# Cryptography from Post-Quantum Assumptions

Raza Ali Kazmi

Doctor of Philosophy

School of Computer Science

McGill University

Montreal, Quebec

2015-04-07

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

**Abstract**

In this thesis we present our contribution in the field of post-quantum cryptography. We introduce a new notion of *weakly Random-Self-Reducible* public-key cryptosystem and show how it can be used to implement secure Oblivious Transfer. We also show that two recent (Post-quantum) cryptosystems can be considered as *weakly Random-Self-Reducible*. We introduce a new problem called Isometric Lattice Problem and reduce graph isomorphism and linear code equivalence to this problem. We also show that this problem has a perfect zero-knowledge interactive proof with respect to a malicious verifier; this is the only hard problem in lattices that is known to have this property.

**Résumé**

Dans cette thèse nous exposons nos contributions au domaine de la cryptographie post-quantique. Nous présentons d'abord la nouvelle notion de système cryptographique aléatoirement-auto-réductible au sens faible et démontrons comment elle peut être utilisée afin d'obtenir une version sécurisée du transfert inconscient. Nous démontrons ensuite que deux systèmes cryptographiques (post-quantiques) récents peuvent être considérés comme exemples de systèmes aléatoirement-auto-réductibles au sens faible. De plus, nous présentons un nouveau problème cryptographique de « treillis isométriques » auquel nous réduisons le problème d'« isomorphisme de graphes » et celui d'« équivalence de codes linéaires ». Nous montrons enfin que ce nouveau problème possède une preuve interactive à connaissance nulle parfaite par rapport à tous les vérificateurs (malveillants) ; celui-ci est le seul problème de treillis connu possédant une telle propriété.

## ACKNOWLEDGEMENTS

I wish to thank, first and foremost, my advisor Professor Claude Crepéau, who put his faith in me as his graduate student. I thank him for his guidance, support and patience. Indeed, without his guidance, I would not be have been able to finish my thesis. Merci Claude.

I would also like to thank Professor Christopher Jason Peikert, for his help and patience, who always took time out of his busy schedule to answer my queries.

Last but not least, I would like to thank my mother Shadab Khatoon and my wife Sahr Kazmi for their unconditional support throughout my degree.

TABLE OF CONTENTS

# Chapter 1
# Introduction

The security of many known public-key cryptosystems is based on the presumed hardness of factoring a composite number or computing a discrete log of a cyclic group (e.g RSA [2], ElGamal encryption [3] etc). Both of these problems can be solved in polynomial time on a quantum computer [1]. Thus, most of the current cryptographic primitives will become insecure once quantum computers are built. **Post-quantum cryptography** refers to research on cryptographic primitives that are believed to be secure against quantum attacks. Currently Post-quantum cryptography is mostly focused on three different approaches:

1. Lattice based cryptography.
2. Code based cryptography.
3. Hash based digital signature schemes.

Lattice based cryptography is considered as one of the most viable options in the Post-quantum world. First of all it is extremely versatile and rich, leading to a large number of applications ranging from public-key encryption schemes [6, 14], digital signature schemes [7, 7, 8, 9], fully homomorphic encryption schemes [10, 13, 15, 16, 17, 18, 20], hierarchical identity-based encryption [21, 22, 23], zero-knowledge proofs [30] to collision-resistant hash functions [24]. From the security point of view, the best attacks on the underlying problems run in exponential time $2^{\Omega(n)}$, in the security

parameter $n$, for both classical and quantum algorithms. Moreover, lattice based cryptography has very strong security proofs based on worst-case hardness. From the point of view of efficiency and implementation, the lattice based cryptographic primitives are relatively simple and efficient to implement.

### 1.1 Our Contribution

Our contribution is mainly in the area of Lattice based cryptography. More precisely,

- We formalize the results of [25], which relied on the *Random-Self-Reducible* encryption (**RSR**) property of certain number theoretic assumptions in order to introduce a new notion of *weakly Random-Self-Reducible* encryption scheme (**wRSR**). We then show how it is possible to construct a secure Oblivious Transfer under the sole assumption that a secure **wRSR** encryption scheme exists. We then show that two recent (Post-quantum) computational assumptions have [13, 14] this weak property.
- We propose a new hard problem in lattices called *Isometric Lattice problem* (**ILP**). We provide interactive proof systems for **ILP**. These proofs are malicious verifier perfect zero-knowledge and have efficient provers. We reduce graph isomorphism (**GI**) and linear code equivalence (**LCE**) to this problem. We also show that **ILP** cannot be NP-complete, unless polynomial hierarchy collapses. We do this by constructing a constant round interactive proof systems for **coILP**, the complementary problem of **ILP**.

## 1.2 Organization of the Thesis

The thesis is as self-contained as possible. Chapter 2 presents the notations and background material required in order to understand the remaining chapters. Chapter 3 introduces the notion of Oblivious Transfer and a formal definition of *Random-Self-Reducible*. The notion of *weakly Random-Self-Reducible* encryption schemes, concrete examples of **wRSR** is also the subject of chapter 3. In this chapter we also show that how one can obtain a secure Oblivious Transfer under the sole assumption that **wRSR** scheme exists. Chapter 4 introduces a new hard problem in lattices. The reduction from **GI** and **LGE** to **ILP** and Zero-Knowledge proofs are also part of this chapter. The summary is given in Chapter 5.

# Chapter 2
# Preliminaries

In this chapter we introduce the notation that we will use throughout the thesis, we also provide some background material and definitions from lattices, coding theory and cryptography.

## 2.1 Notations

- For any matrix $\mathbf{A}$, we denote its transpose by $\mathbf{A^t}$.
- $O(n, \mathbb{R}) = \{Q \in \mathbb{R}^{n \times n} : Q \cdot Q^{\mathbf{t}} = \mathbf{I}\}$ denote the group of $n \times n$ orthogonal matrices over $\mathbb{R}$.
- $e = \sum_{n=0}^{\infty} \frac{1}{n!}$ denote the base of the natural logarithm.
- $GL_k(\mathbb{Z})$ denote the group of $k \times k$ invertible (unimodular) matrices over the integers.
- For any $a \in \mathbb{R}$ we denote the nearest integer to $a$ by $\lfloor a \rceil$.
- $GL_k(\mathbb{F}_q)$ denote the set of $k \times k$ invertible matrices over the finite field $\mathbb{F}_q$.
- $\mathcal{P}_n$ denote the set of $n \times n$ permutation matrices.
- $\sigma_n$ is the set of all permutations of $\{1, \ldots, n\}$. For $\pi \in \sigma_n$, we denote $P_\pi$ the corresponding $n \times n$ permutation matrix.
- $\mathcal{P}(n, \mathbb{F}_q)$ denote 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 $i = 1, \ldots, n$.

**Negligible Function**

A function $negl : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$ is *negligible*, if for any positive polynomial $p(n)$, there exists a positive integer $n_0$ such that for all $n > n_0$

$$negl(n) < \frac{1}{p(n)}.$$

**One-Way function & Hard-Core Predicate**

A function $f : \{1,0\}^* \to \{1,0\}^*$ is one-way if

1. (Easy to compute:) If $f$ can be computed in polynomial time in the size of the input.
2. (Hard to invert:) For every probabilistic polynomial-time algorithm $\mathcal{A}$, every positive polynomial $p(n)$ and for uniformly picked $x \in \{0,1\}^n$

   $$\mathbf{Pr}[f(\mathcal{A}(f(x))) = f(x)] < \frac{1}{p(n)}$$

   i.e. given $f(x)$ it is computationally hard (on average) to recover $x$.

A function $b : \{0,1\}^* \to \{0,1\}$ is a hardcore predicate of a one-way function $f$ if $b$ can be computed in polynomial time, and for every probabilistic polynomial time algorithm $\mathcal{A}$, and uniformly distributed $x \in \{0,1\}^n$ and given $f(x)$ the probability

$$\mathbf{Pr}[\mathcal{A}(f(x)) = b(x)] \leq \frac{1}{2} + negl(n).$$

In simpler words there is no probabilistic polynomial-time algorithm that computes $b(x)$ from $f(x)$ with probability significantly greater than one half over random choice

of $x$. Goldreich and Levin proved that given any one-way function $f$ we can construct a different one-way function $g$ with a hardcore predicate $b_g$ for $g$.

### 2.2 Norms

For a real vector $\mathbf{v} = (v_1, \ldots, v_n)$ we denote its Euclidean norm by $\|\mathbf{v}\|$

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \cdots + v_n^2}$$

and max-norm $\|\mathbf{v}\|_\infty$

$$\|\mathbf{v}\| = max_{i=1}^n |v_i|.$$

We denote the norm of a matrix $\mathbf{B} = [\mathbf{b}_1|\mathbf{b}_2|\ldots|\mathbf{b}_k] \in \mathbb{R}^{n\times k}$

$$\|\mathbf{B}\| = max_{i=1}^n \|\mathbf{b}_i\|.$$

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. Note that $max_{i=1}^n \|\tilde{\mathbf{b}}_i\| \leq max_{i=1}^n \|\mathbf{b}_i\|$.

### 2.3 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 $\mathbf{B}_1 = \mathbf{UB}_2$. The *dual* of a Lattice $\mathcal{L}(\mathbf{B})$, denoted as $\mathcal{L}(\mathbf{B})^*$ is the set of all vectors $\mathbf{v} \in span(\mathbf{B})$ such that $\langle \mathbf{v}, \mathbf{u} \rangle \in \mathbb{Z}$ for all $\mathbf{u} \in \mathcal{L}(\mathbf{B})$. Note that $\mathcal{L}(\mathbf{B})^*$ is also a lattice, with basis $\mathbf{B}^* = \mathbf{B}(\mathbf{B^t B})^{-1}$. The determinant of a lattice $\mathcal{L}(\mathbf{B})$, denoted as $\det(\mathcal{L}(\mathbf{B})) = \sqrt{\det(\mathbf{B^t B})}$, is the volume

of the fundamental parallelepiped $\{\mathbf{Bx} : 0 \leq x_i < 1\}$. The determinant is lattice invariant, i.e., it does not depend on the particular basis used to compute it.

**Hermite Normal Form**

Let $\mathbf{B} \in \mathbb{Z}^{k \times n}$ and $[\mathbf{B}] = \{U\mathbf{B} : U \in GL_k(\mathbb{Z})\}$. There exists a unique $k \times n$ matrix $\mathbf{H} \in [\mathbf{B}]$ such that

- There exists a sequence of integers $j_1 < j_2 < ... < j_n$ such that for all $0 \leq i \leq n$ we have $h_{i,j} = 0$ for all $j < j_i$ (row echelon structure).
- For $0 \leq k < i \leq n$ we have $0 \leq h_{l,j_i} < h_{i,j_i}$ (i.e the pivot element is the greatest along its column and the coefficient above are nonnegative).

The matrix $\mathbf{H}$ is called the Hermite normal form of $\mathbf{B}$ denoted as $\mathbf{HNF}(\mathbf{B})$ [44]. It is easy to see that for any rational matrix $\mathbf{B}' \in \mathbb{Q}^{k \times n}$, there also exists a unique canonical form under unimodular matrices $GL_k(\mathbb{Z})$. Let $r$ denote the least common multiple of the denominators of entries in $\mathbf{B}'$, notice that $r \cdot \mathbf{B}'$ in an integer matrix.

$$\mathbf{HNF}(\mathbf{B}') = \frac{1}{r}\mathbf{HNF}(r \cdot \mathbf{B}').$$

**$q$-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 $\mathbf{v} \mod q$. Let $\mathbf{G} = [\mathbf{g}_1 | \ldots | \mathbf{g}_k] \in \mathbb{F}_q^{n \times k}$ for some positive integers $n, k, q$ with $n \geq k$. We define below two important families of $q$-ary lattices in cryptography

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

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

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

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

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, $k+1 \leq j \leq n$. A basis of $\Lambda_q^\top$ is given by

$$q \cdot (\mathbf{B}^{-1})^{\mathbf{t}}.$$

Hence, these lattices are dual to each other up to normalization; i.e $\Lambda_q(\mathbf{G}) = q \cdot \Lambda_q^\top(\mathbf{G})$ and $\Lambda_q^\top(\mathbf{G}) = q \cdot \Lambda_q(\mathbf{G})$.

**Gaussian distribution and standard tail inequality**

For any real number $\beta > 0$ the *Gaussian distribution* with mean 0 is the distribution on $\mathbb{R}$ having density function $D_\beta(x) = \frac{1}{\beta}\exp(-\pi(x/\beta)^2)$, for all $x \in \mathbb{R}$.

A random variable with normal distribution lies within $\pm\frac{t\cdot\beta}{\sqrt{2\pi}}$ of its mean, except with probability at most $\frac{1}{t}\cdot\exp(-t^2/2)$.

**Discrete Gaussian distribution over $\mathbb{Z}_q$**

For any integer $q \geq 2$, the *discrete Gaussian distribution* $\overline{\psi}_\beta(q)$ over $\mathbb{Z}_q$ with mean 0 and standard deviation $\pm\frac{q\cdot\beta}{\sqrt{2\pi}}$ is obtained by drawing $y \leftarrow D_\beta$ and outputting $\lfloor q \cdot y \rceil \pmod q$.

**Fact**: Let $\beta > 0$ and $q \in \mathbb{Z}$, let the vector $\mathbf{x} \in \mathbb{Z}_q^n$ be chosen as $\mathbf{x} \leftarrow \overline{\psi}_\beta(q)^n$. Let $\mathbf{x} \in \mathbb{Z}^n$ be an arbitrary vector and let $g = \omega\left(\sqrt{\log n}\right)$. Then with overwhelming probability $\|\langle \mathbf{x}, \mathbf{y}\rangle\| \leq \beta q \cdot \|\mathbf{y}\|$ (see [14]).

**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 $\mathcal{L}$

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

The subscript $\mathbf{c} = 0$ when omitted.

**Statistical Distance and Indistinguishability**

Let $X$ and $Y$ be two random variables over some countable set $\Omega$. The statistical distance between $X$ and $Y$ is

$$\Delta(X,Y) = \frac{1}{2}\left\{\sum_{\omega \in \Omega} |Pr[X(\omega)] - Pr[Y(\omega)]|\right\}.$$

We say that two probability ensembles $\mathcal{X} = \{X_k\}_{k \in \mathbb{N}}$ and $\mathcal{Y} = \{Y_k\}_{k \in \mathbb{N}}$ are statistically close ($\mathcal{X} \sim \mathcal{Y}$) if $\Delta(X_k, Y_k)$ is a negligible function in $k$.

**Theorem 1** *Let $n$ be the security parameter and $t(n) \in \omega(\sqrt{\log n})$ be some fixed function (say $t(n) = \log n$.) For any $r(n) \in \omega(\sqrt{\log n})$, $c \in \mathbb{R}$, the algorithm* Sample$\mathbb{Z}$ *samples in PPT according to a distribution that is statistically close to the discrete Gaussian distribution $D_{r(n),c,\mathbb{Z}}$ over the one dimensional integer lattice $\mathbb{Z}$.*

**Proof** (see [21]).

**Theorem 2** *Given a basis $\mathbf{B} = [\mathbf{b}_1|\ldots|\mathbf{b}_k] \in \mathbb{R}^{n \times k}$ 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* Sample**Point** *outputs a sample from a distribution that is statistically close to $D_{s,\mathbf{c},\mathcal{L}}$.*

**Algorithm 1** Sample$\mathbb{Z}$.

– Input security parameter $(n,\, c,\, r(n),\, t(n))$

1. Pick an integer $z$ uniformly from $\mathbb{Z} \cap [c - r(n) \cdot t(n), c + r(n) \cdot t(n)]$.
2. Output $z$ with probability $\rho_{r(n)}(z - c)$.
3. Otherwise Repeat.

**Algorithm 2** Sample**Point**.

– Input $(\mathbf{B},\, \mathbf{c},\, s,\, t(n))$

1. $\mathbf{v}_k \leftarrow 0$ and $\mathbf{c}_k \leftarrow \mathbf{c}$.
2. For $i$ from $k$ to 1 do
   (a) $c'_i \leftarrow \frac{\langle \mathbf{c}_i, \tilde{\mathbf{b}}_i \rangle}{\|\tilde{\mathbf{b}}_i\|^2}$ and $s'_i \leftarrow \frac{s}{\|\tilde{\mathbf{b}}_i\|}$.
   (b) Pick $z_i \leftarrow$ Sample$\mathbb{Z}(n, c'_i, s'_i, t(n))$.
   (c) $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$ and $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i - z_i \mathbf{b}_i$.
3. Output $\mathbf{v}_0$.

**Proof** (see [21]).

**Theorem 3** *There is a deterministic polynomial-time algorithm that, given an arbitrary basis* $\{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ *of an* $n$*-dimensional lattice* $\mathcal{L}$ *and a set of linearly independent lattice vectors* $\mathbf{S} = [\mathbf{s}_1 | \mathbf{s}_2 \ldots | \mathbf{s}_k] \in \mathcal{L}$ *with ordering* $\|\mathbf{s}_1\| \leq \|\mathbf{s}_2\| \leq \cdots \leq \|\mathbf{s}_k\|$, *outputs a basis* $\{\mathbf{r}_1 \ldots \mathbf{r}_k\}$ *of* $\mathcal{L}$ *such that* $\|\tilde{\mathbf{r}}_i\| \leq \|\tilde{\mathbf{s}}_i\|$ *for* $1 \leq i \leq k$.

**Proof** We will only provide a sketch of the proof. For details see [19], page 129. Write $\mathbf{S} = [\mathbf{s}_1 | \ldots | \mathbf{s}_k]$ and $\mathbf{B} = [\mathbf{b}_1 | \ldots | \mathbf{b}_k]$. Since $\{\mathbf{s}_1 \ldots \mathbf{s}_k\} \subset \mathcal{L}$, we can write $\mathbf{S} = \mathbf{BQ}$ for some non-singular integer matrix $\mathbf{Q} \in \mathbb{Z}^{k \times k}$. Suppose that $\mathbf{Q} \notin GL_k(\mathbb{Z})$ (otherwise $\mathbf{S}$ is a basis of $\mathcal{L}$). Find $U \in GL_k(\mathbb{Z})$, such that $\mathbf{T} = U\mathbf{Q} \in \mathbb{Z}^{k \times k}$ is an upper triangular matrix and write $\mathbf{R} = \mathbf{B}U^{-1}$. This can easily be achieved by performing elementary

row operations.[1] Since $U \in GL_k(\mathbb{Z})$ we also have $U^{-1} \in GL_k(\mathbb{Z})$. Hence $\mathbf{R} = \mathbf{B}U^{-1}$ is a basis of $\mathcal{L}$.

**Orthogonal Matrices and Givens Rotations** A Givens rotation is an orthogonal $n \times n$ matrix of the form

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

where $c = \cos(\theta)$ and $s = \sin(\theta)$. The non-zero elements of Givens matrix is given by

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

$g_{i,i} = c$ and $g_{j,j} = c$

$g_{i,j} = s$ and $g_{j,i} = -s$ if $i < j$

$g_{i,j} = -s$ and $g_{j,i} = s$ if $i > j$.

1. We are only allowed to multiply a row by $-1$, add an integer multiple to another row and row switching.

The product $G_{(i,j,\theta)} \cdot \mathbf{v}$ represents a counter clockwise rotation of the vector $\mathbf{v}$ in $(i, j)$ plane by angle $\theta$. 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 $D_\epsilon \in \mathcal{D}_{\epsilon_n}$

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

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

**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 $\{i, j\} = \{y, z\}$.

$$G_{(i,j,\theta_{i,j})} \cdot G_{(j,i,\theta_{j,i})}\mathbf{v} = G_{(j,i,\theta_{j,i})} \cdot G_{(i,j,\theta_{i,j})}\mathbf{v}$$

$$G_{(i,j,\theta_{i,j})} \cdot G_{(x,y,\theta_{x,y})}\mathbf{v} = G_{(x,y,\theta_{x,y})} \cdot G_{(i,j,\theta_{i,j})}\mathbf{v}.$$

3. *Linearity*: For any Givens matrix $G_{i,j}$, any vector $\mathbf{v} \in \mathbb{R}^n$ and any permutation $\pi \in \sigma_n$

$$G_{(\pi(i),\pi(j),\theta_{i,j})}P_\pi \cdot \mathbf{v} = P_\pi G_{(i,j,\theta_{i,j})} \cdot \mathbf{v}$$

   $P_\pi$ is the corresponding permutation matrix of $\pi$.

## 2.4 Linear Codes

Let $\mathbb{F}_q$ be a finite field of order $q$. A $[k, n]$ linear code $C$ over $\mathbb{F}_q$ is a $k$ dimensional subspace of $\mathbb{F}_q^n$. The code $C$ has $q^k$ elements (called codewords) and $\frac{1}{k!}\prod_{i=1}^{k-1}(q^k - q^i)$ distinct bases. Given a basis

$$\mathbf{G} \in \mathbb{F}_q^{n \times k},$$

for $C$ each codeword $\mathbf{c}$ can be written uniquely as a linear combination,

$$\mathbf{c} = \mathbf{G}\mathbf{u} \in \mathbb{F}_q^n,$$

The *Hamming distance* between two codewords $\mathbf{c}_1, \mathbf{c}_2 \in C$, denoted by $d(\mathbf{c}_1, \mathbf{c}_2)$ is the number of places at which $\mathbf{c}_1$ and $\mathbf{c}_2$ differ. If $\mathbf{c}_1 = (x_1, \ldots, x_n)$ and $\mathbf{c}_2 = (y_1, \ldots, y_n)$, then

$$d(\mathbf{c}_1, \mathbf{c}_2) = d(x_1, y_1) + \ldots + d(x_n, y_n)$$

where $d(x_i, y_i) = 0$ if and only if $x_i = y_i$ and $d(x_i, y_i) = 1$ otherwise. The minimum distance of a code $C = [k, n]$, denoted by $d(C)$

$$d(C) = min\{d(\mathbf{c}_1, \mathbf{c}_2) : \mathbf{c}_1, \mathbf{c}_2 \in C \,\&\, \mathbf{c}_1 \neq \mathbf{c}_2\}.$$

The (Hamming) weight of a codeword $\mathbf{c}$, denoted by $wt(\mathbf{c})$, is defined to be the number of nonzero coordinates in $\mathbf{c}$. The minimum (Hamming) weight of $C$, denoted $wt(C)$, is the smallest of the weights of the nonzero codewords of $C$ and is equivalent to the minimum distance of the code

$$wt(C) = d(C).$$

### 2.4.1 Hard Problems in Coding theory and Graph Theory

**Definition 1 (Permutation Code Equivalence PCE)**: *We say that two* $[k, n]$ *linear codes* $C_1$ *and* $C_2$ *with generators* $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{F}_q^{n\times k}$ *are permutation equivalent if there exists an invertible matrix* $\mathbf{N} \in \mathbb{F}_q^{k\times k}$ *and a permutation matrix* $P \in \mathcal{P}_n$ *such that* $\mathbf{G}_2 = P\mathbf{G}_1\mathbf{N}$.

• **The Decision Problem**: Given generator matrices $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{F}_q^{k\times n}$, decide whether they are permutation equivalent.

**Definition 2 (Linear Code Equivalence LCE)**: *We say that two* $[k, n]$ *linear codes* $C_1$ *and* $C_2$ *are linearly equivalent, if there exists an invertible matrix* $\mathbf{N} \in \mathbb{F}_q^{k\times k}$ *and an* $n \times n$ *monomial matrix* $P \in \mathcal{P}(n, \mathbb{F}_q)$, *such that* $\mathbf{G}_2 = P\mathbf{G}_1\mathbf{N}$.

• **The Decision Problem**: Given generator matrices $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{F}_q^{n\times k}$, decide whether they are linearly equivalent.

**Definition 3 (Graph Isomorphism GI)**: *Let* $G = (V, E)$ *and* $G' = (V, E')$ *be two graphs. We say* $G$ *and* $G'$ *are isomorphic if there exists a permutation* $\pi : V \to V$ *such that* $(v, u) \in E$ *if and only if* $(\pi(v), \pi(u)) \in E'$, *for all* $u, v \in V$.

• **The Decision Problem**: Given graphs $G = (V, E)$ and $G' = (V, E')$ decide whether they are isomorphic.

## Chapter 3
## Oblivious Transfer from weakly Random-Self-Reducible Encryption

Oblivious Transfer is a protocol in which a sender transfers one of potentially many pieces of information to a receiver, but remains oblivious as to what piece has been transferred. It is a very important cryptographic primitive, with applications ranging from Oblivious function evaluation, Commitment schemes, two-party computation, private information retrieval and to zero-knowledge proofs. The concept of OT first appeared in the seminal paper of Stephen J. Wiesner entitled *Conjugate Coding*. However, the paper was rejected by IEEE Transactions on Information Theory and was eventually published years later in 1983 in SIGACT News [4]. The notion of OT was introduced to the world of cryptography by Rabin in 1981 [37]. In this variant, a sender sends a message to a receiver who receives the message with probability 1/2. The protocol ensures that the sender remains oblivious as to whether or not the receiver received the message.

Even, Goldreich and Lempel introduced another variant of oblivious transfer called One-out-of-Two oblivious transfer denoted as $\binom{2}{1}$-OT [38]. In this variant a sender inputs two ordered bits $b_0, b_1$ and a receiver inputs a choice bit $c$. The protocol sends $b_c$ to the receiver, without the sender learning $c$, while the receiver learns nothing other than $b_c$. Both of these variants were shown to be equivalent by Crépeau [35]. The early implementations of Oblivious Transfer were very innovative but did not offer very strong security [37, 38]. The very first OT protocols that may

be considered secure were introduced by Berger, Peralta and Tedrick [36] and Ficher, Micali and Rackoff [39].

Later, two methodologies where introduced. A first set of results by Brassard, Crépeau and Robert [25] relied on *Random-Self-Reducibility* (**RSR** for short) of certain number theoretic assumptions such as the Quadratic Residuosity assumption, the RSA assumption or the Discrete log assumption. These results were not extended to very general computational assumptions because the **RSR** property, which was at the heart of the construction, is not very common. In a second set of results by Goldreich, Micali and Wigderson [26], secure Oblivious Transfer protocols were constructed from the generic assumption that (enhanced) [1] Trap-door One-Way permutations exist.

Unfortunately, all constructions that are used to implement secure OT under either of these methodologies fall apart when faced with a quantum computer [1]: none of the so-called Post-Quantum Cryptosystems can directly implement secure OT under these methodologies. Nevertheless, some small modifications to the GMW methodology have led to proposals for OT under the Learning with error LWE assumption [33, 21]. Similarily, Dowsley, van de Graaf, Müller-Quade and Nascimento [31] as well as Kobara, Morozov and Overbeck [32] have proposed Oblivious Transfer protocols based on assumptions related to the McEliece public-key cryptosystem. Both of these papers use generalization of the GMW methodology. However both

1. The enhanced property is not very restrictive, but some examples of candidates Trap-door One-Way permutations seem to escape it [27].

of them also require an extra computational assumption on top of McEliece's to conclude security. Those were the first proposals for OT protocols believed secure against a quantum computer [2] .

More recently, a new methodology has been proposed by Peikert, Vaikuntanathan and Waters [34] using the notion of dual-mode cryptosystems. Their approach can be instantiated using several number theoretic assumptions, and LWE in the Post-Quantum case. This approach leads to universally composable protocols and requires the presence of a Common Reference String.

In this chapter, we first formalize the results by Brassard, Crépeau and Robert [25] which relied on the **RSR** property of certain number theoretic assumptions in order to introduce a new notion of weakly random-self-reducible encryption scheme **wRSR**. We then show how it is possible to construct a secure Oblivious Transfer under the sole assumption that a secure **wRSR** encryption scheme exists. We show that encryption schemes from two (Post-Quantum) computational assumptions [14, 13] have this weak property. We hope that in the future, our methodology may be used for various new computational assumptions as well.

### 3.1 Random-Self-Reducible Encryption Scheme

Informally speaking an encryption scheme is *Random-Self-Reducible* **RSR** if an arbitrary ciphertext $c$ may be efficiently transformed to a uniformly distributed ciphertext $c'$ by a user who only knows the public-key from that system. Moreover,

2. Earlier results accomplished a similar security level using only a One-Way function and Quantum Communication. The motivation of the papers cited above and of the current work is to avoid quantum communication altogether [11].

upon learning the decryption $m'$ of $c'$, the user is able to efficiently compute $m$, the decryption of $c$, from knowledge of the relation between $c$ and $c'$.

**Definition 4** *Let $\xi = (KeyGen, Enc, Dec, \mathcal{M}, \mathcal{C})$ be a public-key cryptosystem and $\lambda$ be the security parameter. The cryptosystem $\xi$ is random-self-reducible if there exists a set $\widehat{\mathcal{M}}$, a pair of probabilistic polynomial-time algorithms $(\mathcal{S}, \mathcal{S}')$, together with a polynomial-time algorithm $\widehat{Dec}$, such that for a key pair $(sk, pk) \leftarrow KeyGen(1^\lambda)$ and uniformly picked string $\mathbf{R}$ from $\widehat{\mathcal{M}}$,*

1. *$\mathcal{S}_{pk} : \widehat{\mathcal{M}} \times \mathcal{C} \to \mathcal{C}$, $\mathcal{S}'_{pk} : \widehat{\mathcal{M}} \times \widehat{\mathcal{M}} \to \mathcal{M}$, and $\widehat{Dec}_{sk} : \mathcal{C} \to \widehat{\mathcal{M}}$,*
2. *$\mathcal{S}_{pk}(\mathbf{R}, c)$ is uniformly distributed over $\mathcal{C}$, for all $c \in \mathcal{C}$,*
3. *$\mathcal{S}'_{pk}(\mathbf{R}, \widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, Enc_{pk}(m)))) = m$, for all messages $m \in \mathcal{M}$.*

### 3.1.1 Examples of RSR Cryptosystem

There are several cryptosystems that satisfy the **RSR** property. For example RSA [2], Goldwasser-Micali [40] and Paillier cryptosystems [41] are all random self-reducible.

**Goldwasser-Micali Cryptosystem**

Let $(x, n)$ and $(p, q)$ denote the public/private keys and $(Enc, Dec)$ denote the encryption/decryption algorithms.

1. $\widehat{\mathcal{M}} = \{0, 1\}$, $\widehat{Dec} = Dec$
2. $\mathcal{S}_{pk}(\mathbf{R}, Enc(b)) = Enc(\mathbf{R}) \cdot Enc(b) \bmod n = Enc(\mathbf{R} \oplus b)$, where bit $\mathbf{R}$ is uniformly chosen.

3. $\mathcal{S}'_{pk}(\mathbf{R}, b) = \mathbf{R} \oplus b$.

**Semantically secure RSA Cryptosystem**

Let $(e, n)$ and $d$ denote the public and private keys and $\big(Enc := (lsb^{-1}(b))^e \bmod n, Dec := lsb(m^d$ denote the encryption/decryption algorithms, where $lsb^{-1}(b)$ is a random element $r$ in $\mathbb{Z}_n^*$ such that $lsb(r) = b$.

1. $\widehat{\mathcal{M}} = \mathbb{Z}_n^*$, $\widehat{Dec}(m) = m^d \mod n$
2. $\mathcal{S}_{pk}(\mathbf{R}, Enc(b)) = \mathbf{R}^e \cdot Enc(b) \bmod n = (\mathbf{R} \cdot lsb^{-1}(b))^e \bmod n$ where $\mathbf{R}$ is uniformly chosen from the message space $\widehat{M}$.
3. $\mathcal{S}'_{pk}(\mathbf{R}, m) = lsb\left(\mathbf{R}^{-1} \cdot m \mod n\right)$.

### 3.2 $\binom{2}{1}$-OT from a RSR Public-Key Cryptosystem

Let $\xi = (KeyGen, Enc, Dec, \mathcal{M}, \mathcal{C})$ be a **RSR** public-key cryptosystem and $\lambda$ be the security parameter. Let $(sk, pk) \leftarrow KeyGen(1^\lambda)$ be sender's private and public-keys. The sender encodes his bits so that $Enc_{pk}(b_0)$ and $Enc_{pk}(b_1)$ are semantically secure encryptions of $b_0, b_1$.

**Protocol 1** $\binom{2}{1}$-OT from **RSR** Cryptosystem.

1: The sender computes $c_0 \leftarrow Enc_{pk}(b_0)$ and $c_1 \leftarrow Enc_{pk}(b_1)$.

2: The sender sends the ordered pair $(c_0, c_1)$ to the receiver.

3: The receiver picks a string $\mathbf{R}$ uniformly from $\mathcal{C}$ and computes $c \leftarrow \mathcal{S}_{pk}(\mathbf{R}, c_i)$ for its choice bit $i$ and sends $c$ to the sender.

4: The sender computes $\widehat{m} \leftarrow \widehat{Dec}_{sk}(c)$ and sends $\widehat{m}$ to the receiver.

5: The receiver obtains the bit $b_i \leftarrow \mathcal{S}'_{pk}(\mathbf{R}, \widehat{m})$.

**Correctness**

We first observe that this protocol correctly computes $\binom{2}{1}$-OT.

$$\mathcal{S}'_{pk}(\mathbf{R}, \widehat{m}) = \mathcal{S}'_{pk}(\mathbf{R}, \widehat{Dec}_{sk}(c)) = \mathcal{S}'_{pk}(\mathbf{R}, \widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, c_i)))$$
$$= \mathcal{S}'_{pk}(\mathbf{R}, \widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, Enc_{pk}(b_i)))) = b_i \text{ by definition 4.}$$

**Theorem 4** *Protocol* 1 *is a secure oblivious transfer in the semi-honest model.*

**Proof**. We will present a simulator for each party. These simulators are given the local input (which also includes the security parameter $\lambda$) and the local output of the corresponding party. The following schematic depiction of the information flow in protocol 1 may be useful towards the constructions of the simulators.

| *Sender* | | *Receiver* |
|---|---|---|
| input $\left((b_0, b_1), 1^\lambda\right)$ | | input $\left(i, 1^\lambda\right)$ |
| $(sk, pk) \leftarrow KeyGen(1^\lambda)$ | $\xrightarrow{pk}$ | |
| $c_0 := Enc_{pk}(b_0), c_1 := Enc_{pk}(b_1)$ | $\xrightarrow{(c_0, c_1)}$ | |
| | $\xleftarrow{c}$ | $c := \mathcal{S}_{pk}(\mathbf{R}, c_i)$ |
| $\widehat{m} \leftarrow \widehat{Dec}_{sk}(c)$ | $\xrightarrow{\widehat{m}}$ | |
| output $\epsilon$ | | output $b_i := \mathcal{S}'_{pk}(\mathbf{R}, \widehat{m})$ |

**Simulator for the sender's view**: We will first present a simulator for the sender's view. On input $\left((b_0, b_1), 1^\lambda, \epsilon\right)$, this simulator uniformly picks $c'$ from $\mathcal{C}$ and outputs $\left((b_0, b_1), 1^\lambda, c'\right)$. Clearly this output distribution is identical to the view of the sender in the real execution. This hold because $c'$ is uniformly distributed over the ciphertext space $\mathcal{C}$. Therefore, the receiver's security is perfect.

**Simulator for the receiver's view**: On input $\left(i, b_i, 1^\lambda\right)$, this simulator generates $(sk', pk') \leftarrow KeyGen(1^\lambda)$ as in protocol 1. It computes $c'_i \leftarrow Enc_{pk'}(b_i)$ and $c'_{1-i} \leftarrow Enc_{pk'}(b)$ (for some $b \in \mathcal{M}$) The simulator then picks a string $\mathbf{R}'$ uniformly. It then computes $c' \leftarrow \mathcal{S}_{pk'}(\mathbf{R}', c'_i)$ and $\widehat{m}' \leftarrow \widehat{Dec}_{sk'}(c')$. The simulator outputs $\left(i, 1^\lambda, pk', c'_0, c'_1, \widehat{m}'\right)$. Note that except for $c'_{1-i}$, this output distribution is identical to the view of the receiver in the real execution. Moreover, since $\xi$ is a semantically secure encryption scheme, it is impossible to distinguish between the encryption of $b_{1-i}$ and $b$ for any probabilistic polynomial time adversary except with negligible probability. Therefore, the sender's security is computational.

**Malicious adversaries**: Of course we are not only interested in the semi-honest case but also to the situation with malicious adversaries. To handle these cases, zero-knowledge proofs are used by the sender to demonstrate that $c_0, c_1$ are well formed encryptions and by the receiver to demonstrate that $c$ is indeed constructed from a single $c_i$ and not a combination of both. We leave it as an exercise to demonstrate the full result including zero-knowledge proofs [27]:

**Theorem 5** *Protocol* 1 *may be compiled to a secure oblivious transfer in the malicious model.*

**Proof** (see [27, 5]).

### 3.3 weakly Random Self-Reducible Encryption

The current state of affairs is that we don't know of any **RSR** cryptosystem believed to be resistant to quantum attacks. The **RSR** property may be considered too strong in its uniformity requirement of the output of $\mathcal{S}$. One can weaken this property to statistical indistinguishability for some pair of probabilistic polynomial

distributions and can still obtain a secure OT protocol provided we have cryptosystems satisfying this weaker property.

In this section we define the notion of *weakly Random-Self-Reducibile* public-key cryptosystem. Informally speaking a *public-key cryptosystem* is weakly Random-Self-Reducible if it is possible efficiently (using the public key) to re-encrypt a ciphertext $c_i$ in a way to make it unrecognizable, regardless of the plaintext it carries. After obtaining decryption of the re-encrypted ciphertext $\widehat{c}$, it is possible to recover the plaintext hidden by the original encryption $c_i$. We accept that the unrecognizability property be statistical indistinguishability instead of perfect indistinguishability as in **RSR** .

Our definition is motivated by the fact that many post-quantum encryption schemes use random errors in the process of encrypting the plaintext. Many of these schemes provide a fair amount of flexibility in choosing the size of the error for a fixed pair of public and private keys. Due to this flexibility, one can easily convert these cryptosystem into a **wRSR** scheme. The encryption algorithm *Enc* involves relatively small errors, while the re-encryption process uses relatively large errors that will hide the original error. The definition is formally stated below.

**Definition 5** *A public-key cryptosystem* $\xi = (KeyGen, Enc, Dec, \mathcal{M}, \mathcal{C})$ *is weakly random-self-reducible if there exist sets* $\widehat{\mathcal{M}}$, $\widehat{\mathcal{C}}$, *a pair of probabilistic polynomial-time algorithms* $(\mathcal{S}, \mathcal{S}')$, *together with a probabilistic polynomial-time algorithm* $\widehat{Dec}$, *and a probabilistic-polynomial time distribution* $\chi$ *on* $\widehat{\mathcal{C}}$ *such that for all* $c_1, c_2 \in \mathcal{C}$, *key pair* $(sk, pk) \leftarrow KeyGen\left(1^{\lambda}\right)$ *and* $\mathbf{R} \xleftarrow{\chi} \widehat{\mathcal{C}}$:

1. $\mathcal{S}_{pk} : \widehat{\mathcal{C}} \times \mathcal{C} \rightarrow \widehat{\mathcal{C}}$, $\mathcal{S}'_{pk} : \widehat{\mathcal{C}} \times \widehat{\mathcal{M}} \rightarrow \mathcal{M}$ *and* $\widehat{Dec}_{sk} : \widehat{\mathcal{C}} \rightarrow \widehat{\mathcal{M}}$,
2. $\mathcal{S}_{pk}(\mathbf{R}, c_1)$ *and* $\mathcal{S}_{pk}(\mathbf{R}, c_2)$ *are statistically indistinguishable,*
3. $\mathcal{S}'_{pk}\Big(\mathbf{R}, \widehat{Dec}_{sk}\left(\mathcal{S}_{pk}(\mathbf{R}, Enc_{pk}\left(m\right))\right)\Big) = m$, *for all messages* $m \in \mathcal{M}$.

Note that **RSR** is the sub-case of **wRSR** where $\widehat{\mathcal{M}} = \mathcal{C}$, $\chi$ is the uniform distribution over $\widehat{\mathcal{C}}$ and $S_{pk}(\mathbf{R}, c)$ is uniformly distributed over $\mathcal{C}$. In section 3.3.2 we show that one can construct a weakly Random-Self-Reducible encryption schemes based on the Approximate Integer GCD assumption [14] or the Learning with Errors assumption [13].

### 3.3.1 $\binom{2}{1}$-OT from a wRSR Cryptosystem

Let $\xi = (KeyGen, Enc, Dec, \mathcal{M}, \mathcal{C})$ be a **wRSR** public-key cryptosystem and $\lambda$ be the security parameter. Let $(sk, pk) \leftarrow KeyGen(1^{\lambda})$ be the sender's private and public-keys. The sender encodes his bits so that $Enc_{pk}(b_0)$ and $Enc_{pk}(b_1)$ are semantically secure encryptions of $b_0$, $b_1$.

**Protocol 2** $\binom{2}{1}$-OT from **wRSR** Cryptosystem.

The sender computes $c_0 \leftarrow Enc_{pk}(b_0)$ and $c_1 \leftarrow Enc_{pk}(b_1)$.

2: The sender sends the ordered pair $(c_0, c_1)$ to the receiver.

The receiver picks a string $\mathbf{R} \xleftarrow{\chi} \widehat{\mathcal{C}}$ and computes $\hat{c} \leftarrow \mathcal{S}_{pk}(\mathbf{R}, c_i)$, where $i$ is the receiver's choice bit.

4: The receiver sends $\hat{c}$ to the sender.

The sender computes $\widehat{m} \leftarrow \widehat{Dec}_{sk}(\hat{c})$ and sends $\widehat{m}$ to the receiver.

6: The receiver obtains the bit $b_i \leftarrow \mathcal{S}'_{pk}(\mathbf{R}, \widehat{m})$.

**Correctness**. We first observe that this protocol correctly computes $\binom{2}{1}$-OT.

$$\widehat{Dec}_{sk}(\widehat{c}) = \widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, c_i)) = \widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, Enc_{pk}(b_i))).$$

$$\implies \mathcal{S}'_{pk}(\mathbf{R}, \widehat{Dec}_{sk}(\widehat{c})) = \mathcal{S}'_{pk}(\mathbf{R}, \widehat{m}) = b_i \text{ (by property 3)}.$$

**Theorem 6** *The Protocol 2 is a secure oblivious transfer between the sender and the receiver, provided both parties follow the protocol honestly.*

**Simulator for the sender's view**: The simulator is very similar to the **RSR** case. On input $\left((b_0, b_1), 1^\lambda, \epsilon\right)$, the simulator picks $\mathbf{S} \xleftarrow{\chi} \widehat{\mathcal{C}}$ and arbitrary ciphertext $c \in \mathcal{C}$. The simulator computes $\widehat{a} \leftarrow \mathcal{S}_{pk}(\mathbf{S}, c)$ and outputs $\left((b_0, b_1), 1^\lambda, \widehat{a}\right)$. The output distribution is statistically close to the view of the sender in the real execution. This holds because the statistical distance between $\widehat{a}$ and $\widehat{c}$ is negligible.

**Simulator for the receiver's view**: The simulator is very similar to the **RSR** case. On input $\left(i, b_i, 1^\lambda\right)$, the simulator computes $(sk', pk') \leftarrow KeyGen(1^\lambda)$, $x_i \leftarrow Enc_{pk'}(b_i)$, $x_{1-i} \leftarrow Enc_{pk'}(b)$ (for some $b \in \mathcal{M}$) and $\widehat{x} \leftarrow \mathcal{S}_{pk}(\mathbf{S}, x_i)$ where $\mathbf{S} \xleftarrow{\chi} \widehat{\mathcal{C}}$. The simulator then computes $\widehat{w} \leftarrow \widehat{Dec}_{sk'}(\widehat{x})$ and outputs $\left(i, 1^\lambda, pk', x_0, x_1, \widehat{w}\right)$. The

output is computationally indistinguishable from the view of the receiver in the real execution. This is because except for $x_{1-i}$, this output distribution is identical to the view of the receiver in the real execution. Furthermore since $\xi$ is semantically secure, no probabilistic polynomial-time adversary can distinguish between the encryption of $b_{1-i}$ and $b$ except with negligible probability. Therefore, the sender's security is computational.

**Malicious adversaries**: We handle the case of malicious adversaries in a similar way as the **RSR** case. However, due to the nature of the cryptosystems used to implement **wRSR**, a zero-knowledge proof that $c_0, c_1 \in \mathcal{C}$ is also necessary. We leave it as an exercise to demonstrate the full result including zero-knowledge proofs:

**Theorem 7** *Protocol* 2 *may be compiled to a secure oblivious transfer in the malicious model.*

### 3.3.2 Instantiation of wRSR public-key Cryptosystems

In this section we provide concrete instantiations of **wRSR** schemes from two different post-quantum assumptions.

1. Approximate Integer GCD problem (**AIGP**)[13].
2. Learning with Errors (**LWE**)[33].

More precisely we show that one can easily construct a **wRSR** from the cryptosystems presented in [13, 14]. Please note that for these encryption schemes, operation ($a \mod n$) means mapping integer $a$ into the interval $[-\lfloor n/2 \rfloor, \lfloor n/2 \rfloor]$, (where $n$ is an odd positive integer).

### 3.3.3 Approximate Integer GCD problem

Let $p$ be a large $\eta$-bit odd integer and $x_i$'s are defined as follows

$$x_i = q_i p + r_i, \quad 0 \leq i \leq \tau$$

where $x_i$ is a $\gamma$-bit number which is much larger than $p$ and $r_i$ is a $\rho$-bit error-term which is much smaller than $p$ in absolute value. W.l.o.g. assume that $x_0$ is the largest of them, and that $x_0$ is odd. Under the Approximate Integer GCD assumption the function

$$f_x(s, z, b) = \left( 2\overline{z} + b + 2 \sum_{i=1}^{\tau} x_i s_i \right) \bmod x_0$$

is one-way for anyone who does not know $p$, where $b \in \{0, 1\}$, $\mathbf{s} \in \{0, 1\}^\tau$ is a random binary vector and $\overline{z}$ is a random error term of appropriate size (see below).

**public-key Cryptosystem from AIGCD Problem**

Van Dijk, Gentry, Halevi and Vaikuntanathan constructed a fully homomorphic encryption scheme based on the problem of finding an approximate integer gcd [13]. The construction below has many parameters, controlling things like the number of integers in the public-key and the bit-length of the various components. Specifically, we use the following five parameters (all polynomial in the security parameter $\lambda$):

- $\eta$ is the bit-length of the secret key $p$.
- $\gamma$ is the bit-length of the integers $x_i$ in the public-key.
- $\rho$ is the bit-length of the noise $r_i$.
- $\rho'$ is the bit-length of the random error $z$.
- $\tau$ is the number of integers in the public-key, (contrary to the other parameters, this is *not* a bit-size.)

These parameters must be set under the following constraints:

- $\rho \in \omega(\log \lambda)$, to protect against brute-force attacks on the noise.
- $\rho' = \Omega(\rho + \log \tau)$ ($\tau$ is a polynomial is $\lambda$, e.g $\tau = \lambda$).

– $\eta \geq \rho \cdot \Theta(\lambda \log^2 \lambda)$ and should satisfy $2^{\eta-2} > 2^{\rho'} + \tau \cdot 2^{\rho}$, to avoid sums of errors passing $p/2$.

– $\gamma \in \omega(\eta^2 \log \lambda)$, to thwart various lattice-based attacks on the underlying approximate-gcd problem.

– $\tau \geq \gamma + \omega(\log \lambda)$, in order to use the leftover hash lemma in the reduction to approximate gcd.

The public-key is the vector $\mathbf{x} = (x_0, x_1, \ldots, x_\tau)$ and the private key is the $\eta$ bit integer $p$. To encrypt a bit $b \in \{0, 1\}$ under the public-key $\mathbf{x}$.

– $Enc_{\mathbf{x}}(b)$

1. Pick uniformly a random bit string $s_1, \cdots, s_\tau$ and pick uniformly a $\overline{\rho}$-bit error-term $\overline{z}$.
2. Output the ciphertext $c \leftarrow \left(2\overline{z} + b + 2\sum_{i=1}^{\tau} x_i s_i\right) \bmod x_0$.

– $Dec_p(c)$

1. $c' \leftarrow c \bmod p$.
2. Output bit $b \leftarrow c' \bmod 2$.

The decryption works, provided the overall distance to the nearest multiple of $p$ does not exceed $p/2$, that is $2(\overline{z} + \sum_{i=1}^{\tau} r_i s_i)$ is less than $p/2$ in absolute value. For the above choice of parameters this will always be the case. We rely on the work of [13] to assess that the resulting cryptosystem is a semantically secure encryption scheme.

**weakly RSR based on Approximate Integer GCD**

The cryptosystem based on **AIGCD** can easily be converted to a **wRSR** encryption scheme. Keeping the same notations as above we set

- $\rho = 2\sqrt{\lambda}$ (is the size of $r_i$'s in the public key).
- $\rho' = 2\rho$ (size of the error term $\mathcal{Z}$ in $\mathcal{S}_{pk}$)
- $\overline{\rho} = \rho/2$ (size of the error term $\overline{z}$ in $Enc$).
- $\eta = \Theta(\lambda)$ (size of the private key).
- $\gamma \in \omega(\eta^2 \log \lambda)$.
- $\tau = \gamma + \rho$ (number of $x_i$'s in public-key $\mathbf{x}$).
- $\mathcal{I} = \left\{[-2^{\rho'}, -2^{\rho'-1}] \cup [2^{\rho'-1}, 2^{\rho'}]\right\} \cap \mathbb{Z}$.
- $\mathcal{R}' = \left\{ 2\mathcal{Z} + 2\sum_{i=1}^{\tau} x_i w_i \bmod x_0 : \mathcal{Z} \in \mathcal{I}, w_i \in \{0,1\} \right\}$.

- $\widehat{\mathcal{M}} = \{0,1\}$ and $\widehat{\mathcal{C}} = \mathcal{C} \times \widehat{\mathcal{M}}$.
- The distribution $\chi$ is induced by picking $\mathbf{r} \xleftarrow{uniform} \mathcal{R}'$, $e \xleftarrow{uniform} \widehat{\mathcal{M}}$ and outputting $\mathbf{R} = (\mathbf{r}, e)$.
- $\mathcal{S}_{pk}(\mathbf{R}, c) := (\mathbf{r} + e + c) \bmod x_0$.
- $\widehat{Dec}_{sk} := Dec_{sk}$.
- $\mathcal{S}'_{pk}(\mathbf{R}, \hat{b}) := (e + \hat{b}) \bmod 2$.

**wRSR Encryption Scheme from AIGCD**

**wRSR Properties**(**Semi-Honest Case**). The scheme clearly satisfies the first and the third properties for the above choice of parameters. For the second property let

$$\mathcal{S}_{pk}(\mathbf{R}, c) = \left(2\mathcal{Z} + e + 2\sum_{i=1}^{\tau} x_i w_i\right) + c \bmod x_0$$
$$\mathcal{S}_{pk}(\mathbf{R}, c') = \left(2\mathcal{Z} + e' + 2\sum_{i=1}^{\tau} x_i w_i\right) + c' \bmod x_0.$$

Since $c, c' \in \mathcal{C}$, there exist $\overline{\rho}$ bit integers $\overline{z}, \overline{z}'$, vectors $\mathbf{s}, \mathbf{s}' \in \{0,1\}^{\tau}$ and bits $b, b' \in \{0,1\}$ such that

$$c = \left(2\overline{z} + b + 2\sum_{i=1}^{\tau} x_i s_i\right) \bmod x_0 \;\; \& \;\; c' = \left(2\overline{z}' + b' + 2\sum_{i=1}^{\tau} x_i s'_i\right) \bmod x_0$$

Note that $\mathcal{S}_{pk}(\mathbf{R}, c)$ and $\mathcal{S}_{pk}(\mathbf{R}, c')$ are perfectly indistinguishable if $\mathbf{r} + b + e$ and $\mathbf{r} + b' + e$ lie in the interval $\mathcal{I}$. Also note that both $\mathbf{r} + b + e$ and $\mathbf{r} + b' + e$ can at most be $2^{\rho'+1} + 2^{\overline{\rho}+1} + \tau \cdot 2^{\rho+2} + 2$ in the absolute value and are guaranteed to lie in $\mathcal{I}$ as far as $\mathcal{Z}$ or $\mathcal{Z}'$ do not lie in

$$\mathbb{Z} \cap \left\{[-2^{\rho'}, (2^{\overline{\rho}+1} + \tau 2^{\rho+2} + 2) - 2^{\rho'}] \cup [2^{\rho'} - (2^{\overline{\rho}+1} + \tau 2^{\rho+2} + 2), 2^{\rho'}]\right\}.$$

Note that $\overline{\rho}$ is $\rho/2$ bits, $\rho = 2\sqrt{\lambda}$ and $\tau = \tilde{O}(\lambda^2)$. The probability of $\mathcal{Z}$ or $\mathcal{Z}'$ lie in this interval is

$$2 \times \left(\frac{2^{\overline{\rho}+1} + \tau \cdot 2^{\rho+1} + 2}{2^{2\rho-1}}\right) = \left(\frac{2^{\sqrt{\lambda}+1} + \tau \cdot 2^{2\sqrt{\lambda}+1} + 2}{2^{4\sqrt{\lambda}-3}}\right) < 2^{-\sqrt{\lambda}} \cdot \tau$$

which is negligible in the security parameter $\lambda$. Hence, $\mathcal{S}_{pk}(\mathbf{R}, c)$ and $\mathcal{S}_{pk}(\mathbf{R}, c')$ are statistically indistinguishable.

### 3.3.4 Learning with Errors (LWE)

Let $n, q \geq 2$ be positive integers and $\chi$ be a distribution on $\mathbb{Z}_q$. For a uniformly chosen vector $\mathbf{s} \in \mathbb{Z}_q^n$ we obtain a distribution $A_{\mathbf{s},\chi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and a noise $x \leftarrow \chi$ and outputting $\left(\mathbf{a}, \mathbf{a}^T\mathbf{s} + x\right)$.

**Definition (LWE)**. For an integer $q = q(n)$ and a distribution $\chi$ on $\mathbb{Z}_q$, the goal of the (average case) **LWE** problem is defined as follows : given $m$ independent samples from $A_{\mathbf{s},\chi}$ (for some uniformly chosen fixed vector $\mathbf{s} \in \mathbb{Z}_q^n$) output $\mathbf{s}$ with non-negligible probability. The **Decision** version LWE problem denoted as $\mathbf{distLWE}_{n,m,q,\chi}$ is to distinguish (with non-negligible advantage) from the $m$ samples chosen according to $A_{\mathbf{s},\chi}$, from $m$ samples chosen uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

In [33] Regev proved that the search version LWE is at least as hard as quantumly approximating certain lattice problems in the worst case. Formally Regev proved the following theorem.

**Theorem 8** *Let $n$, $q$ be integers and $\alpha \in (0,1)$ be such that $q > 2\sqrt{n}$. If there exists an efficient algorithm that solves LWE then there exists an efficient quantum algorithm that approximates the decision version of the shortest vector problem* (**GAPSVP**$_\gamma$) *and the shortest independent vectors problem* (**SIVP**) *to within $\tilde{O}(n/\alpha)$ in the worst case.*

**A Simple BGN-Type Cryptosystem**

The BGN-Type cryptosystem is a semantically secure public-key cryptosystem, whose security is equivalent to the hardness of the LWE problem [14]. It supports polynomially many additions and one multiplication without increasing the ciphertext size.

- $n$ is the security parameter and $c = c(n) > 0$ be any function of $n$.
- $q > 2^{20}(c+4)^3 n^{3c+4} \log^5 n$ is a prime modulus.
- The message space is the set $\mathcal{M} = \{\mathbf{B} \in \mathbb{Z}_2^{m \times m} : \ m = \lfloor 8n \log q \rfloor\}$.
- $\beta = \frac{1}{27n^{1+(3c/2)}\sqrt{qm}\log n \log q}$ specify a discrete normal distribution $\overline{\psi}_\beta(q)$ over $\mathbb{Z}_q$.

The public-key is a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and the private key is a matrix $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$, such that

- $\mathbf{A}$ is statistically close to uniform distribution over $\mathbb{Z}_q^{m \times n}$.
- $\mathbf{T} \cdot \mathbf{A} \mod q = \mathbf{0}$ and $\mathbf{T}$ is invertible over $\mathbb{Z}$.
- The Euclidean norm of all the rows in $\mathbf{T}$ is bounded by $O(n \log q)$.

To encrypt a binary $m \times m$ matrix $\mathbf{B}$ under the public-key $\mathbf{A}$:

– $Enc_{pk}(\mathbf{B})$

1. Pick a matrix $\mathbf{S} \in \mathbb{Z}_q^{n\times m}$ uniformly and an error matrix $\mathbf{X} \in \mathbb{Z}_q^{m\times m}$ with each entry in $\mathbf{X}$ is chosen independently according to the distribution $\overline{\psi}_\beta(q)$.
2. Output the ciphertext $\mathbf{C} \leftarrow \mathbf{AS} + \mathbf{2X} + \mathbf{B} \pmod q \in \mathbb{Z}_q^{m\times m}$.

– $Dec_{sk}(\mathbf{C})$

1. Set $\mathbf{D} \leftarrow \mathbf{TCT}^t \mod q = \mathbf{T}(\mathbf{2X} + \mathbf{B})\mathbf{T}^t \mod q$.
2. Output the plaintext $\mathbf{B} \leftarrow \mathbf{T}^{-1}\mathbf{D}\,(\mathbf{T}^t)^{-1} \pmod 2 \in \mathbb{Z}_2^{m\times m}$.

To see that the decryption works, recall that $\mathbf{T} \cdot \mathbf{A} \mod q = \mathbf{0}$, therefore $\mathbf{TCT}^t \equiv \mathbf{T}(\mathbf{2X} + \mathbf{B})\mathbf{T}^t \mod q$. Moreover, for the above choice of parameters each entry in $\mathbf{T}(\mathbf{2X} + \mathbf{B})\mathbf{T}^t$ will be much smaller than $q/2$ in the absolute value with overwhelming probability [14]. Hence, we have $\mathbf{T}(\mathbf{2X} + \mathbf{B})\mathbf{T}^t \mod q = \mathbf{T}(\mathbf{2X} + \mathbf{B})\mathbf{T}^t$ over the integers.

**wRSR from LWE Problem**

The encryption scheme is very similar to the BGN-Type cryptosystem. The main constraints on the parameters are given by the correctness requirement and hardness requirements ($\beta$ should be large enough such that we can invoke above theorem).

- $q \in \left(2^{4(\log n)^2-1}, 2^{4(\log n)^2}\right)$ is a prime modulus.
- Entries of the error matrix $\mathbf{X}$ in $Enc_{pk}$ are chosen independently according to $\overline{\psi}_\beta(q)$, where

$$\beta = \frac{2^{-2(\log n)^2}}{20m \cdot (\log_2 n - 1) \cdot (20n \log_2 q)^2}.$$

- $\mathcal{I} = \left\{[-2^{3(\log n)^2}, -2^{3(\log n)^2-1}] \cup [2^{3(\log n)^2-1}, 2^{3(\log n)^2})]\right\} \cap \mathbb{Z}$.
- $\mathcal{R}' = \{\mathbf{AW} + 2\mathbf{X} \bmod q : \mathbf{X} \in \mathcal{I}^{m\times m}, \mathbf{W} \in \mathbb{Z}_q^{n\times m}\}$.

These parameters yield an approximation factor of $\tilde{O}(n/\alpha) = \tilde{O}(n^{O(\log n)})$, for lattice problems such as ($\mathbf{GAPSVP}_\gamma$). The best known algorithms for ($\mathbf{GAPSVP}_\gamma$) for $\gamma = \tilde{O}(n^{O(\log n)})$, runs in $2^{\tilde{\Omega}(n)}$.

- $\widehat{\mathcal{M}} = \mathcal{M}$, $\widehat{\mathcal{C}} = \mathcal{C} \times \widehat{\mathcal{M}}$.
- The distribution $\chi$ on $\widehat{\mathcal{C}}$ is induced by picking $\mathbf{r} \xleftarrow{uniform} \mathcal{R}'$, $\mathbf{B}' \xleftarrow{uniform} \widehat{\mathcal{M}}$ and outputting $\mathbf{R} = (\mathbf{r}, \mathbf{B}')$.
- $\mathcal{S}_{pk}(\mathbf{R}, \mathbf{C}) := (\mathbf{r} + \mathbf{B}' + \mathbf{C}) \bmod q$.
- $\widehat{Dec}_{sk} := Dec_{sk}$.
- $\mathcal{S}'_{pk}\left(\mathbf{R}, \widehat{\mathbf{B}}\right) := \left(\mathbf{B}' + \widehat{\mathbf{B}}\right) \bmod 2$.

**wRSR Encryption Scheme from LWE**

**Theorem 9** *Let $n > 339$ be any integer, $q \in \left(2^{4(\log n)^2-1}, 2^{4(\log n)^2}\right)$ be any prime and $\beta = \frac{2^{-2(\log n)^2}}{20m\cdot(\log_2 n-1)\cdot(20n\log_2 q)^2}$. Then $\widehat{Dec}_{sk}$ correctly decrypts with overwhelming probability. Furthermore the above* **LWE** *construction is a* **wRSR** *encryption scheme.*

**Proof** $\widehat{Dec}_{sk}(\mathcal{S}_{pk}(\mathbf{R}, \mathbf{C}))$ will decrypt to $(\mathbf{B} + \mathbf{B}') \bmod 2$, as long as

$$\|\mathbf{T}(2(\mathbf{X} + \mathbf{X}') + (\mathbf{B} + \mathbf{B}'))\mathbf{T}^t\|_\infty < q/2.$$

With overwhelming probability every entry of $\mathbf{T}(\mathbf{X})$ and $\mathbf{T}(\mathbf{B} + \mathbf{B}')$ is at most $40\beta q(\log_2 n - 1)n\log_2 q$ and $40n\log_2 q$. Therefore with overwhelming probability

$$\|\mathbf{T}(2(\mathbf{X} + \mathbf{X}') + (\mathbf{B}' + \mathbf{B}))\mathbf{T}^t\|_\infty < m(40n\log_2 q)^2 \cdot \left(\beta q(\log_2 n) + 2^{3(\log n)^2}\right).$$

from tail inequality $\beta q(\log_2 n) < 2^{2(\log n)^2}$, with overwhelming probability and $m = \lfloor 8n\log q \rfloor$ and $\log q = 4(\log n)^2$, therefore

$$\|\mathbf{T}(2(\mathbf{X}+\mathbf{X}')+(\mathbf{B}'+\mathbf{B}))\mathbf{T}^t\|_\infty < n^3(16\log n)^6\cdot\left(2^{2(\log n)^2}+2^{3(\log n)^2}\right).$$

But $(40n)^6\log_2 n\cdot\left(2^{2(\log n)^2}+2^{3(\log n)^2}\right) < q/2$ for all $n > 339$, hence

$$\|\mathbf{T}(2(\mathbf{X}+\mathbf{X}')+(\mathbf{B}'+\mathbf{B}))\mathbf{T}^t\|_\infty < 2^{n-1} < \tfrac{q}{2}.$$

**wRSR Properties** (**Semi-Honest Case**). The scheme clearly satisfies the first property. The scheme also satisfies the third property whenever $\widehat{Dec}_{sk}$ is correct, which will be the case with overwhelming probability for above choice of parameters. To prove the construction satisfies the second property let $\mathcal{S}_{pk}(\mathbf{R},\mathbf{C}) = \mathbf{AW}+\mathbf{E}+2\mathbf{Z}+\mathbf{C} \mod q$, and $\mathcal{S}_{pk}(\mathbf{R}',\mathbf{C}') = \mathbf{AW}'+\mathbf{E}'+2\mathbf{Z}'+\mathbf{C}' \mod q$. Since, $\mathbf{C}$ and $\mathbf{C}'$ are in the ciphertext space, there exist matrices $\mathbf{S},\mathbf{S}' \in \mathbb{Z}_q^{n\times m}$, $\mathbf{B},\mathbf{B}' \in \mathbb{Z}_2^{m\times m}$ and $\mathbf{X},\mathbf{X}' \in \mathbb{Z}_q^{m\times m}$, such that

$$\mathbf{C} = \mathbf{AS}+\mathbf{B}+2\mathbf{X} \mod q \text{ and } \mathbf{C}' = \mathbf{AS}'+\mathbf{B}'+2\mathbf{X}' \mod q$$

note that as far as each entry in $2(\mathbf{Z}+\mathbf{X})+(\mathbf{E}+\mathbf{B})$ and $2(\mathbf{Z}'+\mathbf{X}')+(\mathbf{E}'+\mathbf{B}')$ lie in the interval $\mathcal{I}$, $\mathcal{S}_{pk}(\mathbf{R},\mathbf{C})$ and $\mathcal{S}_{pk}(\mathbf{R}',\mathbf{C}')$ remain perfectly indistinguishable. The probability that each entry in $2(\mathbf{Z}+\mathbf{X})+(\mathbf{E}+\mathbf{B})$ and $2(\mathbf{Z}'+\mathbf{X}')+(\mathbf{E}'+\mathbf{B}')$ does not lies in $\mathcal{I} = \{[-2^{3(\log n)^2}, -2^{3(\log n)^2-1}] \cup [2^{3(\log n)^2-1}, 2^{3(\log n)^2})]\} \cap \mathbb{Z}$ is

$$\left(\tfrac{2^{3(\log n)^2}-(2^{3(\log n)^2}+\|\mathbf{X}\|_\infty+2)}{2^{3(\log n)^2-1}} + \tfrac{2^{3(\log n)^2}-(2^{3(\log n)^2}+\|\mathbf{X}'\|_\infty+2)}{2^{3(\log n)^2-1}}\right) = \tfrac{\|\mathbf{X}\|_\infty+\|\mathbf{X}'\|_\infty+4}{2^{3(\log n)^2-1}}.$$

Furthermore with overwhelming probability $\|\mathbf{X}\|_\infty$ and $\|\mathbf{X}'\|_\infty$ are at most $2^{2(\log n)^2}$, therefore with overwhelming probability

$$\tfrac{\|\mathbf{X}\|_\infty+\|\mathbf{X}'\|_\infty+4}{2^{3(\log n)^2-1}} = \tfrac{2^{2(\log n)^2+1}+4}{2^{3(\log n)^2-1}} < 2^{-(\log n)^2+1}$$

which is negligible in $n$. Therefore $\mathcal{S}_{pk}(\mathbf{R},\mathbf{C})$ and $\mathcal{S}_{pk}(\mathbf{R}',\mathbf{C}')$ are statistically indistinguishable.

# Chapter 4
# Zero-Knowledge Interactive Proof Systems for Lattice Problems

## 4.1 Interactive Proof Systems

Speaking informally, an **Interactive Proof System** (**IP**) is a challenge-response protocol between two parties in which one party, called the prover, tries to prove a certain statement to the other party, called the verifier [28]. Initially, both parties are given an input $x$. The objective of the **IP** is for the prover to convince the verifier that $x$ satisfy some specified property. For example, $x$ belongs to some language $L$. An **IP** consist of a specified (any polynomially in the size of $x$) number of rounds. During each round, the prover and the verifier alternately do the following:

1. Receive a message from the other party.
2. Perform a private computation.
3. Send a message to the other party.

A typical round of an **IP** consists of a challenge from the receiver and a response by the prover. At the end of the protocol the verifier either **accepts** or **rejects** the claim. The **IP** must satisfy two properties:

- **Completeness**: The verifier always accepts the proof if the statement is true and both parties follow the protocol correctly.
- **Soundness**: If the statement is false, no prover can convince the verifier that it is true, except with some small probability.

**Definition 6** *An interactive proof system with soundness error $s \in [0,1]$ and completeness $c \in [0,1]$, for a language $L \subseteq \{0,1\}^*$ is a pair of algorithms: a prover $P$ (possibly computationally unbounded) and a probabilistic polynomial-time verifier $V$, with the following properties.*

- **Completeness**: *For all inputs $x$ in $L$, the verifier after interacting with the prover ($[P(x) \leftrightarrow V(x)]$) accepts the proof (i.e. $out_V[P(x) \leftrightarrow V(x)] = 1$) with probability at least $c$*

$$\forall x \in L, \ \ \mathbf{Pr}[out_V[P(x) \leftrightarrow V(x)] = 1] \geq c.$$

- **Soundness**: *For every computationally unbounded $P^*$,*

$$\forall x \notin L, \ \ \mathbf{Pr}\left[out_V[P^*(x) \leftrightarrow V(x)] = 1\right] \leq s.$$

It is easy to verify that the interactive proof system, for the linear code equivalence problem, defined below has perfect **completeness** ($c = 1$) and **soundness** error $s = 2^{-l}$. Note that, if $\mathbf{G}_1$ and $\mathbf{G}_2$ are not equivalent then the only way for the prover to deceive the verifier is for him to guess correctly the verifier's choice $j$.

**An IP for Linear Code Equivalence Problem (LCE).**

Input: Generating matrices $\mathbf{G}_1$, $\mathbf{G}_2 \in \mathbb{F}_q^{k \times n}$.

1. Repeat the following steps for $1 \leq i \leq l$.
   (a) Prover picks uniformly an invertible matrix $\mathbf{N}_i \in \mathbf{F}_q^{k \times k}$ and a monomial matrix $P_i \in \mathcal{P}(n, \mathbb{F}_q)$.
   (b) Prover computes $\mathbf{H}_i \rightarrow \mathbf{N}_i \mathbf{G}_1 P_i$ and sends $\mathbf{H}_i$ to the verifier.

(c) Verifier picks uniformly $j_i \in \{1, 2\}$ and sends it to the prover.

(d) Prover sends a non-singular matrix $\mathbf{M}_i \in \mathbf{F}_q^{k \times k}$ and a matrix $P_i' \in \mathcal{P}(n, \mathbb{F}_q)$.

i. If $j_i = 1$, then $\mathbf{M}_i = \mathbf{N}_i$ and $P_i' = P_i$.

ii. Else $\mathbf{M}_i = \mathbf{N}_i\mathbf{N}^{-1}$ and $P_i' = P^{-1}P_i$, where $\mathbf{N} \in \mathbf{F}_q^{k \times k}$ is an invertible matrix and $P \in \mathcal{P}(n, \mathbb{F}_q)$ such that $\mathbf{G}_2 = \mathbf{N}\mathbf{G}_1 P$.

2. Verifier will accept the proof if for all $l$ rounds $\mathbf{H}_i = \mathbf{M}_i \mathbf{G}_{j_i} P_i'$.

## 4.2 Zero-Knowledge Property

For cryptographic applications it is useful for interactive proof systems to have the zero-knowledge property. Speaking informally an interactive proof is zero-knowledge if the verifier cannot not learn anything as a result of the protocol, except the validity of the statement. Moreover at the end of the proof the verifier will have no idea of how to prove himself that the statement is true. Zero-knowledge interactive proofs have many applications in cryptography, such as identification schemes, multiparty computations, etc. We formally prove the zero-knowledge property of an **IP** by a technique known as *simulation*. Before we give the formal definition of zero-knowledge interactive proof systems (**ZKIP**), we will briefly study the above **IP** for **LCE**. What the verifier learns as a result of this proof can be represented as the following transcript

$$T = ((\mathbf{G}_1,\ \mathbf{G}_2); (\mathbf{H}_1, j_1, \mathbf{M}_1, P_1'); \ldots; (\mathbf{H}_l, j_l, \mathbf{M}_l, P_l'))\,.$$

The **IP** for **LCE** will be zero-knowledge if there exits a probabilistic polynomial-time algorithm (simulator) that without participating in the proof can forge transcripts that **look like** real transcripts. The fact that a simulator can forge transcripts has a very important consequence. Anything that the verifier computes can also be computed from a forged transcript. Therefore, participating in the proof system does not enable the verifier to prove that $x$ is in $L$. Hence, the verifier cannot convince someone else by showing him the transcript $T$, since anyone can forge a transcript that looks like a real transcript.

### 4.2.1 Zero-Knowledge Interactive Proofs

**Definition 7** *Suppose that we have a polynomial-time interactive proof system* $[P \leftrightarrow V]$ *for a language* $L \subseteq \{0,1\}^*$. *Let* $V^*$ *denote a (possibly cheating) verifier. Let* $\mathcal{T}(V^*, R_{V^*}, x)$ *be the set of all possible transcripts that could be produced as the result of an interactive proof* $[P \leftrightarrow V^*]$ *on input* $x \in L$ *and* $R_{V^*}$ *be its random coins. Suppose for every such* $V^*$, *there exists an expected polynomial-time simulator* $S$ *and let* $\mathcal{T}(S, R_S, x)$ *denote the set of all possible simulated transcripts that could be produced by* $S$. *Let* $\mathbf{Pr}_{V^*}(\mathcal{T})$ *denote the probability distribution on* $\mathcal{T}(V^*, R_{V^*}, x)$ *as a result of* $[P \leftrightarrow V^*]$, *and* $\mathbf{Pr}_S(\mathcal{T})$ *be the probability distribution on* $\mathcal{T}(S, R_S, x)$ *induced by* $S$. *If the distributions* $\mathbf{Pr}_S(\mathcal{T}) = \mathbf{Pr}_{V^*}(\mathcal{T})$, *then we define the interactive proof to be perfect zero-knowledge* (**PZKIP**) *and if the distributions* $\mathbf{Pr_S}(\mathcal{T})$ *and* $\mathbf{Pr}_{V^*}(\mathcal{T})$ *are statistically close (in the size of the input* $x$*) then we define the interactive proof to be statistical zero-knowledge* (**SZKIP**).

### 4.2.2 Interactive Proofs with Efficient Provers

The definition of **IP** allows the prover to be unbounded. However, for the real world applications, we would like the prover to be efficient. Of course, if the prover is given just the same inputs as the verifier, then it cannot accomplish anything that the verifier cannot accomplish itself. But in many proof systems, the prover can be made efficient by giving it some extra knowledge. For example in the **LCE** proof, the prover can be made efficient by giving him the matrices $\mathbf{N} \in \mathbf{G}_k(\mathbb{F}_q)$ and $P \in \mathcal{P}(n, \mathbb{F}_q)$.

## 4.3 Lattices and Zero-Knowledge Interactive Proofs

The first **IP** for lattice problems was presented by Goldreich and Goldwasser [42]. They show that the complement problems $\mathbf{coGapCVP}_\gamma$ and $\mathbf{coGapSVP}_\gamma$ have constant round interactive proofs. However, these proofs are only honest verifier perfect zero-knowledge and known to have inefficient provers. Micciancio and Vadhan [29] presented interactive proofs for $\mathbf{GapCVP}_\gamma$ and $\mathbf{GapSVP}_\gamma$. These proofs are statistical zero-knowledge and have efficient provers as well.

## 4.4 Isometric Lattice Problem ILP

In this section 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 **ILP** is at least as hard as **Graph Isomorphism** and **Linear Code Equivalence** over $\mathbb{F}_p$. This is the only hard problem known in lattices that has a *malicious* verifier perfect zero-knowledge **IP** system with an *efficient* prover. We also show that **ILP** is unlikely to be **NP-complete**. To do this we present a constant round **IP** system for the

complementary problem (**co-ILP**) of **ILP**. Furthermore, the proof system for the complementary problem is *honest* verifier *perfect* zero-knowledge.

### 4.4.1 Isometric Lattices

**Definition 8** *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 $\mathcal{L}(\mathbf{B}_1) \cong \mathcal{L}(\mathbf{B}_2)$.

## 4.5 Variants of ILP

Let

$$\mathbf{S}_{(\mathbf{B}_1,\mathbf{B}_2)} = \{\mathbf{B} \in \mathbb{R}^{n\times k} : \mathcal{L}(\mathbf{B}) \cong \mathcal{L}(\mathbf{B}_1) \,\&\, \mathcal{L}(\mathbf{B}) \cong \mathcal{L}(\mathbf{B}_2)\}$$

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**. Therefore, it is natural to ask if one can obtain a **PZKIP** for **ILP** by mimicking the **LCE** proof system. [1] However, if we try to mimic the proof system for **LCE** we will face 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 $(GL_k(\mathbb{Z}))$

1. The **IP** for **LCE** is **PZKIP** with an efficient prover see appendix A.

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 $O(n, \mathbb{R})$[43, 50], 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 verifier.

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 $\overline{O(n,\mathbb{R})} \leq O(n,\mathbb{R})$.

- Let $Q \in O(n,\mathbb{R})$. We add $Q$ in $\overline{O(n,\mathbb{R})}$

$$\overline{O(n,\mathbb{R})} \leftarrow \overline{O(n,\mathbb{R})} \cup \{Q\}.$$

- Since $\overline{O(n,\mathbb{R})}$ has to be a multiplicative group, we must add $Q \cdot Q$ and $Q^{\mathbf{T}}$ to it. Hence

$$\overline{O(n,\mathbb{R})} \leftarrow \overline{O(n,\mathbb{R})} \cup \{Q \cdot Q\} \cup \{Q^{\mathbf{T}}\}.$$

- By the same argument $Q \cdot Q \cdot Q$ and $Q^{\mathbf{T}} \cdot Q^{\mathbf{T}}$ must also be added to $\overline{O(n,\mathbb{R})}$. Hence, this process may never end and $\overline{O(n,\mathbb{R})}$ 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 these variants are at least hard as **GI** and **LCE**. We further show that these variants are unlikely to be **NP-complete** unless the polynomial hierarchy collapses [45, 46].

### 4.5.1 Isometric Lattices over $\mathbb{Z}$

**Definition 9** *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*

$$\mathbf{B}_2 = Q\mathbf{B}_1 U.$$

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

### 4.5.2 Isometric Lattices over $\mathbb{R}_{\mathcal{Q}} \subset \mathbb{R}$

**Definition 10** *Let $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}_{\mathcal{Q}}^{n\times k}$ be two bases of rank $k$. We say that two lattices $\mathcal{L}(\mathbf{B}_1) \cong_{\mathbb{R}_{\mathcal{Q}}} \mathcal{L}(\mathbf{B}_2)$ are isometric over $\mathbb{R}_{\mathcal{Q}}$ if there exists a matrix $U \in GL_k(\mathbb{Z})$ and a matrix $Q \in \overline{O(n, \mathbb{R}_{\mathcal{Q}})}$ such that*

$$\mathbf{B}_2 = Q\mathbf{B}_1 U.$$

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

Note that sets $\mathbb{R}_{\mathcal{Q}}$ and $\overline{O(n, \mathbb{R}_{\mathcal{Q}})}$ are defined in section 4.6.

## 4.6 The Set $\mathbb{R}_{\mathcal{Q}}$

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

$$r = a_1 < n_{11}, x_{11}+ < n_{21}, x_{21} + \cdots + < n_{k1}, x_{k1} >> \cdots > +$$

$$a_2 < n_{12}, x_{12}+ < n_{22}, x_{22} + \cdots + < n_{k2}, x_{k2} >> \cdots > +$$

$$\cdots + a_l < n_{1l}, x_{1l}+ < n_{2l}, x_{2l} + \cdots + < n_{kl}, x_{kl} >> \cdots > .$$

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

### 4.6.1 The Set $\overline{O(n, \mathbb{R}_{\mathcal{Q}})}$

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

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

Let $\mathcal{P}$ be any desired publicly known positive polynomial in the size of the input bases $\mathbf{B}_1, \mathbf{B}_2 \in O(n, \mathbb{R}_{\mathcal{Q}})$ and $\delta = \frac{\pi}{2^{\mathcal{P}}}$. We denote $C$ the set of angles

$$C = \{0, \delta, 2\delta, \ldots, \theta, \ldots, 2\pi - \delta\}$$

We denote $\overline{O(n, \mathbb{R}_{\mathcal{Q}})}$ 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

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

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

where $x = \frac{n}{2}$ if $n$ is even, otherwise $x = \frac{n-1}{2}$. Clearly $\overline{O(n, \mathbb{R}_{\mathcal{Q}})}$ 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 $0 \leq n \leq 2^{\mathcal{P}}$ $\sin(\frac{n\pi}{2^{\mathcal{P}}})$ and $\cos(\frac{n\pi}{2^{\mathcal{P}}})$ can be computed in $O(\mathcal{P})$ time (see appendix B). Let $Q \in \overline{O(n, \mathbb{R}_{\mathcal{Q}})}$,

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

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

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

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

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

Hence, $Q' = Q^{\mathbf{T}}$.

### 4.7 Interactive Proof for Isometric Lattice Problem over Integers $\mathbf{ILP}_{\mathbb{Z}}$

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 [44]. 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 10** *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 $\mathcal{L}(\mathbf{B}_1)$ andd $\mathcal{L}(\mathbf{B}_2)$ are not isometric over integers, then the only way for the prover to cheat is to guess $j$ correctly in each round. Since, $j$ is chosen uniformly and independently from $\{1, 2\}$, the probability of prover guessing $j$ 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 [44]. Therefore the expected running time of the prover is polynomial.

**Zero-Knowledge**: Let $V^*$ be any probabilistic polynomial time (a 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 $V^*$ and $S$ 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 $\mathbf{ILP}_{\mathbb{Z}}$.

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

$$\frac{1 - p}{2^n n!}$$

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

$$\mathbf{Pr}_{V^*}(\mathcal{T}) = \mathbf{Pr}_S(\mathcal{T}).$$

### 4.8 Sampling a Lattice Basis in Zero-Knowledge and $\mathbf{ILP}_{\mathbb{R}_\mathcal{Q}}$

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 the algorithm Sample**Point** (chapter 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 $D_{s,\mathbf{c},\mathcal{L}}$ (chapter 2). Sample**Basis** is zero-knowledge in a sense that the output point $\mathbf{v}$ leaks al most no information about the input basis $\mathbf{B}$ except the bound $s$ with overwhelming probability [21]. 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 ([33], 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 $1 \leq i \leq k$ (chapter 2).

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.

**Algorithm 5** Sample$\mathcal{L}$

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

1. Set $t(n) = \log n$.
2. Sample $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n^2}\}$ points independently using the algorithm Sample**Point**$(\mathbf{B}, 0, s, t(n))$.
3. 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.
4. 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 $\mathbf{B} \in \mathbb{R}_{\mathcal{Q}}^{n \times k}$ then so $\mathbf{T} \in \mathbb{R}_{\mathcal{Q}}^{n \times k}$. Since $\mathbf{T}$ and $\mathbf{B}$ are bases of the same lattice, there exists a $U \in GL_k(\mathbb{Z})$ such that

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

## 4.9 An Interactive Proof for $\mathbf{ILP}_{\mathbb{R}_\mathcal{Q}}$

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

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 $Q'_j \leftarrow \overline{O(n, \mathbb{R}_\mathcal{Q})}$.
   (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 $P_j \in \overline{O(n, \mathbb{R}_\mathcal{Q})}$.
      i. if $c_j = 1$, then $P_j = Q'_j$.
      ii. if $c_j = 2$ then $P_j = Q'_jQ^{\mathbf{T}}$, where $Q \in \overline{O(n, \mathbb{R}_\mathcal{Q})}$ 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 11** *The proof system for* $\mathbf{ILP}_{\mathbb{R}_\mathcal{Q}}$ *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 $\mathbf{B}_2 = Q\mathbf{B}_1 U$ for some $Q \in \overline{O(n, \mathbb{R}_{\mathcal{Q}})}$ and $U \in GL_k(\mathbb{Z})$. Clearly,

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

since $\mathbf{B}'_j = Q'_j \mathbf{B}_1 U'_j$ and $\mathbf{B}'_j = Q'_j Q^{\mathbf{T}} \mathbf{B}_2 U U'_j$ 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 $\mathbb{R}_{\mathcal{Q}}$, 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 $\mathbf{ILP}_{\mathbb{R}_{\mathcal{Q}}}$. Let $S_{\mathbb{R}_{\mathcal{Q}}}$ denote the simulator, which will produce the possible set of forged transcripts $\mathcal{T}(S_{\mathbb{R}_{\mathcal{Q}}})$. We denote $\mathbf{Pr}_{V^*}(\mathcal{T})$ the probability distribution on $\mathcal{T}(V^*)$ and we denote $\mathbf{Pr}_{S_{\mathbb{R}_{\mathcal{Q}}}}(\mathcal{T})$ the probability distribution on $\mathcal{T}(S_{\mathbb{R}_{\mathcal{Q}}})$. We will prove that:

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

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

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 $\in \overline{O(n, \mathbb{R}_{\mathcal{Q}})}$.
      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** $S_{\mathbb{R}_{\mathcal{Q}}}$ for $\mathbf{ILP}_{\mathbb{R}_{\mathcal{Q}}}$.

*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 $S_{\mathbb{R}_{\mathcal{Q}}}$ generate for every triple it concatenates to $T$? We note that $Q'Q^{\mathbf{T}}$ and $Q'$ are uniformly distributed over $\overline{O(n, \mathbb{R}_{\mathcal{Q}})}$, 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 $S_{\mathbb{R}_\mathcal{Q}}$ is roughly twice the running time of $V^*$. By definition $V^*$ runs in probabilistic polynomial time. Hence the running time of $S_{\mathbb{R}_\mathcal{Q}}$ is also expected polynomial time.

We will prove that the two probability distributions $\mathbf{Pr}_{V^*}(\mathcal{T})$ and $\mathbf{Pr}_{S_{\mathbb{R}_\mathcal{Q}}}(\mathcal{T})$ 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 [27], 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 $O(n, \mathbb{R}_\mathcal{Q})$ and $S_{\mathbb{R}_\mathcal{Q}}$ also picks $Q'_1$ uniformly over $O(n, \mathbb{R}_\mathcal{Q})$. Hence both $P'_1$ and $Q'_1$ are identically distributed. Also $\mathbf{B}'_1$ and $\mathbf{H}'_1$ computed by Sample$\mathcal{L}$. Therefore they are almost identically distributed.

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

$$\mathbf{Pr}_{V^*}(\mathcal{T}) \sim \mathbf{Pr}_{S_{\mathbb{R}_\mathcal{Q}}}(\mathcal{T}).$$

## 4.10 Isometric Lattice Problem is not Easy

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

**Theorem 12 ILP** *is at least as hard as* **LCE** (*Linear Code Equivalence problem*) *over prime fields* $\mathbb{F}_p$.

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

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

$$\mathbf{G} \longrightarrow \mathbf{B}$$

where $\Lambda_2(\mathbf{G})$ be the corresponding $p$-ary lattice. Recall from chapter 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 $\Lambda_p(\mathbf{G})$. Where $\mathbf{b}_j = (0, ..., p, ..., 0) \in \mathbb{Z}^n$ and the $j$-th coordinate is equal to $p$, for $k + 1 \leq j \leq n$. Clearly the map $\psi$ can be computed in polynomial time. Let $\mathbf{G}_1 = [\mathbf{g}_{11}|\ldots|\mathbf{g}_{1k}] \in \mathbb{F}_p^{n\times k}$ and $\mathbf{G}_2 = [\mathbf{g}_{21}|\ldots|\mathbf{g}_{2k}] \in \mathbb{F}_p^{n\times k}$ 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 $\mathbf{G}_2 = P\mathbf{G}_1'M$, where $\mathbf{G}_1' = D\mathbf{G}_1$ and let $\Lambda_p(\mathbf{G}_1')$ and $\Lambda_p(\mathbf{G}_2)$ be corresponding lattices.

$$\text{For any } \mathbf{v} \in \Lambda_p(\mathbf{G}_2) \Longleftrightarrow \mathbf{v} \equiv \mathbf{G}_2 \cdot \mathbf{s} \pmod{p}, \text{ for some } \mathbf{s} \in \mathbb{Z}^k$$

$$\Longrightarrow \mathbf{v} \equiv P\mathbf{G}_1'M \cdot \mathbf{s} \pmod{p} \equiv P\mathbf{G}_1' \cdot \mathbf{s}' \pmod{p},\ \mathbf{s}' = M\mathbf{s} \in \mathbb{Z}^k$$

$$\Longrightarrow \mathbf{v} \in \Lambda_p(P\mathbf{G}_1')$$

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

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

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

$\Longleftarrow$ 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,

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

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

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

**Theorem 13** $\mathbf{ILP}_{\mathbb{Z}}$ *is at least as hard as the* **GI** (*Graph Isomorphism*) *problem.*

**Proof** Petrank and Roth [47] reduced **GI** to **PCE** (Permutation Code Equivalence). More precisely they provided a polynomial time mapping $\phi$ 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 $\mathbf{G} = [\mathbf{g}_1|\dots|\mathbf{g}_k] \in \mathbb{F}_2^{n\times k}$ be a basis of some $[k, n]$ linear code $C$

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

$$\mathbf{G} \longrightarrow \mathbf{B}$$

where $\Lambda_2(\mathbf{G})$ is the corresponding 2-ary lattice. Recall from chapter 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 $\Lambda_2(\mathbf{G})$. 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 $\psi$ can be computed in polynomial time. Let $\mathbf{G}_1 = [\mathbf{g}_{11}|\dots|\mathbf{g}_{1k}] \in \mathbb{F}_2^{n\times k}$ and $\mathbf{G}_2 = [\mathbf{g}_{21}|\dots|\mathbf{g}_{2k}] \in \mathbb{F}_2^{n\times k}$ 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, $\Lambda_2(Q\mathbf{G}_1) = \Lambda_2(\mathbf{G}_2)$, we also have

$$\mathbf{v} \equiv (Q\mathbf{G}_1)\mathbf{u} \equiv (P\mathbf{G}_1)\mathbf{u} \pmod 2 \text{ for some } \mathbf{u} \in \mathbb{Z}^k.$$

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**.

### 4.10.1 ILP is unlikely to be NP-complete

In this section we show that $\mathbf{ILP}_{\mathbb{S}}$ is unlikely to be NP-complete (where $\mathbb{S} = \mathbb{Z}$ or $\mathbb{S} = \mathbb{R}_{\mathcal{Q}}$ see section 4.5). We do this by constructing a constant round interactive proof for the **Non-Isometric Lattice problem** ($\mathbf{co\text{-}ILP}_{\mathbb{S}}$), 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 $\Pi$ has a constant round interactive proof and $\Pi$ is NP-complete then the polynomial hierarchy collapses [45, 46]. 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 $\mathbf{co\text{-}ILP}_{\mathbb{S}}$**

– 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

$$Q_1, \ldots, Q_l \in O(n, \mathbb{R}_{\mathcal{Q}}).$$

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 4.8.
5. For $1 \leq i \leq l$, the all-powerful prover 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 $\mathbb{S} = \mathbb{R}_{\mathcal{Q}}$.

## Chapter 5
## Conclusion and Future Work

The objective of this thesis has been to study and construct cryptographic primitives under the assumption that an adversary has the power of quantum computing. We proposed a new notion of weakly Random-Self-Reducible encryption scheme and show that one can obtain a secure OT, under a sole assumption that a of **wRSR** encryption scheme exits. We provided concrete instantiation of **wRSR** from two different post quantum assumptions:

– Approximate Integer GCD.

– Learning with Errors.

We also presented a new hard problem from lattices called the Isomteric Lattice Problem **ILP**. We showed that **ILP** is at least as hard as Graph Isomorphism and Linear Code Equivalence. We presented two variants of **ILP** (one over the integers and other over rational radicands and proved that these variants have a Perfect Zero-knowledge and a Statistical Zero-Knowledge interactive proof systems with an efficient prover for **ILP**. Interestingly this is the only problem known from integer lattices to have a Perfect Zero-Knowledge interactive proof system with an efficient prover. We further showed that this problem is unlikely to be NP-complete unless the polynomial hierarchy collapses.

The research I have completed thus far has raised some interesting questions, such as whether the **wRSR** property is more general than the **Dual** mode property

[34]? Are there secure encryption schemes based on linear codes that satisfy the **wRSR** property? I would like to continue my research to investigate these questions further.

I would also like to explore the possibility of constructing a public-key encryption scheme whose security is as hard as the problem of decoding random linear codes: The fundamental security assumption in present code-based cryptosystems is that decoding a random linear code is hard on average, but there is no known proof that breaking current code-based encryption schemes is as hard as the assumption. And perhaps it may very well be the case that breaking code-based systems is easier than decoding random linear codes. The code-based cryptosystems are constructed from particular families of codes that have efficient decoding algorithms and good distances. Hence, breaking these schemes would not necessarily imply breaking schemes whose security is equal to general decoding of linear codes.

Bibliography

[1] Peter W Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, volume 26, issue 5, pages 1484 – 1509 (1997).

[2] Ron Rivest, Adi Shamir, and Leonard Adleman. *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, volume 21, issue 2, pages 120 – 126 (1978).

[3] Taher ElGamal. *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. IEEE Transactions on Information Theory, volume 31, issue 4, pages 469 – 472 (1985).

[4] Stephen J. Wiesner. *Conjugate Coding*. SIGACT News, volume 15, issue 1, pages 78 – 88 (1983).

[5] Gilles Brassard and Claude Crépeau. *Zero-knowledge simulation of Boolean circuits*. In CRYPTO, pages 223 – 233 (1986).

[6] Christpher Peikert. *Public-key cryptosystems from the worst-case shortest vector problem*. In STOC, pages 333 – 342 (2009).

[7] Vadim Lyubashevsky and Daniele Micciancio. *Asymptotically efficient lattice-based digital signatures*. In TCC, pages 37 – 54 (2008).

[8] Xavier Boyen. *Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more*. In PKC, pages 499 – 517 (2010).

[9] Samuel Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. *A group signature scheme from lattice assumptions*. In ASIACRYPT, pages 395 – 412 (2010).

[10] Craig Gentry. *Fully homomorphic encryption using ideal lattices*. In STOC, pages 169 – 178 (2009).

[11] Claude Crépeau. *Quantum Oblivious Transfer*. Journal of Modern Optics, special issue on Quantum Communication and Cryptography, volume 41, number 12, pages 2445 – 2454 (1994).

[12] Nick Howgrave-Graham. *Approximate integer common divisors*. In CaLC, pages 51– 66 (2001).

[13] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. *Fully Homomorphic Encryption over the Integers*. In EUROCRYPT, pages 24 – 43 (2010).

[14] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. *A simple BGN-Type Cryptosystem from Learning with Errors*. In EUROCRYPT, pages 506 – 522 (2010).

[15] Zvika Brakerski and Vinod Vaikuntanathan. *Efficient Fully Homomorphic Encryption from (Standard) LWE*. In FOCS, pages 97 – 106 (2011).

[16] Zvika Brakerski and Vinod Vaikuntanathan. *Fully homomorphic encryption from ring-LWE and security for key dependent messages*. In CRYPTO, pages 505 – 524 (2011).

[17] Craig Gentry and Shai Halevi. *Fully homomorphic encryption without squashing using depth-3 arithmetic circuits*. In FOCS, pages 107 – 109 (2011).

[18] Zvika Brakerski, Craig Gentry and Vinod Vaikuntanathan. *Fully homomorphic encryption without bootstrapping*. In ITCS, pages 309 – 325 (2012).

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

[20] Craig Gentry, Amit Sahai and Brent Waters. *Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based*. In CRYPTO, pages 75 – 92 (2013).

[21] 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).

[22] David Cash, Dennis Hofheinz, Eike Kiltz and Chris Peikert. *Bonsai trees, or how to delegate a lattice basis*. In EUROCRYPT, pages 523 – 552 (2010).

[23] Shweta Agrawal, Dan Boneh, and Xavier Boyen. *Efficient lattice (H)IBE in the standard model*. In EUROCRYPT, pages 553 – 572 (2010).

[24] Christopher Peikert and Alon Rosen. *Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices*. In TCC, pages 145 – 166 (2006).

[25] Gilles Brassard, Claude Crépeau and Jean-Marc Robert. *All-or-nothing disclosure of secrets*. In CRYPTO, pages 224 – 238 (1986).

[26] 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).

[27] Oded Goldreich. *Foundations of Cryptography*. Volume I & II. Cambridge University Press, 2001 – 2004.

[28] 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).

[29] Daniele Micciancio and Salil P. Vadhan. *Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More*. In CRYPTO, pages 282 – 298 (2003).

[30] Christopher Peikert and Vinod Vaikuntanathan. *Noninteractive Statistical Zero-Knowledge Proofs for Lattice Problems*. In CRYPTO, pages 17 – 21 (2008).

[31] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade and Anderson C. A. Nascimento. *Oblivious Transfer Based on the McEliece Assumptions*. In proceedings of the 3rd international conference on Information Theoretic Security. Lecture Notes in Computer Science Volume 5155, pages 107 – 117 (2008).

[32] Kazukuni Kobara, Kirill Morozov and Raphael Overbeck. *Coding-Based Oblivious Transfer*. In Mathematical Methods in Computer Science. Lecture Notes in Computer Science, volume 5393, pages 142 – 156 (2008).

[33] Oded Regev. *On lattices, learning with errors, random linear codes, and cryptography*. In STOC, pages 84 – 93 (2005).

[34] Christopher Peikert, Vinod Vaikuntanathan and Brent Waters. *A Framework for Efficient and Composable Oblivious Transfer*. In CRYPTO, pages 554 – 571 (2008).

[35] Claude Crépeau. *Equivalence between two flavours of oblivious transfer*. In CRYPTO, pages 350 – 354 (1987).

[36] Richard Berger, Rene Peralta, and Tom Tedrick. *A provably secure oblivious transfer protocol*. In EUROCRYPT, pages 379 – 386 (1984).

[37] Michael Oser Rabin. *How to exchange secrets by oblivious transfer*. Technical Memo TR81, Aiken Computation Laboratory, Harvard University, (1981).

[38] Shimon Even, Oded Goldreich and Abraham Lempel. *A randomized protocol for signing contracts*. Communications of the ACM, volume 28, issue 6, pages 637 – 647 (1985).

[39] Michael J. Fischer, Silvio Micali and Charles Rackoff. *A secure protocol for the oblivious transfer*. Journal of Cryptology, volume 9, issue 3, pages 191 – 195 (1996).

[40] Shafi Goldwasser and Silvio Micali. *Probabilistic encryption*. Journal of Computer and System Sciences, volume 28, issue 2, pages 270 – 299 (1984).

[41] Pascal Paillier. *Public-key cryptosystems based on composite degree residuosity classes*. In EUROCRYPT, pages 538 – 554 (1999).

[42] Oded Goldreich and Shafi Goldwasser. *On the Limits of Non-Approximability of Lattice Problems*. In STOC, pages 23 – 26 (1998).

[43] 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).

[44] 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).

[45] Shafi Goldwasser and Michael Sipser. *Private coins versus public coins in interactive proof systems*. In STOC, pages 59 – 68 (1986).

[46] 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).

[47] 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).

[48] Daniele Micciancio and Christopher Peikert. *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller*. In EUROCRYPT, pages 700 – 718 (2012).

[49] Oded Goldreich, Amit Sahai and Salil Vadhan. *Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge*. In STOC, pages 399 – 408 (1998).

[50] George Marsaglia. *Choosing a Point from the Surface of a Sphere*. Annals of Mathematical Statistics, volume 43, number 2, pages 645 – 647 (1972).

[51] Eric Schmutz. *Rational points on the unit sphere*. Central European Journal of Mathematics, volume 6, issue 3, pages 482 – 487 (2008).

[52] Tanner and Thisted. *Applied Statistics*, pages 199 – 206 (1982).

[53] Hans Liebeck and Anthony Osborne. *The generation of all rational orthogonal matrices*. American Mathematical Monthly, volume 98, issue 2, pages 131 – 133 (1991).

[54] 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).

[55] 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).

[56] Robert J. McEliece. *A public-key cryptosystem based on algebraic coding theory*. Technical memo, California Institute of Technology (1978).

[57] Nicolas Sendrier. *The Tightness of Security Reductions in Code-based Cryptography*. IEEE Information Theory Workshop, pages 16 – 20 (2011).

## Appendix A
## A Perfect Zero-Knowledge Interactive Proof for LCE

**An IP for Linear Code Equivalence Problem (LCE).**

Input: Generating matrices $\mathbf{G}_1$, $\mathbf{G}_2 \in \mathbb{F}_q^{k\times n}$.

1. Repeat the following steps $l$ times.
   (a) Prover picks uniformly matrices $\mathbf{N}_i \in \mathbf{G}_k(\mathbf{F}_q)$ and $P_i \in \mathcal{P}(n, \mathbb{F}_q)$.
   (b) Prover computes $\mathbf{H}_i \rightarrow \mathbf{N}_i\mathbf{G}_1P_i$ and sends $\mathbf{H}_i$ to the verifier.
   (c) Verifier picks uniformly $c \in \{1, 2\}$ and sends to the prover.
   (d) Prover sends a non-singular matrix $\mathbf{W} \in \mathbb{F}_q^{k\times k}$ and a matrix $T \in \mathcal{P}(n, \mathbb{F}_q)$.
      i. If $c = 1$, then $\mathbf{W} = \mathbf{N}_i$ and $T = P_i$.
      ii. Else $\mathbf{W} = \mathbf{U}_i\mathbf{N}^{-1}$ and $T = P_iP^{-1}$.
2. Verifier will accept the proof if for all $l$ rounds $\mathbf{H}_i = \mathbf{W}\mathbf{G}_jT$.

**Theorem 14** *The proof system for (***LCE***) is a malicious verifier perfect-zero knowledge interactive proof with an efficient prover.*

**Proof** 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 the prover $P$ and $V^*$ carrying out the interactive proof on a yes instance $(\mathbf{G}_1, \mathbf{G}_2)$ of **LCE**. 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 $\mathcal{T}(S)$.

Input: Two generating matrices $\mathbf{G}_1$, $\mathbf{G}_2 \in \mathbb{F}_q^{k\times n}$, such that $\mathbf{G}_2 = \mathbf{N}\mathbf{G}_1 P$, for $\mathbf{N} \in \mathbb{F}_q^{k\times k}$ and $P \in \mathcal{P}(n, \mathbb{F}_q)$.

1. $T = (\mathbf{G}_1, \mathbf{G}_2)$.
2. **for** $j = 1$ **to** $l$ **do**
   (a) old state $\leftarrow$ state($V$)
   (b) repeat
      i. Pick uniformly $i \in \{1, 2\}$.
      ii. Pick uniformly matrices $\mathbf{N}' \in \mathbb{F}_q^{k\times k}$ and $P'$ from $\mathcal{P}(n, \mathbb{F}_q)$.
      iii. Compute $\mathbf{H}' \leftarrow \mathbf{N}'\mathbf{G}_i P'$.
      iv. Call $\mathbf{V}$ with input $\mathbf{H}'$ and obtain $i'$.
      v. **if** $i = i'$ **then**
         – Concatenate $(\mathbf{H}', i, \mathbf{N}', P')$ to the end of $T$.
         **else**
         – Set state($V$) $\leftarrow$ old state.

vi. until $i = i'$

**Simulator** $S$ for **LCE**.

By definition $V^*$ is PPT and the probability that $i = i'$ is $1/2$. Therefore, on the average $S$ will generate two quadruples $(\mathbf{H}', i, \mathbf{N}', P')$ for every quadruple it concatenates to the transcript $T$. 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 $T = (\mathbf{G}_1, \mathbf{G}_2)$, 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 see what happens at the $j$-th round of our interactive proof. 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 $\mathbf{H}_i$ uniformly (denote the probability by $u$) over its space.

This probability is independent of how verifier picks $c \in \{1, 2\}$. Therefore the probability that at the $j$-th round $(\mathbf{H}', i, \mathbf{N}', P')$ is on the transcript of the **IP**

if $c = 1$ is

$$p \cdot u$$

and if $c = 2$

$$\frac{1-p}{u}$$

The simulator $S$ in any round will pick an orthogonal matrix $\mathbf{H}'$ with probability $u$.

The probability that $i = 1$ and $i' = 1$ is

$$\frac{p}{2}$$

and the probability $i = 2$ and $i' = 2$ is

$$\frac{1-p}{2}.$$

In both cases the corresponding triple $(\mathbf{H}', i, \mathbf{N}', P')$ will be written on the transcript. Note that with probability 1/2 nothing is added to the transcript. The probability that $(\mathbf{H}', 1, \mathbf{N}', P')$ is written on the transcript in $j$-th round during the $m$-th iteration of the **repeat** loop is

$$\frac{p \cdot u}{2^m}.$$

Therefore the total probability that $(\mathbf{H}', 1, \mathbf{N}', P')$ is written on the transcript in the $j$-th round is

$$\begin{aligned}&\frac{p \cdot u}{2} + \frac{p \cdot u}{2^2} + ... + \frac{p \cdot u}{2^m} + ... + ... \\ &= \frac{p \cdot u}{2}\left(1 + \frac{1}{2} + \frac{1}{4} + ... + \frac{1}{2^{m-1}} + .... + ....\right) = \frac{p \cdot u}{2}.\end{aligned}$$

Similarly the total probability that $\left(\mathbf{H}'_j, 2, Q'_j\right)$ is written on the transcript in the $j$-th round is $\frac{(1-p)\cdot u}{2}$. Hence, by induction, the two probability distributions are identical

$$\mathbf{Pr}_{V^*}(\mathcal{T}) = \mathbf{Pr}_S(\mathcal{T}).$$

Clearly, $P$ the expected running time of the prover is polynomial.

## Appendix B
## 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 with 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.