Double Difficulties, Defense in Depth A succinct authenticated key agreement protocol

WenBin Hsieh

Department of Electronic and Computer Engineering, National Taipei University of Technology, Taipei, 106344, Taiwan

Elsevier use only: Received date here; revised date here; accepted date here

Abstract

In 2016, NIST announced an open competition with the goal of finding and standardizing a suitable quantum-resistant cryptographic algorithm, with the standard to be drafted in 2023. These algorithms aim to implement post-quantum secure key encapsulation mechanism (KEM) and digital signatures. However, the proposed algorithm does not consider authentication and is vulnerable to attacks such as man-in-the-middle. In this paper, we propose an authenticated key exchange algorithm to solve the above problems and improve its usability. The proposed algorithm combines learning with errors (LWE) and elliptic curve discrete logarithm problem to provide the required security goals. As forward security is a desirable property in a key exchange protocol, an ephemeral key pair is designed that a long-term secret compromise does not affect the security of past session keys. Moreover, the exchange steps required by the algorithm are very streamlined and can be completed with only two handshakes. We also use the random oracle model to prove the correctness and the security of proposed scheme. The performance analysis demonstrates the effectiveness of the proposed scheme. We believe that the novel approach introduced in this algorithm opens several doors for innovative applications of digital signatures in KEMs.

Keywords: LWE, ECDLP, AKA, PQC, KEM

1. Introduction

In recent years, as government organizations and private enterprises around the world have devoted a lot of resources in researching quantum computers, significant progress in the research and construction of quantum computers has been made. A fully-fledged quantum computer will be able to efficiently solve a distinct set of mathematical problems, such as integer factorization and discrete logarithms, which are the basis for various cryptographic schemes. In 2016, NIST announced an open competition with the goal of finding and standardizing a suitable quantum-resistant cryptographic algorithm, with the standard to be drafted in 2023. These algorithms aim to implement post-quantum secure KEM and digital signatures. However, the proposed algorithm does not consider authentication and is vulnerable to attacks such as man-in-the-middle. Just like the first famous key agreement protocol, the Diffie-Hellman (DH) key agreement protocol [1] is the basic architecture of public key cryptography. It is simple yet elegant. After its invention, countless applications based on the DH key exchange protocol or DH problem have been proposed. Now CRYSTALS-Kyber [2] is specified as a Key Encapsulation Mechanism (KEM) standard, and its security is based on the difficulty of solving the learning with errors (LWE). Nonetheless, Kyber key exchange (Kyber.KE) protocol is not resistant to attacks originally suffered by the DH protocol, such as Man-In-The-Middle (MITM) attack, lack of Perfect Forward Secrecy (PFS), etc. Although in [2], Bos et al. further proposed the authentication key exchange protocol using Kyber (Kyber.AKE). However, full forward secrecy [3] is not achievable in Kyber.AKE.

Based on the above background, we propose an authentication key agreement protocol based on two difficulties: the error learning problem and the elliptic curve discrete logarithm problem. The new protocol uses a key encapsulation mechanism to encrypt the elliptic curve digital signature algorithm, realizing identity verification and key agreement in a succinct two-way handshake.

2. Related works

In 1976, Diffie and Hellman [1] opened the door to the concept of public key algorithms. Subsequently, Rivest, Shamir and Adleman [4] proposed a concrete public key encryption scheme in 1978. After N. Koblitz [5] proposed elliptic curve cryptography (ECC) in 1985, the application of ECC was further integrated into the Diffie-Hellman key exchange algorithm and became ECDH. In addition, Scott Vanstone [6] also proposed the Elliptic Curve Digital Signature Algorithm (ECDSA) in 1992, which is an elliptic curve analog of the Digital Signature Algorithm (DSA). These pioneering algorithms have led decades of research on key exchange protocols. The mathematical problems they are based on, such as the discrete logarithm problem and the integer factorization problem, are regarded as the basis for protocol security. However, in 1994 Peter Shor [7] proposed a quantum algorithm that posed a threat to modern cryptography. With the advancement of quantum computers, post-quantum cryptography (PQC) has also emerged in response.

Among the early works on PQC focus on key encapsulation mechanisms such as Classic McEliece [8], HQC (Hamming Quasi-Cyclic) [9], BIKE (Bit Flipping Key Encapsulation) [10], NTRU Prime [11], etc. However, because the session key is dominated by one party and there is no authentication between the two parties, these mechanisms are vulnerable to many protocol attacks, such as man-in-the-middle attacks. Jintai Ding et al. [13] then proposed a key exchange scheme based on the learning with errors problems. Ding et al. introduced a randomized algorithm to generate the signal and a robust extractor to remove the bias of the distribution of the extracted key. Nonetheless, the proposed scheme is still susceptible to man-in-the-middle attacks. Guilhem et al. [14] presented an unauthenticated and thus CPA-secured secured key exchange protocol, which was selected by Hermelink et al. [15] to be instantiated as a quantum-safe algorithm on the automotive microcontroller platform AURIX^{TM}[16]. The proposed protocol generates an ephemeral key pair that is used to achieve forward. Completing this protocol requires a three-way handshake and can only achieve weak perfect forward secrecy [3]. Joppe Bos et al. then gave Kyber-AKE [2] which only required two handshakes to complete the protocol, but the proposed protocol also had weak perfect forward secrecy only.

In order to conquer the above issues, we propose a more secure and practical key agreement protocol that is called Double-Difficulty Authenticated Key Agreement (D^2AKA) protocol. The focus of this research is to implement an authentication key agreement protocol that integrates the error learning problem and the elliptic curve discrete logarithm problem and resists the attacks suffered by Kyber.KE/AKE. The main contributions of this article are summarized as follows:

(1) We propose a hybrid authenticated authentication protocol based on two different types of difficulties, namely the error learning problem and the elliptic curve discrete logarithm problem. Even if a problem is solved, there is no advantage to the adversary.
(2) We use digital signatures to further provide identity authentication to achieve mutual authentication, implicitly utilizing zero-knowledge proof. Moreover, the proposed protocol is simple and requires only two handshakes. The proposed protocol achieves perfect forward secrecy that cannot be achieved by 2-message protocols [30].
(3) We conduct security and performance analyses of our approach to validate its resilience to security attacks and its computational effectiveness. Furthermore, we compare the performance of our protocol with various existing methods and the results show that our approach is practical in terms of storage, communication, and computation costs.

The rest of this paper is structured as follows. Related works are reviewed in Section 2. In Section 3, we present preliminaries. In Section 4, we give a detailed description of our proposed authentication protocol. Section 5 presents the security analysis and section 6 gives a performance comparison. Finally, we conclude our work in Section 7.

3. Preliminary

Definition 1. Let $n \geq 1$ and $q \geq 2$ be integers, let $\alpha \in (0, 1)$. For $s \in Z_q^n$, let $A_{s,\alpha}$ be the distribution on $Z_q^n \times Z_q$ obtained by selecting a vector $a \in Z_q^n$ uniformly at random, $e \leftarrow D_{Z,\alpha q}$, and outputting $(a, \langle a, s\rangle + e)$.

The Learning with errors (LWE) problem is：for uniformly random $s \leftarrow Z_q^n$, given poly(n) number of samples that are either from $A_{s,\alpha}$ or uniformly random in $Z_q^n \times Z_q$, output 0 if the former holds and 1 if the latter holds.[4]

Definition 2. Let E be an elliptic curve defined over a finite filed F_q, and let $P \in E(F_q)$ be a point of order n. Given $Q \in \langle P\rangle$.

The elliptic curve discrete logarithm problem (ECDLP) is：Find the integer a, $0 \leq a \leq n-1$, such that $Q = aP$.[5]

3.1. LWE – Learning With Errors

The learning with errors (LWE) problem was introduced by Regev [17] as a generalization of the well-known 'learning parity with noise' problem, to larger moduli. The details can refer to M. Ruckert et al. [18] and V. Lyubashevsky et al. [19].

First we define a few parameters used in the cryptosystem: integer dimensions n_1, $n_2 \geq 1$ and an integer modulus $q \geq 2$, which relate to the underlying LWE problems; Gaussian parameters s_k and s_e for key generation and encryption, respectively; and a message alphabet Σ (for example, $\Sigma = \{0, 1\}$) with length $\ell \geq 1$; a discrete Gaussian error distributions $\chi = D_{Z,s_k}$ over the integers with the (relative) error rate $\alpha := s/q \in (0, 1)$, where $s > 0$.

A simple error-tolerant encoder and decoder is designed as follows, an encode function: $\Sigma \to Z_q$ and a decode function: $Z_q \to \Sigma$, such that for some large enough threshold $t \geq 1$, decode(encode(m) + e mod q) = m for any integer $e \in [-t, t)$. For example, if $\Sigma = \{0, 1\}$, then we can define encode(m) $:= m\cdot\lceil q/2\rfloor$, and decode($\bar{m}$) $:= 0$ if $\bar{m} \in [\lfloor -q/4\rfloor, \lfloor q/4\rfloor)$, which is contained in Z_q, and 1 otherwise. This algorithm has error tolerance $t = \lfloor q/4\rfloor$. The output of encode and decode are extended to vectors, component-wise.

- Gen(1^ℓ): With security parameter 1^ℓ, we generate a uniformly random public matrix $\tilde{A} \in Z_q^{n_1 \times n_2}$. Choose $R_1 \leftarrow D_{Z,S_k}^{n_1 \times l}$ and $R_2 \leftarrow D_{Z,S_k}^{n_2 \times l}$. Let $\tilde{P} = R_1 - \tilde{A} \cdot R_2 \in Z_q^{n_1 \times l}$. The public key is $\tilde{P}$ (and $\tilde{A}$, if needed), and the secret key is R_2.

$$[\tilde{A} \ \ \tilde{P}]\cdot\begin{bmatrix} R_2 \\ I \end{bmatrix} = R_1 \bmod q \qquad (1)$$

- Enc($\tilde{A}, \tilde{P}, m \in \Sigma^l$): Choose error vectors $e = (e_1, e_2, e_3) \in Z^{n_1} \times Z^{n_2} \times Z^l$ with each entry drawn independently from D_{Z,S_e}. Let $\tilde{m}$ = encode(m) $\in Z_q^l$, and compute the ciphertext.

$$c^t = [c_1^t \ \ c_2^t] = [e_1^t e_2^t \ e_3^t + \tilde{m}^t]\cdot\begin{bmatrix} \tilde{A} & \tilde{P} \\ I & \\ & I \end{bmatrix} \quad (2)$$

$, where\ c^t \in Z_q^{1\times(n_2 \times l)}$.

- Dec(c^t, R_2): output decode $(c_1^t \cdot R_2 + c_2^t)^t \in \Sigma^l$. Using Equation (2) followed by Equation (1), we can apply decode to

$$[c_1^t \ \ c_2^t]\cdot\begin{bmatrix} R_2 \\ I \end{bmatrix} = (e^t + [0 \ \ 0 \ \ \tilde{m}^t])\cdot\begin{bmatrix} R_1 \\ R_2 \\ I \end{bmatrix}$$
$$= e^t\cdot R + \tilde{m}^t$$

where $R = \begin{bmatrix} R_1 \\ R_2 \\ I \end{bmatrix}$. Therefore, decryption will be correct as long as each $|\langle e, r_j\rangle| < t$, the error threshold of decode. $r_j \in Z^{n_1+n_2+l}$ is the jth column of R.

3.2. ECDSA – Elliptic Curve Digital Signature Algorithm

We give a quick review to the theory of elliptic curves. In 1987, Koblitz [5] provides an introduction to elliptic curves and elliptic curve systems. For more detailed information, consult Blake et al. [20] or Menezes [21]. Some advanced books on elliptic curves are Silverman [22] and Enge [23].

Let $p > 3$ be an odd prime. An elliptic curve E over F_p is defined by an equation of the form

$$y^2 = x^3 + a + b \qquad (3)$$

where a, b $\in F_p$, and $4a^3 + 27b^2 \neq 0$ (mod p).The set $E(F_p)$ consists of all points (x, y), $x \in F_p$, $y \in F_p$, which satisfies the defined equation (3). A special point O is called the point at infinity. The sum of two points and the double of a point are defined in the follow algebraic formula.

(1) $P + O = O + P = P$ for all $P \in E(F_p)$
(2) If $P = (x, y) \in E(F_p)$, then $(x, y) + (x, -y) = O$.
(3) Let $P = (x_1, y_1) \in E(F_p)$ and $Q = (x_2, y_2) \in E(F_p)$, where $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, \text{ and}$$

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

(4) Let $P = (x_1, y_1) \in E(F_p)$, where $P \neq -P$. Then $2P = (x_3, y_3)$, where

$$y_3 = \left(\frac{3x_1^2 - y_1}{2y_1}\right)^2 (x_1 - x_3) - 2x_1$$

$$y_3 = \left(\frac{3x_1^2 - y_1}{2y_1}\right)(x_1 - x_3) - y_1$$

Then ECDSA is summarized as follows:

Domain parameters are comprised of,

(1) a field size q, where either $q = p$, an odd prime, or $q = 2^m$;
(2) an equation of the elliptic curve E over F_q is defined with two field elements a and b in F_q(i.e., $y^2 = x^3 + ax + b$ in the case $p > 3$);
(3) a finite point $G = (x_G, y_G)$ of prime order in $E(F_q)$ is defined with two field elements x_G and y_G in F_q;
(4) the order of the point G with $n > 2^{160}$ and $n > \sqrt[4]{q}$;
(5) the cofactor $h = \#E(F_q)/n$.

The procedure for generating and verifying signature using the ECDSA is described as below,

Key generation. To sign a message m, an entity with domain parameters $D = (q, a, b, G, n, h)$ and a key pair (d, Q) where d is a private key and Q is public key. The entity does the following operations:

(1) Select a random or pseudo random integer k, $1 \leq k \leq n - 1$.
(2) Compute $kG = (x_1, y_1)$ and convert x_1 to an integer $\hat{x}_1$.
(3) Compute $r = \hat{x}_1$ mod n. If $r = 0$ then goto step1.
(4) Compute k^{-1} mod n.
(5) Compute SHA(m) and convert the bit string to an integer e.
(6) Compute $s = k^{-1}(e + dr)$ mod n. If $s = 0$ then goto step 1.
(7) A signature for the message m is (r, s).

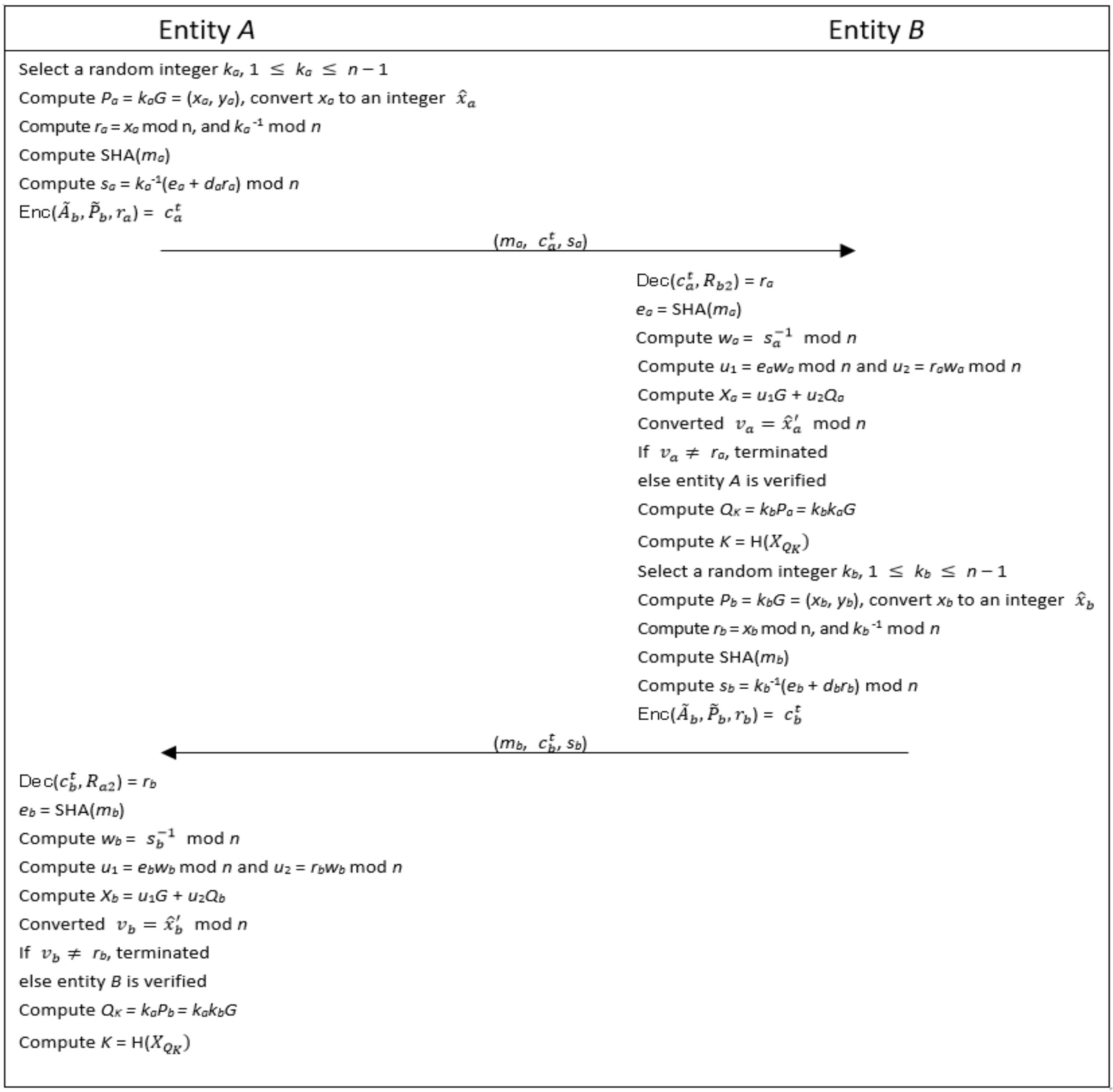
Key verification. To verify the signature (r, s) on m. The receiver does the following:

(1) Verify that r and s are integer in the interval [1, n-1].
(2) Compute SHA(m) and convert the bit string to an integer e.
(3) Compute $w = s^{-1}$ mod n.
(4) Compute $u_1 = ew$ mod n and $u_2 = rw$ mod n.
(5) Compute $X = u_1G + u_2Q$.
(6) If $X = O$, the signature is rejected. Otherwise, the x-coordinate x_1' of X is converted to an integer $v = \hat{x}_1'$ mod n.
(7) If $v = r$, the signature is accepted.

4. The proposed algorithm – Double-Difficulty Authenticated Key Agreement algorithm (D^{2}AKA)

In this section a LWE and ECDSA-based key agreement algorithm is proposed. The main security requirements of D^{2}AKA are mutual authentication (MA) and authenticated key agreement (AKA).

- MA security： D^{2}AKA ensures that the session key is known only to both communication parties involved in establishing



a session key. It allows the participants to mutually authenticate each other on exchanging the key material.

・ AKA security：D[2]AKA guarantees that only communication parties participating in the execution of the algorithm can compute the same session key. It also ensures the semantic security of established session keys.

For the definition of domain parameters, please refer to Sections 3.1.1 and 3.1.2. The detailed procedure of the proposed algorithm is depicted as following,

・Entity **A** possesses

(1) A uniformly random public matrix $\tilde{A}_a \in Z_q^{n_1 \times n_2}$,

(2) a secret key is R_{a2}, where $R_{a2} \leftarrow D_{Z,S_k}^{n_2 \times l}$, which satisfy the following equation

$$[\tilde{A}_a \ \tilde{P}_a] \cdot \begin{bmatrix} R_{a2} \\ I \end{bmatrix} = R_{a1} \bmod q, \ R_{a1} \leftarrow D_{Z,S_k}^{n_1 \times l}$$

(3) a public key $\tilde{P}_a = R_{a1} - \tilde{A}_a \cdot R_{a2} \in Z_q^{n_1 \times l}$

(4) An elliptic curve key pair (d_a, Q_a) where d_a is

a private key, Q_a is public key, and $Q_a = d_aG$.

· Entity **B** possesses

(1) A uniformly random public matrix $\tilde{A}_b \in Z_q^{n_1 \times n_2}$,
(2) a secret key is R_{b2}, where $R_{b2} \leftarrow D_{Z,S_k}^{n_2 \times l}$, which satisfy the following equation

$$[\tilde{A}_b\ \tilde{P}_b]\cdot\begin{bmatrix} R_{b2} \\ I \end{bmatrix} = R_{b1} \bmod q,\ R_{b1} \leftarrow D_{Z,S_k}^{n_1 \times l}$$

(3) a public key $\tilde{P}_b = R_{b1} - \tilde{A}_b \cdot R_{b2} \in Z_q^{n_1 \times l}$
(4) An elliptic curve key pair (d_b, Q_b) where d_b is a private key, Q_b is public key, and $Q_b = d_bG$.

Now suppose an entity ***A*** and an entity ***B*** want to negotiate a session key. The proposed authenticated key agreement algorithm is divided in the following steps,

(1) Take entity's identity as challenge message m_a, i.e. Alisa.
(2) Select a random integer k_a, $1 \le k_a \le n - 1$. Note k_a is an ephemeral private key (material).
(3) Compute $P_a = k_aG = (x_a, y_a)$ and convert x_a to an integer $\hat{x}_a$. Note P_a is an ephemeral public key (material) and if y_a is negative then goto step2.
(4) Compute $r_a = x_a \bmod n$. If $r_a = 0$ then goto step 2.
(5) Compute $k_a^{-1} \bmod n$.
(6) Compute SHA(m_a) and convert the bit string to an integer e_a.
(7) Compute $s_a = k_a^{-1}(e_a + d_ar_a) \bmod n$. If $s_a = 0$ then goto step 2.
(8) A signature for the challenge m_a is (r_a, s_a).
(9) $\mathsf{Enc}(\tilde{A}_b, \tilde{P}_b, r_a \in \Sigma^l)$: Choose error vectors $e = (e_1, e_2, e_3) \in Z^{n_1} \times Z^{n_2} \times Z^l$. Let $\widetilde{m}$ = encode(r_a), and compute the ciphertext.

$$c_a^t = [c_1^t\ \ c_2^t] = [e_1^t e_2^t\ e_3^t + \widetilde{m}^t] \cdot \begin{bmatrix} \tilde{A}_b & \tilde{P}_b \\ I & \\ & I \end{bmatrix}$$

(10) Then m_a, c_a^t and s_a are sent to entity ***B***.

After receiving the key exchange materials (c_a^t, s_a) and entity B (i.e. Bryant) carries out the following operation:

(1) $\mathsf{Dec}(c_a^t, R_{b2})$: Use B's secret key R_2 to decode c_a^t, entity B will get

$$[c_1^t\ \ c_2^t]\cdot\begin{bmatrix} R_{b2} \\ I \end{bmatrix} = (e^t + [0\ \ 0\ \ \widetilde{m}^t])\cdot\begin{bmatrix} R_{b1} \\ R_{b2} \\ I \end{bmatrix}$$
$$= e^t \cdot R_b + \widetilde{m}^t$$

,where $R_b = \begin{bmatrix} R_{b1} \\ R_{b2} \\ I \end{bmatrix}$.

The decryption will be correct as long as each $|\langle e,\ r_j \rangle| < t$, , and then r_a = decode($\widetilde{m}$).

(2) Next entity B computes SHA(m_a) and convert the bit string to an integer e_a.
(3) Then compute $w_a = s_a^{-1} \bmod n$.
(4) Compute $u_1 = e_aw_a \bmod n$ and $u_2 = r_aw_a \bmod n$.
(5) Compute $X_a = u_1G + u_2Q_a$.
(6) If $X_a = O$, the signature is rejected. Otherwise, the x-coordinate x_a' of X_a is converted to an integer $v_a = \hat{x}_a' \bmod n$.
(7) If $v_a = r_a$, the signature is accepted, entity A is verified.
(8) Bring the x-coordinate x_a' back to the curve equation (3) to find the non-negative y-coordinate value y_a' to get P_a.
(9) Select a random integer k_b, $1 \le k_b \le n - 1$. Note k_b is an ephemeral private key (material).
(10) Compute $P_b = k_bG = (x_b, y_b)$ and convert x_b to an integer $\hat{x}_a$. Note P_b is an ephemeral public key (material) and if y_b is negative then goto step8.
(11) A session key K is generated by computing

$$Q_K = k_bP_a = k_bk_aG$$
X_{Q_K} is x-coordinate of Q_K
$$K = \mathrm{H}(X_{Q_K})$$

Now that entity B gets the session key K. B implements the following steps in order to make A get the same session key:

(1) Take his identity as response message m_b to compute SHA(m_b) and convert the bit string to an integer e_b.
(2) Entity B also generates his signature by computing $r_b = x_b$ and $s_b = k_b^{-1}(e_b + d_br_b) \bmod n$.
(3) $\mathsf{Enc}(\tilde{A}_a, \tilde{P}_a, r_b \in \Sigma^l)$: Choose error vectors $e = (e_1, e_2, e_3) \in Z^{n_1} \times Z^{n_2} \times Z^l$. Let $\widetilde{m}$ = encode(r_b), and compute the ciphertext.

$$c_b^t = [c_1^t\ \ c_2^t] = [e_1^t e_2^t\ e_3^t + \widetilde{m}^t] \cdot \begin{bmatrix} \tilde{A}_a & \tilde{P}_a \\ I & \\ & I \end{bmatrix}$$

(4) Then m_b, c_b^t and s_b are sent to entity ***A***.

Finally, entity A receives (m_b, c_b^t, s_b) and carries out the following operations:

(1) $\mathsf{Dec}(c_b^t, R_{a2})$: Use A's secret key R_2 to decode c_b^t, entity A will get

$$[c_1^t \;\; c_2^t]\cdot\begin{bmatrix} R_{a2} \\ I \end{bmatrix} = (e^t + [0 \;\; 0 \;\; \widetilde{m}^t])\cdot\begin{bmatrix} R_{a1} \\ R_{a2} \\ I \end{bmatrix}$$
$$= e^t\cdot R_a + \widetilde{m}^t$$

,where $R_a=\begin{bmatrix} R_{a1} \\ R_{a2} \\ I \end{bmatrix}$.

The decryption will be correct as long as each $|\langle e, r_j\rangle| < t$, , and then r_b = decode($\widetilde{m}$).

(2) Entity A then computes SHA(m_b) and convert the bit string to an integer e_b.
(3) Then compute $w_b = s_b^{-1} \bmod n$.
(4) Compute $u_1 = e_b w_b \bmod n$ and $u_2 = r_b w_b \bmod n$.
(5) Compute $X_b = u_1 G + u_2 Q_b$.
(6) If $X_b = O$, the signature is rejected. Otherwise, the x-coordinate x_b' of X_b is converted to an integer $v_b = \hat{x}_b' \bmod n$.
(7) If $v_b = r_b$, the signature is accepted, entity B is verified.
(8) Bring the x-coordinate x_b' back to the curve equation (3) to find the non-negative y-coordinate value y_b' to get P_b.
(9) Entity A uses the ephemeral private key k_a to get the session key by computing

$K = k_a P_b = k_a k_b G$
X_{Q_K} is x-coordinate of Q_K
$K = \mathrm{H}(X_{Q_K})$

5. Security analysis of D²AKA

In security proofs by reduction, *correctness* means that if all participants follow the protocol honestly, the protocol will provide correct outputs, while *security* means that if all participants follow the protocol honestly, no one can forge a valid output. In this section, we first follow [24] to present the correctness and security of authentication and key agreement of the proposed algorithm. Then, additional security analysis of the D²AKA protocol is given at the rest of this section.

5.1. Authentication

The authentication algorithm takes as input a message-signature pair (m_i, σ_m=(r_i, s_i)), the public key Q_i with the system parameters SP. It returns "reject" if σ_m is not a valid signature of m_i signed with the corresponding private key d_i; otherwise, it returns "accept."

Correctness. Given any (d_i, Q_i, m_i, σ_m), if σ_m is a valid signature of m_i signed with d_i. The authentication algorithm will return "accept" on (Q_i, m_i, σ_m).

Security. Without the private key d_i, it is hard for any probabilistic polynomial time (PPT) adversary to forge a valid signature σ_m' on a new message m_i that can pass the authentication.

5.2. Key agreement

The key agreement algorithm takes the security parameter 1^ℓ as input and outputs a key pair (R_2, P), where R_2 is a private key and $\tilde{P}$ is a public key. An ephemeral key (material) pair (k_i, P_i) is also generated. P_i is a scalar multiplication with a scalar k_i, which means P_i equals $k_i G$. The key (material) P_i is then exchanged to negotiate a session key.

Correctness. The proof is as follows:
$\because k_b P_a = k_b k_a G = k_a k_b G = k_a P_b$
$\therefore \mathrm{H}(k_b P_a) = \mathrm{H}(k_a P_b)$

Security. A key material P_i is encrypted with the Learning With Errors (LWE) which is a quantum robust method of cryptography. Without the corresponding private key R_2, it is hard for any probabilistic polynomial time (PPT) adversary to get the key material and compute the session key.

5.3. Probable Security model

This section discusses a formal security model based on ROM [25] which proves the probable security of the D²AKA protocol. More detailed derivation and proof can be found in [26-28].

Let a PPT adversary $\mathscr{A}$ attempts to breach the semantic security of the D²AKA protocol. A challenge-response game is played between a challenger $\mathscr{C}$ and $\mathscr{A}$, where $\mathscr{C}$ helps $\mathscr{A}$ breach the semantic security of the D²AKA protocol. In this game,

$\mathscr{A}$ poses the following queries and $\mathscr{C}$ in return answers the queries.

- **Setup**(λ) : In this query, $\mathscr{C}$ is given a security parameter λ. With λ, $\mathscr{C}$ outputs a key pair (*PK*, *SK*) and a set of public parameters σ. $\mathscr{C}$ returns (*PK*, σ) to and keeps *SK* secret.
- **Query**(ϵ_i) : In this query, a list $\mathscr{L}$ is kept by $\mathscr{C}$. $\mathscr{L}$ is initially empty. A **Query** with input *u* and outputs *v* is inserted into $\mathscr{L}$ as a tuple (*u*, *v*). In response to this query with the input *u*, $\mathscr{C}$ searches the list $\mathscr{L}$ and returns *v* to $\mathscr{A}$ if the tuple (*u*, *v*) is found. Otherwise, $\mathscr{C}$ selects $v \in z_q^*$ randomly, and insert the new tuple (*u*, *v*) into $\mathscr{L}$. Then $\mathscr{C}$ returns *v* to $\mathscr{A}$
- **Execute**(ϵ_i) : In response to this query, $\mathscr{C}$ executes the D^{2}AKA protocol for the entity ϵ_i.
- **Guess**(ϵ_{ij}) : This query is allowed to ask only once in each session. $\mathscr{C}$ randomly chooses a bit *b* $\in\{0, 1\}$. If $b = 1$, the session key k_{ij} is returned to $\mathscr{A}$ by $\mathscr{C}$. Otherwise, a random value is returned as the session key.

In an active session, $\mathscr{A}$ may ask any number of oracle queries to $\mathscr{C}$ except **Guess** query. After the completion of all queries, $\mathscr{A}$ outputs a bit b'. If b' equals to *b*, $\mathscr{A}$ wins the game.

Definition 3. The probability of breaching the semantic security of session key in the D^{2}AKA protocol by an PPT adversary $\mathscr{A}$ with the polynomial time-bound *t* can be defined as

$$Adv_{A,AKA}^{PQAKA}(t) = |\Pr[b = b'] - \tfrac{1}{2}| \quad (4)$$

Definition 4. The D^{2}AKA protocol ensures the AKA security of the session key if for a PPT adversary $\mathscr{A}$,

$$Adv_{A,AKA}^{PQAKA}(t) \leq \varepsilon \quad (5)$$

Definition 5. The probability of breaching the MA security of session key in the D^{2}AKA protocol by a PPT adversary $\mathscr{A}$ within the polynomial time-bound *t* can be defined by $Adv_{A,MA}^{PQAKA}(t)$.

Definition 6. The D^{2}AKA protocol ensures the MA security of the session key if for a PPT adversary $\mathscr{A}$,

$$Adv_{A,MA}^{PQAKA}(t) \leq \varepsilon \quad (6)$$

Theorem. For any PPT adversary, the D^{2}AKA protocol demonstrates the MA and AKA security in ROM using the Learning from Errors (LWE) problem.

Proof. To prove the formal security of the D^{2}AKA protocol, ROM is used. We assume that an adversary $\mathscr{A}$ run a PPT algorithm φ to break the MA and AKA security of the proposed algorithm. A game is played between a challenger $\mathscr{C}$ and $\mathscr{A}$ in which $\mathscr{C}$ helps $\mathscr{A}$ break the semantic security of our proposal. To break the security of the D^{2}AKA protocol, $\mathscr{A}$ motives to solve the Learning from Errors (LWE) problem on which the security of the proposed algorithm is based.

$\mathscr{A}$ requests various queries to $\mathscr{C}$. In response, $\mathscr{C}$ answers its query in the following ways.

Setup(λ) : In response to this query asked by $\mathscr{A}$, $\mathscr{C}$ runs the **Setup** algorithm of the proposed D^{2}AKA protocol which outputs two key pairs (R_2, $\tilde{P}$) and (d, Q) where R_2 and *d* are private keys with $\tilde{P}$ and *Q* are the corresponding public keys. Global parameters $\sigma = (n, q, \tilde{A}, \mathrm{H}(\cdot))$ are also generated. Public keys and global parameters are then transferred to $\mathscr{A}$.

Query(ϵ_i) : To answer this query, $\mathscr{C}$ keeps a list called $\mathscr{L}$ which is initially empty. The content of this list is in the form of tuples such as (m_i, c_i^t, s_i). An adversary $\mathscr{A}$ requests this query with m_i. $\mathscr{C}$ searches $\mathscr{L}$ for (m_i, c_i^t, s_i). If search is successful, returns (m_i, c_i^t, s_i) as output. Otherwise, $\mathscr{C}$ selects a random integer k_i and $P_i = k_i G$. Then $\mathscr{C}$ generates a signature for m_i, encrypts r_i with ($\tilde{A}_a, \tilde{P}_a$), and outputs c_i^t. Then (m_i, c_i^t, s_i) is inserted into the list $\mathscr{L}$ and returned to $\mathscr{A}$.

Now $\mathscr{A}$ runs the algorithm φ to run the proposed D^{2}AKA protocol for entities *A* and *B*. The result of φ is then returned to $\mathscr{C}$. Next, $\mathscr{C}$ performs **Query**(ϵ_A) and **Query**(ϵ_B) as many times as she/he wants, using inputs m_a and m_b. gets $\mathscr{C}$ the value of c_c^t from the list $\mathscr{L}$ and she/he computes r_c for all the queries. However, $\mathscr{C}$ still cannot find the r_a and r_b used in the key agreement between entities *A* and *B*. In order to get r_a and r_b, $\mathscr{C}$ must solve the LWE problem, which is computationally difficult for any PPT algorithm. Thus, the proposed D^{2}AKA protocol is secure against AKA security under the LWE assumption.

Next, after execution of **Guess**(ϵ_{ij}) query, $\mathscr{A}$ outputs a tuple (c_i^t, s_i) to $\mathscr{C}$. Now $\mathscr{C}$ checks if

$$e^t \cdot R_a + \tilde{m}^t = \text{encode}(s_i^{-1} (e_i + d_i r_i) \cdot G) \quad (6)$$

If equation (6) does not satisfy, $\mathscr{C}$ terminates the execution. Furthermore, $\mathscr{A}$ may output another tuple (c'^{t}_{i}, s'_i). Again $\mathscr{C}$ verifies whether

$$e'^{t}\cdot R_a + \tilde{m}'^{t} = \text{encode}(s'^{-1}_{i}\ (e_i + d_i r'_i)\cdot G) \qquad (7)$$

Subtracting (6) from (7), $\mathscr{C}$ has the following expressions

$$(e^{t}\cdot R_a - e'^{t}\cdot R_a) + (\tilde{m}^{t} - \tilde{m}'^{t}) = \text{encode}(s^{-1}_{i}\ (e_i + d_i r_i)\cdot G) - \text{encode}(s'^{-1}_{i}\ (e_i + d_i r'_i)\cdot G)$$

Now, to help $\mathscr{A}$, $\mathscr{C}$ has to solve the LWE problem by computing

$$R_a = \frac{\delta - \delta' - (\tilde{m}^{t} - \tilde{m}'^{t})}{(e^{t} - e'^{t})}$$

where $\delta = \text{encode}(s^{-1}_{i}\ (e_i + d_i r_i)\cdot G)$, and $\delta' = \text{encode}(s'^{-1}_{i}\ (e_i + d_i r'_i)\cdot G)$. Thus, there is a contradiction with the LWE assumption. Therefore, the D^{2}AKA protocol attains MA security under the LWE assumption.

5.4. Further security analysis

This section describes other security features of the proposed D^{2}AKA.

(1) Man-in-the-middle (MITM) attack：In the proposed D^{2}AKA protocol, both entities A and B verify signatures for mutual authentication. Entities A and B share their messages (r_i, s_i) with each other for verification. In addition, r_i is encrypted with the key encapsulation mechanism based on the LWE problem. The transmitted messages are first decrypted and then verified by either party using the elliptic curve digital signature algorithm. The verification shows the generation of a correct session key among A and B. Suppose an adversary $\mathscr{A}$ wants to perform a MITM attack on the D^{2}AKA protocol. In order to forge a signature, $\mathscr{A}$ must solve the elliptic curve discrete logarithm problem to obtain the long-term private key d_a. Therefore, the proposed D^{2}AKA protocol can protect against MITM attacks.

(2) Unknown key-share (UKS) attack：In the proposed D^{2}AKA protocol, the entities A and B computes the session key using their ephemeral private key k_i and public key-related information r_i. This public key-related information is verified with signature s_i. In addition, r_i is protected against $\mathscr{A}$. Thus, the generated key cannot be known to $\mathscr{A}$. The proposed D^{2}AKA protocol defends the UKS attack.

(3) Known-key security (KKS) attack：In the proposed D^{2}AKA protocol, entities A and B use the ephemeral key materials to calculate the session key as $K = k_a k_b G$. It can be easy to notice that knowing the value of the current session key does not allow $\mathscr{A}$ to compute other session keys, since every session uses different ephemeral values. Therefore, the KKS attack is protected by the proposed D^{2}AKA protocol.

(4) Perfect Forward Secrecy (PFS)：In the proposed D^{2}AKA protocol, it is assumed that an adversary $\mathscr{A}$ wants to recover the past session keys after obtaining the private keys of entities A and B. Since the ephemeral secret values k_a and k_b are known only to their owning entity, $\mathscr{A}$ fails to get previous secret keys. Additionally, k_i and r_i from P_i and c^{t}_{i} due to ECDLP and the LWE difficulties Therefore, the proposed D^{2}AKA protocol exhibits PFS security property.

(5) No key control (NKC)：In the proposed D^{2}AKA protocol, both entities A and B compute the session key as $K = k_a k_b G$. The ephemeral values are k_a and k_b chosen randomly by A and B respectively. Hence A (or B) cannot force another entity B (or A) for choosing K as a pre-selected key or a small value. The pre-selected K is available to the corresponding user only and small k_i might be easily guessed. In both cases, the session key is being misused, by either the user or the adversary. In the proposed D^{2}AKA protocol, the two communicating entities make equal contributions to the establishment of a shared session key, thereby satisfying the NKC security property.

(6) Two different types of difficult problems：The proposed D^{2}AKA protocol utilizes two difficult problems of different nature to improve security. One is the LWE problem, and the other is the ECDLP. The key material

r_i is encrypted with the LWE and the session key computation is based on ECDLP. Now assume that the adversary only has the ability to solve one problem at a time. Let the adversary crack the LWE to get r_i first, then he/she still cannot compute the session key because he/she knows nothing about the ephemeral elliptical private key. Next, if the adversary has the ability to solve ECDLP, he/she still cannot calculate the session key because r_i is protected by the LWE. Two types of puzzles improve the security of the proposed protocol and avoid the risk of a problem being solved.

6. Performance analysis

In this section, the performance of the proposed D²AKA is discussed by measuring the storage as well as communication and computation costs. A comparative analysis of the proposed D²AKA protocol with DH type protocols is shown. We also made some comparisons with directly using public key encryption schemes for key agreement. The basic main idea of using PKA is as follows: for two parties A and B, they have key pairs (pk_A, sk_A) and (pk_B, sk_B) respectively. A selects a bit a uniformly at random, encrypts it using B's public key to get c_B = Enc(pk_B, a), and sends c_B to B. Similarly, B selects a uniform bit b and sends c_A to A by computing c_A = Enc(pk_A, b). A and B use their own private keys to decrypt the ciphertext and calculate $a \oplus b$.

For simplicity, to analyze the performance of the proposed scheme, we choose $n = n_1 = n_2$, q, $s = s_k = s_e$. The comparisons are given in Table 1：

Table 1 Complexity comparisons between LWE-based key agreement protocols

protocol	Pub. Param.	Commun. Comp.	Comput. Comp.	Assumption
Regev [17]	$4(n+1)n\log^2 q$	ILIT	$4n^2\log q$	**SIVP**
R. Lindner et al. [29]	$4n^2\log q$	$4(n^2+n)\log q$	$6n^2$	**SIVP**
Jintai Ding et al. [13]	$n^2\log q$	$2n\log q + 1$	$2n^2$	**SIVP**
DH-type [1]	$\log q$	$2\log q$	$2n^2$	**DHP**
Ours	$n^2\log q + \log p$	$2(n\log q + \log p + 1)$	$8n^2$	**SIVP + ECDLP**

Pub. Param. Means the size of public parameter; Commun. Comp. means the communication complexity; Comput. Comp. means the computation complexity and is estimated by the number of multiplications in Z_q.
F_p is the field on which the elliptic curve is defined.

The security comparison is presented in table 2. As mentioned at the beginning of this article, most post-quantum key exchange protocols do not consider mutual authentication, thereby being subject to various attacks such as man-in-the-middle. In terms of computational complexity analysis, we adopt more stringent cost considerations. In addition to the existing matrix operations, the proposed protocol adds point multiplication (actually scalar multiplication over the elliptic curve) and point addition operations to implement the digital signature algorithm. However, with many improved security properties, the computational cost under strict evaluation is not much higher than other protocols. These comparison analyses guarantee the betterment of the proposed D²AKA protocol that is more suitable for ensuring communication security.

Table 2 Security comparisons between LWE-based key agreement protocols

protocol	Man-in-the-middle	Known-key security	Perfect Forward Secrecy	Mutual Authentication	No key control
Regev [17]	×	×	×	×	✓
R. Lindner et al. [29]	×	×	×	×	✓
Jintai Ding et al. [13]	×	×	×	×	✓
Kyber.KE [2]	×	×	×	×	×
Kyber.AKE [2]	✓	✓	×	✓	✓
Ours	✓	✓	✓	✓	✓

7. Conclusion

This paper proposed an authenticated key agreement protocol by combining the error learning problem and the elliptic curve discrete logarithm problem. The proposed D^2AKA protocol not only provides mutual authentication but also defends against various attacks in communication protocols. Furthermore, two different types of mathematical puzzles make it more difficult for attackers to crack the proposed protocol. We also show that the proposed protocol is provably secure under the random oracle model based on the infeasibility of the LWE assumption. Performance assessment also proves that our protocol is acceptable, especially under Big O evaluation. In summary, the proposed D^2AKA will be more suitable and secure for key agreement. In the future, we will propose a general model integrating KEM and key exchange signature algorithms.

References

[1] W. Diffie and M. E. Hellman, New directions in cryptography. IEEE Transactions on Information Theory, 22(6) (1976) 644–654.

[2] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, John M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM, IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 2018, pp. 353-367.

[3] R. Canetti and H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, In: EUROCRYPT'01, 2001, pp. 453–474.

[4] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2) (1978) 120–126.

[5] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, 48 (1987), 203-209.

[6] S. Vanstone, Responses to NIST's proposal. Communications of the ACM, 35(7) (1992), 50–52.

[7] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994, pp. 124-134.

[8] M. R. Albrecht et al, Classic McEliece: conservative code-based cryptography: cryptosystem specification. Technical report, National Institute of Standards and Technology, 2022. [Online]. Available: https://classic.mceliece.org.

[9] C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichettiand G. Z´ emor, Hamming Quasi-Cyclic (HQC), 2017.

[10] Official Web Page of BIKE suite. Accessed: May 31, 2023. [Online]. Available: https://bikesuite.org.

[11] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Technical report, National Institute of Standards and Technology, 2020. [Online]. Available:https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[12] D. Hankerson and A. MenezesStein, Elliptic curve discrete logarithm problem. Encyclopedia of Cryptography and Security, 2011, pp. 397-400.

[13] J. Ding and X. Lin, A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptology ePrint Archive* 2012/688, 2012.

[14] C. de Saint Guilhem, N. P. Smart, B. Warinschi, Generic forward-secure key agreement without signatures. In: Nguyen, P.Q., Zhou, J. (eds.) Information Security - 20th International Conference, ISC 2017, Ho Chi Minh City,Vietnam, November 22-24, 2017, Proceedings. LNCS, vol. 10599, pp. 114–133. Springer, 2017.

[15] J. Hermelink, T. Pöppelmann, M. Stöttinger, Y. Wang, Y. Wan, Quantum safe authenticated key exchange protocol for automotive application, 2020.

[16] Infineon: AURIXTM32-bit microcontrollers for automotive and industrial applications, 2020. [Online]. Available: https://www.infineon.com/dgdl/ Infineon-TriCore Family BR-BC-v0100-N.pdf?fileId= 5546d4625d5945ed015dc81f47b436c7.
[17] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6) (2009), 1–40. Preliminary version in STOC 2005. On pp. 1, 2, 5, 6, 14, and 16.
[18] M. Ruckert and M. Schneider, Selecting secure parameters for lattice-based cryptography. Cryptology ePrint Archive, Report 2010/137, 2010. http://eprint.iacr.org/. On pp. 3, 4, and 10.
[19] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In EUROCRYPT'10, 2010, pp. 1–23.
[20] I. Blake, G. Seroussi and N. Smart, Elliptic Curves in Cryptography, Cambridge University Press, 1999.
[21] A. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993.
[22] J. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, 1986.
[23] A. Enge, Elliptic Curves and Their Applications to Cryptography — An Introduction, Kluwer Academic Publishers, 1999.
[24] F. Guo, W. Susilo and Y. Mu, Introduction to Security Reduction, Berlin, Germany:Springer, 2018.
[25] Mihir Bellare and Phillip Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. In Proceedings of the 1st ACM conference on Computer and communications security (CCS '93). Association for Computing Machinery, New York, NY, USA, 1993, pp. 62–73.
[26] D. S. Gupta, S. Ray, T. Singh, M. Kumari, Post-quantum lightweight identity-based two-party authenticated key exchange protocol for Internet of Vehicles with probable security, Computer Communications, 181 (2022) 69-79.
[27] Fuchun Guo, Susilo Willy, Yi Mu, Public-Key Encryption with Random Oracles, 2018, 10.1007/978-3-319-93049-7_7.
[28] Sun, Xiaochao, Bao Li, Xianhui Lu and Fuyang Fang. CCA Secure Public Key Encryption Scheme Based on LWE Without Gaussian Sampling, Conference on Information Security and Cryptology, 2015.
[29] R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In CT-RSA, 2011, pp. 319–339.
[30] H. Krawczyk. HMQV: a high-performance secure diffie-hellman protocol. In Proceedings of the 25th annual international conference on Advances in Cryptology CRYPTO'05, 2005. Springer-Verlag, Berlin, Heidelberg, 546–566.