Quantum-Safe Lattice-Based Authentication and Key Agreement Protocol for Internet of Things

Basudeb Bera, Rohini Poolat Parameswarath, *Senior Member, IEEE*, and Biplab Sikdar, *Fellow, IEEE*
Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583
E-mail: b.bera26@nus.edu.sg; rohini.p@nus.edu.sg; bsikdar@nus.edu.sg

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices introduces significant security challenges, especially with the advent of quantum computers. This paper proposes a quantum-safe authentication protocol tailored for IoT environments with constrained resources. The protocol leverages lattice based primitives to ensure secure and efficient device authentication. The proposed protocol offers a secure key agreement framework relying on the Module Learning With Errors (MLWE) problem, which resists quantum attacks. It provides resistance against classical and quantum adversaries while minimizing communication and computation overhead. The proposed protocol's robustness is verified through formal analysis using the Scyther tool. Performance evaluation demonstrates that the protocol has better security features with less computation and communication costs compared to similar protocols.**

***Index Terms*—Authentication, key agreement, post-quantum, security, IoT.**

I. Introduction

The Internet of Things (IoT) refers to a network of interconnected physical devices, sensors, and actuators that communicate and exchange data over the internet. IoT spans domains including smart homes, healthcare, industrial automation, and transportation, enabling real-time monitoring, control, and decision-making [1]. The rapid adoption of IoT technologies has transformed everyday life and industrial operations, driving efficiency, automation, and data-driven insights. Estimates indicate billions of IoT devices will be deployed globally in the coming years, generating massive volumes of data.

This exponential growth of IoT devices highlights the need for secure and reliable communication among devices, as any compromise can have cascading effects [2]. Communication among IoT devices is vulnerable to a wide range of attacks, such as eavesdropping and tampering [3], impersonation [4], and replay attacks [5]. Hence, robust security solutions must be deployed to secure communications in IoT networks [6]. Authentication serves as the first line of defense in securing IoT networks by verifying the identity of devices and users. Mutual authentication prevents unauthorized access, data breaches, and impersonation attacks, which are common in IoT environments. Strong authentication ensures secure communication among IoT devices [7], [8].

Conventional cryptographic techniques are widely used to build authentication protocols for IoT. The security of algorithms such as Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) depend on the difficulty of solving underlying mathematical problems. Though such techniques are effective today, they may become vulnerable with the advancements in quantum computing, which can break algorithms such as RSA and ECC. This emerging threat requires the urgent need for quantum-resistant authentication protocols tailored for IoT devices. A quantum-safe authentication protocol can provide security for IoT communication against future adversaries. The proposed authentication protocol addresses this gap, aiming to develop a practical authentication protocol that combines efficiency with quantum-resistant security.

The security of these algorithms relies on the hardness of underlying mathematical problems such as integer factorization problem (IFP), discrete logarithmic problem (DLP), or elliptic curve discrete logarithmic problem (ECDLP). However, due to the significant advancements in quantum computing and Shor's algorithm [13], which solves the IFP and DLP problems in polynomial time using quantum techniques, a powerful quantum computer can potentially break existing PKC schemes.

The major contribution of this paper as follows:

- In this paper, we propose a lattice-based authentication and key agreement protocol for IoT applications relying on the MLWE problem, which resists quantum attacks.
- A comprehensive performance analysis shows the robustness and efficiency of the proposed model.
- A formal security verification using the Scyther tool proves its correctness and resistance to various potential attacks.

A. Related Work

In this section, we discuss existing IoT authentication schemes. Several authentication schemes for IoT networks have been proposed in the literature. An authentication scheme based on Physical Unclonable Functions (PUFs) was proposed in [9]. Though the scheme in [9] is lighweight and hence suitable for IoT devices, it is vulnerable to Man-in-the-Middle (MITM) and impersonation attacks [10]. The authors of [10] proposed an improved authentication scheme based on Public Key Infrastructure (PKI) and PUFs. However, such PKI-based solutions have high computation costs [11]. An authentication and key agreement protocol based on Diffie-Hellman problem was proposed in [11]. An authentication protocol for IoT was proposed in [12]. Vinoth et al. [13] proposed an authentication scheme for Industrial IoT (IIoT) leveraging a secret sharing technique. Rafique et al. proposed an efficient authenitcaion protocol for IIoT environments [14]. However, the authentication schemes in [13] and [14] do not provide forward security

[15]. It can be noted that none of the above schemes are quantum-secure.

Guo et al. proposed a three-party password-authenticated key agreement protocol whose security is based on the MLWE problem in [16]. It is quantum-secure. However, the scheme in [16] is vulnerable to replay, impersonation, and DoS attacks [17]. Also, this scheme does not ensure mutual authentication completely. An improved three-party AKE scheme based on MLWE was proposed in [17]. It is also quantum-secure. Also, it is secure against several attacks such as impersonation, replay, and DoS attacks. Two multi-server authentication protocols based on the MLWE problem were proposed in [18]. In protocol 1, all servers and user together generate session keys, whereas in protocol 2, both user and servers generate session keys individually. Protocol 1 is suitable for high-security and protocol 2 is suitable for high-efficiency scenarios. However, both protocols are unable to resist replay attacks and face issues like untraceability and scalability. Yang et al. also proposed a password authenticated key exchange protocol based on MLWE in [19]. This protocol fails to resist replay attacks, does not maintain untraceability, and faces scalability issues.

II. Mathematical Preliminaries

Let a $2n$-th cyclotomic polynomial be defined as x^n+1, a polynomial ring $\mathcal{R}$ is $\mathcal{R} = \frac{\mathcal{Z}[x]}{<x^n+1>}$, and a quotient polynomial ring $\mathcal{R}_q$ as $\mathcal{R}_q^k = \frac{\mathcal{Z}_q[x]}{<x^n+1>}$, where n is the positive integer and q is a large prime. The Module Learning with Error (MLWE) was introduced by Langlois and Stehlé in 2015 [20] and extended the Learning with Error (LWE) and Ring-Learning with Error (RLWE) problems by proposing a more flexible structure. Let $\mathcal{R}_q^d$ define a vector having d ring ploynomials in $\mathcal{R}_q$ and $\mathcal{R}_q^{k\times d}$ represent a matrix of ring polynomials with rank k and dimension d. Then, MLWE problem is reduced to LWE problem, if $\mathcal{R}_q$ is set to $\mathcal{Z}_q$; and MLWE problem is reduced to RLWE problem, if $d = k = 1$. A generalized definition of MLWE is given in Definition 1 [17], [20].

Definition 1 (MLWE Problem)**.** *The MLWE problem is defined with the parameters (n, q, k, η), where n defines the degree of a polynomial or dimension of vector, q is the modulus, k defines the dimension of a polynomial matrix, and η is a parameter of a central binomial distribution β_η^k on $\mathcal{R}_q^k$. With these parameters, a vector $b = \mathbf{A}\cdot s+e \in \mathcal{R}_q^k$ is calculated, where $\mathbf{A}$ is matrix of polynomial randomly chosen from $\mathcal{R}_q^{k\times k}$, s is a secret selecting randomly and uniformly from β_η^k, and e is an error vector also chosen from β_η^k.*

Then, the challenging problem of searching MLWE based on module lattice is to find the secret $s \in \beta_\eta^k$, given $(\mathbf{A}, b)$. Another difficult problem of decision MLWE based module lattice is to determine whether a given pair $(\mathbf{A}, b)$ originates from one of two specific distributions: one where $b = \mathbf{A}\cdot s + e$ for a uniformly chosen secret $s \in \beta_\eta^k$ and an error e sampled from the distribution β_η^k; or another where b is chosen uniformly randomly from $\mathcal{R}_q^k$ [17], [18].

Definition 2. *The cross-rounding function $< x >_{q,2}$ is defined as $< x >_{q,2} = \lfloor \frac{4}{q}\cdot x \rfloor \pmod 2$, where $< x >_{q,2}$ is uniformly random when x is uniformly random.*

The modular rounding function $\lfloor x \rceil_{q,2}$ is defined as $\lfloor x \rceil_{q,2} = \lfloor \frac{2}{q}\cdot x \rceil_{q,2}$. Then, $\lfloor x \rceil_{q,2}$ is uniformly random when x is uniformly random [18].

Definition 3 (The reconciliation function (rec(.)))**.** *Given $x \in \mathcal{Z}_q$ and the cross-rounding value $y =< a >_{q,2}$ of a close value $a \in \mathcal{Z}_q$, define the sets $\mathcal{S}_0 = \{0, 1, \cdots, \lfloor \frac{q}{4} \rceil - 1\}$ and $\mathcal{S}_1 = \{-\lfloor \frac{q}{4} \rfloor, \cdots, -1\}$. Let $\mathcal{E} = [-\frac{q}{8}, \frac{q}{8})$, and the function $rec(\cdot)$ is defined as*

$$rec(x, y) = \begin{cases} 0, & if\ x \in \mathcal{S}_y + \mathcal{E} \pmod q \\ 1, & else \end{cases};$$

- *For an even q, if $x \in \mathcal{Z}_q$, $e \in \mathcal{E}$, $x + v = e \pmod q$, then $\lfloor v \rfloor_{q,2} = rec(x, < v >_{q,2})$.*
- *For an odd q, define a randomized doubling function $dbl(\cdot) : \mathcal{Z}_q \to \mathcal{Z}_{2q}$, $dbl(x) = 2x - e_0$, where $e_0 \in \{-1, 0, 1\}$. Then, the probability $Pr(e_0 = 0) = \frac{1}{2}$ and $Pr(e_0 = -1 or 1) = \frac{1}{4}$. If $v \in \mathcal{Z}_q$ is uniformly random and $\bar{v} = dbl(v) \in \mathcal{Z}_{2q}$, then $\lfloor \bar{v} \rfloor_{2q,2} \in \mathcal{Z}_{2q}$ is also uniformly random [18]. The function $dbl(\cdot)$ is extended to polynomials $f \in \mathcal{R}_q$ by applying it to each of f's coefficients.*

III. Proposed Quantum-safe AKE Scheme

This section provides various phases of the proposed scheme, called QSLAKE-IoT.

1) Network Model: The network model of the proposed scheme is shown in Fig. 1. In this model, we consider a general IoT application where smart devices (SD)s are connected with associated edge server (ES). SDs communicate through a wireless communication channel after their successful registration by the ES. During their registration, the ES loads public and private parameters, which will be used for their mutual authentication process, and then they establish a common secure key for their secure communication.

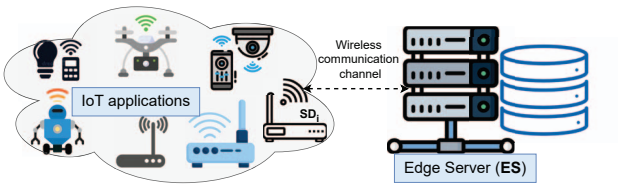

Fig. 1. Network model.

2) Threat Models: In the proposed scheme, SDs communicate with the ES via a wireless public channel, which raises a security concern. Here, we adopt widely accepted threat models, such as the Dolev-Yao (DY) model [21], the Canetti and Krawczyk (CK) model [22], and the extended CK-adversary (eCK) models [23]. Utilizing the DY threat model, an adversary $\mathcal{A}$ is capable of not only eavesdropping on the

communication but also modifying, deleting, or inserting fake content into the communication channel. In the CK-adversary model, $\mathcal{A}$ gains the extra power to not only alter, delete, and inject the false information into the communication channel, but also can hijack or seize the channel. As a result, $\mathcal{A}$ compromises a session state and then discloses both ephemeral and long-term secrets that are used in generating the session key. Whereas, under the eCK threat model, $\mathcal{A}$ guesses a session key by executing a sequence of queries and performing the $Test(sid)$ query by compromising a session state with session id sid. After some time, $\mathcal{A}$ checks whether the session key is original or a random value. In order to win the experiment, $\mathcal{A}$ has to both guess the challenge correctly and the test session has to be clean. Additionally, $\mathcal{A}$ is also capable of launching a quantum lattice reduction attack or other quantum attacks to find a short vector to recover the secret session keys with Grover's search algorithm [24].

3) Initial Phase: In this phase, the edge server ES selects the following initial parameters for the proposed scheme as follows:

Step 1: The ES selects MLWE parameters (n, q, k, η), picks a $2n$-th cyclotomic polynomial as $x^n + 1$ and defines a polynomial ring $\mathcal{R}$ as $\mathcal{R} = \frac{\mathcal{Z}[x]}{<x^n+1>}$ and a quotient polynomial ring $\mathcal{R}_q^n$ as $\mathcal{R}_q^n = \frac{\mathcal{Z}_q[x]}{<x^n+1>}$.

Step 2: The ES selects a central binomial distribution β_η on $\mathcal{R}_q^n$ and defines a one-way random hash oracle, say H_1, as $H_1 : \{0,1\}^r \rightarrow \chi_\gamma$, which takes an arbitrary input and produces a fixed length r as output in distribution χ_γ [25].

Step 3: The ES selects a polynomial matrix $\mathbf{A} \in \mathcal{R}_q^{k \times k}$, a master secret key mk, identity id_s, and chooses a hash function $h(\cdot)$ (here, SHA-256 for quantum secure).

Step 4: Finally, the ES publishes the parameters $\{n,\ q,\ k,\ \eta,\ \mathbf{A},\ \chi_\gamma,\ H_1(\cdot),\ h(\cdot)\}$ as public and stores $\{mk,\ id_s\}$ in its memory as secrets. The ES is placed under physical locking system to avoid physical capture attacks.

4) Enrollment Phase: Step 1: The ES selects a unique and distinct identity id_i, a temporal identity tid_i, and a secret key u_i for each SD_i. The ES then computes authentication factor $A_f = h(id_i \ ||mk \ ||RTS_i \ ||id_s)$, where RTS_i is a registration timestamp.

Step 2: The ES then stores $\{id_i,\ tid_i,\ A_f,\ u_i\}$ into a SD_i's tamper-resistant memory through secure channel or offline mode. The ES also stores $\{\{id_i,\ tid_i,\ A_f\}$ into its memory.

5) Authentication and Key Exchange (AKE) Phase: The ES and a SD_i execute the following process to authenticate each other and to share a secure session key.

Step 1: SD_i selects $(s_i, e_i) \in \beta_\eta^k \times \beta_\eta^k$, a timestamp TS_1, and random nonce $n_i \in \{0,1\}^{256}$. Next, SD_i computes $x_i = h(n_i \ ||u_i \ ||id_i \ ||\ TS_1)$, $x_i' = x_i \oplus A_f$, $b_i = \mathbf{A} \cdot s_i + e_i \in \mathcal{R}_q^k$, $a_i = b_i + H_1(A_f \ ||id_i \ ||TS_1)$, and $c_i = h(a_i \ ||tid_i \ ||A_f \ ||x_i' \ ||TS_1)$. SD_i then generates a message MSG_1 as $\{tid_i,\ TS_1,\ a_i,\ c_i,\ x_i'\}$ and sends it to the ES via public channel.

Step 2: The ES receives MSG_1 at TS_1^* and verifies if $|TS_1^* - TS_1| < \Delta T$, where ΔT is maximum delay of MSG_1 in the network. If this holds, the ES fetches id_i and A_f corresponding to tid_i, derives $x_i = x_i' \oplus A_f$, and computes $c_i' = h(a_i \ ||tid_i \ ||A_f \ ||x_i' \ ||TS_1)$. Next, the ES verifies if $c_i' = c_i$. If so, the ES computes $b_i = a_i - H_1(A_f \ ||id_i \ ||TS_1)$. The ES then picks $(s_j, e_j) \in \beta_\eta^k \times \beta_\eta^k$, a random $n_j \in \{0,1\}^{256}$, a fresh timestamp TS_2, and $(e_l, e_0) \in \beta_\eta$. The ES computes $b_j = \mathbf{A}^T \cdot s_j + e_j \in \mathcal{R}_q^k$, $a_j = b_j + H_1(A_f \ ||id_i \ ||TS_1)$, $d_j = b_i^T \cdot s_j + e_l \in \mathcal{R}_q$, and $f_j = dbl(d_j) (= 2d_j + e_0 \in \mathcal{R}_{2q})$. Next, the ES computes $\alpha_j = <f_j>_{2q,2} \in \{0,1\}^n$, $\delta_j = \lfloor f_j \rceil_{2q,2} \in \{0,1\}^n$, $x_j = h(n_j \ ||mk \ ||id_s \ ||TS_2)$, and $x_j' = x_j \oplus A_f$. The ES then constructs a session key SK as $SK = h(x_j \ ||x_i \ ||\delta_j \ ||id_i \ ||TS_1 \ ||TS_2)$, picks a new pseud-identity tid_n, computes $tid_i' = tid_n \oplus h(SK \ ||tid_i \ ||TS_2)$, $\alpha_j' = \alpha_j \oplus x_j$, and a verifier $SKV = h(SK \ ||tid_i' \ ||\alpha_j' \ ||TS_2 \ ||x_j' \ ||a_j \ ||TS_1 \ ||a_i)$. Next, the ES builds a reply message MSG_2 as $\{SKV,\ TS_2,\ \alpha_j',\ tid_i',\ a_j,\ x_j'\}$ and sends it to SD_i via public channel.

Step 3: SD_i receives MSG_2 at TS_2^* and verifies its freshness by the condition: $|TS_2^* - TS_2| < \Delta T$. If this is satisfied, SD_i derives $b_j = a_j - H_1(A_f \ ||id_i \ ||TS_1)$, computes $x_j = x_j' \oplus A_f$, $d_j' = b_j^T \cdot s_i \in \mathcal{R}_q$, $\alpha_j = \alpha_j' \oplus x_j$, $\delta_j' = rec(2d_j', \alpha_j) \in \{0,1\}^n$, and a session key $SK' = h(x_j \ ||x_i \ ||\delta_j' \ ||id_i \ ||TS_1 \ ||TS_2)$. Next, SD_i derives $tid_n = tid_i' \oplus h(SK' \ ||tid_i \ ||TS_2)$, and the verifier $SKV' = h(SK' \ ||tid_i' \ ||\alpha_j' \ ||TS_2 \ ||x_j' \ ||a_j \ ||TS_1 \ ||a_i)$. After that, SD_i verifies if $SKV = SKV'$. If this holds, SD_i believes that it has established the same session key $SK' (= SK)$ and then updates old tid_i to new tid_n.

6) Dynamic Device Joining Phase: When a new device, say SD_j, joins the network, SD_j follows the below steps:

Step 1: The ES first picks unique and distinct identity id_j, a temporal identity tid_j, and a secret key u_j for each SD_j. The ES then calculates an authentication factor $A_j = h(id_j \ ||mk \ ||RTS_j \ ||id_s)$, where RTS_j is a registration timestamp.

Step 2: The ES then stores $\{id_j,\ tid_j,\ A_j,\ u_j\}$ into a SD_j's tamper-resistant memory through secure channel or offline mode and stores $\{id_j,\ tid_j,\ A_j\}$ into its memory.

IV. Informal Security Analysis

In this section, we describe how the proposed protocol prevents the following attacks.

Replay attack: In this attack, $\mathcal{A}$ sends older messages repeatedly to the receiver ES by capturing previous messages. However, QSLAKE-IoT utilizes fresh timestamps $\{TS_1, TS_2\}$ in message construction for each session for fresh communication between SD and the ES. Therefore, if $\mathcal{A}$ tries to replay previous messages, it can be easily detected by verifying timestamps. Hence, QSLAKE-IoT resists replay attacks.

Man-in-the-Middle (MiTM) attack: In a MiTM attack, $\mathcal{A}$ tries to communicate to the ES on behalf of a SD on the fly. To do so, $\mathcal{A}$ must construct $MSG_1 = \{tid_i,\ TS_1,\ a_i,\ c_i,\ x_i'\}$. Then, $\mathcal{A}$ can pick its own timestamp TS_1', random nonce $n_i' \in \{0,1\}^{256}$, and $(s_i', e_i') \in \beta_\eta^k \times \beta_\eta^k$. However, to compute $x_i = h(n_i' \ ||u_i \ ||id_i \ ||\ TS_1')$, $x_i' = x_i \oplus A_f$, and $b_i' = \mathbf{A} \cdot s_i' + e_i'$, $\mathcal{A}$ needs to know the values of the secrets $\{u_i,\ id_i,\ A_f\}$. Without these secrets, $\mathcal{A}$ cannot generate a valid MSG_1. It is worth noticing that these pieces of information are not shared via public channel as a plaintext; therefore, $\mathcal{A}$ cannot have these secrets. Thus, QSLAKE-IoT is resilient against MiTM attacks.

Device impersonation attack: In this attack, $\mathcal{A}$ impersonates SD_i and then starts communication with the ES. To do so, $\mathcal{A}$ needs to generate a valid message MSG_1. However, to generate a legitimate MSG_1, $\mathcal{A}$ needs the long-term secrets $\{u_i, id_i, A_f\}$ as well as short-term secrets $\{s_i, n_i\}$. However, these long-term secrets are stored in SD_i's tamper-resistant memory and resist any attempt to extract them. Therefore, QSLAKE-IoT resists this attack.

Privileged-insider attack: During the SD enrollment process, the ES loads the registration credentials, and after successful registration, the ES deletes SD's secret u_i from its memory. Therefore, any attacker being an insider cannot get any privilege to have these secrets. Therefore, QSLAKE-IoT resists this attack.

Ephemeral secret leakage (ESL) attack: In this attack, $\mathcal{A}$ aims to capture long-term secrets $\{u_i, id_i, A_f\}$ and ephemeral secrets $\{s_i, n_i\}$ by hijacking a session state using the CK and eCK-adversary model to generate a valid session key SK, where $SK = h(x_j \,||x_i\, ||\delta_j\, ||id_i\, ||TS_1\, ||TS_2)$. To construct the SK, $\mathcal{A}$ needs the values of x_i and x_j, where $x_i = h(n_i\, ||u_i\, ||id_i\, ||\, TS_1)$ and $x_j = h(n_j\, ||mk\, ||id_s\, ||TS_2)$. These secrets are not shared via public channel; instead, they are shared in their protected form using the collision-resistant one-way hash function $h(\cdot)$. In addition, the session key is generated with session-specific secrets $\{s_i, n_i, s_j, n_j, e_i, e_j, e_0\}$ which are different in every session. Therefore, if $\mathcal{A}$ compromises a session, that should not affect the previous and following sessions. Thus, QSLAKE-IoT is not vulnerable to ESL attacks under the CK-adversary model.

Anonymity and untraceability: During the AKE process, a SD and the ES share the communication messages MSG_1 and MSG_2 via public channels. Both parties do not share their real identities over the public channel. SD shares its temporal identity, and in each session, this identity is updated with a new one; therefore, eavesdropping on the channel cannot reveal their real identities. As a result, $\mathcal{A}$ cannot disclose their real identities, and then anonymity is preserved.

Each message is generated with a fresh timestamp and session-specific random nonce, which makes these messages dynamic. In addition, in each final session they update their temporal identities, which makes the message more dynamic. Therefore, if $\mathcal{A}$ captures the messages, it cannot find who communicated with whom. Thus, QSLAKE-IoT preserves the untraceability property.

Quantum attacks: The security of QSLAKE-IoT is based on the computational hardness of the MLWE problem. According to Definition 1, the difficult problem of search MLWE based on module lattice is to find the secret $s \in \beta_\eta^k$, given $(\mathbf{A}, b)$, where $b = \mathbf{A} \cdot s + e$, $s \in \beta_\eta^k$, and $e \in \beta_\eta^k$. In QSLAKE-IoT, we use $(s_i, e_i) \in \beta_\eta^k \times \beta_\eta^k$ and $(s_j, e_j) \in \beta_\eta^k \times \beta_\eta^k$ to generate the session key SK. Therefore, it is a challenging task for $\mathcal{A}$ to find these secrets using both classical and quantum algorithms in polynomial time. It is noticed that the session key SK is generated using SHA-256, a quantum-safe cryptographic hash function. Therefore, $\mathcal{A}$ cannot break the MLWE problem using even the quantum lattice reduction attack and the quantum computing attack. As a result, any $\mathcal{A}$ having a probabilistic polynomial-time machine (a quantum computer) cannot break the session key in polynomial time. Furthermore, the probability of breaking the MLWE by $\mathcal{A}$ is extremely low, making the advantage of breaking the session key negligible. Thus, QSLAKE-IoT protects against quantum attacks.

V. TestBed Experimental Setup, Results, and Implementation

In this section, we provide testbed experimental results for the computation times for cryptographic primitives used in our proposed scheme. We wrote the scripts in Python language and utilized cryptographic library cryptography 37.0.2 for this testbed. For the configuration setup, we consider ES as a laptop configured with Ubuntu 22.04 LTS, featuring 8 GB RAM and an Intel® Core™ i7-9750H processor, CPU running at 2.60 GHz, equipped with 6 cores and 12 threads, operating on a 64-bit architecture with a 256 GB SSD and SD is considered as a Raspberry Pi 4 Model B configured with Raspberry Pi 4 Model B Rev 1.5, featuring a 64-bit Cortex-A72 processor clocked at 1800 MHz with 4 cores and 7.6 GB of RAM, running Ubuntu 20.04.6 LTS on an aarch64 architecture. We also consider NIST standard parameters, such as $n = 256$ and $q = 3329$ for RLWE-based primitives and $\eta = 3$, $k = 3$ for MLWE-based primitives.

Let T_h denote the time (in milliseconds) for computing $h(\cdot)$, T_{senc}/T_{sdec} represent the time for computing AES-128 encryption and decryption, and T_{eca}/T_{ecm} indicate the time for ECC point addition and multiplication using curve secp256r1. In addition, T_g, T_{sm}, T_{pm}, T_{pa}, T_{cha}, and T_{mlwe} represent the time for sampling from χ_γ, a component-wise multiplication with a scalar in $\mathcal{R}_q$, a component-wise polynomial multiplication in $\mathcal{R}_q$, a component-wise polynomial addition in $\mathcal{R}_q$, the characteristic function in $\mathcal{R}_q$, and MLWE-based polynomial generation of the form $b = \mathbf{A} \cdot s + e \in \mathcal{R}_q^k$, respectively. Table I shows the average execution times over 1,000 times for each primitive.

TABLE I
AVERAGE EXECUTION TIMES (IN MS) OF CRYPTOGRAPHIC PRIMITIVES

Operation	Smart device	Server
T_h	0.3621	0.0212
T_{senc}	0.3366	0.0079
T_{sdec}	0.3422	0.0078
T_{ecm}	2.7823	0.5294
T_{eca}	0.3912	0.0742
T_g	0.0344	0.0104
T_{sm}	0.0227	0.0051
T_{pm}	4.3638	0.2835
T_{pa}	0.1337	0.0069
T_{cha}	0.6129	0.0713
T_{mlwe}	6.5610	0.6320

VI. Formal Security Verification under Scyther Tool

We verify the security of our proposed protocol using the Scyther tool. It offers clear termination guarantees for any security protocol with unlimited sessions and infinite state

aggregation. It also supports parallel analysis of multiple protocols. It uses security protocol description language (.spdl) to write a security protocol for verification. It supports predefined security models, including the DY threat model, CK-adversary, eCK-adversary, and others [26]. It supports various claims for security tests, such as secrecy and authentication aspects, including aliveness, weak agreement, agreement, and synchronization. The claim "secret" ensures confidentiality and various authentication claims like "Alive," "Niagree," "Nisynch," and "Weakagree" that help to find attacks, like replay, reflection, and MiTM attacks. The Alive guarantees that all events are successfully executed by both parties, SD (named as DeviceSD) and ES (named as ServerES). Nisynch ensures that all messages are successfully sent and received, whereas Niagree describes a non-injective agreement, that is, both SD and ES believe they have successfully run the protocol and that they both agree on the same session key. Weakagree guarantees that the protocol remains resilient against impersonation attacks. Further details can be accessed in the Scyther manual [27]. The outcome of the simulation is presented in Fig. 2, where two roles, DeviceSD and ServerES, are defined. The result also indicates that Scyther did not find any potential attacks within the proposed scheme.

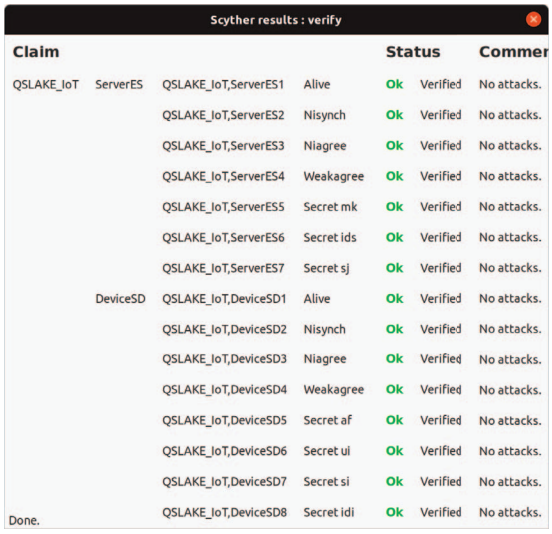

Fig. 2. Simulation results using Scyther tool.

VII. Performance Analysis

1) Communication Costs Comparison: For communication cost calculation, we consider the size of random numbers is 160 bits, identities is 160 bits, timestamps is 32 bits, hash outputs using the SHA-256 is 256 bits, the AES encryption/decryption key is 128 bits, polynomials in R_p is 4096 bits, and cha/w_i is 1 bit.

QSLAKE-IoT requires two messages for the AKE process, $MSG_1 = \{tid_i, TS_1, a_i, c_i, x_i'\}$ and $MSG_2 = \{SKV, TS_2, \alpha_j', tid_i', a_j, x_j'\}$. Based on our assumption, MSG_1 needs (160 + 32 + 4096 + 256 + 256) = 4800 bits and MSG_2 requires (256 + 32 + 256 + 256 + 4096 + 256) = 5152 bits, respectively. Thus, QSLAKE-IoT incurs a total communication cost of 9952 bits. Table II shows the comparison of QSLAKE-IoT with other existing works, and it is worth noticing that the proposed QSLAKE-IoT scheme incurs lower costs compared to the other schemes.

TABLE II
Comparative analysis on communication costs

Scheme	No. of messages	Total cost (in bits)
Guo et al. [16]	4	27874
Park et al. [17]	4	21122
Yang et al. [18]	4	28252
Yang et al. [19]	4	28080
Ahmad and Jagatheswari [28]	4	18210
QSLAKE-IoT	2	9952

2) Computation Costs Comparison: Based on our testbed experimental results mentioned in Section V, we calculate the computation costs. The proposed QSLAKE-IoT scheme requires computation costs of $5T_h + 3T_g + T_{pm} + T_{mlwe} \approx 12.8385$ ms for SD_i, whereas the ES requires costs of $5T_h + 3T_g + T_{pm} + T_{mlwe} \approx 1.0631$ ms. Table III shows the comparison of the computation costs of the proposed scheme along with existing schemes. It is worth noticing that the proposed QSLAKE-IoT requires less computation cost, making it practical and efficient in real-world applications.

TABLE III
Comparative analysis on computation costs

Scheme	SD/smart device	Server
Guo et al. [16]	$8T_h + 5T_g + 4T_{pm}$ $+2T_{pa} + T_{senc} + T_{sdec}$ $+2T_{mlwe} \approx 34.5922$ ms	$4T_h + 5T_g + 2T_{pm} +$ $2T_{pa} + T_{senc} + 2T_{mlwe}$ ≈ 1.9895 ms
Park et al. [17]	$8T_h + 5T_g + 2T_{pm}$ $+2T_{pa} + 2T_{mlwe}$ ≈ 25.1858 ms	$6T_h$ ≈ 0.1272 ms
Yang et al. [18]	$3T_h + 2T_g + 2T_{pm}$ $+2T_{mlwe}$ ≈ 16.4437 ms	$4T_h + 2T_g + 4T_{pm}$ $+3T_{pa} + T_{mlwe}$ ≈ 1.8923 ms
Yang et al. [19]	$4T_h + 2T_g + 2T_{pm}$ $+T_{mlwe}$ ≈ 16.8058 ms	$8T_h + 5T_g + 4T_{pm}$ $+3T_{pa} + 2T_{mlwe}$ ≈ 2.6403 ms
Ahmad and Jagatheswari [28]	$4T_h + 4T_g + 2T_{sm} +$ $4T_{pm} + 2T_{pa} + T_{cha}$ ≈ 19.9669 ms	$5T_h$ ≈ 0.1060 ms
QSLAKE-IoT	$5T_h + 3T_g + T_{pm}$ $+T_{mlwe}$ ≈ 12.8385 ms	$5T_h + 4T_g + T_{pm}$ $+T_{mlwe}$ ≈ 1.0631 ms

3) Security and Functionality (FS) Features: Table IV compares the security and functionality features of the proposed QSLAKE-IoT scheme with other existing protocols. Notably, the proposed QSLAKE-IoT scheme satisfies all FS features, while the others lack support for certain features, indicating that they fall short of achieving the intended security standards.

VIII. Conclusion

The proposed quantum-safe authentication protocol ensures long-term resilience of IoT systems against both classical and quantum threats. Its lightweight design makes it practical for resource-constrained devices without compromising security. Results confirm its security and feasibility for large-scale IoT deployments. This establishes it as a reliable solution for securing next-generation IoT environments. For future work, further optimization of computation and energy costs will be investigated.

TABLE IV
COMPARATIVE ANALYSIS ON VARIOUS FS ATTRIBUTES

Attribute (FS)	[17]	[16]	[18]	[19]	[28]	Proposed scheme
FS_1	✓	×	×	×	×	✓
FS_2	✓	✓	✓	✓	✓	✓
FS_3	✓	×	✓	✓	✓	✓
FS_4	✓	✓	✓	✓	✓	✓
FS_5	✓	×	✓	✓	✓	✓
FS_6	✓	✓	✓	✓	✓	✓
FS_7	✓	✓	✓	✓	✓	✓
FS_8	✓	✓	✓	✓	×	✓
FS_9	✓	×	✓	✓	✓	✓
FS_{10}	×	×	×	×	×	✓
FS_{11}	×	×	×	×	×	✓
FS_{12}	✓	×	×	×	✓	✓
FS_{13}	×	×	×	×	×	✓

"FS_1: Replay attack; FS_2: MITM attack; FS_3: Mutual authentication; FS_4: Key Agreement; FS_5: Device impersonation attack; FS_6: Device physical capture attack; FS_7: ESL attack under the CK-adversary model; FS_8: Anonymity ; FS_9: Privileged-insider attack; FS_{10}: Dynamic device joining addition phase; FS_{11}: Untraceability; FS_{12}: Security verification under Scyther/AVISPA/ProVerif; FS_{13}: Scalability.

✓: A scheme supports an attribute; ×: A scheme does not support an attribute.

IX. ACKNOWLEDGEMENT

This work is fully supported by the Advanced Research and Technology Innovation Centre (ARTIC), the National University of Singapore under Grant (project number: AFP-RP2).

REFERENCES

[1] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, "A Survey on Intelligent Internet of Things: Applications, Security, Privacy, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1238–1292, 2025.

[2] M. Hossain, G. Kayas, R. Hasan, A. Skjellum, S. Noor, and S. M. R. Islam, "A Holistic Analysis of Internet of Things (IoT) Security: Principles, Practices, and New Perspectives," *Future Internet*, vol. 16, no. 2, 2024.

[3] R. P. Parameswarath, N. Venkata Abhishek, and B. Sikdar, "PREVENT: A Mechanism for Preventing Message Tampering Attacks in Electric Vehicle Networks," in *97th Vehicular Technology Conference (VTC'2023)*, Florence, Italy, 2023, pp. 1–5, doi: 10.1109/VTC2023-Spring57618.2023.10200289.

[4] D. D. N. Nguyen, K. Sood, Y. Xiang, L. Gao, and L. Chi, "Impersonation Attack Detection in IoT Networks," in *IEEE Global Communications Conference (GLOBECOM'22)*, Rio de Janeiro, Brazil, 2022, pp. 6061–6066, doi: 10.1109/GLOBECOM48099.2022.10001392.

[5] M. Adil, M. K. Khan, N. Kumar, M. Attique, A. Farouk, M. Guizani, and Z. Jin, "Healthcare Internet of Things: Security Threats, Challenges, and Future Research Directions," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 046–19 069, 2024.

[6] L. A. Maghrabi, "Automated Network Intrusion Detection for Internet of Things: Security Enhancements," *IEEE Access*, vol. 12, pp. 30 839–30 851, 2024.

[7] W. Wang, M. Liu, and M. Chen, "CA_DeepSC: Cross-Modal Alignment for Multi-Modal Semantic Communications," in *IEEE Global Communications Conference (GLOBECOM'23)*, Kuala Lumpur, Malaysia, 2023, pp. 5871–5876.

[8] J. Yao, T. Wang, M. Chen, L. Wang, and G. Chen, "GBS-AKA: Group-Based Secure Authentication and Key Agreement for M2M in 4G Network," in *International Conference on Cloud Computing Research and Innovations (ICCCRI'16)*, Singapore, 2016, pp. 42–48.

[9] P. Gope and B. Sikdar, "Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019.

[10] Z. Siddiqui, J. Gao, and M. Khurram Khan, "An Improved Lightweight PUF-PKI Digital Certificate Authentication Scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19 744–19 756, 2022.

[11] Z. Liu, C. Hu, C. Ruan, Y. Pu, P. Hu, and J. Yu, "LCL-AKA: Lightweight Authentication and Key Agreement Protocol for Power IoT," *IEEE Transactions on Smart Grid*, vol. 16, no. 5, pp. 4128–4142, 2025.

[12] A. Akli and K. Chougdali, "IOTA-Assisted Self-Sovereign Identity Framework for Decentralized Authentication and Secure Data Sharing," *IEEE Access*, vol. 13, pp. 80 191–80 205, 2025.

[13] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure Multifactor Authenticated Key Agreement Scheme for Industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3801–3811, 2021.

[14] F. Rafique, M. S. Obaidat, K. Mahmood, M. F. Ayub, J. Ferzund, and S. A. Chaudhry, "An Efficient and Provably Secure Certificateless Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 8039–8046, 2022.

[15] Y. Han, H. Guo, J. Liu, B. B. Ehui, Y. Wu, and S. Li, "An Enhanced Multifactor Authentication and Key Agreement Protocol in Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 243–16 254, 2024.

[16] S. Guo, Y. Song, S. Guo, Y. Yang, and S. Song, "Three-Party Password Authentication and Key Exchange Protocol Based on MLWE," *Symmetry*, vol. 15, no. 9, 2023.

[17] H. Park, S. Son, Y. Park, and Y. Park, "Provably Quantum Secure Three-Party Mutual Authentication and Key Exchange Protocol Based on Modular Learning with Error," *Electronics*, vol. 13, no. 19, 2024.

[18] Y. Yang, S. Song, and S. Guo, "Multi-server Password authenticated Key Exchange Protocol Based on MLWE," in *Proceedings of 5th International Conference on Computer Network Security and Software Engineering (CNSSE'25)*. Association for Computing Machinery, 2025, pp. 164–177.

[19] Y. Yang., S. Song, and S. Guo, "Two-server password authenticated key exchange protocol based on mlwe," in *Proceedings of 4th International Conference on Cryptography, Network Security and Communication Technology (CNSCT'25)*. Association for Computing Machinery, 2025, pp. 55–68.

[20] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.

[21] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[22] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[23] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*. Wollongong, Australia: Springer Berlin Heidelberg, 2007, pp. 1–16.

[24] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symposium on Theory of Computing (STOC'96)*, Philadelphia, Pennsylvania, USA, 1996, pp. 212–219.

[25] J. Ding, P. Branco, and K. Schmitt, "Key Exchange and Authenticated Key Exchange with Reusable Keys Based on RLWE Assumption," Cryptology ePrint Archive, Paper 2019/665, 2019, https://eprint.iacr.org/2019/665. Accessed on May 2025.

[26] J. Cao, M. Ma, Y. Fu, H. Li, and Y. Zhang, "CPPHA: Capability-Based Privacy-Protection Handover Authentication Mechanism for SDN-Based 5G HetNets," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1182–1195, 2021.

[27] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification*, A. Gupta and S. Malik, Eds. Princeton, NJ, USA: Springer Berlin Heidelberg, 2008, pp. 414–418.

[28] A. Ahmad and S. Jagatheswari, "Lattice-Based Three Party Authenticated Key Agreement Scheme in Medical IoT for Post-Quantum Environment," *IEEE Access*, vol. 12, pp. 157 247–157 259, 2024.