Adversarial Attack Detection and Mitigation in Next-Generation Intelligent Consumer Electronics

Basudeb Bera, Samaresh Bera, *Senior Member, IEEE*, Madhusanka Liyanage, and Biplab Sikdar, *Fellow, IEEE*

***Abstract*—In next-generation consumer electronics (NGCEs), advanced CE devices heavily rely on machine learning and artificial intelligence (AI), where data and models are the basic building blocks for making automated decisions and optimized services. Nowadays, generative adversarial networks has led to model inversion as well as model/data poisoning attacks in NGCEs that can significantly degrade the model performance and accuracy. In addition, the majority of the CEs data are collected through vulnerable wireless public channels, posing significant security challenges to data privacy, integrity, and confidentiality. However, traditional authentication schemes relying on the discrete logarithm problem (DLP) have been utilized form the past few decades for securing CE's data. With the advancement of quantum computing and the adaptation of the quantum Shor's algorithm, these schemes become no more secure and therefore vulnerable to quantum attacks. Moreover, the wireless CEs applications are vulnerable to impersonation attacks by malicious entities.**

To overcome these challenges, we propose a radio frequency fingerprint (RFF)-based adversarial attack detection and mitigation with a lightweight quantum-secure authentication and key agreement protocol for securing CEs communication. We utilize a vector similarity search technique for attack detection and unauthorized access using the CE's RFF. A comprehensive performance analysis and comparative analysis presents its efficiency and scalability for CE applications. A testbed experiment using Raspberry Pi, verification using the Scyther tool, and performance under unknown attacks shows the practicability, correctness, and robustness of the proposed scheme. The network performance using NS-3 highlights the novelty of the proposed scheme.

***Index Terms*—Attack detection, authentication, key agreement, security, consumer electronics, Scyther, RFF, similarity search.**

I. Introduction

The recent advances of Internet of things and consumer electronics enabled end-users to monitor their daily lifestyle through seamless connectivity across various sectors like e-healthcare, smart homes, and smart cities, among others [1]. The next-generation consumer electronic devices (NGCEs) enabled with Internet connectivity often exchange real-time information with cloud servers and third parties. Furthermore, NGCEs are also enabled with artificial intelligence/machine learning (AI/ML) for real-time inferences and decision making [2]. Consequently, the performance of NGCEs heavily rely on the underlying security measures that are adopted to overcome vulnerabilities of users' authenticity, security and privacy [3]. However, the current development on generative AI, for example, generative adversarial networks (GANs), has led to model inversion and model and data poisoning attacks. It has shown the trained AI models are also vulnerable to adversarial attacks that result in the recreation of users' fake data in training processes and degradation of the model performance in terms of accuracy, sensitivity, and specificity [4], [5]. The attackers also would provide malicious input to fool the system, resulting in false prediction. This poses a long-term security threat to the data that requires long-term security, privacy, and confidentiality [6]. In addition, CE devices communicate over the vulnerable wireless open channel, posing significant security challenges, including replay attacks, man-in-the-middle (MITM) attacks, impersonation attacks, and ephemeral secret leakage (ESL) attacks, among others [7], [8].

The traditional authentication schemes [9]–[13] have been utilized for securing the wireless communication system in CE or related applications and most of these protocols rely on the discrete logarithm problem (DLP). Due to the rapid advancement of quantum computing and adaptation of quantum algorithms, such as Shor's algorithm [14] and Grover's algorithm [15], these schemes become no more secure and vulnerable to quantum attacks. Although National Institute of Standards and Technology (NIST) standardized few quantum-secure techniques, including module-lattice-based key-encapsulation mechanism, such as CRYSTALS-KYBER, module-lattice-based digital signature, and stateless hash-based digital signature (for more details, please see FIPS 203, FIPS 204, FIPS 205), provide security in current wireless communication systems. It is noticed that these protocols require significant operational costs in terms of communication overhead, computation costs, and power utilization, making them impractical and inefficient for real-world resource-constrained CE applications [16].

The radio frequency fingerprint identification (RFFI) technique is a promising method that depends on the device's intrinsic hardware impairments of radio frequency (RF) components generated during the manufacturing process [17]. The hardware characteristics of these components cannot affect their normal communication functionality. The hardware characteristics can be considered as a unique and distinct device's fingerprint, meaning no two devices have the same fingerprint, similar to human biometrics, and these are very hard to clone or tamper with [18]. RF uses waveform-level imperfections

B. Bera and B. Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: b.bera26@nus.edu.sg, bsikdar@nus.edu.sg).

S. Bera is with the Department of Computer Science and Engineering, Indian Institute of Technology Jammu, 181221, India. (email: s.bera.1989@ieee.org)

M. Liyanage is with the Network Softwarization and Security Labs (NetsLab), School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland (e-mail: madhusanka@ucd.ie).

(Corresponding author: Basudeb Bera)

imposed by the RF circuit to obtain a fingerprint of the wireless device. The transmitter device generates the carrier signal, which is transmitted into a waveform, and then the receiver can capture the signal. The receiver then identifies transmitters by extracting the device-specific RFFs on a per-packet basis [19]. All the RFFI operations are accomplished at the receiver, and there is no alteration of the transmitter. The impairments generally include oscillator drift, carrier frequency offset (CFO), sampling offset, phase offset, mixer imperfection, in-phase/quadrature (I/Q) imbalance, phase noise, and power amplifier non-linearity, among others, which are usually unique and can hardly be imitated by adversarial devices [20]. Since the RFF supports intrinsic hardware impairments and are inherently difficult to replicate, thereby increasing difficulty of successful spoofing for any adversary. Hence, it is particularly suitable for power-constrained and low-cost CE devices.

Motivated by the security issues with traditional approaches and the advantages of RFFI-based approach, in this paper, we propose an RFF-based attack detection and mitigation model using vector similarity search (VSS) [21]. The proposed model integrates physical-layer device fingerprinting with post-quantum cryptography to detect impersonation attacks and offers a lightweight and reliable device authentication for ensuring secure communication in NGCE networks.

The major technical novelty of this paper are as follows:

- We propose an attack detection framework utilizing VSS, where the model uses the RFF of communicating device to verify its authenticity based on pre-loaded information and then detects any attacks or unauthorized access.
- We also design a quantum-secure key agreement protocol for mitigating the attacks and future-proof communication using the ring learning with errors (RLWE) hardness problem [22]. Our model integrates physical-layer device fingerprinting with post-quantum cryptography to detect impersonation attacks.
- We conduct security verification of the proposed protocol using the Scyther tool and security verification to demonstrate robustness and resistance against active and passive quantum classical attacks.
- We present a comprehensive performance analysis and its efficiency and scalability in real-world CE applications compared to existing approaches. We also present simulation results from networking aspects, such as throughput, latency, and packet delivery ratio. Furthermore, we conduct a testbed experiment using Raspberry Pi to show utilization of cryptographic primitives. It is shown to be practicable for resource-constrained CE applications.

The rest of the paper is organized as follows. Section II provided a comprehensive literature review on the related application by discussing their security issues and advantages. The proposed scheme is discussed in Section IV, and Section V provides security analysis along with formal security verification using the Scyther tool. Section VI highlights the real-time testbed experiment of the cryptographic primitives required for computation costs and power usage calculation, and Section VII provides the performance analysis of the proposed scheme compared with existing models. Section VIII includes the conclusion of the proposed scheme and future research direction.

II. Related Works

Subramanian et al. [23] proposed a post-quantum authentication (PQA) protocol replying on NIST standard algorithm, such as Kyber and Dilithium algorithms. However, these baseline algorithm requires huge operational costs in term of communication, computation, storage, and power consumption, making their scheme inefficient for the resource-constraint CEs. Yang et al. [24] proposed a hybrid post-quantum framework combining AES-128, ECC-P256, CRYSTALS-Kyber, and CRYSTALS-Dilithium for CEs. In their scheme, data confidentiality is maintained by encryption technique using AES-128 and message authentication using Dilithium signatures and ensuring forward secrecy. However, their scheme requires significant operational costs, making it heavy and impracticable for real-world CE applications. Similarly, Shahidinejad et al. [25] proposed a PQA for mobile devices relying on the Kyber algorithm, making it inefficient for lightweight CE devices.

Ahmad and Jagatheswari [26] proposed a PQA and key agreement protocol for medical IoT applications, and their baseline security relies on the ring learning with errors (RLWE) problem. In their scheme, real identities of communicating parties are exchanged over an insecure public channel, disclosing them to an adversary. Therefore, their scheme fails to preserve anonymity and untraceability properties. Moreover, their scheme is vulnerable to key reuse attacks and replay attacks. Mishra et al. [27] suggested a PQA and key agreement model for Internet of drone (IoD) applications, depending on the RLWE-hardness. In their model, a user and drone establish a session key after exchanging a sequence of messages with their real identities over a vulnerable public channel. Thus, their scheme faces leakage of real identities and untraceability issues. In addition, their scheme is vulnerable to key reuse attacks and fails to support dynamic node joining. Another similar PQA and key agreement scheme designed by Rewal et al. [28] for mobile devices relying on RLWE-hardness, where the user and sensor exchange messages and at the end establish a session key. However, in this scheme, real identities are shared over a public channel, and thus, this scheme fails to preserve anonymity and untraceability. In addition, their scheme is vulnerable to key reuse attacks, does not support the dynamic node addition phase, and faces scalability issues.

Guo et al. [29] designed a PQA and key exchange protocol based on the module learning with errors (MLWE)-hardness problem. In their model, client A and client B exchange messages and establish a session key. However, their scheme requires huge operation costs, does not ensure proper mutual authentication, and is vulnerable to replay, impersonation, and DoS attacks [30]. An improved version of [29] is proposed by Park et al. in [30]. However, their scheme fails to preserve the untraceability property and does not support the dynamic node joining phase, making it inefficient. Moreover, utilization of MLWE, their scheme, requires significant operation costs, making it impracticable for real-world resource-constrained applications. Yang et al. [31] proposed an MLWE-based PQA

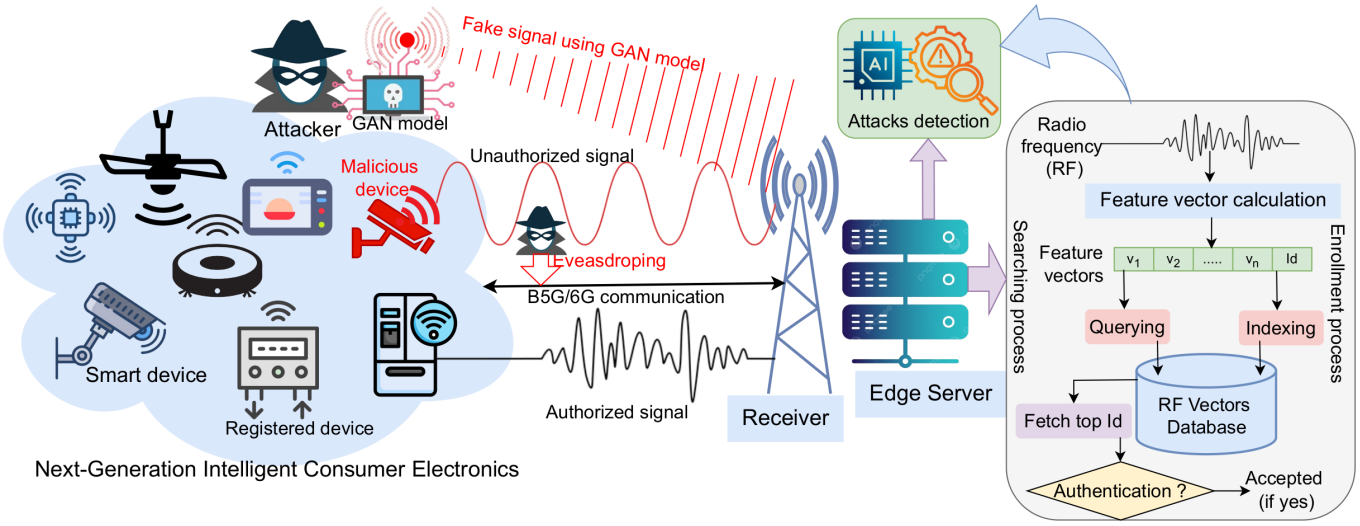


Fig. 1. The network model, showing SD communicating with the ES over a public channel.

and key exchange protocol for multi-server environments; however, their protocol is unable to resist replay attacks and faces untraceability and scalability issues. In addition, due to the requirement of huge communication and computational costs, their protocols become inefficient for lightweight devices, like CEs.

Dwivedi et al. [10] proposed an authentication scheme for IoD applications and the hardness relying on one-way hash function and elliptic curve cryptography. Unfortunately, their scheme becomes vulnerable to quantum attacks and faces scalability issues. Gunda et al. [32] designed an authentication and key exchange protocol for CEs, leveraging biometric systems and the elliptic curve Diffie-Hellman (ECDH) problem. Their protocols use factors like smart cards, passwords, and user biometrics for their authentication techniques. Unfortunately, their model fails to resist quantum attacks and faces scalability issues. The summary of this literature survey is tabulated in Table I, which includes the advantages, limitations, and cryptographic methodologies of the existing schemes.

III. Mathematical Preliminaries

Let $\mathbf{Z}$ be a set of integers and $n \in \mathbf{Z}$ be a large integer, where n is a power of 2 that is characterized by $n = 2^l$, $l > 0$. Let $p \in \mathbf{Z}$ be a large odd prime number, $\mathbf{Z}[x]$ be a polynomial ring over $\mathbf{Z}$, and $\mathbf{Z}_p[x]$ be a polynomial ring under the modulo operation under $\mathbf{Z}_p$. $\mathbf{R}_p$ be a polynomial quotient ring of $\mathbf{R}$ under modulo p, defined as $\mathbf{R}_p = \frac{\mathbf{Z}_p[x]}{<x^n+1>}$, where $< x^n + 1 >$ is a 2n-th cyclotomic irreducible polynomial over $\mathbf{Z}$. χ_γ denote a discrete Gaussian distribution (called error distribution) over $\mathbf{R}_p$, where $\gamma > 0$ is the standard deviation of the distribution. Define $\mathbf{S} = \{-\frac{p}{4}, \cdots, \frac{p}{4}\}$ be a subset of $\mathbf{Z}_p = \{-\lfloor\frac{p-1}{2}\rfloor, \cdots, \lfloor\frac{p-1}{2}\rfloor\}$, which is the middle half $\mathbf{Z}_p$. Let a characteristic function $Cha(x)$ which takes the input x and defined as $Cha(x) = 0$ if $x \in \mathbf{S}$ and 1 otherwise. An auxiliary modular function $Mod_2 : \mathbf{Z}_p \times \{0,1\} \rightarrow \{0,1\}$, is defined as $Mod_2(x,y) = (x + y \cdot \frac{(p-1)}{2}) \pmod p \pmod 2$ [22].

Lemma 1. *Assume p be an odd prime, random samples $x, e \in \mathbf{Z}_p$, s.t., $|e| < \frac{p}{8}$. If $a = x + 2 \cdot e$, then $Mod_2(x, Cha(x)) = Mod_2(a, Cha(x))$ holds [22].*

Proof. The proof of Lemma 1 is provided in [22]. □

Ring Learning With Error (RLWE) Problem: Let s be an element of $\mathbf{R}_p$ and $\mathbf{A}_{s,\chi_\gamma}$ be a distribution of $(a, a \cdot s + e)$, where $a \in \mathbf{R}_p$ and $e \in \chi_\gamma$, respectively. Then, the RLWE problem states that it is hard to distinguish $\mathbf{A}_{s,\chi_\gamma}$ from the uniform distribution on $\mathbf{R}_p^2$ for any probabilistic polynomial time algorithm with only polynomially many samples [22].

IV. Proposed Scheme (ADMICE)

In this section, we discuss the various phases of the proposed attack detection using RFF and mitigation using the post-quantum authentication and key exchange (AKE) model for NG Intelligent CEs, called ADMICE. Since the traditional classical public-key cryptographic schemes based on the DLP, such as RSA and ECC, are mostly vulnerable to quantum attacks due to Shor's algorithm. This algorithm can solve integer factorization and discrete logarithms in polynomial time on a sufficiently powerful quantum computer. Therefore, in our model we replace DLP-based primitives with lattice-based constructions grounded in RLWE hardness assumptions such that no efficient quantum algorithms are currently known for breaking this hardness.

A. Network Model

We consider various intelligent CE devices are deployed and communicate with their associated distributed edge server (ES) network. Before deployment, each smart device (SD) is registered with the ES by a registration authority ensuring initial credentials and RFF profiles, such as oscillator

TABLE I
CRYPTOGRAPHIC METHODS, ADVANTAGES AND LIMITATIONS OF EXISTING SCHEMES IN IOT-BASED CE DEVICES

Scheme	Year	Cryptographic methods/hardness	Advantages	Drawbacks/Limitations
Subramanian et al. [23]	2025	* Kyber Key Encapsulation Mechanism * Dilithium * Based on MLWE problem	* Authentication * Key exchange	* Require huge operational costs * Lack of security verification * Does not support dynamic node addition
Yang et al. [24]	2025	* Symmetric encryption/decryption * ECC * CRYSTALS-Dilithium * One-way hash function * Based on MLWE problem	* Key agreement	* Require huge operational costs * Lack of security verification * Does not support dynamic node addition
Shahidinejad et al. [25]	2025	* Kyber KEM * Symmetric encryption/decryption * One-way hash function * Based on MLWE problem	* Authentication * Key exchange	* Require huge operational costs * Fail to adversarial attack detection * Scalability issues
Ahmad and Jagatheswari [26]	2024	* Lattice-based PQC operations * characteristic function * One-way hash function * Based on RLWE hardness	* Authentication * Key agreement	* Vulnerable to replay attack * Anonymity and untraceability issues * Vulnerable to key reuse attacks * Scalability issues * Fail to adversarial attack detection
Mishra et al. [27]	2023	* Lattice-based PQC operations * One-way hash function * characteristic function * Based on RLWE hardness	* Authentication * Key agreement	* Vulnerable to key reuse attack * Anonymity and untraceability issues * Scalability issues * Fail to adversarial attack detection
Rewal et al. [28]	2023	* Lattice-based PQC operations * One-way hash function * characteristic function * Based on RLWE hardness	* Authentication * Key agreement	* Vulnerable to key reuse attack * Fails to provide anonymity and untraceability * Scalability issues * Fail to advesarial attack detection * Lack of security varification
Guo et al. [29]	2023	* Symmetric key encryption/decryption * One-way hash function * Lattice-based operations * Based on MLWE hardness	* Authentication * Key exchange	* Require huge operational costs * Vulnerable to privileged-insider attack * Scalability issues * Fail to adversarial attack detection
Park et al. in [30]	2024	* Cryptographic hash function * Lattice-based operations Based on MLWE hardness	* Authentication * Key exchange	* Anonymity and untraceability issues * Scalability issues * Fail to adversarial attack detection
Yang et al. [31]	2021	* Cryptographic hash function * Lattice-based operations Based on MLWE hardness	* Authentication * Key exchange	* Cannot resist replay attack * Untraceability and scalability issues * Require huge operational costs * Fail to adversarial attack detection
Dwivedi et al. [10]	2023	* ECC * One-way hash function Based on ECDLP hardness	* Authentication * Key exchange	* Scalability issues * Fail to quantum attacks * Fail to adversarial attack detection
Gunda et al. [32]	2023	* Fuzzy extractor functions * One-way hash function * ECC	* Authentication * Key exchange	* Face scalability issues * Vulnerable to quantum attacks * Fail to adversarial attack detection * Lack of security verification

drift, CFO, sampling offset, phase offset, I/Q imbalance, and phase noise are established. After successful registration, they communicate over the public channel with their RFF. After a sequence of message exchanges, SD and ES establish a quantum-secure session key, which is used to share their sensing information. The RFF is used to monitor their activity and protect any unauthorized access or detect any suspicious activity. Figure 1 represents the network model of the proposed scheme. The attack detection and mitigation module is deployed at the edge server, as depicted in the figure.

B. Threat Models

Since each communication between an SD and the ES is performed over the vulnerable public channel, which poses significant security challenges, including data privacy, integrity, and confidentiality. In this work, we consider the widely recognized adversarial threat models, such as the Dolev-Yao (DY) threat model [33] and the Canetti and Krawczyk (CK) adversary model [34]. In addition, we consider a quantum-capable adversary present in the network, where the adversary can record encrypted traffic during transmission and decrypt it later when quantum computing becomes available. This scenario poses a long-term security threat to the data that requires long-term confidentiality. The details are provided as follows:

DY threat model: Under the DY threat model, an adversary $\mathcal{A}$ can eavesdrop, manipulate, delete, or inject make content into the communication medium.

CK-adversary model: $\mathcal{A}$ gains an additional capability than the DY threat model, including seizing short-term and long-term secrets, along with session states, during a session establishment phase.

In addition, $\mathcal{A}$ can physically capture a device and initiate side-channel attacks, such as power analysis attacks [35], to extract stored information from the compromised device's memory. We assume that $\mathcal{A}$ may try to impersonate a legitimate SD by transmitting fake signals.

C. Initial Setup Phase

The edge server ES is responsible for selecting initial functions for the proposed scheme as follows:

Step 1: The ES selects a large integer $n \in \mathbf{Z}$ and a large prime number $p \in \mathbf{Z}$. Next, the ES chooses polynomial rings $\mathbf{Z}[x]$ and $\mathbf{Z}_p[x]$ over $\mathbf{Z}$ and $\mathbf{Z}_p$, respectively.

Step 2: The ES picks a 2n-th cyclotomic polynomial as x^n+1 and defines a polynomial ring $\mathbf{R}$ as $\mathbf{R} = \frac{\mathbf{Z}[x]}{<x^n+1>}$ and a a quotient ring $\mathbf{R}_p$ as $\mathbf{R}_p = \frac{\mathbf{Z}_p[x]}{<x^n+1>}$.

Step 3: The ES selects a error distribution χ_γ over $\mathbf{R}_p$, where γ is the standard deviation of the distribution.

Step 4: The ES defines a one-way hash function, say H_1, as $H_1 : \{0,1\}^r \rightarrow \chi_\gamma$, which inputs as any random string of fixed length r and outputs an element in χ_γ. This is a random oracle and its outputs are sampled from χ_γ [36].

Step 5: The ES chooses a hash function $h()$ (here, SHA-256 for a quantum secure), picks a master secret key $mk \in \mathbf{Z}_p$, and samples a public polynomial $b \in \mathbf{R}_p$. Finally, the ES publish the parameters $\{n, p, b, \chi_\gamma, H_1, h(\cdot)\}$ as public and stores mk in its memory as secret key.

D. Enrollment and Database Creation Phase

In this phase, the ES enrollments each CE smart device SD_i and creates a vector database of their RFF in a secure and controlled environment, ensuring no malicious device is not enrolled and an attacker cannot interfere with this phase. This process is executed as follows:

• The ES picks a unique and distinct identity Id_i, a pseudo-identity Sid_i, and computes $t_i = h(Id_i || mk || RTS)$, where RTS is the registration timestamp.

• The ES deploys the legitimate SD_is into authorized places and collects the RFFs of these devices using the RFF extractor. These SD_is need to send several packets in this controlled setup so that the ES can extract feature vectors from their RFFs from the received packets using the RFF extractor. Next, the ES creates a RFF database $(RFVD)$ with their individual identities Id_is using Algorithm 1, where k be the total number of features per RFF sample. Note that each SD_i's RFFs is unique and distinct; therefore, no two devices have the same RFF, just as no two human being have same biometrics fingerprint. This uniqueness makes RFFs hard to clone or spoof. The RFFs of a newly joined devices will be updated to the $RFVD$ and the RFFs of a device that leave the system will be deleted form the $RFVD$.

Algorithm 1 RF Vector Database Creation

procedure CREATERFVD(RFF, Id_i)
  $< v_j >|_{j=1}^{k} \leftarrow ExtractFeatures(RFF)$
  $RFVD \leftarrow Indexing(< v_j >|_{j=1}^{k} \cup Id_i)$
  return $RFVD$
end procedure

• Next, the ES chooses a matching threshold value, say η for VSS and then stores the parameters $\{\eta,\ RFVD,\ (Id_i,\ Sid_i)\}$ into its secure memory. The ES then stores $(Id_i,\ Sid_i,\ t_i)$ into SD_i's memory securely.

E. Attack Detection and Mitigation Phase

In this section, we elaborates attack detection and mitigation using RF-based authentication. A detailed is provided as follows.

*1) **Attack Detection Phase**:* In this phase, the ES detects any suspicious activities or any attack attempt by adversary $\mathcal{A}$. The ES continuously monitors for any suspicious RF signals that do not match any enrolled device. An unknown signal could indicate an adversary attempting to communicate. The ES follows the below steps to detect and counter any attempt of attacks by $\mathcal{A}$.

• The ES define a threshold value for attack detection as α, the maximum number of consecutive unrecognized signals tolerated before declaring an attack by $\mathcal{A}$. Let the ES receives a radio frequency signal $RFF_{\mathcal{A}}$ of $\mathcal{A}$.

• The ES then executes the Algorithm 2. In this algorithm, the ES extracts the features vectors of $RFF_{\mathcal{A}}$ as $< v_j >|_{j=1}^{k}$ $ExtractFeatures(RFF_{\mathcal{A}})$ and then runs the querying algorithm to get its identity as Id_i' $Querying(RFVD, < v_i >|_{j=1}^{k})$. Next, the ES verifies whether Id_i' exists in the identity set $\{Id_i\}_{i=1}^{N}$, where N is the total number of CE devices in the network. If this exists, then the ES believes there is no attack. If it does not satisfy the ES, then increase the flag as $flag \leftarrow flag + 1$ and check whether this flag reaches the predefined threshold value α. If it is not reached, then the ES again receives the RF signal and similarly checks the identity and flag value. Whenever the flag value reaches the threshold without matching the identity in any attempt, the ES then believes that there is an attack attempt by the $\mathcal{A}$ and then raises an alarm to mitigate the threat.

Algorithm 2 Adversarial Attack Detection using Similarity Search and RFF

procedure ATTACKDETECTION(η, $RFVD$, α)
  $flag \leftarrow 0$
  while $(flag \leq \alpha)$ **do**
    Receive signal $RF_{\mathcal{A}}$
    $< v_j >|_{j=1}^{k} \leftarrow ExtractFeatures(RFF_{\mathcal{A}})$
    $Id_i' \leftarrow Querying(RFVD, < v_j >|_{j=1}^{k})$
    if $Id_i' \notin \{Id_i\}_{i=1}^{N}$ **then** ▷ N is total CE devices.
      $flag \leftarrow flag + 1$
    else
      return $exit$ ▷ No attack found.
    end if
  end while
  if $(flag \geq \alpha)$ **then**
    return Attack found. ▷ Raise alarm.
  end if
end procedure

*2) **Attack Mitigation Process using RF-Based AKE**:* The attack mitigation process follows the secure authentication process by the ES and a SD_i by establishing a shared secure session key.

Step 1: SD_i selects a fresh timestamp ts_1, random nonce r_i, $e_i \leftarrow \chi_\gamma$, and computes $p_i = H_1(Sid_i || Id_i || ts_1)$. Next, SD_i calculates $x_i = b \cdot (r_i + p_i) + 2 \cdot e_i$, $x_i^* = x_i + p_i$, $u = h(t_i || Id_i || ts_1) \oplus h(Id_i || Sid_i || ts_1 || p_i)$, and $a_i = h(x_i^* || ts_1 || Sid_i || Id_i || u)$. Next, SD_i generates a message M_1 as $\{a_i,\ ts_1,\ Sid_i,\ x_i^*,\ u\}$ and sends it to the ES via open channel along with its RFF.

Step 2: The ES receives M_1 at ts_1^*, verify its freshness by the condition $|ts_1 - ts_1^*| < \Delta T$, where ΔT is a maximum message delay in the network. If it is satisfied, the ES fetch Id_i corresponding to received Sid_i and calculates $a_i' = h(x_i^* || ts_1 || Sid_i || Id_i || u)$. Next, the ES verifies $a_i = a_i'$, if

it is verified, the ES fetch Id_i corresponding to received Sid_i and executes the Algorithm 3 (DEVICEAUTH(RFF, Id_i, η, $RFVD$)). Algorithm 3 takes input as (RFF, Id_i, η, $RFVD$) and then extracts feature vectors as $< v_j >|_{j=1}^{k}$ $ExtractFeatures(RFF)$. After that, these vectors are compared with the existing database $RFVD$; if it matches, then it returns an identity Id_i' along with matching threshold β. Then the algorithm checks whether this β reaches the pre-defined threshold η and matches Id_i' with the supplied identity Id_i, that is, $(\beta \geq \eta)$ & $(Id_i = Id_i')$. If both conditions are satisfied, the algorithm returns a $true$, which means the authentication is done successfully. Then the database is updated with the latest RFF vector on successful authentication, which can help to maintain the device fingerprints over time.

Algorithm 3 Authentication using Similarity Search

procedure DEVICEAUTH(RFF, Id_i, η, $RFVD$)
  $< v_j >|_{j=1}^{k} \leftarrow ExtractFeatures(RFF)$
  $(Id_i', \beta) \leftarrow Querying(RFVD, < v_j >|_{j=1}^{k})$
  if $(\beta \geq \eta)$ & $(Id_i = Id_i')$ **then**
    $RFVD \leftarrow RFVD \cup < v_j >|_{j=1}^{k}$
    return $true$ ▷ Authentication successful.
  else
    return $false$ ▷ Authentication fail.
  end if
end procedure

After successful authentication, the ES derives $h(t_i||\ Id_i||\ ts_1) = u \oplus h(Id_i\ ||Sid_i\ ||ts_1\ ||p_i)$, $p_i = H_1(Sid_i||\ Id_i||\ ts_1)$, and $x_i = x_i^* - p_i$. Next, the ES selects random samples r_j, $e_j \leftarrow \chi_\gamma$, and timestamp ts_2. Next, the ES computes $x_j = b \cdot (r_j + p_i) + 2 \cdot e_j$, $y_j = x_i \cdot (r_j + p_i)$, $x_j^* = x_j + p_i$, $c_j = Cha(y_j)$, $m_j = Mod_2(y_j, c_j)$, and $c_j^* = c_j \oplus h(t_i||\ Id_i||\ ts_1)$. Next, the ES generates a session key as $SK_j = h(Id_i||\ m_j||\ h(t_i||\ Id_i||\ ts_1)||\ p_i||\ ts_1||\ ts_2)$, picks a new pseudo-identity Sid_n and computes $Sid_i^* = Sid_n \oplus h(t_i||\ Id_i||\ ts_1)$ and a verifier $SKV_j = h(c_j^*||\ SK_j||\ ts_1||\ ts_2||\ x_j^*||\ x_i^*||\ Sid_i^*)$. Next, the ES builds a reply message M_2 as $\{SKV_j, ts_2, x_j^*, Sid_i^*, c_j^*\}$ and sends it to SD_i.

Step 3: After receiving the message M_2 at timestamp ts_2^* from the ES, SD_i verifies its freshness with the condition: $|ts_2^* - ts_2| < \Delta T$. If it satisfied, SD_i derives $x_j = x_j^* - p_i$, computes $y_i = x_j \cdot (r_i + p_i), c_j = c_j^* \oplus h(t_i||\ Id_i||\ ts_1)$, and $m_i = Mod_2(y_i, c_j)$. Next, SD_i generates the session key $SK_i = h(Id_i||\ m_i||\ h(t_i||\ Id_i||\ ts_1)||\ p_i||\ ts_1||\ ts_2)$, derives $Sid_n = Sid_i^* \oplus h(t_i||\ Id_i||\ ts_1)$, and computes verifier as $SKV_i = h(c_j^*||\ SK_i||\ ts_1||\ ts_2||\ x_j^*||\ x_i^*||\ Sid_i^*)$. SD_i then checks whether $SKV_i = SKV_j$. If yes, SD_i updates believes that they successfully generate the same session key and then SD_i updates old Sid_i with new Sid_n. Next, SD_i picks new timestamp ts_3, computes acknowledgment as $Ack = h(SK_i||\ ts_3||\ Sid_n||\ ts_2)$, and sends a final message M_3 $\{Ack, ts_3\}$ to the ES via open channel.

Step 4: The ES receives the message M_3 at timestamp ts_3^* from SD_i and then verifies its freshness by the condition: $|ts_3^* - ts_3| < \Delta T$. If it is verified, the ES computes $Ack' = h(SK_j||\ ts_3||\ Sid_n||\ ts_2)$, verifies $Ack' = Ack$? If so, then the ES also believes that they build a same session key and then updates Sid_i with new Sid_n. A summary of this phase is described in Fig. 2. The overall flowchart of the proposed scheme is presented in Fig. 3.

Smart device as SD_i	**Edge server** as ES
Stored: $\{Sid_i, t_i, Id_i\}$	$\{(Sid_i, Id_i), RFVD, \eta\}$
Select r_i, e_i χ_γ, timestamp ts_1, compute $p_i = H_1(Sid_i||\ Id_i||\ ts_1)$, $x_i = b \cdot (r_i + p_i) + 2 \cdot e_i$, $x_i^* = x_i + p_i$, and $u = h(t_i||\ Id_i||\ ts_1) \oplus h(Id_i\ ||Sid_i\ ||ts_1\ ||p_i)$, $a_i = h(x_i^*||\ ts_1||\ Sid_i||Id_i||u)$, $\{a_i, ts_1, Sid_i, x_i^*, u\}$ →	
	Verify $\lvert ts_1^* - ts_1\rvert < \Delta T$, if yes, fetch Id_i corr. to Sid_i, compute $a_i' = h(x_i^*||\ ts_1||Sid_i||Id_i||u)$, verify $a_i' = a_i$, if verified, check $true$ =Algorithm 3, if so, compute $h(t_i||\ Id_i||\ ts_1) = u \oplus h(Id_i\ ||Sid_i\ ||ts_1\ ||p_i)$, $p_i = H_1(Sid_i||\ Id_i||\ ts_1)$, $x_i = x_i^* - p_i$, select r_j, $e_j \leftarrow \chi_\gamma$, timestamp ts_2, compute $x_j = b \cdot (r_j + p_i) + 2 \cdot e_j$, $y_j = x_i \cdot (r_j + p_i)$, $x_j^* = x_j + p_i$, $c_j = Cha(y_j)$, $m_j = Mod_2(y_j, c_j)$, $c_j^* = c_j \oplus h(t_i||\ Id_i||\ ts_1)$. $SK_j = h(Id_i\ ||m_j||h(t_i||Id_i||ts_1)||p_i||ts_1||ts_2)$. Pick Sid_n, compute $Sid_i^* = Sid_n \oplus h(t_i||\ Id_i||\ ts_1)$, $SKV_j = h(c_j^*||\ SK_j||\ ts_1||ts_2||x_j^*||x_i^*||\ Sid_i^*)$ $\{SKV_j, ts_2, x_j^*, Sid_i^*, c_j^*\}$ ←
Verify $\lvert ts_2^* - ts_2\rvert < \Delta T$, if yes, derive $x_j = x_j^* - p_i$, $y_i = x_j \cdot (r_i + p_i), c_j = c_j^* \oplus h(t_i||\ Id_i||\ ts_1)$, $m_i = Mod_2(y_i, c_j)$, $SK_i = h(Id_i||m_i||h(t_i||Id_i||ts_1)||p_i\ ||ts_1||ts_2)$, $Sid_n = Sid_i^* \oplus h(t_i||\ Id_i||\ ts_1)$, $SKV_i = h(c_j^*||\ SK_i||\ ts_1||\ ts_2||\ x_j^*||\ x_i^*||\ Sid_i^*)$, check $SKV_i = SKV_j$, if valid, update Sid_n pick new timestamp ts_3, and compute $Ack = h(SK_i||\ ts_3||\ Sid_n||\ ts_2)$ $\{Ack, ts_3\}$ →	
	Verify $\lvert ts_3^* - ts_3\rvert < \Delta T$, if yes, $Ack' = h(SK_j||\ ts_3||\ Sid_n||\ ts_2)$, verify $Ack' = Ack$, if yes update Sid_i with new Sid_n

Fig. 2. Message flow and computations in the AKE phase.

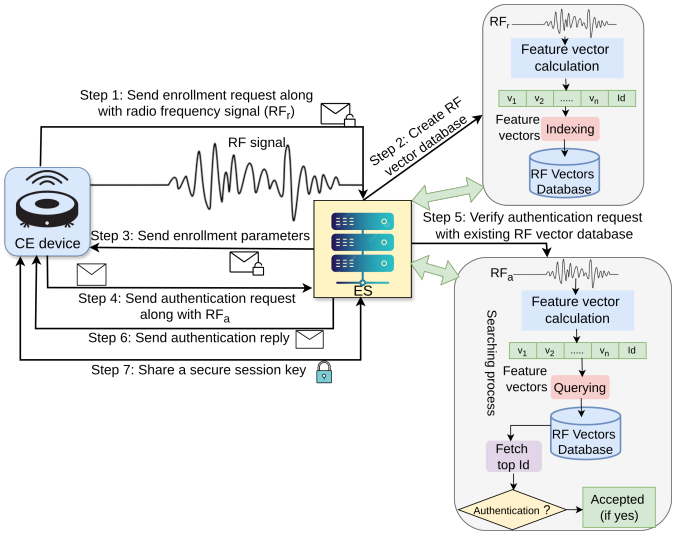


Fig. 3. Overall flowchart of the proposed scheme.

F. Dynamic Device Joining and Database Update Phase

In this phase, the ES registers the new smart devices SD_n with the following process:

- The ES selects a unique and distinct identity Id_n, a pseudo-identity Sid_n, and then calculates $t_n = h(Id_n || mk || RTS_n)$, where RTS_n is the registration timestamp.
- Similarly, the RFFs of a newly joined devices will be updated to the $RFVD$ by the ES.
- Next, the ES shares securely via offline or secure channel the registration information $\{Id_n, Sid_n, t_n\}$ with SD_n, deletes t_n from it's own memory, and stores $\{Id_n, Sid_n, RFVD\}$ into it's own memory.

V. Security Analysis

Here, an informal security analysis and security verification are provided to prove that the proposed scheme is resisting various potential attacks.

A. Informal Security Analysis

1) Replay attack: In this attacks, an adversary $\mathcal{A}$ tries to resend the older messages, which may have been captured previously. In the proposed scheme, we integrated fresh timestamps and random nonce to construct the messages $\{M_1, M_2, M_3\}$, and these parameters are protected with a collision-resistant hash function. Therefore, after receiving these messages, the receiver verifies their freshness before processing them. Even if $\mathcal{A}$ captures the messages, he cannot replay these, and it can be detected easily. Thus the proposed scheme resists replay attack.

2) Man-in-the-Middle (MiTM) attack: In this attack, $\mathcal{A}$ tries to communicate with the receiver ES on behalf of a SD_i on the fly. To do so, $\mathcal{A}$ needs to create a valid message $M_1 = \{a_i, ts_1, Sid_i, x_i^*, u\}$. Constructing this message, $\mathcal{A}$ needs the original value of secrets $\{t_i, Id_i\}$, and these values are utilized with a one-way hash function to build M_1 instead of sharing plain text form. Without having these secrets, $\mathcal{A}$ cannot generate a valid M_1, and to reveal these secrets, $\mathcal{A}$ needs to break the hash function, which is impossible. Thus, the proposed scheme is secure against MiTM attack.

3) Device impersonation attack: In this attacks, $\mathcal{A}$ impersonates a device offline and initiates the communication with ES. To achieve this goal, $\mathcal{A}$ tries to generate a legitimate message M_1. However, it is noticed that to generate the attributes of M_1, it needs the secret values of $\{t_i, Id_i\}$, and these values are utilized with a one-way hash function to protect from $\mathcal{A}$. Therefore, due to the collision-resistant nature of the hash function, $\mathcal{A}$ cannot generate another valid M_1. Thus, $\mathcal{A}$ cannot proceed, and hence, the proposed scheme is secure against this attack.

4) Privileged-insider attack: In this attacks, $\mathcal{A}$ is an insider and tries to hijack all communication by revealing the session key. However, it is noted that, during the registration process, the ES deletes secret information t_i, and therefore, $\mathcal{A}$ cannot retain the long-term secret. However, constructing the session key $\mathcal{A}$ needs to know the secret t_i, and without this value, $\mathcal{A}$ cannot succeed. Thus, the proposed scheme resists the privileged-insider attack.

5) Ephemeral Secret Leakage (ESL) attack: In this attack, $\mathcal{A}$ utilizes CK-adversary threat model capability to hijack a particular session state and then tries to reveal ephemeral secrets and long-term secrets that are utilized in constructing session key $SK_i (= SK_j)$. Therefore, $\mathcal{A}$ can only construct a valid session key for a specific session if and only if he has access to both these secrets. In this proposed scheme, we utilized random nonces for constructing the session key that are unique to each session, ensuring that the session key differs from its subsequent sessions. Consequently, even if $\mathcal{A}$ compromises a session key in a particular session, it cannot reveal the keys for subsequent or previous sessions. Thus, the proposed scheme is secure against ESL attack under the CK-adversary model.

6) Anonymity and untraceability: In this proposed scheme, SD_i communicates with the ES with their pseudo-identity instead of their real one with the public channel, and the real identity is protected by a secure hash function. As a result, $\mathcal{A}$ cannot reveal the real identity from the public channel and is unable to extract the identity from the hashed value. Therefore, the anonymity of SD_i is preserved within the proposed scheme.

In addition, each attribute of the communicated messages is constructed using fresh timestamps and random nonces, which makes the messages random. The temporary identity changes after each successful session. The randomness of the timestamps and nonces ensures that the messages remain unpredictable. As a result, $\mathcal{A}$ cannot trace the messages, and the proposed scheme maintains the untraceability property.

7) Physical capture attack: In this attack, $\mathcal{A}$ captures a device physically and extracts stored information from the compromised device using a power analysis attack. However, it is noted that, in this proposed scheme, the registration credentials are unique and distinct to each device; therefore, if a device is compromised, that should not impact other non-compromised devices. Thus, the proposed scheme resists this attack.

8) Key reuse attacks: In this section, we analyze the security of the proposed `ADMICE` scheme under the key key reuse attacks, which poses significant risks to lattice-based key exchange schemes. The key reuse attack is a combination of two attacks, including the signal leakage attack (SLA) [37] and the key mismatch attack (KMA) [38].

- In the SLA, an adversary $\mathcal{A}$ plays an initiator role and initiates multiple key exchange sessions using a deliberately malformed private key. $\mathcal{A}$ then observes the response signals from the victim to detect any variations.
- Whereas, in a KMA, $\mathcal{A}$ seeks to recover the secret key by sending repeated queries to his oracle that test whether the two parties have derived matching shared keys.

SLA. For this attack simulation, $\mathcal{A}$ impersonates SD_i on the fly and send quires x_j^* to his oracle. Next, his oracle returns a generated public key x_j^* and signal c_j^*. After that, $\mathcal{A}$ can evaluate the correct secret value r_j of ES by counting the signal changes. However, the original public key x_i is hidden with secret functional value p_i, and its masked value is communicated as x_j^* over the public channel. In addition, $\mathcal{A}$ receives the masked signal from the ES as c_j^* and it is

derived as $c_j^* = c_j \oplus h(t_i || Id_i || ts_1)$, which is calculated with the secret shared values. Thus, without having these secrets $\mathcal{A}$ cannot derive the original signal as well as x_j. Hence, the proposed ADMICE scheme is secure against the SLA.

KMA. $\mathcal{A}$ execute this attack when the public key is reused. Here, $\mathcal{A}$ actively plays a role of ES on the fly and his oracle to simulate the SD_i's role in the key exchange process. In this scheme, x_i^* is the public key of SD_i and using this key $\mathcal{A}$ run his oracle in multiple times. It is worth noticing that the original value of the public key x_i is masked with the secret value p_i, where $p_i = H_1(Sid_i || Id_i || ts_1)$. Therefore, with this x_i^* value, $\mathcal{A}$ cannot derive the original x_i without having the value of p_i. Thus, $\mathcal{A}$ cannot launch KMA in the proposed scheme.

B. Formal Security Verification under Scyther Tool

We employ the Scyther tool to verify the security of our proposed scheme. Scyther offers explicit termination for unlimited session and infinite state aggregation protocols, along with support for multi-protocol parallel analysis. Security protocols are modeled using the security protocol description language (.spdl). Scyther incorporates predefined security models such as the DY threat model, CK-adversary, eCK-adversary and others, alleviating the need for users to formalize adversary powers [39]. It provides a range of claims to test various security goals, including secrecy and multiple authentication aspects like aliveness, weak agreement, agreement, and synchronization. The secret claim ensures state confidentiality. Different levels of authentication strength are ensured through various authentication claims like Alive, Niagree, and Nisynch, which help to detect replay, reflection, and man-in-the-middle attacks. Alive ensures that all events are carried out by the communicating parties, and Nisynch ensures that all messages are sent by the sender and received by the recipient. Weakagree ensures that the protocol remains resilient against impersonation attacks. Details can be found in Scyther manual [40].

Scyther results : verify

Claim				Status		Commer
ADMICE	DeviceSD	ADMICE,DeviceSD1	Alive	Ok	Verified	No attacks.
		ADMICE,DeviceSD2	Nisynch	Ok	Verified	No attacks.
		ADMICE,DeviceSD3	Niagree	Ok	Verified	No attacks.
		ADMICE,DeviceSD4	Weakagree	Ok	Verified	No attacks.
		ADMICE,DeviceSD5	Secret ti	Ok	Verified	No attacks.
		ADMICE,DeviceSD6	Secret idi	Ok	Verified	No attacks.
	ServerES	ADMICE,ServerES1	Alive	Ok	Verified	No attacks.
		ADMICE,ServerES2	Secret idi	Ok	Verified	No attacks.
		ADMICE,ServerES3	Nisynch	Ok	Verified	No attacks.
		ADMICE,ServerES4	Niagree	Ok	Verified	No attacks.
		ADMICE,ServerES5	Weakagree	Ok	Verified	No attacks.

Done.

Fig. 4. Simulation results using Scyther tool.

The results of this simulation are presented in Fig. 4. Here, we defined two roles: one is for the CE device (SD), named as **DeviceSD**, and the other is for the edge server (ES), called **ServerES**. The findings indicate that the Scyther did not detect any vulnerabilities within the proposed scheme, as the status of the results shows **ok**. We have considered the verification parameters as 5 for the maximum number of runs and typed matching method for matching type. In advance parameter setting, we have assumed "Find best attack" as searching pruning, 10 as the maximum number of patterns per claim, and null as additional backed parameters.

VI. TestBed Experimental Setup, Results, and Implementation

In this section, we conducted a testbed experiment for computing execution times for cryptographic primitives and we utilize cryptographic library cryptography 37.0.2 for this testbed. We used python code for script writing and the experiment was conducted under two cases, such as case 1: we considered ES as a laptop configuration with "Ubuntu 22.04 LTS, featuring 16 GB of 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"; case 2: a Raspberry Pi 4 Model B is considered as V_i 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".

Let T_h, T_{senc}/T_{sdec}, and T_{eca}/T_{ecm} represents the required execution time (in milliseconds) for "one-way hash function using Secure Hash Algorithm (SHA-256) algorithm", "Advanced Encryption Standard (AES-128) encryption and decryption", and "elliptic curve point addition and multiplication", here we consider a non-singular elliptic curve, namely secp256r1 of the form: $y^2 = x^3 + ax + b \pmod q$. In addition, for lattice-based primitives, such as T_g, T_{sm}, T_{pm}, T_{pa}, and T_{cha} represents the time for sampling from X_γ, a "component-wise multiplication with a scalar in $\mathbf{R}_p$", "a component-wise polynomial multiplication in $\mathbf{R}_p$", "a component-wise polynomial addition in $\mathbf{R}_p$", and "the characteristic function in $\mathbf{R}_p$", respectively. We run the execution for 1,000 times for each primitives and we considered only the average time for each. The testbed result is shown in Table II. We have used the NIST standard parameters for computing operation on $\mathbf{R}_p$ and operation for module learning with errors (MLWE) problem as $n = 256$, $q = 3329$, $k = 3$, $\gamma = 3$.

VII. Performance Analysis

In this section, a comprehensive comparative analysis is conducted with the existing schemes as follows: LPQE [26], QSSA [27], QSLP [28], TPPA [29], PQST [30], and MPKE [31]. We note that the scheme names are used for representation purpose only. The work LPQE [26] proposed RLWE-based AKE protocol for medical IoT applications. QSSA [27] suggested RLWE-based AKE model for IoD applications, and QSLP [28] designed similar RLWE-based AKE protocol for mobile devices. The works, PQST [30], TPPA [29], and MPKE [31] suggested MLWE-based AKE protocols. We name the existing approaches as LPQE [26]. Furthermore, we call the

TABLE II
AVERAGE EXECUTION TIMES AND ENERGY CONSUMPTION OF CRYPTOGRAPHIC PRIMITIVES

Operation	Execution cost (in ms)		Energy consumption cost (in mJ)	
	Smart device	Server	Smart device	Server
T_h	0.1733	0.0199	0.0926	0.0089
T_{senc}	0.0819	0.0071	0.0731	0.0054
T_{sdec}	0.0819	0.0072	0.0743	0.0055
T_{ecm}	2.5892	0.5047	4.8841	1.0576
T_{eca}	0.3616	0.0751	0.6922	0.1495
T_g	0.0310	0.0069	0.0627	0.0142
T_{sm}	0.0211	0.0047	0.0423	0.0099
T_{pm}	3.1726	0.2855	6.4400	0.5686
T_{pa}	0.0745	0.0069	0.1446	0.0139
T_{cha}	0.6008	0.0695	1.1445	0.1389
T_{mlwe}	24.0011	2.5300	48.6979	5.0579

proposed scheme as ADMICE. Henceforth, we use ADMICE to represent the proposed scheme.

The performance analysis is measured in terms of computation costs, communication costs, security features, and performance under unknown attacks.

A. Communication Costs Comparison

Here, we consider the sizes of random numbers, timestamps, identities, hash output, elements in $\mathbf{R}_p$, and signal function $Cha(\cdot)$ to be 160 bits, 32 bits, 160 bits, 256 bits, 4096 bits, and 1 bit for comparison fairness, respectively. To calculate communication costs of the proposed scheme, we consider the AKE phase described in Section IV-E, where three messages are exchanged to establish a secure session key, and these are $M_1 = \{a_i, ts_1, Sid_i, x_i^* u\}$, $M_2 = \{SKV_j, ts_2, x_j^*, Sid_i^*, c_j^*\}$, and $M_3 = \{Ack, ts_3\}$. M_1 requires (256 + 32 + 160 + 4096 + 256) = 4800 bits, M_2 needs (256 + 32 + 4096 + 256 + 256) = 4896 bits, and M_3 requires (256 + 32) = 288 bits, respectively. Thus, the proposed scheme requires a total of 9984 bits. The comparison analysis is presented in Table III, and the scalability is shown in Fig. 5, which shows that the proposed scheme requires lower communication costs compared to the existing schemes. Table III indicates that the proposed ADMICE scheme significantly reduces the communication cost for smart device compared to the existing schemes. In particular, the proposed scheme achieves communication cost advantages of 45.17%, 28.78%, 46.40%, 64.18%, 52.73%, and 64.66% compared to the LPQE [26], QSSA [27], QSLP [28], TPPA [29], PQST [30], and MPKE [31] schemes, respectively. These results clearly demonstrate that the proposed scheme achieves a significant reduction in communication overhead while requiring fewer message exchanges.

B. Computation Costs Comparison

For computation cost calculation, we consider the testbed experiment results described in Section VI for the execution times of various cryptographic primitives. In the proposed scheme, CE device SD_i requires computation costs of $7T_h + 4T_g + T_{sm} + 2T_{pm} + 4T_{pa} \approx 8.002$ ms and ES needs costs of $7T_h + 3T_g + T_{sm} + 2T_{pm} + 3T_{pa} + T_{cha} \approx 0.81984$ ms. Table IV shows that ADMICE incurs the lowest computation cost among all the compared schemes. This highlights the efficiency of our approach. The scalability in terms of computation costs is also presented in Fig. 6, which indicates that the proposed scheme is more scalable compared to others. The Table IV highlights that the proposed ADMICE scheme achieves a significant reduction in computation overhead for smart device compared to the other existing schemes. In particular, the proposed scheme achieves computation cost advantages of 44%, 57.5%, 46.6%, 87.2%, 85.7%, and 74.1% compared to the LPQE [26], QSSA [27], QSLP [28], TPPA [29], PQST [30], and MPKE [31] schemes, respectively.

TABLE III
COMPARATIVE ANALYSIS ON COMMUNICATION COSTS

Scheme	No. of messages	Total cost (in bits)	Advantage (%)
LPQE [26]	4	18210	45.17
QSSA [27]	3	14018	28.78
QSLP [28]	4	18626	46.40
TPPA [29]	4	27874	64.18
PQST [30]	4	21122	52.73
MPKE [31]	4	28252	64.66
ADMICE (Proposed)	3	9984	0

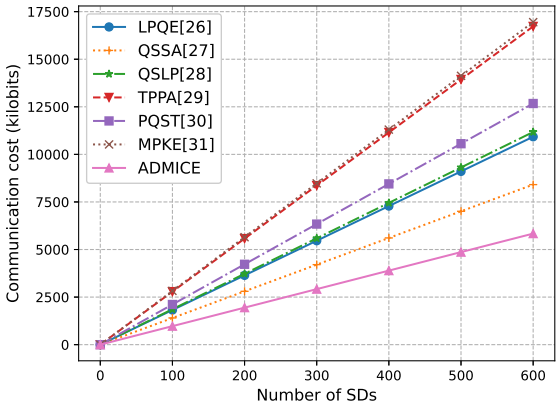


Fig. 5. Communication costs vs number of SDs.

TABLE IV
COMPARATIVE ANALYSIS ON COMPUTATION COSTS

Scheme	SD/smart device	Advantage (%)	Server
LPQE [26]	$4T_h + 4T_g + 2T_{sm} + 4T_{pm} + 2T_{pa} + T_{cha}$ ≈ 14.3005 ms	≈ 44	$5T_h$ ≈ 0.0998 ms
QSSA [27]	$8T_h + 4T_g + 2T_{sm} + 3T_{pm} + 2T_{pa} + 2T_{cha}$ ≈ 18.7672 ms	≈ 57.5	$6T_h + T_{pm}$ ≈ 0.4054 ms
QSLP [28]	$8T_h + 4T_g + 2T_{sm} + 4T_{pm} + 2T_{pa} + T_{cha}$ ≈ 14.9936 ms	≈ 46.6	$6T_h$ ≈ 0.1198 ms
TPPA [29]	$8T_h + 5T_g + 4T_{pm} + 2T_{pa} + T_{senc} + T_{sdec} + 2T_{mlwe}$ ≈ 62.5475 ms	≈ 87.2	$4T_h + 5T_g + 2T_{pm} + 2T_{pa} + T_{senc} + 2T_{mlwe}$ ≈ 5.7669 ms
PQST [30]	$8T_h + 5T_g + 2T_{pm} + 2T_{pa} + 2T_{mlwe}$ ≈ 56.0384 ms	≈ 85.7	$6T_h$ ≈ 0.1198 ms
MPKE [31]	$3T_h + 2T_g + 2T_{pm} + 2T_{mlwe}$ ≈ 30.9284 ms	≈ 74.1	$4T_h + 2T_g + 4T_{pm} + 3T_{pa} + T_{mlwe}$ ≈ 3.7868 ms
ADMICE (Proposed)	$7T_h + 4T_g + T_{sm} + 2T_{pm} + 4T_{pa}$ ≈ 8.002 ms	–	$7T_h + 3T_g + T_{sm} + 2T_{pm} + 3T_{pa} + T_{cha}$ ≈ 0.81984 ms

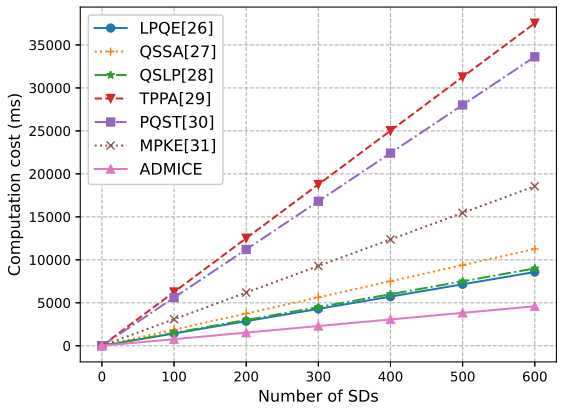


Fig. 6. Computation costs vs number of SDs.

C. Storage Costs and Authentication Latency for Smart Device Comparison

In this section, we calculate the storage costs for the smart device, and it is calculated based on the parameters stored in the smart device's memory. The parameters' sizes are calculated based on the similar assumption as mentioned in Section VII-A. We also calculated the authentication time (in ms) based on the time required for authenticating the smart device to the corresponding server/gateway node. Table V highlighted the comparison of the storage costs and authentication latency for smart devices, and it is noted that the proposed scheme ADMICE requires lower storage costs compared to all other existing schemes. Moreover, the proposed scheme ADMICE incurs lower authentication latency compared to the scheme of QSSA [27], TPPA [29], PQST [30], and MPKE [31]. Although the proposed scheme ADMICE takes slightly more authentication time compared to the schemes of LPQE [26] and QSLP [28], these schemes do not fulfill all the security requirements, and they need huge operational costs in total.

TABLE V
COMPARATIVE ANALYSIS ON STORAGE COSTS AND AUTHENTICATION TIME

Scheme	Storage (in bits)	Authentication time (in ms)
LPQE [26]	928	3.5238
QSSA [27]	928	7.6028
QSLP [28]	768	3.5637
TPPA [29]	768	26.7044
PQST [30]	768	24.243
MPKE [31]	672	24.3111
ADMICE (Proposed)	576	4.0505

D. Energy Consumption Costs Analysis

In this section, we calculate the energy consumption cost of the proposed scheme along with compared schemes. It is evaluated based on the required energy cost in millijoule (mJ) for executing the successful authentication process described in Fig. 2, which means the energy required for executing cryptographic primitives required for the authentication. We calculate the energy consumption costs based on our testbed experiment in Section VI. Table VI shows the results and it is worth noticing that the proposed scheme requires 3830.2145 mJ for smart device and 88.7945 mJ for server. This costs is calculated as $7T_h + 4T_g + T_{sm} + 2T_{pm} + 4T_{pa} + M_c \times E \approx 7 \times 0.0926 + 4 \times 0.0627 + 2 \times 6.4400 + 4 \times 0.1446 + 5088 \times 0.7500 \approx 3830.2145$ mJ. Similarly, we calculate the energy consumption costs for server as 88.7945 mJ. Other schemes also calculated similar way and it can be found in Table VI. Here, $E = 0.7500$ mJ is an energy consumption costs for smart device requires for transmitting a single bit with a transmission rate of 1 Mbps operating at 5.1 V and 3.5 A. For the server, powered at 250V and 3A, transmitting 1 bit at the same rate, the energy consumed is approximately 0.01785 mJ. M_c is the total message transferred by the smart device. The results shows that the proposed ADMICE scheme requires lower energy consumption costs compared to other existing scheme and shows the efficiency in the real-world applications.

TABLE VI
COMPARATIVE ANALYSIS ON ENERGY CONSUMPTION COST (IN MJ)

Schemes	Energy consumption cost (in mJ)	
	Smart device	Server
LPQE [26]	10107.8995	231.3805
QSSA [27]	6829.3544	89.1580
QSLP [28]	13613.7699	9.1926
TPPA [29]	10132.6467	270.7533
PQST [30]	8415.6193	179.4459
MPKE [31]	3373.9811	432.9106
ADMICE (Proposed)	3830.2145	88.7945

E. Security and Functionality Attributes Comparison

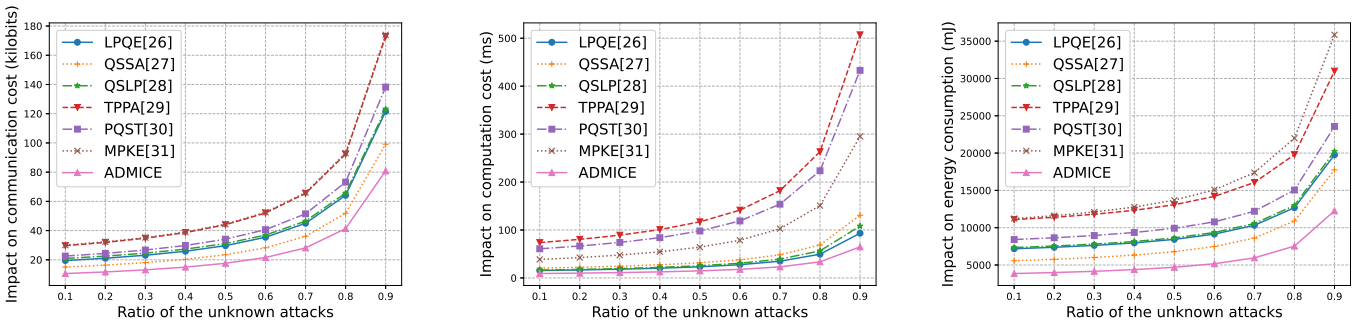
The security and functionality features comparison is presented in Table VII, which highlighted that the proposed scheme satisfies all security and functional attributes. Notably, none of the compared schemes provided physical-layer attack detection (via RFF) or quantum resilience together which is a gap our work addresses.

F. Performance under Unknown Attacks Comparison

In Section V-A, we have discussed that the proposed scheme resists various classical-quantum attacks; however, we consider scenarios with unpredictable, unknown attacks (i.e., attack vectors not specifically addressed by our design) occurring during the protocol execution. This performance is measured on the communication costs, computation costs, and power consumption metrics, and we follow the similar approach as described in [41], [42] to evaluate it. For power consumption evaluation, we consider that for the CE device SD, powered at 5.1V and 3.5A, transmitting 1 bit at a rate of 1 Mbps, the power requirement (Pow_{sd}) is approximately 0.750000 mJ (millijoule). For the server ES, powered at 250V and 3A, transmitting 1 bit at the same rate, the power consumption (Pow_{es}) is approximately 0.01785 mJ.

$$\mathcal{C}_{avg} = \frac{\mathcal{C}_f \times Pr_f + \mathcal{C}_s \times Pr_s}{Pr_s}, \quad (1)$$

$$\mathcal{C}_f = \sum^{N} \frac{\mathcal{C}_i}{N}. \quad (2)$$



(a) Impact on communication cost (b) Impact on computation cost (c) Impact on energy cost

Fig. 7. Performance under the unknown attacks.

TABLE VII
COMPARATIVE ANALYSIS ON VARIOUS FS ATTRIBUTES

Attribute (FS)	[31]	[29]	[30]	[28]	[27]	[26]	ADMICE
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}	NA	NA	NA	×	×	×	✓
FS_{14}	×	×	×	×	×	×	✓

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 and untraceability; FS_9: Privileged-insider attack; FS_{10}: Node addition phase; FS_{11}: Quantum attack; FS_{12}: Security verification under Scyther/AVISPA/ProVerif; FS_{13}: Key reuse attacks; FS_{14}: Attack detection framework.
✓: A scheme is secure or it supports an attribute; ×: A scheme is insecure or it does not support an attribute; N/A: means Not applicable in a scheme.

In (1), C_{avg} denotes the average communication/computation/energy consumption overhead required during unknown attacks, $\mathcal{C}_f$ represents the overhead for an unsuccessful authentication under unknown attack, and $\mathcal{C}_s$ denotes the overhead for a successful authentication. Additionally, Pr_f represents the probability that an unknown attack occurs while the protocol is being executed and Pr_s is probability of success (i.e., no unknown attack), where $Pr_s = 1 - Pr_f$. We assume that the total number of messages in the authentication process is N and that there is a $\frac{1}{N}$ chance that an unknown attack will occur at step i. $\mathcal{C}_i$ indicates the total overhead prior to the occurrence of an unknown attack at step i, and $\mathcal{C}_f$ can be obtained from (2). The results of this analysis are shown in Fig. Fig. 7. Figures 7(a) and 7(b) present the performance on communication and computation costs under the unknown attacks. Whereas Fig. 7(c) shows the performance on power usage under the unknown attacks. However, the proposed scheme's overhead rises more slowly than others'. For instance, even if an attack happens at the worst possible moment, the extra communication cost for our scheme is lower compared to exist schemes. It demonstrates our framework's robustness and efficiency even against unforeseen attack attempts.

G. Network Performance Comparison using NS-3

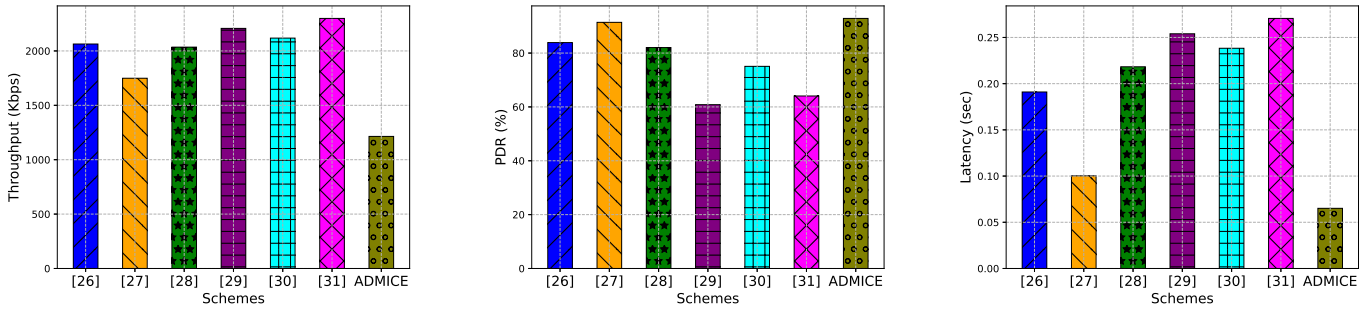
In this section, we measure the network performance metrics of the proposed scheme in terms of throughput, packet delivery ratio (PDR), and latency or end-to-end (E2E) delay using network simulation 3 (NS-3). The simulation was performed under the same configuration described in Section VI, where we considered the simulation time is 2500 seconds and network covers an area of $100m \times 100m$. We considered the network node as 7 for CEs devices and one edge server, routing protocol is OLSR, and the MAC protocol is IEEE 802.11b. These metrics are measured based on the communication messages in the proposed scheme, that is 9728 bits.

1) Throughput measurement: Throughput (Th) is measured based on the amount of data transmitted per second. Figure 8(a) shows the throughput of the proposed scheme, and it is 1214.82 Kbps. The other schemes are slightly better than the proposed scheme, but they lack security features and are vulnerable to various attacks.

2) PDR (%) measurement: The packet delivery ratio (PDR) represents the proportion of packets successfully received to those sent. Figure 8(b) shows the PDR of the proposed scheme is approximately 93%. Notably, the proposed scheme achieves a higher PDR than all other schemes.

3) Latency measurement: The latency refers to the total time required to transmit packets from the source node to the destination node. Figure 8(c) highlights that the proposed scheme requires lower latency compared to existing schemes, and it is approximately 0.065127 seconds. It is important to note that the proposed scheme outperforms other schemes.

The NS-3 simulations demonstrate that the proposed scheme do not significantly degrade network performance. The throughput remains above 1 Mbps, latency is actually improved, which making it practical for real-time IoT/CE applications.



(a) Throughput (b) PDR (c) Latency

Fig. 8. Network performance comparison results using NS-3.

VIII. Conclusion

The VSS technique is utilized to detect adversarial attacks and unauthorized access using RFF. A quantum-secure authentication and key agreement protocol is constructed for mitigate such attack. The security analysis demonstrated the proposed scheme is resistant to various attacks, including replay, man-in-the-middle, and insider attacks, among other threats. A comprehensive comparative analysis and performance analysis under unknown attacks highlighted the scalability and efficiency of the proposed scheme. Formal security analysis under the Scyther tool has shown its robustness, and a testbed experiment using Raspberry Pi devices validated its practicability in real CE devices. The NS-3 simulations confirmed that our proposed protocol do not compromise network performance and highlighted the novelty and efficiency.

In the future, we plan to demonstrate the resistance of the proposed scheme against real-time attack scenarios using a benchmark model, where an attacker may be present in the network and utilize a generative AI model, like a generative adversarial network (GAN), to manipulate the fake signal in the network.

References

[1] G.-Y. Yang, F. Wang, and K.-H. Yeh, "GNN-enhanced Traffic Anomaly Detection for Next-Generation SDN-Enabled Consumer Electronics," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025.

[2] Y. Li, J. Shen, P. Vijayakumar, C.-F. Lai, A. Sivaraman, and P. K. Sharma, "Next-Generation Consumer Electronics Data Auditing Scheme Toward Cloud-Edge Distributed and Resilient Machine Learning," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2244–2256, 2024.

[3] C. K. Wu, C.-T. Cheng, Y. Uwate, G. Chen, S. Mumtaz, and K. F. Tsang, "State-of-the-Art and Research Opportunities for Next-Generation Consumer Electronics," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 937–948, 2023.

[4] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, "De-Pois: An Attack-Agnostic Defense against Data Poisoning Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3412–3425, 2021.

[5] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, "Data Poisoning Attacks on Federated Machine Learning," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 365–11 375, 2022.

[6] B. Mao, J. Liu, Y. Wu, and N. Kato, "Security and Privacy on 6G Network Edge: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1095–1127, 2023.

[7] M. Sookhak, H. Tang, Y. He, and F. R. Yu, "Security and Privacy of Smart Cities: A Survey, Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1718–1743, 2019.

[8] L. M. Alkwai and K. Yadav, "Blockchain-Based Secure 5G/6G Communication for Internet of Things Devices in Consumer Electronic Systems," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 6327–6338, 2024.

[9] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, and J. Liu, "Remotely Access "My" Smart Home in Private: An Anti-Tracking Authentication and Key Agreement Scheme," *IEEE Access*, vol. 7, pp. 41 835–41 851, 2019.

[10] S. K. Dwivedi, M. Abdussami, R. Amin, and M. K. Khan, "D^{3}APTS: Design of ECC-Based Authentication Protocol and Data Storage for Tactile Internet Enabled IoD System With Blockchain," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4239–4248, 2024.

[11] F. Wang, J. Cui, Q. Zhang, D. He, C. Gu, and H. Zhong, "Blockchain-Based Lightweight Message Authentication for Edge-Assisted Cross-Domain Industrial Internet of Things," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 4, pp. 1587–1604, 2024.

[12] K. K. Saha, S. Ray, and M. Dasgupta, "ECMHP: ECC-Based Secure Handshake Protocol for Multicasting in CCN-IoT Environment," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5826–5842, 2024.

[13] W. Wang, Q. Xie, Y. Huang, Y. Ding, L. Zhang, D. Gao, C. Su, and J. J. P. C. Rodrigues, "Attack Analysis and Enhanced Authentication Protocol Design for Vehicle Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 6, pp. 6943–6954, 2025.

[14] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[15] 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.

[16] T. N. Turnip, B. Andersen, and C. Vargas-Rosales, "Towards 6G Authentication and Key Agreement Protocol: A Survey on Hybrid Post Quantum Cryptography," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025, doi: 10.1109/COMST.2025.3567439.

[17] F. Zhou, L. Zhang, Z. Yang, and L. Feng, "Radio Frequency-Enhanced Multi-Factor IoT Device Authentication via Swarm Learning," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 4, pp. 2487–2499, 2025.

[18] G. Shen, J. Zhang, A. Marshall, R. Woods, J. Cavallaro, and L. Chen, "Towards Receiver-Agnostic and Collaborative Radio Frequency Fingerprint Identification," *IEEE Transactions on Mobile Computing*, vol. 23, no. 7, pp. 7618–7634, 2024.

[19] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards Scalable and Channel-Robust Radio Frequency Fingerprint Identification for LoRa," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 774–787, 2022.

[20] Y. Zeng, Y. Gong, J. Liu, S. Lin, Z. Han, R. Cao, K. Huang, and K. B. Letaief, "Multi-Channel Attentive Feature Fusion for Radio Frequency Fingerprinting," *IEEE Transactions on Wireless Communications*, vol. 23, no. 5, pp. 4243–4254, 2024.

[21] K. Echihabi, K. Zoumpatianos, and T. Palpanas, "New Trends in High-D Vector Similarity Search: AI-Driven, Progressive, and Distributed," *Proceedings of the VLDB Endowment*, vol. 14, no. 12, pp. 3198–3201, 2021.

[22] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen, "Authenticated key exchange from ideal lattices," in *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Sofia, Bulgaria, 2015, pp. 719–751.

[23] S. S. N, K. Jain, P. Krishnan, P. P, and P. Ganeshkumar, "Embedded SIM (eSIM) Based PQC-Enabled Authentication and Provisioning Techniques for Enhancing Security in Edge-Based IoT Ecosystem," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025, doi: 10.1109/TCE.2025.3609084.

[24] J. Yang, V. Govindarajan, X. Xu, M. A. Khan, Z. A. Shaikh, S. Ayouni, M. Shabaz, T. R. Gadekallu, and L. Y. Por, "Enhancing Cryptographic Security in Smart Consumer Electronics with a Hybrid Classical-Post-Quantum Framework," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025, 10.1109/TCE.2025.3603827.

[25] A. Shahidinejad, J. Abawajy, and S. Huda, "Quantum-Proof Anonymous Key Exchange With Perfect Forward Secrecy for Mobile Devices," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025, 10.1109/TCE.2025.3582742.

[26] 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.

[27] D. Mishra, M. Singh, P. Rewal, K. Pursharthi, N. Kumar, A. Barnawi, and R. S. Rathore, "Quantum-Safe Secure and Authorized Communication Protocol for Internet of Drones," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 16 499–16 507, 2023.

[28] P. Rewal, M. Singh, D. Mishra, K. Pursharthi, and A. Mishra, "Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices," *Journal of Information Security and Applications*, vol. 75, p. 103505, 2023.

[29] 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.

[30] 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.

[31] 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.

[32] N. K. Gunda, M. E. Seno, B. K. Singh, K. S. Yogi, I. B. Mohamed, A. Dhibi, K. B. V. B. Rao, S. Kiyosov, and S. Gupta, "Digital Forensic Authentication and Key Agreement Protocol for Biometric-based Consumer Electronics Devices," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025, doi: 10.1109/TCE.2025.3587082.

[33] 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.

[34] 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.

[35] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[36] 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.

[37] J. Ding, S. Alsayigh, R. V. Saraswathy, S. Fluhrer, and X. Lin, "Leakage of signal function with reused keys in RLWE key exchange," in *International Conference on Communications (ICC'17)*, Paris, France, 2017, pp. 1–6.

[38] Y. Qin, C. Cheng, X. Zhang, Y. Pan, L. Hu, and J. Ding, "A Systematic Approach and Analysis of Key Mismatch Attacks on Lattice-Based NIST Candidate KEMs," in *Advances in Cryptology – ASIACRYPT 2021*, Singapore, 2021, pp. 92–121.

[39] 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.

[40] 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.

[41] R. Ma, J. Cao, D. Feng, H. Li, and S. He, "FTGPHA: Fixed-Trajectory Group Pre-Handover Authentication Mechanism for Mobile Relays in 5G High-Speed Rail Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2126–2140, 2020.

[42] B. Bera, R. P. Parameswarath, and B. Sikdar, "Fortifying the Security of Smart Grid Networks with Post-Quantum Communication," *IEEE Transactions on Smart Grid*, vol. 16, no. 6, pp. 5430–5445, 2025.