Efficient Two-Party Authentication Key Agreement Protocol Using Reconciliation Mechanism from Lattice

Jinhua Wang(✉), Ting Chen, Yanyan Liu, Yu Zhou, and XinFeng Dong

Science and Technology on Communication Security Laboratory, Chengdu 610041, Sichuan, China
wjhcetc@163.com

Abstract. It is crucial and challenging to design a quantum-secure and efficient authentication key agreement scheme for IoT. The reasons are that not only there are various security requirements need to meet, but also communication party is resource-constrained. Recently, a large number of 2PAKA schemes for IoT have been presented, yet most of them are subject to quantum attack. In this paper, we put forward a quantum-secure 2PAKA protocol using lattice cryptography. The proposed LB-ID-2PAKA protocol makes use of identity-based signature to avoid the complicated certificate management of PKI-based protocol. At the same time, based on the Kyber.KE, we apply Peikert's reconciliation mechanism to save the communication cost. Our LB-ID-2PAKA protocol can be resistance against various attack and provide desired security property, especially support perfect forward secrecy. Moreover, the provable security analysis shows that our LB-ID-2PAKA protocol is provably secure under RO model and the hardness assumption of MLWE.

Keywords: Post-quantum · Key Agreement · Identity-based Signature · MLWE · Peikert's reconciliation mechanism

1 Introduction

Recently, with the breakthrough of 5G, the Internet of Things (IoT) has got rapid development. At the same time, due to the impact of the epidemic, the demand for telecommuting has increased, followed by a large number of devices connected to the Internet. It is estimated that 24.6 billion IoT devices will be connected globally by 2025 [1]. IoT has become the most developed way to share information among many resource-constrained devices that are connected to each other via the internet. While the IoT brings convenience, it also gives adversaries more options to attack. So, it is urgent to design a protocol for secure communication.

However, it is challenging to design an efficient authentication key agreement scheme for IoT. The reasons are that not only there are various security requirements need to meet, but also communication party is resource-constrained. With the advent of the DH

Q. Jiang et al. (Eds.): SPNCE 2022, LNICST 496, pp. 32–47, 2023.
https://doi.org/10.1007/978-3-031-30623-5_3

exchange in 1976 [2], two party authentication key agreement (2PAKA) protocols have got rapid development over the past few decades [3–7]. The existing 2PAKA protocols have been put forward based on the traditional cryptographic primitives (public-key cryptography [8], identity-based cryptography [9]), elliptic curve cryptography [10], bilinear mapping [11], Chinese Remainder Theorem [12] et.) and the hardness assumption of computational Diffie-Hellman (CDH), or integer factorization problem (IFP) or discrete logarithm problem (DLP).

However, Shor [13] pointed that there exists an adversary under the quantum computing environments can solve abovementioned computational hardness problem easily using polynomial-time algorithms. Thus, there is an urgent need to put forward a quantum-secure 2PAKA protocol for IoT. Due to the advantages of simple operations and less computational overheads, lattice-based cryptography attracts extensive attention from home and abroad.

With the proliferation of lattice-based cryptography, there are two designing ways of AKA protocol under quantum computing environments: one is using reconciliation mechanism and the other is using key encapsulation mechanism (KEM). Peikert [14] pointed that the learning with errors (LWE)-based KA protocol was technically feasible, but it did not design a protocol. After, Lindner et al. [15] presented the construction of DH-like key exchange based on LWE. In literature [16, 17], the author tries to extract the same secret from two approximations, so that the communication parties can obtain the same session key through calculation. Thus, based on LWE and its variants 2PAKA protocols are mostly designed through error reconciliation mechanism.

With its simplicity and modularity, 2PAKA protocol is designed using KEM becomes one of the research hotspots. However, using KEM will cause more communication overheads, it cannot applies to IoT directly. Moreover, most of them are based on public key infrastructure, which will lead to the complicated certificate management. To fill up the loophole, identity-based cryptography (IBC) is introduced to eliminate certificate management. Yet, most of existing identity-based 2PAKA protocols fail to provide quantum-secure. Therefore, it is crucial and challenging to devise a quantum-secure identity-based 2PAKA protocol for IoT.

1.1 Contributions

Although researchers have made considerable efforts to design an efficient and provably secure 2PAKA for IoT, most of them are subject to quantum attack. In this paper, we put forward a quantum-secure 2PAKA protocol using lattice cryptography(LB-ID-2PAKA) for IoT. The main contributions of LB-ID-2PAKA protocol are concluded as the following three dots:

- Combining IBC and lattice cryptography designs a LB-ID-2PAKA protocol. The protocol uses IBC to eliminate the cost of certificate management and uses lattice hard assumptions to confirm the quantum-secure.
- Based on the Kyber.KE, we apply Peikert's reconciliation mechanism to save the communication cost. Moreover, the probability of agreement failure is smaller.
- This paper provides the provable security of LB-ID-2PAKA protocol. The analysis result shows that the proposal is provably secure under RO model. Furthermore, the

LB-ID-2PAKA protocol can withstand various attacks and provide desired security property, especially support perfect forward secrecy.

1.2 Related Work

The communication between two parties in IoT could be performed to negotiate a session key. Since Diffie and Hellman [1] devised the first key agreement scheme, a large of 2PAKA protocols [6, 7] were proposed to provide stronger security and better performance. Most of these schemes are based on traditional PKI which need a certification authority to manage public-key certificates that binds the user to his public key. However, the storage, storage, and transmission of certificates require significant overheads. To remove the cost of certificate management, many identity-based 2PAKA (ID-2PAKA) were proposed. Using IBC, ECC and bilinear mapping, Gupta et al. [18] put forward a secure ID-2PAKA protocol for IIoT and claimed their scheme was efficient than other state-of-the-art competing protocols. Dang et al.[19] designed an ID-2PAKA and claimed their scheme was provably secure in eCK model. Unfortunately, Deng et al. [20] pointed that two specific attacks, put forward a novel ID-2PAKA protocol without pair operation and provided the security proofs in eCK model.

In 2012, DING [21] proposed the first provable security key exchange protocol based on LWE, which has high computational efficiency and can be extended to RLWE. The protocol solves the above error elimination problem by rounding the signal function and extracting the shared key from two very close values. In 2016, Zhao et al. [22] put forward an identity-based AKA protocol from the LWE, which applies DH ephemeral key to compute key materials, introduces error item, uses encoding bases of ideal lattice as the tool for analyzing error tolerance, and makes reasonable suggests for parameters setting. In 2019, Li et al. [23] using secure public-key encryption and ciphertext compression technology designs an implicit authentication key exchange protocol. After, combining a post-quantum DH like protocol with an identity-based signcryption scheme Chen et al. [24] put forward the AKA based on elliptic curve cryptography.

In 2020, Banerjee et al. [25] proposed a quantum-secure certificate-less AKA protocol for Transport Layer Security (TLS) handshakes which saves the communication overheads by using identity-based to replace traditional certificate-based cryptography. [26] uses standard Fujisaki-Okamoto transform [27] and ID-KEM generic constructions to convert post-quantum DLP-IBE [28] to CCA-secure IBE and IND-CCA2-secure ID-KEM. Then, combining IND-CCA2-secure ID-KEM and CPA-secure NewHope-KEM [29] designs a quantum-secure ID-AKA protocol based on the FSXY construction [30]. However, although [16] provides the quantum-secure, the AKE protocol constructed by KEM leads to more communication cost. At the same year, Islam et al. [31] proposed a post-quantum 2PAKA protocol using IBC. The scheme is provably security in RO model based on CBi-ISIS and Bi-ISIS.

Recently, Gupta et al. [32] proposed a lattice-based 2PAKA protocol using IBC for IoT. However, the communication overhead is huge.

1.3 Organization of the Paper

The paper of the remaining is organized as follows. In Sect. 2, the preliminaries used in presented protocol are presented. Section 3 describes the LB-ID-AKA protocol in detail. The correctness analysis is given in Sect. 4. Section 5 analyses the sufficient security strength of the proposed scheme. Lastly, Sect. 6 concludes the paper.

2 Preliminaries

In this section, for ease of understanding, we briefly describe the technology used in presented protocol.

2.1 Compression and Decompression

Suppose for $0 < d < \lceil \log(q) \rceil$ be an integer, and q is a modules. The compression technology [33] consists of two polynomial functions: $Compress_q(x, d)$ and $Decompress_q(x, d)$. These two functions are defines as:

- $Compress_q(x, d)$: It inputs $x \in Z_q$ and outputs an integer in $\{0, \ldots., 2^d - 1\}$ where $0 < d < \lceil \log(q) \rceil$. It can be represented as .

$$Compress_q(x, d) = \left\lceil (2^d/q) \cdot x \right\rfloor mod^+ 2^d, x \in Z_q. \tag{1}$$

- $Decompress_q(x, d)$: It inputs the result of $Compress_q$ function and outputs a value x' which is an approximate value of x. It can be described as follows:

$$Decompress_q(x, d) = \left\lceil (q/2^d) \cdot x \right\rfloor, x \in Z_q \tag{2}$$

These two functions satisfy $Decompress_q\big(Compress_q(x, d), d\big) = x + e_x$ where e_x is a much small value.

2.2 Peikert's Reconciliation Mechanism

Because of the efficiency, Peikert's reconciliation mechanism is widely used in the designing of AKA protocol based-LWE problem. We give a briefly introduction as follows. The details are in the [34].

Peikert's reconciliation mechanism consists of three functions, namely modular rounding function n $\lfloor \cdot \rceil_{q,2}$, cross-rounding function $\langle \cdot \rangle_{q,2}$ and reconciliation function $Rec()$. $\lfloor \cdot \rceil_{q,2}$.

Suppose for $x \in Z_q$, cross-rounding function $\langle x \rangle_{q,2}$ represents $Z_q \to Z_2$, which can be computed as $\langle x \rangle_{q,2} = \left\lfloor \frac{4}{q} x \right\rfloor mod\ 2$.

Given an integer q, modular rounding function $\lfloor \cdot \rceil_{q,2}$ represents $Z_q \to Z_2$, which can be computed as $\lfloor \cdot \rceil_{q,2} = \left\lfloor \frac{2}{q} x \right\rceil$.

If q is an odd, above-mentioned function needs to compute in Z_{2q} rather than Z_q to eliminate the error. Thus, the function used to realize $Z_q \to Z_{2q}$ is randomized doubling function $dbl()$, which can be computed as $dbl(v) = 2v + e$.

Error reconciliation mechanism represents $Z_{2q} \times Z_2 \to Z_2$, which can be computed as $Rec(w, \sigma) = \begin{cases} 0 \text{ if} w \in I_\sigma + Emod2q \\ 1 \quad \text{otherwis}e \end{cases}$.

Where $q \geq 2$ is an integer, $Z_q = \left\{-\frac{q}{2}, \ldots, 0, \ldots, \frac{q}{2} - 1\right\}$, two intervals $I_0 = \left\{0, 1, \ldots, \left\lfloor \frac{q}{2} \right\rceil - 1\right\}$, $I_1 = \left\{-\left\lfloor \frac{q}{2} \right\rceil, \ldots, -1\right\}$ and $E = \left[-\frac{q}{4}, \frac{q}{4}\right)$.

2.3 MLWE

Module learning with errors (MLWE) was proposed by Langlois and stehle [35] in 2015 which is an extension of LWE and RLWE. Let k, m, η, n be positive integers, R_q denote $Z_q[x]/(x^n + 1)$, β_η is centered binomial distribution. MLWE is distinguishing uniform samples $(A_i, b_i = A_i s + e) \in (R_q^{m \times n} \times R_q^n)$ from sample $(A, b) \in (R_q^{m \times k} \times R_q^k)$ where $A_i \in R_q^{m \times k}$ is uniform and $b_i = A_i s + e$ with $s \leftarrow \beta_\eta^k$ common to all samples and $e \leftarrow \beta_\eta^m$ fresh for every sample. More precisely, for an algorithm $\mathcal{A}$, we define

$$\mathrm{Adv}_{m,n,q,k,\eta}^{MLWE}(\mathcal{A}) = \left| \Pr\left[b' = 1 : \begin{array}{c} A \leftarrow R_q^{m \times k}, (s, e) \leftarrow (\beta_\eta^k, \beta_\eta^m) \\ b = As + e, b' \leftarrow \mathcal{A}(A, b) \end{array} \right] \right|$$
$$- \Pr\left[b' = 1 : A \leftarrow R_q^{m \times k}, b \in R_q^m, b' \leftarrow \mathcal{A}(A, b) \right]$$

3 The Proposed LB-ID-2PAKA Protocol

In this section, the system model is described firstly. Then, we designed a post-quantum AKA protocol based on the identity cryptography for two-party communication in IoT. The symbols and descriptions used in this paper are presented in Table 1.

The LB-ID-2PAKA protocol is made up of three phases, namely, system setup phase, registration phase and authentication key agreement phase. To illustrate this further, we describe the LB-ID-2PAKA in details.

3.1 System Setup Phase

The input of Setup phase is security parameter λ. At the end of Setup phase, KGC generates the system parameters and master private key and public key pair of KGC. The details are as follows:

a) KGC picks a random vector d_{KGC} as the private key and computes the corresponding public key $P_{KGC} = d_{KGC}^T A$.
b) It selects three one-way secure hash functions $H_i : \{0, 1\}^* \to Z_q^*$ for $i = \{1, 2, 3\}$.
c) It stores d_{KGC} secret and keeps the $pp = \{\lambda, q, m, A, P_{KGC}, H_i : i \in \{1, 2, 3\}$ public.

Table 1. Symbols and corresponding descriptions.

Symbol	Description
U_i/U_j	The user
KGC	Key generate center
λ	System security parameter
q	A positive integer
d_{KGC}	Private key of KGC
P_{KGC}	Public key of KGC
A	Matrix with rank n
ID_i/ID_j	The identity of user U_i/U_j
$r_i/x_i/y_i$	Random vector
D_i	Private key of user U_i
P_i	Public key of user U_i
T_i/T_j	The timestamp of user U_i/U_j
d_x/d_y	An integer in compress function
T	Transpose operations
H_i	One-way secure hash functions
$dbl()$	Randomized doubling function
$\lfloor\cdot\rceil_{p,q}$	Modular rounding function
$\langle\cdot\rangle_{p,q}$	Cross-rounding function
Rec	Reconciliation function

3.2 Registration Phase

Before the user U_i communicates with other entities in the IoT, U_i should initiate a registration request to the KGC through a secure channel. Then KGC computes and returns corresponding private key using identity-based cryptography. As shown in Fig. 1, the details are as follows:

a) U_i selects identifier ID_i and sends the registration request $\{ID_i\}$ to the KGC through a secure channel;
b) Receiving the registration request, KGC checks whether ID_i exists in the registry. If it exists, aborts session. Else, KGC performs the c) steps;
c) KGC firstly picks a vector $r_i \in z_q^m$ at random. Then, KGC calculates the private key $R_i = r_i^T A$ and $d_i = (r_i + h_i d_{KGC}) mod q$ where $h_i = H_1(ID_i, R_i)$, and returns $U_i's$ private key $\{D_i = (d_i, R_i)\}$ to U_i;
d) After receiving D_i, U_i computes $h_i = H_1(ID_i, R_i)$ and verifies the validity of the private key by checking whether $d_i^T A? = R_i + h_i P_{KGC}$ is holds. At last, U_i calculates her/his public key $P_i = d_i^T A = R_i + h_i P_{KGC}$.

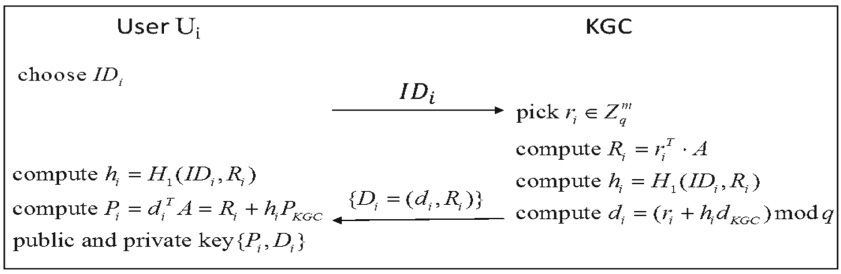

Fig. 1. The registration phase of LB-ID-2PAKA

3.3 Authentication Key Agreement Phase

Before U_i communicates with U_j, they should authenticate mutually and negotiate a common session key. The details are in Fig. 2.

a) U_i picks vectors $x_i, y_i \in z_q^m$ randomly, then calculates $Y_i = y_i^T A$, $X_i = Compress_q(Ax_i + e_1, d_x)$ and $l_i = H_2(X_i||Y_i||R_i||T_i)$, where T_i is the current timestamp. U_i computes the signature $\delta_i = y_i + l_i d_i \bmod q$ and sends the message $m_1 = \{ID_i, \delta_i, R_i, T_i, X_i, Y_i\}$ to U_j.

b) After the receipt m_1 from U_i, U_j checks whether $T_i' - T_i \leq \Delta T$ is holds. If not, the session is aborted. Else, U_j computes $l_i = H_2(X_i||Y_i||R_i||T_i)$ and $U_i's$ public key $P_i = R_i + H_1(ID_i, R_i)P_{KGC}$. Then U_j checks the validity of signature δ_i by checking whether $\delta_i^T A? = Y_i + l_i P_i$ is holds. If it is equal, the user U_j authenticates U_i successfully. Else, the session aborts. After successful authentication, U_j computes $X'_i = Decompress_q(X_i, d_x)$ to obtain the key materials. Then,U_j selects random vectors $x_j, y_j \in z_q^m$, and calculates $Y_j = Compress_q\left(A^T x_j + e_2, d_y\right)$. After that, U_j executes Peikert's reconciliation mechanism to compute session key. U_j calculates $v = X_i'^T x_j + e$, $\bar{v} = dbl(v)$, $c = \langle \bar{v} \rangle_{2q,2}$ and $K_j = [\bar{v}]_{2q,2}$. Then, U_j computes $Y_j = y_j^T A$, $l_j = H_2\left(X_j||Y_j||R_j||T_j\right)$, the signature $\delta_j = y_j + l_j d_j \bmod q$ and sends $m_2 = \left\{ID_j, \delta_j, R_j, T_j, X_j, Y_j, c\right\}$ to U_i, where T_j is the current timestamp. After sending m_2, U_j computes the session key $SK_j = H(K_j||ID_i||ID_j||T_i||T_j)$.

c) After the receipt m_2 from U_j, U_i checks whether $T_j' - T_j \leq \Delta T$ is holds. If not, the session is aborted. Else, U_i computes $l_j = H_2\left(X_j||Y_j||R_j||T_j\right)$ and $U_j's$ public key $P_j = R_j + H_1(ID_j, R_j)P_{KGC}$. Then U_i checks the validity of signature δ_j by checking whether $\delta_j^T A? = Y_j + l_j P_j$ is hold. If it is equal, the user U_i authenticates U_j successfully. Else, the session aborts. After successful authentication, U_i computes $X'_j = Decompress_q\left(X_j, d_y\right)$ to obtain the key materials. After that, the user U_i calculates $v' = 2x_i^T X'_j$ and $K_i = Rec(v', c)$ using reconciliation function. At last, U_i computes the session key $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$.

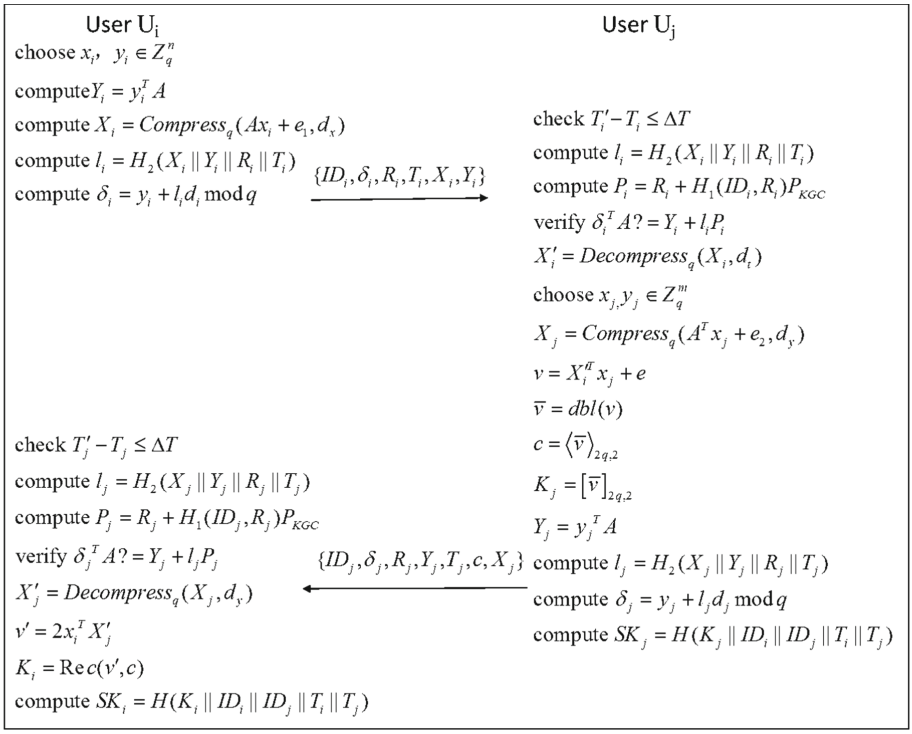

Fig. 2. The authentication key agreement phase of LB-ID-2PAKA

4 Correctness Analysis

In this section, the correctness analysis is given in this section, which consist of signature's correctness and the correctness of session key.

4.1 The Correctness of Signature

The receiver verifies whether the equation $\delta_i^T A? = Y_i + l_i P_i$ is hold to check the authenticity of the message.

Proof:

$$\begin{aligned}
\delta_i^T A &= (y_i + l_i d_i)^T A \\
&= (y_i + l_i d_i)^T A \\
&= (y_i + l_i(r_i + h_i d_{KGC}))^T A \\
&= y_i{}^T A + l_i r_i^T A + l_i h_i d_{KGC}^T A \\
&= Y_i + l_i R_i + l_i h_i P_{KGC} \\
&= Y_i + l_i(R_i + h_i P_{KGC}) \\
&= Y_i + l_i P_i
\end{aligned}$$

The verification of $\delta_j^T A? = Y_j + l_j P_j$ is similar to above-mentioned proof.

4.2 The Correctness of Session Key

It is hypothesized that $X_i \in R_q^k$, the error e_x between X_i and X'_i is very tiny, where X'_i is the result after the operation of compressing and decompressing. That is to say:

$$X'_i = Decompress_q\big(Compress_q(Ax_i + e_1, d_x), d_x\big) = Ax_i + e_1 + e_x$$

Similarly, as for $X_j \in R_q^k$, there is $X'_j = Decompress_q\big(Compress_q\big(A^T x_j + e_2, d_y\big), d_y\big) = A^T x_j + e_2 + e_y$, where e_y is the error e_x between X_j and X'_j.

As shown in Fig. 2, U_i and U_j calculate two approximately equal values $v' = 2x_i^T X'_j$ and $\overline{v} = dbl(X_i'^T x_j + e)$. The difference between them is as follows:

$$\begin{aligned}\overline{v} - v' &= 2(X_i'^T x_j + e) - \overline{e} - 2x_i^T X'_j \\ &= 2\Big((Ax_i + e_1 + e_x)^T x_j + e\Big) - 2\Big(x_i^T (A^T x_j + e_2 + e_y)\Big) - \overline{e} \\ &= 2\Big(e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e\Big) - \overline{e}\end{aligned}$$

According to the definition of Peikert's reconciliation mechanism, if modulus q is odd, the error-tolerance range is $\lfloor q/2 \rfloor$. That is to say, if the difference between $\overline{v}$ and v' satisfied $\left|\overline{v} - v'\right| \le \lfloor q/2 \rfloor$, the output of reconciliation function $Rec\big(v', c\big) = \big\lfloor \overline{v}_{2q,2} \big\rceil$.

$$\left|\overline{v} - v'\right| \le \lfloor q/2 \rfloor$$

Turn to

$$2\Big(e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e\Big) - \overline{e} \le \left\lfloor \frac{q}{2} \right\rfloor$$

$$2\Big(e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e\Big) - \overline{e} \le \left\lfloor \frac{q}{2} \right\rfloor$$

$$e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e - \frac{1}{2}\overline{e} \le \left\lfloor \frac{q}{4} \right\rfloor$$

$$e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e \le \left\lfloor \frac{q}{4} \right\rfloor \tag{3}$$

According to the definition, the $\overline{e}$ is tiny small, so $\frac{1}{2}\overline{e}$ is much smaller. Thus, the inequality can turn to (3). In contrast with the correctness proof of Kyber's public encryption scheme [36], the inequality (3) above is much the same. The inequality in Kyber's public encryption scheme is as follows (4)

$$e_1{}^T x_j + e_x{}^T x_j - x_i^T e_2 - x_i^T e_y + e + e_v < \left\lfloor \frac{q}{4} \right\rfloor \tag{4}$$

It is obvious that there is the slight difference between inequality (3) and (4). The inequality (4) increases an error vector e_v, which is much small. According to the analysis of Kyber, as long as selecting appropriate parameters, the probability that the inequality (4) is true is more than. That is to say, the probability that the inequality (3) fails is less than 2^{-128}. That is to say, the probability that our LB-ID-AKA protocol fails to negotiates the same session key is no more than 2^{-128}.

5 Security Analysis

5.1 Informal Security Analysis

In this section, the informal security analysis of the LB-ID-2PAKA is provided.

User Impersonation Attack. In the disguise of U_i, $\mathcal{A}$ must construct valid $\{ID_i, \delta_i, R_i, T_i, X_i, Y_i\}$ to pass authentication of the user U_j. However, $\mathcal{A}$ cannot acquired the correct signature δ_i without knowing d_i. Likewise, because $\mathcal{A}$ cannot construct valid authenticator δ_j, $\mathcal{A}$ fails to disguises U_j as a valid user. So the proposed LB-ID-AKA protocol can withstand user impersonation attack.

Man in the Middle Attack. As shown in Fig. 2, the user U_i and U_j achieve mutual authentication using identity-based signature. If an attacker $\mathcal{A}$ wants to launch man-in-the-middle attack, $\mathcal{A}$ needs to calculate the key materials $2x_iA^Tx_j$ from messages $\{ID_i, \delta_i, R_i, T_i, X_i, Y_i\}$ and $\{ID_j, \delta_j, R_j, T_j, X_j, Y_j, c\}$ which are from the session $\mathcal{A}$ disguises as the user U_i/U_j communicated with U_j/U_i. However, the difficulty of calculating $2x_iA^Tx_j$ is equivalent to the difficulty of solving LWE lattice hard problem, which is obviously not feasible. As a result, the proposed LB-ID-AKA protocol is immune to the man-in-the-middle attack.

Unknown Key-Share Attack. In our LB-ID-AKA protocol, the user U_i and U_j figure out the session key as $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$ and $SK_j = H(K_j||ID_i||ID_j||T_i||T_j)$ respectively, where $K_i = Rec(2x_i^T X'_j, c)$, $K_j = \left[dbl(X_i'^T x_j + e)\right]_{2q,2}$. The computation of session key consists of identity ID_i and ID_j, the key material K_i and K_j. The key material is verified by the identity-based signature δ_i and δ_j. Moreover, the adversary $\mathcal{A}$ has no knowledge of the private key d_i and d_j of the user U_i and U_j. Thus, $\mathcal{A}$ has no way to know the value of generated key SK. Therefore, our LB-ID-AKA protocol defends the unknown key-share attack.

Known-Key Attack. It is assumed that $\mathcal{A}$ obtains the current session key SK between U_i and U_j. In our LB-ID-AKA protocol, the user U_i and U_j figure out the session key as $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$ and $SK_j = H(K_j||ID_i||ID_j||T_i||T_j)$ respectively, where $K_i = Rec(2x_i^T X'_j, c)$, $K_j = \left[dbl(X_i'^T x_j + e)\right]_{2q,2}$. It is obvious that $\mathcal{A}$ has difficulty in calculating past session key, since $\mathcal{A}$ has no knowledge of ephemeral secret key x_i and x_j. Therefore, the designed LB-ID-AKA protocol is free of known-key attack.

Perfect Forward Secrecy. It is hypothesized that $\mathcal{A}$ obtains the perennial private key of communication parties U_i and U_j at T. If $\mathcal{A}$ wishes to calculate the session key before T, $\mathcal{A}$ needs to compute $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$. Because the value of K_i or K_j is calculated using Peikert's reconciliation mechanism, and the ephemeral secret key x_i and x_j are only public to itself, it's impossible for $\mathcal{A}$ to figure out past session key $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$. Therefore, our LB-ID-AKA protocol guarantees perfect forward secrecy.

No key Control. In our LB-ID-AKA protocol, the user U_i and U_j figure out the session key as $SK_i = H(K_i||ID_i||ID_j||T_i||T_j)$ and $SK_j = H(K_j||ID_i||ID_j||T_i||T_j)$ respectively, where $K_i = Rec(2x_i^T X'_j, c)$, $K_j = \left[dbl(X_i'^T x_j + e)\right]_{2q,2}$. Since the ephemeral secret key x_i and x_j are selected randomly by U_i and U_j respectively. As a result, the user $U_i(U_j)$ is incapable of making the negotiated SK_i be a pre-selected value. That is to say, a pre-selected value SK is only accepted by the party who selects. Thus, our LB-ID-AKA protocol satisfies no key control.

5.2 Security Proof

Based on the Dolev-Yao(DY) threat model [37] and random oracle model (ROM) [38], we present the adversary model for proposed LB-ID-2PAKA.

Threat Model

Participants. In a 2PAKA protocol, the participants consists of two categories: user(U_i) and key generation center(KGC). Both of them can run several instances Π. Π^t represents $t-th$ instance of an executing participant. Thus, Π_{KGC}^{t1} and $\Pi_{U_i}^{t2}$ represent of KGC and U_i respectively.

Queries. The interaction between the adversary $\mathcal{A}$ and challenger $\mathcal{C}$ is performed merely through the corresponding oracle query, which emulates the attack capability of the adversary in real world.

Setup(λ)**:** The adversary $\mathcal{A}$ inputs parameter λ, challenger $\mathcal{C}$ performs all operations in the setup phase. At the end of the query, $\mathcal{A}$ can learn the major public and system parameter.

Hash queries to $\boldsymbol{H_i}$**:** The challenger $\mathcal{C}$ keeps a hash list L_{H_i}, which is empty initially. L_{H_i} is made up of several tuples (x, y), where x is the input of $\boldsymbol{H_i}$, y is the output of $\boldsymbol{H_i}$. In response to $\boldsymbol{H_i}$ asked by $\mathcal{A}$, $\mathcal{C}$ searchs the corresponding y to given x. If L_{H_i} exists tuple (x, y), $\mathcal{C}$ outputs y. Otherwise, $\mathcal{C}$ selects and returns a random $y \in Z_q^*$, and inserts the tuple (x, y) into the L_{H_i}.

Corrupt(ID_i)**:** The challenger $\mathcal{C}$ plays the role of KGC in this query. At the end of this query, $\mathcal{A}$ registers successfully using the identity of ID_i and receives the private key of ID_i.

Unwrap_private_key(U_i)**:** To response this query, challenger $\mathcal{C}$ returns the private key of the registered user $\boldsymbol{U_i}$.

Execute$(\Pi_{i,j}^k)$**:** This query simulates the passive attack capability of $\mathcal{A}$. To reply this query, challenger $\mathcal{C}$ runs the LB-ID-2PAKA protocol Π_U^i with its partner. At the end of running LB-ID-AKA, $\mathcal{A}$ learns the message transmitting in the public channel.

Send$(\Pi_{i,j}^k, m)$: This query simulates the active attack capability of $\mathcal{A}$. To reply this query, challenger $\mathcal{C}$ plays the role of receiving the message m, he/she runs the LB-ID-AKA protocol Π_E^k with its partner. At the end of running LB-ID-2PAKA, $\mathcal{C}$ returns the message to $\mathcal{A}$. Especially, ***Send***$(\Pi_{i,j}^k, start)$ can initiate protocol running, $\mathcal{A}$ will receive the login request.

Reveal$(\Pi_{i,j}^k)$**:** This query simulates the capability of $\mathcal{A}$ initiating known-key attack. Suppose for ***Reveal***$(\Pi_{i,j}^k)$, $\mathcal{A}$ learns the past session key and uses it to perform attack. If $\mathcal{A}$ sends ***Reveal***$(\Pi_{i,j}^k)$ query to $\mathcal{C}$, challenger $\mathcal{C}$ may reply as follows:

If Π_i^k and his partner Π_j^k has negotiated session key $SK_{i,j}$, and Π_i^k and his partner Π_j^k hasn't been performed $Test()$ query, $\mathcal{C}$ outputs the fresh session key $SK_{i,j}$.

Else, $\mathcal{C}$ outputs $\perp$ to $\mathcal{A}$.

Test$(\Pi_{i,j}^k)$**:** This query doesn't simulates the actual attack capability of $\mathcal{A}$, which is used for evaluating the semantic security of the session key. It is noted that the adversary $\mathcal{A}$ ask to $\mathcal{C}$ this query only once for each $\Pi_{i,j}^k$. As for $\Pi_{i,j}^k$, if this instance doesn't generate session key $SK_{i,j}$ at the end of protocol running, $\mathcal{C}$ outputs $\perp$ to $\mathcal{A}$ as a reply; Otherwise, $\mathcal{C}$ will perform coin toss experiment to generate random b. If $b = 0$, $\mathcal{C}$ outputs the actual session key $SK_{i,j}$; Else, $\mathcal{C}$ outputs a random string of the same length with the actual session key $SK_{i,j}$.

Corrupt$(\boldsymbol{\Pi_{i,j}^k})$**:** This query simulates the corrupt user capability of $\mathcal{A}$, which is used to evaluating the perfect forward security of the protocol. As a reply, $\mathcal{C}$ outputs the past session key $SK_{i,j}$ or $\perp$ to $\mathcal{A}$.

Definition (Semantic Security of Session Key): Suppose $\mathcal{A}$ is a probabilistic polynomial time (PPT) adversary, which $\mathcal{A}$ can distinguish the random number of the same size from session key. First, $\mathcal{A}$ executes above-mentioned queries any times and only once ***Test***$(\Pi_{i,j}^k)$. Then, $\mathcal{A}$ output whether the result of ***Test***$\left(\Pi_{i,j}^k\right)$ query is session key. $\mathcal{A}$ wins the tossing coin game if ***Test***$\left(\Pi_{i,j}^k\right)$ query value b' is equal to the guessed value b of $\mathcal{A}$. The probability $\mathcal{A}$ successes in the game can be defined as

$$Adv_{\mathcal{P}}^{AKA}(\mathcal{A}) = |\Pr[\mathrm{Succ}_0] - 1| \tag{5}$$

Provable Security Theorem 1: Suppose $\mathcal{A}$ is a PPT adversary who can execute above-mentioned queries any times and only once ***Test***$(\Pi_{i,j}^k)$. The probability $\mathcal{A}$ breaks semantic security of session key can be defined as

$$Adv_{\mathcal{P}}^{AKA}(\mathcal{A}) = \frac{1}{q_{H_1}^2} + \frac{1}{q_{H_3}^2} + Adv_{p,q}^{MLWE}(\mathcal{A}_2) + Adv_{p,q}^{MLWE}(\mathcal{A}_1 \tag{6}$$

where $\mathcal{A}_1$ is an adversary who solves the decisional MLWE, q_s, q_{H_1} and q_{H_3} are the numbers of adversary executes $send()$ oracle, hash oracle, respectively.

It is proved that the negotiated session key is indistinguishable from the random bits through a series of game G_i $(i = 0, 1, 2, \ldots)$.

Game G_0: The initial game is to the real world.

$$\Pr[Succ_0] = 0 \tag{7}$$

Game G_1: In game G_1, $\mathcal{A}$ lunches $\boldsymbol{Has}\mathbf{hqueriestoH}_1$. This game is similar to G_0, and the only difference is that the l_i and l_j are no longer generated using H_2, but is sampled in the hash list. The challenger $\mathcal{C}$ keeps the tuple of H_1^{list}. The adversary $\mathcal{A}$ who can distinguish between G_0 and G_1 can also distinguish the value computed through hash function from a value searched in the H_1^{list}. Moreover, according to birthday paradox, the probability of hash collision is $1/q_{H_1}^2$. Therefore, the advantage of $\mathcal{A}$ distinguishes G_1 from G_0 is

$$\Delta_0 = |\Pr[Succ_1] - \Pr[Succ_0]| = \frac{1}{q_{H_1}^2} \tag{8}$$

Game G_2: $\mathcal{A}$ lunches $\boldsymbol{Execute}(\Pi_{i,j}^k)$ query, challenger $\mathcal{C}$ replies with $m_1 = \{ID_i, \delta_i, R_i, T_i, X_i, Y_i\}$ and $m_2 = \{ID_j, \delta_j, R_j, T_j, X_j, Y_j, c\}$ to $\mathcal{A}$ during LB-ID-AKA running. Then, $\mathcal{A}$ misuses m_1 and m_2 to learn session key $SK_j = H(K_j||ID_i||ID_j||T_i||T_j)$. The adversary $\mathcal{A}$ who can distinguish between G_2 and G_1 can also breaks MLWE. Due to the indistinguishability of G_2 and G_1, the advantage of $\mathcal{A}$ wins the game is

$$\Delta_1 = |\Pr[Succ_2] - \Pr[Succ_1]| = 0 \tag{9}$$

Game G_3: In game G_3, $\mathcal{A}$ lunches $\boldsymbol{Send}(\Pi_{i,j}^k, m)$ query. $\mathcal{A}$ forges signature $\delta_i^* = Y_i^* + l_i^* P_i$ and c^* generated at random, then lunches $\boldsymbol{Send}(\Pi_{i,j}^k, m)$ query using δ_i^* and c^* replace δ_i and c, respectively. Because the unforgeability of the signature, the probability $\mathcal{A}$ can distinguish between G_3 and G_2 is negligible. As a result, the advantage of $\mathcal{A}$ wins the game is

$$\Delta_2 = |\Pr[Succ_3] - \Pr[Succ_2]| = 0 \tag{10}$$

Game G_4: In game G_4, $\mathcal{A}$ lunches $\boldsymbol{Has}\mathbf{hqueriestoH}_3$. This game is similar to G_3, and the only difference is that the l_i and l_j are no longer generated using H_3, but is sampled in the hash list. The challenger $\mathcal{C}$ keeps the tuple of H_3^{list}. The adversary $\mathcal{A}$ who can distinguish between G_4 and G_3 can also distinguish the value computed through hash function from a value searched in the H_3^{list}. Moreover, according to birthday paradox, the probability of hash collision is $1/q_3^2$. Therefore, the advantage of $\mathcal{A}$ distinguishes G_4 from G_3 is

$$\Delta_3 = |\Pr[Succ_4] - \Pr[Succ_3]| = \frac{1}{q_{H_3}^2} \tag{11}$$

Game G_5: In game G_5, the ephemeral public keys X_i and X_j of user U_i and U_j respectively, are no longer MLWE distribution samples, but are uniformly selected at random. In G_4, $\left(A^T, X_j\right)$ and(A, X_i) are MLWE sample pairs generated using *Compress*(). Whereas in game G_5, $\left(A^T, X_j\right)$ and (A, X_i) are selected from random distribution $U(R_q^{n\times n} \times R_q^n)$ uniformly. Adversary $\mathcal{A}$ who can distinguish between G_4 and G_5 can solve the decisional

MLWE hard problem based on lattice. Therefore, the advantage of $\mathcal{A}_1$ distinguishes G_5 from G_4 is

$$\Delta_4 = |\Pr[Succ_5] - \Pr[Succ_4]| = Adv_{n,q,\eta}^{MLWE}(\mathcal{A}_1) \tag{12}$$

Game G_6: Compared with G_5, $\left(X_i', v\right)$ is no longer a sample pair from MLWE distribution in game G_6, which is a sample picked uniformly from random distribution $U(R_q^n \times R_q)$. Adversary $\mathcal{A}$ who can distinguish between G_6 and G_5 can solve the decisional MLWE hard problem based on lattice. Therefore, the advantage of $\mathcal{A}_1$ distinguishes G_6 from G_5 is

$$\Delta_5 = |\Pr[Succ_6] - \Pr[Succ_5]| = Adv_{n,q,\eta}^{MLWE}(\mathcal{A}_2) \tag{13}$$

As above, the probability $\mathcal{A}$ breaks semantic security of session key is

$$Adv_{\mathcal{P}}^{AKA}(\mathcal{A}) = \frac{1}{q_{H_1}^2} + \frac{1}{q_{H_3}^2} + Adv_{p,q}^{MLWE}(\mathcal{A}_2) + Adv_{p,q}^{MLWE}(\mathcal{A}_1)$$

6 Conclusion

With the proliferation of IoT, two-party communications attracts more and more attention, so how to secure negotiate session key in IoT has become one of the research hotspots. A large number of 2PAKA schemes have been presented recently, yet most of them are subject to quantum attack. In this article, we proposed a quantum-secure 2PAKA protocol using lattice cryptography for IoT. The proposed LB-ID-2PAKA protocol makes use of identity-based signature to avoid the complicated certificate management of PKI-based protocol. At the same time, the LB-ID-2PAKA protocol can provide desired security property and withstand various attacks, especially support perfect forward secrecy. Moreover, the provable security analysis shows that our LB-ID-2PAKA protocol is provably secure under the hardness assumption of MLWE.

Acknowledgments. This research was supported by Sichuan Science and Technology Program (no. 2022JDRC0061), the Stability Program of Science and Technology on Communication Security Laboratory (2022) and Foundation of Science and Technology On Communication Security Laboratory of China (No. 61421030107012102).

References

1. Karati, A., Islam, S.H., Karuppiah, M.: Provably secure and lightweight certificateless signature scheme for IIoT environments. IEEE Trans. Ind. Inform. **14**(8), 3701–3711 (2018)
2. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
3. Hölbl, M., Welzer, T., Brumen, B.: An improved two-party identity-based authenticated key agreement protocol using pairings. J. Comput. Syst. Sci. **78**(1), 142–150 (2012)

4. Chen, L., Cheng, Z., Smart, N.P.: Identity-based key agreement protocols from pairings. Int. J. Inf. Secur. **6**(4), 213–241 (2007)
5. Ni, L., Chen, G., Li, J., Hao, Y.: Strongly secure identity-based authenticated key agreement protocols. Comput. Elect. Eng. **37**(2), 205–217 (2011)
6. Gupta, D.S., Biswas, G.P.: A novel and efficient lattice-based authenticated key exchange protocol in C-K model. Int. J. Commun. Syst. **31**(3), e3473 (2018)
7. Dierks, T., Rescorla, E.: The transport layer security (TLS) protocol version 1.2 (2008)
8. Cui, J., Wang, Y., Zhang, J., Xu, Y., Zhong, H.: Full session key agreement scheme based on chaotic map in vehicular ad hoc networks. IEEE Trans. Veh. Technol. **69**(8), 8914–8924 (2020)
9. Ouada, F.S., Omar, M., Bouabdallah, A., Tari, A.: Lightweight identity-based authentication protocol for wireless sensor networks. Int. J. Inf. Comput. Secur. **8**(2), 121–138 (2016)
10. Gupta, D.S., Biswas, G.: An ECC-based authenticated group key exchange protocol in IBE framework. Int. J. Commun. Syst. **30**(18), 3363 (2017)
11. Gupta, D.S., Biswas, G.: On securing bi-and tri-partite session key agreement protocol using IBE framework. Wirel. Pers. Commun. **96**(3), 4505–4524 (2017)
12. Guo, C., Chang, C.-C.: An authenticated group key distribution protocol based on the generalized Chinese remainder theorem. Int. J. Commun. Syst. **27**(1), 126–134 (2014)
13. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–330 (1999)
14. Peikert, C.: Some recent progress in lattice-based cryptography. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, p. 72. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_5
15. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21
16. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange -- a new hope. In: USENIX Security Symposium, pp.327–343. USENIX, Austin (2016)
17. Ding, J., Alsayigh, S., Lancrenon, J., Saraswathy, R.V., Snook, M.: Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 183–204. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52153-4_11
18. Gupta, D.S., Islam, S.K.H., Obaidat, M.S., et al.: A provably secure and lightweight identity-based two-party authenticated key agreement protocol for IIoT environments. IEEE Syst. J. **PP**(99), 1–10 (2020)
19. Dang, L., et al.: Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks. Int. J. Distrib. Sens. Netw. **14**(4), 1–16 (2018)
20. Deng, L., Shao, J., Hu, Z.: Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks. Peer-to-Peer Netw. Appl. **14**(4), 2236–2247 (2021). https://doi.org/10.1007/s12083-021-01181-8
21. Ding, J., Lin, X.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology Eprint Archive (2013)
22. Zhao, X., Gao, H., Wang, A.: An identity-based authenticated key exchange protocol from RLWE. J. Comput. Res. Dev. **53**(11), 2482–2490 (2016)
23. Li, Z., Xie, T., Zhang, J., Xu, R.: Post quantum authenticated key exchange protocol based on ring learning with errors problem. J. Comput. Res. Dev. **56**(12), 2694–2701 (2019)
24. Chen, M.: A composable authentication key exchange scheme with post-quantum forward secrecy. J. Comput. Res. Dev. **57**(10), 2157–2176 (2020)
25. Utsav, B., Chandrakasan, A.P.: Efficient post-quantum TLS handshakes using identity-based key exchange from lattices. In: 2020 IEEE International Conference on Communications (ICC) (2020)

26. Bos, J., Costello, C., Naehrig, M., et al.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: IEEE. 2015 IEEE Symposium on Security and Privacy, pp. 553–570. IEEE, Piscataway (2015)
27. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10677, pp. 341–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12
28. Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 22–41. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_2
29. Poppelmann, T.: NewHope – algorithm specifications and supporting documentation. NIST Technical Report (2019)
30. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 467–484. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_28
31. Islam, S.H., Zeadally, S.: Provably secure identity-based two-party authenticated key agreement protocol based on CBi-ISIS and Bi-ISIS problems on lattices. J. Inf. Secure Appl. **54** (2020)
32. Daya, S., Sangram, R., Tajinder, S., Madhu, K.: Post-quantum lightweight identity-based two-party authenticated key exchange protocol for internet of vehicles with probable security. Comput. Commun. **181**, 69–79 (2022)
33. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
34. Peikert, C.: Lattice cryptography for the internet. International workshop on post-quantum cryptigraphy, pp. 197–219 (2014)
35. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. **75**, 565–599 (2015). https://doi.org/10.1007/s10623-014-9938-4
36. Bos, J., Ducas, L., Kiltz, E.: CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy (European S&P), pp. 353–367. IEEE (2018)
37. Dolev, D.A., Yao, A.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
38. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_21