INAKA: Improved Authenticated Key Agreement Protocol Based on Newhope

YATAO YANG1,2, JIERUN HUANG1, JIANYUAN CHEN1, XINGUANG HAN1,2, YANG ZHAO1, AND ZHANZHEN WEI1

1Beijing Electronic Science and Technology Institute, Beijing 100070, China
2School of Telecommunications Engineering, Xidian University, Xi’an, 710071, China

Corresponding authors: Yatao Yang (yy2008@163.com) and Zhanzhen Wei (13501308617@163.com)

This work was supported by the China State Cryptography Development Fund of Thirteen Five-year under Grant MMJJ20170110.

ABSTRACT The Newhope scheme is one of the milestones of the study in key agreement protocol but it lacks the anti-active-attack capability. In this article, we propose a mutual authenticated key agreement scheme named INAKA scheme based on the commitment value and lattice hard problem. This scheme improves the key encapsulation mechanism in the Newhope scheme to generating the commitment values for both communication parties and thus achieves mutual authentication, key agreement and identity privacy protection at the same time. Firstly, the INAKA protocol is combinable, i.e. the common traditional and lattice-based cryptographic algorithms (encryption, decryption, hash operation) can both act as the protocol components. What’s more, the INAKA protocol has been analyzed that it can resist the man-in-the-middle attack, replay attack, and other attacks. This scheme satisfies provable security under eCK and indistinguishable game models. Its anti-attack capability and security are significantly enhanced compared with the Newhope scheme. Besides, the INAKA protocol involves the identity authentication feature but keeps at the same level of computational complexity. None of the existing schemes (such as Ding’s and BCNS) are able to satisfy the above feature. Lastly, the test results in this article show the INAKA protocol only needs 8.131 milliseconds to complete mutual authentication and key agreement. The outcome of our work could provide lower operation overhead, handy code implementation, and better efficiency to meet the industrial practical requirements.

INDEX TERMS Mutual authentication, authenticated key agreement, lattice, Newhope, key encapsulation mechanism.

I. INTRODUCTION Key agreement (KA) protocol is designed to enable two or more participants to negotiate a common session key on an insecure channel, which allows participants to build a secure communication channel through cryptographic techniques. The shared session key can be used to encrypt and authenticate the information, which plays an important role on ensuring the security of data transmitted. The key encapsulation mechanism (KEM) enables the sender and the receiver to share session keys securely. In a KEM, the initiator encapsulates the session key firstly, and the sender runs an encapsulation algorithm to generate the session key and the cipher text, then, the sender delivers the encapsulated session key to the receiver, finally, the receiver runs the corresponding de-encapsulation algorithm to get the same session key as the sender. Asymmetric encryption methods can be used in most KEMs. During the process of encapsulation and de-encapsulation, it is necessary to ensure message’s confidentiality and security, and to ensure that the session keys obtained by both participants are consistent. However, implementing different key agreement protocols through KEM is passive security and cannot resist man-in-the-middle attack.

Authenticated key agreement (AKA) protocol, not only can negotiate the session key between different participants, but also can authenticate each other between two users. Besides, AKA is able to resist active attacks on wireless or wire channel. In AKA protocol, each communication participant generates a pair of public key and private key separately to carry out identity authentication and key agreement through a KEM.

In recent years, quantum computing technologies have developed rapidly, and the traditional public key cryptosystems are being threatened. Post-quantum cryptography
becomes a very concerned research field because of its resistance to quantum computing attacks [1], [2]. In particular, post-quantum cryptography research has been further promoted by National Security Agency (NSA) and National Institute of Standards and Technology (NIST), which have announced their plans about post-quantum cryptography. However, as so far, there are still rare lattice-based effective AKA protocols with provable security. In order to solve the issue that Newhope protocol cannot resist the active attack, A mutual authenticated key agreement scheme named INAKA is designed, which can be used in the network environment to achieve privacy preservation and mutual authenticated key agreement between communication participants.

Our contributions include, a mutual identity authentication and key agreement protocol is proposed. After message transmission with two rounds, the two participants perform key agreement with identity authentication. Encryption, decryption module and signature for identity authentication are used in the first round. In the second round, SM3 hash operation for authentication is employed. Moreover, the Mask Factor (MF) is proposed and introduced to this protocol to enhance protection for data transmitted.

The structure of this article is as follows: In the first section, the background and development status of the authenticated key agreement protocol are introduced, the basic knowledge on lattice-based cryptography and KEM is introduced in the second and third section. In the fourth section, the specific algorithm and process in INAKA protocol designed are shown. In the fifth section, security proof for this scheme is given. In the sixth section, we introduce the software implementation for the protocol and carry on the performance analysis, the last section is the summary and prospect about our work.

II. RELATED WORK

In 2014, Peikert [3] proposed a KEM based on ideal lattice, which combined the encryption schemes with a reconciliation mechanism by means of RLWE problem, as a result, a KEM based on chosen plaintext attack security (CPA) was constructed. Bos et al. [4] introduced a key exchange protocol RLWE-based (BCNS). In 2017, Ding et al. [5] constructed two lattice-based authenticated key exchange (PAKE) protocols by using a simple and elegant designing idea, which could be regarded as parallel extension of random oracle model (ROM)-based protocols. Alkim et al. [6] proposed a generalizable scheme for BCNS protocol, called Newhope. The main differences between above protocols and Newhope were the generalized coordination mechanism and different error distribution methods. In 2018, Gjøsteen and Jager [7] proposed an AKA protocol based on digital signature which could guarantee forward security in the client-server model, which was simple and easy to implement. In the same year, Bindel et al. [8] described some AKA protocols, such as FSXY, Peikert, ZZDSD, then, some other schemes were described and compared with each other.

About latest achievements about KA’s software and hardware implementation, D. Abbasinezhad-Mood’s research group has done a lot of work. They [9] proposed an anonymous elliptic curve cryptography-based self-certified key distribution scheme, it was free from the overhead of the certificate management and the key escrow issue, communication and computational costs were comparatively lower. The authors implemented the cryptographic elements on two state-of-the-art ARM chips, which was beneficial for the researches in this field. In 2019, Abbasinezhad-Mood et al. [10] talked about the present key establishment schemes’ security weaknesses, indicated that many solutions suffered from known session-specific temporary information attack, private key leakage, key escrow problem. Then, they proposed a key establishment scheme that could be free from these security challenges and key escrow problem, computational and communication costs were also acceptable. The performance analysis had been presented on a NXP LPC1788 ARM chip. In 2019, his team [11] researched the issues on how to securely read the consumption data while putting the least possible overhead on the smart meters. The authors proposed a key establishment protocol, which were both free from the electricity service provider involvement during the key agreement and benefited from notable reduction in the communication cost. Working efficiency and security analyses had been implemented for proposed security protocol, which got better results. Abbasinezhad-Mood et al. [12] also investigated some typical key management protocols and elaborated the existing errata and security threats, proposed a modified version, which was free from the challenges of these solutions, an anonymous ECC-based self-certified two-factor key management scheme was proposed, which could provide the desired security features, formal security verification and proof also supported their scheme.

III. COMPUTATIONAL PROBLEM ON LATTICES

Let R be a ring, \( \mathbb{R}^n \) denotes the set of units in R with vector \( v = (v_1, \ldots, v_n) \in \mathbb{R}^n \). Euclidean length of vector v can be denoted by \( \|v\| = \sqrt{\sum_{i=1}^{n} v_i^2} \). Lattice is regarded as a discrete subgroup in finite dimensional Euclidean vector space. Let \( L \subseteq \mathbb{R}^n \) be a lattice, the minimum distance in lattice \( \lambda \) is defined as the Euclidean length of the shortest non-zero vector of the lattice, which can be expressed by a formula \( \lambda_1 (L) = \min_{v \in L \setminus \{0\}} \|X\| \) as introduced in [13] and [14].

The shortest vector problem (SVP) and the closest vector problem (CVP) are two fundamental computational problems in lattices. These problems have been used for great amount of cryptographic applications.

When solving the problem CVP, to basic lattice \( B \) and target vector \( t \), shortest non-zero vector \( v \in L(B) \) can be searched from \( B \), that is \( \|v\| \leq \lambda_1(L(B)), \|v - t\| \) is called the shortest distance. In the CVP, with approximate factor \( \gamma \), for the case of \( \gamma \geq 1 \), it is needed to find a lattice vector \( v \in L(B) \) to satisfy \( \|v - t\| \leq \gamma \cdot \text{dist}(t, \Lambda) \), where, \( \text{dist}(t, \Lambda) = \inf \{\|v - t\| : v \in \Lambda\} \) denotes the distance from \( t \) to \( \Lambda \).
A. LWE PROBLEMS ON LATTICES
Learning with errors (LWE) problem was promoted by Regev [15], which showed a conclusion on quantum computing: solving a random LWE instance was just like to solve a hard lattice problem with the worst case. LWE problem could be regarded as a generalization of learning and noise parity (LNP), and it was associated with the solving hard problems. In general, when a sequence of approximate random linear equations \( s \) was given, the secret vector \( s \in \mathbb{Z}_q^n \) would be restored to solve LWE problem. The non-quantum reduction for variant \( q \) was demonstrated from the shortest vector problem to the variant of LWE problem [6]. LWE problems were often employed to construct primitives such as indistinguishability chosen plaintext attack (IND-CPA) or indistinguishability chosen ciphertext attack (IND-CCA) secure public key encryption (PKE), identity based encryption (IBE) and full homomorphic encryption (FHE) schemes. These LWE problems were defined as search LWE problem (sLWE) [6], the sLWE problem required to distinguish the uniform distribution between LWE samples and uniform random samples.

B. LEARNING WITH ERRORS IN A RING (RLWE)
In order to solve the problem of low efficiency in cryptosystems based on LWE problem. In 2010, Lyuhashevsky et al. [17] proposed a variant of LWE problem, namely learning with errors in a ring (RLWE). Two common definitions about RLWE problems in cryptography are given as below.

**Definition 1 (Search RLWE Problem):** Let \( R = \mathbb{Z}_q[x]/(x^n + 1) \), \( n = 2^k \), \( k \geq 1 \), \( q = 1 \mod 2n \), \( a \in R \), \( a \) is uniformly randomly selected. \( e \in R \) is an error vector obeying a normal distribution \( \psi_a \). If \( b \in R \) and \( b = a \cdot s + e \) are known, the problem of solving \( s \) by \( a \cdot b \) is the search RLWE problem.

**Definition 2 (Decision RLWE Problem):** Let, \( R = \mathbb{Z}_q[x]/(x^n + 1) \), \( n = 2^k \), \( k \geq 1 \), \( q = 1 \mod 2n \), \( s \in R \), \( s \) is uniformly randomly selected. \( e \in R \) is an error vector obeying a normal distribution \( \psi_a \). When calculating \( b = a \cdot s + e \), \( b \in R \), \( A_{s,\psi} \) is the distribution of \((a, b)\), the problem of distinguishing the uniform distribution between \( A_{s,\psi} \) and \( R \times R \) is the decision RLWE problem. If the decision RLWE problem is difficult, \( A_{s,\psi} \) is pseudo-random.

| Table 1. Variables, parameters, and symbols. |
|---------------------------------------------|
| **Notations** | **Definitions** |
| \( H() \) | Hash function |
| \((pk_a, sk_a)\) | Private and public keys of Alice |
| \((pk_b, sk_b)\) | Private and public keys of Bob |
| \(v_a, r_a\) | Confusion factors of Alice |
| \(s \) | Mask factors of Bob |
| \(ss\) | Shared session key |
| \(M_a\) | Mask value of Alice |
| \(M_b\) | Mask value of Bob |
| \(Ver_a\) | Authentication value of Alice |
| \(Ver_b\) | Authentication value of Bob |
| \(Sig()\) | Signature information |
| \(o\) | Multiplication operator |
| \(\oplus\) | Exclusive-OR operator |

B. KEM ANALYSIS IN NEWHOPE PROTOCOL
Alkim et al. [6] proposed a generalization for the bcns protocol, called Newhope. The main difference between bcns and newhope was the generalized coordination mechanism and the error distributions. In Newhope, the modulus \( q \) could be very small. Instead of a bounded Gaussian distribution, a central binomial distribution \( \Psi_k \) was used in Newhope, its security was not reduced. In addition, depend on [18] and [19], to prevent backdoor attacks and one-to-one cost attacks, pseudo-random public polynomials were generated to run KEM in Newhope. A 256-bit seed and hash function, such as SHAKE-128, were recommended to generate the pseudo-random polynomials.

V. INAKA PROTOCOL DESIGN
INAKA protocol is designed in this article based on Newhope’s KEM with IND-CCA security, now we describe it with some steps as follows.

1) Build a PKE protocol with IND-CPA security on the basis of Newhope’s KEM with IND-CCA security.
2) Use Fujisaki-Okamoto transform (FOT) introduced in [20], [21] to derive the PKE with IND-CCA security.
3) Build KEM with IND-CCA security Newhope-based by using of PKE with IND-CCA security. Assuming polynomial \( a \) is well known by everyone, furthermore, two hash functions are defined: \( H() : \{0, 1\}^* \rightarrow \{0, 1\}^{256} \).

Variables, parameters, and symbols in this scheme can be defined as shown in Table 1 by referring the style in [9].

The flow about INAKA protocol is shown in Fig. 1.

A. STEP1: KEY GENERATION
Both Alice and Bob randomly sample 32 integers with the range of \( 0 \sim 255 \) as the seeds for random number generator. Then hash function will be performed for processing on
the seeds generated by random sampling. The hash function we used is SM3, which is a commercial hash function recommended by China State Cryptography Administration (CSCA). To the hash function $H(l, d)$, where, $l$ is the number of bits of the input data, and $d$ is the number of bits of the output data. During the process of key generation, the seed is processed by SM3 to get an array $z$, whose value is in the range of $0 \sim 255$. The public/private key pairs for Alice and Bob are $(pk_A, sk_A)$ and $(pk_B, sk_B)$, which are generated by function $GenA()$ and $GenB()$.

### B. STEP2: ALICE GENERATES AUTHENTICATION ID $\text{Ver}_A$ AND SENDS IT TO BOB

$s_A$, $e_A$, and $e'_A$ are polynomials on the $\psi_{16}$ domain. The operator $\circ$ represents the multiplication of two polynomial’s coefficients. If $s, e \in R_q$, then,

$$s \circ e = \sum_{i=0}^{n-1} (s_i \cdot e_i \mod q)X^i.$$  

Hash operation will be executed for the identity of Alice $ID_A$ to get the identity mask value (MV) of Alice $M_A$. $a$ is a polynomial derived from the seed, which is used to calculate $b_A = a \circ s_A + e_A$.

Among them, two confusion factors (CF) $v_A$ and $r_A$ are calculated:

$$v_A \leftarrow b_A \circ s_A + e'_A$$

$$r_A \leftarrow \text{HelpRec}(v_A)$$

The confusion factors can resist active attacks in later processes. $b_A, M_A$ and seed are as input data of hash function. Digital signature value $\text{Sig}_{sk_A}$ can be obtained by calculating $b_A, seed, M_A$ and $r_A$ with Alice’s secret key, after that, the authentication value for Alice $\text{Ver}_A$ will be computed by encrypting the $\text{Sig}_{sk_A}$ with Bob’s public key as described:

$$\text{Ver}_A \leftarrow E_{pk_B} \{ \text{Sig}_{sk_A} [H(b_A, M_A, seed), b_A, seed, M_A, r_A] \}$$

Alice sends its authentication value $\text{Ver}_A$ to Bob.

### C. STEP3: BOB VERIFIES ALICE’S IDENTITY

Bob deciphers $\text{Ver}_A$ by using its secret key, the $b_A$, seed, $M_A$, $r_A$ and Hash value $H()$ can be obtained. Bob will recalculate...
If Bob verifies Alice’s identity successfully, the polynomial $s_B$, $e_B$ and $e_B'$ on the domain $\psi_{16}$ will be selected for the following operations. Identity mask value of Bob $M_B$ can be obtained by hash operation for the identity of Bob ID$_B$. The following operations will be executed:

$$b_B \leftarrow a \circ s_B + e_B$$
$$v_B \leftarrow b_A \circ s_B + e_B'$$

Bob performs HelpRec() function for $v_B$ to get the intermediate value $r_B$. To protect $r_B$ and $b_B$, three MFs are introduced: $b$, $r$ and $M$, where, $b$ is the XOR result of $b_B$ and $b_B'$, $r$ is the XOR result of $r_A$ and $r_B$, and $M$ is the XOR result of Bob’s and Alice’s identity mask values $M_A$, $M_B$.

$$b \leftarrow b_A \oplus b_B$$
$$r \leftarrow r_A \oplus r_B$$
$$M \leftarrow M_A \oplus M_B$$

Then, Bob runs hash transformation for $M_B$, $b_B$ and $r_B$ to obtain his authentication value $\text{Ver}_B$:

$$\text{Ver}_B \leftarrow H(M_B, b_B, r_B)$$

Bob sends three MFs ($b$, $r$, $M$) and Bob’s authentication value $\text{Ver}_B$ to Alice for his identity verification.

E. STEPS: ALICE VERIFIES BOB’S IDENTITY

After receiving the message from Bob, Alice carries out XOR on Alice’s identity mask value $M_A$ and $M$ to get Bob’s identity mask value $M_B$, performing similar XOR operations, Alice can get $b_B$ and $r_B$ as following:

$$M_B \leftarrow M \oplus M_A$$
$$b_B \leftarrow b \oplus b_A$$
$$r_B \leftarrow r_A \oplus r$$

Then, Alice runs hash transformation on $M_B$, $b_B$ and $r_B$, then compares the hash result achieved with Bob’s authentication value $\text{Ver}_B$. If they are consistent, the verification is success. Otherwise, the authentication fails and the process of authenticated key agreement is stopped.

F. STEP6: KEY AGREEMENT BETWEEN ALICE AND BOB

If Alice verifies Bob’s identity successfully, the next steps will be proceed. Alice will calculate $v_A \leftarrow b_B \circ s_A$, then, Alice and Bob perform Rec() function respectively to get the key $k$ and $k'$.

The final shared session key $ss$ can be obtained by performing Hash transformation respectively on $M_A$, $M_B$, and $k$, $k'$. The shared session key generated by two participants is same. That is to say, the authentication and key agreement process is success.

VI. SECURITY ANALYSIS

A. ANALYSIS TO RESIST DIFFERENT ATTACKS

The correctness of INAKA protocol has been reflected on the above protocol flow. Common security attributes for AKA protocol has been introduced in [22]–[24]. We will analyze pivotal security attributes about this protocol in the following parts.

1) PRIVATE KEY RECOVERY ATTACK

The protocol proposed in this article is based on the RLWE hard problem, that is, the security of our protocol can be reduced to the RLWE problem on the lattice. The public key, private key, and authentication information in this protocol are constructed according to the requirements of the RLWE problem. At present, there is no effective cryptographic algorithm that can solve the RLWE hard problem, so, it is impossible for the adversary to recover the private key by intercepting intermediate parameters in this protocol. In other words, if the adversary might crack this protocol by recovering the private key, the adversary also could crack the RLWE problem on the lattice.

2) MAN-IN-THE-MIDDLE ATTACK

If the adversary attacks the protocol by using the man-in-the-middle, the $b$ and the seed that Bob sent to Alice might be actively attacked by the adversary during the data transmission without identity verification between Alice and Bob. Therefore, data integrity could not be guaranteed. $v_B$ or $r_B$ also could not be accurately calculated. Finally the key agreement would be failed. In addition, the MFs $b$, $r$ and $M$ were introduced to data transmission process, even if the adversary acquired the transmitted data, he could not get any useful information about Alice or Bob, because the transmitted data were masked by XOR operation, and could not be directly used for calculation to determine the identity of Alice or Bob. Therefore, in INAKA scheme, the mutual identity authentication for two communication participants and the mask factors prevent the adversary from impersonating any party to destroy the key agreement process. Therefore, INAKA protocol can resist the man-in-the-middle attack.

3) REPLY ATTACK

The replay attack can be regarded as the behavior of an attacker using the public session information that has been intercepted to obtain the secret information in new session. In this protocol, Alice randomly samples 32 integer elements from 0 to 255 as the seed in session, polynomials $s_A$ and $e_A$ on the domain $\psi_{16}$ are valid only in current session, only if the mutual authentication between Alice and Bob is success.
the shared key $ss$ in current session would be generated. This scheme can resist replay attack.

4) WEAKLY PERFECT FORWARD SECURITY
During the key agreement process, if a new node joins, the session key generated would be also inconsistent because the random polynomials in key agreement process are different. As a result, the newly joined node could not know the previous session key, therefore this protocol has weakly perfect forward security.

5) UNKNOWN KEY-SHARED ATTACK
Assuming that there was an adversary Eve in the session channel, he might illegally use Bob’s long-term public key as his long-term public key. Eve could implement an unknown key sharing attack by following manner.

Suppose Alice would initiate session 1 with Bob. When Alice sent authentication message to Bob, Eve might intercept the message in channel and initiate another session 2 to Bob, and also sent the same authentication message to Bob. Bob performed the verification calculation after receiving the message from Eve, then sent the verification result to Eve, Eve forwarded the verification message to Alice. “Secure communication” could be established in session 1, the shared session key could be obtained by both Alice and Eve. In session 2, Eve intercepted the returned value and forwarded it to Bob, “secure communication” could also be established between Eve and Bob, and Eve could also get the shared session key. That is, the adversary Eve can intercept the encrypted message from Alice in session 1 and forward it to Bob, since Bob has a shared key with Eve, the message could be correctly decrypted by Bob. As a result, the protocol could be attacked by an unknown shared key.

However, the hash value and digital signature about identity information were added to the calculation process of session key in this scheme, so that the unknown key-shared attack during the communication process was avoided.

B. SECURITY PROOF
The security of this protocol could be proved under the Canetti and Krawczyk (CK) model introduced in [22], [23]. However, one protocol that had been proven security under CK model still could not be guaranteed to resist key compromise impersonation (KCI) attacks and leakage of ephemeral private key (LEP) attacks, moreover, it could not ensure weak perfect forward security of the protocol (wPFS). In order to make up for these shortcomings, the extended CK model (eCK) introduced in [25] was used to prove the security of the protocol. In general, BR, BPR, CK, and eCK are often used as the provable security models for current AKA protocols, among them, the eCK model covers the strongest attack types and attack methods, which has been widely recognized by many researchers. Therefore, the eCK model also is used to prove the security of the INAKA protocol. In this model, a probabilistic adversary Eve could control the communication. Eve could obtain the secret information stored in the participants’ memory through an explicit attack. Therefore, the security of the key exchange protocol needed to guarantee the leakage probability of secret values to minimize the impact on the security of other secret items. The adversary Eve could interact with a protocol participant $O$, the security of INAKA could be proved by using the following queries [9]:

1) Execute (Alice, Bob): This query is used to simulate a passive attack and return exchange messages that belong to the protocol participants during actual execution.
2) $h_n(a)$: Eve receives a random number as the Hash value of $a$ by this query.
3) Send($O, m$): This query is used to simulate an active attack, Eve can send $m$ to $O$ and receive response according to the protocol description through this query.
4) Long Term Key Reveal($O$): This query lets Eve to inform the long term secret key held by the oracle $O$.
5) Ephermal Key Reveal($O$): This query lets Eve to inform the ephemeral secret held by the oracle $O$.
6) Session Key Reveal($O$): This query lets Eve to inform the session key held by the oracle $O$.
7) Establish Party A: This query lets Eve to achieve the public key of Alice from CA.
8) Corrupt($O$): This query lets Eve to inform the long-term secret held by the oracle $O$.
9) Expire($O$): This query deletes the session key for the full session held by the oracle $O$.
10) Test($O$): This query is used to measure the semantic security of the session key.

Definition 3 (Security of eCK Model): Let $ss^*$ represent the session key constructed in the session $sid^*$, $sid^*$ is a fresh session. Eve guesses the value of $c^*$, $c^* \in \{0, 1\}$, the value $c$ will be output. When responding to the adversary Eve, if $c = 1$, $sid^*$ returns the value of the real session key; Otherwise, $sid^*$ returns a random number with the same length to Eve [25]. The advantage of adversary Eve is:

$$Adv^{RLWE}(Eve) = |2 \text{Pr}[c^* = c] - 1|$$

The AKA protocol is secure if and only if:

1) Two honest protocol participants complete a matched session, both sides calculate the same session key, or both sides output an failed execution identifier.
2) Within arbitrary probability polynomial time, Eve’s advantage $Adv^{RLWE}(Eve)$ is negligible.

Supposing the advantage probability of the adversary Eve to solve the RLWE hard problem is $Adv^{RLWE}(Eve)$, if the advantage of attacking one protocol by the adversary Eve is negligible, that is $Adv^{RLWE}(Eve) \leq \varepsilon$, the scheme is secure.

Proof: In order to prove the semantic security of the protocol, game sequence $GM_0$ to $GM_3$ are defined. $GM_0$ is indicative of the real attack, and $GM_3$ is the game in which Eve has no advantage. Let $S_i$ is the corresponding event to the $GM_i$.

Game $GM_0$: The simulation of this game is identical to the real attack in the random oracle model. Hence, we have:

$$Adv^{RLWE}(Eve) = |2 \text{Pr}[S_0 (Eve)] - 1|.$$
Table 2: Simulation of oracles.

| Scenario | Oracle Action | Verification Equation | Probability |
|----------|---------------|-----------------------|-------------|
| Alice sends (query) | $\sigma = \text{execute}_i$ | $\text{Ver}_e = E_{\sigma} \{ \text{Sig}_j \{ H(b_i, M, \text{seed}), b_i, \text{seed}, M, r_i \} \}$ | $\text{Pr}[\text{Ver}_e] = \frac{1}{2}$ |
| Bob sends (query) | $\sigma = \text{execute}_j$ | $\text{Ver}_e = D_{\text{sk}_e}(\text{Ver}_i)$ | $\text{Pr}[\text{Ver}_e] = \frac{1}{2}$ |
| Session Key Reveal | $\sigma = \text{execute}_i$ | $\text{Ver}_e = H(M, b_i, r_i)$ | $\text{Pr}[\text{Ver}_e] = \frac{1}{2}$ |
| ephemeral key reveal | $\sigma = \text{execute}_i$ | $\text{Ver}_e = \text{sk}_e$ | $\text{Pr}[\text{Ver}_e] = \frac{1}{2}$ |

Game GM1: In this game, the oracles are simulated (see Table 2). Since the oracles Execute, Send, and others are simulated as done in the real attack just as the real execution of the protocol. Therefore, we can conclude:

$$\text{Adv}_{\text{RLWE}}^{\text{GM1}}(\text{Eve}) = |2 \text{Pr}[S_1] - \text{Pr}[S_2]| - 1.$$  

Game GM2: The simulation of this game is the same as GM1, except that if there is a conflict in the script and the hash query, the game will be terminated. According to the birthday paradox, the probability of hash collision is at most $q_h^2/2^{l+1}$. Since $N_A, e_A, e'_A$ are selected randomly from the domain $\mathbb{G}_m$, the probability of collision in the script is at most $(q_h + q_e)^2/2^l$ [26, 27]. As a result, we conclude:

$$|\text{Pr}[S_2] - \text{Pr}[S_1]| \leq q_h^2/2^{l+1} + (q_h + q_e)^2/2^l,$$

$$\text{Adv}_{\text{RLWE}}^{\text{GM2}}(\text{Eve}) < \epsilon.$$  

Game GM3: In this game, the simulation of this game is consistent with GM2, unless Eve is lucky to guess the value of the verifier without asking $h$ oracle and the game is suspended. Therefore, we have:

$$|\text{Pr}[S_3] - \text{Pr}[S_2]| \leq q_h^2/2^{l+1},$$

$$\text{Adv}_{\text{RLWE}}^{\text{GM3}}(\text{Eve}) < \epsilon.$$  

Game GM4: In this game, the session key security is considered that Eve cannot achieve the session key unless one of $(s_A, e_A, e'_A)$ or $(s_B, e_B, e'_B)$ are revealed to him. The goal of Eve is to compute the session key in the following four cases by making Execute and $h$ queries.

Case 0: Corrupt(Alice) and Corrupt(Bob). In this case, Eve achieves the keys of both Alice and Bob, i.e., the skA, skB, but not their MVs, MA or MB.

Case 1: Corrupt(Alice) and Ephemeral Key Reveal(Bob). In this case, Eve only achieves the private key of Alice.

Case 2: Long Term Key Reveal(Alice). In this case, Eve can only achieve the long term key private key.

Case 3: Ephemeral Key Reveal(Alice) and Corrupt(Bob). In this case, Eve only achieves the private key of Bob.

Case 4: Ephemeral Key Reveal(Alice) and Ephemeral Key Reveal(Bob). In this case, Eve cannot obtain their private keys.

In all of the aforementioned four cases, Eve cannot compute the $k$ and $k'$. Therefore, the difference between this game and the previous one is negligible. It is concluded that:

$$|\text{Pr}[S_4] - \text{Pr}[S_3]| \leq q_h^2/2^{l+1}.$$  

Game GM5: The simulation of this game is the same as the game GM4 except that this game will be ended if Eve issues $h$ query. Since Eve can achieve the $k$ with probability of at most $q_h^2/2^{l+1}$, we have $|\text{Pr}[S_5] - \text{Pr}[S_4]| \leq q_h^2/2^{l+1}$. Because Eve has no advantage in distinguishing the real session key from a random one without making the $h$ query with the correct input, we have:

$$\text{Pr}[S_5] = 1/2, \quad \text{Adv}_{\text{RLWE}}^{\text{GM5}}(\text{Eve}) < \epsilon.$$  

Combining all the probabilities, it is concluded that INAKA scheme is provable security.

VII. PERFORMANCE ANALYSIS FOR INAKA

A. THEORETICAL ANALYSIS

In order to illustrate the advantages of this proposed protocol, we compare the working performance and efficiency between typical KA/AKA schemes, such as Ding scheme [28], BCNS scheme [4], Newhope scheme [6], BCD+ scheme [29], Peiker scheme [3], FSXY scheme [16], ZZDSD scheme [30] and BDK+ scheme [31]. In order to highlight the comprehensive performance of the protocols, we analyze these schemes on the computational complexity, forward security, anti-man-in-the-middle attacks, anti-replay attack, and so on. The compared results are shown in Table 3.

B. PERFORMANCE TEST

The protocol was tested on a computer configured with an Intel Core i7 processor, 8 GB of memory, and Windows 7 64-bit operating system. The software working platform is Microsoft Visual Studio 2010 Professional Edition. In the
TABLE 3. Comparison between different KA/AKA protocols.

| Type | Scheme | Modulo q | Security model | Computational complexity | Anti-man-in-the-middle attack | Anti-replay attack | Forward security |
|------|--------|----------|----------------|--------------------------|-----------------------------|-----------------|-----------------|
| KA   | [28]   | $n^4$    | RO             | $O(2^{3n-2})$            | N                           | N               | N               |
|      | [4]    | $O(n^l)$ | SK             | $O(2^{x+l})$             | N                           | Y               | Y               |
|      | [6]    | $O(n^2)$ | RO             | $O(2^{x+1})$             | Unknown                     | Y               | N               |
|      | [29]   | $O(n^2)$ | SK             | $O(2^{x+1})$             | N                           | Y               | Y               |
| AKA  | [3]    | $O(n^2)$ | SK             | $O(2^{x+1})$             | Y                           | Y               | Y               |
|      | [16]   | $O(n^2)$ | CK’            | $O(2^{x+1})$             | Unknown                     | Y               | N               |
|      | [30]   | $O(n^2)$ | BR             | $O(2^{x+2})$             | Y                           | Y               | N               |
|      | [31]   | $O(n^2)$ | RO             | $O(2^{x+1})$             | Y                           | Y               | N               |
| AKA  | INAKA  | $O(n^2)$ | eCK            | $O(2^{x+1})$             | Y                           | Y               | Y               |

TABLE 4. Efficiency test for INAKA.

| Protocol | Alice generates $r_{vA}$ | Bob verifies Alice | Bob generates $r_{vB}$ | Alice verifies Bob | Alice generates $s$ | Total time |
|----------|--------------------------|-------------------|------------------------|-------------------|--------------------|------------|
| INAKA    | 0.916ms                  | 6.864ms           | 0.172ms                | 0.028ms           | 0.151ms            | 8.131ms    |

TABLE 5. Length of key parameters in INAKA protocol.

| Protocol | Seed | $pk$ | $sk$ | $Ver_A$ | $ct$ | $ss$ | $Ver_B$ |
|----------|------|------|------|---------|------|------|---------|
| INAKA    | 384Bit | 14.25KB | 28.75KB | 33KB | 17.25KB | 256Bit | 29KB |

actual test, as an example, RSA1024 algorithm is used as encryption/decryption module in INAKA protocol.

The running time in Table 4 is the mean value of 50 actual test results, where,

1. The average time required for Alice to generate $(pk_A, sk_A)$ and identity authentication information $Ver_A$ is 0.916 ms while the longest time is 1.636 ms, and the shortest time is 0.884 ms.

2. The average time required for Bob to verify Alice’s identity information is 6.864 ms while the longest time is 7.236 ms, and the shortest time is 6.730 ms.

3. The average time required for Bob to derive the key to generate identity authentication information $Ver_B$ is 0.172 ms while the longest time is 0.276 ms, and the shortest time is 0.167 ms.

4. The average time required for Alice to verify Bob’s identity information is 0.028 ms while the longest time is 0.087 ms, and the shortest time is 0.027 ms.

5. The average time required for Alice to derive the key is 0.151 ms while the longest time is 0.222 ms, and the shortest time is 0.148 ms.

The average time required to generate the shared key in a complete session is 8.131 ms while the longest time is 9.457 ms, and the shortest time is 7.956 ms.

After actual testing, following results are shown in the Table 5. The length of the seed randomly generated is 384 bits, the length of the public key $pk$ is 14.25 KB, and the private key $sk$ is 28.75 KB, the length of the identity authentication value $Ver_A$ that Alice sends to Bob in the first round is 33 KB. In the second round, the length of Bob’s identity authentication value $Ver_B$ is 33 KB, the length of intermediate vector $ct$ is 17.25 KB, and the session key $ss$ is 256 bits.

Fig.2 is line charts of 50 times of efficiency test. Fig.3 is respective time consumption to generate the session key for Alice and Bob.

It can be seen from Fig.2 and Fig.3 that it only costs 8.131 milliseconds to complete the mutual authentication and key agreement through 50 times of actual tests without considering transmission time and delay on the communication link.

Several typical protocols’ implementation performance parameters on software platform are presented in Table 6. It can be seen from Table 6 that, all the schemes, including [9], [11], [12] and INAKA all have better security, which can resist man-in-the-middle attack, anti-replay attack or others, and can satisfy forward security, mutual authentication. The time consumption on private key/signature and public key/verification in INAKA are higher than other schemes owing to different working platforms and configurations. It is worth mentioning that, in this protocol, encryption, decryption and hash operation can be replaced by other traditional or lattice-based cryptographic modules, that is...
TABLE 6. Comparison of the related schemes.

| Scheme | F1(m)| F2(m) | F3 | F4 | F5 | F6 | F7 |
|--------|------|------|----|----|----|----|----|
| [9]    | 0.135| 0.01 | Yes| Yes| Yes| Yes| Intel Core (i7-4702MQ processor, 16-GB memory, 256-GB SSD hard drive, 84-bit Ubantu 16.04, and the NanoPi M3 board). |
| [11]   | 0.85 | 0.1925 | Yes| Yes| Yes| Yes| E2200 2.20 GHz Intel Pentium CPU, 2 GB of RAM, and a 32-bit Ubuntu 12.04.1 LTS as the operating system. |
| [12]   | 0.155| 0.04 | Yes| Yes| Yes| Yes| Intel Core (i7 processor, 8 GB memory, SATA hard drive, Windows 7 64-bit operating system. |
| INAKA  | 0.615| 1.894| Yes| Yes| Yes| Yes| Intel Core (i7 processor, 8 GB memory, SATA hard drive, Windows 7 64-bit operating system. |

F1: private/signed; F2: public/verification; F3: Anti-man-in-the-middle attack; F4: Anti-replay attack; F5: Forward security; F6: Mutual authentication; F7: Working platform.

![FIGURE 3. Time consumption comparison between Alice and Bob.](image)

TABLE 6. Comparison of the related schemes.

In this article, we propose a common lattice-based combinable authentication and key agreement protocol named INAKA. It combines post-quantum cryptography, identity authentication, commitment value, identity mask values, and mask factors to protect the user’s identity privacy. This protocol can achieve mutual identity authentication and key agreement in only two rounds of communication transmission. In the round I, the user receives the sender’s authentication message by operating hash, encryption and digital signature. In round II, the user carries out the decryption, digital signature verification, and MFs’ XOR computation. This kind of two-rounds message transmission designing reduces the computational burden and improves the AKA protocol’s efficiency. The test result indicates that the INAKA protocol only needs 8.131 milliseconds to complete the mutual authentication and key agreement, without considering transmission time and delay in the communication link. It also has been proved that the INAKA scheme meets the security under the eCK and indistinguishable game model. This solution can achieve privacy preservation and mutual authenticated key agreement between communication parties, thus we believe our work can benefit the authenticated network application.

REFERENCES

[1] V. Lyubashevsky, “Digital signatures based on the hardness of ideal lattice problems in all rings,” in Proc. Int. Conf. Theory Appl. Crypt. Inf. Secur. (ASIACRYPT), in Lecture Notes in Computer Science, vol. 10032. Hanoi, Vietnam: Springer, Nov. 2016, pp. 196–214.

[2] D. Micciancio and M. Walter, “Gaussian sampling over the integers: Efficient, generic, constant-time,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 10402. San Francisco, CA, USA: Springer, Jul. 2017, pp. 455–485.

[3] C. Peikert, “Lattice cryptography for the Internet,” in Proc. Int. Workshop Post-Quantum Cryptogr. (POQCrypt), in Lecture Notes in Computer Science, vol. 8772. Waterloo, ON, Canada: Springer, Oct. 2014, pp. 197–219.

[4] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, “Post-quantum key exchange for the TLS protocol from the ring learning with errors problem,” in Proc. IEEE Symp. Secur. Privacy, San Jose, CA, USA, May 2015, pp. 553–570.

[5] 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. Cryptogr. Track RSA Conf. (CT-RSA), in Lecture Notes in Computer Science, vol. 10159. San Francisco, CA, USA: Springer, Jan. 2017, pp. 183–204.

[6] E. Alkim, L. Ducas, T. Poppelmann, and P. Schwabe, “Post-quantum key exchange—A new hope,” in Proc. 25th USENIX Secur. Symp. (USENIX Secur.), Austin, TX, USA, Aug. 2016, pp. 327–343.

[7] K. Gjøsteen and T. Jager, “Practical and tightly-secure digital signatures and authenticated key exchange,” in Advances in Cryptology, (Lecture Notes in Computer Science), vol. 10992. Santa Barbara, CA, USA: Springer, Jul. 2018, pp. 95–125.

[8] N. Bindel, J. Buchmann, and S. Rieß, “Comparing apples with apples: Performance analysis of lattice-based authenticated key exchange protocols,” Int. J. Inf. Secur., vol. 17, no. 6, pp. 701–718, 2018.

[9] D. Abbasinezhad-Mood and M. Nikooghadam, “An anonymous ECC-based self-certified key distribution scheme for the smart grid,” IEEE Trans. Ind. Electron., vol. 65, no. 10, pp. 7996–8004, Oct. 2018.

[10] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, and S. M. Mazinani, “A secure and efficient key establishment scheme for communications of smart meters and service providers in smart grid,” IEEE Trans Ind. Informat., vol. 16, no. 3, pp. 1495–1502, Mar. 2020.

[11] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, “Novel anonymous key establishment protocol for isolated smart meters,” IEEE Trans. Ind. Electron., vol. 67, no. 4, pp. 2844–2851, Apr. 2020.

[12] D. Abbasinezhad-Mood and M. Nikooghadam, “Efficient design of a novel ECC-based public key scheme for medical data protection by utilization of NanoPi fire,” IEICE Trans. Rel., vol. 67, no. 3, pp. 1328–1339, Sep. 2018.

[13] R. Kannan, “Improved algorithms for integer programming and related lattice problems,” in Proc. 15th Ann. ACM Symp. Theory Comput. (STOC), Boston, MA, USA, Dec. 1983, pp. 193–206.

[14] O. Regev, “Lattice-based cryptography,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 4117. Santa Barbara, CA, USA: Springer, 2006, pp. 131–141.

[15] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in Proc. 37th ACM Symp. Theory Comput. (STOC), 2005, pp. 84–93.

[16] A. Fujioka, K. Suzuki, K. Kaga, and K. Yoneyama, “Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism,” in Proc. ASIACCS, in Lecture Notes in Computer Science, vol. 6110. Hangzhou, China: ACM, 2010, pp. 83–94.

[17] V. Lyubashevsky, C. Peikert, and O. Regev, “On lattices and learning with errors over rings,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 6110. Hanoi, Vietnam: Springer, Jun. 2010, pp. 1–23.

[18] J. Alawatugoda, C. Boyd, and D. Stebila, “Continuous after-the-fact leakage-resilient key exchange,” in Proc. 19th Australas. Conf. Inf. Secur. Privacy (ACISP), in Lecture Notes in Computer Science, vol. 8544. Wollongong, NSW, Australia: Springer, Jul. 2014, pp. 258–273.
Y. Yang et al.: INAKA: Improved AKA Protocol Based on Newhope

[19] M. Bellare and P. Rogaway, “Entity authentication and key distribution,” in Proc. 13th Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO), in Lecture Notes in Computer Science, vol. 773. Santa Barbara, CA, USA: Springer, 1994, pp. 232–249.
[20] O. Regev, “The learning with errors problem (invited survey),” in Proc. IEEE 25th Annu. Conf. Comput. Complex., Cambridge, MA, USA, Jun. 2010, pp. 191–204.
[21] E. E. Targhi and D. Unruh, “Post-quantum security of the Fujisaki-Okamoto and OAEP transforms,” in Proc. Theory Cryptogr. Conf. (TCC), in Lecture Notes in Computer Science, vol. 9986. Beijing, China: Springer, 2016, pp. 192–216.
[22] R. Canetti and H. Krawczyk, “Analysis of key-exchange protocols and their use for building secure channels,” in Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT), in Lecture Notes in Computer Science, vol. 2045. Innsbruck, Austria: Springer, Apr. 2001, pp. 453–474.
[23] R. Canetti and H. Krawczyk, “Security analysis of IKE’s signature-based key-exchange protocol,” in Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol. (CRYPTO), in Lecture Notes in Computer Science, vol. 2442. Santa Barbara, CA, USA: Springer, Jul. 2002, pp. 143–161.
[24] H. Krawczyk, “HMQV: A high performance secure Diffie–Hellman protocol,” in Advances in Cryptology, Lecture Notes in Computer Science, vol. 3621. Santa Barbara, CA, USA: Springer, 2005, pp. 546–566.
[25] B. Lamacchia, K. Lauter, and A. Mityagin, “Stronger security of authenticated key exchange,” in Proc. Int. Conf. Provably Secur. (ProvSec), in Lecture Notes in Computer Science, vol. 4784. Wollongong, NSW, Australia: Springer, 2007, pp. 1–16.
[26] S. H. Islam, “Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps,” Nonlinear Dyn., vol. 78, no. 3, pp. 2261–2276, Jul. 2014.
[27] J. T. Ding, “A simple provably secure key exchange scheme based on the learning with errors problem,” Int. Assoc. Cryptologic Res., IACR Cryptol. ePrint Arch., New York, NY, USA, Tech. Rep. 2012(688), Dec. 2012, pp. 1–13.
[28] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, “Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), 2016, pp. 1006–1018.
[29] J. Zhang, Z. Zhang, J. Ding, M. Snooek, and O. Dagdelen, “Authenticated key exchange from ideal lattices,” in Proc. 34th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT), in Lecture Notes in Computer Science, vol. 9057. Sofia, Bulgaria: Springer, 2015, pp. 719–751.
[30] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, “CRYSTALS–Kyber: A CCA-secure module-lattice-based KEM,” in Proc. IEEE Eur. Symp. Secure Privacy (EuroS&P), Apr. 2018, pp. 353–367.

YATAO YANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2009. He is currently an Associate Professor with the Beijing Electronic Science and Technology Institute. His main research interests include information security, cryptographic protocol, and algorithm.

JIERUN HUANG was born in 1995. She is currently pursuing the master’s degree with the Beijing Electronic Science and Technology Institute. Her main research interests include information security, cryptographic protocol, and algorithm.

JIANYUAN CHEN was born in 1997. He is currently pursuing the master’s degree with the Beijing Electronic Science and Technology Institute. His main research interests include information security, cryptographic protocol, and algorithm.

JINGYANG HAN was born in 1994. He is currently pursuing the master’s degree with the Beijing Electronic Science and Technology Institute. His main research interests include information security, cryptographic protocol, and algorithm.

YANG ZHAO was born in 1995. He is currently pursuing the master’s degree with the Beijing Electronic Science and Technology Institute. His main research interests include information security, cryptographic protocol, and algorithm.

ZHANZHEN WEI was born in 1971. He is currently a Professor with the Beijing Electronic Science and Technology Institute. His main research interests include information security and cryptography.

* * *