IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 April 2024, accepted 18 May 2024, date of publication 22 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404232

RESEARCH ARTICLE

Anonymous Quantum Safe Construction of Three Party Authentication and Key Agreement Protocol for Mobile Devices

DHARMINDER CHAUDHARY[1], (Member, IEEE), PRADEEP KUMAR DADSENA[2], (Member, IEEE), A. PADMAVATHI[1], (Member, IEEE), MOHAMMAD MEHEDI HASSAN[3], (Senior Member, IEEE), BADER FAHAD ALKHAMEES[3], (Member, IEEE), AND UDDESHAYA KUMAR[4]

[1]Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai 601103, India
[2]Department of Mathematics, Government Engineering College at Jagdalpur, Jagdalpur 494001, India
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Mathematics, Gautam Buddha University, Noida 201312, India

Corresponding author: Dharminder Chaudhary (manndharminder999@gmail.com)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2024R493.

ABSTRACT Once the shared secret key is established, three parties can use it for secure communication using symmetric-key encryption AES (128, 192, 256) algorithms or other cryptographic primitives. Although there are few third-party post-quantum authentication and key agreement protocols exist, but the recent studies in this article show that they are not anonymous or cannot provide forward secrecy. Most of the existing protocols enable adversaries to trace the source of messages. Many of third-party AKA schemes based on conventional public-key cryptosystems are vulnerable to quantum computers. Therefore, this paper contains a forward secure three-party post-quantum authenticated key establishment protocol for mobile devices. The proposed three-party key exchange protocol establishes an authenticated shared key that can be periodically refreshed to maintain forward secrecy. This protocol enables two parties to establish a shared session key even in the presence of quantum adversaries and enables them to communicate confidentially and securely over insecure networks. The protocol is anonymous as both the parties communicate using masked dynamic identities. A contrast study consisting of performance and security assessment is presented, which illustrates the suggested design is more applicable.

INDEX TERMS Key establishment, authentication, ring learning with errors, cryptography.

I. INTRODUCTION

Three-party post-quantum key exchange protocols enable three parties to establish a shared secret key securely, even in the presence of quantum adversaries. These protocols are crucial for scenarios where three entities need to communicate securely over an insecure network. Converting a two-party key exchange protocol into a three-party key exchange protocol involves extending the protocol to accommodate an additional party while maintaining the security properties of the original scheme. Start with a well-established two-party key exchange protocol, such as Diffie-Hellman key exchange, or a post-quantum secure protocol like NTRUEncrypt or SIDH. In the original protocol, two parties (A and B) exchange messages to establish a shared secret key. To extend the protocol to support three parties (A, B, and C), introduce a mechanism for party A to communicate securely with both parties B and C simultaneously. One approach is to modify the original protocol to allow party A to generate separate shared secret keys with parties B and C, which are then combined to form a single shared secret key among all three parties. Alternatively, employ a distributed key

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

generation scheme where parties A, B, and C collaboratively generate a shared secret key without requiring party A to communicate separately with parties B and C. Ensure that the extended protocol maintains the security properties of the original two-party protocol, such as confidentiality, integrity, and resistance to attacks. Many of researchers [1], [17], [23], [25] have tried to design three-party post-quantum key exchange protocols that enable three parties to establish a shared secret key securely, but they are not able to achieve all security attributes (mainly anonymity, forward secrecy, nontraceability). Therefore, it's highly challenging to design a three-party authenticated post-quantum key agreement protocol that enables three parties to establish anonymous and forward secure communication.

II. RELATED WORK

Ding et al. [9] is the first to use the assumption learning with error for designing key exchange. Their key exchange protocol were found more efficient in terms of communication and computation costs compared with key exchange based on encryption schemes. Furthermore, they extended their idea to the ring learning with errors (RLWE) assumption, which results in a small key size and better efficiency. But, their protocol is Diffie-Hellman-like protocol and does not provide authenticity. In the year 2014, Peikert [24] proposed a straightforward, low-bandwidth reconciliation method that enables two parties to ''approximately agree'' on a secret value to come to a precise agreement constitutes one of our primary technological advancements. However, their method is based on encryption technologies based on the LWE problem, and it takes more cost. This method is actually key encapsulation and cannot provide authenticity to users. The difference between [9] and [24] is that the first one uses a biased, and the second one uses an unbiased reconciliation mechanism. In the next year 2015, Zhang et al. [31] proposed an advanced authentication and key establishment protocol. However, their protocol constantly utilizes the long-term server key while establishing communication. It culminates in the widely recognized signal leaking attack. Kirkwood et al. [18] were the first who observed that using secret key more than once is dangerous in designing Ding like key exchange based on ring learning with errors. This idea was a new open track for the researchers, but it lacks in a complete description. In 2016, Fluhrer [12] presented a novel and inventive concept of key mismatch for ring learning with error-based protocols that update public and private keys. The genuine party'sparty's private key can be retrieved using this concept. In order for two parties to retrieve a particular key, the match or mismatch attributes are crucial. However, this attack will not work when the two parties agree on the key using the least significant bits of the same binary length keys. Again in the year 2016, Stebila and Mosca [26] discussed efficiency and safety features individually as well as in relation to the Transport Layer Security (TLS) protocol. They presented the Open Quantum Safe project, an open-source software project for quantum-resistant cryptography prototyping. It consists of our combinations of liboqs, a C library of quantum-resistant techniques, into widely used open-source protocols that are and programs, such as the OpenSSL library. Taking idea [12], in 2017, Ding et al. [7] introduced the idea of signal leakage attack for ring learning with error-based key exchange reusing public/private keys. This paper [7] provides an insight into how a long-term publicly accessible keys reuse in RLWE key exchange protocols might be assaulted, and it is prompted by an attack described in [7]. By starting numerous sessions with the truthful party and examining the signal function's results, this work focuses on the attack on the authenticated key exchange protocols. Ding et al. [8] discussed that an intruder using a faulty key initiates a series of key exchange sessions, and the attack known as ''signaling leakage'' depends on changes in the signal supplied by the respondent using his key. In their work, they offered a new assault on Ding's one pass case that simply used the data of whether both sides agreed on the final key instead of depending on the signal function output. They have also discussed that the existing signal leakage attack can be performed with less number of steps and how it can be extended to Peikert's key exchange. In 2018, Feng et al. [11] suggested an anonymous authenticated key exchange mechanism for mobile phones and tablets. For smartphones and tablets, the protocol is intuitive, straightforward, and effective, although it is susceptible to spoofing, manipulation-based, signal leakage, and it cannot provide anonymous communication for the users. In 2020, Dabra et al. [2] analyzed the security of Feng et al. [11], and they discussed signal leakage attack on the protocol [11]. Dabra et al. cites dabra2020lba proposed an ideal lattice-based anonymous password-authenticated key exchange protocol has been proposed for mobile devices to address the aforementioned issues [11]. Additionally, their protocol supports the features of anonymous communication and complete forward secrecy. In 2020, Dharminder and Chandran [4] proposed learning with errors-based anonymous authentication protocol using ideal in some lattice. But, Dabra et al. [2] supports weak login and authentication phase that results to denial of service. Islam [15] proposed quantum-safe two-party authentic key agreement, but their protocol [15] was susceptible to signal leakage [3]. Wang et al. [28] also introduced an efficient post quantum two factor authentication key agreement fot mobile devices. Their protocol contains three messages of exchange that create communication overheads. In 2022, Dharminder et al. [5] proposed a post quantum reconciliation enabling key exchange for the Internet of Things environment. In 2023, Kumar et al. [19] proposed a quantum-safe key agreement based on a variant of ideal lattice assumption, the ring learning errors. This protocol just requires two messages in exchange for an authenticated key agreement. But, all the above protocols are two-party authenticated key agreements. A two-party authentication protocol involves two entities (often referred to as Alice and Bob) exchanging messages to authenticate each other.

However, when a third party, traditionally referred to as Charlie, is introduced into the authentication process, it becomes a three-party authentication protocol. In a three-party authentication protocol, Charlie plays a crucial role as a trusted intermediary, facilitating the authentication process between Alice and Bob. The involvement of a third party adds an additional layer of security and trust, especially in scenarios where direct communication between Alice and Bob is not feasible or secure. To fill this gap, Xu et al. [30] and Liu et al. [22] proposed a provably secure three-party password-authenticated key exchange protocol based on ring learning with error. But, their password-based protocol is not secure because password guessing is very easy. In the next year, Islam and Basu [17] first put forward a new three-party password-authenticated key exchange protocol based on ring learning with error. They have also shown that the proposed protocol is safe in the post-quantum world based on the hardness assumption of the ring-learning-with-errors. In 2023, Chaudhary et al. [1] proposed a three-party keu agreement using ECC cryptography. Their protocol ensures authentication, anonymity, and forward secrecy. They also analyzed that the approach proposed by Islam and Basu [17] does not guarantee anonymous communication. It is susceptible to impersonation, password guessing, and smart card theft attacks. In the year 2023, Rewal et al. [25] proposed a lattice-based authenticated key establishment protocol to resist attacks like password guessing and increase efficiency. They have done a comparison analysis, evaluation of performance, and security assessment to prove the suitability of the suggested design. But, their scheme [25] does not provide anonymous communication, and secret value is directly xored with a biometric value, which means it can't provide three-factor security. Therefore, designing a three-party post-quantum key agreement protocol that satisfies existing security attributes and requirements is challenging.

III. MOTIVATION AND CONTRIBUTION

Designing a three-party post-quantum key agreement protocol addresses specific scenarios and requirements where communication involves three distinct entities (see model in Fig. (1)). In many real-world scenarios, communication involves more than two parties. For example, in group messaging applications, collaborative environments, or distributed systems, three or more parties need to establish a shared secret key to communicate securely. As quantum computing capabilities advance, it is essential to develop post-quantum secure key agreement protocols that can withstand quantum attacks. Designing three-party protocols with post-quantum cryptographic primitives ensures long-term security against quantum adversaries. Designing three-party post-quantum key agreement protocols contributes to ongoing research and innovation in the field of cryptography. It allows researchers to explore new cryptographic techniques, protocols, and security models tailored to multi-party communication scenarios. But, the researchers need help with developing efficient three-party post-quantum key

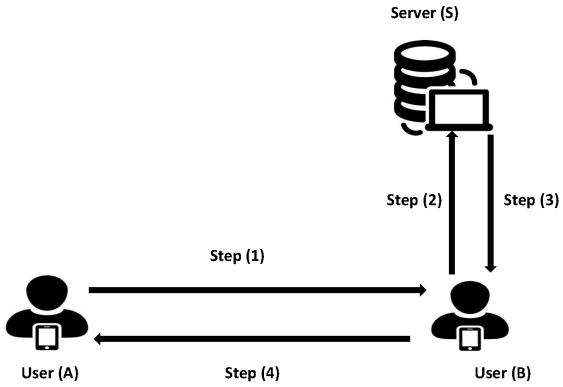

FIGURE 1. Steps involved in three-party post-quantum key agreement protocol.

agreement protocols that meet the specific requirements for mobile devices. Secondly, there exist few post-quantum authenticated key establishment protocols namely [1], [17], [23], [25], but either they are not anonymous [17], [23], [25] or their computation cost is very high [1]. Furthers these protocols [17], [23], [25] cannot provide forward secrecy. Therefore, this paper contains a three-party post-quantum key agreement protocol to fill the gaps.

- Anonymous communication refers to the ability to send messages or interact with others without revealing one's identity or other identifying information. This concept is important in various contexts, including privacy protection, freedom of speech, whistleblowing, and political activism. The proposed three-party post-quantum key agreement protocol provides anonymous communication.
- The proposed post-quantum key agreement protocol offers several advantages over traditional cryptographic algorithms, particularly in light of the potential future development of quantum computers capable of breaking many currently used cryptographic schemes. Overall, the adoption of post-quantum key exchange protocols offers a proactive and strategic approach to addressing the security challenges posed by quantum computing advancements, ensuring the continued confidentiality, integrity, and availability of sensitive information in the digital age.
- Forward secrecy, also known as perfect forward secrecy (PFS), is a property of cryptographic protocols that ensures that past communication sessions remain secure even if long-term secret keys are compromised in the future. In other words, forward secrecy provides protection against retroactive decryption of encrypted communication. The proposed post-quantum key agreement protocol offers forward secrecy, and the session key for each session is independent of previous sessions.
- There are many post-quantum key agreement protocols with varying numbers of message exchanges. In practice, the efficiency and security of a key exchange protocol depend on factors such as the desired security

properties, computational complexity, and network constraints. The proposed post-quantum key agreement protocol needs four messages of exchange, which is minimal message exchanges as compared to state-of-art.

IV. ROADMAP OF THE PAPER

In this paper, section (I) is about the introduction to the Internet of Drones and the need for security against quantum computers. The section (II) contains related work to analyze the work in this area, and the section (III) is all about motivation and contribution. Section (V) contains basic notations and definitions, and section (VI) is all about the proposed work. Section (VII) and Section (VIII) contain security analysis of the work in the random oracle, section (IX) analyzes the performance of the protocol with the cost of used operations, and Section (X) concludes the paperwork.

V. PRELIMINARIES

Before delving into the specifics of designing three-party post-quantum key agreement protocols, it's essential to establish some preliminaries. This section provides the necessary definitions of technical terms used in the context of this study. Post-quantum cryptography refers to cryptographic algorithms and protocols designed to resist attacks from quantum computers. Unlike classical cryptography, post-quantum cryptography relies on mathematical problems that are believed to be hard even for quantum computers to solve efficiently, such as lattice-based cryptography. Certain computational problems underlying the chosen post-quantum cryptographic primitives are hard to solve efficiently, even for classical or quantum computers. The security of the protocol relies on the computational hardness of specific mathematical problems, such as lattice problems, short integer solution, laerning with error, or ring learning with error. These assumptions provide a framework for analyzing the security and functionality of three-party post-quantum key agreement protocols and guide the design choices to ensure their effectiveness in practical deployment.

A. LEARNING WITH ERROR

The Learning with Errors (LWE) assumption is a central concept in modern cryptography, particularly in the realm of lattice-based cryptography. LWE is a computational problem that forms the basis for many cryptographic constructions.

Definition 1: In the LWE problem, you're given pairs of the form $(\alpha,\ \alpha.s+e)$, where "α" is a randomly chosen vector, "s" is a secret vector, and "e" is a small random noise vector. The challenge is to recover the secret vector "s" from these pairs.

The security of many cryptographic schemes based on LWE relies on the assumption that it's computationally hard to distinguish these pairs from random noise. This assumption forms the foundation for various cryptographic primitives like encryption, key exchange, and digital signatures. LWE is often used in constructing post-quantum secure cryptographic systems, as it's believed to be resistant to attacks from quantum computers. However, as with any cryptographic assumption, its security depends on the specific parameters chosen and the efficiency of known algorithms for solving it.

B. RING LEARNING WITH ERRORS

Ring Learning with Errors (RLWE) is an extension of the Learning with Errors (LWE) problem, but it involves polynomial rings rather than vector spaces. In RLWE, instead of working with vectors, you work with polynomials modulo some polynomial a prime number.

Definition 2: The Ring Learning with Errors problem involves given samples $(\alpha,\ \alpha.s+e)$, where "α" is a randomly chosen polynomial, "s" is a secret polynomial, and "e" is a small random noise polynomial. The task is to recover the secret polynomial "s" from these pairs.

RLWE is particularly significant in lattice-based cryptography because it provides a hard problem believed to be resistant to attacks by both classical and quantum computers. This makes it a prime candidate for constructing post-quantum secure cryptographic schemes. In the context of ideal lattices, RLWE is often defined over the ring of integers modulo some ideal in a number field. This allows for more efficient implementations and provides additional mathematical structure that can be exploited in cryptographic constructions.

C. BASIC TERMINOLOGY

The protocol uses lattice based assumption "Ring Learning with Error (RLWE)" that is assumed to be secure against quantum attacks. Thereore, the reader need to understand some basic mathematics, and notations involved in the design of the protocol. Let us consider $q > 2$ to be the random prime number, R to be the set of real numbers and Z to be the set of integers, respectively. Ring of polynomial are $Z[x]$ and $Z_q[x]$ having integers coefficients, and integer modulo (q) coefficients respectively. Let security parameter n is chosen such that $n = 2^\ell$, where $\ell \in Z$. Let us consider an irreducible polynomial $x^n+1 \in Z[x]$, and polynomial rings $\mathbb{K} = \frac{Z[x]}{<x^n+1>}$ and another polynomial ring $\mathbb{K}_q = \frac{Z_q[x]}{<x^n+1>}$ with integer coefficients reduced modulo (q), and finite degree. As a result the element of ring Q is defined as $a = \sum_{i=0}^{n-1} a_i x^i \in \mathbb{K}$. We define two types of norms; (1) norm L_2 and (2) L_∞, defined as $||a||_2 = \{\sum_{i=0}^{n-1} a_i^2\}^{\frac{1}{2}}$ and $||a||_\infty = max|a_i|$ for $0 \leq i \leq n-1$. To define learning with error assumption, we need a system of equations peturbed with error whose coefficients are small integers. The discrete Gaussian distribution Y_β is also defined over $\mathbb{K}_q$ and $\beta > 0$ represents some random fixed real number called standard deviation of the Gaussian distribution.

Lemma 1: For any $x,\ y \in \mathbb{K}$ there exist following inequalities is defined as $||x.y||_a \leq n^b ||x||_a.||y||_a$ where $(a,\ b) \in \{(2,\ \frac{1}{2}),\ (\infty,\ 1)\}$ satisfies [6].

Lemma 2: For some numbers $\beta = \omega(\sqrt{\log_2 n})$ and $\alpha = \beta\sqrt{n}$, the inequality $pr_{d \leftarrow X_\beta}[||d||_2 > \alpha] \leq \frac{2}{2^n}$ holds true.

Let $T = \{-\lfloor \frac{q}{4} \rfloor, \ldots, \lfloor \frac{q}{4} \rfloor\}$ be the half middle subset of $Z_q = \{-\frac{(q-1)}{2}, \ldots, \frac{(q-1)}{2}\}$. We have characteristic function $\psi(.)$ defined on compliment of $T = \{-\lfloor \frac{q}{4} \rfloor, \ldots, \lfloor \frac{q}{4} \rfloor\}$, that is defined as $\psi(z) = 0$ if $z \in T$, and $\psi(z) = 1$ otherwise. The auxiliary modular function $\Phi_2 : Z_q \times \{0, 1\} \rightarrow \{0, 1\}$ is defined as $\Phi_2(u, d) = (u + d.\frac{(q-1)}{2}) \text{mod (q) mod (2)}$, such that $u \in Z_q$ and $d = \Psi(u)$, respectively.

Lemma 3: For some given number q and the elements $u, e \in Q_q$ such that there exist $|e| < q/8$ and we have $\Phi_2(u, \Psi(u)) = \Phi_2(w, \Psi(u))$ where w is defined as $w = u + 2.e$.

Following the above concept, one can do the generalization of the functions ψ and Φ_2 over $\mathbb{K}_q$. Let us assume an element $d = d_0 + d_1 x + \ldots + d_{n-1}x^{n-1} \in \mathbb{K}$, which can be used to represent the vector d. This vector d will be defined as follows $d = (d_0, d_1, \ldots, d_{n-1})$. Similarly for another vector u is defined as $u = (u_0, u_1, \ldots, u_{n-1}) \in \{0, 1\}^n$ The auxiliary and characteristic functions can be written in the generalized format as follows $\Phi(d) = (\Phi(d_0), \Phi(d_1), \ldots, \Phi(d_{n-1}))$ and $\Phi_2(d, u) = (\Phi_2(d_0, u_0), \Phi_2(d_1, u_1), \ldots, \Phi_2(d_{n-1}, u_{n-1}))$. As stated by the RLWE problem, for a polynomial times known pairs $(\alpha, b = \alpha.s + e) \in \mathbb{K}_q \times \mathbb{K}_q$ in the rings, it is practically infeasible to search for the unknown $s, e \in \mathbb{K}_q$ inside the ring which were selected from the Gaussian Distribution over $\mathbb{K}_q$, and naturally induced over Z^n.

Definition 3: Let us consider $A_{s, Y_\delta} = (u_i, v_i)$ to be some sample from $\mathbb{K}_q \times \mathbb{K}_q$ where u_i is taken uniformly from $\mathbb{K}_i$ and $v_i = u_i.s + e$, such that $s, e \rightarrow Y_\delta$. The RLWE problem refers to the way that differs elements A_{s, Y_β} from the uniform random distribution on the finite ring $\mathbb{K}_q$, where there are many polynomial samples.

Definition 4: Pairing with error Problems or PWE problems: There is a function named $\gamma : \mathbb{K}_q \times \mathbb{K}_q \rightarrow \{0, 1\}$ is such that it is defined by $\gamma(u, s) = \Phi_2(u.s, \Phi(u.s))$. The goal of the PWE for the given parameters $u, v, c \leftarrow \mathbb{K}_q$ is to find the $\gamma(u, s)$ for the unknown $s, e' \in Y_\delta$ where $v = c.s + 2.e'$ are defined.

Definition 5: Decision pairing with error problem or DPWE problem. For some given parameters $d, y, z, c \in \mathbb{K}_q$ the DPWE problem aims to check if (d, z) where $d = y.s + 2.g$ and $z = c.s + 2.e$ are defined uniformly random in $\mathbb{K}_q \times \mathbb{K}_q$ where some unknowns are also there like $s, g, e \in Y_\delta$.

VI. PROPOSED ANONYMOUS AND FORWARD SECURE QUANTUM-SAFE THREE PARTY AUTHENTICATED KEY ESTABLISHMENT PROTOCOL

Three-party post-quantum key exchange protocols enable three parties to establish a shared secret key securely, even in the presence of quantum adversaries. These protocols are crucial for scenarios where three entities need to communicate securely over an insecure network. This section is consisting of four phases; (1) initialization, (2) user (A) registration, (3) user (B) registration, and (4) login and authentication phases.

TABLE 1. Notations table.

Notation	Description
S	Server
U_i for $i \in \{A, B\}$	User
id_{U_i} for $i \in \{A, B\}$	User's IDs
pw_{U_i} for $i \in \{A, B\}$	Passwords
bio_{U_i}	User's Biometric Data
D_{U_i} for $i \in \{A, B\}$	Device specific user data
r_{U_i}, e_{U_i} for $i \in \{A, B\}$	Random values
$x_{U_i}, t_{U_i}, w_{U_i}, k_{U_i}, \sigma_{U_i}$ for $i \in \{A, B\}$	Computed values
pid_{U_i} for $i \in \{A, B\}$	User's Masked Identity
$s\kappa, s\kappa'$	Session Keys
H	Hash function
Y_δ	Gaussian distribution
a	$a \in \mathbb{K}_p$ publicly known polynomial
s	S secret key
$\vartheta = a \cdot s + 2 \cdot e$	S public key

A. INITIALIZATION PHASE

The initialization phase of a control server (CS) involves setting up and preparing the server for operation. This phase is crucial for ensuring that the server is properly configured, secure, and ready to manage the intended tasks. Here's a general outline of the initialization phase for a control server:

1) The process involves choosing an element at random from the ring $\mathbb{K}_p = \frac{\mathbb{Z}[x]}{\langle x^n+1 \rangle}$, a discrete Gaussian distribution Y_δ, an odd prime number $p > (4y(\sqrt[n]{3}))^2$, and an integer $n \in \mathbb{Z}_p$ such that $p \mod (2n) = 1$.
2) Choose a collision resistant hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^\ell$, in which ℓ represents the output of fixed length. The CS chooses $\alpha, s \in \mathbb{K}_p$, and $e \in Y_\delta$, and it cpmputes public key $\varrho = \alpha \cdot s + 2 \cdot e$. Fianlly, the parameters $\{n, p, \alpha, \varrho, Y_\delta, SHA2 - 256(.)\}$ are published in a public domain while keeping the secret key s safe.

B. USER REGISTRATION

To register on a server, you typically need to follow specific instructions provided by the server administrator or outlined in the server's documentation. This might involve creating an account with a username and password, providing an email address, or completing other registration steps. The user U_i for $i \in \{A, B\}$ establishes a secure channel to complete registration with control server S. The user follows the following steps to complete the registration phase.

C. USER REGISTRATION PHASE

User U_i for $i \in \{A, B\}$ register with the server S by providing their masked password and identifying information during the following registration steps:

1) *RG1:* User U_i for $i \in \{A, B\}$ selects a unique identity id_i for $i \in \{A, B\}$, a (preferably strong) password pw_i for $i \in \{A, B\}$, and chooses secrets $n_i \in Z$ for $i \in \{A, B\}$ randomly. After that, U_i computes $z_i = Sha - 256(id_i||pw_i||n_i)$ and sends the information $\{id_i, z_i\}$ to the sever S securely.
2) *RG2:* After receiving the registration request from U_i, the server S first checks its database against $Sha - 256(id_i)$. If it is already in the database, it rejects the registration request; otherwise, it computes

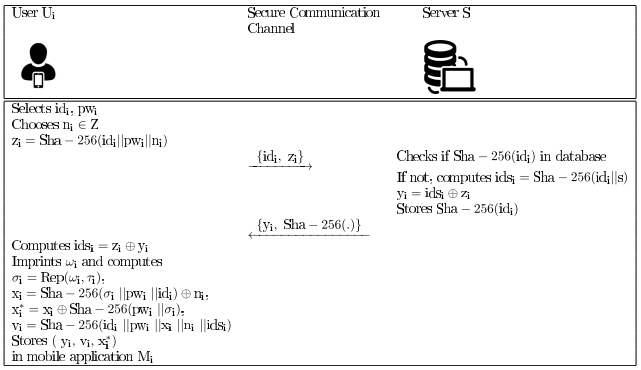

FIGURE 2. Summary of the user registration phase.

$ids_i = Sha-256(id_i||s)$ with the help of its own master secret key s. After computing $y_i = ids_i \oplus z_i$, S stores $Sha-256(id_i)$ in its database and sends $\{y_i, Sha-256(\cdot)\}$ to U_i via secure channel.

3) *RG3:* After getting the message from S, U_i computes $ids_i = z_i \oplus y_i$ and imprints his/her biometric ω_i at sensor of the mobile device and generates a biometric secret σ_i and its associated public reproduction parameter τ_i using fuzzy extractor probabilistic generation function $Gen(\cdot)$ as $\omega_i = \{\sigma_i, \tau_i\}$. Next, U_i computes $x_i = Sha-256(\sigma_i||pw_i||id_i) \oplus n_i$, $x_i^* = x_i \oplus Sha-256(pw_i||\sigma_i)$ and a verification value $v_i = Sha-256(id_i||pw_i||x_i||n_i||ids_i)$ for the user. Finally, U_i stores the registration credentials $(y_i, v_i, x_i^*, Sha-256(.)$ into his/her mobile application. Figure 2 presents the summary of the user registration phase.

D. LOGIN AND AUTHENTICATION PHASE

This section contains a description of the procedures involved in the authentication phase (Figure (3)). The login and authentication phases are critical components of the user authentication process in computer systems. These phases ensure that only authorized users can access the system or application. Here's an overview of the login and authentication phases:

1) The user U_A gives inputs id_A, pw_A and imprints ω_A. The device computes $n_A = x_A \oplus Sha-256(\sigma'_A||pw_A||id_A)$, and the value $x_A = x_A^* \oplus Sha-256(pw_A||\sigma'_A)$. Furthers it computes $n_A = x_A \oplus Sha-256(\sigma_A||pw_A||id_A)$, and masked $ids'_A = y_A \oplus Sha-256(id_A||pw_A||n_A)$ for the user. Furthers it computes the value $v'_A = Sha-256(id_A||pw_A||x_A||n_A||ids'_A)$, and verifies $v_A = v'_A$ to get logged into the device.
2) After getting successful login, U_A chooses r_A, $v_A \leftarrow Y_\delta$, computes $z_A = \alpha r_A + 2v_A$, $k_A = r_A Q$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A, c_A)$, and a dynamic masked identity $aid_A = (id_A||id_B) \oplus Sha-256(m_A||z_A)$ along with verification factor $v_A = Sha-256(aid_A||z_A||m_A||id_A||id_B)$ to check the values z_A, v_A, aid_A, c_A, sent to U_B are correct.
3) The user U_B receives z_A, v_A, aid_A, c_A,, then he gives input id_B, pw_B and imprints ω_B. The device computes $n_B = x_B \oplus Sha-256(\sigma'_B||pw_B||id_B)$, $x_B = x_B^* \oplus Sha-256(pw_B||\sigma'_B)$, $n_B = x_B \oplus Sha-256(\sigma_B||pw_B||id_B)$, and masked identity $ids'_B = y_B \oplus Sha-256(id_B||pw_B||n_B)$ for the user. Furthers it computes the value $v'_B = Sha-256(id_B||pw_B||x_B||n_B||ids'_B)$, and verifies $v_B = v'_B$ to get logged into the device.
4) The user U_B samples random r_B, $v_B \leftarrow Y_\delta$, and it computes $z_B = \alpha r_B + 2v_B$, $k_B = r_B Q$, $c_B = \Psi(k_B)$, $m_B = \Phi_2(k_B, c_B)$, $\kappa_B = Sha-256(m_B||z_B)$, and derives masked identity $aid_B = e_{\kappa_B}(aid_A, id_B)$, and a verification factor $v_B = Sha-256(aid_A||z_A||m_A||z_B||m_B||id_A||id_B)$ to validate the values z_A, v_B, aid_B, c_A,, z_B, c_B, sent over public channel to the server S.
5) The server receives information z_A, v_B, aid_B, c_A,, z_B, c_B, it computes $k'_A = z_A.s$, $m'_A = \Phi_2(k'_A, c_A)$, $k'_B = z_B.s$, $m'_B = \Phi_2(k'_B, c_B)$, $\kappa_B = Sha-256(m_B||z_B)$, and computes masked identity for user U_A, and real identity of user U_B by decryption as $(aid_A, id_B) = d_{\kappa_B}(aid_B)$, then it computes real identity of user U_A $(id_A||id_B) = aid_A \oplus Sha-256(m_A||z_A)$, and estimates this request is sent for id_B. Furthermore, it verifies the received values by equating $v_B = Sha-256(aid_A||z_A||m_A||z_B||m_B||id_A||id_B)$, and computes a dynamic session key $s\kappa = s\kappa_A = s\kappa_B = Sha-256(id_A||id_B||z_A||m_A||z_B||m_B||s)$, and masked session keys $s\kappa_B^* = (s\kappa, id_A) \oplus Sha-256(m_B||z_B||id_B)$, and $s\kappa_A^* = s\kappa \oplus Sha-256(m_A||z_A||id_A||id_B)$ along with verification factor $v_B^* = Sha-256(z_B||m_B||id_A||id_B||s\kappa||s\kappa_A^*)$, $s\kappa_A^*$, $s\kappa_B^*$, v_B^*, and sends it to U_B.
6) The user U_B computes session key as $s\kappa_B^* \oplus Sha-256(m_B||z_B||id_B) = (s\kappa, id_A)$, and estimates this session key is to establish a session with user U_A, and he also verifies the validity of session key by equating $v_B^* = Sha-256(z_B||m_B||id_A||id_B||s\kappa||s\kappa_A^*)$, and computes a verification factor for U_A as $v_A^* = Sha-256(id_A||id_B||z_A||s\kappa||s\kappa_A^*)$, and sends v_A^*, $s\kappa_A^*$ to U_A.
7) The user U_A receives the values v_A^*, $s\kappa_A^*$, and computes session key as $s\kappa_A^* \oplus Sha-256(m_A||z_A||id_A||id_B) = s\kappa$, and verifies the validity by equating $v_A^* = Sha-256(id_A||id_B||z_A||s\kappa||s\kappa_A^*)$.

E. PROOF OF CORRECTNESS

The proof of correctness of a cryptographic scheme demonstrates that the scheme achieves its intended security goals and functionality according to its design specifications. The proof typically involves formal mathematical reasoning and analysis to establish that the scheme operates as intended and provides the desired security properties.

Theorem 1: If U_A and U_B agreed on $s\kappa$ in accordance with our method, then U_A and U_B can execute the scheme effectively with an identical session key $s\kappa$ with high probability.

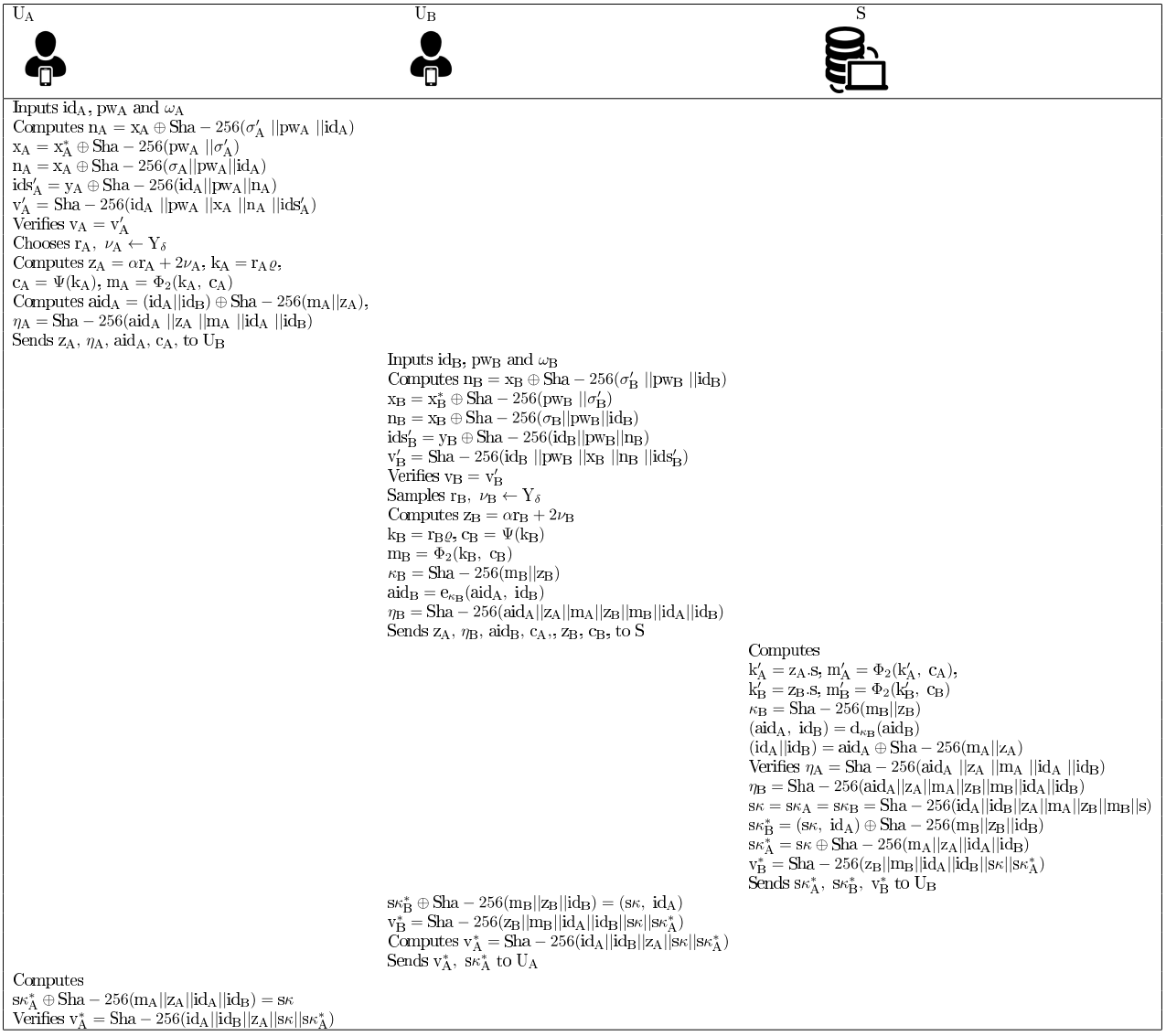

FIGURE 3. A stepwise discription of the proposed login and authentication protocol.

Proof: It sufficient to demonstrate that for user U_A, we have $|k_A - K'_A| < \frac{q}{8}$, and for user U_B, we have $|k_B - K'_B| < \frac{q}{8}$, then correctness of the protocol is obvious. To show the correctness, one have to compute error difference $|k_A - K'_A| = |r_A\varrho - z_A s| = |2r_A.e - 2\nu_A.s|$, and $||2r_A.e - 2\nu_A.s||_2 \leq ||r_A.e||_2||\nu_A.s||_2 \leq \sqrt{m}||r_A||_2||e||_2 + \sqrt{m}||\nu_A||_2||s||_2 < 2\sqrt{m}\gamma.\sqrt{m}\gamma.\sqrt{m} = 2m^{\frac{3}{2}}\gamma^2 < \frac{q}{8}$, hence the equality by lemma (VII.3). ■

VII. INFORMAL SECURITY ANALYSIS

1) IMPERSONATION ATTACK

The proposed scheme involves three parties, each playing a distinct role. Adversaries possess the capability to impersonate any of these parties, thereby orchestrating an impersonation attack. However, it's noteworthy that our scheme boasts robust security measures specifically designed to withstand such impersonation attacks. Through stringent authentication mechanisms and cryptographic protocols, our scheme ensures resilience against malicious attempts to deceive or manipulate the identities of involved parties. In the proposed protocol, the user U_A chooses r_A, $\nu_A \leftarrow Y_\delta$, and computes $z_A = \alpha r_A + 2\nu_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A, c_A)$, $aid_A = (id_A||id_B) \oplus Sha - 256(m_A||z_A)$, $\eta_A = Sha - 256(aid_A ||z_A ||m_A ||id_A ||id_B)$, and sends z_A, η_A, aid_A, c_A, to U_B which is random for each session. Similarly, U_B computes randomised z_A, η_B, aid_B, c_A,, z_B, c_B, and sends to the server S. Therefore, the impersonation is not possible in the protocol.

2) MAN IN THE MIDDLE ATTACK

In a man-in-the-middle attack, an adversary seeks to impersonate a legitimate participant, positioning themselves between two parties. This allows the attacker to intercept, modify, or inject new information into the ongoing

conversation or data transmission, all while remaining undetected by the intended recipient who assumes the communication is proceeding as usual. In the proposed protocol, the user U_A chooses $r_A,\ v_A \leftarrow Y_\delta$, and computes $z_A = \alpha r_A + 2v_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A,\ c_A)$, $aid_A = (id_A||id_B) \oplus Sha - 256(m_A||z_A)$, $\eta_A = Sha - 256(aid_A\ ||z_A\ ||m_A\ ||id_A\ ||id_B)$, and sends z_A, η_A, aid_A, c_A, to U_B which is random for each session. Similarly, U_B computes randomised z_A, η_B, aid_B, c_A,, z_B, c_B, and sends to the server S. Therefore, our approach employs an authentication mechanism at every stage of communication or data transmission to mitigate such threats.

3) REPLAY ATTACK

A replay attack, orchestrated by an adversary, occurs when intercepted data transmitted between two parties over a network is maliciously retransmitted to the receiver. This nefarious tactic is often employed to gain unauthorized access to a system or facilitate illicit activities. However, it's essential to highlight that our scheme is fortified against replay attacks. In the proposed protocol, the user U_A chooses $r_A,\ v_A \leftarrow Y_\delta$, and computes $z_A = \alpha r_A + 2v_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A,\ c_A)$, $aid_A = (id_A||id_B) \oplus Sha - 256(m_A||z_A)$, $\eta_A = Sha - 256(aid_A\ ||z_A\ ||m_A\ ||id_A\ ||id_B)$, and sends z_A, η_A, aid_A, c_A, to U_B which is random for each session. The user U_B samples random $r_B,\ v_B \leftarrow Y_\delta$, and it computes $z_B = \alpha r_B + 2v_B$, $k_B = r_B\varrho$, $c_B = \Psi(k_B)$, $m_B = \Phi_2(k_B,\ c_B)$, $\kappa_B = Sha - 256(m_B||z_B)$, and derives masked identity $aid_B = e_{\kappa_B}(aid_A,\ id_B)$, and $v_B = Sha - 256(aid_A||z_A||m_A||z_B||m_B||id_A||id_B)$ to validate the values z_A, v_B, aid_B, c_A,, z_B, c_B, sent over public channel to the server S. By choosing random samples, our scheme effectively mitigates the risk of data interception and malicious replay, ensuring the integrity and authenticity of the communication process.

4) PERFECT FORWARD SECURITY

To thwart potential eavesdropping on past communications, many key exchange schemes utilize perfect forward secrecy. Typically, the server employs a long-term secret key for encryption. However, this static (long-term) secret key is vulnerable to theft, enabling adversaries to masquerade as either the server or the client. Subsequently, they can orchestrate a Perfect Forward Security (PFS) attack by reconstructing the session key. In the proposed protocol, the user U_A chooses $r_A,\ v_A \leftarrow Y_\delta$, and computes $z_A = \alpha r_A + 2v_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A,\ c_A)$, $aid_A = (id_A||id_B) \oplus Sha - 256(m_A||z_A)$, $\eta_A = Sha - 256(aid_A\ ||z_A\ ||m_A\ ||id_A\ ||id_B)$, and sends z_A, η_A, aid_A, c_A, to U_B which is random for each session. The user U_B samples random $r_B,\ v_B \leftarrow Y_\delta$, and it computes $z_B = \alpha r_B + 2v_B$, $k_B = r_B\varrho$, $c_B = \Psi(k_B)$, $m_B = \Phi_2(k_B,\ c_B)$, $\kappa_B = Sha - 256(m_B||z_B)$, and derives masked identity $aid_B = e_{\kappa_B}(aid_A,\ id_B)$, and $v_B = Sha - 256(aid_A||z_A||m_A||z_B||m_B||id_A||id_B)$ to validate the values z_A, v_B, aid_B, c_A,, z_B, c_B, sent over public channel to the server S. The session key is derived using these random numbers which are different for each session. Therefore, the proposed protocol preserve forward secrecy.

5) PASSWORD GUESSING ATTACK

A password-guessing attack represents a type of cyber assault wherein an adversary systematically tries various character combinations in an effort to decipher a user's password. The user U_i for $i \in \{A,\ B\}$ selects a unique identity id_i for $i \in \{A,\ B\}$, a (preferably strong) password pw_i for $i \in \{A,\ B\}$, and chooses secrets $n_i \in Z$ for $i \in \{A,\ B\}$ randomly. After that, U_i computes $z_i = Sha - 256(id_i||pw_i||n_i)$ and sends the information $\{id_i,\ z_i\}$ to the sever S securely. After receiving the registration request from U_i, the server S first checks its database against $Sha - 256(id_i)$. If it is already in the database, it rejects the registration request; otherwise, it computes $ids_i = Sha - 256(id_i\ ||s)$ with the help of its own master secret key s. After computing $y_i = ids_i \oplus z_i$, S stores $Sha - 256(id_i)$ in its database and sends $\{y_i,\ Sha - 256(\cdot)\}$ to U_i via secure channel. After getting the message from S, U_i computes $ids_i = z_i \oplus y_i$ and imprints his/her biometric ω_i at sensor of the mobile device and generates a biometric secret σ_i and its associated public reproduction parameter τ_i using fuzzy extractor probabilistic generation function $Gen(\cdot)$ as $\omega_i = \{\sigma_i, \tau_i\}$. Next, U_i computes $x_i = Sha - 256(\sigma_i\ ||pw_i\ ||id_i) \oplus n_i$, $x_i^* = x_i \oplus Sha - 256(pw_i\ ||\sigma_i)$ and a verification value $v_i = Sha - 256(id_i\ ||pw_i\ ||x_i\ ||n_i\ ||ids_i)$ for the user. Finally, U_i stores the registration credentials $(y_i,\ v_i,\ x_i^*,\ Sha - 256(.)$ into his/her mobile application. Therefore, the password guessing is not possible for the proposed protocol.

6) NO CLOCK SYNCHRONIZATION

In our proposed scheme, we opt for a random nonce instead of employing a time stamp, thereby eliminating the need for clock synchronization. This strategic choice enhances the scheme's flexibility and resilience, as it mitigates potential challenges associated with coordinating time across different devices or networks. By leveraging a random nonce, our scheme maintains robustness and efficiency, ensuring seamless operation in diverse environments without the reliance on synchronized clocks. The proposed scheme uses random numbers in the place of time stamp, so it avoids problem of synchronization.

7) USER ANONYMITY AND UNTRACEABLE

The proposed scheme prioritizes user anonymity by safeguarding their identities during interactions across public channels. Through stringent privacy measures, the scheme ensures that users' identities remain concealed, bolstering confidentiality and trust in the communication process. By maintaining anonymity in every interaction, our scheme offers users enhanced security, fostering a safer and more secure communication environment. The user U_A chooses $r_A,\ v_A \leftarrow Y_\delta$, and computes $z_A = \alpha r_A + 2v_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A,\ c_A)$, and masked identity $aid_A = (id_A||id_B) \oplus Sha - 256(m_A||z_A)$, and $\eta_A = Sha - 256(aid_A$

$||z_A\ ||m_A\ ||id_A\ ||id_B)$, and sends z_A, η_A, aid_A, c_A, to U_B which is random for each session. The user U_B samples random $r_B,\ v_B \leftarrow Y_\delta$, and it computes $z_B = \alpha r_B + 2v_B$, $k_B = r_B\varrho$, $c_B = \Psi(k_B)$, $m_B = \Phi_2(k_B,\ c_B)$, $\kappa_B = Sha-256(m_B||z_B)$, and derives masked identity $aid_B = e_{\kappa_B}(aid_A,\ id_B)$, and $v_B = Sha - 256(aid_A||z_A||m_A||z_B||m_B||id_A||id_B)$ to validate the values z_A, v_B, aid_B, c_A,, z_B, c_B, sent over public channel to the server S. Since the masked identities are sent over the public channel, so anonymity is preserved in the protocol. Moreover, the random number helps to randomise the messages over public channel, so the protocol provides untraceability.

VIII. FORMAL SECURITY ANALYSIS

Formal security analysis of authentication protocols involves using mathematical modeling and formal methods to rigorously analyze the security properties and vulnerabilities of authentication mechanisms. Unlike informal security analysis, which relies on qualitative assessments and general principles, formal security analysis provides a systematic and precise evaluation of the protocol's security guarantees, often using formal verification techniques. Define the threat model, which describes the capabilities and objectives of potential attackers, as well as the assumptions made about the security of underlying cryptographic primitives and communication channels. This helps determine the scope and context of the security analysis.

A. SECURITY MODEL

Security models serve as foundational tools for designing, analyzing, and evaluating the security of information systems, protocols, and cryptographic algorithms. By providing a structured framework for understanding security requirements, threats, and mitigation strategies, security models help guide the development of secure and resilient systems that can withstand a wide range of security challenges and adversarial threats.

B. ADVERSARY CAPABILITIES IN CK01

Similar to the BR models, the adversary in the CK01 model operates in a fundamentally similar way. In particular, it regulates all parties' communications. There is no distinct notation for instances in the CK01, which is a departure from the BR models' nomenclature. Instead, sessions are named using unique session IDs at any party and may be recognised as instances. More precisely, given a principal P_i wanting to establish a connection with a principal P_j via a session identifier s, a session can be identified by a tuple (P_i, P_j, s). Keep in mind that in the BR notation, the role of the instance number s and the session identifier s are entirely different. In addition to arranging message exchanges with parties (sessions) and monitoring their responses, the adversary might pose certain targeted inquiries that surpass the scope of the BR models.

1) **Party Corruption**: Similar to the BPR00 and BR95 models, the corrupt query enables the adversary to access a party's long-term key. The long-term key of the party is returned by the query, together with all of the memory, which could contain session or ephemeral keys.
2) **Session Key Reveal**: Similar to the BR models, the attacker can use a reveal query to get the session key of any finished session.
3) **Session state reveal**: The BR model prevents the adversary from obtaining information that could be stored during (or after) the session key computation, exceptÂ forÂ malformed queries. An unfinished session may be queried using the CK01 session stateÂ query, which returns the internal state. The protocol can define what constitutes the session state thanks to the model; an ephemeral Diffie-Hellman exponent is a common example.
4) **Session expire**: After making the necessary modifications to the new queries, the definition of security in the CK01 model roughly resembles the BR definitions we have seen in earlier parts. The concept revolves around the lack of an effective adversary capable of differentiating between a random string in the session key space and the session key in a newly created session. Rather than adopting the nomenclature from the CK01 article, we will stick with the BR model's terminology to emphasise the similarities when describing the security game.

Provable security of authentication protocols involves demonstrating mathematically that the protocol achieves its security objectives under specific assumptions and threat models. Unlike empirical security, which relies on testing and observation, provable security provides formal guarantees of security based on rigorous mathematical analysis and proof techniques. To simulate the proof, we have taken the protocol E along with three components (1) U_a, (2) U_b and (3) S_j with a random arbitrary i^{th} instance for a participant P^i.

C. FORMAL SECURITY PROOF

Theorem 2: Let us consider an active/passive adversary $\mathcal{A}$ is allowed to execute at most s_q sends queries, e_q executes queries and h_q hashings, and $Advt_{\mathcal{A}}(t)$ be advantage gain by it. Let us consider another notation $Advt_{\mathcal{A}}^{RLWE}(t)$ for the advantage to solve Ring Learning with Error assumption with polynomial time t, then

$$Advt_{\mathcal{A}}(t) \leq \frac{2h_q^2}{q} + \frac{2s_q}{q} + \frac{(e_q + s_q)^2}{q} + (2h_q)Advt_{\mathcal{A}}^{RLWE}(t) + \frac{2h_q}{q} + 2(c_z \cdot s_q^{s_z}). \quad (1)$$

where c_z and s_z denote Zipf's parameter [27].

Proof: An active adversary $\mathcal{A}$ employs a variety of attack techniques to compromise the security of systems, networks, or organizations. These attacks are often sophisticated, targeted, and designed to achieve specific objectives, such as unauthorized access, data theft, or service disruption.

To show the idea of attack the challenger $\mathcal{C}$ sends a challenge and an adversary $\mathcal{A}$ sends response back. These queries are submitted by $\mathcal{A}$ to $\mathcal{C}$, and $\mathcal{C}$ is responsible to give response back to $\mathcal{A}$.

Now, the adversary $\mathcal{A}$ analyses responses erecived, and breaches the semantic security of established session key. One can prove this fact by playing games from GM_1 to GM_6. Let us consider ϱ_i denotes the event corresponding to game GM_i and the advantage of ϱ_i to be denoted by probability $Pr(\varrho_i)$. One thing should be noted that ϱ_i stands for corresponding GM_i, and it does not care for $\mathcal{A}$ succeeds or failed in breaking the semantic security of the proposed protocol π in GM_i. Let us explain that if $\mathcal{A}$ run, then E being independent to ϱ_i may be executed, E can be used by $\mathcal{C}$. One can Observe that until E is executed, GM_i and GM_{i+1} are identical. Therefore we have

$$|Pr[\varrho_{i+1}] - pr[\varrho_i]| \leq Pr[E]$$

Defining the Games:

1) **GM$_1$**: The game GM_1 similated in random Oracle is identical to simulation of a actual attack on the protocol. Therefore, one can claim

$$Advt_{\mathcal{A}}(t) = |Pr(\varrho_0) - \frac{1}{2}| \tag{2}$$

Now, suppose $D_i = |Pr(\varrho_{i-1}) - Pr(\varrho_i)|$, then one can have transformation from equation (2) as

$$\begin{aligned} Advt_{\mathcal{A}}(t) &= |Pr(\varrho_0) - \frac{1}{2}| \\ &= |Pr(\varrho_0) - Pr(\varrho_4) + Pr(\varrho_4) - \frac{1}{2}| \\ &= |\sum_{i=1}^{5} D_i + Pr(\varrho_4) - \frac{1}{2}| \end{aligned}$$

2) **GM$_2$**: In GM_2, the adversary $\mathcal{A}$ executes hashing queries which is asked in different manner then GM_1, and apart from this fact, it is not distinguishable from GM_1. In the game GM_2, we have hashing table L_h that includes pair (a, b) following $b = h(a)$. Whenever the adversary $\mathcal{A}$ asks hashing queries $h(.)$ for input a, then the challenger $\mathcal{C}$ looks into the L_h for existence of pair (a, b), if found, then $\mathcal{C}$ returns with b, otherwise b', where b' is random, and it includes (a, b') into L_h, then returns with b' to $\mathcal{A}$. Therefore,

$$Pr[\varrho_1] = Pr[\varrho_0] \tag{3}$$

3) **GM$_3$**: This game cannot be distinguished from GM_2, it has only exception that it can abort if collision found for massages $\{z_A, v_A, aid_A, c_A\}$, $\{z_B, v_B, aid_B, c_B\}$, $\{s\kappa_A^*, s\kappa_B^*, v_B^*\}$, $\{s\kappa_A^*, v_A^*\}$. According the Birthday Paradox, the optimized chances that hashing oracle will give identical output is at most $\frac{h_q^2}{2q}$, and for two arbitrary samples at most $\frac{(e_q+S_q)^2}{2q}$. Therefore,

$$D_2 = |Pr(\varrho_1) - Pr(\varrho_2)| \leq (\frac{h_a^2}{2q} + \frac{(e_q + S_q)^2}{2q}). \tag{4}$$

4) **GM$_4$**: This game cannot be distinguished from GM_3, it has only exception that a random chosen instance by user $\pi_s^{U_i}$ or server $\pi_s^{S_j}$ disregards a conforming to the rules for authentication values. The game GM_3 can be terminated if $\mathcal{A}$ correctly guesses bit b without taking help of Oracle $h(.)$, and the phase Tests query. Therefore,

$$D_3 = |Pr[\varrho_2] - Pr[\varrho_3]| = \frac{h_q}{2q} \tag{5}$$

5) **GM$_5$**: This game cannot be distinguished from GM_4, it has only exception that session key $s\kappa$ uses the concept of ring learning with error, it means the key does dependent on hashing $h(.)$ and simulation Oracle $h(.)$. In the given protocol, $s\kappa = Sha - 256(id_A||id_B||z_A||m_A||z_B||m_B||s)$, where $z_A = \alpha r_A + 2v_A$, $k_A = r_A\varrho$, $c_A = \Psi(k_A)$, $m_A = \Phi_2(k_A, c_A)$, and $z_B = \alpha r_B + 2v_B$, $k_B = r_B\varrho$, $c_B = \Psi(k_B)$, $m_B = \Phi_2(k_B, c_B)$, respectively. Therefore, it can be observed that simulation is done using self-reducible instance of Ring Learning with Error assumption, and it concludes that if $\mathcal{A}$ correctly guesses $s\kappa$, then $\mathcal{A}$ solves Ring Learning with Error assumption efficiently. Therefore,

$$D_4 = |Pr(\varrho_3) - Pr(\varrho_4)| \leq h_q.Advt_{\mathcal{A}}^{RLWE}(t) + \frac{h_q}{q}. \tag{6}$$

6) **GM$_6$**: This game cannot be distinguished from GM_5, it has only exception that hashing Oracle queried with input $s\kappa$. The chances of guessing actual bit b correctly in tests query is at most $\frac{h_q^2}{2p}$. Moreover, the adversary $\mathcal{A}$ is not able to distinguish the real session key from random string unless hashing Oracle is similated correct inputs. Therefore, $Pr(\varrho_5) = \frac{1}{2}$, and guessing password with low entropy [27] could be used. According to this rule, if one considers $s_q = 10^7$ or 10^8, then $\mathcal{A}$'s probability to win is greater than $\frac{1}{2}$. On the contrary, if $s_q \leq 10^6$, then $\mathcal{A}$'s probability to win more than $\frac{1}{2}$, and off-line password-guessing probability is $\leq c_z \cdot s_q^{s_z}$ [27]. So we have

$$D_5 = |Pr(\varrho_4) - Pr(\varrho_5)| \leq \frac{h_q^2}{2q} + c_z \cdot s_q^{s_z}. \tag{7}$$

Therefore, one can get equation (1), if one combines equations (2) to (7), and uses mathematical inequalities correctly. This confirms the security of the proposed protocol under Ring Learning with Error assumption. ■

IX. SECURITY COMPARISON AND PERFORMANCE ANALYSIS

In this section, the computation cost of the proposed protocol is calculated. The protocol is implemented in C/C++, where a type of processing can be either multi-threading or parallel. Certain libraries like latticeCrypto and MIRACLE are used for the implementation purpose. An HP laptop

TABLE 2. Average run time of various cryptographic primitives in nanoseconds (*ns*).

Operation	User side	Server side
T_{Ge}	540.865	72.504
T_{smul}	6.343	0.260
T_{pma}	29.764	2.346
T_{pmul}	12.088	0.291
T_{cha}	33.346	0.646
T_{h}	178.369	13.458
T_{c}	1121	135
T_{sym}	0.4635	0.0089
T_{ecpm}	3363	405

with Linux(ubuntu) operating system (Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz and 8GB RAM) will be a server. A Samsung mobile device, GT-I9300, having Android 4.3 OS (1.4GHz Quad-Core and 1 GB RAM) is the end user (see Table (5) for server, and see Table (6) for device configurations).

This section assess the efficiency of our approach concerning communication costs, computation time, and security features when compared to both existing post-quantum and non-post-quantum protocols. Specifically, we set n to be 1024 bits, $\log \delta = 17.01$ for the Gaussian distribution, where q is an odd large prime, and δ denotes the standard deviation. We use Charm-crypto, Lattice-crypto, Miracle, NumPy, and hash lib libraries. The specified operations are executed on an x64-based PC featuring an Intel(R) Core(TM) 1.00 GHz i5-1035I CPU with 8 GB RAM and NVIDIA GeForce RTX 2060 operating system Linux(ubuntu). The mean cost associated with sampling from the Gaussian distribution χ_δ is denoted as T_{Ge}, while the average cost of a single scalar multiplication is represented by T_{smul}. T_{pmul} is employed to express the average cost of a single multiplication in Q_q. For both multiplication and addition in Q_q, we use T_{pma}, T_{cha} signifies the cost of the characteristic function, and T_{sym} is utilized for symmetric encryption/decryption. Moreover, T_h symbolizes the cost associated with hashing, t_{exp} signifies exponentiation, T_{ecpm} represents the cost of elliptic curve point multiplication, and T_c is designated for chaotic map operations. Furthermore, we make the assumption that $T_{sym} \approx 0.0139 T_{cha}$ as proposed in [14]. Additionally, we take into account that the time needed for elliptic curve point multiplication is threefold the cost of a chaotic map operation, and according to [20], the fuzzy extractor function is nearly equivalent to elliptic curve point multiplication. The expense of each operation is then documented in Table 2.

We conduct a comparison between the communication and computation costs of the proposed protocol and classical authentication schemes. The computational cost comparison, specifically focusing on each protocol's authentication and key agreement components, is presented in Table 3 and Fig. 4. The outcomes presented in Table 3 reveal that the proposed approach results in lower computational costs when contrasted with other classical authentication schemes under consideration.

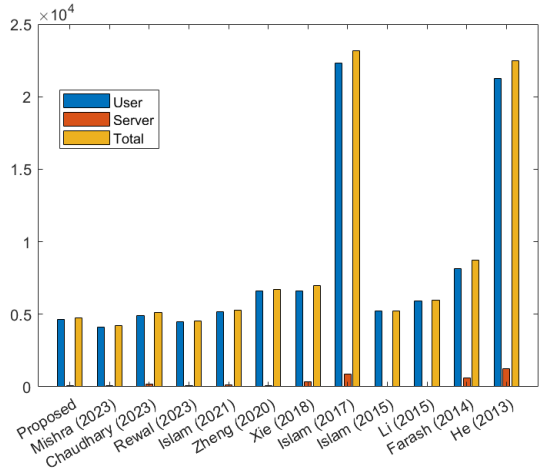

FIGURE 4. Comparative analysis on computation costs.

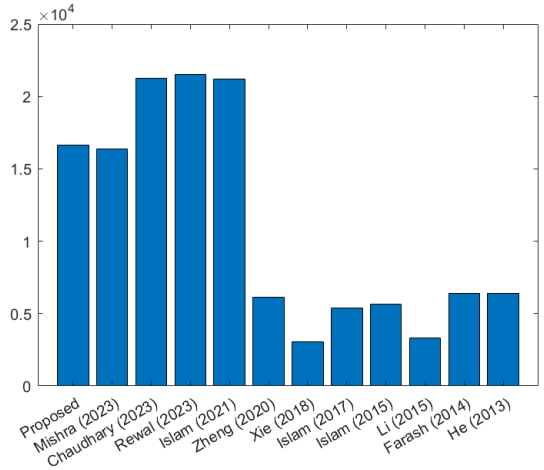

FIGURE 5. Comparative analysis on communication costs.

```
ubuntu@ubuntu-virtual-machine:~$ python3 server.py
T(Ge) : 72.5036865883552984
T(smul) : 0.259758464378438
T(pmul) : 0.290874676256378
T(pma) : 2.34567875434567456
T(cha) : 0.64567875434567456
T(h) : 13.4577656723434234
T(fe) : 405
T(bp) : 8190
T(ecca) : 10
T(ecpm) : 405
```

FIGURE 6. Server Computation cost in nanoseconds.

```
ubuntu@ubuntu-virtual-machine:~$ python3 user.py
T(Ge) : 540.865456787654
T(smul) : 6.343453434
T(pmul) : 12.087656734
T(pma) : 29.76456765456
T(cha) : 33.34567654567
T(h) : 178.3693667878
T(fe) : 3363
T(bp) : 51369
T(ecca) : 72
T(ecpm) : 3363
```

FIGURE 7. User Computation cost in nanoseconds.

We have conducted a comparison of the computational costs of the proposed scheme with post-quantum authentication schemes, including those proposed by Mishra et al. [23],

TABLE 3. Computation costs comparison with non post-quantum authentication schemes.

Scheme	Users-side computations cost (in ns)	Servers-side computations cost (in ns)	Total computation cost (in ns)
Proposed	$13T_h + 4T_{Ge} + 3T_{pmul} + 2T_{pma} + 2T_{smul} + 2T_{cha} \approx 4657.427$	$7T_h + T_{pmul} + 2T_{cha} \approx 97.552$	4754.979
Zheng et al. [32]	$12T_h + 4T_c \approx 6624.428$	$7T_h \approx 94.206$	6718.634
Xie et al. [29]	$12T_h + 4T_c + 5T_{sym} \approx 6626.7455$	$7T_h + 2T_c + 5T_{sym} \approx 364.251$	6990.9965
Islam et al. [16]	$12T_h + 6T_{ecpm} \approx 22318.428$	$4T_h + 2T_{ecpm} \approx 863.832$	23182.26
Islam et al. [14]	$4T_h + 4T_c + 10T_{sym} \approx 5202.111$	$3T_h + 4T_{sym} \approx 40.410$	5242.521
Li et al. [21]	$8T_h + 4T_c + 4T_{sym} \approx 5912.806$	$3T_h + 4T_{sym} \approx 40.410$	5953.216
Farash et al. [10]	$6T_c + 8T_h \approx 8152.952$	$4T_c + 4T_h \approx 593.832$	8706.784
He et al. [13]	$6T_h + 6T_{ecpm} \approx 21248.214$	$3T_h + 3T_{ecpm} \approx 1255.374$	22503.588

TABLE 4. Computation costs comparison with post-quantum authentication schemes.

Scheme	Users-side computation cost (in μs)	Servers-side computation cost (in μs)	Total computations cost (in μs)
Proposed	$13T_h + 4T_{Ge} + 3T_{pmul} + 2T_{pma} + 2T_{smul} + 2T_{cha} \approx 4657.427$	$7T_h + T_{pmul} + 2T_{cha} \approx 97.552$	4754.979
Mishra et al. (2023) [23]	$10T_h + 4T_{Ge} + 3T_{pmul} + 2T_{pma} + 2T_{smul} + 2T_{cha} \approx 4122.32$	$6T_h + T_{pmul} \approx 83.093$	4205.413
Chaudhary et al. (2023) [1]	$2(7T_h + 2T_{Ge} + 2T_{pmul} + T_{pma} + T_{smul} + 2T_{cha}) \approx 4914.576$	$10T_h + T_{Ge} + 2T_{pmul} \approx 211.776$	5126.352
Rewal et al. (2023) [25]	$2(6T_h + 2T_{Ge} + T_{pmul} + T_{pma} + T_{smul} + T_{cha}) \approx 4466.97$	$6T_h \approx 80.748$	4547.718
Islam and Basu (2021) [17]	$2(8T_h + 2T_{Ge} + T_{pmul} + T_{pma} + T_{smul} + T_{cha}) \approx 5180.446$	$8T_h \approx 107.664$	5288.11

TABLE 5. Server configuration.

Components	Details
Processor	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
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	Linux(ubuntu)

TABLE 6. Device configuration.

Components	Details
Operating System	Android 13, MIUI 14
Android Version	13 TKQ1.221114.001
CPU	Snapdragon 680, Octa-core Max 2.40GHz
Model	2201117TI
RAM	6.0+2.0 GB
Kernel Version	4.19.157-perf-gcb1ffc010755
Cores	8

TABLE 7. Comparison of communication costs with non-post-quantum authentication schemes.

Scheme	Communication cost (in bits)
Proposed	16640
Zhang et al. [32]	6144
Xie et al. [29]	3072
Islam et al. [16]	5376
Islam et al. [14]	5632
Li et al. [21]	3328
Farash et al. [10]	6400
He et al. [13]	6400

TABLE 8. Comparison of communication costs with post-quantum authentication schemes.

Scheme	Communication cost (in bits)
Proposed	16640
Mishra et al. (2023) [23]	16384
Chaudhary et al. (2023) [1]	21248
Rewal et al. (2023) [25]	21504
Islam and Basu (2021) [17]	21210

TABLE 9. Security attribute comparative analysis with non-post-quantum authentication schemes.

Attribute	[13]	[10]	[21]	[14]	[16]	[29]	[32]	Proposed
Smart Card Attack	✓	✓	✓	✓	×	✓	✓	✓
MiTM Attack	✓	✓	✓	✓	✓	✓	✓	✓
User anonymous	×	×	×	✓	×	×	✓	✓
Mutual Authentications	✓	✓	✓	✓	✓	✓	✓	✓
Forward secure	✓	✓	✓	✓	✓	✓	✓	✓
Impersonation attack	✓	✓	✓	✓	×	✓	✓	✓
Replays attack	✓	×	✓	×	✓	✓	✓	✓
Insider's attack	✓	×	✓	✓	✓	✓	✓	✓
Passwords guess	×	×	×	✓	×	✓	✓	✓
Quantum computer attack	×	×	×	×	×	×	×	✓

Chaudhary et al. [1], Rewal et al. [25], and Islam and Basu [17]. The results are summarized in Table 4. It is apparent that our protocol incurs lower computational costs for both the user and server sides compared to those of Chaudhary et al. [1] and Islam and Basu scheme [17]. Meanwhile, it demonstrates comparable computational cost requirements with Mishra et al. [23] and Rewal et al. scheme [25]. To depict the communication costs linked with relevant existing authentication protocols, each binary string is considered to be 256 bits, encompassing passwords, nonces, timestamps, and identities. In the hashing process, we employ a 512-bit hash output using the Secure Hash Algorithm (SHA-512). For symmetric encryption/decryption, a 256-bit key is used, and AES-256 algorithm is employed for symmetric encryption/decryption. The chaotic map utilizes a 256-bit size, an elliptic curve point is of 320 bits, and each element from Q_q utilizes a 4094-bit size. Table 7 and Fig. 5 outline the communication costs of the proposed scheme, juxtaposed with classical authentication schemes [10], [13], [14], [16], [21], [29], [32].

It is observed that the proposed scheme incurs higher communication costs compared to classical authentication schemes [10], [13], [14], [16], [21], [29], [32]. This increase is attributed to the polynomial size involved in the lattice-based cryptographic scheme. In comparing security characteristics, our primary considerations encompass security such as replay attacks, man-in-the-middle attacks, password guessing, impersonation, privileged-insider threats, known-key

TABLE 10. Security attribute comparative analysis with post-quantum authentication schemes.

Attribute	[17]	[25]	[1]	[23]	Proposed
Smart Attack	×	✓	✓	✓	✓
Mutual Authentication	✓	✓	✓	✓	✓
User anonymous	×	×	✓	×	✓
MiTM Attack	✓	✓	✓	✓	✓
Impersonations	×	✓	✓	✓	✓
Password guessing	×	✓	✓	✓	✓
Forward secure	✓	✓	×	×	✓
Replay attack	✓	✓	✓	✓	✓
Malicious Insider	✓	✓	✓	✓	✓
Traceability	✓	✓	×	×	✓
Quantum computer attack	✓	✓	✓	✓	✓

security, mutual authentication, and perfect forward secrecy of the session key. Table 9 presents a comparison of security attributes between the proposed protocol and other relevant existing classical authentication protocols. On the flip side, Table 10 offers a contrast of diverse security attributes between the proposed protocol and other contemporary post-quantum authentication schemes. In this context, the symbol "✓" denotes the fulfilment of a condition, whereas "×" signifies that a property is not met. It is critical to emphasise that the suggested protocol excels in meeting all of the mentioned security criteria. In contrast, some existing authentication techniques fall short of fully implementing these criteria. This demonstrates the proposed protocol's resilience and efficacy in resolving a variety of security challenges.

X. CONCLUSION

The suggested protocol makes use of the robustness of quantum-resistant cryptographic primitives, especially those based on lattice structures, to guarantee the forward secrecy and integrity of key agreement procedures for three party protocol. We have shown that user (A) can safely establish session keys with user user (B), and session key is fresh for each of sessions. Therefore, the proposed protocol is forward secure. A forward-secure protocol is a cryptographic protocol that provides forward secrecy, also known as perfect forward secrecy (PFS). Forward secrecy ensures that if long-term secret keys are compromised in the future, past communication sessions remain secure and cannot be retroactively decrypted. For example, Transport Layer Security (TLS), used for securing web browsing sessions, can benefit from forward-secure protocols to provide perfect forward secrecy. Forward secrecy in TLS ensures that past HTTPS sessions remain secure even if the server's private key is compromised in the future. This protocol provides anonymous communication which is used to anonymous email services like ProtonMail and Tutanota offer end-to-end encrypted email communication while anonymizing user metadata. Additionally, a comparison with current protools is shown in the perforamnce section. The evaluation demonstrates that the proposed approach has reasonable less computation costs and offers sufficient security against known quantum attacks. In the near future, it might be possible to improve computational effectiveness by cutting down on the number of operations while maintaining security. This protocol may have application to provide security in Internet of Drones, Internet of Vehicles, and Internet of Things etc.

REFERENCES

[1] D. Chaudhary, U. Kumar, and K. Saleem, "A construction of three party post quantum secure authenticated key exchange using ring learning with errors and ECC cryptography," *IEEE Access*, vol. 11, pp. 136947–136957, 2023.

[2] V. Dabra, A. Bala, and S. Kumari, "LBA-PAKE: Lattice-based anonymous password authenticated key exchange for mobile devices," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5067–5077, Dec. 2021.

[3] V. Dabra, A. Bala, and S. Kumari, "Flaw and amendment of a two-party authenticated key agreement protocol for post-quantum environments," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102889.

[4] D. Dharminder and K. P. Chandran, "LWESM: Learning with error based secure communication in mobile devices using fuzzy extractor," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 10, pp. 4089–4100, Oct. 2020.

[5] D. Dharminder, C. B. Reddy, A. K. Das, Y. Park, and S. S. Jamal, "Post-quantum lattice-based secure reconciliation enabled key agreement protocol for IoT," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2680–2692, Feb. 2023.

[6] J. Ding, S. Alsayigh, J. Lancrenon, S. Rv, and M. Snook, "Provably secure password authenticated key exchange based on RLWE for the post-quantum world," in *Proc. Cryptographers' Track RSA Conf.*, San Francisco, CA, USA. Springer, Feb. 2017, pp. 183–204.

[7] J. Ding, S. Alsayigh, R. V. Saraswathy, S. Fluhrer, and X. Lin, "Leakage of signal function with reused keys in RLWE key exchange," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[8] J. Ding, S. Fluhrer, and S. Rv, "Complete attack on RLWE key exchange with reused keys, without signal leakage," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Wollongong, NSW, Australia. Cham, Switzerland: Springer, Jul. 2018, pp. 467–486.

[9] J. Ding, X. Xie, and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," *Cryptol. ePrint Arch.*, Jan. 2012.

[10] M. S. Farash and M. A. Attari, "An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps," *Nonlinear Dyn.*, vol. 77, nos. 1–2, pp. 399–411, Jul. 2014.

[11] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2775–2785, Sep. 2019.

[12] S. Fluhrer, "Cryptanalysis of ring-LWE based key exchange with key share reuse," *Cryptol. ePrint Arch.*, Jan. 2016.

[13] D. He, Y. Chen, and J. Chen, "An id-based three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments," *Arabian J. Sci. Eng.*, vol. 38, no. 8, pp. 2055–2061, Aug. 2013.

[14] S. H. Islam, "Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps," *Inf. Sci.*, vol. 312, pp. 104–130, Aug. 2015.

[15] S. H. Islam, "Provably secure two-party authenticated key agreement protocol for post-quantum environments," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102468.

[16] S. K. H. Islam, R. Amin, G. P. Biswas, M. S. Farash, X. Li, and S. Kumari, "An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 29, no. 3, pp. 311–324, Jul. 2017.

[17] S. H. Islam and S. Basu, "PB-3PAKA: Password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments," *J. Inf. Secur. Appl.*, vol. 63, Dec. 2021, Art. no. 103026.

[18] D. Kirkwood, B. C. Lackey, J. McVey, M. Motley, J. A. Solinas, and D. Tuller, "Failure is not an option: Standardization issues for post-quantum key agreement," in *Proc. Workshop Cybersecurity Post-Quantum World*, 2015, p. 21.

[19] U. Kumar, M. Garg, S. Kumari, and D. Dharminder, "A construction of post quantum secure and signal leakage resistant authenticated key agreement protocol for mobile communication," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 1, Jan. 2023, Art. no. e4660.
[20] S. Kumari, A. K. Das, X. Li, F. Wu, M. K. Khan, Q. Jiang, and S. K. H. Islam, "A provably secure biometrics-based authenticated key agreement scheme for multi-server environments," *Multimedia Tools Appl.*, vol. 77, no. 2, pp. 2359–2389, Jan. 2018.
[21] X. Li, J. Niu, S. Kumari, M. K. Khan, J. Liao, and W. Liang, "Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol," *Nonlinear Dyn.*, vol. 80, no. 3, pp. 1209–1220, May 2015.
[22] C. Liu, Z. Zheng, K. Jia, and Q. You, "Provably secure three-party password-based authenticated key exchange from RLWE," in *Proc. 15th Int. Conf. Inf. Secur. Pract. Exper. (ISPEC)*, Kuala Lumpur, Malaysia. Springer, Nov. 2019, pp. 56–72.
[23] D. Mishra, M. Singh, P. Reval, K. Pursharthi, N. Kumar, A. Barnawi, and R. Rathore, "Quantum-safe secure and authorized communication protocol for Internet of Drones," *IEEE Trans. Veh. Technol.*, 2023.
[24] C. Peikert, "Lattice cryptography for the internet," in *Proc. 6th Int. Workshop Post-Quantum Cryptogr.*, Waterloo, ON, Canada. Springer, Oct. 2014, pp. 197–219.
[25] P. Rewal, M. Singh, D. Mishra, K. Pursharthi, and A. Mishra, "Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices," *J. Inf. Secur. Appl.*, vol. 75, Jun. 2023, Art. no. 103505.
[26] D. Stebila and M. Mosca, "Post-quantum key exchange for the internet and the open quantum safe project," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Springer, 2016, pp. 14–37.
[27] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.
[28] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 193–208, Jan. 2023.
[29] Q. Xie, Y. Lu, X. Tan, Z. Tang, and B. Hu, "Security and efficiency enhancement of an anonymous three-party password-authenticated key agreement using extended chaotic maps," *PLoS ONE*, vol. 13, no. 10, Oct. 2018, Art. no. e0203984.
[30] D. Xu, D. He, K.-K. R. Choo, and J. Chen, "Provably secure three-party password authenticated key exchange protocol based on ring learning with error," *Cryptol. ePrint Arch.*, Jan. 2017.
[31] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen, "Authenticated key exchange from ideal lattices," in *Proc. 34th Annu. Int. Conf. Theory Appl. Cryptograph. Tech. Adv. Cryptol. (EUROCRYPT)*, Sofia, Bulgaria. Springer, Apr. 2015, pp. 719–751.
[32] Y. Zheng, S. Hu, L. Wei, Y. Chen, H. Wang, Y. Yang, Y. Li, B. Xu, W. Huang, and L. Chen, "Design and analysis of a security-enhanced three-party authenticated key agreement protocol based on chaotic maps," *IEEE Access*, vol. 8, pp. 66150–66162, 2020.

DHARMINDER CHAUDHARY (Member, IEEE) received the Ph.D. degree in cryptography and network security. He is currently an Assistant Professor (Senior Grade) with the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India. He has published 30 sci/scopus-indexed articles in the areas cryptography and network security and the Internet of Drones/vehicles security.

PRADEEP KUMAR DADSENA (Member, IEEE) is currently pursuing the Ph.D. degree in cryptography and network security with the Department of Mathematics, Government Engineering College at Jagdalpur, Jagdalpur, India. He has published four sci/scopus-indexed articles in the areas cryptography and network security and the Internet of Drones/vehicles security.

A. PADMAVATHI (Member, IEEE) received the Ph.D. degree in cryptography and network security. She is currently an Assistant Professor (Selection Grade) with the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India. She has published 12 sci/scopus-indexed articles in the areas cryptography and network security and the Internet of Drones/vehicles security.

MOHAMMAD MEHEDI HASSAN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in February 2011. He is currently a Professor with the Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored around 340 publications, including refereed IEEE /Springer/Elsevier journals conference papers, books, and book chapters. His research interests include edge/cloud computing, the Internet of Things, cyber security, deep learning, artificial intelligence, body sensor networks, 5G networks, and social networks.

BADER FAHAD ALKHAMEES (Member, IEEE) received the bachelor's degree in computer science from King Saud University, Riyadh, Saudi Arabia, in 2003, the master's degree in software systems from Heriot Watt University, Scotland, U.K., in 2008, and the Ph.D. degree in biomedical informatics from Rutgers University, Newark, NJ, USA. Currently, he is an Associate Professor with the Department of Information System, College of Computer and Information Sciences, King Saud University. His research interests include biomedical informatics, medical imaging and diagnosis, machine learning, fuzzy systems, cloud and edge computing, the Internet of Things, and computer networks.

UDDESHAYA KUMAR received the B.Sc. and M.Sc. degrees in mathematics from CCS University. He is currently an Assistant Professor with Gautam Buddha University, Noida, India. His research interests includes cryptography, vehicular communications, security and privacy in IoV, and the IoT.