IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/46SP/2025/4629125/
Vol. 2025 (37), Article ID 4629125

Hierarchical Identity-Based Encryption over q-Ary Lattices in the Random Oracle Model*

Weronika SUMERA and Mariusz JURKIEWICZ

Military University of Technology, 2 Gen. S. Kaliski St., Warsaw, Poland

Correspondence should be addressed to: Weronika SUMERA, w.sumek01@gmail.com

* Presented at the 46th IBIMA International Conference, 26-27 November 2025, Ronda, Spain

Abstract

This paper investigates a Hierarchical Identity Based Encryption ($HIBE$) scheme, with emphasis on the construction of D. Cash, D. Hofheinz, E. Kiltz, and Ch. Peikert. The scheme is founded on q-ary lattices and its security relies on the hardness of the Learning With Errors (LWE) problem, which is regarded as resistant to all known quantum algorithms. We propose a modified variant of the original design and provide a formal security analysis in the random oracle model, confirming that the modification preserves its theoretical guarantees. The central contribution of the study, however, lies in the implementation. We evaluate the scheme with respect to correctness, execution time, and computational overhead, offering a concrete assessment of its efficiency and scalability. The results show that the modified scheme combines post-quantum security with practical performance, making it a compelling candidate for applications that require hierarchical key management in a quantum aware environment.

Keywords: Post-quantum cryptography, q-ary lattice, HIBE

Introduction

In the present era of rapid advances in computational technologies, and in particular the accelerating progress of quantum computing, the demand for cryptographic systems that remain secure under these new paradigms has become urgent. Quantum algorithms such as Shor's algorithm demonstrate that many of the mathematical problems on which classical public key cryptography is built, including integer factorization and discrete logarithms in both finite fields and elliptic curve groups, could be solved in polynomial time given sufficiently powerful quantum devices. This realization places at risk the foundations of the cryptographic infrastructure that currently secures communication, authentication, and digital commerce.

To mitigate these threats, the research community has turned its attention to the design of post-quantum cryptography, that is, schemes whose security is maintained even against adversaries equipped with quantum computation. Among the candidate families of problems, lattice-based cryptography has emerged as one of the most promising. Hard lattice problems, and in particular the Learning with Errors (LWE) problem, have thus far resisted both classical and quantum algorithmic advances. At the same time, lattice techniques permit the construction of versatile primitives with strong security guarantees and practical efficiency. The use of q-ary lattices further enriches this framework by providing additional structural properties that facilitate flexible key management and scalable system design.

Within the landscape of advanced cryptographic primitives, Identity Based Encryption (IBE), introduced by D. Boneh and X. Boyen (2004), represents a significant conceptual shift. By using identifiers such as email addresses, national identification numbers, or usernames as public keys, IBE eliminates the need for certificates and reduces

Cite this Article as: Weronika SUMERA and Mariusz JURKIEWICZ, Vol. 2025 (37) "Hierarchical Identity-Based Encryption over q-Ary Lattices in the Random Oracle Model " Communications of International Proceedings, Vol. 2025 (37), Article ID 4629125, https://doi.org/10.5171/2025.4629125

reliance on traditional public key infrastructure (PKI). This simplification of key management is particularly attractive in large distributed systems.

Hierarchical Identity Based Encryption ($HIBE$) extends the IBE framework by introducing hierarchical delegation of cryptographic authority. In such systems, higher-level entities can derive secret keys for subordinate identities, enabling fine-grained and distributed access control. The design of $HIBE$ schemes that are both secure and efficient in the post-quantum setting is crucial for modern applications ranging from secure cloud storage to identity management in large-scale networks. Lattice-based $HIBE$ constructions are especially compelling, since they combine the functional advantages of hierarchical delegation with conjectured quantum resistance.

The goal of this work is to analyze and implement a $HIBE$ scheme based on q-ary lattices, with particular emphasis on the construction of D. Cash, D. Hofheinz, E. Kiltz, and Ch. Peikert (2012). We introduce a modified version of their original scheme and provide a security analysis in the random oracle model, showing that the essential guarantees are preserved in the modified design. The principal contribution of this paper lies in the implementation study. We evaluate the correctness of the scheme and analyze its concrete performance in terms of execution time, computational cost, and scalability. Our findings illustrate both the strengths and the limitations of lattice-based $HIBE$ in practice, thereby contributing to the broader effort to bridge the gap between theoretical post-quantum constructions and efficient real-world deployment.

Preliminaries

Lattices

Let $n, m \in Z_{>0}$ with $n \leq m$. Consider the set $B = \{v_1, \dots, v_n\} \subset R^m$ consisting of linearly independent vectors. A lattice in the Euclidean space R^m is defined as the set of all integer linear combinations of the vectors in B. Because the vectors of B are linearly independent, this set spans an n-dimensional subspace of R^m that contains the lattice L. The set B is called a basis of the lattice, the integer n is the dimension of the lattice, and m is its rank. When $n = m$, the lattice is called a full rank lattice. Writing the vectors of B as the columns of a matrix, the lattice can be expressed compactly as

$$L = L(B) = \{Ba \mid a \in Z^n\}.$$

A fundamental region of a lattice is a domain of the ambient space that contains exactly one representative of each coset modulo the lattice. For a lattice $L \subset R^m$ of rank n with basis $B = \{v_1, \dots, v_n\}$, one standard choice is the centered parallelepiped $F(B) = \{Bt \mid t \in [-1/2, 1/2)^n\}$. Another equally valid choice is the half-open parallelepiped $P(B) = \{Bt \mid t \in [0,1)^n\}$. Although $F(B)$ and $P(B)$ are not identical as sets, both serve as fundamental regions in the sense that their translates under L partition the entire subspace spanned by B without overlap. The particular choice of region is usually determined by analytic or algorithmic convenience.

A special and particularly important family of lattices arises from modular linear algebra. A q-ary lattice is a lattice that arises naturally from modular linear algebra. One standard example is obtained from the kernel of a matrix $A \in Z_q^{n \times m}$. In this case we define

$$\Lambda_q(A) = \{x \in Z^m \mid Ax \equiv 0 \ (mod\ q)\}.$$

This lattice consists of all integer vectors that are mapped to zero under the action of A modulo q. In this framework, the matrix A is referred to as the parity check matrix that defines the lattice. It is important to note, however, that A is not a basis matrix of $\Lambda_q(A)$ but rather specifies the modular constraints that determine it.

Lattices of the q-ary type play a central role in modern lattice based cryptography. They provide the structural foundation for the Learning With Errors (LWE) problem, whose presumed hardness against both classical and quantum algorithms has established it as one of the most reliable assumptions for post quantum security. The algebraic structure and geometric complexity of q-ary lattices support the design of expressive primitives, including identity based and hierarchical identity based encryption. These properties are essential for the present work, in which a modified $HIBE$ scheme is studied. The main contribution lies in an implementation oriented analysis that evaluates correctness, execution time, and computational efficiency, thereby connecting theoretical post quantum security with the practical demands of real world cryptographic deployment.

The definitions introduced in this subsection establish the mathematical foundation for the remainder of this work. Our study builds directly on these lattice concepts in order to analyze a modified $HIBE$ scheme. Beyond theoretical considerations, the primary contribution of this paper lies in an implementation oriented evaluation.

By examining efficiency, execution time, and computational cost in practice, we seek to clarify the extent to which lattice based hierarchical encryption can meet the dual challenge of strong post quantum security and real world deployability.

Discrete Gaussian

For $\mu \in R$ and $\sigma \in R_{>0}$, consider the function

$$\rho_{\sigma,\mu}(x) = \frac{1}{S_{\sigma,\mu}} \hat{\rho}_{\sigma,\mu}(x), \quad x \in Z.$$

Let $E_{\sigma,\mu}$ be a discrete random variable such that, for every $x \in Z$,

$$Pr[E_{\sigma,\mu} = x] = \rho_{\sigma,\mu}(x).$$

The probability distribution $E_{\sigma,\mu}$ is called the discrete Gaussian distribution and is denoted by $D_{\sigma,\mu}$. As in the continuous case, the parameter μ represents the center of the distribution, while σ controls the spread. More precisely, μ determines the expected value, and σ provides a discrete analogue of the standard deviation, shaping how rapidly the probability mass decays as one moves away from μ. The discrete Gaussian distribution plays a central role in lattice based cryptography, where it is used both for sampling and for ensuring the statistical properties required by reduction proofs.

When $\mu = 0$, the distribution is referred to as the centered discrete Gaussian distribution with parameter σ. It is denoted by D_σ. In this case the Gaussian function simplifies to the form:

$$\rho_\sigma(x) = exp\left(-\frac{x^2}{2\sigma^2}\right),$$

and the normalization factor S_σ is given by

$$S_\sigma = S_{\sigma,0} = \sum_{k=-\infty}^{\infty} exp\left(-\frac{k^2}{2\sigma^2}\right) = 1 + 2\sum_{k=1}^{\infty} exp\left(-\frac{k^2}{2\sigma^2}\right).$$

Consequently, the discrete random variable E_σ is said to follow D_σ if

$$Pr[E_\sigma = x] = \frac{1}{S_\sigma} exp\left(-\frac{x^2}{2\sigma^2}\right).$$

To analyze and approximate the behavior of the discrete Gaussian distribution, one makes use of the discrete Gaussian tail bound (Banaszczyk, 1983):

$$Pr_{x \leftarrow D_{L,\sigma}}\left[\| x \|_2 > c\sigma\sqrt{n}\right] < (c\sqrt{2\pi e}\, e^{-\pi c^2})^n,$$

where $L \subset Z^n$ is a lattice of rank n, $\sigma > 0$, and $c > 1/\sqrt{2\pi}$. It is important to note that this inequality holds for every admissible value of c, thereby offering a family of tail estimates that can be tuned to the application at hand. Such bounds are of fundamental importance in lattice based cryptography, since they guarantee that discrete Gaussian samples concentrate in a predictable region, a property that underpins both theoretical security reductions and the efficiency of practical implementations.

To generate samples from the discrete Gaussian distribution, one typically relies on an inversion sampling technique based on the cumulative distribution function (CDF). The idea is straightforward: draw a number u uniformly at random from the interval $(0,1]$, and then determine the smallest integer $x \in Z$ such that the CDF of the discrete Gaussian satisfies $F(x) \geq u$. In other words, one computes the inverse of the CDF at the point u, which ensures that the resulting x is distributed according to the desired law. Formally, the inverse mapping is given by

$$F^{-1}(u) = inf\{x \in Z \mid F(x) \geq u,\ u \in (0,1)\}.$$

A simple way to realize this procedure is to enumerate the probability masses ρ_x of the distribution in increasing order of x, while maintaining the running sum of their cumulative values. Once the cumulative sum exceeds the sampled threshold u, the corresponding value of x is returned. This sequential inversion approach is conceptually simple and ensures correctness, though its efficiency depends on the rate at which the distribution's tails decay. In practice, more advanced techniques, such as rejection sampling or precomputation methods, are often employed to accelerate Gaussian sampling, especially in lattice based cryptography where large numbers of samples are required.

Learning With Errors Problem

In the Learning With Errors (LWE) problem, one is given a matrix $A \in Z_q^{n \times m}$, an unknown secret vector $s \in Z_q^m$, and an error vector $e \in Z^n$. The samples are generated in the form

$$y \equiv As + e \ (mod\ q).$$

The error vector e introduces random noise of small norm, typically sampled from a discrete Gaussian distribution. This perturbation is precisely what makes the problem computationally difficult: although the relation between y, A, and s is linear, the injected noise obscures the hidden structure to the extent that efficient recovery of s is conjectured to be infeasible for both classical and quantum adversaries.

Computational version. In the computational formulation, the adversary is provided with many independent samples (A_i, y_i) of the form

$$y_i \equiv A_i s + e_i \ (mod\ q),$$

and the goal is to recover the secret vector s. This search problem generalizes the classical task of solving noisy systems of modular linear equations and is believed to be intractable under standard parameter regimes.

Decisional version. In the decisional formulation, the adversary must distinguish whether a given pair (A, y) was generated according to the LWE distribution, or whether y was drawn uniformly at random from Z_q^n independently of A. This variant captures the hardness of distinguishing structured noisy samples from purely random data, and it is the version most often employed in the design of cryptographic protocols.

Connection to q-ary lattices. The LWE problem also admits a geometric interpretation. A q-ary lattice is defined as the set of all vectors $x \in Z^m$ such that $Ax \equiv 0 \ (mod\ q)$. From this perspective, each LWE sample can be seen as a perturbation of the linear relations defining such a lattice. The task of solving LWE then reduces to identifying the lattice point closest to a noisy target vector, which places it in the family of approximate lattice problems. This geometric viewpoint highlights why LWE serves as a natural bridge between abstract hardness assumptions and concrete lattice constructions that underpin post quantum cryptography.

The Notion of Hierarchical Identity-Based Encryption

Hierarchical Identity-Based Encryption

The scheme studied in this work is a hierarchical identity-based encryption ($HIBE$) system, designed to enable secure communication within a hierarchy of users whose identities determine their cryptographic credentials. In such a system, a user's identity string functions directly as the public key, which eliminates the need for a traditional public key infrastructure and simplifies key management across large, distributed environments.

Formally, a $HIBE$ scheme is defined by the following algorithms:

Setup: This PT algorithm generates the global system parameters together with the master key pair, which is held by the root authority of the hierarchy.

Key Derivation (KeyDerivation): Given the master secret key and a user's identity, this PPT algorithm produces the private key corresponding to that identity. More generally, a user who already possesses a private key for some identity at a higher level in the hierarchy can use this procedure to derive private keys for subordinate identities, enabling controlled delegation.

Encryption (Enc): This PPT algorithm takes as input a plaintext message and the public identity of the intended recipient, and outputs a ciphertext that can only be decrypted using the private key associated with that identity.

Decryption (Dec): This DPT algorithm takes as input a ciphertext and the corresponding private key, and recovers the original plaintext message.

These components together realize a cryptographic system that supports both confidentiality and hierarchical delegation. The ability to derive subordinate keys is particularly well suited for applications such as organizational access control, secure cloud storage, and distributed identity management, where authority naturally follows a layered structure.

Security model

The fundamental security notion considered for $HIBE$ schemes is semantic security under chosen ciphertext and chosen identity attacks, denoted by $IND - ID - CCA$ (*Indistinguishability under Identity Chosen Ciphertext Attack*). This notion extends the standard CCA security model, as introduced in Section II.3.3, by incorporating the additional dimension of identity based encryption.

Formally, the security model is expressed as a game between a challenger and an adversary Adv, where the adversary may issue decryption queries under adaptively chosen identities and challenge messages. The experiment defining this notion is denoted by $Exp_{HIBE}^{IND-ID-CCA}$. The adversary's advantage in this experiment quantifies the extent to which it can distinguish encryptions of chosen messages under chosen identities, despite access to decryption oracles. Achieving negligible advantage in this game is regarded as the baseline requirement for a $HIBE$ scheme to be considered secure in practice.

For a scheme $HIBE = (Setup, KeyDerivation, Enc, Dec)$ and an adversary A, the security experiment is defined as a game that unfolds in the following phases:

$Exp_{HIBE}^{IND-ID-CCA}(1^n)$:

- The challenger runs $Setup$. The resulting public system parameters are given to A, while the master secret key msk remains hidden.
- **Phase 1.** The adversary A may adaptively issue a polynomial number of queries of two types:
    - *Private key extraction:* In the i-th query, A provides an identity $id_{I(i)}$. The algorithm $KeyDerivation$ is executed on $id_{I(i)}$, producing the corresponding private key $pk_{id_{I(i)}}$, which is then returned to A.
    - *Decryption query:* In the j-th query, A submits an identity $id_{I(j)}$ together with a ciphertext c_j. The algorithm $KeyDerivation$ is first invoked on $id_{I(j)}$ to derive the private key $pk_{id_{I(j)}}$. Using this key, the decryption algorithm Dec is applied to c_j, producing $\hat{m}_j$, which is returned to A.
- **Challenge.** At the end of Phase 1, A outputs a target identity id^* together with two challenge messages M_0, M_1. The adversary is not permitted to choose id^* or any of its ancestors if a corresponding key extraction query was previously made. The challenger selects a random bit $b \in \{0,1\}$ and runs Enc to compute the ciphertext c of M_b under identity id^*. The ciphertext c, called the *challenge*, is given to A.
- **Phase 2.** The adversary may continue to issue polynomially many private key extraction and decryption queries, subject to the restriction that they cannot involve the target identity id^*, any ancestor of id^*, or the challenge ciphertext c. The semantics of the queries are identical to those in Phase 1.
- **Guess.** Finally, A outputs a bit $\hat{b} \in \{0,1\}$. The adversary is said to win the game if $\hat{b} = b$.

The advantage of an adversary A in the game defined above is the following function of the security parameter:

$$Adv_{HIBE}^{IND-ID-CCA}(n) = |Pr[Exp_{HIBE}^{IND-ID-CCA}(1^n) = 1] - 1/2|.$$

As can be seen, the adversary's advantage is the probability of achieving success in the presented experiment that differs significantly from mere guessing (which has probability $1/2$).

Definition 1. *A scheme HIBE is said to be secure in the sense of* $IND - ID - CCA$ *if for every efficient adversary A its advantage is bounded by a negligible function:*

$$Adv_{HIBE}^{IND-ID-CCA}(n) \leq negl(n).$$

Construction of Hierarchical Identity-Based Encryption

The construction of our proposed $HIBE$ scheme is a modification of the work of D. Cash, D. Hofheinz, E. Kiltz, and Ch. Peikert (2012). To introduce the scheme in a clear manner, we begin with a simple illustrative example involving three hierarchical levels.

At the top of the hierarchy is the root authority. At this level, the master key pair is generated, consisting of a master public key and a master secret key. These keys serve as the foundation for the entire system and enable the generation of user keys for the levels below.

The first level contains users who each receive a private key derived from the master secret key. Every user at this level is associated with a unique identifier, denoted by $id_0, \dots, id_z$, where z indexes the set of first-level identities.

The hierarchy extends naturally to deeper levels, in which users are associated with identifiers that reflect their lineage. Each identity at level l (with $l \geq 1$) consists of l components. For instance, identities such as $id_{0,0}$ or $id_{0,1,0}$ indicate users that descend from higher-level identities.

The process of key derivation mirrors this hierarchical structure. From the master keys at the root, private keys are generated for first-level users. For users at deeper levels, private keys are derived recursively: a user at level l obtains a private key using the private key of its immediate predecessor at level $l - 1$. In this setting, a predecessor is defined as the parent identity from which the current identity directly descends. For example, the users identified by $id_{0,0}$ and $id_{0,1}$ share the parent id_0. Similarly, the users $id_{1,0,0}, id_{1,0,1}, id_{1,0,2}$ all descend from the parent $id_{1,0}$.

This hierarchical structure captures the essential idea of $HIBE$: authority is concentrated at the root, but it can be progressively delegated to lower levels of the system. The resulting scheme supports fine-grained access control and efficient key management in environments where trust relationships follow a layered organizational model.

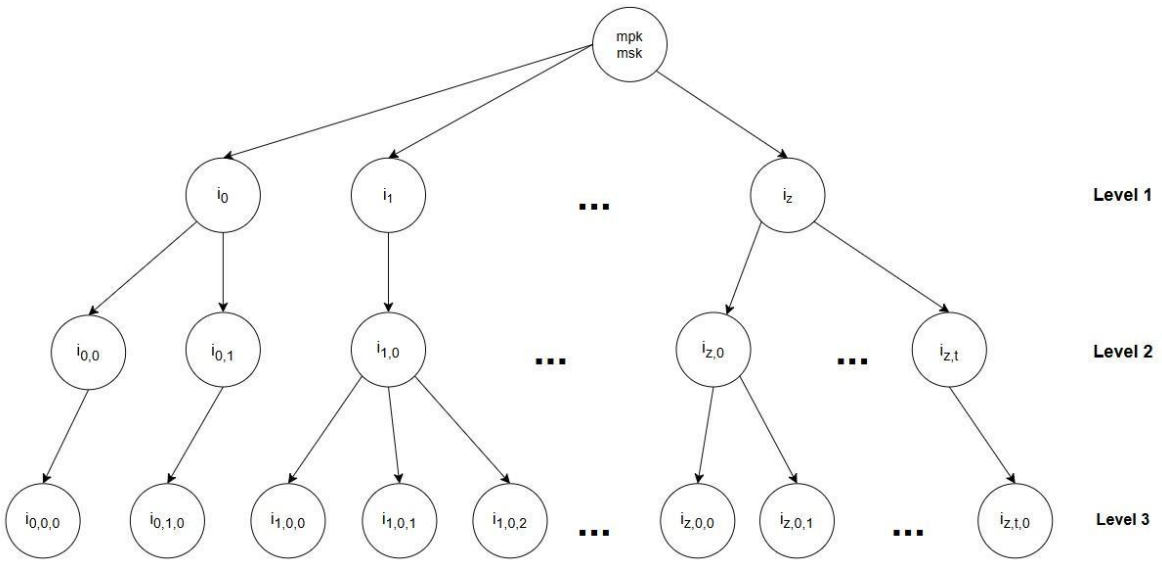

Fig 1. Visualization of an example scheme

Setup Algorithm

The $Setup$ algorithm is a probabilistic polynomial-time (PPT) algorithm used for initializing the scheme. Its purpose is to generate the master parameters of the scheme: the master secret key (msk) and the master public key (mpk).

Let $Setup$ denote the maximum depth of the hierarchy (i.e., the number of levels), and let $H = (H_n)_n$ be a family of hash functions such that $H: \{0,1\}^n \to \{0,1\}^n$. The algorithm proceeds as follows:

- Using the $TrapGen$ algorithm, generate a parity-check matrix $A \in Z_q^{n \times m}$ and a corresponding short lattice basis $T_A \subset \Lambda_q(A)$.
- Sample a vector $e \leftarrow D_{Z^m,s}$ and compute $y = Ae \in Z_q^n$.
- Select hash functions $H_1, \ldots, H_d \leftarrow H_n$.

As a result, the keys are defined as follows:

- master public key:

$$mpk = (A, y),$$

- master secret key:

$$msk = (mpk, T_A).$$

KeyDerivation Algorithm

The $KeyDerivation$ algorithm allows the generation of a private key for a user with identity id located at level $l \leq d$ in the hierarchy. The output of the algorithm is a set of parameters enabling the user to decrypt messages.

For a given identity id, a matrix is sampled as follows:

$$A' \leftarrow^{\$} Z_q^{n \times m}.$$

Next, the matrix and vector are defined as:

$$A_{id} = A||A', \quad y_{id} = H_l(id).$$

where $||$ denotes concatenation.

Using the appropriate parameters, a short basis $T_{A_{id}}$ and a vector e_{id} are generated, which constitute the user's private key:

$$T_{A_{id}} \leftarrow RandBasis(ExtBasis(A_{id}, T_A), s_l),$$

$$e_{id} \leftarrow SampleISIS(A_{id}, T_A, y_{id}, s_l),$$

such that:

$$y_{id} = A_{id} e_{id}.$$

Consequently, the user's public and private keys are defined as:

$$pk_{id} = y_{id}, \quad sk_{id} = e_{id}.$$

Thanks to the hierarchical structure of the $HIBE$ system, it is possible to derive the private key of a user at level l from the private key of its immediate predecessor at level $l-1$. If a parent user possesses the identity $id_{|l-1}$ and the corresponding private key, one can use its short basis $T_{A_{|l-1}}$ to derive a new private key for the user id.

For the given identity id, a matrix is sampled:

$$A' \leftarrow^{\$} Z_q^{n \times m}.$$

Then, the concatenated matrix is defined as:

$$A_{id} = A||\ldots||A' = A_{id_{|l-1}}||A'.$$

Based on this, the short basis $T_{A_{id}}$ and the vector e_{id} are generated using the appropriate parameters, forming the user's private key:

$$T_{A_{id}} \leftarrow RandBasis(ExtBasis(A_{id}, T_{A_{|l-1}}), s_l),$$

$$e_{id} \leftarrow SampleISIS(A_{id}, T_{A_{id}}, y_{id}, s_l).$$

Enc Algorithm

The Encryption (Enc) algorithm enables the encryption of a single-bit message $m \in \{0,1\}$ for a user with identity id, using the master public key. The resulting ciphertext consists of two components, b and p, which are computed as follows:

- Sample a vector $s \leftarrow^{\$} Z_q^n$.
- Compute the product of the transposed matrix A_{id}^T with the vector s, and add noise drawn from a distribution χ. In our construction, the noise follows a discrete Gaussian distribution. This yields the vector b:

$$b = A_{id}^T \cdot s + error_1.$$

- Similarly, compute the value p as:

$$p = (y_{id}^T \cdot s + m \cdot q/2) + error_2.$$

Finally, the ciphertext of the message m is given by:

$$c = (b, p).$$

Dec Algorithm

The Decryption (Dec) algorithm allows a user with identity id to decrypt a ciphertext using their private key $sk_{id} = (e_{id})$. The decryption process consists of computing an intermediate value m', which is then interpreted as the original bit of the single-bit message:

$$m' = p - e_{id}^T \cdot b \ mod \ q.$$

Expanding the above expression, m' can be written as:

$$m' = y_{id}^T \cdot s + m \cdot \frac{q}{2} + error_2 - e_{id}^T \cdot A_{id}^T \cdot s - e_{id}^T \cdot A_{id}^T \cdot error_1.$$

After simplification, this reduces to:

$$m' = m \cdot \frac{q}{2} + error_2 - e_{id}^T \cdot A_{id}^T \cdot error_1.$$

To correctly interpret the result, we extend the range to the set of real numbers in $[0,2)$ by scaling:

$$\frac{2m'}{q} = \frac{2}{q}\left(m \cdot \frac{q}{2} + error_2 - e_{id}^T \cdot A_{id}^T \cdot error_1\right).$$

Simplifying further, we obtain:

$$\frac{2m'}{q} = m + \frac{2}{q}(error_2 - e_{id}^T \cdot A_{id}^T \cdot error_1).$$

Assuming that:

$$|error_2 - e_{id}^T \cdot A_{id}^T \cdot error_1| < \frac{q}{4},$$

it follows that:

$$\left|\frac{2}{q}\left(error_2 - e_{id}^T \cdot A_{id}^T \cdot error_1\right)\right| < \frac{1}{2}.$$

Finally, the scaled value $\frac{2m'}{q}$ is interpreted as follows:

- if $\frac{2m'}{q}$ is closer 0, then the decrypted message i $sm = 0$,
- if $\frac{2m'}{q}$ is closer 2, then the decrypted message is $m = 1$.

Thanks to appropriate bounds on the errors, reliable reconstruction of the original message bit is guaranteed.

Auxiliary Algorithms

In the described scheme, auxiliary algorithms are used, such as $TrapGen$, $RandBasis$, $SampleISIS$ and $ExtBasis$, which will be described below.

TrapGen Algorithm

The $TrapGen$ algorithm is a probabilistic polynomial-time (PPT) algorithm. It is used to generate a parity-check matrix $A \in Z_q^{n\times m}$ and the so-called trapdoor $T_A \in Z^{m\times m}$, which is a short basis of the q-ary lattice $\Lambda^{\perp}(A)$. Let $C > 1$ be a constant. There exists a $TrapGen$ algorithm that, on input parameters n, $m \geq Cnlgq$ and q, outputs $A \in Z_q^{n\times m}$ and $T_A \in Z^{m\times m}$ (Cash et al, 2012) . More precisely, we have:

$$TrapGen(1^n, 1^m, q) \rightarrow (A, T_A),$$

where:

- A is a parity-check matrix,
- T_A is a short basis of the q-ary lattice $\Lambda^{\perp}(A)$,
- The condition is satisfied: $AT_A \equiv 0\ (mod\ q)$ $(\in Z_q^{n\times m})$.

In the following, one version of this algorithm will be presented in detail by Alwen (2011) . Let $\delta > 0$ be a fixed arbitrarily small constant, and let q be a modulus that is a power of a prime number. Then, there exists a probabilistic polynomial-time algorithm (PPT)) which, for any $m_1 \geq d = (1+\delta)nlog(q)$ and any uniformly random matrix $A_1 \in Z_q^{n\times m_1}$ and integers $r \geq 2$, $m_2 \geq m_1 \cdot l$, where $l = \lceil log_r q \rceil$, computes matrices U, G, R, P and C with the following properties:

- U is a unimodular matrix of dimension $m_2 \times m_2$ of the form:

  $$U = diag(T_l, \dots, T_l, I),$$

  i.e., U is a block diagonal matrix, which on the main diagonal contains m_1 matricies $T_l \in Z^{l\times l}$ and the identity matrix of dimension $m_2 - m_1 l$. The matrix T_l s an upper-triangular matrix with ones on the diagonal, elements equal to $-r$ bove the diagonal, and zeros below the diagonal.

- G is a matrix whose elements form an increasing geometric sequence. Similarly to U, it is a block matrix consisting of m_1 blocks $G(i)$, each containing l columns, and a zero matrix containing $m_2 - m_1 l$ columns. We write this as:

  $$G = [G(1) \mid \cdots \mid G(m_1) \mid 0(m_1+1) \mid \cdots \mid 0(m_2)] \in Z_q^{m_1\times m_2}.$$

  To describe the structure of the blocks $G(i)$, we assume that each matrix $G(i)$ has a column form, and its j-th column is denoted by $g_j^{(i)}$. Then the last column, i.e., $g_l^{(i)}$, corresponds to the i-th column of the matrix $HNF(\Lambda_q(A_1)) - I$ while the remaining columns are computed recursively according to the formula:

$$g_j^{(i)} = \lfloor \frac{g_{j+1}^{(i)}}{r} \rfloor, \quad dla\ j = l-1, \ldots, 1.$$

- P is a matrix of dimension $m_2 \times m_1$, constructed as a concatenation of cyclically chosen canonical vectors $e_i \in Z^{m_2}$. More precisely, the j-th column of the matrix P is of the form:

$$p_j = e_{jl} \in Z^{m_2}, \quad where\ j \in [m_1].$$

- R is a matrix of dimension $m_1 \times m_2$, whose elements are values of the random variable $X \in \{0, \pm 1\}$ with distribution:

$$Pr[X = 0] = \frac{1}{2}, \quad Pr[X = \pm 1] = \frac{1}{4}.$$

  The elements are sampled independently.

- C is a zero matrix of dimension $m_1 \times m_1$.

The parity-check matrix $A \in Z_q^{n \times (m_1 + m_2)}$ takes the form:

$$A = [A_1 \mid A_2], \quad where\ A_2 = -A_1(R + G) \in Z_q^{n \times m_2}.$$

The corresponding basis (the so-called trapdoor basis) takes the form:

$$T_A = ((G + R)U\ RP - C\ (U\ P\ ).$$

ExtBasis Algorithm

The $ExtBasis$ algorithm (Cash et al, 2012) is used to extend an existing lattice to a new lattice and to generate a short basis for it. Let $T_A \in Z^{m \times m}$ be a short basis of the q-ary lattice $\Lambda_q(A)$, where $A \in Z_q^{n \times m}$ is a matrix generating the entire space Z_q^n. Furthermore, let $A_0 \in Z_q^{n \times m_0}$ be an arbitrary matrix.

There exists a probabilistic polynomial-time algorithm (PPT), denoted as $ExtBasis$, which takes as input the matrix A and its corresponding short basis T_A, and outputs a new short basis T_A' of the q-ary lattice q-ary $\Lambda_q(A')$, where $A' = [A \mid A_0]$.

$$A' = A||A_0.$$

Formally:

$$ExtBasis(T_A, A') \rightarrow (T_A').$$

In addition, the generated short basis satisfies the following condition:

$$||T_A|| = ||T_A'||.$$

The algorithm constructs the basis T_A' in the following way:

$$T_A' = (T_A\ W\ 0\ I\ ),$$

where

- $I \in Z^{m_0 \times m_0}$ is the identity matrix
- 0 is the identity matrix
- $W \in Z^{m \times m_0}$ is a matrix that is a solution to the equation:

$$AW \equiv -A_0 \pmod q.$$

Algorithm 1: ExtBasis

Input: Matrix A, vector A_0, basis T_A

Output: Extended basis T'_A

Compute W such that $A \cdot W = -A_0 \pmod{q}$;

Generate matrices I and 0;

Construct extended basis T'_A:

$$T'_A = \begin{bmatrix} T_A & W \\ 0 & I \end{bmatrix}$$

return T'_A;

Fig 2. Algorithm ExtBasis

SampleISIS Algorithm

The $SampleISIS$ algorithm (Jurkiewicz, 2024) occupies a central position in post quantum cryptography, particularly in constructions whose security is based on the computational hardness of the Inhomogeneous Short Integer Solution ($ISIS$) problem by Gentry (2008). The purpose of this algorithm is to efficiently generate short vectors e that satisfy the relation $Ae \equiv y \ (mod\ q)$, where $A \in Z_q^{n \times m}$ and $y \in Z_q^n$.

The $ISIS$ problem is believed to be computationally intractable in general and therefore serves as a hardness assumption underlying the security of numerous lattice based cryptographic schemes. The situation changes, however, when the matrix A is accompanied by auxiliary information in the form of a trapdoor, typically represented as a short basis of the q-ary lattice $\Lambda^{\perp}(A)$. With access to such a trapdoor, it becomes possible to efficiently solve the equation while ensuring that the output vector e remains short. This property makes trapdoor constructions indispensable for the design of practical cryptographic primitives, including identity based and hierarchical encryption schemes, as well as digital signatures and advanced functionalities in the post quantum setting.

The algorithm takes as input a matrix $A \in Z_q^{n \times m}$, T'_A a short basis of the q-ary lattice the Gaussian parameter s of the Discrete Gaussian Distribution, and a vector $y \in Z_q^n$. It then proceeds as follows:

- Select an arbitrary vector t, that satisfies the equation: $At \equiv u \ (mod\ q)$. This solution does not need to be short; its purpose is solely to guarantee the correctness of the subsequent steps of the algorithm. The existence of such a t is ensured by the fact that for every $y \in Z_q^n$, provided A is of full rank, there exists at least one solution to this equation $(mod\ q)$. Such a solution can be found using standard linear algebra techniques.
- Using the algorithm $SampleD(T_A, s, -t)$ compute the vector $v \leftarrow D_{s,-t}$.
- Finally, return the vector of the form $e = t + v$.

Algorithm 2: SampleISIS

Input: Matrix A, short basis T_A, parameter s, target vector y

Output: Vector e such that $A \cdot e = y \pmod{q}$

Find t such that $A \cdot t = y \pmod{q}$;

Sample $v \leftarrow \text{SampleD}(T_A, s, -t)$;

$e \leftarrow t + v$;

return e;

Fig 3 Algorithm SampleISIS

SampleD Algorithm

The $SampleD$ algorithm (Cash et al, 2012) is an algorithm that takes as input three parameters:

- T_A - a short basis of the q-ary lattice $\Lambda^{\perp}(A)$.
- $s > 0$ - the Gaussian parameter of the discrete Gaussian distribution,
- $\mathbf{y} \in \RR^n$ - the center of the discrete Gaussian distribution.

It outputs a vector $e \in Z^m$. which is computed as follows:

- Initialize $v_n \leftarrow 0$ and $y_n \leftarrow y$.
- Dla $i \leftarrow n,\ldots,1$ perform the following steps:
    - Compute:

$$y_i' = \frac{\langle y_i, \tilde{t}_i \rangle}{\langle \tilde{t}_i, \tilde{t}_i \rangle},$$

      where $\tilde{t}_i$ denotes the i-th vector obtained from the Gram–Schmidt orthogonalization of the basis T_A.
    - Compute:

$$s_i' = \frac{s}{||\tilde{t}_i||}.$$

    - Sample an integer from the discrete Gaussian distribution with center y_i' and parameter s_i'.

$$z_i \sim D_{s_i', y_i'}.$$

    - Update the values:
        - $y_{i-1} \leftarrow y_i - z_i t_i$
        - $v_{i-1} \leftarrow v_i + z_i t_i$
- Return the value $e = v_0$.

Algorithm 3: SampleD

Input: Basis T_A, parameter s, target vector y

Output: Vector e sampled from discrete Gaussian

$v_n \leftarrow 0$, $y_n \leftarrow y$;
for $i = n, n-1, \ldots, 1$ **do**
  $y_i' \leftarrow \frac{\langle y_i, \tilde{t}_i \rangle}{\langle t_i, t_i \rangle}$;
  $s_i' \leftarrow s / \|\tilde{t}_i\|$;
  Sample $z_i \sim D_{s_i', y_i'}$;
  $y_{i-1} \leftarrow y_i - z_i \cdot t_i$;
  $v_{i-1} \leftarrow v_i + z_i \cdot t_i$;
$e \leftarrow v_0$;
return e;

Fig 4. Algorithm SampleD

RandBasis Algorithm

The $RandBasis$ algorithm (Cash et al, 2012) is a PPT algorithm designed to generate a random, yet still short, basis of the q-ary lattice $\Lambda^{\perp}(A)$. It takes as input the parameters: T_A a short basis of the q-ary lattice $\Lambda^{\perp}(A)$, the Gaussian parameter of the discrete Gaussian distribution $s > 0$ and outputs a new short basis of this lattice according to the following procedure:

- Initialize $i \leftarrow 0$
- While $i < m$:
    - Sample a vector $v \leftarrow SampleD(T_A, s)$
    - If v is linearly independent of $v_1, \ldots, v_i$ then set $i \leftarrow i + 1$ and $v_i = v$
- Compute $HNF(T_A)$
- Return $T'_A = ToBasis(V, HNF(T_A))$

The auxiliary algorithm $ToBasis$ (Cash et al, 2012) is responsible for constructing a new basis from the set of linearly independent vectors V and the lattice structure defined by $HNF(T_A)$. It ensures that the following condition is satisfied: $||\widetilde{t_{A_i}}'|| = ||\widetilde{t_{A_i}}||$.

Algorithm 4: RandBasis

Input: Basis T_A, parameter s

Output: Randomized basis T'_A

```
i ← 0;
while i < m do
    Sample v ← SampleD(T_A, s);
    if v is linearly independent of v_1, ..., v_i then
        i ← i + 1;
        v_i ← v;
HNF ← HNF(T_A);
T'_A ← ToBasis({v_1, ..., v_m}, HNF);
return T'_A;
```

Fig 5. Algorithm RandBasis

Security Proof

Let n be the security parameter, and let $HIBE = (Setup, KeyDerivation, Enc, Dec)$ denote a hierarchical identity-based encryption scheme based on the LWE problem.

It is assumed that if there exists an efficient adversary A who breaks the security of the given scheme in the sense of $IND - ID - CCA$ with non-negligible probability, then there also exists an algorithm B that solves an instance of the LWE problem with non-negligible probability, in time comparable to the runtime of A. This intuition leads to the following formal theorem:

Theorem 1. *If the LWE problem is secure, then the HIBE scheme is $IND - ID - CCA$ secure, i.e.,* $Adv_{A,HIBE}^{IND-ID-CCA}(n) \leq Adv_B^{LWE}(n)$.

Proof. Assume that A is an adversary attacking the security of the $HIBE$ scheme, and let $\epsilon = Adv_{HIBE}^{IND-ID-CCA}(n)$ be its advantage. We construct a simulator B that uses A to solve an instance of the LWE

problem. **Initialization Phase**:
The simulator B receives the LWE problem parameters, consisting of a matrix $A \in Z_q^{n \times m}$ and a vector $y \in Z_q^n$, which is one of two equally likely forms:

- $y = A \cdot s + error$, where s is a short vector from Z_q^n, and $error$ is a small-norm error,
- a random element of Z_q^n (independent of A).

B samples hash functions $H_1, \ldots, H_d$ and then sets the master public key as $mpk = (A, y)$. B does not know the short basis T_A, so it cannot compute the master secret key msk. B sends the master public key to A.

B must simulate, up to a negligible probability, a randomly generated $HIBE$ scheme in front of A, to which a randomly generated $HIBE$ scheme in front of A has the right to make queries according to the model described in Section 3.2.

Query Phase 1:

A may issue the following types of queries:

1. Generate the private key for a user with identity id.
2. Decrypt a given ciphertext c using the private key of a user with identity id.

Response to type i queries:

1. Using the $TrapGen$ algorithm, B generates a random matrix A' and a short basis $T_{A'}$.
2. Constructs the matrix $A_{id} = A' || A$.
3. Using $T_{A'}$ and A_{id} using the algorithms $RandBasis$, $ExtBasis$ and $SampleISIS$ generates a short basis $T_{A_{id}}$ and the corresponding private key e_{id}.
4. Returns the value e_{id} to A.

Response to type ii queries:

B performs the same steps as in the private key query. Then, instead of returning the private key e_{id}, it uses the Dec algorithm to compute the decrypted message and sends the decrypted value to A.

Challenge Phase:
A chooses an identity id^* and two messages m_0 and m_1, which it sends to B, samples a bit $b \in \{0,1\}$, a vector s, and $error_1$ and $error_2$. Then B computes: $b^* = A_{id}^T \cdot s + error_1$ and $p^* = (y_{id}^T \cdot s + m \cdot q/2) + error_2$ and returns the ciphertext $c^* = (b^*, p^*)$ to A.
Query Phase 2:
The Adversary A may again issue queries, analogously to Phase 1, with certain restrictions. In particular, it may not request the private key or decryption for the challenge identity id^* or any of its ancestors in the hierarchy.
Guessing Phase:
A outputs a guess bit b'. If $b' = b$, B concludes that the vector v comes from the LWE distribution; otherwise, it concludes that the vector was random.
Thus, if adversary A can distinguish ciphertexts with non-negligible advantage ϵ, the simulator B can also distinguish, with non-negligible probability, whether the given vector comes from the LWE distribution or is random.
This implies that the ability of an adversary to break the $IND - ID - CCA$ security of $HIBE$ implies the ability to solve the LWE problem.
According to the results presented in Section 2.3, the LWE problem is computationally hard for appropriately chosen parameters and is based on the worst-case hardness of lattice problems. This hardness implies that for

any probabilistic polynomial-time algorithm, its success probability in solving an LWE instance is negligible, i.e.:

$$Adv_B^{LWE}(n) \leq negl(n).$$

Consequently, it follows that:

$$Adv_{A,HIBE}^{IND-ID-CCA}(n) \leq negl(n).$$

Implementation Results

We now present the results of our implementation. It is important to emphasize that the current implementation should be regarded as a simulation of the $HIBE$ scheme rather than a production level deployment. The primary objective is to assess the feasibility of the construction and to provide insight into its efficiency under realistic parameter choices.

To examine both correctness and performance, we conducted a series of experimental tests across several parameter sets:

- $n = 13,\ q = 8209,$
- $n = 10,\ q = 16411,$
- $n = 9,\ q = 32771.$

For these sets, both master keys and user keys for the first and second hierarchical levels were generated. Additional tests were performed for:

- $n = 21,\ q = 32771,$
- $n = 25,\ q = 32771,$

where only master keys and first-level user keys were generated. Due to hardware limitations, the full planned set of 1,000 independent trials per parameter set could not be executed. Instead, each set was tested in 70 trials, organized into blocks of 20, allowing the collection of partial results for global analysis. In each trial, a single random bit (0 or 1) was encrypted and decrypted, and the output was compared with the original to verify correctness.

For each parameter set, we recorded the following performance indicators:

- the number of messages that were successfully decrypted,
- the average running times of the encryption and decryption procedures,
- the standard deviations associated with these running times.

To determine the global average encryption time and its standard deviation, the following formal approach was adopted. Let A be a finite set, and let $A_1, A_2, \ldots, A_k$ be its disjoint subsets such that $\#A_i = n_i$ and $A = \bigcup_{i=1}^{k} A_i$ with $A_i \cap A_j = \emptyset$ for $i \neq j$. Then $\#A = \sum_{i=1}^{k} \quad n_i$. For each subset A_i, let μ_i be the mean and standard deviation σ_i. The global mean is given by:

$$\mu = \frac{1}{n} \sum_{i=1}^{k} \quad n_i \cdot \mu_i.$$

The global standard deviation is:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{k} \quad n_i(\sigma^2 + (\mu_i - \mu)^2)}.$$

Tables 1 and 2 summarize the experimental results obtained for the first and second hierarchical levels of the proposed $HIBE$ scheme. For each configuration, the average encryption and decryption times, their corresponding standard deviations, and the number of correctly decrypted messages out of 1,400 trials are reported.

Table 1: Summary of global results f or the first hierarchical level

No.	Case	Avg. Encryption Time [s]	Std. Dev. of Encryption [s]	Avg. Cecryption Time [s]	Std. Dev. of Decryption [s]	Number of Correct Decryptions (out of 1400)
1	$q = 8209, n = 13$	12,476	0,41578	0,00085	9,46E-05	720
2	$q = 16441, n = 10$	11,029	0,37320	0,00076	8,53E-05	705
3	$q = 32771, n = 9$	11,297	0,38480	0,00078	9,07E-05	764
4	$q = 32771, n = 21$	26,441	0,69112	0,00176	1,46E-04	695
5	$q = 32771, n = 25$	31,388	0,73953	0,00208	1,57E-04	696

Table 2: Summary of global results for the second hierarchical level

No.	Case	Avg. Encryption Time [s]	Std. Dev. of Encryption [s]	Avg. Decryption Time [s]	Std. Dev. of Decryption [s]	Number of Correct Decryptions (out of 1400)
1	$q = 8209, n = 13$	18,686	0,53286	0,00126	1,19E-04	702
2	$q = 16441, n = 10$	16,527	0,46734	0,00112	1,24E-04	718
3	$q = 32771, n = 9$	16,932	0,48265	0,00078	1,17E-04	662

The experimental results demonstrate that increasing the parameter n consistently results in longer encryption and decryption times. This behavior aligns with the expected computational complexity of the scheme, since larger values of n correspond to higher dimensional matrices, which in turn require a greater number of arithmetic operations. The observed trend therefore provides an empirical confirmation of the theoretical performance characteristics of the construction.

Across all experiments, the average decryption success rate varied between approximately 47% and 55%. While the system does not achieve perfect decryption accuracy under the tested conditions, this limitation arises primarily from the specific parameter values selected for the experiments. With access to greater computational resources, it would be possible to employ larger values of n and q, which are expected to enhance the reliability of decryption and bring the success rate closer to the theoretical guarantees of the scheme.

At the same time, increasing n and q inevitably raises both memory consumption and execution times, which must be carefully considered in the design and deployment of $HIBE$ systems.

A key contributor to the computational cost of the scheme is the $TrapGen$ algorithm executed during the setup phase. $TrapGen$ generates large matrices together with their short lattice bases, imposing substantial memory and computational demands. Because the size of these initial parameters directly influences the dimensions of private keys, ciphertexts, and intermediate structures, $TrapGen$ plays a decisive role in system performance. Improving the efficiency of $TrapGen$ - or replacing it with a more lightweight trapdoor generation method - could significantly reduce parameter sizes, thereby lowering computational overhead and enhancing the practical feasibility of $HIBE$ implementations.

Conclusion

The modification introduced in our construction of the $HIBE$ scheme appears to improve efficiency while preserving the theoretical guarantees of post quantum security derived from the LWE assumption. The experimental results confirm correctness under the tested parameter sets and reveal predictable trade offs between

performance and parameter growth. In particular, the increase in execution time with larger values of n and q reflects the expected computational complexity of lattice based algorithms, while the moderate decryption accuracy observed in our simulations is primarily a consequence of the limited resources and parameter sizes employed. With more powerful computational platforms and carefully tuned parameters, substantially higher correctness rates can be achieved.

These findings suggest that the proposed modification brings the $HIBE$ construction closer to practical deployment. At the same time, the results highlight the need for continued research on algorithmic improvements, especially in the areas of trapdoor generation, Gaussian sampling, and key derivation, which remain the principal bottlenecks in efficiency. Addressing these challenges is essential if lattice based hierarchical encryption is to scale effectively in real world applications.

It should also be emphasized that, alongside traditional $HIBE$ systems in which all security rests on a single master secret, alternative paradigms have emerged, such as registered identity-based encryption, where trust is distributed in more flexible ways. Nevertheless, master key dependent $HIBE$ constructions remain highly relevant. In strongly hierarchical institutions such as governmental agencies, military organizations, or large corporations-security policies often require higher level authorities to retain direct control over the cryptographic capabilities of their subordinates. In such contexts, schemes with a single root of trust are not only natural but indispensable.

In conclusion, our study shows that master key dependent $HIBE$ systems continue to offer a compelling direction for post quantum cryptography. Future work should focus on refining their efficiency, exploring parameter regimes that strike an appropriate balance between performance and security, and investigating hybrid designs that combine the strict hierarchical control of classical $HIBE$ with the flexibility of registered or distributed identity-based approaches. Advancing research along these lines is essential to ensure that hierarchical encryption remains a robust and deployable tool for safeguarding complex infrastructures in the post quantum era.

References

- Ahmad, K., Ahmad, K. & Dulhare, U., 2021. Functional encryption. *Springer.*
- Alwen, J. & Peikert, C., 2011. Generating shorter bases for hard random lattices. *Theory of Computing Systems.*
- Banaszczyk, W., 1983. Gaussian measures on locally convex topological vector spaces. *Studia Mathematica.*
- Boneh, D. & Boyen, X., 2024. Efficient selective-id secure identity-based encryption without random oracles. *International conference on the theory and applications of cryptographic techniques.*
- Boneh, D., Boyen, X. & Goh, E., 2005. Hierarchical identity based encryption with constant size ciphertext. *Annual international conference on the theory and applications of cryptographic techniques.*
- Boneh, D. & Shoup, V., 2023. *A graduate course in applied cryptography..* draft 0.6. ed. s.l.:s.n.
- Cash, D., Hofheinz, D., Kiltz, E. & Peikert, C., 2012. *Bonsai trees, or how to delegate a lattice basis.* s.l.:Journal of cryptology.
- Gentry, C., Peikert, C. & Vaikuntanathan, V., 2008. Trapdoors for hard lattices and new cryptographic constructions. *Proceedings of the fortieth annual ACM symposium on Theory of computing.*
- Goldwasser, S. & Micali, S., 1982. Probabilistic encryption. *Proceedings of the fourteenth annual ACM symposium on Theory of computing.*
- Joye, M. & Neven, G., 2009. Identity-based cryptography. *IOS press.*
- Jurkiewicz, M., 2024. Quantum-resistant forward-secure digital signature scheme based on q-ary lattices. *Journal of Telecommunications and Information Technology.*
- Katz, J. & Lindell, Y., 2014. Introduction to modern cryptography. *Chapman and hall/CRC.*
- Lidl, R. & Pilz, G., 2012. Applied abstract algebra. *Springer Science & Business Media.*
- Luther, M., 2008. Introduction to identity-based encryption. *Artech house.*
- Micciancio, D. & Goldwasser, S., 2002. Complexity of lattice problems: a crypto- graphic perspective. *Springer Science & Business Media.*
- Micciancio, D. & Peikert, C., 2012. Trapdoorsforlattices: Simpler, tighter, faster, smaller. *Annual International Conference on the Theory and Applications of Cryptographic Techniques.*
- Michel, A. & Herget, C., 2009. *Algebra and analysis for engineers and scientists.* s.l.:Springer Science & Business Media.
- Rao, K., 2017. *Numerical methods for scientists and engineers.* s.l.:PHI Learning Pvt. Ltd..