School of Information Technology and
Engineering at the ADA University

School of Engineering and Applied Science
at the George Washington University

LATTICE-BASED IDENTITY-BASED ENCRYPTION: FOUNDATIONS, CONSTRUCTIONS AND IMPLEMENTATION

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Agha Aghayev

April, 2025

This Thesis by: Agha Aghayev
Entitled: ***Lattice-based Identity-based encryption: Foundations, Constructions and Implementation***

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

(Advisor)	(Date)
(Program Director)	(Date)
(Dean)	(Date)

Cryptographers seldom sleep at nights.

— Silvio Micali

ABSTRACT

Traditional public key encryption (PKE) is widely used over the internet and overcomes the key distribution problem. However, accessing the correct public keys of users still remain as a challenge. The Public Key Infrastructure (PKI) addresses this problem by managing the creation, distribution, secure storage of public keys, and their verification through trusted Certificate Authorities (CA). Moreover, Certificate Authorities can be corrupted and fake certificates can be issued.

Identity-Based Encryption (IBE) solves this problem by generating secret keys from user identities, enabling any string — such as an email address, phone number, or other identifier to serve as a valid public key for a user. Additionally, it encompasses several advantages over traditional public key encryption, offering easier key revocation and flexibility for delegation of keys. Hence, this study focuses on examining the underlying mathematical foundations of Identity-Based Encryption (IBE) schemes and analyzing their structural properties.

On the other hand, the recent developments in quantum computing pose a significant threat against contemporary cryptographic constructions, including PKE and IBE. This trend brings the necessity of developing quantum-resistant schemes, and Lattice-based cryptography has gained significant attention from researchers, as there is no known polynomial time algorithm to break its security. Additionally, operations on lattices are highly parallelizable and efficient, and more importantly, they offer strong **worst-case** security guarantees. Furthermore, our work focuses on the foundational principles of lattice-based cryptography and its application to Identity-Based Encryption (IBE) schemes. To understand the basic building blocks of lattice-based IBE schemes, we study two notions of **trapdoor sampling** - a critical step in lattice IBE construction. We implement two variants of lattice-based trapdoor sampling and experimentally evaluate their performances.

Keywords: cryptography, identity-based encryption, trapdoors, lattices, quantum

TABLE OF CONTENTS

List of Figures

LIST OF ABBREVIATIONS

PKI	Public Key Infrastructure
PKE	Public Key Encryption
CA	Certificate Authorities
IBE	Identity-Based Encryption
NIST	National Institute of Standards and Technology
MSK	Master Secret Key
MPK	Master Public Key
PPT	Probabilistic Polynomial Time
CPA	Chosen Plaintext Encryption
CCA	Chosen Ciphertext Encryption
PK	Public Key
sk	Secret Key
QR	Quadratic Residuosity
HIBE	Hierarchical Identity-Based Encryption
HNF	Hermite Normal Form
SVP	Shortest Vector Problem
LWE	Learning With Errors
SIS	Short Integer Solutions
PSF	Preimage Sampleable Functions

1 INTRODUCTION

The practices of secure communication are applied since the last 4000 years and the challenging part was the distribution of secret keys. The historical cryptosystems relied on algorithms in which parties used the same key for encryption and decryption. Additonally, key sharing require physical meetings. The paradigm using same keys is also known as **symmetric key encryption** which is still widely used today, such as AES algorithm. The idea of sharing secret key publicly has been thought as a big challenge and first approach was due to Diffie and Hellman [17], where they introduced the famous Diffie-Hellamn key exchange protocol which gained a significant attention. Their seminal work has formed the fundamentals of **asymmetric key encryption**, also known as public key encryption which plays the crucial role in today's cryptography standards and applications in secure communication, transactions, protecting sensitive data, etc. In the asymmetric cryptography settings, the schemes consist of two key pairs: public and private key. The receiver party outputs their public key which is used by other parties to encrypt message, and only the receiver who knows the corresponding private key can decrypt the message.

The public key encryption schemes rely on some ***hardness assumptions***, using mathematical problems that are unsolvable in polynomial time. There have been a significant work on cryptographic tools via hardness assumptions from number theory for traditional public key cryptography. Moreover, the problem of accessing to an internet user's **correct** and **trusted** public key still remains as an issue, where it is not straighfoward to overcome some adversary to adjust the public information so that they could read the encrypted message. This issue raises from the **key authenticity**, which ensures that a public key truly belongs to the claimed owner. Building a public key encryption scheme in larger scale would require having a sort of mechanism to store the user identites (***ID***) and their public keys (***PK***). Obviously, the first part of the tuple is a unique identification information of a user and when some other parties want to send a message, they pick the user's public key from the directory and encrypt the message to the ***PK***, informally. Generally speaking, The set of id and public key pairs is a major part of public-key infrastructure (also known as PKI) and it has to be trusted and authenticated. Furthermore, no adversary should be able to insert another ***ID,PK'*** where he presumably also knows the corresponding secret key ***SK'***. The PKI centers work on a hierarchical trust model, where **Certificate Authorities (CA)** as trusted parties issue digital signatures for public keys, confirming ownership of the public key to the user, and any certificate is trusted by all other entities that trust the CA. While the **PKIs** seems to solve the key substitution problems, they bring centralization, trust, and security risks. We have seen real life examples of succesfull attacks against Certificate Authorities, where a lot of fake certificates have been issued.

1.1 Identity-based encryption

There exist different approaches to address the certificate issue, and Identity-based encryption appears as an applicable alternative where any string can act as a public key of the user. Thus, one of the main objectives of this is to investigate alternatives for PKIs. Identity-based encryption (IBE) is a form of public-key cryptography where a public key is derived directly from a user's identity, such as an e-mail address or unique identifier. This negates the need for traditional Public Key Infrastructure (PKI), where trusted certificate authorities manage public key distribution, however it assumes a trusted key generation center as we shall see later. IBE offers several advantages:

- **Simplified Key Management**: Public keys are derived from identities, eliminating the need for certification processes, reducing cost.
- **Verification**: Enables users to securely message without exchanging public and private keys

The novel approach was originally due to Shamir [36] where he also introduced the identity-based signatures , however, did not solve this problem and it remained as an open question for over 17 years. Boneh and Franklin [13] introduced the first IBE scheme using the hardness of ***Decisional Bilinear Diffie-Hellman*** assumption which was practically doable, followed by Cock's work [16] under the Quadratic Residuosity (QR) assumption in the same year. Moreover, several approaches followed by these fundamental approaches are presented based on bilinear mappings and elliptic-curves. When cryptosystems are designed, it is assumed that the power of adversaries are bounded by polynomial time computation and time on ***classical*** computers.

1.2 Lattice-based cryptography

However, there is a great movement on ***Quantum Computing*** which represents a revolutionary change in computational sources and computation, using a new class of algorithms that is capable of solving some problems faster than any classical computer, and a well-known of such algorithms is **Shor's algorithm** [37]. The algorithm can ***efficiently*** solve integer factorization and discrete-logarithm problems in polynomial time, which are the widely used assumptions of public key cryptography. As advancements made in quantum computing, this will also bring challenges for latter attempts of identity-based encryption schemes. Hence, there is a great research on investigation of quantum-proof cryptosystems in the literature. For instance, The U.S. Department of Commerce's National Institute of Standards and Technology (NIST) carries the responsibility for post-quantum cryptography standardization project where they develop a set of cryptographic protocols for key-encapsulation mechanism, digital signatures, etc. building secure cryptographic quantum-proof tools against quantum attacks. This trend brings the necessity of secure systems in quantum settings and the need for post-quantum cryptography. The most promising candidate for post-quantum cryptography is seem to be lattice-based cryptography, a subfield of cryptography that relies on lattices and the core hard problem of finding a short vector in this lattice. Lattice-based cryptosystems have attractive features of being highly parallelizable and fast due to their relations with matrices, and they have ***worst-case*** security guarantees. In simple terms, solving lattice-based average-hard problems is as hard as solving other related hard problems in the worst case. There are also applications of lattices for achieving quantum-resistant identity-based encryption and it is worth to investigate.

The basic tool in this work is a set of ***trapdoor functions*** that appear useful for our purposes. Trapdoor functions are type of functions that is efficient to compute in forward for eveyone, but hard to invert without the ***trapdoor*** information, which changes for the context of hardness assumptions. The notion of trapdoor information can vary from context to context. In lattice-based cryptography, it is usually accepted a full-rank "good" basis. As we are trying to find a secret key for given identities of users, we will use many-to-one and surjective trapdoor functions, as we want to sample preimages for all of the possible space for user identities. To build the important building blocks of IBE, we have investigated two main types of lattice trapdoors, the traditional ones with ***short full rank bases***, and gadget-based trapdoors, which are much faster, parallelizable , and consumes less memory. The trapdoor inversion algorithms sample one of the preimages under a

specific distribution, the so-called ***discrete Gaussian distribuiton*** that enables preimage sampling over lattices. Basically, we want to sample among many preimages with the same distribution, and not to expose our secret short basis. The advantage of these sampling methods is that they do not reveal the geometric properties of the lattice basis and sampled vector only depends on its euclidian length from origin. The discrete Gaussian distribution is also very useful for ***worst-case to average-case*** reductions, a nice property of lattices. Furthermore, the ***main objectives*** of this thesis include defining the identity-based encryption, its advantages and disadvantages, and the security definitions in different settings. We investigate the key recovation for differet situations, and using IBE for different delegation duties. On the other hand, we studied the foundations lattice-based cryptography, their security guarantees in quantum settings, hardness assumptions, and a twisted public key encryption scheme for enabling IBE. Using lattice-based hardness assumptions, we review two trapdoors constructions for evaluating trapdoor functions. We show preimage sampling using a randomized nearest-plane algorithm with a full-rank basis and use the perturbation method for achieving the desired distribution with gadget-based trapdoors. Finally, we experimentally evaluated their performances via different security parameters and tested their usability for real life IBE schemes.

2 LITERATURE REVIEW

The notion of identity-based encryption was introduced in 1984 [36] by Shamir after the hype of public key encryption, where he could not solve the problem itself using RSA function but described the features of such a cryptosystem. There have been several proposed schemes in the literature that rely on three main mathematical objects such as bilinear mappings, quadratic resudies, and lattices.

Moreover, constructing the initial implementation of an IBE scheme remained an open problem for over ten years. In 2001, Boneh and Franklin [13] proposed the first fully functional IBE scheme using the Bilinear Diffie-Hellman problem. The proposed algorithm is based on a bilinear mapping overa group defined by an elliptic curve, enabling smaller key sizes and forming the foundation for most IBE schemes. It was followed by [11, 10, 38], and [33], relying on various assumptions on this common structure. Shortly after Boneh et al. work, in 2001, Cocks [16] also proposed IBE scheme that uses the Quadratic Residues (QR) problem with modulo a large composite integer and it encrypts messages bit by bit, resulting in ciphertext expansion. There have been other works using the quadratic residues such as [14], [8], still in random oracle model. A more recent study uses the Computational Diffie-Hellman assumption and introducing a completely different approach by Dottling and Sarg [18].

Hierarcial identity-based encryption (HIBE) is known as a more functional type of IBE which enables users to compute the secret key for an extended identity string. The proposed approaches [27, 38, 24, 10, 22, 12] and many more also rely on a bilinear mapping defined on a group for achieving Hierarcial IBE.

All of the mentioned works rely on well pairing principle (e.g. bilinear maps) and quadratic residue problem as their underlying hardness assumptions. However, the issue with those schemes is that they rely on hardness assumptions that are not believed to be secure against quantum attacks, making them potentially vulnerable to future quantum computers. Using lattice-based hardness assumptions is one of the most promising approaches to building quantum-resistant cryptographic

tools. In literature, the proposed schemes mostly rely on hardness of **Learning With Errors** problem. The first IBE from lattices were introduced in 2008 by Gentry, Peikert and Vaikuntanathan, namely the GPV scheme [23] which uses trapdoor inversion for finding secret key for given ID, and will be used our primary approach for building a prototype. It was followed by Awragal and Boyen [3], removing the need for random oracle model. The disadvantage of the latter methods is their key size compared to classical variants and is not practical. Ducas et al. [19] proposed an IBE scheme based on NTRU lattices using a more suitable distribution making GPV-based schemes more practica and achieving smaller key sizes, and followed by a recent work by Ji et al. [28]. These approaches rely on the GPV scheme, which involves inverting with traditional full-rank lattice trapdoors like Ajtai's [5]. Furthermore, Micciancio et al. [30] proposed a different way of achieving lattice trapdoors by introducing a of trapdoors that are smaller in size and have the same quality. Although not being directly related to IBE schemes, it enhances efficiency and reduces complexity by offering smaller key sizes, paralleized algorithms for inversions, and it serves as an additional prototype we aim to develop. On the other hand, there are Hierarchial IBE schemes based on LWE assumption, Bonasi et al. [15] proposed the first HIBE scheme from lattices in the standard model, and other works with random oracles ([1],[2]). On the implementation side, researchers tend to use different ring settings for smaller key size anf faster calculations, such as [29, 9], improving efficiency of trapdoors.

3 PRELIMINARIES

3.1 Notations

We denote the integers as $\mathbb{Z}$ and $\mathbb{R}$ represents the set of the real numbers. We denote the set $\{1, \ldots, m\}$ as $[m]$ for a positive integer m. We use lowercase bold letters $\mathbf{a}$ for vectors and $\mathbf{A}$ for matrices. Given two matrices/vectors, $\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \end{array}\right]$ represents horizontal and $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}$ represents vertical concatenation.

We define ***singular value decomposition*** of real matrix $\mathbf{B} \in \mathbb{R}^{n\times k}$as $\mathbf{B} = \mathbf{QDP}^\top$, in which $\mathbf{Q} \in \mathbb{R}^{n\times n}$, $\mathbf{P} \in \mathbb{R}^{k\times k}$ are square and orthogonal matrices, and $\mathbf{D} \in \mathbb{R}^{n\times k}$ is a uniqely defined diagonal matrix with positive entries $s_i \geq 0$ on the diagonal, in non-increasing order. The s_i are called the singular values of $\mathbf{B}$ and the following holds

$$s_1(\mathbf{B}) = \max_{\|u\|=1} \|\mathbf{B}u\| = \max_{\|u\|=1} \|\mathbf{B}^\top u\| \geq \|\mathbf{B}\|, \|\mathbf{B}^\top\|,$$

where unit vectors represent $u \in \mathbb{R}^k$.

The ***Moore-Penrose*** pseudoinverse of a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is denoted by $\mathbf{A}^+$ that

$$(\mathbf{AA}^+)\mathbf{A} = \mathbf{A}, \quad \mathbf{A}^+(\mathbf{AA}^+) = \mathbf{A}^+,$$

and its mutliplication with original matriA is both symmetric, $\mathbf{AA}^+$ and $\mathbf{A}^+\mathbf{A}$. Both matrices span the same space $\text{span}(\mathbf{A}) = \text{span}(\mathbf{A}^+)$, and If $\mathbf{A}$ is non-singular the pseudoinverse is the same as inverse of the matrix $\mathbf{A}^+ = \mathbf{A}^{-1}$.

Let $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric matrix. We say $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ is ***positive definite*** (or ***semidefinite***) if for every nonzero real vector $x \in \mathbb{R}^n$, $x^\top \mathbf{\Sigma} x > \mathbf{0}$ holds ($x^\top \mathbf{\Sigma} x \geq \mathbf{0}$ for semidefinite) .

We define positive definitene matrices by $\mathbf{\Sigma} > 0$, and $\mathbf{\Sigma} \geq \mathbf{0}$ for positive semidefinite matrices. A matrix $\mathbf{\Sigma}$ satisfies this property if and only if its inverse is also positive definite and the same holds for semidefinite matrice when its pseudoinverse $\mathbf{\Sigma}^{+} \geq \mathbf{0}$.

The comparison positive definiteness matrices follows as if $(\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2) > 0$ then $\mathbf{\Sigma}_1 > \mathbf{\Sigma}_2$ and similarly for positive semidefiniteness. The oppostie holds for their pseudoinverses, if $\mathbf{\Sigma}_2^{+} \geq \mathbf{\Sigma}_1^{+} \geq 0$, then the inequality $\mathbf{\Sigma}_1 \geq \mathbf{\Sigma}_2 \geq 0$ holds in the case of semidefiniteness, and same suffices for strict inequalities.

The multiplication of any matrix $\mathbf{B}$ with its transpose yields positive semidefinite matrix $\mathbf{\Sigma} = \mathbf{B}\mathbf{B}^{\top}$, thus we call $\mathbf{B}$ the square root of $\mathbf{\Sigma} > 0$, written as $\mathbf{B} = \sqrt{\mathbf{\Sigma}}$. We can calculate the square root of any $\mathbf{\Sigma} \geq 0$ has a square root using the ***Cholesky*** decomposition.

Given two matrices, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times q}$, the tensor product of $\mathbf{A}$ and $\mathbf{B}$ is denoted as $\mathbf{A} \otimes \mathbf{B}$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

which is an $mp \times nq$ block matrix.

Definition 3.1. Given two random variables X and Y over a countable set A, we define their statistical distance as follows:

$$\Delta(X, Y) = \frac{1}{2} \sum_{a \in A} |\Pr\{X = a\} - \Pr\{Y = a\}|$$

For a positive parameter s , the Gaussian function is defined on range $\mathbb{R}^n$ parametrized with mean $\mathbf{c}$ as:

$$\forall x \in \mathbb{R}^n, \quad \rho_{s,c}(x) = \exp\left(-\pi \frac{\|x - c\|^2}{s^2}\right)$$

3.2 Lattices

We will first start with basic notations of lattices.

Definition 3.2. Let $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$ be a set of linearly independent vectors. The ***lattice*** $\mathcal{L} \subset \mathbb{R}^m$ is the set of all linear combinations of integer coefficients

$$\mathcal{L} = \{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_m\mathbf{v}_m \mid a_1, a_2, \ldots, a_m \in \mathbb{Z}\}.$$

In this section, we will briefly define lattices. The set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ is a ***basis*** of the lattice $\boldsymbol{L}$ Moreover, the same lattice can be represented with infinite bases. The number of vectors represents the ***dimension*** of the lattice. We can express the basis of the lattice $\mathcal{L} \in \mathbb{R}^n$ with dimension of n with column vectors

$$\mathbf{B} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \\ | & | & & | \end{bmatrix}$$

where $\mathbf{B}$ is the matrix whose columns are the basis vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and the lattice generated by $\mathbf{B}$ is defined as:

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^m\}.$$

The ***dual lattice*** Λ^* is the set of vectors whose elements yield integer inner products:

$$\Lambda^* = \{x \in \mathbb{R}^n : \forall\, v \in \Lambda, \langle x, v\rangle \in \mathbb{Z}\}$$

Micciancio et al. [31] introduced the smoothing parameter:

Definition 3.3. For any n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$ and a positive real $\varepsilon > 0$, the ***smoothing parameter*** $\eta_\varepsilon(\Lambda)$ is defined as the smallest real number $s > 0$ such that

$$\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \varepsilon.$$

where Λ^* denotes the dual lattice.

Definition 3.4. Given a lattice basis $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ of $\mathcal{L} \subseteq \mathbb{R}^n$, its ***Gram-Schmidt*** orthogonalization of $\widetilde{\mathbf{B}}$ is the set of vectors $(\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2, \ldots, \widetilde{\mathbf{b}}_n)$, where $\widetilde{\mathbf{b}}_1 = \mathbf{b}_1$, and for $i = 1, 2, \ldots, n$, calculated iteratively as:

$$\widetilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{b}_i, \widetilde{\mathbf{b}}_j\rangle}{\langle \widetilde{\mathbf{b}}_j, \widetilde{\mathbf{b}}_j\rangle} \widetilde{\mathbf{b}}_j$$

For all of the basis vectors, it's clear that $\|\widetilde{\mathbf{b}}_i\| \leq \|\mathbf{b}_i\|$. Moreover, the orthogonalized vectors do not necessarily lie in the lattice.

3.2.1 Hard Random Lattices

We can define a set of lattice problems introduced by [4] in more natural form using ***parity check matrix*** $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for some values of n, m and q, a notion from coding theory.

We define the lattice associated with $\mathbf{A}$ as as

$$\Lambda^{\perp}(\mathbf{A}) = \left\{ \mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \sum_{j \in [m]} x_j \cdot \mathbf{a}_j = \mathbf{0} \in \mathbb{Z}_q^n \right\} \subseteq \mathbb{Z}^m.$$

The lattice $\Lambda^{\perp}(\mathbf{A})$ is also called a q***-ary lattice***, which means that $q \cdot \mathbb{Z}^m \subseteq \Lambda^{\perp}(\mathbf{A})$ for every $\mathbf{A}$. In other words, all integer vectors that are multiples of q belong to the lattice $\Lambda^{\perp}(\mathbf{A})$ and a vector belongs to the lattice if its entries modulo q satisfy the equality.

3.2.2 Hermite Normal Form

The Hermite Normal Form (HNF) is essentially reduced echolong form of some matrices over integers. We say that a square nonsigualar integer matrix $\mathbf{H} \in \mathbb{Z}^{m \times m}$ is in the hermite normal form if

1. the entries under diagonal i.e., $h_{i,j} = 0$ for all $i > j$, that the matrix H is upper triangular.
2. diagonal entries are positive$h_{i,i} > 0$ for all $i \in [m]$

3. the upper triangular elements $i < j, 0 \leq h_{i,j} < h_{i,i}$, that is, each element is reduced modulo the diagonal entry in its row.

$$H = \begin{bmatrix} h_{1,1} & \cdots & \cdots & \cdots & \cdots & \cdots \\ & \ddots & & & & \\ & & h_{i,i} & \cdots & h_{i,i} & \cdots \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & h_{n,n} \end{bmatrix}$$

Every nonsingular square integer matrix can be converted to a matrix in HNF via elementary column operations. Given parity-check matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we can compute the hermite normal form of $\mathbf{H} \in \mathbb{Z}^{m \times m}$ of its basis efficiently.

4 IDENTITY-BASED ENCRYPTION

In this section, we will define identity-based encryption, its foundations, advantages, and security definitions. As mentioned earlier, despite the fact that the public-key infrastructure is widely used, a directory is needed to store the identity and public key pairs. Identity-based encryption addresses this issue by eliminating the directory and replacing the public keys with identifying strings of users, such as emails, phone numbers, street line addresses, social security numbers, etc. Without exchanging keys, it enables two parties to communicate and verify each other's signatures. The scheme assumes a trusted key generator center that is responsible for producing **master secret key (MSK)** and **master public key (MPK**, and issues secret keys to the users.

In traditional public key cryptosystems, a third party randomly generate public and private keys for users. The major difference here is the user chooses his own public key which uniquely identifies him, and the key center calculates the corresponding secret key for the user using the MSK, and it will be used for decryption.

4.1 Applications of IBE

Aside from its advantage over PKIs, here are some additional properties of IBES.

4.1.1 Key Revocation

Public key infrastructure systems often issue certificates for keys with a defined expiration date. In IBE settings, this process is much simpler: Bob will have a different public key for different time periods, for instance, "bob@example.com$\|$current $-$ year" where the current year changes over the time and Bob needs to get a new secret key every year. We have the following observations:

- Bob can read the encrypted messages with only corresponding secret keys issued by the private key generator.
- Alice doesn't have to receive any certificate for new public keys.

Moreover, this process would become much complex if there was a need for daily change on Bob's public key and issue new certificates in PKI settings. The interesting feature of IBE is that there is no need for any communication with either Bob or a third party to know his daily public key. Hence, IBE is a promising solution for short-term keys. One more interesting feautre is that sending messages into the future is possible, whereas Bob can only decrypt the ciphertext when the specified time arrives which is responsibilty of the key generator system.

4.1.2 Delegation of keys

Another feature of IBE systems is that it allows for controlling the decryption for different duties. In the following cases, Bob will act as the trusted key generator, where he will know the master secret key to extract secret keys and outputs his public parameters.

- Suppose Bob goes to a travel for a week and Alice wants to send him messages on day-based public keys. Bob suspects that his laptop may get stolen during the trip. Since Bob knows the master secret key, he will generate the corresponding secret keys for every day of the trip and install them on his laptop. If the laptop gets stolen, only the secret keys for this week will be compromised and the master secret key will be unharmed, and the adversary could only read the messages that was sent to these specific dates. In case of PKI, the adversary would be able to read all the messages as he has the private key.
- Suppose that Bob wants his assistant to only be able to decrypt a subset of messages with a tag ***ID=1234***. He will simply issue the corresponding secret key to his asssistant so that he could read the messages for ***ID***

4.2 Definition of IBE

In this section, we will define the syntax of identity-based encryption schemes. An Identity-based encryption scheme IBE consists of four components: (Setup, Extract, Enc, Dec)

- $\mathsf{Setup}(1^\lambda)$: based on the security parameter λ, return master public key (MPK), with other public parameters params which are used for encryption and master secret key (MPK) for generating secret keys as it remains private.
- $\mathsf{Extract}(\mathsf{params}, \mathsf{MSK}, ID)$: the private key generator function which uses master secret key and outputs corresponding secret key sk_{ID} for ID
- $\mathsf{Enc}(\mathsf{params}, \mathsf{MPK}, ID, m)$: encrypts the message m and returns ciphertext c with respect to the ID
- $\mathsf{Dec}(\mathsf{params}, sk_{ID}, c)$: the decryption algorithm is a deterministic function that recovers the message m

The decryption functions must satisfy the **correctness** property in most schemes, meaning that if $c = \mathsf{Enc}(\mathsf{params}, \mathsf{MPK}, ID, m)$ is the encryption of message m, then

$$\mathsf{Dec}(\mathsf{params}, sk_{ID}, c) = m$$

should satisfy. However, the same doesn't hold for the lattice-based schemes, as there is always some small negiligible probability that it may not hold. A careful reader may realize that the master authority is able to read all messages for any identity, as it issues secret keys, so it is very powerful. The figure one represents a visualization of IBE scheme.

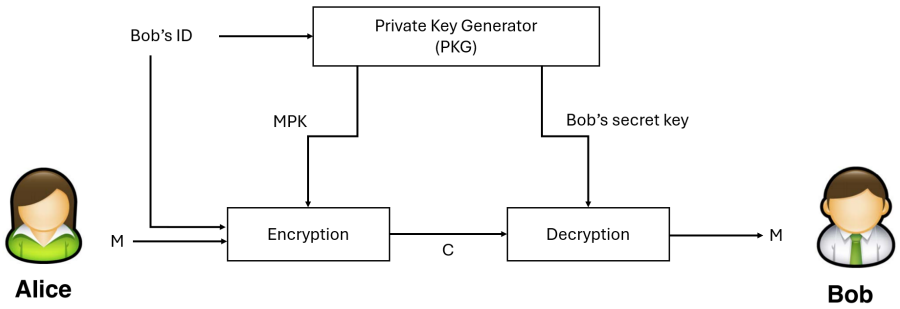


Figure 1: IBE Scheme

Another important property of IBE schemes that we will define is ***anonymity***, where we require the identity used for encryption to be remain unrecognizable for anyone except the intended recipient. The main objective of this property is to make ciphertexts sent to different identities **indistinguishable** for adversaries , enhancing the privacy and security of communications.

4.3 Security definitions and games for Identity-based encryption scehemes

The purpose of cryptographic schemes is to make messages secure or inrecognizable in the presence of adversaries. We assume that the adversary can have polynomially many computational power and time, and still is unable to break the security of our game. In the case of public key encryption, the adversary receives public parameters, and chooses two messages for the game. The challenger chooses one of them uniformly random, encrypts the message and gives the ciphertext back. If the adversary can guess correctly guess which message was encrypted with a probability more than 0.5, he wins the game. However, the identity-based encryption differs from public key encryption and it has its own security game. Suppose we have a Probabilistic Polynomial Time (PPT) adversary who is trying to break our scheme. As we try to enable secure communication for some ID, the adversary should have access to secret keys for other $ID_1, ID_2, ..., ID_n$. We want the encrypted message to ID should still be secure even the adversary has access to the secret keys for polynomially many $ID' \neq ID$. This security notion is also known as ***full*** or ***adaptive security***. A weaker security notion in the context of IBE is called selective security, in which the adversary has to pick the ID before the game starts. We will define the Chosen Plaintext Encryption (CPA) and Chosen Ciphertext Encryption (CCA) security for identity-based encryption as [13].

4.3.1 IND-ID-CPA ***security for IBE***

Consider the following experiment:

> **Setup**. The challenger picks a security parameter λ, and runds the Setup to obtain (MPK, MSK, params), and gives (MPK, params) while keeping MSK to himself as secret.

Phase 1. The PPT adversary $\mathcal{A}$ asks for polynomially many queries for extraction of secret keys for $Q = \{ID_1, ID_2, ..., ID_m\}$, and receives corresponding $sk_{ID_1}, sk_{ID_2}, .., sk_{ID_m}$. The queries can be made adaptively as well, based on the results of previous queries.

Challenge. After the **Phase 1** ends, the adversary generates an equal length of two messages from the message space $m_0, m_1 \in \mathcal{M}$, an $ID^* \notin Q$ of its choice, and send them to the challenger. The challenger extracts the secret key sk_{ID^*}, randomly chooses a bit $b \in \{0, 1\}$, and encrypts $c^* = \mathsf{Enc}(\mathsf{params}, \mathsf{MPK}, ID^*, m_b)$. The challenger sends c^* as the challenge ciphertext to the adversary.

Phase 2. As in **Phase 1**, the adversary continues to ask for extraction queries $\{ID_{m+1}, ID_{m+2}$, $..., ID_n\}$ while $ID_i \neq ID^*$, and receives
$\{sk_{ID_{m+1}}, sk_{ID_{m+2}}, .., sk_{ID_n}\}$.

Guess. The adversary outputs a bit b', we say that he wins if $b = b'$ Consider the advantage function on scheme $\mathcal{E}$, adversary $\mathcal{A}$, and the security parameter λ:

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}}(\lambda) = |\Pr(b = b') - \frac{1}{2}|$$

Definition 4.1. We say that the IBE scheme $\mathcal{E}$ is IND-ID-CPA secure if for all the PPT adversaries $\mathcal{A}$ the advantage function $\mathrm{Adv}_{\mathcal{E},\mathcal{A}}(\lambda)$ is negligible

To achieve full security, we can adjust the game as the adversary outputs the challenge identity ID^* before the game start.

4.3.2 IND-ID-CCA *security for IBE*

Here we define a stronger notion of $CCAsecurity$, where we allow adversary to ask for decryption of any ciphertext under any identity except the challenge identity. Consider the following experiment:

Setup. The challenger picks a security parameter λ, and runs the Setup to obtain (MPK, MSK, params), and gives (MPK, params) while keeping MSK to himself as secret.

Phase 1. Here the challenger needs to provide for two queries:

1. **Key extraction**: The PPT adversary $\mathcal{A}$ asks for polynomially many queries for extraction of secret keys for $\{ID_1, ID_2, \ldots, ID_m\}$. The challenger runs Extract(params, MSK, $\boldsymbol{ID}_i$) and outputs secret keys $sk_{ID_1}, sk_{ID_2}, \ldots, sk_{ID_m}$.
2. **Decryption**: Additionally, the adversary may query for decryption of any ciphertext c under any identity ID. The challenger extracts the secret key sk_{ID}, runs decryption $m = \mathsf{Dec}(\mathsf{params}, sk_{ID}, c)$, and returns m.

The queries can be made adaptively as well, based on the results of previous queries.

Challenge. After **Phase 1** ends, the adversary generates two messages of equal length from the message space $m_0, m_1 \in \mathcal{M}$, and an identity ID^* which wasn't previously queried in Phase 1, and sends them to the challenger. The challenger randomly chooses a bit $b \in \{0, 1\}$, encrypts $c^* = \mathsf{Enc}(\mathsf{params}, \mathsf{MPK}, ID^*, m_b)$, and sends c^* as the challenge ciphertext to the adversary.

Phase 2. As in **Phase 1**:

1. **Key extraction**: The adversary continues to ask for extraction queries $\{ID_{m+1}, ID_{m+2}, \ldots, ID_n\}$ while $ID_i \neq ID^*$, and receives $\{sk_{ID_{m+1}}, sk_{ID_{m+2}}, \ldots, sk_{ID_n}\}$.
2. **Decryption**: The adversary may also continue making decryption queries on any ciphertext $c \neq c^*$ under any identity $ID \neq ID^*$.

Guess. The adversary outputs a bit b', if $b = b'$ he wins the game.

Consider the advantage function on scheme $\mathcal{E}$, adversary $\mathcal{A}$, and the security parameter λ:

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}}(\lambda) = \left|\Pr(b = b') - \frac{1}{2}\right|$$

Definition 4.2. We say that the IBE scheme $\mathcal{E}$ is IND-ID-CCA secure if for all PPT adversaries $\mathcal{A}$, the advantage function $\mathrm{Adv}_{\mathcal{E},\mathcal{A}}(\lambda)$ is negligible.

To achieve full security, we can adjust the game such that the adversary outputs the challenge identity ID^* before the game starts.

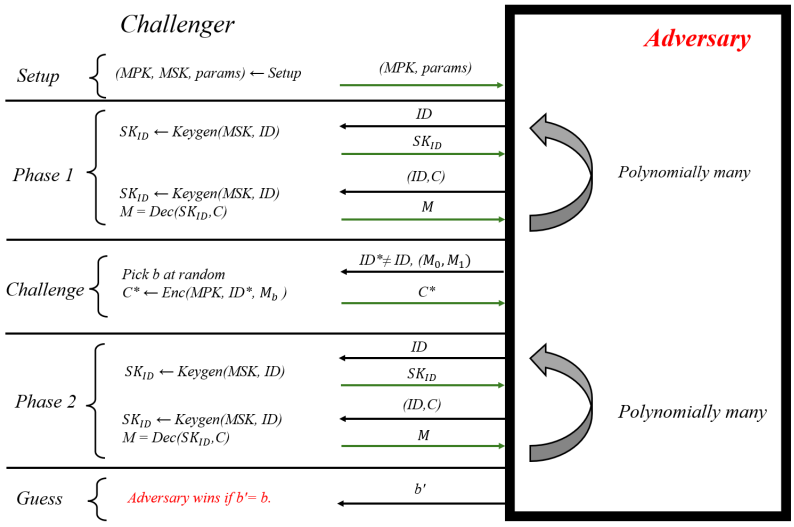


Figure 2: IND-ID-CCA security game for IBE

4.4 Hierarchial IBE

Hierarchical Identity-Based Encryption (HIBE) is a more functional version of IBE which provides a way for delegating secret keys for lower level of users in a hierarchical structure. As standard IBE, the Hierarchical IBEs also consist of four functions, : (Setup, Extract, Enc, Dec). In HIBE systems, identities are vectors of dimension k, at depth k and master secret key is assigned to depth 0. To generate a secret key for depth k, he Extract function takes ID $= (I_1, I_2, \ldots, I_k)$ at depth k and the secret key of parent identity
$sk_{\text{ID}|k-1}$ of the parent identity $\text{ID}_{|k-1} = (I_1, I_2, \ldots, I_{k-1})$ at depth $k-1$., and returns sk_k. Note that by default, the standard identity-based encryption systems are Hierarchical IBE where all identities have the same level 1. Now we will define the selective security game for HIBEs [10]:

Init. The adversary outputs the ID that he wants to be challenged on.

Setup. The challenger chooses the λ, and runs the Setup to obtain (MPK, MSK, params), and gives (MPK, params) while keeping MSK to himself as secret.

Phase 1. Here the challenger needs to provide for $q_1, q_2, \ldots, q_m$ queries, and q_i is for either decryption or key extraction query:

1. **Key extraction**: The PPT adversary $\mathcal{A}$ asks queries for extraction of secret keys for $\{ID_1, ID_2, \ldots, ID_m\}$, where $ID_i \neq ID^*$ or any prefix of ID^*. The challenger runs Extract(params, MSK, $\boldsymbol{ID}_i$) and returns the corresponding $sk_{ID_1}, sk_{ID_2}, \ldots, sk_{ID_m}$.
2. **Decryption**: Additionally, the adversary may query for decryption of any ciphertext c under any identity ID, where $ID \neq ID^*$ or any prefix of ID^*. The challenger extracts sk_{ID}, decrypts $m = \text{Dec}(\text{params}, sk_{ID}, c)$, and returns m.

The queries can be made adaptively as well, based on the results of previous queries.

Challenge. After **Phase 1** ends, the adversary generates two messages of equal length from the message space $m_0, m_1 \in \mathcal{M}$, and an identity ID^* which wasn't previously queried in Phase 1, and sends them to the challenger. The challenger randomly chooses a bit $b \in \{0, 1\}$, encrypts $c^* = \text{Enc}(\text{params}, \text{MPK}, ID^*, m_b)$, and sends c^* as the challenge ciphertext to the adversary.

Phase 2. As in **Phase 1**, the challenger responds to $q_{m+1}, q_{m+2}, \ldots, q_n$, and q_i is for either decryption or key query:

1. **Key extraction**: The adversary continues to ask for extraction queries $\{ID_{m+1}, ID_{m+2}, \ldots, ID_n\}$ where $ID_i \neq ID^*$ or any prefix of ID^*, and receives $\{sk_{ID_{m+1}}, sk_{ID_{m+2}}, \ldots, sk_{ID_n}\}$.
2. **Decryption**: The adversary may also continue making decryption queries on any ciphertext $c \neq c^*$ under identity $ID \neq ID^*$ and any prefix of ID^*.

Guess. The adversary computes a bit b' and wins if $b = b'$.

Consider the advantage function on scheme $\mathcal{E}$, adversary $\mathcal{A}$, and the security parameter λ:

$$\text{Adv}_{\mathcal{E},\mathcal{A}}(\lambda) = \left| \Pr(b = b') - \frac{1}{2} \right|$$

Definition 4.3 ([10]). A HIBE system can be defined in a selective identity and adaptive CCA setting, where a t-time adversary making up to q_{ID} private key and q_C decryption queries has an advantage less than ε.

5 DISCRETE GAUSSIAN SAMPLING

The Gaussian sampling is an efficient algorithm that builds the foundation for cryptographic tools upon lattices that we will use in our work. Given an appropriate basis, the algorithm samples vectors from the so-called discrete Gaussian distribution. The distribution serves as an analytical tool for studying properties of lattices, worst-case to average-case reductions, and also is useful for richer constructions, such as trapdoor functions, digital signatures, and identity-based encryption. In this section, we will define its basic properties, how to efficiently sample over integers and vectors over given appropriate bases, based on discrete Gaussian distribution.

Given an arbitrary basis **B** of the lattice Λ, the mean $c \in \mathbb{R}^n$ and $s > 0$, the algorithm will return a lattice vector distributed based on discrete Gaussian $D_{\Lambda,s,c}$, as long as the parameter s exceeds the lengths of longest ***Gram-Schmidt*** vectors. The crucial part of this sampling algorithm is that sampled vector is determined by the quality(e.g. how short and orthogonal) of the basis and only depends on the maximum length of its gram-schmidt vectors, meaning that it is oblivious to its geometry. From another perspective, the sampling algorithm can be viewed as a randomized ***decoder*** that chooses a lattice vector close to c. It is essentially a probabilistic variant of Babai's "nearest-plane" algorithm [7], in which the algorithm selects a plane with a probability based on its distance from the target point.

We need a subroutione to sample an integer over a one dimensional lattices on a discrete Gaussian, the first attempt could be sampling from ***continues*** Gaussian N_s with input s, and rounding it to the closest integer. However, this method does not produce a discrete Gaussian distribution and is not even statistically close to it.

Lemma 5.1. ***The statistical distance among the discrete Gaussian distribution $D_{\mathbb{Z},s}$ and the roundied samples from the continuous Gaussian N_s is at least $1/s^3$, which is not negligible.***

Here we will show how to "directly" sample from a lattice under the target distribution. Achieving statistical clossenes is hard even sampling from one dimensional case, as the closest approximation may not have a good representation of $D_{\mathbb{Z},s}$. Now, we will define an algorithm that correctly samples from the integer lattice $\mathbb{Z}$ due to [23].

Theorem 5.1 ([23]). ***There is a probabilistic polynomial-time algorithm that, given a basis B of an n-dimensional lattice $\Lambda = L(B)$, a parameter $s \geq |\widetilde{B}| \cdot \omega(\sqrt{\log n})$, and a center $c \in \mathbb{R}^n$, produces a sample from a distribution that is indistinguishable from $D_{\Lambda,s,c}$ up to a negligible statistical distance.***

5.1 Sampling Integers

We define a subroutine SampleZ, which samples from the discrete Gaussian $D_{\mathbb{Z},s,c}$ over $\mathbb{Z}$ and will be used later for n-dimensional vectors. Let $t(n) \geq \omega(\sqrt{\log n})$ be a function of n, e.g. $t(n) = \log n$. The $\mathbb{Z}$ subroutine relies on rejection sampling :

- Given (s, c) and the security parameter n, choose x from the range $Z = \mathbb{Z} \cap [c - s \cdot t(n), c + s \cdot t(n)]$ uniformly random.
- With probability $\rho_s(x - c) \in (0, 1]$, return x, othervise repeat till it terminates

SampleZ indeed correctly samples from $D_{\mathbb{Z},s,c}$, because the probability of rejecting a sample decreases exponentially as t increases. Additionally, the most of probability mass of the distribution $D_{\mathbb{Z},s,c}$ is captured within Z. The following lemma is about correctness of SampleZ is due to the tail inequality on the distribution $D_{\mathbb{Z},s,c}$.

Lemma 5.2 ([23]). ***For any*** $\varepsilon > 0$***, any*** $s \geq \eta_\varepsilon(\mathbb{Z})$***, and any*** $t > 0$***, we have***

$$\Pr_{x \sim D_{\mathbb{Z},s,c}}[|x - c| \geq t \cdot s] \leq 2e^{-\pi t^2} \cdot \frac{1+\varepsilon}{1-\varepsilon}.$$

In particular, for $\varepsilon \in \left(0, \frac{1}{2}\right)$ ***and*** $t \geq \omega(\sqrt{\log n})$***, the probability that*** $|x-c| \geq t \cdot s$ ***is negligible.***

We want our sampler to be efficient, thus it is important to prove that the subroutine SampleZ that samples from the discrete Gaussian distribution $D_{\mathbb{Z},s,c}$ will initially terminate after a reasonable number of iterations. We refer to the following lemma due to [23] about SampleZ that it halts with overwhelming probability:

Lemma 5.3. ***For any*** $0 < \varepsilon < \exp(-\pi)$***, any*** $s \geq \eta_\varepsilon(\mathbb{Z})$ ***and*** $c \in \mathbb{R}$***, and any*** $\omega(\log n)$ ***function,*** SampleZ ***terminates within*** $t(n) \cdot \omega(\log n)$ ***iterations with a overwhelming probability. Furthermore, its output distribution is statistically close to*** $D_{\mathbb{Z},s,c}$.

Moreover, the Gaussian parameter s does not affect the number of iterations till the termination.

Proof. Define a probability distribution D on Z that $D(x)$ is $\rho_s(x - c)$ when $x \in Z$, and $D(x) = 0$ otherwise. Moreover, the distributions D and $D_{\mathbb{Z},s,c}$ are statistically close. Clearly, so the output distribution of SampleZ is identical to D, so its statistically close to $D_{\mathbb{Z},s,c}$. □

5.2 Sampling from Arbitrary Lattices

Now we will define an algorithm for sampling over arbitrary lattices statistically close to $D_{\Lambda,s,c}$. The algorithm can be summarized as a randomized version of nearest-plane algorithm, where instead of rounding to the closes plane, it samples an integer over adaptive discrete Gaussian over integers $D_{\mathbb{Z},s,c}$ at random. The SampleDGS works on given a basis $\mathbf{B}$ of a lattice, the mean vector c and a parameter s, and computes as follows:

Algorithm 1 Gaussian Sampling Algorithm with `SampleDGS`

procedure SampleDGS($\mathbf{B}$, c, s)
  Set $v_n \leftarrow 0$ and $c_n \leftarrow c$
  for $i \leftarrow n$ **to** 1 **do**
    Compute $c'_i = \langle \mathbf{c}_i, \widetilde{\mathbf{b}_\mathbf{i}} \rangle / \langle \widetilde{\mathbf{b}_\mathbf{i}}, \widetilde{\mathbf{b}_\mathbf{i}} \rangle \in \mathbb{R}$ and $s'_i = s / \|\widetilde{b_i}\| > 0$
    Sample $z_i \sim D_{\mathbb{Z},s',c'}$
    $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i - z_i \mathbf{b}_i$
    $\mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$
  end for
  Return $\mathbf{v}_0$
end procedure

The algorithm assumes an oracle for 5th step, which exactly samples from $D_{\mathbb{Z},s',c'}$ for any c' and $s' > 0$, if s' is large enough, it can be implemented by SampleZ. As each scalar operation takes constant time, the running time of the latter algorithm is $\mathcal{O}(n^2)$, plus the time required for n calls to the oracle and the result of the algorithm is always a lattice vector. The figure 3 shows an example for the following parameters, let the basis be:

$$\mathbf{B} = \begin{bmatrix} 8 & 2 \\ 3 & 4 \end{bmatrix}$$

and the Gram-Schmidt basis is

$$\widetilde{\mathbf{B}} = \begin{bmatrix} 8.0 & -1.06849315 \\ 3.0 & 2.84931507 \end{bmatrix}$$

with $s \approx 10.544$, and $t = 1$.

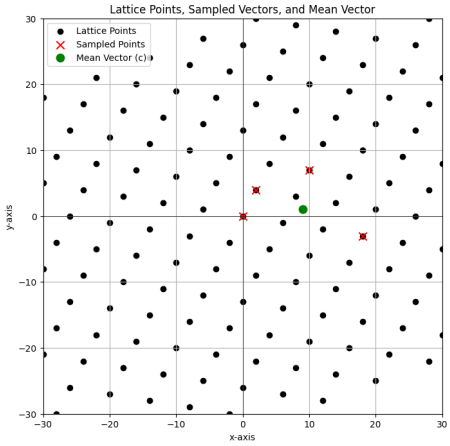


Figure 3: Example of Discrete Gaussian Sampling

6 LATTICES

In this section, we will mention the basic notions of lattices, hardness assumption that will be used through our work. We refer reader to the preliminary section for the basic knowledge of lattices. At the core of the lattice problems and cryptosystems, we seek for the knowledge of short vectors in the lattice. The ability to generate a basis with short vector is important for decryption and many applications, while it should remain hard with only knowledge of the public basis.
The ***minimum distance*** of the lattice $\mathcal{L}$ defined by is the smallest distance among two different lattice points and denoted by $\lambda_1(\mathcal{L})$:

$$\lambda_1(\mathcal{L}) = \min\{\text{dist}(x, y) : x \neq y \in \mathcal{L}\} = \min\{\|x\| : x \in \mathcal{L} \setminus \{0\}\}$$

We can generalize it as follows:

Definition 6.1. Let $\mathcal{L} \subseteq \mathbb{R}^n$ be a lattice of rank n. The $\lambda_i(\mathcal{L})$ is the **i-th successive minimum** and equals the smallest radius r such that the lattice $\mathcal{L}$ contains at least i linearly independent vectors.

$$\lambda_i(\mathcal{L}) = \inf\{r > 0 \mid \dim(\text{span}(\mathcal{L} \cap B_r(0))) \geq i\}$$

where $B_r(0)$ denotes the closed ball of radius r centered at the origin.

Generally speaking, the geometry of the lattices plays an important role on its security, the public basis is a "bad" basis with consist of long vectors which are highly parallel to each other, which makes solving ha problems such as **Shortest vector problem (SVP)**, or finding the closest lattice point to a target vectors, e.g. **Closest Vector Problem** hard. On the other hand, the secret basis is a "good" basis consisting of relatively short and orthogonal vectors. The knowledge of a short basis is the helpful to solve lattice problems and the following tool is useful for analyzing the lattice and gives us nice approximations.

6.1 Lattice Problems and Hardness Assumptions

As mentioned earlier, many cryptosystems rely on hard problems whose it is easy to verify when given a solution, but hard to find one. The problems related to lattices are also known as geometrical problems, and they are highly related to each other. We will use de-facto lattice problems such as the **Learning With Errors (LWE)** and **the Short Integers Solution** problem that are used for building cryptographic applications. We first start by introducing some core lattice-based worst-case approximation problems such as SIVP and GapSVP_γ that are used for proving security of LWE and SIS. These problems are "relaxed" in some sense, due to some approximation factor with function $\gamma = \gamma(n)$ in dimension.

Definition 6.2 (Shortest Independent Vectors Problem). Let **B** a full-rank basis of an n-dimensional lattice, the SIVP_γ problem asks to return a set of n linearly independent lattice vectors $S \subset \mathcal{L}(\mathbf{B})$ such that $\|S\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.

Definition 6.3 (Shortest Vector Problem (Decision Version)). Given a full rank n dimensional lattice with basis **B**, GapSVP_γ asks to return YES instance if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n)$.

6.1.1 *Short Integer Solution (*SIS*)*

Ajtai in his seminal work [4], introduced the The Short Integer Solution problem and the concept of **worst-case to average case reductions**, one of the works using lattices for building cryptographic tools rather using for cryptnalasys. He also showed how to generate hard instances of lattice problems efficiently and used this tool as a one-way function for building a simple collision-resistant hash function, it is also referred as ***minicrypt*** There are other applications of this problem such as identification schemes, digital signatures, etc. serving as a tool of ***minicrypt***. In IBE schemes, it is the building block for finding the secret key for given ID.

Definition 6.4 (Inhomogeneous Short Integer Solutions). Given a uniformly generated matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, q, and a syndrome $\mathbf{u}$, $\mathsf{ISIS}_{n,m,q,\beta}$ (in ℓ_2 norm) asks to find a nonzero integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\|\mathbf{e}\| \leq \beta < q$ and

$$f_{\mathbf{A}}(\mathbf{e}) := \mathbf{Ae} = \mathbf{u} \pmod{q} \in \mathbb{Z}_q^n$$

The right-side of the equation $\mathbf{u}$ is also called ***syndrome***. In its ***homogeneous*** version, the right side of the equation is a zero vector. It is proven by Ajtai that solving homogenius variant of SIS is as hard as solving some other lattice problems in their ***worst-case***.

Theorem 6.1 ([4]). ***For any*** $m = \mathbf{poly}(n)$***, any*** $\beta > 0$***, and for sufficiently large*** $q \geq \beta \cdot \mathbf{poly}(n)$***, solving any instance of*** $\mathsf{SIS}_{n,m,q,\beta}$ ***is at least as hard as solving the decisional approximate shortest vector problem*** GapSVP_γ ***and the approximate shortest independent vectors problem*** SIVP_γ ***on arbitrary*** n***-dimensional lattices, where*** $\gamma = \beta \cdot \mathbf{poly}(n)$.

In other words, the latter theorem confirms that any instance of SIS problem is hard. We want to recall reader that in IBE schemes the identity is chosen by user. Thus, given a random syndrome (identity), we need to have a solution $\mathbf{e}$ for that. Given a random syndrome, ISIS problem admits a solution with overwhelming probability, and it will also be useful when inverting the identity.

Lemma 6.1 ([35]). ***For all but at most a*** q^{-n} ***fraction of matrices*** $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$***, where*** $m \geq 2n\log q$***, the subset sums of the columns of*** $\mathbf{A}$ ***span*** $\mathbb{Z}_q^n$***. That is, for every*** $\mathbf{u} \in \mathbb{Z}_q^n$***, there exists*** $\mathbf{e} \in \{0,1\}^m$ ***such that*** $\mathbf{Ae} = \mathbf{u} \mod q$.

The $\mathsf{ISIS}_{n,m,q,\beta}$ problem remains hard and provides a solution with appropriate parameters, $\beta \geq \sqrt{m}$ and $m \geq 2n\log q$ (for prime q). If the error term chosen over a discrete gaussian distribution on $\mathbb{Z}^m$, the following lemma says that the distribution of syndromes is statistically close to uniform.

Lemma 6.2 ([35]). ***Let*** $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ ***be a matrix whose columns generate*** $\mathbb{Z}_q^n$***, and*** $\varepsilon \in (0, \frac{1}{2})$ ***and any*** $s \geq \eta_\varepsilon(\Lambda^\perp(\mathbf{A}))$***. If*** $\mathbf{e} \sim D_{\mathbb{Z}^m,s}$***, the syndrome*** $\mathbf{u} = \mathbf{Ae} \mod q$ ***has a distribution differs with statistical distance at most*** 2ε ***of the uniform distribution over*** $\mathbb{Z}_q^n$.
For any $\mathbf{u} \in \mathbb{Z}_q^n$***, and let*** $\mathbf{t} \in \mathbb{Z}^m$ ***be arbitary solution to*** $\mathbf{At} = \mathbf{u} \mod q$***. Then, conditioned on*** $\mathbf{Ae} = \mathbf{u} \mod q$***, the distribution of*** $\mathbf{e} \sim D_{\mathbb{Z}^m,s}$ ***is exactly*** $\mathbf{t} + D_{\Lambda^\perp,s,-\mathbf{t}}$.

The second part of the lemma will be used when inverting the ISIS problem.

6.1.2 *Learning With Errors (LWE)*

A generalization of learning with parity problem, also known as ***Learning With Errors*** problem was introduced by Regev [35], one of the building blocks of lattice-based tools, such as CCA-secure

encryption, fully holomorphic encryption (FHE). It serves as a "***cryptomania***", meaining that it used for more sophisticated cryptographic tools. There are two versions of of LWE ***Learning With Errors*** the The LWE problem is another fundamental assumption in lattice-based cryptography. In a nutshell, the problem looks for the secret vector in a set of noisy equations in its search version and it generalizes the well known ***Learning with parity noise*** problem. Now we will define the LWE distribution:

Definition 6.5. [LWE Distribution] Let $s \in \mathbb{Z}_q^n$ be a secret vector. The ***LWE distribution*** $A_{s,\chi} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is defined as follows:

- Sample a uniformly random vector $\mathbf{a} \in \mathbb{Z}_q^n$
- Sample an error term e from some error distribution χon$\mathbb{Z}_q$, and return

$$b = \langle \mathbf{s}, \mathbf{a} \rangle + e \pmod{q}.$$

Definition 6.6 (Search-LWE$_{n,q,\chi,m}$). Given arbitrary number of instances $\{(a_i, b_i)\}_{i=1}^m \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ drawn from $\mathcal{A}_{\mathbf{s},\chi}$, with uniformly random $\mathbf{s}$, the problem asks to find $\mathbf{s}$.

We can also write it in matrix notation, given the equation:

$$\mathbf{As} + \mathbf{e} = \mathbf{b} \pmod{q}$$

where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, secret vector $\mathbf{s} \in \mathbb{Z}_q^m$, and $\mathbf{e} \in \mathbb{Z}_q^n$ is the error vector from χ^n, and $\mathbf{b} \in \mathbb{Z}_q^n$ is perturbed result of the multiplication.

Definition 6.7 (Decision version). Let $q = q(n)$ be an integer polynomial in n and χ be a distribution over $\mathbb{Z}_q$. The Search Learning With Error problem LWE$_{q,\chi}$ asks to distinguish between m independent samples from ***LWE distribution*** $\mathcal{A}_{\mathbf{s},\chi}$ where $\mathbf{s}$ is chosen uniformly at random from $\mathbb{Z}_q^n$, and the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, with non-negligible probability. In short, $\mathcal{A}_{\mathbf{s},\chi}$ is pseudorandom if the LWE problem is hard,

$$(\mathbf{A}, \mathbf{u}) \approx (\mathbf{A}, \mathbf{b})$$

is indistinguishable, where $\mathbf{u} \in \mathbb{Z}_q^n$.

In its core, LWE is a bounded-decoding problem on the dual lattice Λ^*, a variant of closest vector problem that asks finding a nearby lattice point bounded by a threshold.

6.2 Dual-Regev scheme

The dual scheme is essentially a variant of Regev's cryptosystem where every point in $\mathbb{Z}_q^n$ can serve as a public key. The secret key is a vector $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ and public key is its syndrome $\mathbf{Ae} = \mathbf{u} \pmod{q}$. The ciphertext consists of two parts: the encryption algorithm chooses a uniform secret $\mathbf{s} \in \mathbb{Z}_q^n$, error vector $\mathbf{x} \leftarrow \chi^m$ and $x \leftarrow \chi$, and calculates an pseudorandom LWE vector $\mathbf{p} = \mathbf{A}^T\mathbf{s} + \mathbf{x}$ as shown in the definition 6.5. It generates another LWE instance $p = \mathbf{u}^T\mathbf{s} + x$ using public key (syndrome) as a "pad" to hide the message . Based on the lemma 6.2 public key syndrome $\mathbf{u}$ is nearly uniform, the adversary's view in the dual system is also indistinguishable from uniform, under the

hardness of LWE, as we call the scheme ***anonymous***; the ciphertext hides the public key to which it was encrypted.

Compared to the original scheme of Regev, the set of public keys are dense. Moreover, every syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ is actually a valid public key and each may have many corresponding decryption keys $\mathbf{e} \in \mathbb{Z}_q^m$, as the error is large enough. Furthermore, a trapdoor for $\mathbf{A}$ allows a trusted authority to efficiently sample a secret key $\mathbf{e}$ corresponding to any syndrome $\mathbf{u}$, under the same distribution as in the dual cryptosystem. The scheme is essentially useful for identity-based encryption schemes, as the hash function maps identities to random points in $\mathbb{Z}_q^n$.

6.2.1 Definition of the scheme

The secret keys are sampled from the discrete Gaussian $D_{\mathbb{Z}^m,r}$, in which the parameter $r \geq \omega(\sqrt{\log m})$. The matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ chosen uniformly at random defines the lattice and is public, also declares the very same function $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{Ae} \bmod q$. We define the scheme as [23], and all operations are calculated module q.

- DualKeyGen : Choose an error vector $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ (i.e., the input distribution to $f_{\mathbf{A}}$), which is the secret key. The public key is the syndrome $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e})$.
- DualEnc$(\mathbf{u}, b)$: given a bit message $m \in \{0,1\}$, sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \chi^m$, and $x \leftarrow \chi$. Compute
$$\mathbf{c_1} := \mathbf{A}^T\mathbf{s} + \mathbf{x} \in \mathbb{Z}_q^m, \quad c_2 := \mathbf{u}^T\mathbf{s} + x + \left\lfloor \frac{q}{2} \right\rfloor m \mod q.$$
The ciphertext is $c = (\mathbf{c_1}, c_2) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$.
- DualDec$(\mathbf{e}, c)$: Given ciphertext $c = (\mathbf{c_1}, c_2)$ and secret key $\mathbf{e}$, compute $c_2 - \mathbf{c_1 e} \mod q$. If
$$|c_2 - \mathbf{c_1 e} \mod q| < \frac{q}{4},$$
output 0; otherwise, output 1.

With appropriate parameters, the above scheme is ***CPA-secure***:

Theorem 6.2 ([23]). ***Let*** $q \geq 5r(m+1)$, $\alpha \leq 1/r\sqrt{m+1} \cdot \omega(\sqrt{\log n})$, ***let*** $\chi = \overline{\Psi}_\alpha$, ***and*** $m \geq 2n \log q$. ***Assuming that*** $\mathrm{LWE}_{q,\chi}$ ***is hard, the IBE system is anonymous and CPA-secure.***

7 TRAPDOORS FOR LATTICES

The notion of trapdoors in lattices often refers to a matrix with short norm. In the literature, two types of trapdoors are known. The "strong" trapdoors are a full-rank basis for a basis lattice given with a parity-check matrix $\mathbf{S} \subseteq \Lambda^{\perp}(\mathbf{A})$, which can be used for constructing preimage sampleable trapdoor functions (PSF) and a newer notion of trapdoors known as the gadget-based trapdoors, in which the trapdoor matrix transforms the public basis to a special type of gadget matrix that has useful properties for inverting hard problems. They are also much faster and highly parallelizable.

7.1 Preimage Sampleable Trapdoor Functions (PSFs)

We can say that the origin of PSFs, one way of invertion is an abstraction called ***trapdoor permutations***, which is a public bijective (e.g. every domain elements corresponds to a unique image on the image range) function f together with a trapdoor f^{-1}. Given x, $y = f(x)$ should be efficiently computable by everyone. However, it should remain hard to find x given any random y, without the trapdoor f^{-1}. The preimage sampleable functions appear as a relaxation of trapdoor permutation, where a random input may have many preimages.

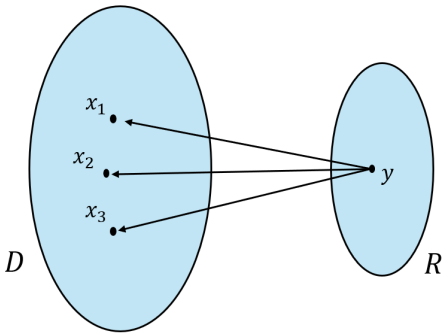


Figure 4: Preimage Sampleable Functions (PSFs)

Moreover, there are several elements that map to the same element on range R since the domain is bigger. We assume that there is a distribution over the domain, and that a value x chosen according to this distribution should produce a uniform distribution over the output range. Moreover, it still should be hard to find any of preimages for given a random sample from range. The trapdoor function f^{-1} has a nice property that it allows to randomly sample over the same distribution of domain. In the case of lattices, we apply this intuition for building a public function $f_{\mathbf{B}}$ on lattice $\Lambda_{\mathbf{B}}$, selecting a lattice point $\mathbf{v}$ and perturbing it with a small error vector $\mathbf{e}$. Given a random point $\mathbf{y}$, the trapdoor function $f^{-}1_{\mathbf{B}}$ with a trapdoor basis $\mathbf{T}$ will sample among (sufficiently close) nearby lattice points under some appropriate distribution such as discrete gaussian distribution over $\Lambda_{\mathbf{B}}$,conditioned on $f_{\mathbf{B}}(\mathbf{e}) = \mathbf{y}$. We choose the perturbation errors from a narrower distribution, however, the error is large enough that many preimages exist.

7.2 Definitions of PSFs

We define the preimage sampleable functions as authors of [23], they consist of 3 polynomial-time algorithms TrapGen, SampleDom, SamplePre:

1. **Trapdoor generation**: TrapGen(1^n) returns (a, t), that a is a public parameter and describes

   $$f_a : D_n \to R_n$$

   for some domain D_n and range R_n that depend on n and the value t serves as a trapdoor for f_a.

2. **Domain sampling**: The function SampleDom(1^n) samples an element x from D_n over some distribution, ensuring that the distribution of $f_a(x)$ is uniform over R_n.

3. **Sampling preimages with trapdoor**: Given $y \in R_n$, the function $\mathsf{SamplePre}(t, y)$ samples from $x \leftarrow \mathsf{SampleDom}(1^n)$, given the condition $f_a(x) = y$.

A PSF collection also satisfies the **one-wayness** property where for any probabilistic polynomial-time adversary $\mathcal{A}$, the probability that $A(1^n, a, y) \in f_a^{-1}(y) \subseteq D_n$ is negligible. This probability is taken over the choice of a, the uniformly random selection of $y \leftarrow R_n$, and the coin randomness of A.

7.3 Constructing PSF on lattices

Advanced cryptographic tools such as identity-based encryption from lattices can be realized using trapdoor functions. In this section, we will demonstrate how to invert Short Integer Solution problem using different trapdoor notions. The initial trapdoor notion was due to Goldreich et al. [25] where they introduced the idea of using a short basis as a secret key for signature schemes, given a point, they were applying nearest-plane algorithm with good basis to find a nearby lattice point as a signature. Despite the use of short basis remained by other approaches, the scheme was totally broken by Nguyen and Regev [32], as the scheme leaks the secret short basis after some number of signatures. Ajtai showed [5] how to generate a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ statistically close to uniform with a full rank trapdoor $\mathbf{T} \in \mathbb{Z}^{m \times m}$ that $\mathbf{AT} = \mathbf{0} \mod q$, also how to generate a basis with one or more short lattice vectors in it. It was then improved by [6], [30],[21] by shortening the bound distance L on trapdoor, the dimension, etc.

7.3.1 Generating basis with a trapdoor

In this section, we briefly show the trapdoor construction of [6] for generating hard random lattices with good basis. The approach of the construction is to extend auniformly random matrix $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times m_1}$ by generating a structured $\mathbf{A}_2 \in \mathbb{Z}_q^{n \times m_2}$ matrix, and concetanating them into a random looking parity check matrix $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2] \in \mathbb{Z}_q^{n \times m}$ for the lattice, with the short basis $\mathbf{T} \in \mathbb{Z}^{m \times m}$ which satisfies $\mathbf{AT} = \mathbf{0} \pmod q$. The number of columns m of the matrix $\mathbf{A}$ should be sufficiently large that it is hard to solve the SIS problem.

Lemma 7.1 ([6]). ***Let*** $\ell = \lceil \log_r q \rceil$. ***Given*** $\delta > 0$***, there exists a PPT algorithm on given inputs the integer*** $r \geq 2$***, a uniformly random*** $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times m_1}$ ***with*** $m_1 \geq d = (1 + \delta) n \log q$***, and any integer*** $m_2 \geq m_1 \cdot \ell$ ***(in unary), outputs matrices*** $\mathbf{U}, \mathbf{G}, \mathbf{R}, \mathbf{P}, \mathbf{C} = \mathbf{I}$

- $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$ ***has statistical distance*** $(m_2 \cdot q^{-\delta n/2})$ ***with uniformly random over*** $\mathbb{Z}_q^{n \times m}$
- ***the short basis*** $\mathbf{T}$ ***where*** $\|\mathbf{T}\| \leq 2r\sqrt{m_1 + 1}$***, which is short.***

Construction. Start by computing the Hermite normal form $\mathbf{H} \in \mathbb{Z}^{m_1 \times m_1}$ be the of basis of the lattice $\Lambda^{\perp}(\mathbf{A}_1)$. First, we find a basis for the lattice of $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times m}$ which consists of set of all vectors satisfying the following equality

$$\Lambda^{\perp}(\mathbf{A}_1) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}_1 \mathbf{z} = \mathbf{0} \bmod q\}.$$

Canonical Basis We extract the first n columns of matrix $\mathbf{A}$, forming an $n \times n$ square submatrix that is invertible modulo q, in other case, we can apply permuation on its columns. Having $\mathbf{A} = [\mathbf{H} \mid \mathbf{A}']$ with invertible $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, we can change the parity-check matrix $[\mathbf{I}_n \mid \bar{\mathbf{A}}]$ without changing the lattice where $\bar{\mathbf{A}}$ represents

$$\bar{\mathbf{A}} = \mathbf{H}^{-1}\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-n)}.$$

Using this matrix, we can convert it to a basis for the lattice $\mathcal{L} = L^{\perp}([\mathbf{I}_n \mid \bar{\mathbf{A}}]) \subset \mathbb{Z}^m$ in Hermite normal form efficiently:

$$\mathbf{H} = \begin{bmatrix} q\mathbf{I}_n & -\bar{\mathbf{A}} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{bmatrix} \in \mathbb{Z}^{m \times m}.$$

After finding the HNF basis, set $\mathbf{H}' = \mathbf{H} - \mathbf{I}_m$

Construction of G. We construct $\mathbf{G}$ as the concatenation of m_1 blocks of matrices $\mathbf{G}^{(i)}$ of dimension $n \times \ell$, followed by a zero matrix with $m_2 - m_1 \cdot \ell$ columns to complete the total of m_2 columns:

$$\mathbf{G} = \left[\mathbf{G}^{(1)} \mid \cdots \mid \mathbf{G}^{(m_1)} \mid \mathbf{0}\right] \in \mathbb{Z}_q^{m_1 \times m_2}$$

We denote the columns of matrices $\mathbf{G}^{(i)}$ with $\mathbf{g}_j^{(i)}$ and $\mathbf{H}'$ with $\mathbf{h}'_j$. We compute the columns of $\mathbf{G}^{(i)}$ $i \in [m_1]$ with recursion: assign $\mathbf{g}_\ell^{(i)} = \mathbf{h}'_i$, and the lefthand columns are defined recursively

$$\mathbf{g}_j^{(i)} = \left\lfloor \frac{\mathbf{g}_{j+1}^{(i)}}{r} \right\rceil = \left\lfloor \frac{\mathbf{h}'_i}{r^{\ell - j}} \right\rceil,$$

for each $j = \ell - 1, \ldots, 1$ and the entries of $\mathbf{g}_1^{(i)}$ will fit in the range $[0, r-1]$.

Construction of P. The columns of this matrix will be extract columns of standart basis for $\mathbb{Z}^{m_2}$ where we assign $\mathbf{p}_j = \mathbf{e}_{j\ell} \in \mathbb{Z}^{m_2}$ for $j \in [m_1]$. We specificaly design $\mathbf{P}$ that its j-th column matches with the rightmost column of $\mathbf{G}^{(j)}$, resulting $\mathbf{GP} = \mathbf{H}'$. Since we use the standard basis, all column vectors of $\mathbf{P}$ are shot $\|\mathbf{p}_j\|_2 = 1$, for $j \in [m_1]$.

Construction of U. We first construct a smaller submatrix $\mathbf{K}_\ell \in \mathbb{Z}^{\ell \times \ell}$, which is a upper-triangular matrix, having diagonal entries equal to 1 and upper diagonal entries equal to $-r$ (i.e., $t_{i,i+1} = -r$ for every $i \in [\ell - 1]$), and fill the rest entries with 0. Let $\mathbf{K}_\ell \in \mathbb{Z}^{\ell \times \ell}$ be defined as:

$$\mathbf{K}_\ell = \begin{bmatrix} 1 & -r & 0 & \cdots & 0 & 0 \\ 0 & 1 & -r & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -r \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

It is easy to see that $\mathbf{K}_\ell$ is unimodular. The matrix $\mathbf{U} \in \mathbb{Z}^{m_2 \times m_2}$ is a block-diagonal matrix of m_1 blocks $\mathbf{K}_\ell$

$$\mathbf{U} = \operatorname{diag}(\mathbf{K}_\ell, \ldots, \mathbf{K}_\ell, \mathbf{I}) = \begin{bmatrix} \mathbf{K}_\ell & & & \\ & \ddots & & \\ & & \mathbf{K}_\ell & \\ & & & \mathbf{I} \end{bmatrix}$$

followed by a identity matrix of dimension $m_2 - m_1 \cdot \ell$ to reach m_2.

We observe that $\mathbf{U}$ is also unimodular and its columns are bounded by $\|\mathbf{u}_j\|_2 \leq r^2 + 1$ for all j. Its multiplication with $\mathbf{G}$ yields

$$\mathbf{GU} = \left[\mathbf{G}^{(1)}\mathbf{K}_\ell \mid \cdots \mid \mathbf{G}^{(m_1)}\mathbf{K}_\ell \mid \mathbf{0}\right].$$

whose columns are in the range $[0, r-1]$.

Construction of R. Let $d = (1+\delta)n \log q$. The first d rows of $\mathbf{R}$ are chosen from a random variable over $\{0, \pm 1\}$, where 0 appears $1/2$ probability, ± 1 each with probability $1/4$ which can be implemented using ***rejection sampling*** (however, only random 0-1 values would also work). The remaining part is zero matrix of shape $(m_1 - d) \times m_2$. Clearly, column vectors of $\mathbf{R}$ are bounded by d, that is $\|\mathbf{r}_j\|_2 \leq d$ for all j

Algorithm 2 Constructing $\mathbf{A}$ and Basis $\mathbf{T}$ of $\Lambda^{\perp}(\mathbf{A})$

1: **Input:** $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times m_1}$ and dimension m_2 (in unary)
2: Compute matrices $\mathbf{G}, \mathbf{R} \in \mathbb{Z}^{m_1 \times m_2}$, $\mathbf{U} \in \mathbb{Z}^{m_2 \times m_2}$, and $\mathbf{P} \in \mathbb{Z}^{m_2 \times m_1}$
3: Compute $\mathbf{A}_2 = -\mathbf{A}_1 \cdot (\mathbf{R} + \mathbf{G}) \in \mathbb{Z}_q^{n \times m_2}$, and let $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$
4: Compute $\mathbf{T} = \begin{pmatrix} (\mathbf{G}+\mathbf{R})\mathbf{U} & \mathbf{RP} - \mathbf{C} \\ \mathbf{U} & \mathbf{P} \end{pmatrix} \in \mathbb{Z}^{m \times m}$
5: **Return** $\mathbf{A}$ and $\mathbf{T}$

The generated matrix $\mathbf{A}_2 = -\mathbf{A}_1 \cdot (\mathbf{R} + \mathbf{G})$ is close to uniformly random over integers modula q, and based on the ***Left Over Hash Lemma*** [26], we have that $\mathbf{A} = [\mathbf{A}_1 \mid -\mathbf{A}_1(\mathbf{G} + \mathbf{R})]$ has the statistical distance $(m_2 \cdot q^{-\delta n/2})$ with $\mathbb{Z}_q^{n \times m}$. By the design of the matrices, $\mathbf{G}$ has the m_1 columns of $\mathbf{H}'$, and $\mathbf{P}$ have the identity vectors that selects exact such columns to satisfy $\mathbf{GP} = \mathbf{H}'$. We refer readers to the original paper for further reading [6]. The following figure shows the block structure of the result of the latter algorithm.

$$n\left\{\underbrace{\left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{A}_2 \end{array}\right]}_{m_1 \quad m_2}\right. \underbrace{\left[\begin{array}{c|c} (\mathbf{G}+\mathbf{R})\mathbf{U} & \mathbf{RP}-\mathbf{C} \\ \hline \mathbf{U} & \mathbf{P} \end{array}\right]}_{m_2 \quad m_1} \begin{array}{l} \left.\vphantom{\begin{array}{c} a \\ b \end{array}}\right\} m_1 \\ \left.\vphantom{\begin{array}{c} a \\ b \end{array}}\right\} m_2 \end{array} = \mathbf{0} \in \mathbb{Z}_q^{n \times m}$$

Figure 5: Hard lattice $\mathbf{A}$ with full-rank trapdoor $\mathbf{T}$, that $\mathbf{AT} = \mathbf{0} \mod q$

7.3.2 *PSFs from lattices*

The collection of PSFs is typcially parametrized by a the width of Gaussian measure $s \geq L \cdot \omega\left(\sqrt{\log m}\right)$, depending on the trapdoor. One can use the trapdoor construction from previous

section to define a hard instance of ISIS problem with a short basis and then use to build a PSF as shown in [23].

- The $\mathsf{TrapGen}(1^n)$ function may use one of trapdoor construction methods to generate a pair of basis $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ which is statistically close to uniform and full-rank matrix $\mathbf{T} \subset \Lambda^{\perp}(\mathbf{A})$ which is a short basis where its Gram-Schmidt basis is bounded by L, and it outputs $(\mathbf{A}, \mathbf{T})$ pairs. The matrix $\mathbf{A}$ defines the public function $f_{\mathbf{A}}(\cdot)$, and the short basis $\mathbf{T}$ serves as the trapdoor.

- For the matrix $\mathbf{A}$, f_A computes $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{Ae} \mod q$ on domain $\mathcal{D}_n = \{\mathbf{e} \in \mathbb{Z}^m : \|\mathbf{e}\| \leq s\sqrt{m}\}$ where the vector $\mathbf{e}$ is bounded and distributed on $D_{\mathbb{Z}^m, s}$ as it can be sampled with standart basis of $\mathbb{Z}^m$ using SampleD. The range $\mathcal{R}_n = \mathbb{Z}_q^n$, set of n dimensional vectors module q.

- We define $\mathsf{SampleISIS}(\mathbf{A}, \mathbf{T}, s, \mathbf{u})$ which samples from $f_m^{\mathbf{A}^{-1}}(\mathbf{u})$:

  1. Using linear algrebra, compute an arbitrary $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{At} \equiv \mathbf{u} \mod q$
  2. Sample $\mathbf{v} \sim D_{\Lambda^{\perp}, s, -\mathbf{t}}$ via $\mathtt{SampleDGS}(\mathbf{T}, s, -\mathbf{t})$ using the "good" basis, and output $\mathbf{e} = \mathbf{t} + \mathbf{v}$, extracting a close lattice point from the solution yields a short solution.

The described procedure gives us a PSF under hardness of ISIS problem and will be used for IBE schemes:

Theorem 7.1 ([23]). ***The described procedure gives a collection of one-way PSFs if*** $\mathsf{ISIS}_{n,m,q,s\sqrt{m}}$ ***is hard.***

7.4 Gadget-based trapdoor functions

Micciancio and Peikert [30] introduced a new notion of trapdoor with plenty of attractive features. Theoretical trapdoor constructions include finding Hermite Normal Form (HNF) or inverse calculations of matrices with large dimentions, and the norm of such trapdoors is much larger. The gadget-based trapdoors are much simpler and efficient, easy to implement using matrix multiplications, and have smaller key sizes in constant factors with respect to previous schemes. This trapdoor is a invertible ***linear*** transform that converts the uniform basis to a gadget matrix, rather than an actual basis.

Theorem 7.2 ([30]). ***Given integer parameters*** $n \geq 1$, $q \geq 2$, ***large enough*** $m = O(n \log q)$, ***a randomized polynomial-time*** $\mathsf{GenTrap}(1^n, 1^m, q)$ ***algorithm exists that yields a parity-check matrix*** $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ ***which is negligibly close to the uniform distribution and a 'trapdoor' short matrix*** $\mathbf{R}$.

Furthermore, we have polynomial time algorithms Invert ***and*** SampleD ***that satisfy following inversion and sampling algorithm for all random choises with overwhelming probability:***

- ***Given an*** LWE ***instance*** $\mathbf{b}^{\top} = \mathbf{s}^{\top}\mathbf{A} + \mathbf{e}^{\top}$, ***in which*** $\mathbf{s} \in \mathbb{Z}_q^n$ ***is arbitrary and either the error is bounded*** $\|\mathbf{e}\| < q/O(\sqrt{n \log q})$ ***or sampled from*** $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$ ***with*** $1/\alpha \geq \sqrt{n \log q} \cdot \omega(\sqrt{\log n})$, ***the algorithm*** $\mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b})$ ***inverts*** $\mathbf{s}$ ***and*** $\mathbf{e}$ ***deterministically.***

- ***Given any vector*** $\mathbf{u} \in \mathbb{Z}_q^n$ ***and large enough Gaussian parameter*** $s = O(\sqrt{n \log q})$***, the output distribution of randomized sampling algorithm*** $\mathsf{SampleD}(\mathbf{R}, \mathbf{A}, \mathbf{u}, s)$ ***has negligible statistical distance from*** $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}),\, s \cdot \omega(\sqrt{\log n})}$.

For our purposes in IBE, we are interested in the second property for randomly sampling from vector domain on given identity and define how to sample with a different trapdoor notion. First we define the gadget matrix, also known as ***primitive matrix*** is a highly structued and publicly known to everyone, and the trapdoors for this parity check matrix is precomputed.

The properties of the gadget matrix will be publicly known and it also allows for a lot of optimizations. Let $q \geq 2$ be an integer modulus and $k \geq 1$ be an integer dimension where q is polynomial in n, e.g. $q = q(n)$ in most constructions. We first start with defining a ***primitive vector*** $g \in \mathbb{Z}_q^k$, whose entries satisfy that $\gcd(g_1, \ldots, g_k, q) = 1$. The vector g forms a k-dimensional lattice $\Lambda^{\perp}(g^t) \subset \mathbb{Z}^k$ which has determinant $\det\left(\Lambda^{\perp}(g^T)\right) = q$. We can define a concrete instantiation of the gadget vector whose entries form a geometrically increasing sequence as follows, due to [30]:

$$\mathbf{g} := \begin{bmatrix} 1 \\ 2 \\ 4 \\ \vdots \\ 2^{k-1} \end{bmatrix} \in \mathbb{Z}_q^k, \quad k = \lceil \log_2 q \rceil$$

We now consider the primitive lattice $\Lambda^{\perp}(\mathbf{g}^T) \subset \mathbb{Z}^k$, where the modulus q may be either a power of two or a prime, where the power-of-two case be generalized to arbitrary prime moduli q and give concrete short basis for it. Untill more recently, there were not known reductions of security for q in case of being a power of small prime despite its advantages that we will see. to Since our applications rely on prime modulus, we will only define the general case. Now notice the following full rank matrix

$$\mathbf{S}_k := \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{bmatrix} \in \mathbb{Z}_q^{k \times k},$$

The matrix $\mathbf{S}_k$ is a basis of $\Lambda^{\perp}(\mathbf{g}^{\mathsf{T}})$ of $\mathbf{g}^T$ as $\mathbf{g}^T \cdot \mathbf{S}_k = \mathbf{0} \bmod q$ holds, and its determinant $\det(\mathbf{S}_k) = q$. Furthermore, the basis vectors are short and have squared length $\|\mathbf{s}_i\|^2 \leq 5$ for $i \leq k$. The good Gram–Schmidt orthogonalization is another observation we have as $\widetilde{\mathbf{S}_k} = 2\mathbf{I}_n$.

Gaussian Sampling for $f_{\mathbf{g}^T}$. If q is a power-of-two, then sampling algorithms become very trivial as we will define. Now we desire to efficently sample for the function $f_{\mathbf{g}^T}$ under Gaussian like distribution from the given coset of $\Lambda^{\perp}(\mathbf{g}^T)$, e.g. to sample a vector with probability of $\rho_s(\mathbf{x})$ from the set

$$\Lambda_u^{\perp}(\mathbf{g}^T) := \left\{ \mathbf{x} \in \mathbb{Z}^k : \langle \mathbf{g}, \mathbf{x} \rangle = u \bmod q \right\}$$

for a syndrome $u \in \mathbb{Z}_q$. A Gaussian parameter bigger than the optimal bound on the smoothing parameter of $\Lambda^{\perp}(\mathbf{g}^T)$, $s \geq \|\widetilde{\mathbf{S}}_k\| \cdot r = 2 \cdot \omega(\sqrt{\log n})$ suffices to sample over desired discrete Gaussian distribution. We mention two approaches for this purpose, due to [30]

The first method, also known as 'bucketing' approach, involves storing a large number of independent samples $\mathbf{x} \leftarrow D_{\mathbb{Z}^k,s}$ in advance, and keeping the tuples of $\mathbf{x}$ and the value of $\mathbf{u} = \langle \mathbf{g}, \mathbf{x} \rangle \in \mathbb{Z}_q$ in a storage, until we have every possible image in $\mathbb{Z}_q$. However, this approach is expensive in the storage and precomputation as a large q may imply lots of such tuples to be computed.

The second approach is to use discrete Gaussian sampling algorithm with the good basis $\mathbf{S}_k$ as it becomes very simple and efficient. Due to the geometrical structure of the basis, it is identical to the following algoerithm:

Algorithm 3 Sampleg - Recursive Gaussian Sampling for $\Lambda_u^{\perp}(\mathbf{g}^T)$

1: **Input**: Syndrome $u \in [q-1]$, Gaussian parameter s, dimension k
2: Initialize empty vector $\mathbf{x} \in \mathbb{Z}^k$
3: **for** $i = 0$ to $k-1$ **do**
4: Sample $x_i \leftarrow D_{2\mathbb{Z}+u,s}$
5: Update $u \leftarrow (u - x_i)/2$
6: **end for**
7: **Return** $\mathbf{x} = (x_0, \ldots, x_{k-1})$

The algorithm simply chooses a random Gaussian, and computes a single entry of the preimage vector.

Gadget matrix. The gadget matrix consist of $\mathbf{g}$, the associated trapdoor basis $\mathbf{S}_k$ is used to define the bigger basis:

$$\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}^T = \begin{bmatrix} \cdots \mathbf{g}^T \cdots & & & \\ & \cdots \mathbf{g}^T \cdots & & \\ & & \ddots & \\ & & & \cdots \mathbf{g}^T \cdots \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}$$

and the associated basis for $\mathbf{G}$ will be the block-diagonal matrix, the tensor product of identity matrix with $\mathbf{S}_k$

$$\mathbf{S} = \mathbf{I}_n \otimes \mathbf{S}_k = \begin{bmatrix} \mathbf{S}_k & & & \\ & \mathbf{S}_k & & \\ & & \ddots & \\ & & & \mathbf{S}_k \end{bmatrix} \in \mathbb{Z}^{nk \times nk}.$$

We can also think of $\mathbf{G}$, $\Lambda^{\perp}(\mathbf{G})$, and $\mathbf{S}$ as n copies of $\mathbf{g}^{\top}$, $\Lambda^{\perp}(\mathbf{g}^{\top})$, and $\mathbf{S}_k$, respectively. The matrix $\mathbf{G}$ is also a primitive matrix, its lattice $\Lambda^{\perp}(\mathbf{G}) \subset \mathbb{Z}^{nk}$ has determinant q^n, and hence $\mathbf{S}$ is a natural basis for this lattice. Moreover, it is immediate that $\|\mathbf{S}\| = \|\mathbf{S}_k\|$.

It is worth to mention that $\mathbf{G}$ is not full rank, as full rank for $\mathbb{Z}_q$ is not well defined when the modula element q is not prime. However, the columns of $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ generates all of the vectors in $\mathbb{Z}_q^n$ where $\mathbf{G} \cdot \mathbb{Z}^m = \mathbb{Z}_q^n$, as it is a primitive matrix.

We mention the properties of the gadget matrix in the following theorem:

Theorem 7.3 ([30]). ***Let*** $q \geq 2$, $n \geq 1$, $k = \lceil \log_2 q \rceil$ ***some integers and set*** $m = nk$. ***Then, there exists a primitive matrix*** $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ ***such that:***

- ***The lattice generated by parity check matrix*** $\Lambda^{\perp}(\mathbf{G})$ ***has a good basis*** $\mathbf{S} \in \mathbb{Z}^{m \times m}$. ***If*** $q = 2^k$, ***we have*** $\widetilde{\mathbf{S}} = 2\mathbf{I}$ ***(so*** $\|\widetilde{\mathbf{S}}\| = 2$***) and*** $\|\mathbf{S}\| = \sqrt{5}$. ***If*** q ***is an arbitrary number then we have*** $\|\widetilde{\mathbf{S}}\| \leq \sqrt{5}$ ***and*** $\|\mathbf{S}\| \leq \max\{\sqrt{5}, \sqrt{k}\}$. ***Moreover, both*** $\mathbf{G}$ ***and*** $\mathbf{S}$ ***matrices require little storage and sparse.***
- ***Preimages for*** $f_{\mathbf{G}}(\mathbf{x}) = \mathbf{G}\mathbf{x} \bmod q$ ***can be sampled in quasilinear*** $O(n \cdot \log^c n)$ ***time with Gaussian parameter*** $s \geq \|\widetilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$. ***Additionally, it can parallelized to polylogarithmic*** $O(\log^c n)$ ***time using*** n ***processors using the preimage sampler for*** $f_{\mathbf{g}^T}(\mathbf{x})$.

The above statements hold for any integer $b \geq 2$ with $k = \lceil \log_b q \rceil$, $\|\widetilde{\mathbf{S}}\| \leq \sqrt{b^2+1}$, and $\|\mathbf{S}\| \leq \max\{\sqrt{b^2+1}, (b-1)\sqrt{k}\}$. When $q = b^k$ is exact power, it yields $\widetilde{\mathbf{S}} = b\mathbf{I}$ and $\|\mathbf{S}\| = \sqrt{b^2+1}$.

7.4.1 Gaussians with covariance

We define a different continuous n-dimensional Gaussian function as $\tilde{\rho}(\mathbf{x}) \triangleq \exp(-\pi \cdot \|\mathbf{x}\|^2) = \exp(-\pi \cdot \langle \mathbf{x}, \mathbf{x} \rangle)$. We can represent the Gaussian function transformed by a atrix $\mathbf{B}$ with linearly independent columns as

$$\tilde{\rho}_{\mathbf{B}}(\mathbf{x}) \triangleq \begin{cases} \rho(\mathbf{B}^{+}\mathbf{x}) = \exp(-\pi \cdot \mathbf{x}^T \Sigma^{+} \mathbf{x}) & \text{if } \mathbf{x} \in \text{span}(\mathbf{B}) = \text{span}(\Sigma), \\ 0 & \text{otherwise,} \end{cases}$$

where $\Sigma = \mathbf{B}\mathbf{B}^T \geq 0$ and we refer to it as $\tilde{\rho}_{\sqrt{\Sigma}}$. We get continuous Gaussian distribution $D_{\sqrt{\Sigma}}$ by normalizing $\rho_{\sqrt{\Sigma}}$ over its span, having covariance $\mathbb{E}_{\mathbf{x} \leftarrow D_{\sqrt{\Sigma}}}[\mathbf{x}\mathbf{x}^T] = \frac{\Sigma}{2\pi}$ (the $\frac{1}{2\pi}$ factor is omitted). We denote Σ as the covariance matrix of $D_{\sqrt{\Sigma}}$.

Given a point $\mathbf{c} \in \mathbb{R}^n$ and a positive semidefinite $\Sigma \geq 0$ matrix that $(\Lambda + \mathbf{c}) \cap \text{span}(\Sigma)$ is nonempty, the discrete Gaussian distribution $D_{\Lambda+\mathbf{c},\sqrt{\Sigma}}$ is proportional to the continuous Gaussian distribution $D_{\sqrt{\Sigma}}$ assign nonzero probability to the coset $\Lambda + \mathbf{c}$, that is

$$D_{\Lambda+\mathbf{c},\sqrt{\Sigma}}(\mathbf{x}) = \frac{\rho_{\sqrt{\Sigma}}(\mathbf{x})}{\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c})} \propto \rho_{\sqrt{\Sigma}}(\mathbf{x}).$$

7.4.2 Trapdoor construction from Gadget matrices

In GPV sampling algorithm and subsequent approaches, the sampling algorithm works on a random looking parity check matrix and a "strong" high ***quality*** matrix trapdoor for finding a nearby lattice point. The quality of lattice trapdoor can be the maximal euclidian length of the basis itself or its Gram-Schmidt orthogonalization. Here, the trapdoor quality is measured with the maximal ***singular value*** of the basis, defined as $s_1(\mathbf{T})$. Now we show how to build sampler algorithm SampleD defined in the 7.2 using the gadget matrices. Typically we set $w = n \cdot \lceil \log q \rceil$.

Definition 7.1 ([30]). For $m \geq w \geq n$, consider two matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{G} \in \mathbb{Z}_q^{n \times w}$. A matrix $\mathbf{R} \in \mathbb{Z}_q^{(m-w) \times w}$ is called $\mathbf{G}$-***trapdoor*** for $\mathbf{A}$ such that

$$\mathbf{A} \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{H}\mathbf{G}$$

for non-singular matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, known as the ***label*** or ***tag*** of the trapdoor. As mentioned earliner, the largest singular value of $s_1(\mathbf{R})$ defines its the quality.

We have constructed the matrix $\mathbf{G}$ to be a primitive matrix, hence if $\mathbf{A}$ has a $\mathbf{G}$-trapdoor, then $\mathbf{A}$ is also primitive matrix. Moreoever, it has the same determinant with $\det(\Lambda^{\perp}(\mathbf{A})) = \det(\Lambda^{\perp}(\mathbf{G})) = q^n$. For simplicity, we assume $\mathbf{H}$ is identity matrix.

The following algorithm generates a pair of pseudorandom matrix $\mathbf{A}$ with a $\mathbf{G}$-trapdoor. Given a random matrix, it is extended with the primitive matrix $\mathbf{G}$ into $\mathbf{A}' = [\bar{\mathbf{A}} \mid \mathbf{G}]$ semirandom matrix. It is followed by a random linear transformation defined by

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{Z}_q^{m\times m}$$

to the semi-random lattice . The unimodular matrix $\mathbf{T}$ has unimodular with inverse $\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$. Applying the linear transformation to the semi-random lattice $\Lambda^{\perp}(\mathbf{A}')$ yields

$$\mathbf{T} \cdot \Lambda^{\perp}(\mathbf{A}_0) = \Lambda^{\perp}(\mathbf{A}_0 \cdot \mathbf{T}^{-1})$$

associated with the parity-check matrix

$$\mathbf{A} = \mathbf{A}' \cdot \mathbf{T}^{-1} = [\bar{\mathbf{A}} \mid \mathbf{HG} - \mathbf{AR}].$$

Based on the Left over hash lemma, the distribution of $[\bar{\mathbf{A}} \mid \mathbf{0}] \cdot \mathbf{T}^{-1} = [\bar{\mathbf{A}} \mid -\mathbf{AR}]$ is close to uniform, so is $\mathbf{A}$.

Algorithm 4 GenTrap – Generate a Matrix with a $\mathbf{G}$-Trapdoor

Input: Random matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times \bar{m}}$, primitive matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$, distribution $\mathcal{D}$ over $\mathbb{Z}_q^{\bar{m}\times w}$
Output: Matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with a $\mathbf{G}$-trapdoor, and trapdoor matrix $\mathbf{R}$

1: Sample a random matrix $\mathbf{R} \in \mathbb{Z}_q^{\bar{m}\times w}$ from distribution $\mathcal{D}$
2: Set: $\mathbf{A}' = [\bar{\mathbf{A}} \mid \mathbf{G}]$
3: Apply transformation: $\mathbf{A} = \mathbf{A}' \cdot \mathbf{T}^{-1} = [\bar{\mathbf{A}} \mid \mathbf{G} - \bar{\mathbf{A}}\mathbf{R}]$
4: Output matrix $\mathbf{A}$ and trapdoor $\mathbf{R}$

The distribution $\mathcal{D}$ used in above algorithm is a subgaussian with some parameter $s > 0$ (or δ-subgaussian for some small δ), yielding $s_1(\mathbf{R}) = s \cdot \mathcal{O}(\sqrt{\bar{m}} + \sqrt{w})$ with overwhelming probability. We can use a previus probability distribution used in [6] trapdoor generation where $\mathcal{P}$ is random variable over $\mathbb{Z}$ that outputs 0 with probability $1/2$, and ± 1 each with probability $1/4$. Then $\mathcal{D}$ is 0-subgaussian with parameter $\sqrt{2\pi}$. Also for a uniformly random matrix $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n\times \bar{m}}$,

$$\mathbf{A} = [\bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R}]$$

should be δ-uniform for some negligibly small parameter $\delta = \text{negl}(n)$. Running the GenTrap algorithm, we will have a parity check matrix $\mathbf{A}$ and its $\mathbf{G}$-trapdoor $\mathbf{R}$. To sample a preimage for given a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ from the ***spherical*** discrete Gaussian $D_{\Lambda_u^{\perp}(\mathbf{A}),s}$, we can sample from a narrower Gaussian over the primitive lattice $\Lambda^{\perp}(\mathbf{G})$.

We use a subroutine for Gaussian sampling from any coset of $\Lambda^{\perp}(\mathbf{G})$ with fixed parameter $\sqrt{\Sigma_{\mathbf{G}}} \geq \eta_{\varepsilon}(\Lambda^{\perp}(\mathbf{G}))$. We have sampling algorithms from previous subsection for which $\sqrt{\Sigma_{\mathbf{G}}}$ is either 2 or $\sqrt{5}$.

First sample a Gaussian $\mathbf{z}$ from $\Lambda_u^{\perp}(\mathbf{G})$ and compute $\mathbf{y} = \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\mathbf{z}$ as

$$\mathbf{A}\mathbf{y} = (\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix})\mathbf{z} = \mathbf{G}\mathbf{z} = \mathbf{u}$$

as desired. However, Micciancio et al. [30] showed that the output of $\mathbf{y}$'s distribution will not be spherical since $\mathbf{y}$'s distribution is an $s_1(\mathbf{R})$ times wider than the distribution of $\mathbf{z}$ over $\Lambda_u^{\perp}(\mathbf{G})$.

To fix the "skewed" distribution fo $\mathbf{y}$ into a spherical Gaussian over all of $\Lambda_u^{\perp}(\mathbf{A})$, we use the ***convolution*** technique from [34]. We perturb the desired syndrome by a short Gaussian perturbation $\mathbf{p} \in \mathbb{Z}^m$ with covariance $\Sigma_{\mathbf{p}} = s^2\mathbf{I} - \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\Sigma_{\mathbf{G}}\begin{bmatrix}\mathbf{R}^T & \mathbf{I}\end{bmatrix}$ and compute an adjusted syndrome $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p}$. We then do the same procedure for adjusted syndrome and sample $\mathbf{y} = \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\mathbf{z}$, and return $\mathbf{x} = \mathbf{p} + \mathbf{y}$. Now the support of $\mathbf{x}$ is all of $\Lambda_u^{\perp}(\mathbf{A})$, and because the covariances of $\mathbf{p}$ and $\mathbf{y}$ are additive, the overall distribution of $\mathbf{x}$ is spherical with Gaussian parameter s that can be as small as $s \approx s_1(\mathbf{R}) \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}})$.

The vector $\mathbf{x}$ lies in the full coset $\Lambda_u^{\perp}(\mathbf{A})$ and combinination of covariance of $\mathbf{p}$ and $\mathbf{y}$ yields a spherical Gaussian distribution as desired. Hence, $\mathbf{x}$ is sampled from a spherical Gaussian with parameter $s \approx s_1(\mathbf{R}) \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}})$. The algorithm below samples from a spherical Gaussian distribution for given syndrome.

Algorithm 5 SampleD- Sampling from $D_{\Lambda_u^{\perp}(\mathbf{A}),s}$ Using a $\mathbf{G}$-Trapdoor

Input: $\mathbf{A}$ with trapdoor $\mathbf{R}$, syndrome $\mathbf{u} \in \mathbb{Z}_q^n$, parameter s

Output: Preimage $\mathbf{x} \in \Lambda_u^{\perp}(\mathbf{A})$ from a spherical discrete Gaussian distribution

1: Let $\sqrt{\Sigma_{\mathbf{G}}}$ be a fixed covariance matrix such that $\sqrt{\Sigma_{\mathbf{G}}} \geq \eta_{\varepsilon}(\Lambda^{\perp}(\mathbf{G}))$
2: Compute adjusted syndrome: $\mathbf{v} = \mathbf{u} - \mathbf{A}\mathbf{p}$, where $\mathbf{p} \sim D_{\mathbb{Z}^m,\sqrt{\Sigma_{\mathbf{p}}}}$
3: Sample $\mathbf{z} \leftarrow D_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}),\sqrt{\Sigma_{\mathbf{G}}}}$
4: Compute $\mathbf{y} = \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\mathbf{z}$
5: Output $\mathbf{x} = \mathbf{p} + \mathbf{y}$

Furthermore, we can also generate a full-rank basis with knowledge of $\mathbf{R}$.

Full-rank basis with G-Trapdoors. It is also possible to generate that a good basis trapdoor $\mathbf{R}$ for the lattice $\Lambda^{\perp}(\mathbf{A})$.

Lemma 7.2 ([30]). ***Let*** $\mathbf{S} \in \mathbb{Z}^{w\times w}$ ***be any basis for*** $\Lambda^{\perp}(\mathbf{G})$. ***Let*** $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ ***have a trapdoor*** $\mathbf{R} \in \mathbb{Z}_q^{(m-w)\times w}$ ***have the basis***

$$\mathbf{S}_{\mathbf{A}} = \begin{bmatrix}\mathbf{I} & \mathbf{R}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{W} & \mathbf{S}\end{bmatrix},$$

where $\mathbf{W} \in \mathbb{Z}^{w \times \bar{m}}$ ***is any solution to*** $\mathbf{GW} = -\mathbf{A}[\mathbf{I} \mid \mathbf{0}]^T \mod q$. ***The basis*** $\mathrm{S_A}$ ***satisfies the following when*** $\mathrm{S_A}$ ***is orthogonalized in a suitable order.***

$$\|\widetilde{\mathbf{S_A}}\| \leq s_1 \left(\begin{bmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right) \cdot \|\widetilde{\mathbf{S}}\| \leq (s_1(\mathbf{R}) + 1) \cdot \|\widetilde{\mathbf{S}}\|,$$

8 IBE REALIZATION OVER LATTICES

Aside from general definition, the lattice-based IBE schemes differ in correctness definition which may fail with negligible probability. The first IBE scheme from lattices, namely the GPV scheme, is introduced by Gentry et al. [23], which forms the ground of many furhter schemes relying on full-rank "strong" trapdoors and also is the main construction of our interest. The IBE system relies on preimage sampleable functions and parametrized by $r \geq L \cdot \omega(\sqrt{\log m})$ to guarantee preimage sampleability. The second scheme is upon the MP-trapdoor from [30] whose inversion algorithm relies on gadget-based trapdoors. The security of both schemes rely on ***random oracle***, a black box model that enables secure schemes in theory, and in this context it is defined as a random function $H : \{0,1\}^* \to \mathbb{Z}_q^n$ that maps identities to uniform public keys in $\mathbb{Z}_q^n$ for the the dual cryptosystem defined in section 6.2.

- $\mathsf{Setup}(1^n)$: Use any of trapdoor functions $f_\mathbf{A}$ with a trapdoor described in previous section. The master public key is the matrix $\mathbf{A}$ which will be the public parameter for dual-Regev scheme, and the master secret key:
  - ***Option A:*** will be the short full-rank basis $\mathbf{T}$
  - ***Option B:*** will be the $\mathbf{G}$-Trapdoor matrix $\mathbf{R}$
- $\mathsf{Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{id})$: Computes the hash of identity string $\mathbf{u} = H(\mathsf{id}) \in \mathbb{Z}_q^n$ and samples a secret key $\mathbf{e} \leftarrow f_\mathbf{A}^{-1}(\mathbf{u})$ using
  - ***Option A:*** short $\mathbf{T}$, use the ISIS sampler with $\mathsf{SampleDGS}(\mathbf{T}, \mathbf{u}, \mathbf{s})$ as described in 7.3.2
  - ***Option B:*** subgaussian $\mathbf{R}$, perturb the syndrome and sample with SampleD desrcibed in 7.4.2

  Store the pair $(\mathsf{id}, \mathbf{e})$ for reuse, and output $\mathbf{e}$.
- $\mathsf{Enc}(\mathsf{MPK}, \mathsf{id}, b)$: Encrypts a bit $b \in \{0,1\}$ using dual-Regev scheme for given identity id, computes the syndrome $\mathbf{u} = H(\mathsf{id})$ and encrypts with $c = (\mathbf{c_1}, c_2) \leftarrow \mathsf{DualEnc}(\mathbf{u}, b)$.
- $\mathsf{Dec}(\mathsf{params}, \mathsf{sk_{id}}, c)$: Decrypts the ciphertext $c = (\mathbf{c_1}, c_2)$ using secret key $\mathbf{e} = \mathsf{sk_{id}}$, and output the message bit $b \leftarrow \mathsf{DualDec}(\mathbf{e}, c)$.

The above scheme can be extended to a multi-bit IBE using the multi-bit extension of the dual cryptosystem with mapped tags H that we assumed to be identity matrix.

Theorem 8.1. ***If the dual-Regev cryptosystem is secure and anonymous as described [23], and the distribtion of its public keys are statistically close to uniform over*** $\mathbb{Z}_q^n$ ***for all matrices*** $\mathbf{A}$ ***(except a negligible fraction of them), the described IBE constructions are also anonymous and secure in the random oracle model for both trapdoor functions.***

9 EXPERIMENTAL EVALUATION

9.1 Methodology

In this section, we aim to experimentally evaluate necessary building blocks of IBEs. We start with a simple realization of random oracles for mapping identites to uniformly random identity vectors over $\mathbb{Z}_q^n$, using hash functions with extendable outputs.The second step involves designing the discrete Gaussian sampling algorithm, which samples lattice points closer to the mean vector with a given Gaussian parameter s. The sampling algorithm for $n \geq 2$ dimensional spaces uses a subroutine SampleZ for one-dimensional discrete Gaussians over integers, and we evaluate its statistical distance from the corresponding discrete Gaussian distribution. Furthermore, we build a concrete realization of the Alwen-Peikert trapdoor construction and measure its execution performance and key size across different security parameters. We also measure the same metrics for gadget-based trapdoors, including the size and extraction time of secret keys.

9.2 Evaluating hash function

As mentioned earlier, the IBE scheme proposed by Gentry et al. uses the random oracle is to convert an identity to a uniform vector in $\mathbb{Z}_q^n$, however, in practice it remains as a challenge. To overcome this problem, we have implemented several approaches using hash functions. The underlying primitive of our algorithm is the SHAKE-256 (Secure Hash Algorithm Keccak) function, which provides variable-length output. It belongs to the **SHA-3 family**, standardized by the National Institute of Standards and Technology (NIST) under FIPS 202 [20]. Our idea is to extend the hash of identity to nk bits where n is the dimension of the vector and $k = \lceil \log_2(q) \rceil$ is the bit length required to represent the module element, as we need k bits to span the space $\mathbb{Z}_q$. After dividing the sequence to n parts, we achieve a distribution close to uniform for syndromes, required by the dual-regev scheme.

First Approach The basic idea was to use a result of a fixed length hash function for splitting into equal pieces. However, as the dimension increases, the bits per element remain so small to span $\mathbb{Z}_q$, which could be vulnerable as the adversary would have to search collisions in a smaller range.

Second Approach The first approach was to map values from the range $[0 : 2^k]$ to $\mathbb{Z}_q$ by reducing modulo q. Since q does not divide 2^k evenly, this introduces a bias, resulting in the elements near zero appearing more frequently.

The given example illustrates the case for 10^6 random samples strheced for $n = 15$ dimensional vectors in $\mathbb{Z}_q$, where $q = 1009$ (a prime number). The result shows that the elements in range 0-15 appeared twice, however, there were no duplicated vectors. We also tested the closeness of our distribution with respect to uniform distribution using techniques as Chi-squarred statistics which is a measure of how much the observed distribution differs from the expected (uniform) distribution and ***p-value*** which helps to understand whether the observed difference is statistically significant. For our case, a small chi-value and relatively big p-value (e.g. greater than 0.05) is enough to interpret the results. Our tests showed that chi-value were around 14766 which is relatively high for big number of samples over a small range of q, and p-value around 10^{-36} which is almost zero.

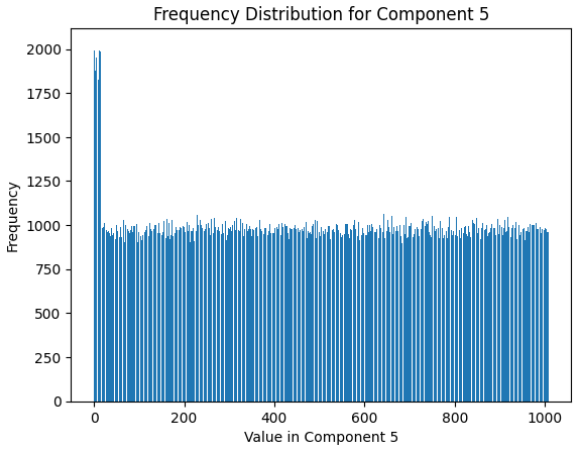


Figure 6: Reducing the range $[0 : 2^k] \mod q$

The above statistic measurements would help adversary to improve his chances, knowing that the first element have a more probability to appear.

Third Approach We employed the ***rejection sampling*** approach for achieving the desired uniform distribution. In general, we introduce the following algorithm for reducing a "uniform" distribution over $[0; b]$ to a smaller "uniform" range $[0 : a]$. Let $t = \lfloor b/a \rfloor$. if the sampled value x from first distribution issmaller than ta, reject it, otherwise, accept $x \mod a$. Following the strategy, $t = \lfloor 2^k/q \rfloor$. However, we may run out of samples from the stretched bits, thus we extend it by $1.5nk$, and assuming that the output of shake-256 is close to uniform, then we should have enough samples for sampling over the desired distribution. We applied the same experiments with rejection sampling, the chi-value was 992, and the p-value was 0.633, indicating the improvement.

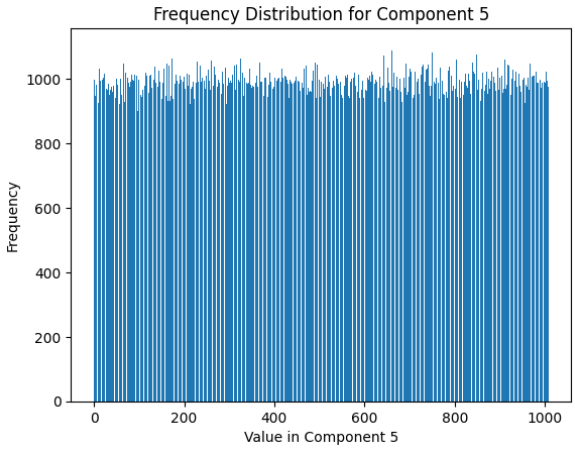


Figure 7: Rejection sampling

The measurements for 100000 samples over $q = 100003$ (the next prime after 10^5), Chi-squared

value was around 100097 and p-value was around 0.414: The low number of samples for a bigger q explains a high chi-squared value, however, it is expected to decrease as we decrease number of samples as we couldn't afford it due to computation power.

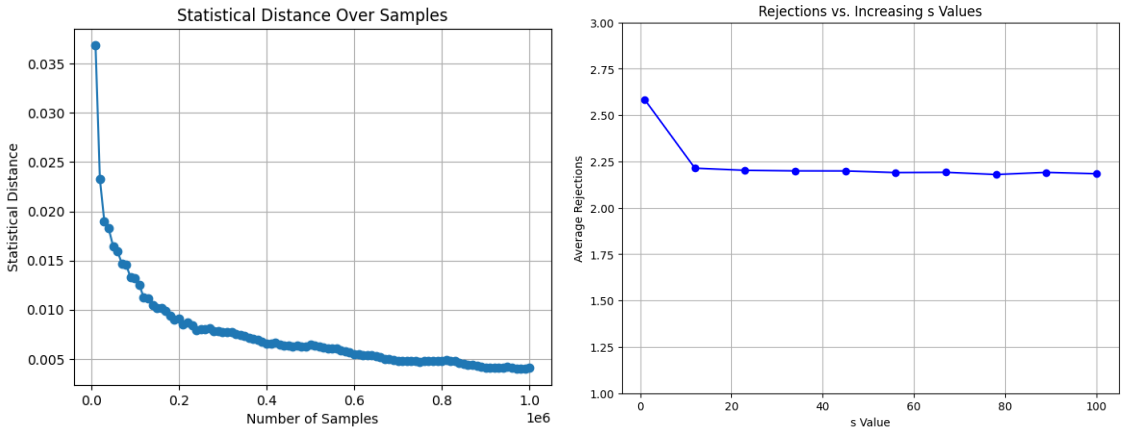
9.3 Discrete Gaussian sampling

We have implemented the discrete Gaussian sampling algorithm over integer lattices in one dimension, using rejection sampling defined in section 5.1. We have calculated the statistical distance over set $Z = \mathbb{Z} \cap [c - s \cdot t(n),\ c + s \cdot t(n)]$ between our sampling algorithm and a real discrete gaussian sampler as follows. In the following formula, μ_i denotes the empirical probability (or relative frequency) of the value i, defined as $\mu_i = n_i/n$, where n_i is the number of occurrences of i, and n is the total number of samples.

$$\Delta(D, X) = \frac{1}{2} \sum_{a \in Z} |\Pr\{D = a\} - \Pr\{X = a\}| = \frac{1}{2} \sum_{a \in Z} \left|\exp\left(-\pi \|a - c\|^2 / s^2\right) - \mu_i\right|$$

Algorithm 6 Rejection Sampling from Discrete Gaussian: SampleZ

procedure SampleZ(s, c, t)
  Set $a \leftarrow \lfloor c - s \cdot t \rceil$ and $b \leftarrow \lfloor c + s \cdot t \rceil$
  repeat
    Sample an integer $x \in \{a, \ldots, b\}$ uniformly at random
    Compute $\rho \leftarrow \exp\left((x - c)^2 / s^2\right)$
    Sample $u \sim \text{Uniform}(0, 1)$
  until $u < \rho$
  Return x
end procedure



(a) Statistical distance

(b) Rejected samples over s

Figure 8: Visualizations related to statistical sampling and distribution properties.

The decrease in statistical distance was expected as the number of samples increased, confirming that our algorithm converges toward the target distribution with more samples. Hence, our tests show that it is practically possible to sample from discrete Gaussian distribution for desired parameters. The termination of rejection sampling plays a crucial role in execution time, as it is a sequential operation due to our choose of algorithm. The second graph implies that number of rejected over samples shouldn't depend on s, meaning that a wider range of distribution support doesn't necessarly affect its peformance.

Our next experiment shows the execution performance of discrete Gaussian sampling algorithm over n dimensional lattices, with randomly generated basis. The execution time also includes computing the Gram-Schmidt orthogonalization of the basis and considering its inherently sequential property, it works in $\mathcal{O}(n^2)$ time as reflected in the following graph.

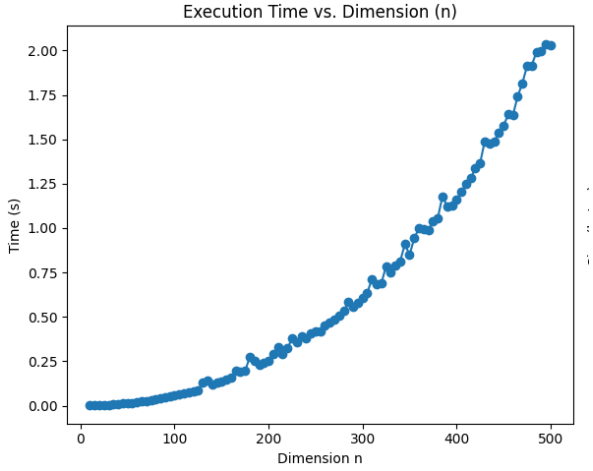


Figure 9: Execution time

9.4 Alwen-Peikert trapdoor sampling

Building the Alwen-Peikert trapdoor include many matrix multiplication, computing the Hermite Normal Form of the matrix, and the dimensions of the parity check matrix and hence its basis grow faster on security parameter: given n, the first block of parity check matrix is supposed to be bigger than are obvious in execution time and key size.In the construction, let $m_1 \geq d = (1+\delta)n \log q$, and any integer $m_2 \geq m_1 \cdot \ell = nk^2$, where $\ell = \lceil \log_2 q \rceil = k$. Therefore, the smallest dimension is approximately $\dim(\mathbf{A}) \approx n \times nk(k+1)$. Clearly, the dimension of the trapdoor is at least $\dim(\mathbf{T}) \approx nk(k+1) \times nk(k+1)$, resulting in large key sizes and high execution times due to the use of subgaussian distributions, as reflected in Figure 10. For the security parameter $n = 45$, the size of master puplic and secret keyse are approximately 110 MB, and key generation took 80 seconds to execute.

We solve the system of linear equations $\mathbf{Ax} = \mathbf{u} \mod q$ to find an arbitrary solution $\mathbf{t}$ using modified Gaussian elimination with modulus q and obtain an arbitrary solution. Moreover, our experiments showed that the result of modified Gaussian elimination tends to produce a short solution, with most elements being zero. Using the trapdoor with a large normal form $\widetilde{\mathbf{T}}$, we applied

the key extraction algorithm with SampleDGS, where the Gaussian parameter $s \geq \widetilde{\mathbf{T}}$ and the mean vector is $-\mathbf{t}$. The sampled vector was too far from the arbitrary solution, and their sum resulted in a noticeably long vector. The length of the preimages is bounded by $\beta \approx s\sqrt{m}$ as needed, however, it caused issues in the encryption process.

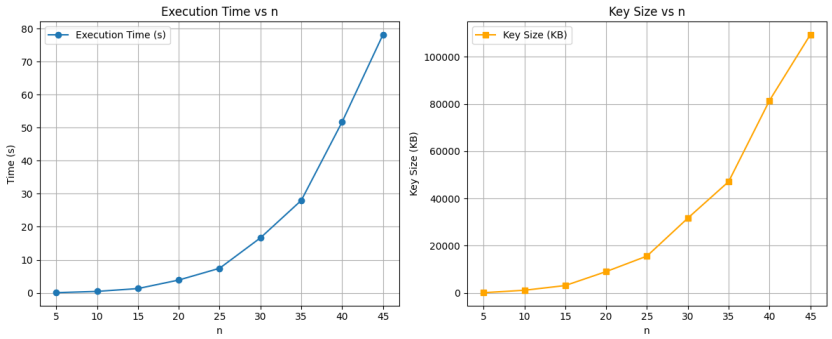


Figure 10: Execution time and trapdoor size over n

9.5 Gadged-based trapdoor sampling

The experimental measurements also imply that gadget-based trapdoors are far more efficient traditional trapdoors. We have succesffuly adapted the covariance sampling to fix the ***skewed*** distribution $\mathbf{y} = \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}$. Given the trapdoor $\mathbf{T} = \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix}$, we sample the perturbation using the following subroutine:

Algorithm 7 Generate Perturbation Vector

Input: Trapdoor matrix $\mathbf{T} \in \{0,1\}^{n \times m}$
Output: Perturbation vector $\mathbf{p} \in \mathbb{Z}^n$
Compute $\Sigma_{\mathbf{T}} \leftarrow \mathbf{T} \times \mathbf{T}^T$
Set $s \leftarrow 2 \times 2s_1(\Sigma_{\mathbf{T}})$ to be sufficently large
Define $\Sigma_p \leftarrow (s^2) \cdot \mathbf{I}_n - 2 \cdot \Sigma_T$
Compute $\mathbf{L} \leftarrow \text{Cholesky}(\Sigma_p)$
Sample $\mathbf{k} \sim \mathcal{N}(0, \mathbf{I}_n)$
Compute $\mathbf{p} \leftarrow \text{round}(\mathbf{L} \times \mathbf{k})$
Return: $\mathbf{p}$

The perturbation algorithm can be pre-processed, as we don't need to compute everything from scratch for new samples except for the fresh Gaussian vector $\mathbf{k}$. However, we include it in the key extraction process, as we must also consider the initial case of key extraction. The key generation of gadget-based trapdoors is exceptionally faster compared to the Alwen-Peikert trapdoor construction. On the other hand, they have approximately the same key size when the security parameter is $n = 150$ in gadget-based trapdoors, whereas it is $n = 45$ for the Alwen-Peikert trapdoor.

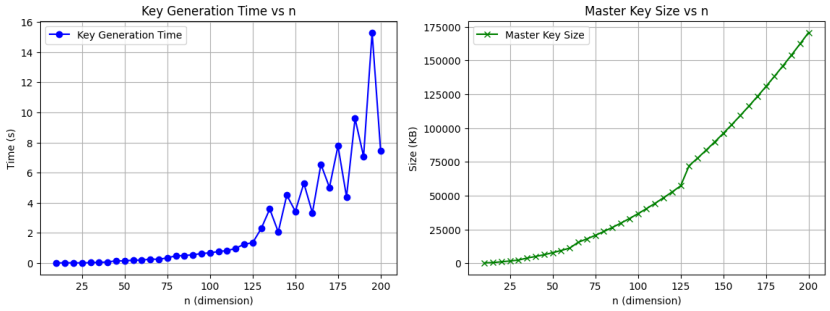


Figure 11: Benchmark of master key generation over n

As mentioned earlier, the time-consuming part of key extraction is the pre-processing of the perturbation algorithm. However, it still includes the randomized preimage sampling over the gadget matrix, which can be improved by parallelizing the algorithm in 3. We see that the size of secret keys has a linear relationship with the security size.

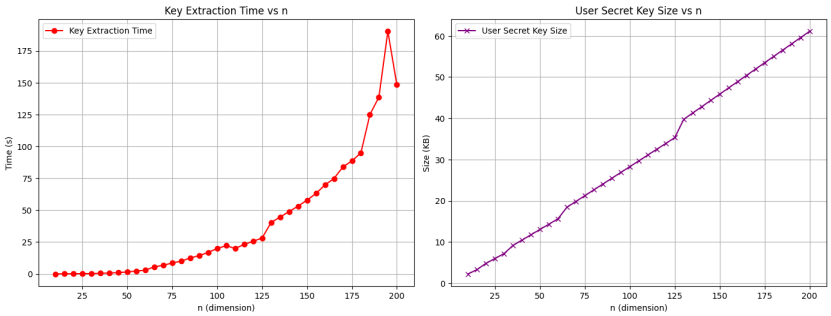


Figure 12: Benchmark of secret key extraction over n

Finally, we can say that the sizes of public parameters, key sizes, etc. are too large for contemporary cryptography standarts. There are existing works on using lattices with different ring settings for achieving faster schemes, while using smaller keys.

10 CONCLUSIONS

To conclude, we investigated how identity-based encryption scheme could solve the problem of finding the correct key and eleminate the need for digital certificates. We saw that via a trusted key generation system, it is possible to use the identities of users to serve as a valid public key. Without key exchange, it is possible for two parties to communicate effectively. In adddition, the key revocation is much simpler compared to traditional public key approaches, by extending the users identity by a specific time extension without the need for certificates for new public keys. Especially, we gave concrete uses cases for its effectiveness on situations with frequent revocation and delegation of keys. However, we have not explored the constructions for Hierarchical Identity-Based Encryption (HIBE), which offer several advantages over traditional IBE and could be a focus for future work.

Furthermore, we have conducted a deep study on foundations of lattice-based cryptography, gaining a significant knowledge on using lattices as a tool for secure communication. Lattice-based assumptions seem to be secure even against powerful quantum computers, as there is no polynomial time classical or quantum algorithm which can solve them efficiently. Operations over lattices can be implemented with matrix and vectors, which are highly effective and parallelizable especially with the new architectures such as GPUs. More importantly, some problems related to lattices enjoy from the average-case hardness, in which solving any random instance of such hard problems is no easier than solving another related worst-case hard problem. Aside from the thesis, we have also investigated the worst-case to average-case reductions for proving the security of problems including SIS and LWE problems. The average-case property of lattices make them a good candidate for cryptographic constructions while generating hard instances of them is easy. It is worth mention that there is a significant number of open problems in the field and and more research is needed to ensure the security of lattice-based cryptographic schemes in quantum contexts.

We observed that using the traditional public key over lattices, we can fix it to work of identity-based encryption via dual-Regev scheme, under LWE assumption. Additionally, it is also possible to generate hard instances of SIS problem with secret short basis enabling for richer constructions. The good bases of hard random lattices can serve as a secret information for trapdoor functions. We investigated two main families of lattice trapdoors, the full-rank bases and gadget-based trapdoors. The use of randomized preimage sampling algorithms is necessary to not to reveal our secret trapdoors. We emphasize the dominant advantages of gadget based trapdoors over traditional trapdoors, both experimentally and theoretically. The gadged-based trapdoors are easier to undertsand and implement, while providing the same level of security. Using two different trapdoor construction, we built trapdoor preimage sampleable functions, in which the preimages have a desired Gaussian like distribution. Our experiments show that the "quality" and dimension of the gadget-based trapdoor sampling is much better in constant time factors.

In practical evaluation, we examined the challenges of building a working implementation of such advanced cryptographic schemes. We experimented different ways of building of random oracles to map user identities to uniformly random vectors over integers. We built the trapdoor functions for both cases, mostly from scratch, as existing software does not support the specific needs of highly structured cryptographic tools. We observed that the choices of correct parameters and distributions is crucial as small mistakes in practice can lead to security leaks and may cause schemes to fail.

For future work, we aim to improve our implementation to a scalable IBE system for real world uses. Espeically, we can paralellize many algorithms for gadget-based trapdoor sampling for improving the efficency of the implementation. Also, we aim to use different set of lattices over different algebraic structures for faster execution and smaller public parameters, key sizes, and ciphertexts. We mention the upcoming challenges of quantum computing, and hence highlight the necessity of further theoretical research in lattice-based cryptography. Unfortunately, the large key sizes are bottleneck of this paradigm, and we should do more research on improving schemes on this regard while keeping the same security levels.

REFERENCES

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h) ibe in the standard model. In ***Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29***, pages 553–572. Springer, 2010.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In ***Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30***, pages 98–115. Springer, 2010.

[3] Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. ***Manuscript, July***, 3, 2009.

[4] Miklós Ajtai. Generating hard instances of lattice problems. In ***Proceedings of the twenty-eighth annual ACM symposium on Theory of computing***, pages 99–108, 1996.

[5] Miklós Ajtai. Generating hard instances of the short basis problem. In ***Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings***, pages 1–9. Springer, 2002.

[6] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. ***Theory of Computing Systems***, 48:535–553, 2011.

[7] László Babai. On lovász'lattice reduction and the nearest lattice point problem. ***Combinatorica***, 6:1–13, 1986.

[8] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In ***Proceedings of the 1st ACM Conference on Computer and Communications Security***, pages 62–73, 1993.

[9] Pauline Bert, Gautier Eberhart, Lucas Prabel, Adeline Roux-Langlois, and Mohamed Sabt. Implementation of lattice trapdoors on modules and applications. In ***Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings 12***, pages 195–214. Springer, 2021.

[10] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In ***Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23***, pages 223–238. Springer, 2004.

[11] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In ***Annual International Cryptology Conference***, pages 443–459. Springer, 2004.

[12] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In ***Annual international conference on the theory and applications of cryptographic techniques***, pages 440–456. Springer, 2005.

[13] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In ***Annual international cryptology conference***, pages 213–229. Springer, 2001.

[14] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption-without pairings. In ***48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)***, pages 647–657. IEEE, 2007.

[15] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. ***Journal of cryptology***, 25:601–639, 2012.

[16] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In ***Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001 Proceedings 8***, pages 360–363. Springer, 2001.

[17] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. ***IEEE Transactions on Information Theory***, 22(6):644–654, 1976.

[18] Nico Döttling and Sanjam Garg. Identity-based encryption from the diffie-hellman assumption. In ***Annual international cryptology conference***, pages 537–569. Springer, 2017.

[19] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over ntru lattices. In ***Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7-11, 2014, Proceedings, Part II 20***, pages 22–41. Springer, 2014.

[20] Morris J Dworkin et al. Sha-3 standard: Permutation-based hash and extendable-output functions. 2015.

[21] Nicholas Genise and Daniele Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In ***Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37***, pages 174–203. Springer, 2018.

[22] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In ***Theory of Cryptography Conference***, pages 437–456. Springer, 2009.

[23] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In ***Proceedings of the fortieth annual ACM symposium on Theory of computing***, pages 197–206, 2008.

[24] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In ***Advances in Cryptology—ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security Queenstown, New Zealand, December 1–5, 2002 Proceedings 8***, pages 548–566. Springer, 2002.

[25] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In ***Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17***, pages 112–131. Springer, 1997.

[26] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. ***SIAM Journal on Computing***, 28(4):1364–1396, 1999.

[27] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In ***International conference on the theory and applications of cryptographic techniques***, pages 466–481. Springer, 2002.

[28] Weidan Ji, Zhedong Wang, Haoxiang Jin, Qi Wang, Geng Wang, and Dawu Gu. Identity-based encryption from lattices with more compactness in the standard model. ***Cryptology ePrint Archive***, 2024.

[29] Sarah McCarthy, Neil Smyth, and Elizabeth O'Sullivan. A practical implementation of identity-based encryption over ntru lattices. In ***Cryptography and Coding: 16th IMA International Conference, IMACC 2017, Oxford, UK, December 12-14, 2017, Proceedings 16***, pages 227–246. Springer, 2017.

[30] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In ***Annual International Conference on the Theory and Applications of Cryptographic Techniques***, pages 700–718. Springer, 2012.

[31] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. ***SIAM journal on computing***, 37(1):267–302, 2007.

[32] Phong Q Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures. In ***Annual international conference on the theory and applications of cryptographic techniques***, pages 271–288. Springer, 2006.

[33] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In ***Annual cryptology conference***, pages 191–208. Springer, 2010.

[34] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In ***Annual Cryptology Conference***, pages 80–97. Springer, 2010.

[35] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. ***Journal of the ACM (JACM)***, 56(6):1–40, 2009.

[36] Adi Shamir. Identity-based cryptosystems and signature schemes. In ***Advances in Cryptology—CRYPTO '84***, volume 196 of ***Lecture Notes in Computer Science***, pages 47–53. Springer, 1984.

[37] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. ***SIAM review***, 41(2):303–332, 1999.

[38] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In ***Annual international cryptology conference***, pages 619–636. Springer, 2009.