SISSRP: A Post-Quantum Augmented PAKE Based on the Short Integer Solution Problem

Sergei Simonovi[1], Giorgi Iashvili[2], Avtandil Gagnidze[3] and Malkhaz Bibiluri[4]

[1] Caucasus University, Paata Saakadze street 1, Tbilisi, 0102, Georgia
[2] Caucasus University, Paata Saakadze street 1, Tbilisi, 0102, Georgia
[3] East-West University, Vakhtang Gorgasali street 9, Tbilisi, 0105, Georgia
[4] Georgian Technical University, Merab Kostava street 77, Tbilisi, 0108, Georgia

Abstract

Most of modern cryptographic algorithms like RSA, Diffie-Hellman KE, Elgamal, SRP are in danger due to the rapid development of quantum computers, which are capable of solving underlying mathematical problems in a reasonable amount of time. The goal of the whitepaper is to offer a post-quantum secure PAKE scheme based on SRP, which is not prone to quantum attacks due to the usage of mathematical problems which are complex to solve with a quantum and classic computer.

Keywords

Post-Quantum cryptography, PAKE, SRP, Lattice based cryptography.

1. Introduction

Modern public-key cryptography relies on a small number of computational hardness assumptions, most notably the integer factorization problem and the discrete logarithm problem. For decades, these problems have provided the mathematical foundation for commonly used cryptographic primitives such as RSA, Diffie–Hellman key exchange, elliptic curve cryptography (ECC), and numerous digital signature schemes. The security of such schemes relies on complexity of underlying mathematical problems and difficulty of solving these problems using classical computation for sufficiently large parameters. However, the rapid theoretical and experimental progress in quantum computing challenges current cryptographic paradigm. In 1994, Peter Shor demonstrated that a sufficiently powerful quantum computer can solve both integer factorization and discrete logarithms in polynomial time. Shor's algorithm transforms these problems into instances of period finding, which can be efficiently solved using the Quantum Fourier Transform. Although current quantum hardware does not yet possess the scale and error-correction capabilities necessary to threaten large production cryptosystems, advancements in qubit stability, fault tolerance, and architecture design are made. As a result, cryptographic systems deployed today may become retrospectively vulnerable once large-scale quantum computers emerge. One of quantum computer related risks commonly referred to as "harvest now, decrypt later".

Password-Authenticated Key Exchange (PAKE) protocols, including the Secure Remote Password (SRP) protocol, rely on discrete logarithm–based constructions for authenticated key establishment. Once these algebraic assumptions are broken by quantum adversaries, PAKE schemes built upon them lose their security guarantees. Given that PAKE protocols are widely used in authentication systems where human-memorable passwords remain the main credential form, the quantum threat directly impacts real-world authentication infrastructures. PAKE protocols occupy a particularly important niche in applied cryptography. They enable two parties sharing only a low-entropy secret (a password) to establish a high-entropy session key while preventing offline dictionary attacks. Unlike naive password-based authentication mechanisms, properly designed PAKE protocols ensure that observing protocol transcripts does not allow an adversary to verify password guesses without interacting with a legitimate participant. Also, augmented PAKE variants, such as SRP, are resilient against server compromise, as they store only a verifier derived from the password, instead of the

*IVUS 2026: Information Society and University Studies 2026, May 12, Kaunas, Lithuania

s_simonovi@cu.edu.ge (S. Simonovi); giiashvili@cu.edu.ge (G. Iashvili); gagnidzeavto@yahoo.com (A. Gagnidze); bibiluri@gtu.ge (M. Bibiluri)

Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-06

CEUR-WS.org/Vol-4317/paper26.pdf

password itself. As adversary cannot impersonate clients without performing active attacks, the design of augmented PAKE limits the damage caused by data breaches.

The main advantages of PAKE protocols are resistance against offline brute-force attacks by embedding password validation inside cryptographic key exchange rather than exposing reusable authentication material. Also, PAKE protocols offer mutual authentication, ensuring that both client and server confirm knowledge of the shared secret. They also provide forward secrecy, preventing compromise of secrets caused by revealing previously established session keys. These properties make PAKE an attractive primitive in untrusted network environments where credential exposure can have severe operational and financial consequences.

Due to the quantum threat, there is a need to re-evaluate PAKE constructions and adapt them to post-quantum assumptions. The objective of this paper is to propose a post-quantum adaptation of the Secure Remote Password protocol that preserves its augmented PAKE structure while eliminating reliance on discrete logarithm–based group operations. Specifically, we replace the classical Diffie–Hellman exponentiation framework with a lattice-based construction grounded in the Short Integer Solution (SIS) problem. Lattice-based assumptions are widely regarded as strong candidates for post-quantum cryptography because no efficient quantum algorithms are known for solving the underlying problems in polynomial time, and their security can be related to worst-case hardness results in lattice theory.

The proposed construction aims to maintain the essential security invariants of SRP: resistance to offline dictionary attacks, mutual authentication, verifier-based server storage, and session key indistinguishability. At the same time, it removes dependence on algebraic group structures vulnerable to Shor's algorithm. The transformation preserves the augmented PAKE properties but substitutes its vulnerable mathematical foundation with a post-quantum secure problem, based on high-dimensional lattice spaces. By modernizing SRP, we provide a structurally conservative password-based authentication protocol that can be used in post-quantum world. The resulting protocol can be deployed in environments where long-term confidentiality, resistance to credential compromise, and quantum resilience are required.

2. PAKE Algorithms

Password-Authenticated Key Exchange (PAKE) is a class of cryptographic protocols that enable two parties who share only a low-entropy secret, like password, to establish a high-entropy cryptographic session key. PAKE protocols are also protected against passive eavesdroppers and active adversaries. A naive approach to password authentication permits offline dictionary attacks; PAKE protocols are designed precisely to prevent this type of attacks. The modern study of PAKE began in the early 1990s. The work of Steven Bellovin and Michael Merritt introduced Encrypted Key Exchange (EKE) in 1992 [1], which combined Diffie–Hellman key exchange [2] with password-based encryption to prevent adversaries from testing password guesses offline. Around the same time, Bellovin and Merritt also proposed Augmented EKE (A-EKE) [3]. Shortly thereafter, Victor Boyko, Philip MacKenzie, and Sarvar Patel formalized the security of EKE-like constructions in the random oracle model. In late 1990s and early 2000s, Mihir Bellare, David Pointcheval, and Phillip Rogaway introduced influential formalizations of authenticated key exchange, while Katz, Ostrovsky, and Yung provided the first provably secure PAKE under standard assumptions without idealized random oracles.

PAKE protocols consist of two broad families: balanced and augmented. In a balanced PAKE, both parties store the same password. Each side proves knowledge of the shared secret during the exchange, and security relies on the protocol structure preventing offline verification of password guesses. Examples include EKE, SPEKE (Simple Password Exponential Key Exchange) [4], and J-PAKE (Password Authenticated Key Exchange by Juggling). In an augmented PAKE, the server stores not the password itself but a verifier derived from it. Even if the server database is compromised, the adversary cannot impersonate the client without performing an online attack against the server. SRP, the Secure Remote Password protocol designed by Thomas Wu [5] in the mid-1990s, is the canonical augmented PAKE; it allows the server to store a verifier computed from the password, while ensuring that theft of this verifier does not immediately enable client impersonation.

PAKE protocols eliminate offline dictionary attacks: an adversary observing the protocol transcript cannot test password guesses without interacting with a legitimate party. Also, they provide mutual authentication: both client and server are assured of the other's knowledge of the password. PAKE protocols yield a high-entropy session key derived from fresh randomness and Diffie–Hellman–like computations, ensuring forward secrecy in many constructions. It should be emphasized that augmented variants protect against server compromise by avoiding storage of plaintext passwords or directly reusable password hashes. Well-designed PAKE protocols offer resistance to man-in-the-middle attacks, since any tampering invalidates the password-based authentication checks.

One of the important developments in a field Is SPEKE [4], that was introduced by David Jablon in 1996. SPEKE replaced encrypted Diffie–Hellman values with password-derived generators. SRP protocol, on the other side, refined the augmented model and was later used in TLS ciphersuites. J-PAKE, proposed by Feng Hao and Peter Ryan, used zero-knowledge proofs to ensure that Diffie–Hellman exponents were well formed. SPAKE2, developed by Abdalla and Pointcheval among others, was simple and efficient protocol with clean security proofs. The Dragonfly protocol, later standardized as part of WPA3's Simultaneous Authentication of Equals (SAE), is another balanced PAKE derived from SPEKE principles. More recently, OPAQUE, designed by Jarecki, Krawczyk, and Xu [6], represents a modern augmented PAKE with strong security proofs in the Universal Composability framework and is being standardized within the IETF. OPAQUE separates password authentication from key exchange using Oblivious Pseudorandom Function, resulting in strong protection even in case of a server compromise.

IETF has published RFCs for SPAKE2 and OPAQUE, and PAKE-based ciphersuites have appeared in TLS 1.3 extensions. The theoretical literature distinguishes between balanced PAKE, augmented PAKE (aPAKE), verifier-based PAKE, and asymmetric PAKE variants. There are also threshold PAKE constructions, where password verification is distributed across multiple servers, and post-quantum PAKE proposals that replace classical Diffie–Hellman with lattice-based or isogeny-based primitives.

In summary, PAKE protocols elevate a low-entropy password into a cryptographically strong session key without exposing it to offline guessing.

3. SRP Algorithm

The Secure Remote Password (SRP) protocol constitutes an augmented password-authenticated key exchange (aPAKE) mechanism, originally introduced by Thomas Wu in 1998 and formalized in RFC 2945 (version 3), with subsequent refinements culminating in SRP-6a as the currently most widely referenced variant. We work in a large cyclic group G of prime order q, usually instantiated as a subgroup of $\mathbb{Z}_N^*$ where N is a large safe prime and g is a generator. Hash function $H(\cdot)$ is modeled as a cryptographic hash (often SHA-512 in modern deployments). SRP algorithm consists of several steps. The first step is setup and registration. During this step, public parameters N, g and H are chosen. N is a large prime number, g is a generator modulo N and H is a hash function. The user chooses identity I and a password P. The server generates a random salt s. Then, a private password hash is calculated:

$x = H(s \| H(I \| P))$. x is also called a password exponent.

Then, a verifier is computed:

$$v = g^x \mod N$$

The server stores (I, s, v). Crucially, the server does not store P, nor x. Only v. This is why SRP is called an augmented PAKE.

The next phase is called authentication phase. A client chooses a random ephemeral secret

$$a \xleftarrow{\$} \mathbb{Z}_q$$

and computes

$$A = g^a \mod N$$

Client sends to the server (I, A). A server retrieves s, v and chooses it's own ephemeral secret

$$b \xleftarrow{\$} \mathbb{Z}_q$$

Then, a server computes $B = kv + g^b \mod N$ where $k = H(N \| g)$. A server sends (s, B). During the next step, both parties compute $u = H(A \| B)$. This value "mixes" both ephemeral contributions. It prevents an attacker from precomputing tables or manipulating one side independently. The next step in SRP algorithm is a computation of a session key on a client computer. A client computes

$$x = H(s \| H(I \| P))$$

After, it computes a premaster key

$$S_c = (B - kg^x)^{(a+ux)} \mod N.$$

$$\text{As } B = kv + g^b = kg^x + g^b \Rightarrow B - kg^x = g^b \Rightarrow S_c = \left(g^b\right)^{a+ux} \Rightarrow g^{b(a+ux)}$$

During the next step, a server computes a session key.

$$S_s = (Av^u)^b \mod N.$$

$$A = g^a, v = g^x \Rightarrow Av^u = g^a(g^x)^u = g^{a+ux}$$

$$S_s = (g^{a+ux})^b = g^{b(a+ux)} \Rightarrow S_c = S_s = S.$$

At this moment, both parties share the same premaster secret. The shared key is $K = H(S)$. To prevent unknown key-share attacks and confirm both sides derived the same key, SRP includes explicit authentication. Client computes

$$M_1 = H(H(N) \oplus H(g) \| H(I) \| s \| A \| B \| K)$$

and server verifies it. A server responds with

$$M_2 = H(A \| M_1 \| K)$$

A client verifies M_2.

It should be emphasized that SRP completely relies on a discrete logarithm problem, which can be solved with a quantum computer in a reasonable amount of time.

4. Quantum attacks on a discrete logarithm problem

Shor's Algorithm is a quantum algorithm developed by mathematician Peter Shor in 1994 [7]. It is designed to solve two specific problems efficiently: integer factorization (finding the prime factors of a number) and discrete logarithms. Both of these problems form the foundation of many widely used cryptographic systems, such as RSA and elliptic curve cryptography (ECC). Classical computers take an impractical amount of time to solve these problems for large numbers, which is why they are considered secure. However, Shor's Algorithm makes it possible for quantum computers to solve them exponentially faster, making these cryptographic systems vulnerable. According to Shor's Algorithm, quantum computers can exploit the mathematical structure of these problems using quantum properties like superposition and entanglement. The algorithm is divided into two main parts. First, it transforms the factorization problem into a periodicity problem, which involves finding the period of a specific function. Then, it uses a quantum subroutine called the Quantum Fourier Transform (QFT) [8] to find this period efficiently. Once the period is identified, classical computations can use it to determine the factors of the given number. The most significant implication of Shor's Algorithm is its ability to break modern encryption. For instance, RSA encryption relies on the difficulty of factoring large numbers, and ECC relies on solving discrete

logarithms. Both are problems that Shor's Algorithm can solve. This means that if a sufficiently powerful quantum computer is developed, it could decrypt data protected by these encryption methods, compromising secure communications, financial transactions, and sensitive information worldwide. However, implementing Shor's Algorithm in practice requires a quantum computer with a large number of qubits and extremely low error rates. Current quantum computers are far from meeting these requirements, but advancements are being made. For example, estimates suggest that breaking a 2048-bit RSA encryption key would require around 4,000 qubits and sufficient error correction, which is not yet feasible with today's quantum technology.

5. Lattice based cryptography, SIS

Lattice-based cryptography [9] is a class of cryptographic constructions that rely on the assumed hardness of certain problems in lattice theory, rather than on factoring integers or solving discrete logarithms. A lattice, in a mathematical sense, is a regular grid of points in multidimensional space generated by linear combinations of basis vectors with integer coefficients. Computational problems related to lattices can be extremely difficult to solve, especially in high dimensions. Because no efficient quantum algorithm is known for solving the main hard lattice problems, lattice-based cryptography is considered one of the leading approaches to post-quantum cryptography. One of the central problems in this area is the Short Integer Solution problem, commonly abbreviated as SIS. The Short Integer Solution problem can be described in linear algebraic terms. Given a large random matrix with entries taken modulo an integer q, the task is to find a nonzero integer vector with relatively small coefficients such that, when multiplied by the matrix, the result is the zero vector modulo q. In other words, we are searching for a short vector that lies in the kernel of the matrix modulo q. The difficulty arises from the requirement that the solution must be short. Without that restriction, many solutions exist and are easy to describe. The constraint on smallness makes the problem computationally hard. The security intuition behind SIS is geometric. The set of all integer solutions to the modular equation forms a lattice. Finding any vector in this lattice is straightforward, but finding one that is unusually short corresponds to solving a difficult geometric optimization problem in high dimensions. This is closely related to classical lattice problems such as finding the shortest vector in a lattice. These problems are believed to be hard even for quantum computers, and their worst-case hardness can be formally related to the average-case hardness of SIS. This worst-case to average-case reduction is one of the most important theoretical strengths of lattice-based cryptography, because it ties the security of practical constructions to well-studied mathematical problems. Lattice-based cryptography more broadly includes encryption and key exchange mechanisms built on related problems, such as learning with errors, which can be seen as a noisy variant of linear equations over lattices. Together, these constructions provide analogues of classical public-key systems but rely on fundamentally different mathematical assumptions. Unlike systems based on factoring or discrete logarithms, there is currently no known quantum algorithm that solves SIS or related lattice problems in polynomial time.

6. SIS based key exchange

SIS problem can be used in a key exchange algorithm [10]. Having two parties A and B, let's assume A and B want to establish shared secret K. The following algorithm can be used to achieve this:

1. A random matrix $\mathrm{R} \in \mathbb{Z}_q^{\mathrm{nxm}}$ is generated.
2. A picks a secret $\mathrm{s_A} \in Z_q^m$ with a norm $\|\mathrm{s_A}\| \leq \beta$. Then, $\mathrm{P}_A = \mathrm{Rs_A}$ is calculated and sent to B
3. B picks a secret $\mathrm{s_B} \in Z_q^n$ with norm $\|\mathrm{s_B}\| \leq \beta$. Then, $\mathrm{P}_B = s_B^T\mathrm{R}$ is calculated and sent to A
4. Getting a P_b, A calculates $\mathrm{K}_A = s_A^T P_B^T = s_A^T \mathrm{R}^\top \mathrm{S_B}$
5. Getting a P_A, B calculates $\mathrm{K}_B = P_A^T\ \mathrm{S_B} = s_A^T \mathrm{R}^\top \mathrm{S_B}$
6. $\mathrm{K}_A = \mathrm{K}_B = \mathrm{K}$

This approach can be used instead of a traditional Diffie-Hellman key exchange to eliminate discrete logarithm related vulnerabilities.

7. SISSRP

The SISSRP protocol represents a post-quantum adaptation of the classical Secure Remote Password (SRP) protocol in which the discrete logarithm–based Diffie–Hellman key exchange is replaced with a lattice-based construction grounded in the Short Integer Solution (SIS) problem. The objective is to preserve the augmented PAKE structure of SRP while eliminating reliance on algebraic groups vulnerable to Shor's algorithm.

7.1 Parameters

Let:

- q be a large modulus,
- n be a lattice dimension parameter,
- $A \in \mathbb{Z}_q^{n \times n}$ be a uniformly random public matrix,
- $H(\cdot)$ be a cryptographic hash function,
- vectors are sampled from a bounded norm distribution $\| \cdot \| \leq \beta$.

The hardness assumption underlying the protocol is the SIS problem: given A, it is computationally infeasible to find a nonzero short vector z such that

$$Az \equiv 0 \pmod{q}$$

unless specific trapdoor information is known.

7.2 Registration

The client selects:

- identity I,
- password P.

The server generates a random salt s.

The client computes a password-derived vector:

$$x = \text{MapToShort}(H(s \| H(I \| P))) \in \mathbb{Z}_{\square}^{n}$$

where MapToShort($\cdot$) deterministically maps the hash output to a short integer vector in $\mathbb{Z}^n$.

The verifier is computed as:

$$v = Ax \pmod{q} \in \mathbb{Z}_q^n$$

The server stores:

$$(I, s, v)$$

Importantly, the server does not store P nor x. As in classical SRP, this preserves the augmented PAKE property.

7.3 Authentication

Step 1: Client ⟶ Server

The client samples a short ephemeral vector:

$$a \leftarrow \chi \text{ where } a \in \mathbb{Z}^n, \|a\| \leq \beta$$

where χ is a distribution bounded by β.

The client calculates:

$$T_c = Aa \pmod q \in \mathbb{Z}_q^n$$

The client sends:

$$(I, T_c)$$

Step 2: Server ⟶ Client

The server retrieves s, v.

The server samples its own short ephemeral vector:

$$b \leftarrow \chi \text{ where } b \in \mathbb{Z}^n, \|b\| \leq \beta$$

It calculates:

$$T_s = Ab + kv \pmod q \in \mathbb{Z}_q^n$$

where

$$k = H(A \| T_c) \in \mathbb{Z}q$$

The server sends:

$$(s, T_s)$$

7.4 Scrambling parameter

Both parties compute:

$$u = H(T_c \| T_s) \in \mathbb{Z}q$$

The parameter binds the session values and prevents replay or precomputation attacks.

7.5 Shared secret derivation

The client recomputes:

$$x = \text{MapToShort}(H(s \| H(I \| P))) \in \mathbb{Z}_q^n$$

The client calculates:

$$S_c = T_s - kv \pmod{q} \in \mathbb{Z}_q^n$$

Since:

$$T_s = Ab + kv$$
$$v = Ax$$

we obtain:

$$S_c = Ab \text{ (mod q)}$$

The client then derives:

$$K_c = a^T \cdot S_c = a^T \cdot (Ab) = a^T Ab (\text{mod } q)$$

The server computes:

$$K_s = b^T \cdot T_c = b^T \cdot (Aa) = b^T Aa (\text{mod } q)$$

$$b^T Aa = (b^T Aa)^T = a^T A^T b = a^T Ab$$

thus:

$$K_c = K_s = K' \in \mathbb{Z}_q$$

The final session key is:

$$K = H(K')$$

7.6 Mutual authentication

To make sure that both parties derived the same key and prevent unknown key-share attacks, the protocol includes explicit authentication.

The client computes:

$$M_1 = H(T_c \| T_s \| K)$$

The server verifies M_1, then responds with:

$$M_2 = H(T_c \| M_1 \| K)$$

The client verifies M_2. If the verification is successful, mutual authentication is achieved.

7.7 Comparison with a classical SRP

Classical SRP embeds the password exponent into multiplicative group operations:

$$g^{b(a+ux)}$$

SISSRP replaces exponentiation with linear algebra over lattices:

$$K' = a^T Ab$$

The security of SISSRP depends on hardness of SIS and collision resistance of hash function H. SISSRP operates in additive lattice space rather than multiplicative cyclic groups. The algebraic structure differs fundamentally, but the core PAKE invariants remain intact. The server stores only a verifier v. Password knowledge is required to derive the shared secret. Offline dictionary attacks remain infeasible without interaction.

The construction removes vulnerability to Shor's algorithm. SISSRP represents a structurally conservative but mathematically modernized version of SRP suitable for post-quantum environments.

8. Security Analysis

The security of SISSRP is analyzed in the framework of augmented PAKE security, taking in consideration both passive eavesdroppers and active adversaries, compromised servers, or attempt offline dictionary attacks. The standard security requirements for augmented PAKE protocols include:

1. **Resistance to offline dictionary attacks:** An adversary who intercepts protocol transcripts cannot efficiently test password guesses without interacting with a protocol participant.
2. **Forward secrecy:** Compromise of long-term secrets (verifiers) does not compromise the confidentiality of previously established session keys.
3. **Server compromise resistance:** An adversary who obtains the server's verifier database cannot impersonate a client without performing an online attack.
4. **Mutual authentication:** Both parties are assured of the other's possession of the correct password.
5. **Session key security:** The established session key is indistinguishable from random to an adversary who does not know the password and does not actively interfere with the session.

9. Future Plans

The future scope of the current project can be expanded by conducting a formal security analysis of the SISSRP protocol based on existing models for augmented PAKE, and verification of the protocol's ability to protect itself from both classical and quantum adversaries. Particularly important would be to carefully examine parameter selection for SISSRP (including lattice dimension, modulus size, and noise distribution). In addition, implementing and benchmarking the SISSRP protocol in a real-world environment and integrating it with existing standards (e.g., TLS) would yield insightful information regarding performance overhead, scalability, and usability. SISSRP's resistance to side-channel attacks, as well as analysis of weakness in the MapToShort transformation or hash environment, would strengthen the overall design. Finally, further research comparing SISSRP with other new post-quantum PAKE constructions and expanding the scope of SISSRP to advance scenarios (e.g., multi-factor authentication) or supporting distributed (threshold) PAKE systems would greatly improve SISSRP's suitability for secure authentication infrastructure on an industry-wide basis.

10. Acknowledgements

This work was supported by the Shota Rustaveli National Foundation of Georgia (YS-24-3272).

11. Declaration on Generative AI

The authors have not employed any Generative AI tools.

12. References

[1] Steiner, M., Tsudik, G., & Waidner, M. (1995). Refinement and extension of encrypted key exchange. *ACM SIGOPS Operating Systems Review*, *29*(3), 22-30.
[2] Exchange, D. H. K. (1976). Diffie-Hellman Key Exchange. *Diffie*.
[3] Bellovin, S. M., & Merritt, M. (1993, December). Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (pp. 244-250).
[4] Zhang, M. (2004). Analysis of the SPEKE password-authenticated key exchange protocol. *IEEE Communications Letters*, *8*(1), 63-65.
[5] Wu, T. D. (1998, March). The Secure Remote Password Protocol. In *NDSS* (Vol. 98, pp. 97-111).
[6] Jarecki, S., Krawczyk, H., & Xu, J. (2018, March). OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 456-486). Cham: Springer International Publishing.
[7] Ugwuishiwu, C. H., Orji, U. E., Ugwu, C. I., & Asogwa, C. N. (2020). An overview of quantum cryptography and shor's algorithm. *Int. J. Adv. Trends Comput. Sci. Eng*, *9*(5).
[8] Weinstein, Y. S., Pravia, M. A., Fortunato, E. M., Lloyd, S., & Cory, D. G. (2001). Implementation of the quantum Fourier transform. *Physical review letters*, *86*(9), 1889.
[9] Regev, O. (2006, August). Lattice-based cryptography. In *Annual International Cryptology Conference* (pp. 131-141). Berlin, Heidelberg: Springer Berlin Heidelberg.
[10] Amirkhanova, D. S., Iavich, M., & Mamyrbayev, O. (2024). Lattice-based post-quantum public key encryption scheme using ElGamal's principles. *Cryptography*, *8*(3), 31.