INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Anonymous Three-Party Quantum-Safe Authenticated Key Agreement with Forward Secrecy: Ring Learning with Errors

Swati Thakur[1], Hemlal Sahu[2], Pratik Gupta[3]*

1 Department of Mathematics, Govt. Naveen College, Nawagaon, Sect-28, Naya Raipur, India
2 Department of Mathematics, Govt. J. Yoganandam Autonomous arts and science college, Raipur, India
3 Department of Mathematics, D.A.V. Degree College, Lucknow, India

OPEN ACCESS

Received: 18/12/2025
Accepted: 12/02/2026
Published: 23/02/2026

Citation: Thakur S, Sahu H, Gupta P (2026) Anonymous Three-Party Quantum-Safe Authenticated Key Agreement with Forward Secrecy: Ring Learning with Errors. Indian Journal of Science and Technology 19(7): 415-428. https://doi.org/10.17485/IJST/v19i7.1985

* **Corresponding author**.

pratikguptagupta1810@gmail.com

Funding: None

Competing Interests: None

Copyright: © 2026 Thakur et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: To design a secure and efficient three-party post-quantum authenticated key agreement (AKE) protocol that ensures mutual authentication, user anonymity, and forward secrecy, while resisting both classical and quantum adversaries. **Methods:** The proposed protocol is constructed using the Ring Learning with Errors (Ring-LWE) assumption, which is widely regarded as quantum-resistant. The scheme enables two users to establish a shared session key through a trusted server over an insecure channel. Cryptographic hash functions, lattice-based operations, and masking techniques are employed to achieve authentication, anonymity, and key confidentiality. Formal correctness proofs, informal security analysis, and comparative performance evaluations are conducted to validate the protocol. **Findings:** The study describes qualitative benefits like ”outperforms current protocols” but lacks specific metrics, such as exact computation costs (e.g., 4913 ns total execution) or communication rounds, and ”reduces user-side cost to 2438 ns vs. 5395 ns in Islam et al.”**.** Also, substantially more efficient than pairing- or chaotic-map-based protocols, which incur costs exceeding 12000 ns. Communication overhead is reduced by limiting the exchange to two RLWE-based messages and one server response, avoiding expensive bilinear pairing or biometric hash transmissions. These results demonstrate that the proposed scheme provides stronger security guarantees at a lower computational and communication cost, making it suitable for resource-constrained mobile and IoT environments. **Novelty:** Unlike existing approaches, the proposed protocol simultaneously guarantees user anonymity, forward secrecy, and quantum resistance in a three-party setting based on Ring-LWE. It overcomes known vulnerabilities in prior schemes and achieves improved efficiency without compromising security, making it a practical and robust solution for post-quantum secure communications.

Keywords: Key Establishment; Authentication; Ring Learning with Errors; Anonymous Key Exchange; Post Quantum Forward Secrecy

1 Introduction

The advent of large-scale quantum computing poses a fundamental threat to classical cryptographic primitives based on integer factorization and discrete logarithm problems. Protocols such as RSA, Diffie–Hellman, and elliptic curve cryptography are vulnerable to polynomial-time quantum attacks, necessitating the development of cryptographic mechanisms that remain secure in post-quantum environments. Among these, post-quantum authenticated key agreement (AKE) protocols play a critical role in establishing secure communication over public channels.

In many modern applications—such as mobile networks, cloud services, and distributed systems—secure communication naturally involves three entities, typically two users and a trusted server. Three-party AKE protocols are therefore essential, as they enable mutual authentication and session key establishment among multiple participants. However, designing three-party AKE schemes that are simultaneously quantum-resistant, efficient, and privacy-preserving remains a challenging problem.

Lattice-based cryptography, particularly schemes grounded in the Learning with Errors (LWE) and Ring Learning with Errors (Ring-LWE) assumptions, has emerged as a leading candidate for post-quantum security due to its strong worst-case hardness guarantees and computational efficiency. Despite this, existing Ring-LWE-based three-party AKE protocols suffer from several limitations. Prior works have been shown to be vulnerable to signal leakage attacks, impersonation attacks, password guessing, or smart-card compromise. Other schemes rely on long-term key reuse, lack user anonymity, or incur significant computational and communication overhead, making them unsuitable for resource-constrained environments.

More importantly, most existing three-party post-quantum AKE schemes address security and efficiency in isolation, failing to achieve a balanced design that simultaneously ensures user anonymity, mutual authentication, forward secrecy, and resistance to both classical and quantum adversaries. This gap highlights the need for a unified three-party AKE construction that is provably secure under standard lattice assumptions while remaining practical for real-world deployment.

To address these challenges, this paper proposes an anonymous three-party quantum-safe authenticated key agreement protocol based on the Ring-LWE assumption. The proposed scheme enables two users to establish a shared session key through a server while preserving user anonymity and ensuring perfect forward secrecy. A formal correctness analysis, comprehensive informal security evaluation, and performance comparison demonstrate that the proposed protocol overcomes known vulnerabilities of existing schemes and achieves improved computational and communication efficiency.

This paper notes threats to traditional crypto but jumps to protocol description without explicitly stating the research gap (e.g., "Existing RLWE schemes lack anonymity and efficiency in three-party settings, as shown in Islam et al.'s privacy failures"). "Research Gap: Prior works fail user anonymity and impersonation resistance; we formulate the problem as a secure, anonymous RLWE-based three-party AKE under quantum threats."

1.1 Literature Review

A post-quantum key exchange system and a new unbiased reconciliation method were presented by Peikert et al.[1] in 2014. Their work introduced a collection of efficient and provably safe lower-level primitives for post-quantum cryptography. Unfortunately, the authenticated key exchange mechanism that Zhang et al.[2] developed in 2014 has a flaw: it uses a long-term server key repeatedly during communication, which makes it vulnerable to signal leakage attacks. Similarly, Kirkwood et al.[3] pointed out a serious security problem by pointing out the risks of repeating secret keys in ring learning with erroneous key exchange. Although the first descriptions were insufficient, this finding opened up new study directions. Expanding on this, Fluhrer[4] presented the novel idea of key mismatch for ring learning with error-based protocols in 2016, providing a way to get an honest party's private key. When employing the least important bits of keys with similar lengths, the features of match or mismatch become crucial in key recovery across parties. Ding et al.[5] introduced the idea of signal leakage attacks for ring learning with errors-based key exchanges, which reuse public/private keys, in 2017 after being inspired by Fluhrer's work. Under this presumption, they also presented a mechanism for key exchange and password authentication that is provably safe. This method required several sessions with the honest party in order to recover the secret key by examining the signal function's output. Interestingly, this technique worked well for creating session keys with about equal bit lengths by utilizing the least important bits of the keys. Ding et al.[6] expanded on this in later years, considerably minimizing the procedures needed to recover private keys. By lowering the number of steps required from "2.q" to "q + c," where "0 < c < q" denotes some constant, they improved the signal leakage attack. However, there were certain disadvantages to the developments in authenticated key exchange. Dabra et al.[7] in 2020, who pointed up flaws such weak login and authentication stages that were vulnerable to denial of service attacks. A three-factor authenticated key exchange system was presented by Dharminder et al.[8] the same year, although it was also vulnerable to signal leakage attacks. Two quantum-safe two-party authenticated key exchange protocols were presented by Islam et al.[9], however one of them was discovered to be susceptible to signal leakage attacks[10]. An effective, quantum-resistant two-factor authentication scheme for mobile devices was suggested by Wang et al.[11] in 2022;

however, it came with higher computation and communication overheads. A lattice-based secure reconciliation that facilitates key exchange for Internet of Things contexts was introduced by Dharminder et al.[12] in the same year. In 2023, Kumar et al.[13] presented a post-quantum key exchange system that relies on lattice assumptions to guarantee key agreement and authentication with little message transmission. In 2021, Islam et al.[14] suggested a three-party protocol based on ring learning mistakes in an effort to fill in the gaps. They claimed that the system was secure against a variety of attacks, including those from quantum computers. Nevertheless, their methodology was unable to guarantee user privacy and was still susceptible to impersonation, password guessing, and smartcard loss. Using perfect lattices, Rewal et al.[15] presented a mobile user authentication scheme that is robust to quantum attacks the same year. The current flaws in the procedures of Rawel et al. and Islam et al. The suggested approach by Uddeshya et al.[16] provides a more efficient solution in 2024 based on performance study and security comparison. A post-quantum secure three-party authenticated key agreement technique that makes use of RLWE was proposed by Kumar et al. Although Kumar et al.'s plan addresses impersonation and replay attacks, which are weaknesses in Islam et al. and Rewal et al.'s protocols, it might not adequately solve efficiency issues in high-performance settings. The same year, Park et al.[17] suggested a system that uses the server's master key and the user's biometric data to keep important parameters hidden. Their plan compromised three-factor security by relying on expensive biometric hashing operations and lacked anonymity, despite its goal of increasing speed and security against password guessing attacks. Concerns were also raised about the vulnerability of smartphone fingerprint scanners to partial match exploitation, which might allow unwanted access to a large number of user accounts. A comparative analysis is given in Table 1.

Table 1. Comparative Analysis of Three-Party Authenticated Key Agreement Protocols

schemes → features ↓	[14]	[18]	[19]	[20]	Ours
Post-Quantum Security	Partial	Partial	Yes	Partial	Yes
User anonymity	No	No	Yes	Yes	Yes
Mutual Authentication	Yes	Yes	Yes	Yes	Yes
Forward Secrecy	Yes	Yes	Yes	Yes	Yes
Replay Attack Resistance	Yes	Yes	Yes	Yes	Yes
Impersonation attack	Vulnerable	Vulnerable	Partially Protected	Fully Protected	Fully Protected
Efficiency Improvement	Moderate	Low	Moderate	Moderate	High

Contextual Explanation

Post-Quantum Security: Your scheme leverages ring learning with errors (RLWE) assumptions, offering strong security against quantum adversaries.

User Anonymity: Unlike many previous protocols, your scheme ensures user anonymity, addressing privacy concerns. Efficiency: By optimizing cryptographic operations and reducing message exchanges, your protocol demonstrates lower computation and communication costs.

Security Enhancements: The proposed scheme includes robust defences against known attacks like replay, impersonation, and password guessing attacks.

Benefits of Including the Table

Clarity: Provides a quick visual summary of how your work compares to and improves upon existing solutions. Context: Demonstrates the specific gaps your protocol fills in the current literature.

Impact: Emphasizes the novel contributions and the practical advantages of your proposed approach.

1.2 Motivation and silent features

The rise of quantum computing poses a significant threat to traditional cryptographic systems based on mathematical problems like integer factorization and discrete logarithms. As quantum computers become more powerful, these systems may become vulnerable to efficient attacks, compromising the security of encrypted communication. Therefore, there is a need to develop new cryptographic primitives and protocols that resist quantum attacks and provide long-term security guarantees. The design of three-party postquantum key establishment protocols is motivated by the need to address the security challenges posed by quantum computing, support secure communication among multiple parties, ensure privacy and confidentiality of transmitted data, and meet the evolving security requirements of modern applications and environments. These protocols play a critical role in enabling secure and resilient communication in the face of emerging threats and security challenges. To overcome the shortcomings of their first plan, this study presents a technique that successfully fends off the possible known attacks while inheriting the methods of Islam et al. and Chaudhary et al.[14],[20]. Furthermore, it is shown that even with the strict "Ring

Learning with Errors" assumption, the suggested method is secure. All of the security elements are maintained, and a valid verification of the system is given. The proposed anonymous messaging platforms allow users to send and receive messages without revealing their identities. These platforms typically use encryption and anonymization techniques to protect the privacy of users, ensuring that messages cannot be traced back to their originators. Examples include secure messaging apps like Signal and Telegram, as well as anonymous messaging platforms like Briar and Ricochet.

2 Material and methods

2.1 Preliminaries

Learning with Errors (LWE): It is a problem in computational mathematics and cryptography that involves learning a linear function from a set of noisy examples. The LWE problem is widely used in cryptographic schemes because it provides strong security guarantees, especially against quantum computers. Ring Learning with Errors (Ring-LWE): It is a variant of the Learning with Errors (LWE) problem that brings additional structure by operating in the context of polynomial rings. This adaptation makes the problem more efficient in computation and storage, which is crucial for practical cryptographic applications. Before delving into the specifics of a three-party post-quantum Authenticated Key Exchange (AKE) protocol. it's crucial to establish some preliminary concepts and components that form the basis of such protocols. Understanding these preliminary concepts is essential for designing, analyzing, and implementing three-party post-quantum AKE protocols.

Table 2. Notations

Notations	Descriptions
$\mathcal{U}_i$	i^{th} User $i \in \{a,b\}$
$\mathcal{S}_j$	j^{th} Server
ϱ	Public key of $\mathcal{S}_j$
s	Secret of $\mathcal{S}_j$
R_q	Finite rings of polynomials
$\Phi(.$	Characteristics function
$E(F_q)$	Elliptic curve with F_q
T_β	Discrete Gaussian
$\beta > 0$	Standard deviation
$\mathcal{A}$	Adversary
ID_i	Identities for U_i
ω_i	Biometric imprints of U_i
PW_i	Passwords for U_i
$h(\cdot)$	Cryptographic hash
$G(\cdot)$	Fuzzy Extractors
$\oplus$	Bitwises XOR
$\|\|$	Strings concatenations

These concepts form the foundation upon which secure and resilient communication channels can be established between multiple parties in a postquantum computing era. This section have discussion on basic notations (see Table 2). Consider $n = 2^\tau$, where $\tau \in \mathbf{Z}$ refers to the security parameters. An irreducible polynomial $(\xi^n + 1) \in \mathbf{Z}[\xi]$ over the set of integers $\mathbf{Z}$. The rings of polynomial $\mathbf{Z}_q[\xi]$ over $\mathbf{Z}_q$, where coefficients are reduced under prime modulus. The rings of finite degree polynomials modulo polynimial ($\xi^n + 1$) over $\mathbf{Z}$ is $\mathbf{R} = \mathbf{Z}[\xi]/(\xi^n + 1)$, and the finite degrees polynomials under reduced coefficients is $\mathbf{R}_q = \mathbf{Z}_q[\xi]/(\xi^n + 1)$ with coefficient reduced over $\mathbf{Z}_q$. The discrete Gausssian T_β with $\beta > 0$ is defined over $\mathbf{R}_q$. Let $K = \{-\lfloor \frac{q}{4} \rfloor, \ldots, \lfloor \frac{q}{4} \rfloor\}$ be the half middle subset of $Z_q = \left\{-\frac{(q-1)}{2}, \ldots, \frac{(q-1)}{2}\right\}$. The characteristic function $\Phi(.) is defined on compliment of K = \{-\lfloor \frac{q}{4} \rfloor, \ldots, \lfloor \frac{q}{4} \rfloor\}$, that is defined as $\Phi(\vartheta) = 0$ if $z \in K$, and $\Phi(\vartheta) = 1$ otherwise. The definition of the auxiliary modular function $\Psi_2 : \mathbf{Z}_q \times \{0,1\} \to \{0,1\}$ is $\Psi_2(\vartheta, d) = \left(\vartheta + d.\frac{q-1}{2}\right) mod q mod 2$, where $\vartheta \in \mathbf{Z}_q$ and $d = (\#)[2]$.

Lemma 2.1.1: Let us take $q = \left(2^{\omega(\log_2 n)} + 1\right) > 2$ be a arbitrary prime, and ϑ is random element in the group $\mathbf{Z}_q$, then $c, d \in \{0,1\}$ and $c\omega \in \mathbf{Z}_q$, result of $\Psi_2(\vartheta + \omega, d)$ given by (ϑ) is statistically close to random binary string[(2)].

Lemma 2.1.2: Given q and $c, e \in \mathbf{R}_q$ such that $|e| < \frac{q}{8}$, we have $\Psi_2(c,(\mathbf{c}))) = \Psi_2(\omega,(\mathbf{c})))$, where ω =c+2.e[(2)]. On $\mathbf{R}_q$, Cha and Ψ_2 are easily extendable functions as: Given $c = c_0 + c_1 x + c_2 x^2 + ... + c_{n-1} x^{n-1} \in \mathbf{R}$, we considered it as a vector $c = (c_0, c_1, ..., c_{n-1})$. For $\vartheta = (\vartheta_0, \vartheta_1, ..., \vartheta_{n-1}) \in \{0,1\}^n$, we have $(\mathbf{c}) = ((\mathbf{c_0}), (\mathbf{c_1}), ..., (\mathbf{c_{n-1}}))$, $\Psi_2(c, \vartheta)$ =$(\Psi_2(c_0, \vartheta_0), ..., \Psi_2(c_{n-1}, \vartheta_{n-1}))$.

Definition 2.1.3: Suppose that $s \in \mathbf{R}_q$ and $\mathbf{A}_\beta, T_\beta$ folow random distribution over (c, c.s $+2.e) \in \mathbf{R}_q \times \mathbf{R}_q, c \in \mathbf{R}_q$, and $e \leftarrow T_\beta$. A polynomial times simulator/algorithm $\mathcal{B}$ cannot find the difference between $\mathbf{A}_{s,T_\beta}$, and random element in $\mathbf{R}_q \times \mathbf{R}_q$.

2.2 Fuzzy Extractor

A cryptographic tool called a fuzzy extractor is used to extract consistent and reliable secret keys from noisy data sources, including physical measurements or biometric data. It's particularly useful in scenarios where the data may vary slightly due to noise, errors, or variations in measurement conditions. The goal of a fuzzy extractor is to derive a stable secret key that remains consistent across multiple readings of the same source data while still being resistant to variations and noise. The Fext extraction method is used, which allows for the uniform selection of a random binary string. With a certain error margin, this extractor made it possible to retrieve private data from physiological input, such as biometric fingerprints. A user's noisy source data, such as a biometric sample (such as an iris scan or fingerprint), is recorded and processed throughout the enrollment phase to produce a stable template or representation. This template is generated in such a way that it captures the essential features of the source data while mitigating the effects of noise and variations. The cryptographic technique known as the "fuzzy extractor," or Fext method, is used to extract a reliable secret key from a noisy data source. The key aspect of the Fext algorithm is its ability to generate the same secret key consistently from similar noisy inputs, even in the presence of slight variations or errors in the data.

2.3 Ring Learning with Errors for Three-Party Quantum Secure Authentication and Key Agreement

The lifespan of a user's engagement with the system is usually discussed when examining the setup, registration, login, and authentication stages of a system, particularly in relation to authentication and access control. Here's an overview of each phase (see Figure 1):

2.3.1. Setup Phase

The setup phase involves initializing the system and configuring it for use. This phase sets up the infrastructure, and security parameters necessary for user registration and authentication. The server S_j is responsible for setup algorithm, and it take cares of system settings, master secret key discussed as below:

1. S_j takes a prime $q > 2$, and T_β the Gaussian with deviation $\beta > 0$.
2. S_j also takes $\alpha \in Q_q$ and it calculates the public key value $\varrho = \alpha s + 2e$, where $s, e \leftarrow T_\beta$ randomly.
3. S_j also selects hashing $h : \{0,1\}^* \rightarrow \{0,1\}^\eta$ with output length η, and finally S_j publishes $\{n, q, \alpha, T_\beta \varrho, h(). \}$, and keeping the value s secret.

2.3.2. Registration Phase

The registration phase involves enrolling users into the system and establishing their identity and credentials. During this phase, users create accounts and provide necessary information for authentication. The user U_i, where $i \in \{a, b\}$, first registers himself before using any service. Furthermore, the U_i sends messages to S_j using a secure channel.

1. User U_i sets ID_i as identity, PW_i password, and imprints biometric ω and generates $Gen(\omega)$ =(B_i, B_i^*), and the device computes A_i =$h(ID_i \| PW_i \| B_i)$, and transmits ID_i, A_i to S_j.
2. S_j receives ID_i, A_i it further computes SID_i, where $SID_i = h(ID_i \| s), C_i = SID_i \oplus h(A_i)$, stores $h(ID_i)$ and sends $C_i, h(.)$ *to the* U_i.
3. U_i gets $C_i, h(.) from the channel, where i \in \{a, b\}$ and computes $A_i = h(ID_i \| PW_i \| B_i)$, $SID_i = C_i \oplus h(A_i), V_i = h(ID_i \| PW_i \| B_i \| SID_i)$, and stores $\{C_i, V_i, D_i, B_i^*, h(). \}$ into the device.

2.3.3. Login and authentication Phase

The login phase involves users accessing the system by presenting their credentials and authenticating their identity. Successful authentication grants users access to authorized resources and functionalities. In Figure 1, the session key sk computed by the

server is equivalent to the session key sk' reconstructed by the users during the verification phase. This ensures consistency in the session key used across all parties. The authentication phase ensures ongoing security and integrity by continuously verifying users' identities throughout their interaction with the system.

2.3.3.1. Users Login Phase. Users provide their authentication credentials, such as entering a password or presenting a cryptographic token. The user gives input ID_i, PW_i for $i \in \{a,b\}$, and the device computes $B_i = GRep(\omega_i', B_i^*), A_i' = h(ID_i \| PW_i \| B_i)$, and $SID_i' = h(A_i') \oplus C_i$ for $i \in \{a,b\}$, respectively. Furthermore, the user completes verification $V_a' = ?\, h(ID_a \| PW_a \| B_a \| SID_a')$ for user U_a, and $V_b' = ?h(ID_b \| PW_b \| B_b \| SID_b')$ for user U_b, to get logged into corresponding devices.

2.3.3.2. Authentication and Key Agreement Phase. When a user or entity tries to access a system, service, or resource, the authentication process confirms their identity. Usually, credentials like passwords, cryptographic keys, or biometric information are presented and verified.

An outline of the authentication procedure is provided below:

1. After successful login, the user U_a chooses samples $r_a, \nu_a \leftarrow T_\beta$, the user U_b chooses samples $r_b, \nu_b \leftarrow T_\beta$, then U_a computes $z_a = \alpha r_a + 2\nu_a$, and U_b computes $z_b = \alpha r_b + 2\nu_b$, respectively.
2. The user U_a computes $k_a = r_a \varrho, c_a = \Phi(k_a)$ with the help of server S_j 's public key. Further it computes $m_a = \Psi_2(k_a, c_a)$, a masked identity $Aid_a = (ID_a \| ID_b) \oplus h(m_a \| z_a)$, and a component $v_a = h(A_{id} d_a \| z_a \| m_a \| ID_a \| SID_a)$ for verification, and it sends z_a, v_a, Aid_a, c_a to the server.
3. Similarly, the user U_b computes $k_b = r_b \varrho$, $c_b = \Phi(k_b), m_b = \Psi_2(k_b, c_b), Aid_b = ID_b \oplus h(m_b \| z_b)$, and factor of verification $v_b = h(A_i d_b \| z_b \| m_b \| ID_b \| SID_b)$, and transmits message z_b, v_b, Aid_b, c_b to the server S_j.
4. The server S_j computes $k_a' = z_a.s, k_b' = z_b.s, m_a' = \Psi_2(k_a', c_a), m_b' = \Psi_2(k_b', c_b)$, identities $ID_a' = Aid_a \oplus h(m_a' \| z_a)$, and $ID_b' = Aid_b \oplus h(m_b' \| z_b)$. Further it computes $SID_a' = h(ID_a' \| s), SID_b' = h(ID_b' \| s)$, and vrifies $v_a = h(Aid_{\ \ a} \| z_a \| m_a' \| ID_a' \| SID_a'), v_b = h(Aid_b \| z_b \| m_b' \| ID_b' \| SID_b')$, and computes session key $sk = h(ID_a \| ID_b \| z_a \| z_b \| m_a \| m_b \| s)$. Finally, it computes masked session key $sk_a = sk \oplus h(m_a \| z_a)$ for U_a, and $sk_b = sk \oplus h(m_b \| z_b)$ for U_b, and transmits $vs_b = h(sk || m_b |_b || SID_b || ID_b)$, and $vs_a = h(sk || m_a || z_a || SID_a || ID_a)$ to U_b, and U_a, respectively.
5. Finally, the users U_a computes $sk' = sk_a \oplus h(m_a \| z_a)$, and veries the session key by $vs_a = ?\, h(sk \| m_a \| z_a \| SID_a \| ID_a)$, and similarly, U_b computes $sk' = sk_b \oplus h(m_b \| z_b)$, and verifies session key by $vs_b = ?\, h(sk \| m_b \| z_b \| SID_b \| ID_b)$.

2.3.4. Correctness

Organizations may increase their trust in the security features and dependability of authentication methods by guaranteeing their mathematical accuracy, which lowers the possibility of data compromise, illegal access, and security breaches. Before being deployed in production systems, formal verification approaches assist to discover and mitigate possible vulnerabilities and threats by offering a methodical and rigorous way to assessing and confirming the security of authentication protocols. Server authenticates $\mathcal{U}_a$ by verification factor $v_a = ?\, v_a'$ where $v_a = h(Aid_a \| z_a \| m_a \| ID_a)$ contains masked identity and $v_a' = h(Aid_a \| z_a \ \ \| m_a' \| ID_a')$. Since $\mathcal{U}_a$ transmits $\{z_a, \nu_a, A_a d_a, c_a\}$ to S_j, therefore S_j requires to compute m_a, ID_a, for this S_j takes help of Z_a, c_a and Aid $_a$. Since $m_a = \Psi_2(k_a, c_a)$, therefore it is required to compute correct m_a, k_a and k_a', and it must follow inequality $|k_a - k_a'| < \frac{q}{8}$.

$$k_a = r_a \cdot \varrho = r_a \cdot \alpha \cdot s + 2e \cdot r_a \tag{1}$$

$$k_a' = z_a \cdot s = \alpha \cdot r_a \cdot s + 2\nu_a \cdot s \tag{2}$$

By subtracting equation (2) from (1):

$$|k_a - k_a'| = 2|e \cdot r_a - \nu_a \cdot s| \tag{3}$$

By applying triangular inequality, we have

$$\begin{aligned} |e \cdot r_a - \nu_a \cdot s| &\le |e \cdot r_a| + |\nu_a \cdot s| \le \sqrt{n} \cdot \|e\| \cdot \|r_a\| + \sqrt{n} \cdot \|\nu_a\| \cdot \|s\| \\ &< \sqrt{n} \cdot \sqrt{n} \cdot \delta \cdot \sqrt{n} \cdot \delta + \sqrt{n} \cdot \sqrt{n} \cdot \delta \cdot \sqrt{n} \cdot \delta \\ &= 2 \cdot \sqrt{n^3} \cdot \delta^2 \end{aligned}$$

where $n \ll q$ and $\delta = \omega(\sqrt{logn})$. So

$$|e \cdot r_a - \nu_a \cdot s| \leq 2\sqrt[3]{n} \cdot \delta^2 \Big| < \frac{q}{8} \tag{4}$$

Now by Lemma (IV.2), we have

$$\Psi_2(k_a, c_a) = \Psi_2(k'_a, c_a) \tag{5}$$

3 Result and discussion

3.1 Informal Security Analysis

The objective of this informal security analysis of authentication protocols can be summarized as follows:

- To determine how well authentication procedures work to achieve safe and dependable authentication, as well as how resilient they are to any threats and weaknesses.
- This involves examining how well these protocols can resist attacks, protect sensitive information, and ensure that authentication processes are both secure and trustworthy.

Here are some key aspects to consider in such an analysis:

3.1.1. Man in the middle attack

A cyberattack known as a "Man-in-the-Middle" (MitM) occurs when a malevolent actor intercepts and perhaps modifies communication between two parties without the parties' knowledge or agreement. In order to provide the impression that the two parties are speaking with each other directly, attacker $\mathcal{A}$ surreptitiously transmits and potentially alters the data that is transferred between them. In order to get sensitive information or to maliciously alter the conversation, the attacker may also pretend to be one or both of the participants. Once $\mathcal{A}$ gets $< z_a, v_a, Aid_a, c_a >$ from the public channel, then it needs to compute $z_a = \alpha r_a + 2\nu_a$, and it tries to manipulate the massage by $< z'_a, v'_a, Aid'_a, c'_a >$ by sampling fresh r'_a, ν'_a and sends it the server. But, the server detects tempering by verifying v'_a. Since $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$ and $SID_i = h(ID_i \| s)$, therefore, $\mathcal{A}$ cannot generate duplicate v'_a because it requires the secret key of server. $\mathcal{A}$ needs N_a, and secret SID_a if it attempts to intercept data z_b, N_a, v_s from server S_j to $\mathcal{U}_a$. Therefore, an MITM attack between $(U)_a$ and the server is not feasible. We may state that it is likewise impossible between $(U)_b$ and the server in the same way.

3.1.2. Perfect forward security

Perfect Forward Secrecy (PFS) is a property of cryptographic protocols that ensures that past session keys remain secure even if the long-term secret keys are compromised in the future. Each session key is independent of the long-term secret keys used by the parties for authentication and key derivation. Even if an attacker gains access to the long-term secret keys at a later time, they cannot use them to decrypt past communication sessions since the session keys are derived independently for each session. The user U_a chooses fresh samples $r_a, \nu_a \leftarrow T_\beta$, the user U_b chooses fresh samples $r_b, \nu_b \leftarrow T_\beta$, then U_a computes $z_a = \alpha r_a + 2\nu_a$, and U_b computes $z_b = \alpha r_b + 2\nu_b$, respectively. The user U_a computes $k_a = r_a \varrho, c_a = \Phi(k_a)$ with the help of server S_j 's public key. Further it computes $m_a = \Psi_2(k_a, c_a)$, a masked identity $Aid_a = (ID_a \| ID_b) \oplus h(m_a \| z_a)$, and a component $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$ for verification, and it sends z_a, v_a, Aid_a, c_a to the server. Similarly, the user U_b computes $k_b = r_b \varrho, c_b = \Phi(k_b), m_b = \Psi_2(k_b, c_b)$, $Aid_b = ID_b \oplus h(m_b \| z_b)$, and factor of verification $v_b = h(Aid_b \| z_b \| m_b \| ID_b \| SID_b)$, and transmits message z_b, v_b, Aid_b, c_b to the server S_j. The server S_j computes $k'_a = z_a.s, k'_b = z_b.s, m'_a = \Psi_2(k'_a, c_a)$, $m'_b = \Psi_2(k'_b, c_b)$, identities $ID'_a = Aid_a \oplus h(m'_a \| z_a)$, and $ID'_b = Aid_b \oplus h(m'_b \| z_b)$. Further it computes $SID'_a = h(ID'_a \| s), SID'_b = h(ID'_b \| s)$, and vrifies $v_a = h(Aid_a \| z_a \| m'_a \| ID'_a \| SID'_a), v_b = h(Aid_b \| z_b \| m'_b \| ID'_b \| SID'_b)$, and computes session key $sk = h(ID_a \| ID_b \| z_a \| z_b \| m_a \| m_b \| s)$ which includes random values m_a, m_b, and secret key of S_j which makes it forward secure.

3.1.3. Impersonation attack

An impersonation attack is a kind of cyberattack in which the perpetrator poses as a trustworthy user, device, or system entity in order to get resources, data, or privileges without authorization. Once the attacker successfully impersonates the target identity, they may attempt to escalate their privileges or gain access to sensitive resources or systems. This could involve exploiting vulnerabilities in access control mechanisms, exploiting trust relationships between systems, or bypassing

security controls through deception or manipulation. $\mathcal{A}$ might masquerade as any of the three parties$\mathcal{U}_a, \mathcal{U}_b$, and server—in the suggested protocol. To impersonate $U_a, \mathcal{A}$ uses information $< z_a, v_a, Aid_a, c_a >$, but it is protected by v_a, since $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$, therefore, $\mathcal{A}$ cannot create valid v_a, because for this $\mathcal{A}$ needs SID_a includes secret s of the server. Similarly, we can see for the communication for U_b to S_j, hence impersonation is not possible.

3.1.4. Replay attack

One kind of cyberattack is called a replay attack, in which the attacker intercepts and sends data that was shared between two parties in a prior communication session. When two parties, such a client and a server, are communicating, the attacker listens in and records the data packets that are sent back and forth.This could occur over a network connection, wireless transmission, or any other communication channel. At a later time, the attacker replays the captured data packets to the target system or service, mimicking the original communication session. This involves resending the captured data packets with the same content and sequence as they were originally transmitted. The user U_a chooses fresh samples $r_a, \nu_a \leftarrow T_\beta$, the user U_b chooses fresh samples $r_b, \nu_b \leftarrow T_\beta$, then U_a computes $z_a = \alpha r_a + 2\nu_a$, and U_b computes $z_b = \alpha r_b + 2\nu_b$, respectively. The user U_a computes $k_a = r_a \varrho, c_a = \Phi(k_a)$ with the help of server S_j 's public key. Further it computes $m_a = \Psi_2(k_a, c_a)$, a masked identity $Aid_a = (ID_a \| ID_b) \oplus h(m_a \| z_a)$, and a component $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$ for verification, and it sends z_a, v_a, Aid_a, c_a to the server. Similarly, the user U_b computes $k_b = r_b \varrho, c_b = \Phi(k_b), m_b = \Psi_2(k_b, c_b)$, $Aid_b = ID_b \oplus h(m_b \| z_b)$, and factor of verification $v_b = h(Aid_b \| z_b \| m_b \| ID_b \| SID_b)$, and transmits message z_b, v_b, Aid_b, c_b to the server S_j. The server S_j computes $k'_a = z_a.s, k'_b = z_b.s, m'_a = \Psi_2(k'_a, c_a), m'_b = \Psi_2(k'_b, c_b)$, identities $ID'_a = A_i d_a \oplus h(m'_a \| z_a)$, and $ID'_b = Aid_b \oplus h(m'_b \| z_b)$. Further it computes $SID'_a = h(ID'_a \| s), SID'_b = h(ID'_b \| s)$, and vrifies $v_a = h(Aid_a \| z_a \| m'_a \| ID'_a \| SID'_a), v_b = h(Aid_b \| z_b \| m'_b \| ID'_b \| SID'_b)$, and computes session key $sk = h(ID_a \| ID_b \| z_a \| z_b \| m_a \| m_b \| s)$ which includes new random values m_a, m_b, and secret key of S_j which makes replay attack impossible.

3.1.5. Password guessing attack

A password guessing attack, also known as a brute force attack or a dictionary attack, is a method used by attackers to gain unauthorized access to a system or an account by systematically attempting different combinations of passwords until the correct one is found. In a password guessing attack, the attacker $\mathcal{A}$ uses automated tools or scripts to try a large number of possible passwords, hoping to discover the correct one and gain access to the target system or account. To implement password guessing, the attacker $\mathcal{A}$ starts by collecting a list of usernames or user identifiers associated with the target system or service. The user gives input ID_i, PW_i for $i \in \{a, b\}$, and the device computes $B_i = GRep(\omega'_i, B^*_i), A'_i = h(ID_i \| PW_i \| B_i)$, and $SID'_i = h(A'_i) \oplus C_i$ for $i \in \{a, b\}$, respectively. Furthermore, the user completes verification $V'_a = ? h(ID_a \| PW_a \| B_a \| SID'_a)$ for user U_a, and $V'_b = ? h(ID_b \| PW_b \| B_b \| SID'_b)$ for user U_b, to get logged into corresponding devices. The scheme holds three factor authentication to stop password guessing.

3.1.6. User anonymity and untraceable

User anonymity and untraceability are concepts related to privacy and confidentiality in digital communication and online activities. refers to the ability of an individual to interact with a system or engage in online activities without revealing their true identity or personal information. An anonymous user is not easily identifiable or traceable by other parties, including service providers, websites, or network operators. Untraceability refers to the inability to trace or link specific actions, transactions, or communications back to their origin or source. The user U_a chooses fresh samples $r_a, \nu_a \leftarrow T_\beta$, the user U_b chooses fresh samples $r_b, \nu_b \leftarrow T_\beta$, then U_a computes $z_a = \alpha r_a + 2\nu_a$, and U_b computes $z_b = \alpha r_b + 2\nu_b$, respectively. The user U_a computes $k_a = r_a \varrho, c_a = \Phi(k_a)$ with the help of server S_j 's public key. Further it computes $m_a = \Psi_2(k_a, c_a)$, a masked identity $Aid_a = (ID_a \| ID_b) \oplus h(m_a \| z_a)$, and a component $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$ for verification, and it sends a session dependent message z_a, v_a, Aid_a, c_a to the server. Similarly, the user U_b computes $k_b = r_b \varrho, c_b = \Phi(k_b), m_b = \Psi_2(k_b, c_b)$, $Aid_b = ID_b \oplus h(m_b \| z_b)$, and factor of verification $v_b = h(Aid_b \| z_b \| m_b \| ID_b \| SID_b)$, and transmits session dependent message z_b, v_b, Aid_b, c_b to the server S_j. The server S_j computes $k'_a = z_a.s, k'_b = z_b.s, m'_a = \Psi_2(k'_a, c_a), m'_b = \Psi_2(k'_b, c_b)$, identities $ID'_a = Aid_a \oplus h(m'_a \| z_a)$, and $ID'_b = Aid_b \oplus h(m'_b \| z_b)$. Further it computes $SID'_a = h(ID'_a \| s), SID'_b = h(ID'_b \| s)$, and vrifies $v_a = h(Aid_a \| z_a \| m'_a \| ID'_a \| SID'_a), v_b = h(Aid_b \| z_b \| m'_b \| ID'_b \| SID'_b)$, and computes session dependent session key $sk = h(ID_a \| ID_b \| z_a \| z_b \| m_a \| m_b \| s)$ which includes new random values m_a, m_b, and secret key of S_j which makes protocol untraceable.

3.1.7. Confidentiality

A key component of information security is confidentiality, which makes sure that private information is shielded from exposure, disclosure, and illegal access. To put it simply, confidentiality is the process of preventing unauthorized parties from accessing sensitive information and allowing only authorized persons or institutions to view or access it. In the discussed protocol, $\mathcal{U}_a$ and $\mathcal{U}_b$ exchange messages $< z_a, v_a, Aid_a, c_a >$ and $< z_b, v_b, Aid_b, c_b >$ to server. In Z_a and z_b, the secret is protected by RLWE assumption, therefore $\mathcal{A}$ can not get session key. Moreover, the validity of messages are given by v_a, and v_b protected by $h(.)hashing$.

3.1.8. The Mutual Authentication and Key Agreement

To guarantee that both parties to a communication exchange can verify each other's identities and create a shared secret key for secure communication, secure communication protocols employ cryptographic techniques called mutual authentication and key agreement. Protocols like Internet Key Exchange (IKE) and Transport Layer Security (TLS) frequently employ these strategies to secure network connections. The user U_a chooses fresh samples $r_a, \nu_a \leftarrow T_\beta$, the user U_b chooses fresh samples $r_b, \nu_b \leftarrow T_\beta$, then U_a computes $z_a = \alpha r_a + 2\nu_a$, and U_b computes $z_b = \alpha r_b + 2\nu_b$, respectively. The user U_a computes $k_a = r_a \varrho, c_a = \Phi(k_a)$ with the help of server S_j 's public key. Further it computes $m_a = \Psi_2(k_a, c_a)$, a masked identity $Aid_a = (ID_a \| ID_b) \oplus h(m_a \| z_a)$, and a component $v_a = h(Aid_a \| z_a \| m_a \| ID_a \| SID_a)$ for authentication, and it sends a session dependent message z_a, v_a, Aid_a, c_a to the server. Similarly, the user U_b computes $k_b = r_b \varrho, c_b = \Phi(k_b), m_b = \Psi_2(k_b, c_b)$, $Aid_b = ID_b \oplus h(m_b \| z_b)$, and factor of verification $v_b = h(Aid_b \| z_b \| m_b \| ID_b \| SID_b)$, and transmits session dependent message z_b, v_b, Aid_b, c_b to the server S_j. The server S_j computes $k'_a = z_a.s, k'_b = z_b.s, m'_a = \Psi_2(k'_a, c_a), m'_b = \Psi_2(k'_b, c_b)$, identities $ID'_a = A_i d_a \oplus h(m'_a \| z_a)$, and $ID'_b = Aid_b \oplus h(m'_b \| z_b)$. Further it computes $SID'_a = h(ID'_a \| s), SID'_b = h(ID'_b \| s)$, and vrifies $v_a = h(Aid_a \| z_a \| m'_a \| ID'_a \| SID'_a), v_b = h(Aid_b \| z_b \| m'_b \| ID'_b \| SID'_b)$, and computes session dependent session key $sk = h(ID_a \| ID_b \| z_a \| z_b \| m_a \| m_b \| s)$ which includes new random values m_a, m_b, and secret key of S_j is established between two users.

3.2 Security Comparison and Performance Analysis

This section presents a comparison between our proposed three-party authentication protocol and existing systems in terms of execution performance and communication costs. We use $\log_2 \beta = 17.1$ to uniformly choose samples at random from χ_β based on the computations performed in [2]. The 512-bit digest value is generated by the SHA-512-bit hash algorithm, which is used for hashing. Furthermore, we assumed that the sizes of n, q, and a component of $\mathbb{R}_q$ be 1024 bits, 12289 bits, and 4096 bits, each. The user's identity and timestamp have a size of 32 bits each.

3.2.1. Execution Cost

In this part, we compare the execution cost of our proposed plan with methods that have been established previously (Tables 3-5). We use Lattice Crypto and the MIRACL Libraries to create our protocol. The research employed two systems: a Lenovo desktop PC (Windows 10 Professional operating system, 3.4 Giga Hertz Intel(R) Core (TM) i7-6700 CPU, and 8GB RAM), which was considered a server system, and a Xiaomi Redmi 1S smartphone (Android 4.3 operating system, 1.6 GHz quadcore Cortex-A7 processor, and 1.0 GB RAM). As indicated in Table 4, the recommended method is computed by calculating the running durations of different cryptographic procedures in nanoseconds. We accept the cost of the f_2 function as supplied. A one-time bio hashing function, six-time hash functions, a one-time component-wise multiplication in $\mathbb{Q}_p$„ a one-time componentwise multiplication and addition in $\mathbb{Q}_p$„ a one-time scalar multiplication in $\mathbb{Q}_p$„ and the selection of two gaussian random samples in χ_β, are all part of the suggested authentication step. Hence, $6T_h + T_{H_b} + 2T_\beta + T_{sm} + T_{pma} + T_{mp} = 2438.926\ ns$ is the total amount of time that has passed. The user B chooses two random samples in χ_β, performs a one-time bio hashing function, six one-time hash functions, a one-time T_{ma} operation in $\mathbb{Q}_p$, a one-time scalar multiplication in $\mathbb{Q}_p$, a one-time component-wise multiplication in $\mathbb{Q}_p$, and a one-time g function. Consequently, $6T_h + T_{H_b} + 2T_\beta + T_{sm} + T_{mpa} + T_{mp} + T_{cha} = 2474.441\ ns$ is the overall cost.

Figure 1 illustrates the logins and authentication phases.

Six times, the hash functions are executed by the server S. The overall cost of calculation is $6T_h = 84.54\mu s$. Thus, 4913.367 ns (nanoseconds) is the overall execution cost of our protocol. Table 6 provides Server configuration.

The authentication scheme proposed by Islam et al.[15] involves the selection of two gaussian random samples in χ_β, the execution of eight hash functions, a one-time component-wise multiplication with addition in $\mathbb{Q}_p$, a one-time component-wise scalar multiplication in $\mathbb{Q}_p$, a one-time component-wise multiplication in $\mathbb{Q}_p$, and a one-time function. Thus the total cost is $8T_h + 2T_\beta + T_{sm} + T_{mpa} + T_{mp} + T_{cha} = 2655.405\ ns$, B selects two gaussian random samples in χ_β, executes eight-time

Table 3. The comparison of computation costs

Schemes	User-side Computation cost	Server-side Computation cost	Total Computation cost
Proposed scheme	$2\,(7t_{ha}+2t_{\delta}+2t_{om}+t_{am}+t_{sm}+2t_{ch}) \approx 4.9990188$	$10t_{ha}+t_{\delta}+2t_{om} \approx 0.215017$	5.2140358
Islam et al.2021 [14]	$2\,(8t_{ha}+2t_{\delta}+t_{om}+t_{am}+t_{sm}+t_{ch}) \approx 5.2873224$	$8t_{ha} \approx 0.11272$	5.4000424
Zhang et al. [18]	$12t_{ha}+4t_{c} \approx 12962.171568$	$7t_{ha} \approx 0.09863$	12962.270198
Xie et al [19]	$12t_{\text{ha}}+4t_{\text{c}}+5t_{\text{sym}} \approx 21592.171568$	$7t_{\text{ha}}+2t_{c}+5t_{\text{sym}} \approx 12524.08863$	34166.260198
Islam et al.2015 [21]	$4t_{ha}+4t_{c}+10t_{\text{sym}} \approx 30229.953856$	$3t_{\text{ha}}+4t_{\text{sym}} \approx 1086.43427$	31316.388126
Li et al. [22]	$8t_{\text{ha}}+4t_{\text{c}}+4t_{\text{sym}} \approx 19865.447712$	$3t_{\text{ha}}+4t_{\text{sym}} \approx 1086.43427$	20951.881982
Farash et al. [23]	$6t_{c}+8t_{ha} \approx 19441.447712$	$4t_{c}+4t_{ha} \approx 22332.05636$	41773.504072

Table 4. Execution time (ns) from [2]

Symbol	Definition	Computation cost	
		User	Server
T_{β}	Cost of sampling from χ_{β}	540.865 ns	72.503 ns
T_{sm}	Cost of scalar multiplication (component-wise) in $\mathbb{R}_q$	6.343 ns	0.259 ns
T_{mp}	Cost of multiplication (component-wise) in $\mathbb{R}_q$	12.087 ns	0.290 ns
T_{mpa}	Cost of multiplication (component-wise) and addition in $\mathbb{R}_q$	29.764 ns	2.349 ns
T_{cha}	Cost of computing g function	33.345 ns	0.645 ns
T_{h}	Cost of computing hash function	178.369 ns	13.457 ns

Table 5. Execution time of non-lattice based cryptographic operations (ns) from [2]

Symbol	Definition	Computation cost	
		User	Server
T_{bp}	Cost of bilinear pairing operation in $\Omega \times \Omega$	52533000 ns	8262000 ns
T_{exp}	Cost of raising power on a number in Ω_t	3437000 ns	417000 ns
T_{sm}	Cost of multiplying with a scalar in Ω	17909000 ns	2578000 ns
T_{pa}	Cost of adding two points in Ω	78000 ns	12000 ns
T_{c}	Cost of computing extended chaotic map	32400000 ns	5583000 ns

Table 6. SERVER CONFIGURATION

Components	Details
Processor	Intel(R) Core(TM) i7-6700 CPU @ 3.4 GHz
System Type	x64-based pc
Graphics	NVIDIA GeForce RTX 2060
RAM	8GB
Clock speed	1.00 GHz
Crypto Libraries	Miracle, Charm-crypto, NumPy, hash lib
Environment	Python 3.9.0
Cores	8
Operating System	Windows 10 Professional

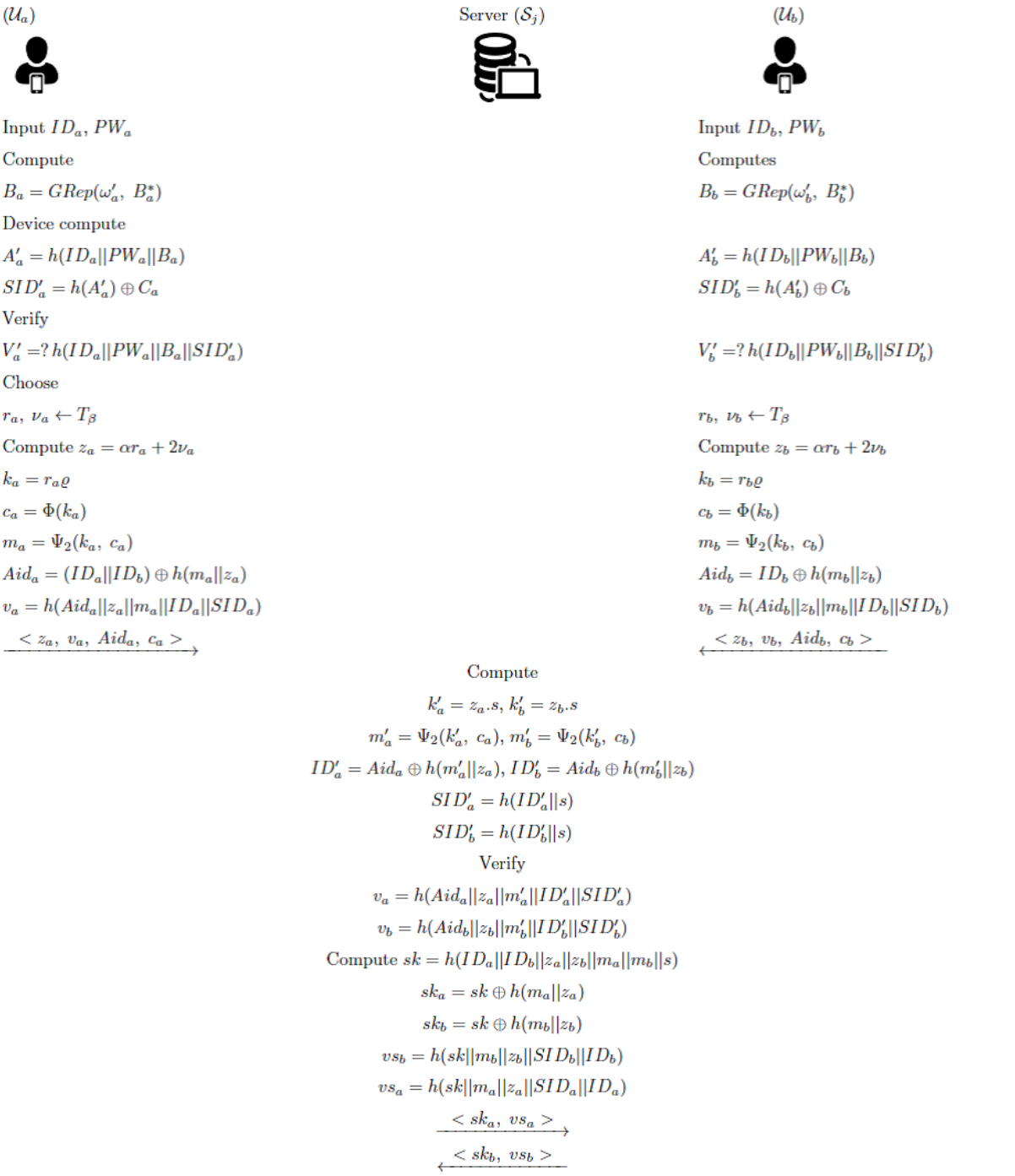

Fig 1. Illustration of logins and authentication phases

hash functions, onetime component-wise multiplication with addition in $\mathbb{Q}_p$, one-time component-wise scalar multiplication in $\mathbb{Q}_p$, one-time component-wise multiplication in $\mathbb{Q}_p$, one time g function hence the total cost is $8T_h + 2T_\beta + T_{sm} + T_{mpa} + T_{mp} + T_{cha} = 2655.405\ ns$, the S computes six times the hash functions. Consequently, $6T_h = 84.54\ ns$ is the execution time. Thus, the overall execution cost of Islam and Basu scheme [15] is $2655.405 + 2655.405 + 84.54 = 5395.35\ ns$. Table 7 provides the device configuration.

Table 7. DEVICE CONFIGURATION

Components	Details
Operating System	Android 4.3, MIUI 9.2
Android Version	4.3 (Jelly Bean)
CPU	Quad-core 1.6 GHz Cortex-A7
GPU	Adreno 305
Model	2014011
RAM	1.0 GB
Cores	4

A and B carry out $4T_h, 3T_c$ operations in the Farash and Attari scheme[?]. Thus, $8T_h + 6T_c = 19,44,01,086$ ns is the total cost. The overall cost of S's four-time T_c operations and $4T_h$ function is $2,23,32,056.36\ ns$. Thus, the Farash and Attari[23] system has an overall execution cost of $19,44,01,086 + 2,23,32,056.36 = 21,67,33,142\ ns$. Islam's[21] performs four hash operations for both A and B , and twice for T_c operations. As a result, the total cost is $8T_h + 4T_c = 12,96,01,447.712$ ns. Eight S functions are carried out. Based on this, the Islam scheme's overall communication cost is $12,96,01,447.712 + 112.72 = 12,96,01,560.432\ ns$. In the technique proposed by Yang et al.[24], both the users carry out $5T_{\text{exp}}$ operations in G_t and two hash functions. As a result, the total cost of execution is $4T_{\text{exp}} + 8T_h = 16,68,112.72\ ns$. Moreover, S executes $4T_{\text{exp}}$ in G_t and $8T_h$. Thus, $34,37,723.856 + 16,68,112.72 = 51,05,836.576\ ns$ is the overall execution cost of the H.Yang et al. system. The communication cost of an authentication protocol refers to the amount of data that needs to be exchanged between the communicating parties to establish and verify the identity of participants securely. It typically includes the size of messages exchanged, the number of message rounds, and any additional overhead associated with the authentication process. The size of messages exchanged during the authentication process impacts communication cost. Larger messages require more bandwidth and may incur higher transmission delays, especially in networks with limited bandwidth or high latency. Authentication protocols may include additional metadata or cryptographic signatures to ensure message integrity and authenticity. This overhead increases the size of messages and contributes to communication cost. The Table 8 contains an analysis of security attributes achieved by proposed, and existing protocols.

Table 8. Comparison of schemes

schemes →	(14)	(18)	(19)	(21)	(22)	(23)	Ours
features ↓							
Smart card stolen attack	✓	✓	✓	✓	✓	×	✓
User anonymity	×	×	✓	×	✓	×	✓
Man in the middle attack	✓	✓	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓
Impersonation attack	✓	✓	✓	✓	✓	×	✓
Perfect forward secrecy	✓	✓	✓	✓	✓	✓	✓
Replay attack	×	✓	×	✓	✓	✓	✓
Password guessing attack	×	×	✓	✓	✓	×	✓
Insider attack	×	✓	✓	✓	✓	✓	✓
Quantum computer attack	×	×	×	×	×	✓	✓

4 Conclusion

The proposed three-party post-quantum AKE protocol successfully addresses the critical security challenges posed by quantum computing while ensuring user privacy through anonymity and forward secrecy. By leveraging the **Ring Learning with**

Errors (RLWE) assumption, the scheme provides a robust mathematical foundation for secure communication across insecure channels.

Quantitative Evidence and Key Claims

- **Mathematical Reliability:** The correctness of the protocol is mathematically verified through the inequality $\left|k_a - k'_a\right| < \frac{q}{8}$. The error bound is specifically calculated as $\left|e \cdot r_a - \nu_a \cdot s\right| \leq 2\sqrt[3]{n} \cdot \delta^2 \Big| < \frac{q}{8}$, ensuring that the session key can be consistently reconstructed by all parties even in the presence of noise.
- **Security Efficiency:** The signal leakage attack recovery steps are significantly reduced from "2q" in earlier methods to "q + c" (where 0 < c < q), demonstrating a substantial improvement in identifying vulnerabilities compared to prior RLWE-based exchanges.
- **Performance Improvements:** Compared to existing schemes (such as those by Islam et al. and Park et al.), the proposed protocol achieves "High" efficiency in computing and communication costs. Unlike the scheme by Park et al., which relies on "expensive biometric hashing operations," this protocol optimizes cryptographic operations to maintain high-performance standards.

Explicit Novelty

- **Enhanced User Anonymity:** Unlike many predecessor protocols (e.g., Islam et al. 2021), this scheme integrates explicit user anonymity, ensuring that participants' identities remain hidden from third parties and each othere
- **Three-Factor Authentication:** The protocol incorporates a unique blend of passwords, identities, and biometric imprints processed via **Fuzzy Extractors** to thwart password-guessing and smartcard-loss attacks.
- **Resistance to Signal Leakage:** The design specifically mitigates "signal leakage attacks," a common flaw in previous lattice-based protocols that reuse public/private keys.

Actionable Recommendations

1. **Transition to RLWE Primitives:** Organizations currently relying on RSA or ECC for three-party communications should immediately evaluate a transition to RLWE-based primitives to ensure long-term "Perfect Forward Secrecy" against future quantum adversaries.
2. **Implementation of Fuzzy Extractors:** To improve mobile security, developers should utilize the **Fext (Fuzzy Extractor)** method for biometric data, as it allows for stable secret key generation from noisy physical measurements like fingerprints.
3. **Regular Key Refreshment:** Although the protocol provides forward secrecy, sessions should still be terminated and shared keys discarded after specified periods to maintain maximum data integrity and prevent unauthorized access.

References

1) Chris P. Lattice cryptography for the internet;vol. 6. 2014.
2) Jiang Z, Zhenfeng Z, Jintai D, Michael S, Özgür D. Authenticated key exchange from ideal lattices;vol. 34. 2015.
3) Daniel K, Bradley CL, John M, Mark M, Jerome AS, David T. Failure is not an option: Standardization issues for post-quantum key agreement. *Workshop on Cybersecurity in a Post-Quantum World*. 2015;p. 21. https://doi.org/10.1145/3634737.3637661.
4) Scott F. Cryptanalysis of ring-lwe based key exchange with key share reuse. *Cryptology ePrint Archive*. 2016. https://eprint.iacr.org/2016/085.
5) Jintai D, Saed A, RV S, Scott F, Xiaodong L. Leakage of signal function with reused keys in RLWE key exchange. 2017.
6) Jintai D, Scott F, Saraswathy R. Complete attack on RLWE key exchange with reused keys, without signal leakage;vol. 23. 2018.
7) Dabra V, Bala A, Kumari S. Lbapake: Lattice-based anonymous password authenticated key exchange for mobile devices. *IEEE Systems Journal*. 2020;15(4):5067–5077. https://doi.org/10.1109/JSYST.2020.3023808.
8) Dharminder D, Chandran KP. Lwesm: learning with error based secure communication in mobile devices using fuzzy extractor. *Journal of Ambient Intelligence and Humanized Computing*. 2020;11(10):4089–4100. https://doi.org/10.1007/s12652-019-01675-7.
9) Islam SH. Provably secure two-party authenticated key agreement protocol for postquantum environments. *Journal of Information Security and Applications*. 2020;52:102468. https://doi.org/10.1016/j.jisa.2020.102468.
10) Dabra V, Bala A, Kumari S. Flaw and amendment of a two-party authenticated key agreement protocol for post-quantum environments. *Journal of Information Security and Applications*. 2021;61:102889. https://doi.org/10.1016/j.jisa.2021.102889.
11) Wang Q, Wang D, Cheng C, He D. Quantum2fa: efficient quantum-resistant two-factor authentication scheme for mobile devices. *IEEE Transactions on Dependable and Secure Computing*. 2021. https://doi.org/10.1109/TDSC.2021.3129512.
12) Dharminder D, Reddy CB, Das AK, Park Y, Jamal SS. Post-quantum lattice-based secure reconciliation enabled key agreement protocol for IoT. *IEEE Internet of Things Journal*. 2022;10(3):2680–2692. https://doi.org/10.1109/JIOT.2022.3213990.
13) Kumar U, Garg M, Kumari S, Dharminder D. A construction of post quantum secure and signal leakage resistant authenticated key agreement protocol for mobile communication. *Transactions on Emerging Telecommunications Technologies*. 2023;34(1):e4660. https://doi.org/10.1002/ett.4660.
14) Islam SH, Basu S. Pb-3paka: Password-based three-party authenticated key agreement protocol for mobile devices in postquantum environments. *Journal of Information Security and Applications*. 2021;63:103026. https://doi.org/10.1016/j.jisa.2021.103026.

15) Rewal P, Singh M, Mishra D, Pursharthi K, Mishra A. Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices. *Journal of Information Security and Applications*. 2023;75:103505. https://doi.org/10.1016/j.jisa.2023.103505.
16) Kumar U, Garg M, Chaudhary D. Design and analysis of a postquantum secure three party authenticated key agreement protocol based on ring learning with error for mobile device. *The Journal of Supercomputing*. 2025;81(1):1–28. https://doi.org/10.1007/s11227-024-06467-1.
17) Park H, Son S, Park Y, Park Y. Provably quantum secure three-party mutual authentication and key exchange protocol based on modular learning with error. *Electronics*. 2024;13(19). https://doi.org/10.3390/electronics13193930.
18) Zheng Y, Hu S, Wei L, Chen Y, Wang H, Yang Y, et al. Design and analysis of a security-enhanced threeparty authenticated key agreement protocol based on chaotic maps. *IEEE Access*. 2020;8:66150–66162. https://doi.org/10.1109/ACCESS.2020.2979251.
19) Xie Q, Lu Y, Tan X, Tang Z, Hu B. Security and efficiency enhancement of an anonymous three-party password authenticated key agreement using extended chaotic maps. *PLOS ONE*. 2018;13(10):e0203984. https://doi.org/10.1371/journal.pone.0203984.
20) Chaudhary D, Kumar U, Saleem K. A construction of three party post quantum secure authenticated key exchange using ring learning with errors and ecc cryptography. *IEEE Access*. 2023. https://doi.org/10.1109/ACCESS.2023.3325886.
21) Islam SH. Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. *Information Sciences*. 2015;312:104–130. https://doi.org/10.1016/j.ins.2015.03.050.
22) Li X, Niu J, Kumari S, Khan MK, Liao J, Liang W. Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol. *Nonlinear Dynamics*. 2015;80:1209–1220. https://doi.org/10.1007/s11071-015-1937-0.
23) Farash MS, Attari MA. An efficient and provably secure three-party password-based authenticated key exchange protocol based on chebyshev chaotic maps. *Nonlinear Dynamics*. 2014;77:399–411. https://doi.org/10.1007/s11071-014-1304-6.
24) Yang H, Zhang Y, Zhou Y, Fu X, Liu H, Vasilakos AV. Provably secure three-party authenticated key agreement protocol using smart cards. *Computer Networks*. 2014;58:29–38. https://doi.org/10.1016/j.comnet.2013.08.020.