 (1989).
7. Nicolas Sendrier and Dimitris E. Simos. *The Hardness of Code Equivalence over* $\mathbb{F}_q$ *and Its Application to Code-Based Cryptography.* In Post-Quantum Cryptography, volume 7932 of LNCS, pages 203-216. Springer, 2013.
8. Oded Goldreich, Silvio Micali, and Avi Wigderson. *How to play any mental game or a completeness theorem for protocols with honest majority.* In STOC, pages 218 – 229 (1987).
9. Oded Goldreich. *Foundations of Cryptography.* Volume I & II. Cambridge University Press, 2001 – 2004.
10. Daniele Micciancio and Salil P. Vadhan. *Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More.* In CRYPTO, pages 282 – 298 (2003).
11. Christopher Peikert and Vinod Vaikuntanathan. *Noninteractive Statistical Zero-Knowledge Proofs for Lattice Problems.* In CRYPTO, pages 17 – 21 (2008).
12. Oded Regev. *On lattices, learning with errors, random linear codes, and cryptography.* In STOC, pages 84 – 93 (2005).
13. Oded Goldreich and Shafi Goldwasser. *On the Limits of Non-Approximability of Lattice Problems.* In STOC, pages 23 – 26 (1998).
14. G.W. Stewart. *The Efficient Generation of Random Orthogonal Matrices with an Application to Condition.* SIAM Journal on Numerical Analysis, volume 17 Number 3 (1980).
15. George Marsaglia. *Choosing a Point from the Surface of a Sphere.* Annals of Mathematical Statistics, volume 43, number 2, pages 645 – 647 (1972).
16. Eric Schmutz. *Rational points on the unit sphere.* Central European Journal of Mathematics, volume 6, issue 3, pages 482 – 487 (2008).
17. Clément Pernet and William Stein. *Fast Computation of Hermite Normal Forms of Random Integer Matrices.* Journal of Number Theory, volume 130, issue 7, pages 1675 – 1683 (2010).
18. Shafi Goldwasser and Michael Sipser. *Private coins versus public coins in interactive proof systems.* In STOC, pages 59 – 68 (1986).
19. Ravi B. Boppana, Johan Håstad and Stathis Zachos. *Does co-NP have short interactive proofs?* Journal of Information Processing Letters, volume 25, issue 2, pages 127 – 132 (1987).
20. Erez Petrank and Ron M. Roth. *Is code equivalence easy to decide?* IEEE Transactions on Information Theory, volume 43, issue 5, pages 1602 – 1604 (1997).
21. Davis Cash, Dennis Hofheinz, Eike Kiltz and Christopher Peikert. *Bonsai Trees, or How to Delegate a Lattice Basis.* In EUROCRYPT, pages 523 – 553 (2010).
22. Daniele Micciancio and Christopher Peikert. *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller.* In EUROCRYPT, pages 700 – 718 (2012).

23. Oded Goldreich, Amit Sahai and Salil Vadhan. *Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge*. In STOC, pages 399 – 408 (1998).
24. Tanner and Thisted. *Applied Statistics*, pages 199 – 206 (1982).
25. Hans Liebeck and Anthony Osborne. *The generation of all rational orthogonal matrices*. American Mathematical Monthly, volume 98, issue 2, pages 131 – 133 (1991).
26. Daniel J Bernstein, Johannes A. Buchmann and Erik Dahmen. *Post-Quantum Cryptography*. Number Theory and Discrete Mathematics, Springer, ISBN 978-3-540-88701-0 (2008).
27. Michael Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co, ISBN 0-7167-1045-5 (1990).
28. Robert J. McEliece. *A public-key cryptosystem based on algebraic coding theory*. Technical memo, California Institute of Technology (1978).

A Computing sine and cosine efficiently

Let $p(n)$ be any desired publicly known positive polynomial. Recall that

$$\sin\left(\frac{\pi}{2^{p(n)}}\right) = \frac{1}{2}\underbrace{< \frac{1}{2}, 2- < \frac{1}{2}, 2+\cdots+ < \frac{1}{2}, 2 >> \cdots >}_{p(n)-1}$$

$$\cos\left(\frac{\pi}{2^{p(n)}}\right) = \frac{1}{2}\underbrace{< \frac{1}{2}, 2+ < \frac{1}{2}, 2+\cdots+ < \frac{1}{2}, 2 >> \cdots >}_{p(n)-1}.$$

Suppose we have to compute $\sin\left(\frac{l\cdot\pi}{2^{p(n)}}\right)$ for some $0 \leq l \leq 2^{p(n)}$.

$$\sin(\alpha+\beta) = \sin(\alpha)\cos(\beta) + \sin(\beta)\cos(\alpha)$$
$$\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$$

Write $l = \sum_{i=0}^{k} x_i \cdot 2^i$, $x_i \in \{0,1\}$ and $k \leq p(n)$. WLOG we can assume that l is not even.

$$\sin\left(\tfrac{l\cdot\pi}{2^{p(n)}}\right) = \sin\left(\tfrac{\pi}{2^{p(n)-k}} + \cdots + \tfrac{\pi}{2^{p(n)}}\right)$$
$$= \sin\left(\tfrac{\pi}{2^{p(n)-k}}\right)\cos\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right) + \sin\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right)\cos\left(\tfrac{\pi}{2^{p(n)-k}}\right).$$

Note that $\sin\left(\frac{\pi}{2^{p(n)-k}}\right)$ and $\cos\left(\frac{\pi}{2^{p(n)-k}}\right)$ can be computed directly. Now we can recursively compute $\cos\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right)$ and $\sin\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right)$. But since $\sin(\theta)^2 = 1-\cos^2(\theta)$, in recursion we will only have to compute either $\cos\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right)$ or $\sin\left(\frac{\left[\sum_{i=0}^{k-1} x_i 2^i\right]\pi}{2^{p(n)}}\right)$.

Clearly depth of the recursion is $k \leq p(n)$ and for each recursive step we will have four values, with each value is of size $O(p(n))$. Hence in total running time is at most $O(p(n))$ operations. Similarly, one can show that $\cos\left(\frac{l\cdot\pi}{2^{p(n)}}\right)$ for any $0 \leq l \leq 2^{p(n)}$, can be computed in polynomial time as well.