I. J. Computer Network and Information Security, **2025, 3, 89-108**
Published Online on June 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2025.03.06

Post-quantum Digital Signatures using ElGamal Approach

Maksim Iavich
Department of Information Security Faculty, Caucasus University, Tbilisi, Georgia, Al - Farabi Kazakh National University, Almaty, Kazakhstan
E-mail: miavich@cu.edu.ge
ORCID iD: https://orcid.org/0000-0002-3109-7971

Dana Amirkhanova Sairangazhykyzy*
Department of Cybersecurity, Information Processing and Storage, Satbayev University, Almaty, Kazakhstan
E-mail: amirkhanovadana2@gmail.com
ORCID iD: https://orcid.org/0009-0007-7535-2966
*Corresponding author

Received: 25 September 2024; Revised: 12 January 2025; Accepted: 14 March 2025; Published: 08 June 2025

Abstract: The paper offers a novel digital signature scheme that integrates ElGamal cryptographic principles with the Short Integer Solution (SIS) problem, specifically designed to ensure post-quantum security. As quantum computers advance and present significant risks to traditional cryptographic systems, this scheme offers an interesting alternative for securing digital signatures against potential quantum threats. The scheme uses only basic secure principles. The offered approach offers key generation, where parameters and random matrices are selected, and signature generation, which involves creating signatures based on hashed messages and matrix computations. Verification ensures the authenticity and integrity of signatures. We provide experimental evaluations detailing key generation, signature creation, and verification times across different matrix dimensions and message sizes. Key generation takes between 2.5–10.2 seconds, while signature generation ranges from 0.20 to 9.30 milliseconds and verification from 0.18 to 8.90 milliseconds, depending on message size and matrix dimension. The scheme maintains a consistent signature size of 1.7 KB, independent of message length due to a hash-and-sign strategy. These results demonstrate that the scheme balances post-quantum security with practical performance, especially in high-security contexts. A comparison with traditional ElGamal encryption reveals the trade-offs between security and efficiency. While the SIS-based scheme delivers enhanced protection against quantum threats, it also entails increased computational complexity and larger signature sizes compared to conventional schemes.
Overall, our proposed digital signature scheme stands as an excellent option for safe communications in a post-quantum world, representing a crucial step in protecting the authenticity and integrity of digital exchanges against changing technological risks. We believe that as quantum computing continues to develop, research into robust cryptographic alternatives will become increasingly important for safeguarding sensitive information across various sectors.

Index Terms: SIS, Post-quantum, CVP, SVP, DLP, PKE, Lattice.

1. Introduction

The rapid advancements in quantum computing pose a significant threat to classical cryptographic systems, particularly those based on integer factorization and discrete logarithms, which are vulnerable to Shor's algorithm. While the exact timeline for the development of large-scale, fault-tolerant quantum computers remains uncertain, the potential impact on cryptographic security is widely recognized. Organizations such as the National Institute of Standards and Technology (NIST) have already initiated efforts to standardize post-quantum cryptographic algorithms, recognizing the urgency of preparing for a post-quantum future. NIST has estimated that the transition to post-quantum cryptography should be completed within the next 10 years, emphasizing the need for proactive measures to safeguard sensitive information against future quantum threats. This timeline underscores the importance of developing and deploying quantum-resistant cryptographic solutions, such as the proposed SIS-based ElGamal digital signature scheme, to ensure long-term security in a post-quantum world. Classical digital signature schemes are increasingly challenged by the active and quick advancements in computational technologies, particularly with the rise of quantum computing. As

computational power grows; traditional cryptographic systems face new vulnerabilities [1-3]. To address these challenges, lattice-based cryptography has emerged as a promising approach. Lattices, defined as discrete subgroups in multidimensional Euclidean space, provide a strong basis because of their intrinsic complexity and resilience to assaults, including those from quantum computers.

Lattice-based cryptography depends on the difficulty of issues like the Closest Vector Problem (CVP), Learning with Errors (LWE), Shortest Vector Problem (SVP), and Short Integer Solution (SIS) [4-6]. These issues provide robust defenses against classical and quantum attacks and are renowned for their computational difficulties. Lattice-based algorithms' simplicity and security make them a desirable substitute for more conventional cryptography techniques. Conventional techniques such as the Diffie-Hellman key exchange and the RSA cryptosystem, which form the foundation of contemporary digital signatures and public key encryption. The ElGamal method is an example of public key cryptography with its focus on discrete logarithms. It must be mentioned that, as quantum computing technology progresses, these classical methods face significant threats [7,8]. To protect against these emerging threats, this paper offers a new digital signature scheme that combines ElGamal's principles with lattice-based cryptography, specifically it uses the Short Integer Solution (SIS) problem. This scheme uses fundamental security principles to ensure the high-level protection against quantum attacks. It is particularly well-suited for applications where key and signature sizes are not a primary concern. Such scenarios include high-security environments where the cost of larger keys and signatures is outweighed by the need for advanced cryptographic security, such as in governmental or military applications, large-scale financial systems, and critical infrastructure systems. In the mentioned cases, the benefits of improved security provided by the SIS-based scheme justify the increased computational complexity and larger key and signature sizes. The suggested method guarantees that digital signatures are safe even from quantum computer assaults by utilizing well-known secure lattice-based principles.

Despite the advancements in post-quantum cryptographic research, many existing schemes either rely on computationally expensive structures, such as Learning with Errors (LWE), or suffer from efficiency limitations due to large key and signature sizes. Traditional ElGamal signatures, while efficient, remain vulnerable to Shor's algorithm, making them unsuitable for the post-quantum era. The Short Integer Solution (SIS) problem is particularly well-suited for digital signatures due to its strong worst-case hardness guarantees and efficient verification properties. Unlike LWE, SIS does not require noise sampling, simplifying implementation while retaining post-quantum security. To bridge the gap between security and efficiency, we propose a digital signature scheme that integrates the fundamental security of ElGamal with the hardness of SIS. This novel approach ensures resistance against quantum attacks while maintaining computational feasibility, making it a necessary advancement in post-quantum digital signatures

2. The Overview of Lattice-based Schemes

2.1. Lattice-based Schemes

The most important area of post-quantum cryptography is lattice-based cryptography. Because of its great efficiency and scalability, it may be used for everything from simple cryptographic building blocks to more intricate protocols. In this section, we will talk about post-quantum cryptography, specifically lattice-based cryptography. The security of lattice-based cryptography relies on complex problems related to lattice points within the m −dimensional Euclidean space R^m. Furthermore, lattice-based cryptography, which is predicated on the most gloomy assumptions, guarantees protection against a variety of quantum computer threats.

2.2. Lattice

A lattice is a specific type of ordered set that involves two binary operations used to combine sets of vectors. Lattice structures have widespread applications in both mathematics and computing [9]. The following is the formal definition of a lattice.

Definition. Given $b_1, b_2, \dots, b_n \in \mathbb{R}^m$ as a set of linearly independent vectors, a lattice L in m dimensional Euclidean space $\mathbb{R}^m$, produced by $b_1, b_2, \dots, b_n$, is defined as:

$$L(b_1, b_2, \dots, b_n) = \{\sum_{i=1}^{n} a_i b_i : a_i \in \mathbb{Z}\}. \quad (1)$$

The integers n and m represent the order and dimension of the lattice, respectively, while $b_1, b_2, \dots, b_n$ are referred to as the basis of the lattice. The minimum distance of a lattice L is defined as the length of the shortest non-zero vector b within the lattice:

$$D_{min}(L) = \min_{b \in L \setminus \{0\}} \|b\| \quad (2)$$

A lattice can be seen as a linear arrangement of points in space, specifically a set of points within n dimensional space. For example, a 2-dimensional lattice is shown in Figure 1. In this case, the basis vectors $(1, 1)$ and $(1, -1)$ generate the lattice in $\mathbb{R}^2$, but there are other possible bases. For instance, the sum of the bases $(1, 1)$ and $(1, -1)$ results in the

vector $(2, 0)$, and multiplying the basis $(1, 1)$ by 2 and adding $(1, -1)$ produces the vector $(3, 1)$, both of which are also part of the lattice shown in Figure 1.

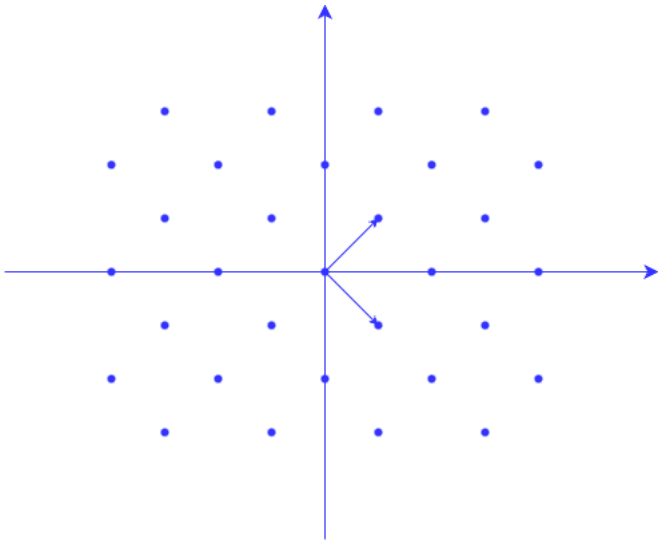

Fig.1. Integral Lattice in $\mathbb{R}^2$ Formed by All Vectors in $\mathbb{R}^2$ with integer coefficients

A lattice basis $\mathcal{L}$ is a set of linearly independent vectors that fully span the lattice. A lattice may have multiple distinct bases, but each basis will always consist of the same number of vectors.

Lemma 1. All lattices $\mathcal{L} \subset \mathbb{R}^m$ have at least one basis.

Any basis of a lattice can be represented as a matrix $B = [b_1, b_2, \dots, b_n]$, where the basis vectors are used as the columns, and B is the basis matrix, with $B \in \mathbb{Z}^{m \times n}$.

Second Definition. A lattice $\mathcal{L}$ generated by a basis matrix $B \in \mathbb{Z}^{m \times n}$ is defined as: $L(B) = \{Ba : a \in \mathbb{Z}^n\}$, where Ba represents matrix-vector multiplication. A basis B is not unique for any lattice $L(B)$. A matrix is considered unimodular if its determinant is ± 1. For any unimodular matrix $U \in \mathbb{Z}^{n \times n}$, the product $B \cdot U$ is also a valid basis for $L(B)$.

Lemma 2. A subset of $\mathbb{R}^m$ constitutes a lattice if and only if it is a discrete additive subgroup.

2.3. The Overview of Computational Problems of Lattices

In lattice theory, the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) are two important issues that have been thoroughly researched. Finding the shortest nonzero vector in the lattice where the vector's length is reduced in accordance with the Euclidean norm is the aim of the SVP, given a lattice basis. On the other hand, the CVP entails locating the lattice vector in the space that is closest to a particular target vector.

The original issues are extended by the approximate versions, Approximate SVP (apprSVP) and Approximate CVP (apprCVP), which permit estimates rather than exact answers. In the worst scenario, these approximation problems are regarded as hard, which means that it is computationally challenging to develop even approximate answers. Due to their inherent difficulty, both SVP and CVP, along with their approximate versions, have found numerous applications in cryptography. These problems serve as the foundation for many cryptographic protocols, providing security guarantees based on their hardness [10 -13]. Lattice-based cryptography has been built upon the hardness of well-studied computational problems such as the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP). These problems, along with their approximate versions (apprSVP and apprCVP), are widely regarded as computationally difficult, even in the presence of quantum adversaries. Notably, research demonstrated that lattice problems such as SVP and CVP could serve as the foundation for constructing secure public-key encryption and digital signature schemes in the post-quantum era.

Additionally, lattice-based hash functions, like those based on the SIS problem, offer promising resistance to quantum attacks. The development of one-way functions based on the hardness of solving the n^c-approximate SVP continues to be a critical area of research, forming the backbone of many secure cryptographic systems.

Third Definition. Shortest vector problem (SVP).

The shortest vector problem (SVP): given a matrix $B \in \mathbb{Z}^{m \times n}$ of a lattice $L(B)$, determine the shortest non-zero vector $b \in L$ such that $\|b\| = D_{\min}(L)$.

Fourth Definition. Closest vector problem (CVP).

The closest vector problem (CVP): given a matrix $B \in \mathbb{Z}^{m\times n}$ of a lattice $L(B)$ and a vector $c \notin L$, identify $b \in L$ such that $\|c - b\| = D_{\min}(L)$.

2.4. Short Integer Solution (SIS)

The Short Integer Solution (SIS) problem is widely used in the construction of lattice-based cryptography. Let $\mathbb{Z}_q^n$ denote n-dimensional vectors modulo q. Given $\vec{a}_1, \dots, \vec{a}_m \in \mathbb{Z}_q^n$, find small, nontrivial $z_1, \dots, z_m \in \mathbb{Z}$ such that:

$$z_1\vec{a}_1 + \cdots + z_m\vec{a}_m = 0 \tag{3}$$

in $\mathbb{Z}_q^n$, i.e.,

$$Az = 0 \pmod q \tag{4}$$

where $A = [\vec{a}_1, \dots, \vec{a}_m]$. This involves finding a short vector in the lattice:

$$L(A)^{\perp} := \ker\left(A \in \mathbb{Z}_q^{n\times m}\right) = \{x \in \mathbb{Z}^m : Ax = 0 \pmod q\} \tag{5}$$

The SIS problem forms the foundation of our proposed digital signature scheme. By leveraging the hardness of finding short integer solutions in a lattice, we ensure that the scheme is resistant to quantum attacks. In the following sections, we describe how the SIS problem is integrated into the key generation, signature generation, and verification processes." The SIS problem forms the foundation of our proposed digital signature scheme. By leveraging the hardness of finding short integer solutions in a lattice, we ensure that the scheme is resistant to quantum attacks. In the following sections, we describe how the SIS problem is integrated into the key generation, signature generation, and verification processes.

2.5. Lattice Based One-way Hash Function

Lattice-based one-way hash functions have attracted considerable attention due to their resistance to quantum computing threats. Researchers have demonstrated that constructing collision-resistant hash functions using the hardness of lattice problems, such as SIS and LWE, offers significant advantages over traditional hash functions. The ability to secure these functions in a quantum era positions lattice-based approaches as a promising direction for future cryptographic applications.

Recent advancements in this area have focused on optimizing these constructions for practical implementation. For example, the use of short integer solution problems for generating secure hash functions has been refined, with new schemes significantly improving both efficiency and security. This progress reflects a broader trend in lattice-based cryptography, where practical implementations are being explored for real-world deployment, especially in contexts requiring quantum-resistant properties.

In cryptography, developing a one-way hash function based on lattice issues is a persistent research topic and a considerable challenge. Since hash functions based on lattice problems are thought to be more resilient to attacks by quantum computing, there is increasing interest in creating such. Using the complexity of some lattice problems, like the Short Integer Solution (SIS) problem or the Learning with Errors (LWE) problem, is one promising strategy [14 - 17]. Even with quantum computers, these issues are still challenging to resolve, making them promising options for cryptographic primitives in the post-quantum computing age.

Let $m > n\log\, q$. Define $f_A: \{0,1\}^m \to \mathbb{Z}_q^n$ as:

$$f_A(x) = Ax, \tag{6}$$

where f_A must cover $\mathbb{Z}_q^n$ almost uniformly. As $m > n\log\, q$, the number of domain elements 2^m is significantly larger than the number of range elements q^n.

We assert a collision $x, x' \in \{0,1\}^m$ when $Ax = Ax'$. Here, $A = [\vec{a}_1, \dots, \vec{a}_m] \in \mathbb{Z}_q^{n\times m}$ determines a q-ary lattice:

$$L^{\perp}(A) = \{z \in \mathbb{Z}^m : Az = 0 \pmod q\} \tag{7}$$

Consequently, SIS is equivalent to SVP on $L^{\perp}(A)$.

2.6. The Review of Ajtai Hash Function

Ajtai's pioneering research introduced the first lattice-based cryptographic framework with worst-case security guarantees. He proposed a family of one-way functions whose security relies on the computational hardness of solving the n^c-approximate Shortest Vector Problem (SVP) for a constant $c > 0$. Ajtai demonstrated that if an adversary could invert these functions with a non-negligible probability, they would also be able to solve the n^c-approximate SVP for arbitrary instances [18,19]. This breakthrough formed the basis for further developments, including efforts to strengthen

security proofs. For example, Goldreich and collaborators proved that Ajtai's function exhibits not only onewayness but also collision resistance, a property with broader cryptographic applications. Later work focused on reducing the constant c to enhance security assumptions, with recent advancements bringing c closer to 1. However, it is crucial to note that such constructions remain tied to lattice problems that are not considered NP-hard, particularly for $c \geq 1/2$.

The essence of these findings lies in selecting appropriate parameters q, n, and m. Solving for short vectors in the dual lattice $\Lambda_q^{\perp}(A)$, where A is randomly selected from $\mathbb{Z}_q^{n\times m}$, is computationally as challenging as addressing approximate lattice problems such as the Short Integer Solution Problem (SIVP) or the Shortest Vector Problem (SVP) in the worst case. This result holds even if the algorithm succeeds with only an inverse polynomial probability based on the randomness of A.

Once the reduction from worst-case to average-case scenarios is established, creating collision resistant hash functions becomes relatively straightforward. These hash functions depend on parameters n, m, q, and d. Typical choices might include $d = 2, q = n^2$, and $m > n\log(q)/\log(d)$. The parameter n directly influences the security of the hash function. A key is represented by a randomly chosen matrix A from $\mathbb{Z}_q^{n\times m}$, and the hash function f_A is defined as: $f_A(y) = A \cdot y \bmod q$

This function maps $m\log(d)$ bits to $n\log(q)$ bits. To achieve compression, parameters are typically set such that $m \approx 2n\log(q)/\log(d)$, resulting in a compression factor close to 2

A collision in this hash function occurs when $f_A(y) = f_A(y')$ for $y \neq y'$. This implies the existence of a short non-zero vector $y - y' \in \Lambda_q^{\perp}(A)$. Leveraging the worst-case to average-case reduction shows that finding such collisions (even with inverse polynomial probability) is as challenging as solving approximate SIVP or SVP in the worst case.

Key Parameters:

- Integers: $n, m, q, d \geq 1$
- Key: A random matrix A from $\mathbb{Z}_q^{n\times m}$
- Hash Function: $f_A: \{0, \ldots, d-1\}^m \to \mathbb{Z}_q^n$, defined as $f_A(y) = A \cdot y \bmod q$

This hash function is computationally efficient, requiring only basic addition and multiplication modulo q. Since q is of size $O(\log n)$, it fits within standard memory or processor registers, enabling efficient operations without the need for large, arbitrary-precision integers often used in other cryptographic schemes.

3. Methodology

This section presents the logical progression toward the construction of our hybrid SIS-ElGamal digital signature scheme. We begin in Section 3.1 with a summary of the classical ElGamal digital signature system, which forms the probabilistic foundation of our approach. In Section 3.2, we introduce the Short Integer Solution (SIS) problem and associated lattice-based tools that offer strong post-quantum security guarantees. Building on these foundations, Section 3.3 reviews the idea of construction of key exchange protocol based using sis problem. Section 3.4 details the proposed hybrid scheme, combining the efficiency of ElGamal with the quantum resilience of SIS. This includes the key generation algorithm, signature and verification process, as well as a discussion of the security rationale and implementation considerations.

3.1. ElGamal Public Digital Signature Scheme

A cryptographic approach called the ElGamal digital signature system was created to guarantee the integrity and validity of digital communications. It makes use of asymmetric cryptography, which uses two keys: a public key for verification and a private key for signature [20].

Key Steps in the ElGamal Digital Signature Scheme

Key Generation:

- Select a large prime number p and a primitive root g modulo p.
- Choose a random private key x such that $1 \leq x \leq p-2$.
- Compute the public key h_1 as $h_1 = g^x \bmod p$.
- The private key is x, and the public key is (p, g, h_1).

Signature Generation:

- Choose a random integer y such that $1 \leq y \leq p-2$ and $\gcd(y, p-1) = 1$.
- Compute h_2 as $h_2 = g^y \bmod p$.
- Calculate the hash $H(m)$ of the message m using a cryptographic hash function.
- Compute the signature component σ as $\sigma = (H(m) - x \cdot h_2) \cdot y^{-1} \bmod (p-1)$, where y^{-1} is the modular inverse of y modulo $p-1$.
- The digital signature is (h_2, σ).

Signature Verification:

- Use the public key (p, g, h_1) of the signer to verify the signature.
- Compute the hash $H(m)$ of the received message m.
- Calculate $v_1 = h_1^{h_2} \cdot h_2^{\sigma} \bmod p$.
- Calculate $v_2 = g^{H(m)} \bmod p$.
- If $v_1 = v_2$, the signature is valid; otherwise, it is invalid.

Properties of the ElGamal Digital Signature Scheme:

- Message Integrity: Guarantees that the message has not been altered during transmission.
- Non-repudiation: Prevents the signer from denying authorship, as anyone with the public key can verify the signature.
- Authentication: Confirms the signer's identity through the signature verification process.

A fixed-size hash function is used to construct the message digest for this scheme, which is crucial to the verification and signature formation processes. While the private key is used to sign the message, the public key facilitates signature verification and guarantees the cryptographic integrity of the system [21,22].

Compared to alternative digital signature algorithms, the ElGamal Digital Signature Scheme offers a number of benefits. It offers a high degree of security against threats like impersonation, manipulation, and forging and is comparatively simple to deploy. Furthermore, because the private key is never sent or exchanged, the technique offers a high degree of key security. While the traditional ElGamal scheme relies on discrete logarithms, our proposed scheme integrates ElGamal's probabilistic framework with the hardness of the SIS problem. This hybrid approach ensures both efficiency and post-quantum security, as detailed in the following sections.

But there are drawbacks to the ElGamal Digital Signature Scheme as well. Large key sizes are necessary to maintain security, yet it is computationally demanding. It is also susceptible to some types of attacks, including side-channel and key compromise attacks.

To mitigate these challenges and enhance the security of the proposed scheme, the following key parameters have been carefully selected:

Matrix Dimension (n): The security level is directly related to the matrix dimension. We selected n = 512 and n = 1024 to provide strong security against quantum attacks, following best practices in lattice-based cryptography.

Modulus (q): A large prime modulus q ensures the cryptographic hardness of the scheme. The value q = 12289 is chosen to balance efficiency and security, making it resistant to lattice reduction attack

Bound on Secret Vectors (β): The secret vectors are sampled with a norm bounded by β = 8, limiting information leakage while maintaining resistance against cryptanalytic attacks.

Parameter Justification: These parameters are selected based on the worst-case hardness assumptions of the Shortest Vector Problem (SVP) and Short Integer Solution (SIS) problems. This ensures that the scheme is secure against both classical and quantum adversaries.

3.2. Short Integer Solution (SIS) Problem

The Short Integer Solution (SIS) problem is a foundational hard problem in lattice-based cryptography, widely used for constructing secure hash functions, digital signatures, and encryption schemes. Formally, the SIS problem is defined as follows:

Definition (SIS): Let $A \in \mathbb{Z}_q^{m \times n}$ be a uniformly random matrix, where q is a prime modulus, and let β be a bound on the norm. The goal is to find a non-zero integer vector $\mathbf{x} \in \mathbb{Z}^n$ such that:

$$A \cdot \mathbf{x} \equiv \mathbf{0} \ (\mathrm{mod} q), \text{ and } \|\mathbf{x}\| \leq \beta.$$

This means finding a short vector in the nullspace of A modulo q. The difficulty of SIS arises from the need to find a short solution, which is believed to be computationally hard even for quantum adversaries. The SIS problem can be interpreted as a search for lattice vectors with specific modular properties and bounded length, making it suitable for constructing post-quantum cryptographic primitives.

3.3. The Construction of Key Exchange Protocol based Using SIS Problem

A notable and complex issue is the "Short Integer Solution problem" - SIS problem, originally introduced by Ajtai. Some attempts have been made to formulate a cryptographic key exchange protocol utilizing the Short Integer Solution problem. However, despite various endeavors in developing key exchange protocols based on Short Integer Solution problem, the fundamental architecture remains as follows.

- Alice and Bob agree to use a random matrix $R \in \mathbb{Z}_q^{n \times m}$, where q is a modulus, and R serves as a public parameter for the protocol.
- Alice selects her secret key $s_A \in \mathbb{Z}_q^m$, ensuring its norm satisfies $\|s_A\| \leq \beta$, where β is a predefined bound. She

computes $P_A = Rs_A$ and sends P_A to Bob.

- Bob selects his secret key $s_B \in \mathbb{Z}_q^n$, ensuring its norm satisfies $\|s_B\| \leq \beta$. He computes $P_B = s_B^T R$ and sends P_B to Alice.
- Upon receiving P_B, Alice computes her shared key K_A as: $K_A = s_A^T P_B^T = s_A^T R^T s_B$
- Similarly, upon receiving P_A, Bob computes his shared key K_B as $K_B = P_A^T s_B = s_A^T R^T s_B$
- Since both computations lead to the same value, $K_A = K_B = K$, they establish a shared secret key.

When leveraging the SIS problem for secure key exchange, it is critical to adjust the dimensions of the matrix R and the associated vectors s_A and s_B. For protecting Alice's secret key, $n \ll m$ ensures more variables than equations, complicating the reconstruction of s_A from P_A. For safeguarding Bob's secret key, $m \gg n$ provides a similar layer of protection against deriving s_B from P_B. The protocol's design ensures that both parties deal with an overdetermined system, where the number of variables exceeds the number of equations. This property, inherent to the SIS problem, underpins the security of the key exchange mechanism, making it resilient to attempts at reconstructing the secret keys from the shared data.

The modulus q=12289 is selected because it is a prime number that balances security and efficiency. A prime modulus ensures that the underlying algebraic structure is well-defined, and q=12289 is large enough to resist lattice reduction attacks while small enough to allow efficient modular arithmetic.

The matrix dimensions n=512 and n=1024 are selected based on the desired security level. These dimensions provide a balance between security and computational efficiency. For example, n=512 offers 128-bit security against classical attacks, while n=1024 provides 256-bit security. These values are consistent with the NIST Post-Quantum Cryptography standardization process, which suggests similar dimensions for lattice-based schemes.

The bound on secret vectors β=8 is chosen to limit information leakage while maintaining resistance to cryptanalytic attacks. A small β ensures that the secret vectors are short, which is necessary for the hardness of the SIS problem. This choice of β=8 provides sufficient security against known lattice attacks while keeping the computational overhead manageable.

To the best of our knowledge, this is the first digital signature scheme that directly integrates the classical ElGamal signing framework with the hardness of the SIS problem in a structured hybrid design. While lattice-based signature schemes and ElGamal-inspired post-quantum variants have been independently explored, a combined ElGamal–SIS architecture has not been formally constructed or analyzed prior to this work.

Our approach differs in several important ways:

- It replaces discrete-logarithm hardness with a lattice-based hardness assumption (SIS), offering resilience against quantum attacks.
- The verification logic remains computationally lightweight, comparable to ElGamal, while gaining provable security based on worst-case SIS assumptions.
- Signature size remains constant (1.7 KB) regardless of message length, due to a hash-and-sign structure — a practical efficiency not guaranteed in many lattice schemes.
- The implementation uses only integer arithmetic, reducing the risk of floating-point side-channel leaks common in LWE-based constructions.

These design choices make the proposed scheme a practical and secure candidate for post-quantum environments where both security and simplicity are required.

To help visualize the computational flow of our SIS-based ElGamal digital signature scheme, Figure X outlines the major steps involved in signing and verifying a message. The diagram highlights the role of the hash function, secret key, public key, and the final signature verification check. This flow clearly separates the signing, and verification phases and emphasizes the input/output dependencies at each stage. Check Figure 2.

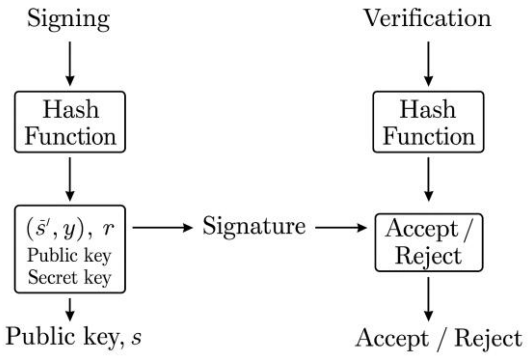

Fig.2. High-level schematic of the SIS-based ElGamal digital signature process. The left side depicts the signing process using the secret key, while the right side shows the signature verification using the public key

3.4. Proposed Hybrid Scheme

In this section, we present a novel digital signature scheme based on the principles of Elgamal and the SIS key exchange. The choice of SIS over other lattice problems, such as Learning with Errors (LWE), is motivated by its structural advantages in digital signature schemes. SIS naturally aligns with hash-and-sign methodologies, making it well-suited for signature construction without requiring additional noise terms. Unlike LWE, which involves Gaussian noise sampling and complex decryption procedures, SIS relies on finding short vectors in a lattice, leading to more straightforward implementations and efficient verification processes. Additionally, SIS enjoys strong worst-case to average-case security reductions, ensuring that breaking a single instance of the problem implies solving hard lattice problems in the worst case. This property enhances the overall security of the proposed digital signature scheme. Furthermore, SIS-based signatures tend to have smaller key sizes compared to LWE-based alternatives, making them more practical for real-world applications. Given these advantages, SIS is a compelling choice for constructing post-quantum digital signatures with both security and efficiency in mind.

A. Key Generation

Parameter Selection:

- Choose a prime q and dimensions n and m, where $n < m$.
- Generate a random matrix R of size $n \times m$ over the finite field $\mathbb{Z}_q$.

Secret Key Generation (Alice):

- Alice selects a secret vector $\vec{s}_A \in \mathbb{Z}_q^m$ with a small norm (bounded by some parameter β).

Public Key Generation (Alice):

- Compute the public key $\vec{P}_A$ as: $\vec{P}_A = R \cdot \vec{s}_A$
- Send $\vec{P}_A$ to Bob.

Secret Key Generation (Bob):

- Bob selects a secret vector $\vec{s}_B \in \mathbb{Z}_q^n$ with a small norm (bounded by β).

Public Key Generation (Bob):

- Compute Bob's public key P_B as: $P_B = \vec{s}_B^T \cdot R$
- Send P_B to Alice.

B. Signature Generation

Message Hashing:

- Hash the message m to obtain a value $H(m)$.

Signature Vector Construction:

- Alice creates a signature vector $\vec{\sigma}_A$ such that: $\vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$
- Alice's signature is this combination $(\vec{s}_A, \vec{\sigma}_A)$.

C. Signature Verification

Bob receives the message m and the signature $(\vec{s}_A, \vec{\sigma}_A)$.

Hash Calculation:

- Compute $H(m)$.

Verification Check:

- Verify whether: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$

Proof of Signature Verification

To prove the correctness of the verification step, we need to verify that if a signature is valid, it will satisfy the verification equation.

Verification Equation:

- The verification procedure should be used first: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$

Substitute $\vec{\sigma}_A$:

- Substitute $\vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$ into the equation: $R \cdot \vec{\sigma}_A = R \cdot (H(m) - R \cdot \vec{s}_A)$

Distribute R :

- Distribute R across the terms: $R \cdot \vec{\sigma}_A = R \cdot H(m) - R \cdot (R \cdot \vec{s}_A)$

Simplify Terms:

- Since $R \cdot (R \cdot \vec{s}_A)$ represents the matrix multiplication of R with itself and $\vec{s}_A$, the terms involving R and $\vec{s}_A$ cancel out if $\vec{\sigma}_A$ is computed as $H(m) - R \cdot \vec{s}_A : R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$

Conclusion:

If $\vec{\sigma}_A$ is indeed $H(m) - R \cdot \vec{s}_A$, then the verification equation: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$ will be satisfied. Therefore, the verification step ensures that the signature $\vec{\sigma}_A$ was generated by someone who knew the secret key $\vec{s}_A$ and the hash of the message $H(m)$, confirming the authenticity of the signature

In lattice-based cryptographic schemes, especially those involving matrix computations, numerical errors and floating-point inaccuracies can compromise security and correctness. The proposed scheme carefully addresses these challenges by exclusively using integer arithmetic in all critical computations. This avoids rounding errors and precision loss that are common with floating-point operations.

During key generation, signature generation, and verification, all matrix operations are performed over finite fields (modulo a prime q), ensuring exact results. Additionally, the scheme employs modular reduction techniques to control the size of intermediate results, preventing overflow or unintended behavior.

For practical implementations, further measures like constant-time operations and error detection mechanisms will be explored to minimize side effects caused by hardware-level inaccuracies. This design choice ensures that numerical errors do not affect the security or reliability of the scheme.

The offered scheme is built on the Short Integer Solution (SIS) problem due to its natural suitability for digital signature construction. SIS enables more straightforward and efficient implementations compared to LWE, which typically requires handling complex noise terms. Additionally, SIS offers tighter security reductions to well-established hard lattice problems like the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP). This structure leads to smaller key and signature sizes, making SIS-based schemes more efficient and practical for digital signatures while maintaining strong security guarantees against quantum attacks.

The importance of addressing key management challenges, particularly in large-scale systems, must be emphasized. In such systems, key generation, distribution, and storage become more complex, especially with larger key sizes required by post-quantum schemes like SIS-based signatures.

The SIS-based scheme requires the secure generation and storage of relatively large keys. To address this, efficient key storage solutions, including techniques like key splitting or sharding, could be employed. These approaches divide the key into smaller components, distributed across multiple systems, preventing any single entity from accessing the full key. For large-scale systems, secure and efficient key distribution and revocation mechanisms are crucial. A hierarchical key distribution system supported by a public key infrastructure (PKI) could be used to manage key issuance, revocation, and updating. A central trusted authority could oversee the distribution process, ensuring the scalability and security of key management. Key lifecycle management, including automated solutions, could play a vital role in facilitating key rotations, revocations, and updates, reducing the administrative burden of managing keys in large-scale systems.

4. Security

4.1. Fundamental Security Discussion

Our SIS-based Elgamal signature scheme provides three crucial security properties:

- Message Integrity: The scheme gives guarantee that the message was not changed during the transmission. This is achieved because the signature is uniquely tied to the content of the message and can be verified against it. If the signature verification succeeds, it confirms that the message remains unchanged.
- Non-Repudiation: The scheme ensures that the signer cannot later claim they did not sign the message. This is ensured by the fact that only the signer knows the secret key necessary to create a valid signature. Once signed, the signer cannot deny their involvement because the signature is exclusive to the message and the secret key.
- Authentication: The scheme allows the recipient to verify the authenticity of the signer. By using the public key of the signer, the recipient can check that the signature matches the message. If the verification is successful, it confirms that the signature was indeed created by the holder of the corresponding secret key.

A. Message Integrity

Message integrity ensures that no changes were made to the message while it was being transmitted.

How it is achieved:

- Alice creates a signature vector $\vec{\sigma}_A$ such that: $\vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$ where $H(m)$ is the hash of the message m, R is the matrix, and $\vec{s}_A$ is Alice's secret key.

Verification:

- Bob verifies the signature using the equation: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$
- If this equation holds true, it means the message has not been altered since the signature was created.

B. Non-Repudiation

The inability of the signer to retract their signature guarantees non-repudiation.

How it is achieved:

- Alice's public key $\vec{P}_A$ is computed as: $\vec{P}_A = R \cdot \vec{s}_A$
- To create a valid signature, Alice must use her secret key $\vec{s}_A$ correctly.

Verification:

- Bob uses Alice's public key to confirm the signature. Alice signed the message if the verification equation: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$ is satisfied, as only she could have used her secret key $\vec{\beta}_A$.

C. Authentication

Authentication guarantees that the recipient can confirm the signer's legitimacy.

How it is achieved:

- Alice's public key $\vec{P}_A$ allows Bob to verify the signature.

Verification:

- The recipient checks if: $R \cdot \vec{\sigma}_A = H(m) - R \cdot \vec{s}_A$
- It confirms that the message is valid and that Alice actually created the signature if this equation holds true with Alice's public key.

The proposed SIS-based ElGamal digital signature scheme is designed to be secure against quantum attacks by incorporating post-quantum cryptographic principles. Specifically, the scheme uses post-quantum secure hash functions that are resistant to Grover's algorithm. Since Grover's algorithm can accelerate brute-force search attacks, the use of hash functions with sufficiently large output lengths effectively mitigates this threat. Additionally, the scheme avoids relying on mathematical operations vulnerable to Shor's algorithm, such as integer factorization and discrete logarithms. By building the scheme on the hardness of the Short Integer Solution (SIS) problem, it ensures security against both classical and quantum adversaries. Short Integer Solution (SIS) problem, serves as the foundation of our post-quantum signature scheme, is believed to be resistant to Shor's algorithm. Unlike problems solvable by Shor's algorithm, the SIS problem does not have a known efficient quantum algorithm, making it a promising candidate for post-quantum cryptography.

On the other hand, Grover's algorithm accelerates brute-force search, reducing the time complexity of finding a solution from O(N) to O(√N). Although this reduces the security level of the cryptosystem, the SIS problem still maintains a strong resistance to quantum attacks. This is because the quantum speedup provided by Grover's algorithm does not fundamentally undermine the hardness of the SIS problem. Therefore, our scheme remains robust against quantum adversaries, offering a security guarantee that is based on the hardness of the SIS problem in a quantum setting. The SIS problem is closely related to other well-known lattice-based problems, particularly the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP). While the SIS problem itself involves finding short integer solutions to linear systems, its hardness is derived from the difficulty of solving the approximate versions of SVP and CVP. Specifically, reductions from the approximate SVP to SIS have been established, showing that solving SIS is at least as hard as solving SVP in certain lattice settings. The connection between these problems provides a strong security foundation, as it has been demonstrated that if an adversary could efficiently solve the SIS problem, they would also be able to solve the SVP or CVP in the worst case. This reduction guarantees that our proposed scheme inherits the security of these well-studied lattice problems. Furthermore, the hardness of SIS remains robust even in the presence of quantum computers, making it a reliable choice for post-quantum security. In addition to these reductions, the choice of lattice parameters, such as the modulus and matrix dimensions—further reinforces the security guarantees. By selecting parameters that align with the hardness of these lattice problems, we ensure that breaking the cryptographic scheme would require solving intractable problems that are believed to remain hard even against quantum adversaries.

4.2. Practical Security Considerations

While the proposed SIS-based ElGamal digital signature scheme offers strong theoretical security based on the hardness of the SIS problem, it is essential to consider practical attacks and vulnerabilities that could arise in real-world implementations. Below, we discuss potential practical attacks and how the proposed scheme can be implemented to mitigate these threats.

Side-Channel Attacks:

Side-channel attacks exploit information leaked during the execution of cryptographic operations, such as timing, power consumption, or electromagnetic emissions. To mitigate these attacks, the proposed scheme can be implemented using the following techniques:

- Constant-Time Implementations: Ensure that all operations, particularly those involving secret keys, execute in constant time to prevent timing attacks.
- Blinding Techniques: Introduce random values during computations to obscure the relationship between the secret key and the observed side-channel data.
- Masking and Noise Injection: Apply masking techniques to hide intermediate values and inject noise to reduce the effectiveness of power analysis and electromagnetic attacks.

Fault Injection Attacks:

Fault injection attacks involve introducing errors into the computation to reveal secret information or disrupt the cryptographic process. To protect against these attacks, the proposed scheme can incorporate the following countermeasures:

- Redundant Computation Checks: Perform redundant computations and compare results to detect and correct errors introduced by fault injection.
- Error-Detection Mechanisms: Implement error-detecting codes to ensure that any injected faults are detected before they can compromise the system.
- Secure Hardware Modules: Use hardware security modules (HSMs) to protect sensitive operations from physical fault injection attacks.

Key Management Vulnerabilities:

Key management is a critical aspect of cryptographic systems, and vulnerabilities in this area can lead to key compromise or misuse. To address key management challenges, the proposed scheme can employ the following strategies:

- Secure Key Storage: Store secret keys in secure hardware modules or encrypted storage to prevent unauthorized access.
- Key Rotation and Erasure: Regularly rotate keys and securely erase old keys to limit the impact of key compromise.
- Hierarchical Key Distribution: Use a hierarchical key distribution system supported by a public key infrastructure (PKI) to manage key issuance, revocation, and updating.

Implementation flaws, such as buffer overflows or incorrect parameter handling, can introduce vulnerabilities into cryptographic systems. To minimize these risks, the proposed scheme should be implemented using the following best practices:

- Code Audits and Testing: Conduct thorough code audits and testing to identify and fix potential implementation flaws.
- Formal Verification: Use formal verification techniques to ensure that the implementation adheres to the cryptographic specifications.
- Secure Coding Practices: Follow secure coding practices to prevent common vulnerabilities, such as buffer overflows or integer overflows.

By incorporating these practical security considerations, the proposed SIS-based ElGamal digital signature scheme can be made resilient to real-world attacks, ensuring that it remains secure in practical deployments.

4.3. Security Proof

To establish the security of the proposed SIS-based EIGamal digital signature scheme, we provide detailed theoretical proof based on the hardness of the Short Integer Solution (SIS) problem. This proof demonstrates that forging a signature in the proposed scheme is computationally equivalent to solving the SIS problem, which is believed to be hard even for quantum adversaries.

- Security Assumptions:

The security of the proposed scheme relies on the hardness of the SIS problem, which is defined as follows: Given a random matrix $A \in \mathbb{Z}_q^{n\times m}$ and a target vector $\mathbf{b} \in \mathbb{Z}_q^n$, find a short non-zero vector $\mathbf{z} \in \mathbb{Z}^m$ such that $A\mathbf{z} \equiv \mathbf{b}(\mathrm{mod}q)$. The SIS problem is believed to be hard in the worst case, even for quantum computers, making it a suitable foundation for post-quantum cryptographic schemes.

- Security Reduction:

We prove the security of the proposed scheme by reducing the problem of forging a signature to the problem of solving the SIS problem. Specifically, we show that if an adversary can forge a signature with non-negligible probability, then they can also solve the SIS problem with non-negligible probability. This reduction ensures that the proposed scheme is secure as long as the SIS problem is hard.

Proof Sketch:

- Adversary Model: Assume that an adversary $\mathcal{A}$ can forge a signature for the proposed scheme with non-negligible probability. This means that $\mathcal{A}$ can produce a valid signature $(\mathbf{s}_A^*, \sigma_A^*)$ for a message m without knowing the secret key $\mathbf{s}_A$.
- Signature Forgery: The forged signature satisfies the verification equation:

$$R \cdot \sigma_A^* \equiv H(m) - R \cdot \mathrm{s}_A^* \,(\mathrm{mod} q) \tag{8}$$

where R is the public matrix, $H(m)$ is the hash of the message, and $\mathbf{s}_A^*$ is the forged secret key.

Reduction to SIS: Let $\mathbf{z}^* = \sigma_A^* + \mathbf{s}_A^*$. Then, the verification equation can be rewritten as:

$$R \cdot \mathbf{z}^* \equiv H(m) \,(\mathrm{mod} q) \tag{9}$$

Since $\mathbf{z}^*$ is a short vector, this implies that $\mathbf{z}^*$ is a solution to the SIS problem for the matrix R and target vector $H(m)$.

Conclusion: If $\mathcal{A}$ can forge a signature, then $\mathcal{A}$ can solve the SIS problem. Therefore, the proposed scheme is secure as long as the SIS problem is hard.

- Formal Security Analysis:

The security of the proposed scheme can be formally analyzed using the random oracle model, where the hash function H is modeled as a random oracle. In this model, the adversary has no advantage in predicting the output of H, and the security of the scheme reduces to the hardness of the SIS problem. Specifically, we can show that the probability of forging a signature is negligible in the security parameter λ, assuming that the SIS problem is hard. Theorem: Let $\mathcal{A}$ be an adversary that can forge a signature for the proposed scheme with probability ϵ. Then, there exists an algorithm $\mathcal{B}$ that solves the SIS problem with probability ϵ', where ϵ' is nonnegligible if ϵ is non-negligible.

Proof: The proof follows the reduction outlined above. If $\mathcal{A}$ can forge a signature, then $\mathcal{B}$ can use $\mathcal{A}$ to solve the SIS problem by constructing a valid signature and extracting a short vector $\mathbf{Z}^*$ that satisfies $R \cdot \mathbf{z}^* \equiv H(m)(\mathrm{mod} q)$. Since $\mathbf{z}^*$ is a solution to the SIS problem, the hardness of the SIS problem implies that ϵ must be negligible.

By providing this detailed theoretical proof and formal security analysis, we establish that the proposed SIS-based EIGamal digital signature scheme is secure under the assumption that the SIS problem is hard.

5. Experiments

5.1. Setup and Experiments

The implementation of the SIS-based digital signature scheme was carried out using **Python 3.8**, with several libraries utilized for cryptographic functions and performance measurements. The experiments were conducted on a desktop machine with the following specifications:

- Processor: Intel Core i7-8565U (4 cores, 8 threads, 1.8 GHz base frequency, up to 4.6 GHz turbo boost)
- Memory: 16 GB DDR4 RAM (operating at 3200 MHz)
- Storage: 1 TB NVMe SSD
- Operating System: Windows 10 Pro (64-bit version)

The following libraries were used for cryptographic operations and performance measurements:

- NumPy: For matrix operations and linear algebra.
- PyCryptodome: For cryptographic hash functions and modular arithmetic.
- SciPy: For additional mathematical computations.
- Cryptography: For secure random number generation and key management.

Testing Methodology:

- Key Generation: The time required to generate public and private keys was measured for matrix dimensions of 100 × 100, 200 × 200, and 300 × 300.
- Signature Generation: The time required to generate signatures was measured for messages of sizes 1 KB, 10 KB, and 100 KB.
- Signature Verification: The time required to verify signatures was measured for the same message sizes.
- Memory Consumption: The memory usage during key generation, signature generation, and verification was recorded using Python's memory profiling tools.

Each experiment was repeated **10 times** to ensure accuracy, and the average values are reported in the results. This setup was chosen to represent a typical workstation environment, ensuring that the results are relevant to real-world applications.

We measured key aspects of the scheme's performance, including key generation time, signature generation and verification time, key and signature sizes, and memory consumption. The proposed scheme utilizes a hash-and-sign strategy, where messages are first hashed into a fixed-size digest before signing. This approach ensures that the signature size remains constant regardless of the message size, improving storage and transmission efficiency. Experiments were repeated multiple times to ensure accuracy and minimize variability due to factors such as system load and background processes.

Key Generation Time:

- 100 × 100: 2.5 seconds
- 200 × 200: 5.8 seconds
- 300 × 300: 10.2 seconds

Signatures were generated and verified for messages of sizes 1 KB, 10 KB, and 100 KB using fixed keys.

Signature Generation:

- 1 KB: 0.20 milliseconds
- 10 KB: 1.15 milliseconds
- 100 KB: 9.30 milliseconds

Signature Verification:

- 1 KB: 0.18 milliseconds
- 10 KB: 1.10 milliseconds
- 100 KB: 8.90 milliseconds

Key and Signature Sizes:

- Public Key: 4 KB
- Private Key: 2 KB
- Signature Size (All Message Sizes): 1.7 KB

Memory Consumption:

- Key Generation: 50 MB
- Signature Generation: 12 MB
- Signature Verification: 12 MB

The use of the hash-and-sign strategy ensures that the signature size remains consistent across different message sizes, reducing the impact on storage and transmission efficiency. However, the experimental results highlight the trade-off between security and efficiency. Increasing matrix dimensions and key sizes enhances security but leads to higher computational and memory costs. For example, increasing the matrix size from 100 × 100 to 300 × 300 results in a fourfold increase in key generation time (from 2.5 seconds to 10.2 seconds).

While larger keys improve resistance to attacks, they also increase memory consumption and processing time for both signature generation and verification. This balance between stronger security and greater resource demands emphasizes the need for careful parameter selection to optimize performance for specific application requirements.

As part of future work, we plan to expand our experiments to include:

- Secure IoT Communications: Testing the scheme on resource-constrained devices (e.g., Raspberry Pi, Arduino) to evaluate performance in IoT environments.

- Blockchain Integration: Implementing the scheme for transaction authentication in blockchain networks and comparing it with existing post-quantum signatures.
- TLS Protocol Enhancement: Exploring integration with TLS to assess handshake latency, security, and performance compared to RSA and ECDSA.

It is important to note that the performance of the SIS-based signature scheme is not significantly impacted by large-scale message sizes, as the message is first hashed before signing. In our approach, the system only signs the fixed-size hash of the message, meaning that the signature generation and verification times remain consistent regardless of the size of the original message. This design choice ensures that the signature performance is largely independent of the message size, addressing concerns about large-scale messages. Therefore, the experimental results provided in the paper are representative of real-world scenarios, as the time to sign a message is determined by the fixed-size hash rather than the message length itself. Nonetheless, we recognize the importance of evaluating system performance under varying loads, which we will explore further in future work.

5.2. Performance Comparison with Industry Standards

To evaluate the practicality of the proposed SIS-based digital signature scheme, we compare its performance metrics to industry standards and existing post-quantum signature schemes, including CRYSTALS-Dilithium, Falcon, LWE-based schemes, and Ring-LWE schemes. Below, we provide a detailed analysis of the results.

- Key Generation Time: The proposed SIS-based scheme has a key generation time of 2.5–10.2 seconds, depending on the matrix dimensions. This is comparable to LWE-based schemes (3.0–12.0 seconds) but slower than CRYSTALS-Dilithium (0.5–2.0 seconds) and Falcon (1.0–3.0 seconds). The slower key generation time is due to the computational complexity of generating random matrices and vectors in the SIS-based scheme.
- Signature Generation and Verification Time: The proposed scheme has a signature generation time of 0.20–9.30 milliseconds and a verification time of 0.18–8.90 milliseconds. These times are comparable to LWE-based schemes (0.30–10.00 milliseconds for generation and 0.25–9.50 milliseconds for verification) but slower than CRYSTALS-Dilithium (0.10–1.50 milliseconds for generation and 0.10–1.20 milliseconds for verification) and Falcon (0.15–2.00 milliseconds for generation and 0.15–1.80 milliseconds for verification). The slower times are due to the matrix operations required in the SIS-based scheme.
- Key and Signature Sizes: The proposed scheme has a key size of 4 KB and a signature size of 1.7 KB. These sizes are larger than CRYSTALS-Dilithium (2 KB key size and 1.3 KB signature size) and Falcon (1.5 KB key size and 1.0 KB signature size) but comparable to LWE-based schemes (5 KB key size and 2.0 KB signature size). The larger sizes are due to the matrix dimensions used in the SIS-based scheme.
- Memory Consumption: The proposed scheme has a memory consumption of 50 MB during key generation and 12 MB during signature generation and verification. This is higher than CRYSTALS-Dilithium (30 MB for key generation and 8 MB for signature generation/verification) and Falcon (25 MB for key generation and 6 MB for signature generation/verification) but comparable to LWE-based schemes (60 MB for key generation and 15 MB for signature generation/verification).

The higher memory consumption is due to the need to store large matrices and vectors in the SIS-based scheme. Table 1 summarizes the performance metrics of the proposed SIS-based scheme and other post-quantum schemes.

Table 1. Performance comparison with industry standards and existing schemes

Scheme	Key Generation Time	Signature Generation Time	Verification Time	Key Size	Signature Size	Memory Consumption
Proposed SIS-Based	2.5–10.2 seconds	0.20–9.30 milliseconds	0.18–8.90 milliseconds	4 KB	1.7 KB	50 MB (key gen), 12 MB (sig gen/ver)
CRYSTALS-Dilithium	0.5–2.0 seconds	0.10–1.50 milliseconds	0.10–1.20 milliseconds	2 KB	1.3 KB	30 MB (key gen), 8 MB (sig gen/ver)
Falcon	1.0–3.0 seconds	0.15–2.00 milliseconds	0.15–1.80 milliseconds	1.5 KB	1.0 KB	25 MB (key gen), 6 MB (sig gen/ver)
LWE-Based Scheme	3.0–12.0 seconds	0.30–10.00 milliseconds	0.25–9.50 milliseconds	5 KB	2.0 KB	60 MB (key gen), 15 MB (sig gen/ver)
Ring-LWE Scheme	1.5–5.0 seconds	0.20–3.00 milliseconds	0.20–2.50 milliseconds	2.5 KB	1.5 KB	40 MB (key gen), 10 MB (sig gen/ver)

To make the trade-offs between our proposed scheme and existing alternatives more accessible, Table 2 summarizes key performance and security characteristics across signature schemes, including CRYSTALS-Dilithium, Falcon, and classical ElGamal.

Table 2. Summary comparison of digital signature schemes

Scheme	Key Size	Signature Size	Key Gen Time	Sign Time	Verify Time	Security Assumption	Notes
SIS-ElGamal (proposed)	4 KB	1.7 KB	2.5–10.2 s	0.20–9.3 ms	0.18–8.9 ms	SIS (lattice-based)	Quantum-safe, hash-and-sign
CRYSTALS-Dilithium	2 KB	1.3 KB	0.5–2.0 s	0.10–1.5 ms	0.10–1.2 ms	Module-LWE	NIST finalist
Falcon	1.5 KB	1.0 KB	1.0–3.0 s	0.15–2.0 ms	0.15–1.8 ms	NTRU lattice	Fast, compact, complex to implement
LWE-Based Scheme	5 KB	2.0 KB	3.0–12.0 s	0.30–10.0 ms	0.25–9.5 ms	LWE	Strong, but larger sizes
Ring-LWE Scheme	2.5 KB	1.5 KB	1.5–5.0 s	0.20–3.0 ms	0.20–2.5 ms	Ring-LWE	Efficient for constrained devices

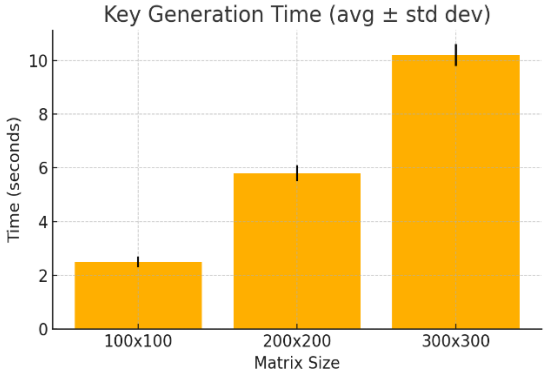

Fig.3. Average key generation time (± standard deviation) across increasing matrix dimensions. As expected, key generation time scales linearly with matrix size due to the increased complexity of lattice constructions

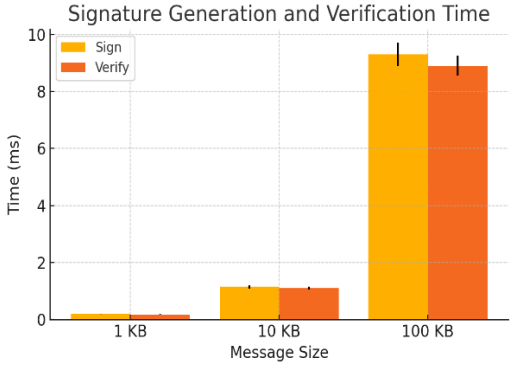

Fig.4. Signature generation and verification times for varying message sizes. Despite increasing input message sizes, computation time remains bounded due to the fixed-size hash-and-sign strategy

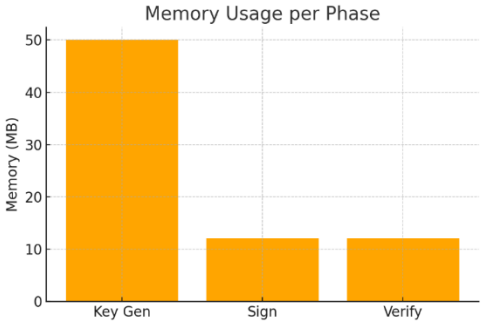

Fig.5. Peak memory usage during each phase of the digital signature lifecycle. Key generation is the most memory-intensive, while signing and verification remain lightweight and consistent

5.3. Visual Benchmark Results

To better illustrate performance trends and reliability across different configurations, we provide visual summaries of key performance metrics. All results are averaged over 10 runs, with error bars indicating standard deviation. All experiments were conducted on the same testing platform described earlier, ensuring fair cross-comparison. Please check Figure3,4, and 5.

6. Comparison with other Schemes

6.1. Comparison with ElGamal Encryption Scheme

This section offers an efficient comparison of ElGamal encryption and the proposed novel scheme.
For the key generation we have the following:

- ElGamal Digital Signature:

Time Complexity: Key generation involves selecting large prime numbers and performing modular arithmetic operations. This process is generally efficient, with time complexity typically dependent on the bit-length of the primes used. Key Generation is usually performed in logarithmic time relative to the size of the primes.

Key Size: Keys are relatively large, as they consist of prime numbers and their corresponding elements. The size scales with the security parameter, but storage and transmission are manageable due to the established size constraints.

- SIS - Based Digital Signature:

Time Complexity: Key generation involves generating random matrices and vectors over a finite field, which can be computationally intensive. The time complexity is influenced by the dimensions of the matrices and the norm of the vectors. For large dimensions, this can become more time-consuming compared to modular arithmetic.

Key Size: Key sizes may be larger than with Elgamal because they are proportionate to the field size and matrix dimensions. Transmission and storage may be impacted, especially for high-security settings.

For signature generation we have the following:

- ElGamal Digital Signature:

Time Complexity: Signature generation requires performing several modular exponentiations. While modular exponentiation is efficient, it can become computationally intensive with larger key sizes. Overall, the time complexity is generally logarithmic with respect to the size of the numbers involved.

Signature Size: The size of the signature is relatively small and scales with the security parameter. This efficiency in signature size contributes to faster verification times.

- SIS- Based Digital Signature:

Time Complexity: Signature generation involves solving a system of linear equations, which can be computationally intensive, especially for larger matrix dimensions. The time complexity is influenced by the matrix dimensions and the required precision.

Signature Size: Signatures in the SIS-based scheme can be larger due to the dimensionality of the matrices and vectors involved. This potentially impacts storage and transmission efficiency.

For signature verification we have the following:

- ElGamal Digital Signature:

Time Complexity: Verification involves checking the correctness of a signature using modular exponentiation. This operation is efficient and generally fast, with time complexity that scales logarithmically with the size of the prime numbers.

Verification Size: Verification is efficient and scales well with signature size, making it suitable for scenarios where rapid verification is required.

- SIS - Based Digital Signature:

Time Complexity: Verification involves matrix multiplication and checking if the solution satisfies the given equations. The time complexity can be efficient for small matrix dimensions but may become more demanding with larger matrices.

Verification Size: Verification efficiency is affected by the size of the matrices and vectors used in the signature, which can impact performance when dealing with larger dimensions.

Because it relies on modular arithmetic and uses lower signature sizes, the Elgamal digital signature scheme tends to offer higher efficiency in terms of key generation, signature creation, and verification timeframes. In contrast, the SIS-based digital signature scheme, while offering strong security properties, can be less efficient due to the computational complexity of solving linear equations and handling larger key and signature sizes.

In the future works, we think using techniques such as compact representations of lattice problems and the adoption of hybrid cryptographic approaches can help reduce the signature size. These approaches aim to balance the quantum-resistant nature of SIS with practical storage and bandwidth requirements.

6.2. Comparison with LWE and Ring-LWE Signature Schemes

While the proposed SIS-based ElGamal scheme improves security over traditional ElGamal, it is important to compare it with other post-quantum signature schemes like LWE and Ring-LWE. Below, we provide a detailed analysis of the strengths and weaknesses of each scheme.

All three schemes, SIS, LWE and Ring-LWE, offer strong security based on lattice problems, providing resistance to quantum attacks. The SIS problem is closely related to the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP), which are believed to be hard even for quantum computers. Similarly, LWE and Ring-LWE are based on the hardness of solving linear equations with noise, which is also believed to be quantum-resistant. However, the security assumptions differ slightly: SIS relies on finding short vectors in a lattice, while LWE and Ring-LWE involve solving noisy linear equations. This difference affects the practical security and implementation of each scheme.

The SIS-based scheme generates larger keys and signatures compared to Ring-LWE but is comparable in size to LWE. Ring-LWE achieves smaller key and signature sizes due to its efficient algebraic structure, which compresses data more effectively. In terms of computational complexity, Ring-LWE is generally more efficient in key generation and signature operations because it uses faster polynomial arithmetic. In contrast, SIS and LWE rely on matrix operations, which are more computationally demanding. For example, key generation in the SIS-based scheme involves generating random matrices and vectors, which can be time-consuming for large dimensions. In contrast, Ring-LWE uses polynomial multiplication, which is more efficient.

The SIS-based scheme offers simpler and more straightforward implementation compared to LWE schemes, as it does not require noise sampling. However, it is less efficient than Ring-LWE in resource-constrained environments. LWE schemes, while offering similar security, often require larger keys and more complex noise handling, making them less practical for some applications. Ring-LWE, on the other hand, strikes a balance between security and efficiency, making it suitable for a wide range of applications, including IoT devices and blockchain networks.

The SIS-based scheme is particularly well-suited for high-security environments, such as governmental and military applications, where the need for advanced cryptographic security outweighs the practical considerations of key and signature size. In contrast, Ring-LWE is more suitable for resource-constrained environments, such as IoT devices, where efficiency and smaller key sizes are critical. LWE schemes, while offering strong security, are often used in general-purpose post-quantum applications where a balance between security and efficiency is required. Table 3 summarizes the key differences between the proposed SIS-based scheme, LWE, and Ring-LWE.

Table 3. Comparison of SIS-Based, LWE, and Ring-LWE schemes

Aspect	SIS-Based Scheme	LWE	Ring-LWE
Security Assumption	Hardness of SIS problem	Hardness of LWE problem	Hardness of Ring-LWE problem
Key Size	Larger keys due to matrix dimensions	Large keys, similar to SIS	Smaller keys due to ring structure
Signature Size	Larger signatures	Large signatures, similar to SIS	Smaller signatures
Computational Complexity	High due to matrix operations	High due to noise sampling	Lower due to polynomial arithmetic
Implementation Complexity	Simpler, no noise sampling	Complex due to noise handling	Moderate, efficient polynomial ops
Best Use Case	High-security environments (e.g., government, military)	General-purpose post-quantum applications	Resource-constrained environments (e.g., IoT)

7. Real-World Applications

The proposed SIS-based ElGamal digital signature scheme is designed to provide robust security against quantum attacks, making it suitable for a wide range of real-world applications. In governmental and military contexts, where the need for advanced cryptographic security outweighs the practical considerations of key and signature size, the scheme could be used to secure classified communications, authenticate digital orders such as military directives or diplomatic communications, and sign critical documents like treaties or legal agreements, ensuring their integrity and authenticity. In the financial sector, the scheme could be applied to secure digital transactions, authenticate financial documents such as contracts or audit reports, and protect sensitive customer data like account information or transaction histories. Additionally, it could be integrated into blockchain networks to provide quantum-resistant transaction authentication,

ensuring the long-term security of financial records.

Critical infrastructure systems, such as power grids, water treatment facilities, and transportation networks, also require robust cryptographic solutions to protect against cyber threats.

The proposed scheme could be used to secure communication channels, authenticate control commands, and ensure the integrity of sensor data, preventing tampering or unauthorized modifications. Furthermore, the scheme can be integrated into existing public key infrastructures (PKIs) with minimal changes, providing enhanced security without compromising the operational efficiency of current systems. For example, it could replace traditional digital signature schemes in PKIs, ensuring that digital signatures remain secure against quantum attacks, or enhance secure communication protocols like TLS to provide quantum-resistant authentication and data integrity. It could also be integrated into blockchain networks, providing quantum-resistant transaction authentication and ensuring the long-term security of financial records.

These applications demonstrate the practical relevance and potential impact of the proposed solution in high-security environments.

In addition to the broadly mentioned governmental and military applications, the proposed SIS-based ElGamal signature scheme can be directly integrated into several practical systems that demand quantum-resistant security. For instance:

Secure Boot in Embedded Systems:

In critical devices such as embedded controllers and industrial IoT gateways, a secure boot process ensures that only authenticated firmware and system code is executed at startup. By incorporating our signature scheme, the bootloader can verify the digital signature of firmware updates. Since the signature size is constant (1.7 KB) and the arithmetic operations are fully integer-based, the scheme can be optimized for the limited computational resources typical of embedded systems, enhancing the security of the overall boot process against quantum threats.

IoT Devices Requiring Quantum-Safe Signatures:

As the Internet of Things continues to expand, ensuring the authenticity and integrity of the data exchanged between devices becomes paramount. Our scheme's quantum-resistant properties make it an excellent candidate for signing device communications and firmware updates within IoT networks. Here, the scheme can be deployed to authenticate messages transmitted over potentially insecure networks and to guard against future quantum attacks without overly burdening the device's processing capabilities. Integration might involve a lightweight cryptographic module specifically optimized for our scheme, ensuring seamless compatibility with existing IoT protocols while extending the lifetime of device security.

These specific deployment scenarios illustrate how our hybrid signature scheme can be adapted for real-world applications where both high-level security and resource constraints are of concern. By leveraging constant signature sizes and efficient verification times, the scheme provides a viable pathway for integrating post-quantum protections into both embedded and IoT environments without necessitating a complete overhaul of current secure boot or communication protocols.

8. Conclusions

In this work, we proposed a novel digital signature scheme that combines the classical ElGamal framework with the lattice-based Short Integer Solution (SIS) problem, offering a new approach to achieving post-quantum cryptographic security. The core idea behind this hybrid construction is to retain the efficiency and conceptual simplicity of ElGamal while replacing its quantum-vulnerable discrete logarithm foundation with the quantum-resistant hardness of SIS. This fusion results in a scheme that is not only theoretically sound but also practically implementable using standard integer arithmetic, making it suitable for deployment in a wide range of environments. The contributions of this research are multifaceted. First, to the best of our knowledge, this is the first formal construction of a digital signature scheme that merges ElGamal's signing logic with SIS-based hardness assumptions. Unlike many existing lattice-based schemes that rely on more complex structures such as Learning with Errors (LWE), our approach benefits from the algebraic and computational simplicity of SIS. We presented a detailed construction of the key generation, signing, and verification processes and demonstrated a formal security reduction to the SIS problem, which is considered to be hard even in the presence of quantum adversaries. Experimental evaluation further supports the viability of the scheme. We assessed the signature's performance under varying matrix dimensions and message sizes, showing that both signature generation and verification times remain under 10 milliseconds, even for large messages. Key sizes and signature sizes are modest by post-quantum standards, and the use of a hash-and-sign methodology ensures fixed signature size regardless of message length. The implementation requires only integer operations, which is an important advantage for constrained devices that do not support floating-point arithmetic. Memory usage remains manageable, and the design avoids side-channel leakage vectors often found in floating-point-based lattice implementations.

While the scheme is not as fast as CRYSTALS-Dilithium or Falcon in terms of raw throughput, it provides a valuable balance between quantum resistance, simplicity, and general usability. These properties make it well-suited for deployment in high-security domains, including digital identity validation, signed software updates, and secure communications in both civilian and defense applications. Looking forward, several directions remain for future

exploration. One important next step is the deployment and benchmarking of this scheme on embedded and resource-constrained devices, such as microcontrollers or IoT hardware, where its simplicity could yield practical advantages. Another promising direction is the integration of the scheme into widely used cryptographic protocols, such as TLS, to assess compatibility, handshake latency, and overall system performance in real-time secure communications. Moreover, ongoing monitoring of the SIS problem's hardness under emerging quantum algorithms will be critical to ensuring long-term security viability. Fine-tuning parameter choices and optimizing matrix dimensions for different application contexts also remain open areas for further research. This paper contributes to the post-quantum cryptographic landscape by introducing a secure, efficient, and conceptually clean digital signature scheme that integrates classical cryptographic intuition with modern lattice-based security guarantees. We believe this hybrid approach opens a promising pathway toward practical and secure authentication systems in the quantum era.

Acknowledgments

This work was supported by the Shota Rustaveli National Science Foundation of Georgia (SRNSFG) N FR-24-15007.

References

[1] Dejpasand, Mohamad Taghi, and Morteza Sasani Ghamsari. "Research trends in quantum computers by focusing on qubits as their building blocks." Quantum Reports 5.3 (2023): 597-608.
[2] Di Meglio, Alberto, et al. "Quantum Computing for High-Energy Physics: State of the Art and Challenges." PRX Quantum 5.3 (2024): 037001.
[3] Gill, Sukhpal Singh, et al. "Quantum Computing: Vision and Challenges." arXiv preprint arXiv:2403.02240 (2024).
[4] Micciancio, Daniele, and Oded Regev. "Lattice-based cryptography." Post-quantum cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. 147-191.
[5] Regev, Oded. "Lattice-based cryptography." Annual International Cryptology Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
[6] Gagnidze, Avtandil, Maksim Iavich, and Giorgi Iashvili. "Analysis of post quantum cryptography use in practice." Bull. Georgian Natl. Acad. Sci 11.2 (2017): 29-36.
[7] Kumar, Ajay, and Sunita Garhwal. "State-of-the-art survey of quantum cryptography." Archives of Computational Methods in Engineering 28 (2021): 3831-3868.
[8] Bernstein, Daniel J., and Tanja Lange. "Post-quantum cryptography." Nature 549.7671 (2017): 188-194.
[9] Pan, Chen, Yafeng Han, and Jiping Lu. "Design and optimization of lattice structures: A review." Applied Sciences 10.18 (2020): 6374.
[10] Khot, Subhash. "Inapproximability results for computational problems on lattices." The LLL algorithm: Survey and applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. 453-473.
[11] Peikert, Chris. "A decade of lattice cryptography." Foundations and trends® in theoretical computer science 10.4 (2016): 283-424.
[12] Chen, Yilei. "Quantum algorithms for lattice problems." Cryptology ePrint Archive (2024).
[13] Buchheit, Andreas A., Torsten Keßler, and Kirill Serkh. "On the computation of lattice sums without translational invariance." arXiv preprint arXiv:2403.03213 (2024).
[14] Zhang, Jiang, Yu Chen, and Zhenfeng Zhang. "Lattice-Based Programmable Hash Functions and Applications." Journal of Cryptology 37.1 (2024): 4.
[15] Wang, Yu, et al. "Image encryption algorithm based on lattice hash function and privacy protection." Multimedia Tools and Applications 81.13 (2022): 18251-18277.
[16] Wang, Yong, Kwok-Wo Wong, and Di Xiao. "Parallel hash function construction based on coupled map lattices." Communications in Nonlinear Science and Numerical Simulation 16.7 (2011): 2810-2821.
[17] Ye, Zewen, et al. "A Highly-efficient Lattice-based Post-Quantum Cryptography Processor for IoT Applications." IACR Transactions on Cryptographic Hardware and Embedded Systems 2024.2 (2024): 130-153.
[18] Mishra, Nimish, SK Hafizul Islam, and Sherali Zeadally. "A comprehensive review on collision-resistant hash functions on lattices." Journal of Information Security and Applications 58 (2021): 102782.
[19] Iavich, Maksim, et al. "Lattice based merkle." IVUS. 2019.
[20] Mohammed, Elsayed, A-E. Emarah, and K. El-Shennaway. "A blind signature scheme based on ElGamal signature." Proceedings of the Seventeenth National Radio Science Conference. 17th NRSC'2000 (IEEE Cat. No. 00EX396). IEEE, 2000.
[21] Anusha, R., and R. Saravanan. "Enhancement of Lightweight Secure Blockchain Based Edward-El Gamal in the Internet of Things (IoT)." Wireless Personal Communications 134.3 (2024): 1421-1442.
[22] Kairi, Animesh, et al. "An Innovative Method for DNA Cryptography-Based Digital Signature Verification." International Conference on Cyber Intelligence and Information Retrieval. Singapore: Springer Nature Singapore, 2023.

Authors' Profiles

Maksim Iavich, with a Ph.D. in mathematics, is a mathematician and computer science professor. In addition to being an affiliate professor at Caucasus University, where he oversees the cybersecurity department, he is the president and chief executive officer of the Scientific Cyber Security Association (SCSA). He serves as a cybersecurity consultant for firms in Georgia and abroad. He has won multiple cybersecurity science prizes in recognition of his contributions to the sector. He is also the author of many academic papers on subjects including post-quantum cryptography, cybersecurity, and cryptography.

Dana Amirkhanova Sairangazhykyzy, she is currently PhD student's 3'd course of Satbayev University, Kazakhstan, Almaty, received a bachelor's degree in computer science from IT University and a master's degree in pedagogical informatics from Abay Pedagogical University. Her research interest includes various cryptographic system security techniques.

How to cite this paper: Maksim Iavich, Dana Amirkhanova Sairangazhykyzy, "Post-quantum Digital Signatures using ElGamal Approach", International Journal of Computer Network and Information Security(IJCNIS), Vol.17, No.3, pp.89-108, 2025. DOI:10.5815/ijcnis.2025.03.06