Check for updates

Received: 19 August 2022 | Revised: 6 February 2023 | Accepted: 27 February 2023

DOI: 10.4218/etrij.2022-0320

ORIGINAL ARTICLE

ETRI Journal WILEY

Post-quantum identity-based authenticated multiple key agreement protocol

Yang Yang[1] | **Hongji Yuan**[2] | **Linbo Yan**[1] | **Yinglan Ruan**[1]

[1]School of Electronics and Information, Nanchang Institute of Technology, Nanchang, China

[2]School of Aeronautics and Astronautics, Nanchang Institute of Technology, Nanchang, China

Correspondence
Hongji Yuan, School of Aeronautics and Astronautics, Nanchang Institute of Technology, Nanchang, China.
Email: yuanhongji2022@163.com

Funding information
This work was supported by the Science and Technology Project of Provincial Education Department of Jiangxi under Grants GJJ212104 and GJJ212105.

Abstract

Authenticated multiple key agreement (AMKA) protocols provide participants with multiple session keys after one round of authentication. Many schemes use Diffie–Hellman or authenticated key agreement schemes that rely on hard integer factorizations that are vulnerable to quantum algorithms. Lattice cryptography provides quantum resistance to authenticated key agreement protocols, but the certificate always incurs excessive public key infrastructure management overhead. Thus, a lightweight lattice-based secure system is needed that removes this overhead. To answer this need, we provide a two-party lattice- and identity-based AMKA scheme based on bilateral short integer or computational bilateral inhomogeneous small integer solutions, and we provide a security proof based on the random oracle model. Compared with existing AMKA protocols, our new protocol has higher efficiency and stronger security.

KEYWORDS
identity-based cryptography, lattice-based cryptography, multiple key agreement

1 | INTRODUCTION

A key agreement is a fundamental cryptographic primitive that enables entities to obtain session keys for secure communications over nonsecure channels. Diffie and Hellman [1] proposed the first key agreement scheme. Unfortunately, owing to the lack of authentication, their scheme is vulnerable to man-in-the-middle attacks. In addition to exchanging session keys, authenticated key agreement (AKA) protocols [2, 3] force participants to ensure the authenticity of the other party.

A public key cryptography (PKC)-based AKA uses certificates to ensure participant authenticity. However, the public key infrastructure (PKI) incurs a large overhead due to PKC certificate management. Identity-based cryptography (IBC) [4] can eliminate this problem by using an arbitrary identity string (e.g., email address) as the public key, whereas a trusted private key generator (PKG) produces a secret key. IBC was employed to provide the first AKA protocol [5], but with the advent of quantum computing, the security of these and other cryptographic protocols has been greatly challenged. A lattice-based cryptosystem can force the worst-case solvable hard problem difficulty; thus, lattice-based protocols offer stronger security. Moreover, they involve simple matrix and vector operations that incur lower computational costs.

To improve the efficiency of key agreement protocols, Harn and Lin designed the first authenticated multiple key agreement (AMKA) scheme [6], which produces multiple keys that are used in future sessions.

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (http://www.kogl.or.kr/info/licenseTypeEn.do).
1225-6463/$ © 2023 ETRI

1.1 | Motivation and contribution

Key agreement protocols that combine the advantages of simple identity-based public key management with lattice-based security are needed to protect against quantum computers. Therefore, we leverage IBC-based, two-party lattice-based AMKA protocols in this study. However, owing to the generation of multiple keys per authentication, security grows in complexity. Notably, the free key escrow provided by IBC adds security risks.

Therefore, mutual secrecy is also needed so that the disclosure of some session keys will not compromise all keys. We assume that the adversary can learn session and private keys, but not ephemeral keys, by compromising an entity. Our objective in this study is to provide an identity-based AMKE protocol that satisfies the security properties discussed in the literature [7, 8].

Based on the hardness assumptions of lattice bilateral (Bi) small integer solution (SIS) and computational Bi (CBi) inhomogeneous SIS (ISIS) problems, we apply IBC to design a lattice- and identity-based AMKA (LIAMKA) protocol. The main contributions of this study are as follows:

- We combined IBC with lattice-based cryptography to construct the LIAMKA protocol, which avoids the huge PKC management overhead. The average-case hardness assumptions (i.e., Bi-SIS and CBi-ISIS) are unsolvable in polynomial time by quantum computers.
- The proposed LIAMKA protocol is secure in a random oracle model (ROM). A detailed informal security analysis further demonstrates that the proposed LIAMKA protocol satisfies the security requirements of identity-based AMKE protocols.
- Because only matrix and vector addition and multiplication are required, the proposed protocol incurs a very low computational overhead. Compared with state-of-the-art AMKA protocols, ours show obvious performance improvements.
- The proposed LIAMKA protocol resists more attacks, including future quantum types, than current AMKA protocols.

1.2 | Related work

To improve the authentication efficiency, Mohammadali and others [9] presented an identity-based key agreement protocol based on elliptical curve cryptography (ECC). Gupta and Biswas [10] proposed an efficient identity-based key agreement protocol that uses pairing over an additive group. Dang and others [11] presented secure identity-based AKA protocols without pairing. Ren and Yoneyama [12] designed a hierarchical identity-based authenticated key exchange using a standard model. Tseng and others [13] proposed a leakage-resilient identity-based authenticated key-exchange protocol for resource-limited devices. Deng and others constructed an identity-based AKA protocol [14] for vehicular ad hoc networks.

Because all of the abovementioned AKA protocols use traditional cryptographic constructs, they are vulnerable to quantum attacks. To enhance their security, lattice-based cryptography has been applied to resist quantum attacks. Lattice primitives have stronger security than worst-case hardness assumptions and are relatively efficient [15]. Ajtai [16] introduced the first lattice hardness assumption. Wang and others [17] expanded the SIS/ISIS problems to lattice Bi-SIS/Bi-ISIS and CBi-ISIS/DBi-ISIS problems, and based on their hardness assumptions, a lattice-based two-party key-agreement protocol was designed. However, Gupta and Biswas [18] found that the proposed scheme of Wang and others [17] did not provide authentication. Furthermore, it fails to resist man-in-the-middle attacks [18]. Gupta and Biswas [19] then proposed two improved protocols using lattice-based signatures and signcryption. Islam and Zeadally [20] designed a novel two-party AKA protocol [20], and based on the CBi-ISIS and Bi-SIS hardness assumptions, it was found to be provably secure in the ROM. Gupta and others [21] presented an efficient lattice-based two-party key-agreement protocol with a formal security proof. Rana and Mishra [22] exploited ring-learning-with-errors problems to construct a new key agreement protocol. However, these lattice-based two-party AKA schemes are not efficient [19–22]. Moreover, they can only generate one session key when run at a time.

To solve the problem of generating only one key at a time when the protocol is running, Harn and Lin designed the first AMKA scheme [6]. Since the pioneering work on AMKE [6], many related protocols have been proposed [23]. Yen and Joye [24] showed that the Harn–Lin protocol [6] suffers from forgery attacks. Wu and others [25] found that Yen–Joye's improved protocol [24] suffers from the same vulnerability as that of Harn–Lin. Furthermore, the improved Harn–Lin protocol [26] cannot provide perfect forward security for all session keys [27]. In 2008, Lee and others [28] proposed an AMKA protocol based on ECC that uses bilinear pairings. However, Vo and others [29] demonstrated that Lee and others' protocol suffers from impersonation attacks, and Farash and others [30] confirmed that it suffers from forgery attacks. Hence, if the long-term private keys of both participants and one session key are revealed, other session keys will also be exposed. Farash and others [30] showed that Vo and others' protocol [29] suffers from forgery and reflection attacks, whereas Cheng [31]

proved that Farash and others' protocol [32] cannot resist a combination of key compromise impersonation and parallel session attacks.

A few identity-based AMKA protocols have been constructed [27, 33]. However, to the best of our knowledge, there are no lattice- and identity-based two-party AMKA protocols in the literature.

1.3 | Article structure

The remainder of this article is constructed as follows. Section 2 introduces related lattice-based cryptographic security assumptions, Section 3 describes the proposed LIAMKA protocol, and Section 4 defines the security model and confirms its accuracy and sufficient security. Section 5 presents a performance analysis of the proposed protocol, and Section 6 concludes this paper.

2 | PRELIMINARIES

In this section, the various concepts, mathematical definitions, and related cryptographic security assumptions used in the proposed LIAMKA protocol are briefly reviewed.

2.1 | Lattice

In geometry, a lattice is a set of points with a periodic arrangement structure in m-dimensional space. In algebra, a lattice is a discrete additive subgroup of R^m.

Definition 1. Given a set of linearly independent vectors, $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n R^m$, a lattice can be defined as the set of all integral combinations of these vectors:

$$\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} \mathbf{b}_i x_i | x_i \quad \mathbb{Z} \right\}, \tag{1}$$

where integers m and n are the dimension and rank of the lattice, respectively. If $m = n$, the lattice is a full rank lattice. By matrix $\mathbf{M}R^{m\times n}$, we denote $[\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n]$. Thus, its generated lattice can be defined as $\mathcal{L}(\mathbf{M}) = \{\mathbf{Mx} : \mathbf{x} Z^n\}$.

Definition 2. We assume that $q \in Z$ is the modulus. The q-ary lattice, $\mathcal{L}$, is an integer lattice that satisfies $qZ^n \subseteq L \subseteq Z^n$. Given an integer modular, q, and a modular matrix, $\mathbf{M}Z_q^{m\times n}$, where $m < n$ (e.g., $n = O(n\log n)$ and $q = O(n^2)$), two m-dimensional q-ary lattices, Λ_q and $\Lambda_q^{\perp}$, can be written as

$$\begin{aligned} \Lambda_q &= \left\{ a : \mathrm{a} = \mathbf{M}^{\mathrm{T}} \mathbf{b} \bmod q, \forall \mathbf{b} \right\}, \\ \Lambda_q^{\perp} &= \{ a : \mathbf{M}\mathrm{a} = 0 \bmod q \}. \end{aligned} \tag{2}$$

2.2 | Cryptographic assumptions

In lattice-based cryptography, Definition 1 lattice inherits basic cryptographic hardness assumptions about the shortest vector problem (SVP) and closest vector problem. The SVP finds a short nonzero vector whose Euclidian norm is at its minimum on a given lattice, and the closest vector problem finds the closest vector to a given vector in the given lattice. A typical SVP problem can be extended to two standard shortest independent vector problems (SIVPs) in the worst case (i.e., SIVPγ and GapSVPγ, γ denotes the approximation factor). GapSVP is a decisional version of SVP, and SIVP can be seen as an extension of SVP. For space limitations, we omit their formal definitions, which can be found in many studies, such as Ren and Yoneyama [12] and Tan [23].

For a q-ary lattice, there are also two hardness assumptions: SIS and ISIS problems.

Definition 3. (SIS problem in a q-ary lattice): Given an integer modular, q, two integers, $m, n\,(m < n)$, a modular matrix, $\mathbf{M} \in Z_q^{m\times n}$, and $\beta > 0$, the goal is to find a nonzero vector, $\mathbf{x} \in Z^n \setminus \{\mathbf{0}\}$, that satisfies the equation, $\mathbf{Mx} = \mathbf{0}\,(\mathrm{mod}\, q)$, such that the l_2-norm, $||\mathbf{x}|| \le \beta$. The SIS problem involves solving a system of Diophantine equations. However, it is still difficult to obtain a small-norm solution.

Definition 4. (ISIS problem). Given a modular basic matrix, $\mathbf{M} \in Z_q^{m\times n}$, $\beta \in Z^+$, and a random vector, $\mathbf{b} \in Z_q^m$, the goal is to find a nonzero vector, $\mathbf{x} \in Z^n$, that satisfies the equation, $\mathbf{Mx} = \mathrm{b} \bmod q$, such that the l_2-norm, $||\mathbf{x}|| \le \beta$. As shown in the literature [15, 16, 34], the SIS and ISIS problems are just as difficult as SVP or SIVP.

Subsequently, Wang and others [17] extended the SIS/ISIS problems in $Z_q^{m\times n}$ to Bi-SIS problems and its inhomogeneous Bi-ISIS in $Z_q^{n\times n}$. The main difference is that Bi-SIS/Bi-ISIS problems use one rectangular modular basic matrix, $\mathbf{M}$, over Z_q, and SIS/ISIS problems use one square modular basic matrix, $\mathbf{M}$, over Z_q.

Definition 5. (Bi-SIS problem). Given a modulus, q, a modular basic matrix, $\mathbf{M} \in Z_q^{n\times n}$, with rank m, and $\beta \in Z^+$, the goal is to find two nonzero integer vectors, $\mathbf{x}, \mathbf{y} \in Z^n$, such that

$$\begin{aligned} \mathbf{Mx} &= 0 \bmod q, \ \|\mathbf{x}\| \le \beta, \\ \mathbf{y}^{\mathrm{T}}\mathbf{M} &= 0^{\mathrm{T}} \bmod q, \ \left\|\mathbf{y}\right\| \le \beta. \end{aligned} \tag{3}$$

Definition 6. (Bi-ISIS problem). Given a modulus, q, a modular basic matrix, $\mathbf{M} \in Z_q^{n\times n}$, with rank m, $\beta \in Z^+$, and two vectors, $\mathbf{b}, \mathbf{c} \in Z_q^n$, the goal is to find two nonzero integer vectors, $\mathbf{x}, \mathbf{y} \in Z^n$, such that

$$\begin{aligned} \mathbf{Mx} &= \mathbf{b} \bmod q, \ \left\|\mathbf{x}\right\| \le \beta, \\ \mathbf{y}^{\mathrm{T}}\mathbf{M} &= \mathbf{c}^{\mathrm{T}} \bmod q, \ \|\mathbf{y}\| \le \beta. \end{aligned} \tag{4}$$

Proposition 1. *([17]). Given any poly bounded $n, \beta = poly(m)$, and prime $q \ge \beta\sqrt{\omega(m\log m)}$, typical Bi-SIS/Bi-ISIS problems are as difficult as the approximate SIVPγ and GapSVPγ problems in the worst case with certain $\gamma = \beta O(\sqrt{n})$.*

Bi-SIS/Bi-ISIS problems are considered difficult and inherit hardness assumptions. CBi-ISIS problem and assumption are defined in Wang and others [17] as follows:

Definition 7. (CBi-ISIS problem). Given a modulus, q, a modular basic matrix, $\mathbf{M} \in Z_q^{n\times n}$, with rank m, $\beta \in R^+$, and a tuple $< \mathbf{M}, \mathbf{Mx}, \mathbf{y}^{\mathrm{T}}\mathbf{M} >$, where $\mathbf{x}$ and $\mathbf{y}$ are two nonzero integer vectors in Z^n, $\left\|\mathbf{x}\right\| \le \beta$, and $\left\|\mathbf{y}\right\| \le \beta$, the goal is to find $\mathbf{y}^{\mathrm{T}}\mathbf{Mx} \bmod q$.

Definition 8. (CBi-ISIS hardness assumption): Given a security parameter, m, $n \approx poly(m)$, prime $q \approx poly(m)$, and $\beta \approx poly(m)$, such that $q \ge \beta\sqrt{\omega(m\log m)}$, $\mathbf{D} = \left\{\mathbf{z} \in Z^n : \left\|\mathbf{z}\right\| \le \beta\right\}$, a random modular basic matrix, $\mathbf{M} \in Z_q^{n\times n}$, with rank m for any probabilistic polynomial-time (PPT) algorithm, A, $\Pr[A(\mathbf{M}, \beta, \mathbf{Mx}, \mathbf{y}^{\mathrm{T}}\mathbf{M}) = \mathbf{y}^{\mathrm{T}}\mathbf{Mx} : \mathbf{x}, \mathbf{y} \in \mathbf{D}] \le negl(m)$, holds where $negl(\cdot)$ is a negligible function.

3 | PROPOSED LIAMKA SCHEME

Here, we present our LIAMKA protocol in which both participants (Bob and Alice) negotiate session keys in a secure manner. We assume that Alice has identity ID_1 and initiates the protocol, and Bob has identity ID_2 and responds to Alice's sender request. A trusted PKG center is responsible for extracting the private key corresponding to the identity of each participant in an offline mode. The proposed LIAMKA scheme comprises three phases: Setup, Private-key-Extract, and Session-key-Agreement.

3.1 | Setup phase

During the Setup phase, a trustworthy PKG generates the system parameters. Given security parameter m, the PKG executes the following operations:

- Given security parameter m, set $\beta \approx poly(m)$, and select a prime, $q \ge \beta\sqrt{\omega(m\log m)}$, and an integer, $n \ge 2m\log q$. Let $\mathbf{M} \in Z_q^{n\times n}$ be a modular basic matrix with rank m, and let $\mathbf{D} = \left\{\mathbf{d} \in Z^n : \left\|\mathbf{d}\right\| \le \beta\right\}$.
- Choose a random vector, $\mathbf{d} \in \mathbf{D}$, and calculate the master public key, $\mathbf{P} = \mathbf{d}^{\mathrm{T}}\mathbf{M} \bmod q$.
- Choose three cryptographic hash functions:

$$\begin{aligned} H_1 &: \{0,1\}^* \times Z_q^{1\times n} \times Z_q^{1\times n} \mathrm{Z}_{\mathrm{q}}^{1\times \mathrm{n}} \to Z_q^*, \\ H_2 &: \{0,1\}^* \times \{0,1\}^* \times Z_q^{n\times 1} \times Z_q^{1\times n} \mathrm{Z}_{\mathrm{q}}^{1\times \mathrm{n}} \to Z_q^*, \\ H3 &: \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times Z_q^{1\times n} \times Z_q^{1\times n} \\ &\quad \times Z_q^{n\times 1} \times Z_q^{1\times n} \times Z_q^n \to Z_q^* \mathrm{Z}_{\mathrm{q}}^{1\times \mathrm{n}}. \end{aligned}$$

The PKG maintains $\mathbf{d}$ as the master secret key, and the public system parameters are $\Delta = \{n, q, \mathbf{M}, \mathbf{P}, H_1(\cdot), H_2(\cdot), H_3(\cdot)\}$.

3.2 | Private-key-extract phase

Each participant obtains a private key by interacting with the PKG over a secure channel. This phase consists of the following operations:

- Each participant sends their identity $ID_i, i = 1, 2$ to the PKG.
- After receiving ID_i, the PKG first verifies its authenticity.
- The PKG chooses a random vector $\mathbf{r}_i \in Z_q^n$ and computes

$$\begin{aligned}\mathbf{P}_i &= \mathbf{r}_i^{\mathrm{T}}\mathbf{M},\\ h_i &= H_1(ID_i||\mathbf{P}_i||\mathbf{P}),\\ \mathbf{d}_i &= (\mathbf{r}_i + h_i\mathbf{d}) \bmod q.\end{aligned} \tag{5}$$

- The PKG sends the public/private key pair $(\mathbf{P}_i, \mathbf{d}_i)$ to the participant with identity ID_i via a secure channel.
- The participant with identity ID_i first calculates $h_i = H_1\left(ID_i \middle\| \mathbf{P}_i \middle\| \mathbf{P}\right)$ and validates the key pair $(\mathbf{P}_i, \mathbf{d}_i)$ by checking whether the following equation holds:

$$\mathbf{d}_i^{\mathrm{T}}\mathbf{M} = \mathbf{P}_i + h_i\mathbf{P} \bmod q. \tag{6}$$

3.3 | Session-key-agreement phase

Suppose that Alice with ID_1 and Bob with ID_2 attempt to agree on session keys over a public channel. Assume that they have their respective public–private key pairs $(\mathbf{P}_1, \mathbf{d}_1)$ and $(\mathbf{P}_2, \mathbf{d}_2)$. The participants can agree on eight keys in one execution of the proposed scheme using the following procedures.

Step 1: Alice first chooses two random vectors, $\mathbf{x}_1, \mathbf{x}_2 \in Z_q^n$, such that $\left\|\mathbf{x}_i\right\| \le \beta$, $i = 1, 2$, and calculates

$$\mathbf{U}_{1i} = \mathbf{M}\mathbf{x}_i \bmod q, \mathbf{V}_{1i} = \mathbf{x}_i^{\mathrm{T}}\mathbf{M} \bmod q, \tag{7}$$

$$\mathbf{S}_1 = \mathbf{d}_1 + \sum_{i=1}^{2} H_2(ID_i||\mathbf{T}_1||\mathbf{U}_{1i}||\mathbf{V}_{1i})\mathbf{x}_i \bmod q. \tag{8}$$

Step 2: Alice issues her identity, ID_1, her key, $\mathbf{P}_1$, and the message $\{\mathbf{U}_{11}, , \mathbf{V}_{11}, \mathbf{U}_{12}, \mathbf{V}_{12}\mathbf{S}_1\}$ with time stamp $\mathbf{T}_1$ to Bob.

Step 3: Upon receiving the request from Alice, Bob first validates whether time stamp $\mathbf{T}_1$ falls in the validation interval. Bob authenticates Alice and validates the message $\{\mathbf{U}_{11}, \mathbf{V}_{11}, \mathbf{U}_{12}, \mathbf{V}_{12}, \mathbf{S}_1\}$ by checking whether the following equation holds:

$$\mathbf{S}_1^{\mathrm{T}}\mathbf{M} = \mathbf{P}_1 + h_1\mathbf{P} + \sum_{i=1}^{2} H_2\left(ID_i \middle\| \mathbf{T}_1 \middle\| \mathbf{U}_{1i} \middle\| \mathbf{V}_{1i}\right)\mathbf{V}_{1i} \bmod q. \tag{9}$$

Step 4: If (9) does not hold, then Bob aborts. Otherwise, he randomly chooses two random vectors, $\mathbf{y}_1, \mathbf{y}_2 \in Z_q^n$, such that $\left\|\mathbf{y}_i\right\| \le \beta$, $i = 1, 2$, and calculates

$$\mathbf{U}_{2i} = \mathbf{M}\mathbf{y}_i \bmod q, \mathbf{V}_{2i} = \mathbf{y}_i^{\mathrm{T}}\mathrm{M} \bmod q, \tag{10}$$

$$\mathbf{S}_2 = \mathbf{d}_2 + \sum_{i=1}^{2} H_2\left(ID_i \middle\| \mathbf{T}_2 \middle\| \mathbf{U}_{2i} \middle\| \mathbf{V}_{2i}\right)\mathrm{y}_i \bmod q. \tag{11}$$

Step 5: Bob issues his identity, ID_2, his key, $\mathbf{P}_2$, and the message $\{\mathbf{U}_{21}, \mathbf{V}_{21}, \mathbf{U}_{22}, \mathbf{V}_{22}, \mathbf{S}_2\}$ with time stamp $\mathbf{T}_2$ to Alice. Finally, Bob computes the shared values.

$$K_{ij}^{01} = \mathbf{V}_{1i}\mathbf{y}_j \bmod q, \tag{12}$$

$$K_{ij}^{11} = \mathbf{y}_i^{\mathrm{T}}\mathbf{U}_{1j} \bmod q, \tag{13}$$

where $i,j = 1, 2$ and the session keys are as follows:

$$\begin{aligned}
K_{B1} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{21}||\mathbf{V}_{11}||K_{11}^{01}\right),\\
K_{B2} &= H_3\left(ID_1 \middle\| ID_2 \middle\| \mathbf{T}_1 \middle\| \mathbf{T}_2 \middle\| \mathbf{P}_1 \middle\| \mathbf{P}_2 \middle\| \mathbf{U}_{22} \middle\| \mathbf{V}_{11} \middle\| K_{12}^{01}\right),\\
K_{B3} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{21}||\mathbf{V}_{12}||K_{21}^{01}\right),\\
K_{B4} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{22}||\mathbf{V}_{12}||K_{22}^{01}\right),\\
K_{B5} &= H_3\left(ID_1 \middle\| ID_2 \middle\| \mathbf{T}_1 \middle\| \mathbf{T}_2 \middle\| \mathbf{P}_1 \middle\| \mathbf{P}_2 \middle\| \mathbf{U}_{11} \middle\| \mathbf{V}_{21} \middle\| K_{11}^{11}\right)\\
K_{B6} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{12}||\mathbf{V}_{21}||K_{12}^{11}\right),\\
K_{B7} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{11}||\mathbf{V}_{22}||K_{21}^{11}\right),\\
K_{B8} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{12}||\mathbf{V}_{22}||K_{22}^{11}\right).
\end{aligned} \tag{14}$$

Step 6: Upon receiving the message $\{\mathbf{U}_{21}, \mathbf{V}_{21}, \mathbf{U}_{22}, \mathbf{V}_{22}, \mathbf{S}_2, ID_2, \mathbf{T}_2, \mathbf{P}_2\}$, Alice first checks whether time stamp $\mathbf{T}_2$ falls in the validation interval. She authenticates Bob and checks whether the message is valid by using the following equation:

$$\mathbf{S}_2^{\mathrm{T}}\mathbf{M} = \mathbf{P}_2 + h_2\mathbf{P} + \sum_{i=1}^{2} H_2\left(ID_i \middle\| \mathbf{T}_2 \middle\| \mathbf{U}_{2i} \middle\| \mathbf{V}_{2i}\right)\mathbf{V}_{2i} \bmod q. \tag{15}$$

Step 7: If (15) does not hold, Alice abandons this authentication request. Otherwise, Alice calculates the shared values.

$$K_{ij}^{02} = \mathbf{x}_i^{\mathrm{T}}\mathbf{U}_{2j} \bmod q, \tag{16}$$

$$K_{ij}^{12} = \mathbf{V}_{2i}\mathbf{x}_j \bmod q, \tag{17}$$

where $i,j = 1,2$ and the session keys are as follows:

$$\begin{aligned}
K_{A1} &= H_3\left(ID_1 \middle\| ID_2 \middle\| \mathbf{T}_1 \middle\| \mathbf{T}_2 \middle\| \mathbf{P}_1 \middle\| \mathbf{P}_2 \middle\| \mathbf{U}_{21} \middle\| \mathbf{V}_{11} \middle\| K_{11}^{02}\right)\\
K_{A2} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{22}||\mathbf{V}_{11}||K_{12}^{02}\right),\\
K_{A3} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{21}||\mathbf{V}_{12}||K_{21}^{02}\right),\\
K_{A4} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{22}||\mathbf{V}_{12}||K_{22}^{02}\right),\\
K_{A5} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{11}||\mathbf{V}_{21}||K_{11}^{12}\right),\\
K_{A6} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{12}||\mathbf{V}_{21}||K_{12}^{12}\right),\\
K_{A7} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{11}||\mathbf{V}_{22}||K_{21}^{12}\right),\\
K_{A8} &= H_3\left(ID_1||ID_2||\mathbf{T}_1||\mathbf{T}_2||\mathbf{P}_1||\mathbf{P}_2||\mathbf{U}_{12}||\mathbf{V}_{22}||K_{22}^{12}\right).
\end{aligned} \tag{18}$$

4 | SECURITY ANALYSIS

This section confirms the correctness, security model, and security proofs of the proposed LIAMKA protocol.

4.1 | Correctness

Theorem 1. *The proposed LIAMKA protocol is correct if $H_1(\cdot)$, $H_2(\cdot)$, and $H_3(\cdot)$ are collision-resistant hash functions. That is, after the successful execution of the protocol, both Alice and Bob agree on common session keys $K_{At} = K_{Bt}$, $t = 1,2,...,8$.*

Proof. First, participants can verify the authenticity of messages issued by one another using (9) and (15). For $j = 1,2$, we have

$$\begin{aligned}
\mathbf{S}_j^{\mathrm{T}}\mathbf{M} &= \left(\mathbf{d}_j + \sum_{i=1}^{2} H_2\left(ID_i||\mathbf{T}_j||\mathbf{U}_{ji}||\mathbf{V}_{ji}\right)\mathbf{y}_i\right)^{\mathrm{T}}\mathbf{M} \bmod q,\\
&= \mathbf{d}_j{}^{\mathrm{T}}\mathbf{M} + \sum_{i=1}^{2} H_2\left(ID_i||\mathbf{T}_j||\mathbf{U}_{ji}||\mathbf{V}_{ji}\right)\mathbf{y}_i^{\mathrm{T}}\mathbf{M} \bmod q,\\
&= \mathbf{P}_j + h_j\mathbf{P} + \sum_{i=1}^{2} H_2\left(ID_i \middle\| \mathbf{T}_j \middle\| \mathbf{U}_{ji} \middle\| \mathbf{V}_{ji}\right)\mathbf{V}_{ji} \bmod q.
\end{aligned}$$

Second, Bob can compute the shared values, $K_{ij}^{01} = \mathbf{V}_{1i}\mathbf{y}_j \bmod q$ and $K_{ij}^{11} = \mathbf{y}_i^{\mathrm{T}}\mathbf{U}_{1j} \bmod q$, whereas Alice can calculate $K_{ij}^{02} = \mathbf{x}_i^{\mathrm{T}}\mathbf{U}_{2j} \bmod q$ and $K_{ij}^{12} = \mathbf{V}_{2i}\mathbf{x}_j \bmod q$, where $i,j = 1,2$.

$$\begin{aligned}
K_{ij}^{01} &= \mathbf{V}_{1i}\mathbf{y}_j \bmod q = \mathbf{x}_i^{\mathrm{T}}\mathbf{M}\mathbf{y}_j \bmod q\\
&= \mathbf{x}_i^{\mathrm{T}}\mathbf{U}_{2j} \bmod q = K_{ij}^{02},
\end{aligned}$$

$$\begin{aligned}
K_{ij}^{11} &= \mathbf{y}_i^{\mathrm{T}}\mathbf{U}_{1j} \bmod q = \mathbf{y}_i^{\mathrm{T}}\mathbf{M}\mathbf{x}_j \bmod q\\
&= \mathbf{V}_{2i}\mathbf{x}_j \bmod q = K_{ij}^{12}.
\end{aligned}$$

Hence, keys K_{At} and K_{Bt} are equal for $t = 1,2,...,8$.

4.2 | Security model

In this section, we discuss the formal security model for identity-based AMKA schemes. Assuming that PKG is a trustworthy entity, it reliably validates users' real identities. Moreover, a secure channel must exist between the PKG and the participants to protect their private keys. In the proposed protocol, the two communicating parties communicate using a nonsecure channel.

A session identifier is composed of the identities of the participants and the messages transmitted during AMKA execution. Specifically, based on session identifier $\prod_{ID,ID^*}^{f}$, we denote the f^{th} key agreement between initiator ID and partner ID^*. Let $\prod_{ID,ID^*}^{f} = (role, ID, ID^*, message_1, ..., message_n)$, where $role \in \{\text{Initiator}, \text{Partner}\}$ annotates its AKA role, ID and ID^* are the identities of one party and the partner, respectively, and $message_i$ is the i^{th} message sent by the parties. If one session and the other executed by the other party transmit the same message, even in a different order, they are still matching sessions. For example, in the proposed two-round protocol, if the ID entity executes session $(\text{Initiator}, ID, ID^*, message_1, message_2)$, then the matching session is executed by ID^*, and the session identifier is $(\text{Partner}, ID^*, ID, message_1, message_2)$.

The AKA security model is formalized in Bellare and others [3]. In this section, we further extend this and other security models [35] to the identity-based AMKA version [36] based on the popular Real-or-Random model [37]. Let a PPT adversary, $\mathcal{A}$, attempt to breach the security of the identity-based AMKA protocol. Probabilistic Turing machine $\mathcal{A}$ can control all communications over the open channel. The security model is simulated using a challenge–response game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$. This allows the adversary to mount all possible attacks during the run. We simulate various security attacks using all possible oracle queries, which model the capabilities of a real attacker. In this game, $\mathcal{A}$ issues any sequence of the oracle queries listed below to $\mathcal{C}$, which responds to the queries in turn.

- *Setup*(m). Given security parameter m, C outputs public parameters Δ, a pair of master secret and public keys, (msk, mpk), and returns (mpk, Δ) to A, keeping msk a secret.
- *Extract*(ID_i). In response to this query of identity ID_i, C extracts the private key and returns it to $\mathcal{A}$.
- *Execute*(ID_i, ID_j). This query models eavesdropping attacks on an honest execution between participants ID_i and ID_j. The output is assigned one session identifier, $\prod^{f}_{ID_i,ID_j}$.
- *Send*(ID_i, ID_j, com). This query models an active attack. The oracle query model sends message com to ID_i on behalf of ID_j and receives an actual response from participant ID_i.
- *Reveal*$\left(\prod^{f}_{ID_i,ID_j}\right)$. This query models a known session-key attack. If no session key is defined, for instance, $\prod^{f}_{ID_i,ID_j}$, then the output of this query is the invalid symbol, $\perp$. Otherwise, it returns some or all of the current session keys of $\prod^{f}_{ID_i,ID_j}$ using the AMKA protocol.
- *Corrupt*(ID). The query models the capability of $\mathcal{A}$, which can acquire the static private key of participant ID.
- *Ephemeral_Key_Reveal* $\left(\prod^{f}_{ID_i,ID_j}, ID_i\right)$. The query models the capability of $\mathcal{A}$, which can get the ephemeral key of participant ID_i during session $\prod^{f}_{ID_i,ID_j}$ (possibly incomplete).
- *Test* $\left(\prod^{f}_{ID_i,ID_j}\right)$. This is invoked once per fresh oracle. $\mathcal{A}$ chooses an accepted fresh oracle, $\prod^{f}_{ID_i,ID_j}$, as an authenticator or one session key. In response to this query, $\mathcal{C}$ randomly selects bit $b \in \{0,1\}$. If $b = 1$, C outputs one session key exchanged between ID_i and its partner during session $\prod^{f}_{ID_i,ID_j}$. Otherwise, $\mathcal{C}$ outputs a random value as the session key.

The definition of the freshness of $\prod^{f}_{ID_i,ID_j}$ is given below.

Definition 9. Let $\prod^{f}_{ID_i,ID_j}$ be a completed session owned by an honest party, ID_i, partnered with an honest peer, ID_j. Let $\prod^{f*}_{ID_j,ID_i}$ represent matching sessions. Session sid is fresh if none of the following conditions hold: any *Reveal* $\left(\prod^{f}_{ID_i,ID_j}\right)$ or *Reveal* $\left(\prod^{f*}_{ID_j,ID_i}\right)$ queries, if one has been made. If session $\prod^{f*}_{ID_j,ID_i}$ exists, $\mathcal{A}$ makes either both *Corrupt* (ID_i) and *Ephemeral_Key_Reveal* $\left(\prod^{f}_{ID_i,ID_j}, ID_i\right)$ queries or both *Corrupt*(ID_j) and *Ephemeral_Key_Reveal* $\left(\prod^{f*}_{ID_j,ID_i}, ID_j\right)$ queries. If session $\prod^{f*}_{ID_j,ID_i}$ does not exist, $\mathcal{A}$ makes either *Corrupt* (ID_j) or both *Corrupt* (ID_i) and *Ephemeral_Key_Reveal* $\left(\prod^{f}_{ID_i,ID_j}, ID_i\right)$ queries.

Session $\prod^{f}_{ID_i,ID_j}$ is not fresh if at least one participant in the session is controlled by the adversary. Thus, adversary $\mathcal{A}$ can reveal the controlled participant's long-term and ephemeral keys. In the formal security model of the identity-based AMKA protocol, $\mathcal{A}$ may issue any number of oracle queries to $\mathcal{C}$, but $\mathcal{A}$ is allowed to make the *Test* query only once per fresh session. At the end of any sequence of queries, $\mathcal{A}$ outputs a bit, b'. If $b' = b$, $\mathcal{A}$ wins the game.

Definition 10. The advantage, $\mathrm{Adv}^{IAKA}_{\mathcal{A}}$, of breaking the semantic security of the identity-based AMKA protocol within polynomial time-bound t can be defined by the probability of adversary $\mathcal{A}$'s successful output of bit b' for a fresh session,

$$\mathrm{Adv}^{IAKA}_{\mathcal{A}}(t) = \left|\Pr[b' = b] - \frac{1}{2}\right|.$$

Definition 11. An identity-based AMKA protocol is *secure* if

- In the presence of a benign adversary in a session and its matching session, both participants always believe that they have the same session key, which is distributed uniformly.
- For any PPT-bounded adversary, $\mathcal{A}$, the advantage, $\mathrm{Adv}^{IAKA}_{\mathcal{A}}(t)$, in the game above is negligible.

4.3 | Formal security analysis

In this section, we analyze the security of the proposed LIAMKA protocol using the security model of the identity-based AMKA protocol.

Kudla and Paterson [38] proved that AKA Protocol II security can be transformed to the proof of a related AKA protocol, π, which is identical. However, Protocol II produces a hashed session key, whereas π employs the input string of the hash function as the session key.

Theorem 2. *Given security parameter m, integer* $n \approx poly(m)$, *prime* $q \approx poly(m)$, *and*

β *poly(m) such that* $q \geq \beta$, *let* $\mathbf{D} = \left\{\mathbf{z} \in Z^n : \left\|\mathbf{z}\right\| \leq \beta\right\}$. *Let* $\mathbf{M}$ *be a random modular basic matrix in* $Z_q^{n\times n}$ *with rank m. Next, assume there exists a polynomial time-bounded adversary,* $\mathcal{A}$, *that breaches the security of the LIAMKA scheme with a nonnegligible advantage. Hence, there is an instance of the Bi-ISIS or CBi-ISIS problem that can be solved with nonnegligible probability. That is, for any PPT adversary,* $\mathcal{A}$, *the proposed LIAMKA protocol is secure upon both Bi-ISIS and CBi-ISIS assumptions on the lattice in the ROM.*

Proof: *As described in Section 4.1, the proposed LIAMKA protocol satisfies the first condition of Definition 11. Next, we use the techniques in Bellare and Rogaway [37] and a proof by contradiction to show that the proposed LIAMKA protocol satisfies the second condition of Definition 11.*

Assume that adversary $\mathcal{A}$ runs PPT algorithm $\mathcal{C}$ to break the security of the proposed protocol. The model is explained by a game between challengers $\mathcal{C}$ and $\mathcal{A}$, where $\mathcal{A}$ utilizes $\mathcal{C}$ to break the security of our proposed scheme; however, $\mathcal{C}$ uses $\mathcal{A}$ as a subroutine to solve the Bi-ISIS and CBi-ISIS problems. An example of the Bi-ISIS problem computes $\mathbf{y} \in Z_q^n$ from two given triples $(\mathbf{M}, \beta, \mathbf{My})$ and $(\mathbf{M}, \beta, \mathbf{y}^T\mathbf{M})$, where a modular basic matrix, $\mathbf{M} \in Z_q^{n\times n}$, has a rank, m, whereas the instance of the Bi-ISIS problem computes $\mathbf{y}^T\mathbf{Mx}$ from the given triple $<\mathbf{M}, \mathbf{Mx}, \mathbf{y}^T\mathbf{M}>$, where $\mathbf{M}$ is a modular basic matrix in $Z_q^{n\times n}$ with rank m. x and y are two unknown nonzero integer vectors.

$\mathcal{A}$ can request various oracle queries from $\mathcal{C}$, which return a response in the following ways. For each type of query, an initially empty list is maintained by $\mathcal{C}$ to store the responses.

- *Setup*(m). To answer the query issued by adversary $\mathcal{A}$, $\mathcal{C}$ executes the setup of the proposed LIAMKA protocol. Then, $\mathcal{C}$ chooses a random vector, $\mathbf{d} \in Z^n$, and generates the global public parameters, $\Delta = \{n, q, \mathbf{M}, \mathbf{P}, H_1(\cdot), H_2(\cdot), H_3(\cdot)\}$. $\mathcal{C}$ returns Δ to $\mathcal{A}$ and keeps the master key, d, secret.
- H_1*_Query*. To answer this query, $\mathcal{C}$ maintains H_1*_list*, which is initially empty but takes the form of tuples, such as $(ID_i, \mathbf{P}_i, \mathbf{P}, h_i)$. If H_1*_Query* about the triple $(ID_i, \mathbf{P}_i, \mathbf{P})$ is requested, $\mathcal{C}$ first searches H_1*_list* for the triple. If $(ID_i, \mathbf{P}_i, \mathbf{P}, *)$ exists in the H_1*_list*, $\mathcal{C}$ returns the corresponding fourth element of the record. Otherwise, $\mathcal{C}$ selects an element, $h_i \in Z_q^*$, and appends a tuple $(ID_i, \mathbf{P}_i, \mathbf{P}, h_i)$ to the H_1*_list*. Then, $\mathcal{C}$ returns h_i in response to H_1*_Query*.
- H_2*_Query*. $\mathcal{C}$ maintains an initially empty H_2*_list* with entries $(ID, \mathbf{T}, \mathbf{U}, \mathbf{V}, \delta)$. If $\mathcal{A}$ requests H_2*_Query* for message $(ID_i, \mathbf{T}, \mathbf{U}_i, \mathbf{V}_i)$, $\mathcal{C}$ first searches H_2*_list* for the tuple. If $(ID_i, \mathbf{T}, \mathbf{U}_i, \mathbf{V}_i, *)$ exists in H_2*_list*, $\mathcal{C}$ returns the last element of the record to $\mathcal{A}$. Otherwise, $\mathcal{C}$ selects an integer, $\delta_i \in Z_q^*$, and appends a tuple $(ID_i, \mathbf{T}, \mathbf{U}_i, \mathbf{V}_i, \delta_i)$ to H_2*_list*. Then, $\mathcal{C}$ returns δ_i as the output to $\mathcal{A}$.
- H_3*_Query*. $\mathcal{C}$ maintains an initially empty H_3*_list* with entries $\left(ID_i, ID_j, \mathbf{T}_i, \mathbf{T}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{U}, \mathbf{V}, K_{ij}^f, K\right)$. If $\mathcal{A}$ requests H_3*_Query* with $\left(ID_i, ID_j, \mathbf{T}_i, \mathbf{T}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{U}, \mathbf{V}, K_{ij}^f\right)$, $\mathcal{C}$ first searches H_3*_list* for the tuple $\left(ID_i, ID_j, \mathbf{T}_i, \mathbf{T}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{U}, \mathbf{V}, K_{ij}^f, *\right)$. If a tuple exists in the list, $\mathcal{C}$ returns the last element as the output. Otherwise, $\mathcal{C}$ selects an element, $K \in Z_q^*$, and appends the tuple $\left(ID_i, ID_j, \mathbf{T}_i, \mathbf{T}_j, \mathbf{P}_i, \mathbf{P}_j, \mathbf{U}, \mathbf{V}, K_{ij}^f, K\right)$ to H_3*_list*. Then, $\mathcal{C}$ returns K as the output to $\mathcal{A}$.
- *Extract*(ID_i). To respond to this query, $\mathcal{C}$ maintains *Ext_list*, which is initially empty. When one oracle query about ID_i is requested, $\mathcal{C}$ first searches *Ext_list* for the tuple $(ID_i, *, \mathbf{P}, *)$. If $(ID_i, *, \mathbf{P}, *)$ exists in *Ext_list*, $\mathcal{C}$ returns the second and fourth elements of the record as a key pair. Otherwise, $\mathcal{C}$ chooses a random vector, $\mathbf{r}_i \in Z_q^n$, computes $\mathbf{P}_i = \mathbf{r}_i^T\mathbf{M}$, and produces response h_i to H_1*_Query* about $\left(ID_i \left\| \mathbf{P}_i \right\| \mathbf{P}\right)$. It then appends a tuple $(ID_i, \mathbf{P}_i, \mathbf{P}, h_i)$ to H_1*_list*. Finally, $\mathcal{C}$ calculates $\mathbf{d}_i = (\mathbf{r}_i + h_i\mathbf{d}) \bmod q$ and appends $(ID_i, \mathbf{P}_i, \mathbf{P}, \mathbf{d}_i)$ to *Ext_list*. $\mathcal{C}$ then returns the key pair $(\mathbf{P}_i, \mathbf{d}_i)$ to $\mathcal{A}$.
- *Execute*$\left(ID_i, ID_j\right)$. To answer this query, $\mathcal{C}$ maintains *Exe_list*, which is initially empty. $\mathcal{C}$ first checks whether ID_i or ID_j exists in *Ext_list*. If neither, $\mathcal{C}$ extracts the respective private keys and appends $(ID_i, \mathbf{P}_i, \mathbf{P}, \mathbf{d}_i)$ and $\left(ID_j, \mathbf{P}_j, \mathbf{P}, \mathbf{d}_j\right)$ to *Ext_list*. Then, $\mathcal{C}$ randomly chooses two vectors, $\mathbf{x}_1, \mathbf{x}_2 \in Z_q^n$, such that $||\mathbf{x}_i|| \leq \beta$, $i = 1, 2$, and calculates

$$\mathbf{U}_{1i} = \mathbf{M}\mathbf{x}_i \bmod q, \mathbf{V}_{1i} = \mathbf{x}_i^T\mathbf{M} \bmod q,$$

$$\mathbf{S}_1 = \mathbf{d}_1 + \sum_{i=1}^{2} H_2\left(ID_i \left\| \mathbf{T}_1 \right\| \mathbf{U}_{1i} \left\| \mathbf{V}_{1i}\right.\right)\mathbf{x}_i \bmod q.$$

Let $message_1$ be $\{\mathbf{U}_{11}, \mathbf{V}_{11}, \mathbf{U}_{12}, \mathbf{V}_{12}, \mathbf{S}_1, \mathbf{T}_1\}$. Similarly, $\mathcal{C}$ generates $\{\mathbf{U}_{21}, \mathbf{V}_{21}, \mathbf{U}_{22}, \mathbf{V}_{22}, \mathbf{S}_2, \mathbf{T}_2\}$ as $message_2$. Let $\prod_{ID_i, ID_j}^f$ be $\left\{\text{Initiator}, ID_i, ID_j, message_1, message_2\right\}$ and $\prod_{ID_j, ID_i}^{f*}$ be $\left\{\text{Partner}, ID_j, ID_i, message_1, message_2\right.$. Then, $\mathcal{C}$ appends $\left(\prod_{ID_i, ID_j}^f, \prod_{ID_j, ID_i}^{f*}, sek\right)$ to *Exe_list* where *sek* is one or all session keys of $\prod_{ID_i, ID_j}^f$ and $\prod_{ID_j, ID_i}^{f*}$.

- *Send*(ID_i, ID_j, com). If message *com* is not empty, the query is responded to by $\mathcal{C}$, as in the proposed LIAMKA protocol. Otherwise, $\mathcal{C}$ searches for tuples $(ID_i, *, \mathbf{P}, *)$ and $(ID_j, *, \mathbf{P}, *)$ in *Ext_list*. If $(ID_i, *, \mathbf{P}, *)$ or $(ID_j, *, \mathbf{P}, *)$ are not in *Ext_list*, $\mathcal{C}$ executes *Extract*() and H_1*_Query* for ID_i or ID_j. Next, $\mathcal{C}$ verifies whether session $\prod^{f}_{ID_i,ID_j}$, which is partnered with $\prod^{f*}_{ID_j,ID_i}$, is in *Exe_list*. If the tuple $\left(\prod^{f}_{ID_i,ID_j}, \prod^{f*}_{ID_j,ID_i}, *\right)$ exists in *Exe_list*, $\mathcal{C}$ returns $\prod^{f*}_{ID_j,ID_i}$; otherwise, $\mathcal{C}$ completes *Execute*(ID_i, ID_j) query for ID_i and ID_j.
- *Reveal*(sid). To respond to this query, $\mathcal{C}$ maintains an initially empty list, *K_list*, with entries $\left(\prod^{f}_{ID_i,ID_j}, \prod^{f*}_{ID_j,ID_i}, K\right)$. In response, $\mathcal{C}$ first checks whether $\prod^{f}_{ID_i,ID_j}$ exists in *K_list*. If no session key of instance $\prod^{f}_{ID_i,ID_j}$ is defined or it has been requested by *Test* query, this query will output $\perp$. Otherwise, it returns the current session key of session $\prod^{f}_{ID_i,ID_j}$.
- *Corrupt*(ID_i). In response to this query, $\mathcal{C}$ first searches *Ext_list* for ID_i. If a quadruple $(ID_i, \mathrm{P}_i, \mathrm{P}, \mathrm{d}_i)$ containing ID_i is in *Exe_list*, $\mathcal{C}$ returns it to $\mathcal{A}$ as the output. Otherwise, $\mathcal{C}$ executes *Extract* query and returns the output to $\mathcal{A}$.
- *Ephemeral_Key_Reveal* $\left(\prod^{f}_{ID_i,ID_j}, ID_i\right)$. In response to this query, $\mathcal{C}$ returns the ephemeral keys, $\mathbf{x}_i \in Z_q^n$ $(i = 1, 2)$, of participant ID_i in session $\prod^{f}_{ID_i,ID_j}$ to $\mathcal{A}$.
- *Test*. When $\mathcal{A}$ executes *Test* query to oracle $\prod^{f}_{ID_i,ID_j}$, $\mathcal{C}$ randomly chooses four elements, $a_1, a_2, b_1, b_2 \in Z_q^*$, and calculates

$$\begin{aligned} \mathbf{U}_1 &= a_1\mathbf{M}\mathbf{y} \bmod q, \mathbf{U}_2 = a_2\mathbf{M}\mathbf{y} \bmod q \\ \mathbf{V}_1 &= a_1\mathbf{y}^{\mathrm{T}}\mathbf{M} \bmod q, \mathbf{V}_2 = a_2\mathbf{y}^{\mathrm{T}}\mathbf{M} \bmod q \\ \mathbf{U}_1' &= b_1\mathbf{M}\mathbf{y} \bmod q, \mathbf{U}_2' = b_2\mathbf{M}\mathbf{y} \bmod q \\ \mathbf{V}_1' &= b_1\mathbf{y}^{\mathrm{T}}\mathbf{M} \bmod q, \mathbf{V}_2' = b_2\mathbf{y}^{\mathrm{T}}\mathbf{M} \bmod q. \end{aligned} \tag{19}$$

Then, $\mathcal{C}$ executes H_2*_Query* about $(ID_i, \mathbf{T}_1, \mathbf{U}_1, \mathbf{V}_1)$. Let h_{21} be the response to H_2*_Query* regarding $(ID_i, \mathbf{T}_1, \mathbf{U}_1, \mathbf{V}_1)$. Then, $\mathcal{C}$ executes H_2*_Query* about $(ID_i, \mathbf{T}_1, \mathbf{U}_2, \mathbf{V}_2)$. If tuple $(ID_i, \mathbf{T}_1, \mathbf{U}_2, \mathbf{V}_2, *)$ exists in H_2*_list*, $\mathcal{C}$ randomly chooses another element, a_2, in Z_q^* and computes $\mathbf{U}_2 = a_2\mathbf{M}\mathbf{y} \bmod q$, and $\mathbf{V}_2 = a_2\mathbf{y}^{\mathrm{T}}\mathbf{M} \bmod q$ until tuple $(ID_i, \mathbf{T}_1, \mathbf{U}_2, \mathbf{V}_2, *)$ does not exist in H_2*_list*. $\mathcal{C}$ computes $h_{22} = (q - (h_{21}a_1 \bmod q))a_2^{-1} \bmod q$ as the response to H_2*_Query* about $(ID_i, \mathbf{T}_1, \mathbf{U}_2, \mathbf{V}_2)$. Similarly, $\mathcal{C}$ obtains h'_{21} as the response to H_2*_Query* about $(ID_i, \mathbf{T}_2, \mathbf{U}_1', \mathbf{V}_1')$ and calculates $h'_{22} = (q - (h'_{21}b_1 \bmod q))$ $b_2^{-1} \bmod q$ as the response to H_2*_Query* about $(ID_i, \mathbf{T}_2, \mathbf{U}_2', \mathbf{V}_2')$. $\mathcal{C}$ then returns $(\mathbf{U}_1, \mathbf{V}_1, \mathbf{U}_2, \mathbf{V}_2, *, \mathbf{T}_1; , \mathbf{U}_1', \mathbf{V}_1', \mathbf{U}_2', \mathbf{V}_2', * \mathbf{T}_2)$ to $\mathcal{A}$ and appends them to H_2*_list*.

After the completion of *Test*, $\mathcal{A}$ outputs tuple $(\mathbf{U}_1, \mathbf{V}_1, \mathbf{U}_2, \mathbf{V}_2, \mathbf{S}_1, \mathbf{T}_1)$ or $(\mathbf{U}_1', \mathbf{V}_1', \mathbf{U}_2', \mathbf{V}_2', \mathbf{S}_2, \mathbf{T}_2)$ to $\mathcal{C}$, which then checks whether the following equation holds:

$$\mathbf{S}_1^{\mathrm{T}}\mathbf{M} = \mathbf{P}_1 + h_1\mathbf{P} + \sum_{i=1}^{2} H_2(ID_i \,||\, \mathbf{T}_1||\mathbf{U}_i||\mathbf{V}_i)\mathbf{V}_i \bmod q. \tag{20}$$

If the above equation does not hold, $\mathcal{C}$ aborts the execution. Otherwise, according to the fork lemma [39], $\mathcal{C}$ may obtain another tuple $(\mathbf{U}_1, \mathbf{V}_1, \mathbf{U}_2, \mathbf{V}_2, \mathbf{S}_1^*, \mathbf{T}_1)$ with another hash value, h_{21}^* for message $(ID_i, \mathbf{T}_1, \mathbf{U}_1, \mathbf{V}_1)$ and h_{22}^* for message $(ID_i\mathbf{T}_1, \mathbf{U}_2, \mathbf{V}_2)$. Thus, $\mathbf{S}_1^{*\mathrm{T}}\mathbf{M} = \mathbf{P}_1 + h_1\mathbf{P} + \sum_{i=1}^{2} h_{2i}^*\mathbf{V}_i \bmod q$. By subtracting the above equation from (20), $\mathcal{C}$ can obtain

$$(\mathbf{S}_1^{*\mathrm{T}} - \mathbf{S}_1{}^{\mathrm{T}})\mathbf{M} = \sum_{i=1}^{2} (h_{2i}^* - h_{2i})\mathbf{V}_i \bmod q.$$

From (18), $\mathcal{C}$ can compute

$$\mathbf{y} = \frac{(\mathbf{S}_1^* - \mathbf{S}_1)}{[a_1(h_{21}^* - h_{21}) + a_2(h_{22}^* - h_{22})]} \bmod q. \tag{21}$$

Thus, $\mathcal{C}$ finds solution $\mathbf{y} \in Z_q^n$ to the Bi-ISIS problem for two given triples $(\mathbf{M}, \beta, \mathbf{M}\mathbf{y})$ and $(\mathbf{M}, \beta, \mathbf{y}^{\mathrm{T}}\mathbf{M})$. This contradicts the Bi-ISIS hardness assumption.

If adversary $\mathcal{A}$ can successfully guess bit b', one of the session keys $< ID_i, ID_j, \mathbf{T}_1, \mathbf{T}_2, \mathbf{U}, \mathbf{V}, K >$ corresponding to message $(\mathbf{U}_1, \mathbf{V}_1, \mathbf{U}_2, \mathbf{V}_2, \mathbf{d}_1, \mathbf{T}_1; \mathbf{U}_1', \mathbf{V}_1', \mathbf{U}_2', \mathbf{V}_2', \mathbf{d}_2, \mathbf{T}_2)$ is forged. Then, $\mathcal{C}$ computes $\mathbf{y}^{\mathrm{T}}\mathbf{M}\mathbf{x} = (a_1b_2)^{-1}K$ or $(a_2b_1)^{-1}K$. Thus, the given CBi-ISIS instance on input $< \mathbf{M}, \mathbf{M}\mathbf{x}, \mathbf{y}^{\mathrm{T}}\mathbf{M} >$ can be solved.

In summary, if the adversary can break the security of the proposed LIAMKA protocol with advantage $\mathrm{Adv}_{\mathcal{A}}^{IAKA}(t) = \varepsilon$, one instance of the CBi-ISIS/Bi-ISIS problem can be solved with a probability of at least $(1/2)\varepsilon$.

5 | PERFORMANCE ANALYSIS

In this section, we discuss the performance efficiency of the proposed LIAMKA protocol in terms of computation, storage, and communication costs. We provide a comparative analysis of the proposed protocol against existing multiple-key agreement protocols from the literature. For simplicity, we choose $n = m \log q$ and $q = m^2$ as the parameters in the proposed scheme, where m is a security parameter.

- *Computation cost*: When we analyze the computational cost, only time-consuming operations during the session key agreement phase are considered. The order of computing the private key for a participant is $2n^2 \cdot |q^2| + n \cdot |q|$, where $|q^2|$ is the cost of multiplying the two numbers in Z_q^*. The message generation of Alice includes the computing overhead:

$$\mathbf{U}_{1i} = \mathbf{M}\mathbf{x}_i \bmod q, \mathbf{V}_{1i} = \mathrm{x}_i{}^{\mathrm{T}}\mathbf{M} \bmod q,$$
$$\mathbf{S}_1 = \mathrm{d}_1 + \sum_{i=1}^{2} H_2(ID_i || \mathbf{T}_1 || \mathbf{U}_{1i} || \mathbf{V}_{1i})\mathbf{x}_i,$$
$$\mathrm{S}_2^{\mathrm{T}}\mathbf{M}? = \mathbf{P}_2 + h_2\mathbf{P} + \sum_{i=1}^{2} H_2\left(ID_i \left\| \mathbf{T}_2 \right\| \mathbf{U}_{2i} \left\| \mathbf{V}_{2i} \right.\right)\mathbf{V}_{2i} \bmod q.$$

Hence, message generation incurs a cost of $3n^2 \cdot |q^2| + 4n \cdot |q|$. Additionally, the generation of any session key requires the computation of shared values $K_{ij}^{02} = \mathbf{x}_i{}^{\mathrm{T}}\mathbf{U}_{2j} \bmod q$ or $K_{ij}^{12} = \mathbf{V}_{2i}\mathbf{x}_j \bmod q$, where $i,j = 1,2$. The order of the one-session key generation is estimated as $n|q^2|$. Hence, the total order of computation of the proposed LIAMKA protocol for one participant is $5n^2|q^2| + 5n|q| + 8n|q^2| = 80m^2\log^4 m + 20m\log^2 m + 64m\log^3 m$ for $n = m\log q$ and $q = m^2$.

- *Communication cost*: To measure the transmitted message size, the identity and time stamp are assumed to be 128 bit (16 B) and 16 bit (2 B) long, respectively. During the session-key agreement phase, the transmission overhead includes two message tuples $\{\mathbf{U}_{11}, \mathbf{V}_{11}, \mathbf{U}_{12}, \mathbf{V}_{12}, \mathbf{S}_1, ID_1, \mathbf{T}_1, \mathbf{P}_1\}$ and $\{\mathbf{U}_{21}, \mathbf{V}_{21}, \mathbf{U}_{22}, \mathbf{V}_{22}, \mathbf{S}_2, ID_2, \mathbf{T}_2, \mathbf{P}_2\}$. Thus, the message requires that $(10n+2)|q| + 244 = (40m\log m + 4)\log m + 244$.
- Storage cost. Each participant must store identity ID_i, key pair $(\mathbf{P}_i, \mathbf{d}_i)$, and system parameter M. Hence, the overhead of storing these values requires $n^2 \cdot |q| + 2n \cdot |q| + 128$. Thus, the total storage cost incurred by both Alice and Bob in the proposed protocol is approximately $2(n^2 \cdot |q| + 2n \cdot |q| + 128) = 16m^2\log^3 m + 16m\log^2 m + 256$.

AMKA protocols in the literature fall into two categories: certificate-based [28, 29, 40, 41] and identity-based [27, 39]. The AMKA protocols in the literature [40, 41] apply a cryptosystem based on discrete logarithms over the general field with a modulus, $q_1 \approx 2^{m'}$, for security parameter m', whereas the AMKA protocols [27–29, 39] depend on an ECC of order n' over GF(q_2). Assume that the proposed lattice-based AMKA protocol has a security parameter of $m = 32$ bits ($n = 2m\log m$). Thus, to achieve the same level of security, the size of m', n', and q_2 are chosen as 1024 bits (i.e., $m' = |q_1| = 1024$ bits, $n' = 160$ bits, and $q_2 = 512$ bits, respectively).

Table 1 shows a comparison between LIAMKA and extant AMKA schemes in terms of computation, storage, and communication costs. Computation costs include the execution of the session key agreement and generation of session keys for one participant. Storage costs include the cost of one party storing a private or public key. Communication costs include the total communication cost of both parties during the session-key agreement phase. Identity-based two-party authenticated key exchange protocols [20, 21] are lattice-based, but they can only produce one session key for each run, and the lattice-based protocol [21] is more effective for both schemes [20, 21]. Next, we compare the proposed LIAMKA protocol with a lattice-based version [21].

The total computation overhead of one participant in Gupta and others [21] is estimated as $96m^2\log^4 m + 12m\log^2 m$. Hence, the proposed LIAMKA protocol is more efficient than the lattice-based protocol [21], and it is easily seen from Table 1 that the storage and communication cost of the existing AMKA schemes are better than the proposed LIAMKA scheme. However, due to the hardness assumptions of CBi-ISIS or Bi-ISIS problems, the proposed protocol has much stronger security than the existing AMKA schemes. Furthermore, LIAMKA's computation cost is more efficient. The lattice-based protocol [21] performs better than LIAMKA in terms of storage and communication costs, but it can produce only one session key for each run.

Comparisons of the proposed LIAMKA scheme with extant AMKA schemes and the lattice-based key exchange protocol [21] in terms of security properties, cryptosystems, and number of session keys are shown in Table 2. AMKA schemes [28, 29, 40, 41] are PKI-based, whereas those in the literature [27, 39] and the proposed scheme are IBC-based AMKA schemes. Compared with identity-based schemes, PKI-based schemes require extra certificate management burdens. Notably, the latter removes the management of public keys. Moreover, the proposed LIAMKA protocol satisfies all AMKA security requirements. The comparison analysis listed in Table 2 confirms that the proposed LIAMKA scheme outperforms the existing AMKA protocols and the lattice-based key agreement protocol [21].

To compare the computational costs, we calculated the number of time-consuming operations during session key agreement in the pairing-based protocols [28, 29, 39] and the proposed LIAMKA scheme. Based on the results in Rahulamathavan and others [42], we chose the parameters of these schemes for 80-bit security. Because the pairing-based protocol [28, 29, 39] generates only four session keys in one run, the time required is the total time of two runs for the generation of eight session keys. These results were obtained using the well-known cryptography library,

TABLE 1 Computational/storage/communication cost comparisons.

	Computation cost	Complexity (in bits)	Storage cost (in bits)	Communication cost (in bits)
Hwang et al. [40]	$5\|q_1^3\| + 2\|q_1\|$	5.37×10^9	$\|q_1\|$	$12\|q_1\|$
Hwang et al. [41]	$6\|q_1^3\| + 2\|q_1\|$	6.44×10^9	$\|q_1\|$	$12\|q_1\|$
Lee et al. [28]	$174\|q_2^2\| + 2\|n'^2\|$	4.57×10^7	$3\|q_2\| + 2\|n'\|$	$8\|q_2\| + 4\|n'\|$
Vo et al. [29]	$5964\|q_2^2\| + 116\|q_2\|$	1.56×10^9	$3\|q_2\| + 2\|n'\|$	$10\|q_2\| + 4\|n'\|$
Pointecheval and Stern [39]	$4486.24\|q_2^2\|$	1.18×10^9	$3\|q_2\|$	$6\|q_2\|$
Dehkordi and Alimoradi [27]	$4428\|q_2^2\|$	1.16×10^9	$3\|q_2\|$	$4\|q_2\|$
Gupta et al. [21]	$6n^2\|q^2\| + 3n\|q\|$	2.56×10^6	$8m^2\log^3 m + 4m\log^2 m$	$(16m\log m + 2)\log m$.
Proposed	$5n^2\|q^2\| + 5n\|q\| + 8n\|q^2\|$	1.46×10^6	$16m^2\log^3 m + 16m\log^2 m + 256$	$(40m\log m + 4)\log m + 244$

TABLE 2 Security and other performance comparisons.

PKI/IBC-based Number of session keys	Hwang et al. [40] PKI 4	Hwang et al. [41] PKI 4	Lee et al. [28] PKI 4	Vo et al. [29] PKI 4	Pointecheval and Stern [39] IBC 4	Dehkordi and Alimoradi [27] IBC 4	Gupta et al. [21] IBC 1	Proposed IBC 8
C1	×	√	×	×	√	×	√	√
C2	√	√	×	×	√	×	—	√
C3	—	—	—	—	×	×	√	√
C4	×	×	×	×	√	×	√	√
C5	√	√	√	√	√	√	√	√

Note: C1, mutual authentication; C2, mutual secrecy; C3, private key generator forward secrecy; C4, perfect forward secrecy; C5, no key control.
Abbreviations: IBC, identity-based cryptography; PKI, public key infrastructure.

MIRACL [43], using a tablet computer running Ubuntu 16.04 with an Intel Core i5-4210U processor (3M Cache, 1.7 GHz) with 4-GB RAM. The duration of the session key agreement in the proposed LIAMKA scheme was 79.35 ms, whereas those of Vo and others [29], Lee and others [28], and Pointecheval and Stern [39] were 154.75 ms, 93.24 ms, and 68.40 ms, respectively. Hence, in addition to resisting quantum attacks, the proposed scheme is also very practical.

6 | CONCLUSION

Based on the hardness assumptions of CBi-ISIS or Bi-ISIS problems, we devised a two-party lattice-based LIAMKA protocol using IBC, which is suitable for quantum computing settings. The LIAMKA protocol can resist various attacks that the existing AKA protocol always suffers from. Furthermore, the LIAMKA protocol has a relatively higher computational efficiency.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest.

ORCID

Hongji Yuan https://orcid.org/0000-0002-5118-0182

REFERENCES

1. W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory **22** (1976), 644–654.
2. S. Blake-Wilson and A. Menezes, *Authenticated DiffieHellman key agreement protocols*, (Proc. 5th Ann. Int. Worksh., SAC'98, Kingston, Ontario, Canada), Aug. 17–18, 1998, pp. 339–361.
3. M. Bellare, D. Pointcheval, and P. Rogaway, *Authenticated key agreement secure against dictionary attacks*, Cryptol. ePrint Arch. (2000), 139–155.

4. A. Shamir, *Identity-based cryptosystems and signature schemes*, In *Advances in cryptology: proceedings of CRYPTO*, Vol. **84**, Springer Berlin Heidelberg, 1984, pp. 47–53.
5. N. P. Smart, *An identity based authenticated key agreement protocol based on the Weil bilinear pairing*, Electron. Lett. **38** (2002), 630–632.
6. L. Harn and H. Y. Lin, *An authenticated key agreement protocol without using oneway function*, (Proc. 8th Inform. Sec. Conf.), 1988, pp.155–160.
7. L. Chen and C. Kudla, *Identity based key agreement protocols from bilinear pairings*, (Proc. 16th IEEE Comput. Sec. Foundat. Worksh.), 2002, pp. 219–213.
8. Z. W. Tan, *Identity-based authenticated multiple key agreement protocol with PKG forward security*, Int. J. Commun. Sys. **28** (2015), no. 3, 534–545.
9. A. Mohammadali, M. S. Haghighi, M. H. Tadayon, and A. Mohammadi-Nodooshan, *A novel identity-based key establishment method for advanced metering infrastructure in smart grid*, IEEE Trans. Smart Grid **9** (2016), no. 4, 2834–2842.
10. D. S. Gupta and G. Biswas, *On securing bi- and tri-partite session key agreement protocol using IBE framework*, Wireless Person. Commun. **96** (2017), no. 3, 4505–4524.
11. L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang, and X. Fu, *Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks*, Int. J. Distrib. Sens. Netw. **14** (2018), no. 4, 1550147718772545.
12. I. Ren and K. Yoneyama, *Adaptive-ID secure hierarchical ID-based authenticated key exchange under standard assumptions without random oracles*, IEICE Trans. Fund. Electron. Commun. Comput. Sci. **105** (2022), no. 9, 1252–1269.
13. Y. M. Tseng, J. L. Chen, and S. S. Huang, *A lightweight leakage-resilient identity-based mutual authentication and key exchange protocol for resource-limited devices*, Comput. Netw. **196** (2021), 108246.
14. L. Deng, J. Shao, and Z. Hu, *Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks*, P2P Netw. Appl. Ther. **3** (2021), 2236–2247.
15. C. Gentry, C. Peikert, and V. Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions* (proc. 40th Ann. ACM Symp. Theory Comput), 2008, pp. 197–206.
16. M. Ajtai, *Generating hard instances of lattice problems* (Proc. 28th Ann ACM Symp. Theory Comput.), 1996, pp. 99–108.
17. S. Wang, Y. Zhu, D. Ma, and R. Feng, *Lattice-based key exchange on small integer solution problem*, Sci. China Inform. Sci. **57** (2014), no. 11, 1–12.
18. D. S. Gupta and G. Biswas, *Cryptanalysis of Wang et al.'s lattice-based key exchange protocol*, Perspec. Sci. **8** (2016), 228–230.
19. D. S. Gupta and G. Biswas, *A novel and efficient lattice-based authenticated key exchange protocol in C-K model*, Int. J. Commun. Sys. **31** (2018), no. 3, e3473.
20. S. H. Islam and S. Zeadally, *Provably secure identity-based two-party authenticated key agreement protocol based on CBi-ISIS and bi-ISIS problems on lattices*, J. Inform. Sec. Applic. **54** (2020), 102540.
21. D. S. Gupta, S. Ray, T. Singh, and M. Kumari, *Post-quantum lightweight identity-based two-party authenticated key exchange protocol for internet of vehicles with probable security*, Comput. Commun. **181** (2022), 69–79.
22. S. Rana and D. Mishra, *Lattice-based key agreement protocol under ring-LWE problem for IoT-enabled smart devices*, Sādhanā **46** (2021), no. 2, 1–11.
23. Z. Tan, *Efficient identity-based authenticated multiple key exchange protocol*, Comput. Elect. Eng. **37** (2011), 191–198.
24. S. M. Yen and M. Joye, *Improved authenticated multiple-key agreement protocol*, Electron. Lett. **34** (1998), no. 18, 1738–1739.
25. T. S. Wu, W. H. He, and C. L. Hsu, *Security of authenticated multiple-key*, Electron. Lett. **35** (1999), no. 5, 391–392.
26. L. Harn and H. Y. Lin, *Authenticated key agreement without using one-way hash function*, Electron. Lett. **37** (2001), no. 10, 629–630.
27. M. H. Dehkordi and R. Alimoradi, *Identity-based multiple key agreement scheme*, KSII Trans. Intern. Inform. Syst. **5** (2011), no. 2, 2392–2402.
28. N. Y. Lee, C. N. Wu, and C. C. Wang, *Authenticated multiple key exchange protocols based on elliptic curves and bilinear pairings*, Comput. Elect. Eng. **34** (2008), no. 1, 12–20.
29. D. L. Vo, H. Lee, C. Y. Yeun, and K. Kim, *Enhancements of authenticated multiple key exchange protocol based on bilinear pairings*, Comput. Elect. Eng. **36** (2010), 155–159.
30. M. S. Farash, M. Bayat, and M. A. Attari, *Vulnerability of two multiple-key agreement protocols*, Comput. Elect. Eng. **37** (2011), 199–204.
31. Q. F. Cheng, *Cryptanalysis of a new efficient authenticated multiple-key exchange protocol from bilinear pairings*, Int. J. Netw. Sec **1** (2014), no. 6, 494–497.
32. M. Farash, M. Attari, R. Atani, and M. Jami, *A new efficient authenticated multiple-key exchange protocol from bilinear pairings*, Comput. Elect. Eng. **39** (2013), no. 2, 530–541.
33. K. W. Kim, E. K. Ryu, and K. Y. Yoo, ID-Based authenticated multiple-key agreement protocol from pairings (Proc. ICCSA), (2004), 672–680.
34. D. S. Gupta and G. Biswas, *Design of lattice-based ElGamal encryption and signature schemes using SIS problem*, Trans. Emerg. Telecommun. Technol. **29** (2018), no. 6, e3255.
35. B. LaMacchia, K. Lauter, and A. Mityagin, Stronger security of authenticated key exchange, (Proc. ProvSec 2007, Wollongong, Australia), Nov. 1–2, 2007, pp. 1–16.
36. T. Zuowen, *An enhanced ID-based authenticated multiple key agreement protocol*, Inform. Technol. Ctrl. **42** (2013), no. 1, 21–28.
37. M. Bellare, and P. Rogaway, *Random oracles are practical a paradigm for designing efficient protocols* (proc. 1st ACM Conf. Comput. Commun. Sec., Fairfox, VA, USA), 1993, pp. 62–73.
38. C. Kudla and K. G. Paterson, *Modular security proofs for key agreement protocol* (Proc. ASIACRYPT), 2005, pp. 549–565.
39. D. Pointecheval, and J. Stern, *Security of proofs for signatures* (Proc. EUROCRYPT'96, Saragossa, Spain), May 12–16, 1996, pp. 387–398.
40. R. J. Hwang, S. H. Shiau, and C. H. Lai, *An enhanced authentication key exchange protocol* (proc. 17th Int. Conf. Adv. Inform. Netw. Applic.), 2003, pp. 20–25.
41. M. S. Hwang, T. Y. Chang, S. C. Lin, and C. S. Tsai, *On the security of an enhanced authentication key exchange protocol* (proc. 18th Int. Conf. Adv. Inform. Netw. Applic.). 2004, pp. 160–163.

42. Y. Rahulamathavan, S. Dogan, X. Shi, R. Lu, M. Rajarajan, and A. Kondoz, *Scalar product lattice computation for efficient privacy-preserving systems*, IEEE IoT J. **8** (2021), no. 3, 1417–1427.
43. MIRACL Ltd., MIRACL cryptographic SDK: multiprecision integer and rational arithmetic cryptographic library, version 7.0.0, 2012. https://github.com/miracl/MIRACL

AUTHOR BIOGRAPHIES

Yang Yang received his master's degree in engineering from Nanchang University in 2019. Since 2006, he has been working at School of Electronics and Information, Nanchang Institute of Technology, Nanchang, Jiangxi Province, P.R. China, successively serving as an assistant, lecturer, and associate professor. His main research interests include electronic information technology and computer simulation.

Yuan Hongji received his doctorate from Harbin Institute of Technology in 1995 and served as an associate professor at Harbin Institute of Technology from 1995 to 1999 and a professor at Nanchang Hangkong University from 2000 to 2005. Since 2005, he has been with the School of Aeronautics and Astronautics, Nanchang Institute of Technology, Nanchang, Jiangxi Province, P.R. China, where he is now a professor. The main research interests include flight control and simulation.

Yan Linbo received her master's degree in engineering from Jiangxi Normal University of Science and Technology in 2016. Since 2005, she has been working at School of Electronics and Information, Nanchang Institute of Technology, Nanchang, Jiangxi Province, P.R. China, successively serving as an assistant, lecturer, and associate professor. The research interests include signal and information processing.

Ruan Yinglan received her bachelor's degree in engineering from Nanchang Institute of Technology in 2016. Since 2016, she has been working at School of Electronics and Information, Nanchang Institute of Technology, Nanchang, Jiangxi Province, P.R. China, where she is a teaching assistant. The main research interest is robotics engineering.

How to cite this article: Y. Yang, H. Yuan, L. Yan, and Y. Ruan, *Post-quantum identity-based authenticated multiple key agreement protocol*, ETRI Journal **45** (2023), 1090–1102. DOI 10.4218/etrij.2022-0320