

A provably secure authentication framework for quantum-safe message transmission in wireless sensor networks

Received: 5 April 2026

Accepted: 24 August 2026

Published online: 31 August 2026

Cite this article as: Abduljabbar Z.A., Hussien Z.A., Nyangaresi V.O. *et al.* A provably secure authentication framework for quantum-safe message transmission in wireless sensor networks. *Sci Rep* (2026). <https://doi.org/10.1038/s41598-026-68879-8>

Zaid Ameen Abduljabbar, Zaid Alaa Hussien, Vincent Omollo Nyangaresi, Ali Hasan Ali, Mustafa A. Al Sibahee, Mohammed Abdulridha Hussain, Iman Qays Abduljaleel, Junchao Ma, Abdulla J. Y. Aldarwish, Dhafer G. Honi & Maryam Mahdi Alhusseini

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

A Provably Secure Authentication Framework for Quantum-Safe Message Transmission in Wireless Sensor Networks

Zaid Ameen Abduljabbar^{1,2}, Zaid Alaa Hussien³, Vincent Omollo Nyangaresi^{4,5}, Ali Hasan Ali^{6,7,8,*}, Mustafa A. Al Sibah^{9,10}, Mohammed Abdulridha Hussain¹, Iman Qays Abduljaleel¹, Junchao Ma¹¹, Abdulla J. Y. Aldarwish¹, Dhafer G. Honi^{1,12}, Maryam Mahdi Alhusseini¹³

¹Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

² Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 518000, China

³Department of Management Information Systems, College of Administration and Economics, University of Basrah, Basrah 61004, Iraq.

⁴Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science & Technology, Bondo 40601, Kenya

⁵Department of Applied Electronics, Saveetha School of Engineering, SIMATS, Chennai, Tamil Nadu 602105, India

⁶Institute of Mathematics, University of Debrecen, Pf. 400, Debrecen, 4002, Hungary

⁷Department of Mathematics, College of Education for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

⁸Technical Engineering College, Al-Ayen University, Thi-Qar, 64001, Iraq

⁹Department of Management and Marketing, College of Industrial Management for Oil and Gas, Basrah University for Oil and Gas, Basrah, 61004, Iraq

¹⁰National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, 518060, China

¹¹College of Big Data and Internet, Shenzhen Technology University, Shenzhen, 518118, China

¹²Doctoral School of Informatics, University of Debrecen, Debrecen, H-4032, Hungary

¹³Polytechnic College of Engineering – Baghdad, Middle Technical University, Baghdad, Iraq

*Correspondence: ali.hasan@science.unideb.hu

Abstract

Wireless Sensor Networks (WSNs) have been heavily deployed to facilitate continuous monitoring of their environments as they collect the necessary data which enable autonomous decision making or predictive control. As such, these networks have found applications in fields such as smart agriculture, military, industrial automation, and healthcare. During the message transfers among sensors or between sensors and mobile sinks, public communication channels are deployed. This exposes the collected data to a myriad of security threats such as tampering, eavesdropping, and replay. Although the recent past has witnessed the development of numerous security frameworks for this ecosystem, most of the current solutions are defenseless against threats such as session hijacking, side-channeling, ephemeral secret leakages and quantum attacks. In this paper, we leverage on lattice cryptography to develop an effective authentication framework for the WSNs. Formal security analysis confirms the robustness of our protocol under the Random Oracle Model (ROM). In addition, extensive semantic security analyses demonstrate the resilience of the proposed framework against typical attacks such as denial of service, key compromise and forgery. In terms of performance, we show that the developed framework incurs slightly lower computation and energy overheads when compared with its peers.

Keywords: Authentication, quantum, lattice, RLWE, WSN, security, privacy.

1. Introduction

Wireless Sensor Networks (WSNs) is the most critical building block of Internet of Things (IoT) that is comprised of distributed sensor nodes. These sensors collect a variety of data from their environment and transmit it to the sink, where it is processed [1]. In addition, sensors can exchange data amongst themselves or with other network elements over wireless communication channels in a multi-hop manner. As explained in [2], IoT and WSNs have facilitated the continuous monitoring of their vicinities and collecting data necessary for autonomous decision making or predictive control. As such, these WSNs have been heavily deployed in environments that require remote control, real-time monitoring and data collection. Such application domains include meteorology, commerce, agriculture, healthcare, smart cities, military and industry [3], [4], [5]. For instance, in smart manufacturing, WSNs facilitate the constant monitoring and effective control of various resources.

On the other hand, the integration of sensor networks and Digital Twins (DTs) can result in optimized resource usage, enhanced operational efficiency, as well as superior predictive maintenance [6], [7].

Although sensor networks have yielded promising results in various fields such as medical services, smart agriculture, smart cities and industrial automation, they are plagued by numerous security, performance and privacy issues. One of the main contributing factor to WSNs vulnerabilities is the exchange of data across open wireless communication channels [8]-[11]. As such, these networks are susceptible to security threats such as impersonation, Denial of Service (DoS), replay, eavesdropping, tampering, node capture, and Man-in-the-Middle attacks [4]. The proliferation of these sensor networks in diverse application domains have served to increase the surface area for adversarial compromise [12]. These security threats result in varied repercussions, such as reduced throughput, large network latencies, and high energy consumptions [5].

To counter the security threats above, robust security mechanisms must be put in place. Such security techniques might include intrusion detection systems [13] or cryptographic approaches such as authentication and encryption. Unfortunately, the WSN ecosystem is characterized by tiny sensor devices that are mostly battery powered. As such, these sensors cannot handle sophisticated cryptographic operations that could potentially offer robust security protection. The energy, computation and communication limitations of sensory devices call for lightweight security solutions that could prolong network functionality [4], [14].

1.1 Motivation

Wireless sensor networks have been deployed in highly sensitive application domains such as healthcare and military. In these application domains, any successful attack or leakage of sensitive data can have serious repercussions. As such, numerous authentication schemes have been put forward to offer security protection for the exchanged messages. Nevertheless, majority of these authentication protocols are still defenseless against numerous security threats such as quantum attacks. In addition, some of them incur high computation and communication overheads which drain sensor battery power. Since most sensors are deployed in hard-to-reach or hazardous locations, frequent battery replacement is problematic. As such, there is need for the development of not only robust authentication protocols but also efficient security solutions for wireless sensor networks.

1.2 Contribution

- We leverage on lattice cryptography's Ring Learning With Error (RLWE) to develop an authentication framework that can defend some of the most recent security threats such as quantum attacks. This is unlike most of the conventional authentication approaches based on techniques such as pure elliptic curve cryptography.
- Session-specific ephemerals and pseudonyms are incorporated in the publicly exchanged messages to support untraceability, confidentiality, perfect key secrecy and anonymity. These are security characteristics that majority of the wireless sensor network authentication approaches rarely consider.
- The negotiated session key comprises of mutually derived security tokens by the network participants. As such, no single network entity can solely derive the session key in the proposed framework. This effectively mitigates against single point of failure risks which are common in traditional authentication approaches that deploy verification tables or centralized entities such as trusted servers for the derivation of session keys.
- During the keying procedures, we incorporate stochastic values obtained via sampling from the discrete Gaussian distribution. This is shown to resist security threats such as known key attacks, owing to the random nature of deployed sampling.
- Extensive formal and informal security analyses are executed, with the outcome showing the robustness of the proposed framework under the ROM and Extended Canetti-Krawczyk (eCK) attack model. As shown in sub-section 5.5, some of the security attacks such as side-channeling, session hijacking, and DoS are rarely considered in majority of the mutual authentication techniques.
- We carry out performance evaluations which demonstrate that the proposed protocol offers robust protection at slightly lower communication and computation overheads.

The rest of this paper is structured as follows: Section 2 presents the related works while Section 3 details the procedures for the proposed authentication framework. On the other hand, Section 4 presents the security analyzes of our framework, while Section 5 details its comparative performance evaluation. Towards the end of this paper, section 6 provides the conclusion and future research directions.

2. Related Works

The recent past has seen the development of numerous security solutions for the WSN ecosystem. For instance, an authentication protocol for safe communication in cloud-based WSN is developed in [15]. However, this scheme fails to preserve user untraceability and is defenseless against threats such as DoS, session key disclosure, sensor node capture and user impersonation [8]. Therefore, an enhanced security technique is presented in [8]. Unfortunately, the cryptanalysis by the authors in [16] revealed that this protocol is vulnerable to sensor node and user impersonation. To prevent against sensor node capture attacks, Physically Unclonable Function (PUF) based authentication schemes are presented in [17], [18] [19], [20] and [21]. However, the protocol in [17] executes repeated fuzzy extractions during the authentication phase, resulting in high computational overheads and latencies. In addition, the deployed bilinear pairing operations further diminish its performance. On the other hand, the protocol developed in [19] cannot withstand stolen verifier threats [22]. In addition, PUF-based authentication protocols have stability challenges due to the sensitivity to environmental conditions of the challenge-responses.

To resist DoS threats, rate-limiting and lightweight authentication scheme is presented in [23] based on Elliptic Curve Cryptography (ECC) and Obfuscated PUF (OPUF). This scheme is shown to resist security threats such as impersonation, Ephemeral-Secret Leakage (ESL) and modeling attacks. However, it inherits the instability features of PUF and hence the same hardware can generate different responses when exposed to diverse environmental conditions. Similarly, ECC and PUF have been deployed in the development of a privacy-preserving authentication scheme in [24]. For enhanced security and privacy protection, this protocol also incorporates biometric, smart cards, and passwords in the authentication procedures. Similarly, the three-factor authentication technique in [25] integrates ECC, fuzzy extraction, biometrics and smart cards. However, the deployment of biometrics requires special hardware which increases implementation complexity. In addition, repeated fuzzy extractions and scalar point multiplications result in high computation overheads. To minimize these computational costs, a scheme based on only one-way hashing and XOR operations is developed in [26]. Nevertheless, this protocol is susceptible to offline password guessing, replay and privileged insider threats [27].

To address ephemeral-secret leakages as well as high communication and computational overheads, an authentication protocol is introduced in [28]. On the other hand, a cloud-centric authenticated key exchange protocol is developed in [29] for enhanced security during the communication between autonomous vehicles and the cloud. On the flip side, this scheme is defenseless against privileged insider threats [30]. Table 1 provides the comparative summary table of these existing approaches.

Table 1: Comparative summary of related works

Protocol	Year	Cryptographic technique	Research gap (s)
Kim et al. [8]	2026	PUF + Fuzzy extraction	• Susceptible to sensor node and user impersonation
Keshta [15]	2025	Hashing + ECC	• Does not support user untraceability
			• Is defenseless against threats such as DoS, session key disclosure, sensor node capture and user impersonation
Subramani et al. [17]	2021	PUF	• High computational overheads & latencies
Nyangareshi et al. [18]	2021	PUF + ECC	• Instability challenges
Shao et al. [19]	2022	PUF + Fuzzy extraction	• Cannot withstand stolen verifier threats
Wu et al. [20]	2025	PUF	• Instability challenges
Wang et al. [21]	2025	PUF + ECC	• Instability challenges
Kuo et al. [23]	2026	OPUF + ECC	• Instability challenges
Guo et al. [24]	2024	PUF + ECC + Fuzzy extraction	• Requires special hardware for biometrics capture, which increases implementation complexity
Arpitha et al. [25]	2024	PUF + Fuzzy extraction	• Requires special hardware for biometrics capture, which increases implementation complexity
Masud et al. [26]	2021	Hashing + XOR	• Vulnerable to offline password guessing, replay and privileged insider threats
Shang et al. [28]	2025	ECC + Fuzzy extraction	• Not evaluated against attacks such as session-hijacking & DoS
Mishra et al. [29]	2023	RLWE	• Cannot defend against privileged insider threats
Ahmad et al. [30]	2024	Post-quantum fuzzy commitment (PQFC)	• Lacks evaluation against attacks such as DoS & ESL
Dabra et al. [31]	2021	Lattice + Password	• Not evaluated against threats such as session hijacking
			• Susceptible to signal leakage attacks
Bahache et al. [32]	2024	Kyber	• Lacks evaluation against attacks such as session disclosure
Rewal et al. [33]	2023	Lattice + Password	• Not evaluated against threats such as side-channeling & ESL
			• High computation overheads
Chaudhary et al. [34]	2024	RLWE + Password	• High communication overheads
			• Lacks evaluation against attacks such as UKS
Mishra et al. [35]	2023	RLWE + Password	• Not evaluated against attacks such as DoS & ESL
Islam et al. [36]	2021	Lattice + Password	• Lacks evaluation against attacks such as eavesdropping

Most of the security schemes discussed so far are based on conventional cryptographic techniques such as ECC, which are easily broken by quantum algorithms. As such, authentication protocols based on post-quantum cryptography have been introduced in [30], [31], [32], [33], [34], [35] and [36]. However, the security analyzes of these schemes failed to consider security threats such as session hijacking, DoS, ESL, session disclosure and side-channeling. In addition, the lattice-based security scheme in [31] is vulnerable to signal leakage attacks [37]. On the other hand, the lattice-based protocol in [33] implements biometric hashing, which results in high computational overheads [30]. Although the scheme in [34] supports user anonymity, a single session is accomplished via numerous message transformations, which results in high communication overheads [30]. Similarly, the Kyber algorithm-based protocol in [32] incurs high communication and computation overheads [30].

3. The Proposed Framework

In this section, we present the mathematical preliminaries of the basic cryptographic operations. This is followed by the descriptions of the attack model as well as the network model. Thereafter, the major phases of the proposed framework are described.

3.1 Mathematical preliminaries

The proposed security scheme is based on the Ring Learning With Error (RLWE) and hashing functions. Let us take Z as a set of integers and security parameter $a = 2^s$, where security token $s \in Z$. In addition, we denote auxiliary modular function, irreducible polynomial, the rings of integer polynomials, fixed real number, discrete Gaussian distribution and the rings of I modulo as F , P , I , n , $\tilde{G}$ and $\mathcal{R}$ respectively. We then consider P defined over Z , defined as $(x^a + 1) \in Z[x]$. The ring of this polynomial over Z_q is denoted as $Z_q[x]$, where $q \bmod 2a \equiv 1$, $q = 2^{b(\log_2 a)} + 1$ is the prime modulus, Z_q is the modular group, and $b \in Z_q$. On the other hand, $\mathcal{R}$ is expressed as $(x^a + 1)$ over Z , that is $\mathcal{R} = Z[x]/(x^a + 1)$. The rings of polynomials $\mathcal{R}_q = Z_q[x]/(x^a + 1)$ with similar modulus and each multiplier is reduced modulo q over Z_q . Here, $\mathcal{R}$ comprise of q^a elements, each of which is basically a polynomial whose degree is less than a . On the other hand, the coefficients of each of these elements are $Z_q = \{0, 1, \dots, q-1\}$. Considering $n > 0$, the $\tilde{G}$ is represented by x_n over $\mathcal{R}_q$. Let $c \in Z_q$ and $e = Cha(c)$ and hence the definition of $F: Z_q \times \{0, 1\} \rightarrow \{0, 1\}$ can be written as $F(c, e) = (c + e \cdot \frac{q-1}{2}) \bmod q \bmod 2$ [38].

Lemma 1: Suppose that c is uniformly selected at random from Z_q and $q = 2^{b(\log_2 a)} + 1 > 2$ is an odd prime. For $\lambda b \in Z_q$, $\lambda e \in \{0, 1\}$, the output distribution of $F(c + b, e)$ given $Cha(c)$ is statistically close to homogeneous on $\{0, 1\}$ [38].

Lemma 2: Based on [38], given q and $\lambda d \in \mathcal{R}_q$ such that $|d| < \frac{q}{8}$, we have $F(\lambda, Cha(\lambda)) = F(b, Cha(\lambda))$, where $b = \lambda + 2 \cdot d$ [38]. On rings of polynomials $\mathcal{R}_q$, Cha and F are effortlessly expandable functions:

Given $\lambda = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \dots + \lambda_{a-1} x^{a-1} \in \mathcal{R}$, this can be regarded as a vector $\lambda = (\lambda_0, \lambda_1, \dots, \lambda_{a-1})$. For $c = (c_0, c_1, \dots, c_{a-1}) \in \{0, 1\}^a$, we have $Cha(\lambda) = (Cha(\lambda_0), Cha(\lambda_1), \dots, Cha(\lambda_{a-1}))$, $F(\lambda, c) = (F(\lambda_0, c_0), \dots, F(\lambda_{a-1}, c_{a-1}))$.

Definition 1: In RLWE, we consider $y \in \mathcal{R}_q$ and ρ_{n, x_n} are distributions over $(\lambda, \lambda \cdot y + 2 \cdot d) \in \mathcal{R}_q \times \mathcal{R}_q$, where $\lambda \in \mathcal{R}_q$ is chosen uniformly at random, and $d \leftarrow x_n$. Given a fixed sample size from x_n and many polynomial samples ρ , a polynomial time-bound algorithm $\tilde{A}$ cannot distinguish ρ_{ϕ, x_n} from the uniform distribution on $\mathcal{R}_q \times \mathcal{R}_q$.

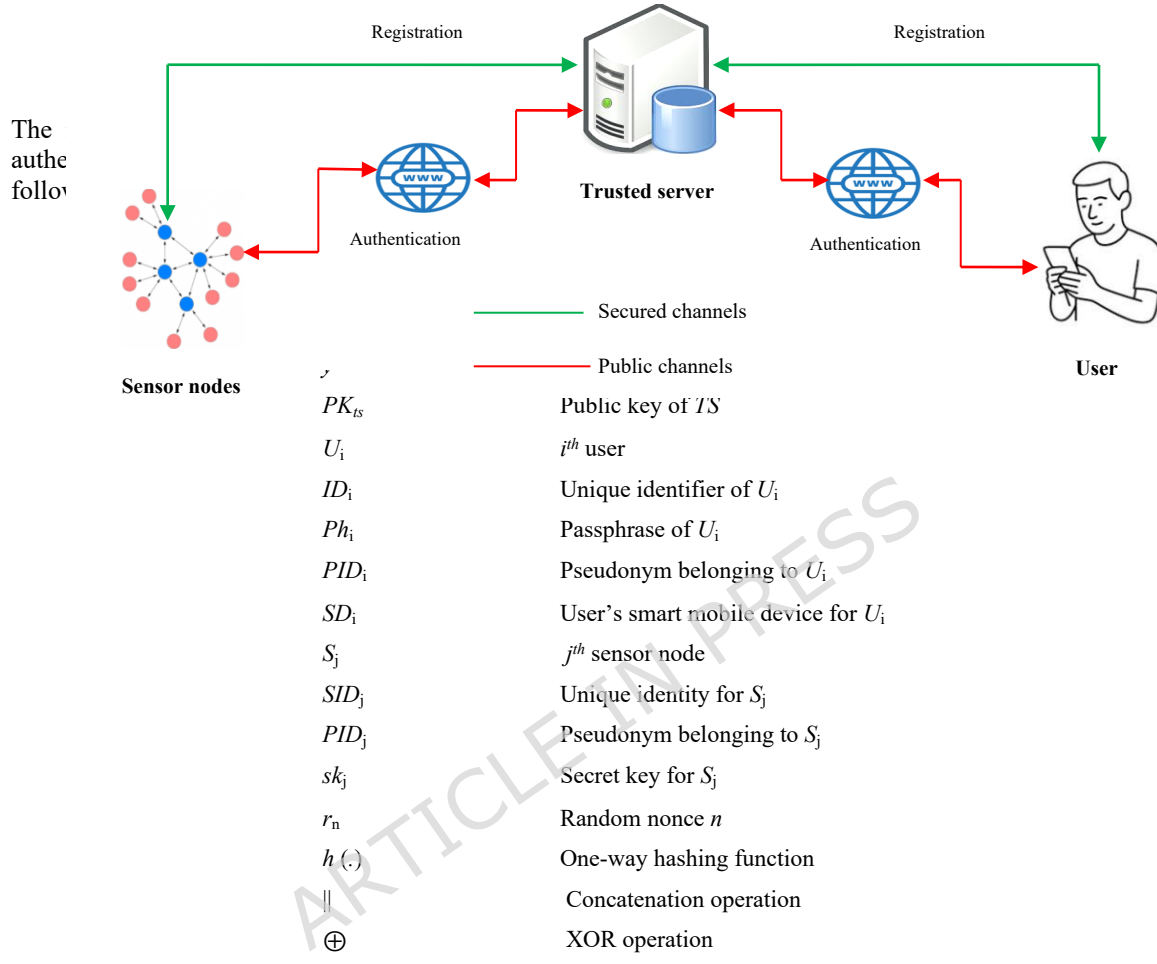
3.2 Attack model

To represent the various adversarial capabilities, we deploy the Extended Canetti-Krawczyk (eCK) attack model. This choice is informed by its rigor and wider deployment during the analysis of Authenticated Key Exchange (AKE) protocols. In addition, eCK captures a broader range of attacker capabilities compared to its peers. Specifically, adversary $\tilde{A}$ can:

- Take full control of the network and inject malicious messages into the network. In addition, $\tilde{A}$ can intercept, erase, or alter the transmitted messages.
- Learn the session-specific secrets (such as random nonces) deployed in a particular session.
- Get access to temporary session-specific information such as partial derivations or intermediate values.
- Obtain long-term (static) secret keys of any network participant.
- Access non-trivial amalgamation of long-terms and ephemeral secret keys.

3.3 Network model

The network architecture of the proposed authentication framework comprises of the Trusted Server (*TS*), sensor node S_j , and user U_i that deploys smart mobile device SD_i to interact with other network participants. Fig.1 depicts the interaction among these network entities.



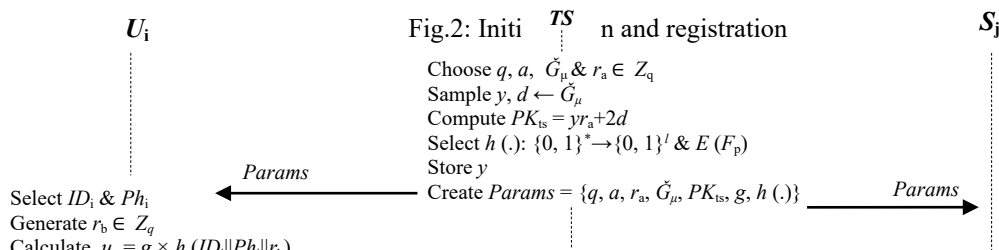
3.4 System initialization

This phase is executed by the Trusted Server (*TS*) and encompasses the generation of security tokens deployable in the subsequent registration, login, mutual authentication and key negotiation phases. The steps that follow are executed during this system setup phase.

Step 1: The *TS* selects some large prime number q , integer a , and Gaussian distribution $\tilde{G}_\mu$. Here, μ is the standard deviation of this Gaussian distribution, where $\mu > 0$ and parameter a assume the form of power of 2.

Step 2: The Trusted Server *TS* chooses random nonce $r_a \in \mathbb{Z}_q$. It then stochastically samples $y, d \leftarrow \tilde{G}_\mu$ and derives its public key as $PK_{ts} = yr_a + 2d$. This is followed by the selection of cryptographic one-way hashing function $h(.)$: $\{0, 1\}^* \rightarrow \{0, 1\}^l$, where l is the length of the hashing output.

Step 3: The *TS* chooses an elliptic curve $E(F_p)$, where F_p is the finite field and g is the generator of the elliptic curve group. At the end, the trusted server secretly stores y as its secret key prior to forwarding $Params = \{q, a, r_a, \tilde{G}_\mu, PK_{ts}, g, h(.)\}$ to the U_i and S_j as shown in Fig.2.



3.5 User registration

In the proposed protocol, all users must register at the trusted server TS prior to accessing the sensors or utilizing any WSN services. This user registration phase is executed across secured communication channels. The specific details of this phase are detailed in the steps that follow.

Step 1: User U_i chooses his/her unique identity ID_i and passphrase Ph_i . This is followed by the generation of random nonce $r_b \in Z_q$ and the derivation of $u_a = g \times h(ID_i || Ph_i || r_b)$. Next, it creates registration message $reg_a = \{ID_i, u_a\}$ that is transmitted to the trusted server TS as evidenced in Fig.2.

Step 2: After receiving message reg_a , the TS derives $s_a = h(ID_i)$, the user's pseudonym as $PID_i = h(ID_i || y)$ that it utilizes to compute value $s_b = PID_i \oplus h(u_a)$. Next, it stores s_a in its database before forwarding response $res_a = \{s_b\}$ towards user U_i .

Step 3: Upon getting res_a , user U_i derives $u_a = g \times h(ID_i || Ph_i || r_b)$, $PID_i = s_b \oplus h(u_a)$, and $u_b = h(ID_i || Ph_i || r_b || PID_i)$. Finally, he/she stores parameters $\{s_b, u_b, r_b, h(.)\}$ in smart mobile device SD_i .

3.6 Sensor node registration

All the sensor nodes within the network in the proposed authentication framework must register at the trusted server TS prior to their deployment to their respective field. The specific details of these procedures are described below.

Step 1: The sensor node S_j chooses its unique identity SID_j and secret key sk_j . Next, it generates random nonce and derives $r_c \in Z_q$ and derives $n_a = g \times h(SID_j || sk_j || r_c)$. This is followed by the construction of registration message $reg_b = \{SID_j, n_a\}$ that is forwarded to the trusted server TS .

Step 2: Upon getting reg_b , the TS calculates $s_c = h(SID_j)$, the sensor node pseudonym as $PID_j = h(SID_j || y)$ that it employs to compute value $s_d = PID_j \oplus h(n_a)$. Next, it stores s_c in its database before forwarding response $res_b = \{s_d\}$ towards sensor node S_j as shown in Fig.2.

Step 3: On receiving response res_b , S_j derives values $n_a = g \times h(SID_j || sk_j || r_c)$, $PID_j = s_d \oplus h(n_a)$ and $n_b = h(SID_j || sk_j || r_c || PID_j)$. Finally, sensor node S_j stores parameters $\{s_d, n_b, r_c, h(.)\}$ in its memory.

3.7 Login, authentication and key setup

In this phase, the user, trusted server and sensor node jointly verify each other's identity. After successful mutual validation procedures, a common session key is established to encipher all the exchanged data among these network participants. The specific steps below are carried out to effectively verify the network participants and setup a common session key among them.

Step 1: User U_i inputs his/her unique identity ID_i and passphrase Ph_i into smart mobile device SD_i . This is followed by the derivation of $u_a^* = g \times h(ID_i || Ph_i || r_b)$, $PID_i^* = s_b \oplus h(u_a^*)$ and $u_b^* = h(ID_i || Ph_i || r_b || PID_i^*)$. The SD_i then extracts u_b from its memory and confirms whether $u_b^* \stackrel{?}{=} u_b$, halting the session if this condition does not hold. Otherwise, it generates $r_d, c_u \leftarrow \tilde{G}_\mu$.

Step 2: The smart mobile device SD_i derives $u_c = r_a r_d + 2c_u$, $u_d = r_d PK_{ts}$, $\lambda_u = Cha(u_d)$, $u_e = F(u_d, \lambda_u)$, $u_f = ID_i \oplus h(u_c || u_e)$ and $u_g = h(u_f || u_c || u_e || ID_i || PID_i)$. It then creates authentication request $auth_a = \{u_c, u_g, u_f, \lambda_u\}$ that is forwarded to the trusted server TS across public communication channels as evidenced in Fig.3.

Step 3: Upon receiving $auth_a$, the TS calculates $u_d^* = u_c y$, $u_e^* = F(u_d^*, \lambda_u)$, $ID_i^* = u_f \oplus h(u_e^* || u_c)$, $PID_i^* = h(ID_i^* || y)$ and $u_g^* = h(u_f || u_c || u_e^* || ID_i^* || PID_i^*)$. It then checks if $u_g^* \stackrel{?}{=} u_g$, terminating the session if this verification flops. Otherwise, it generates $z_i \leftarrow \tilde{G}_\mu$.

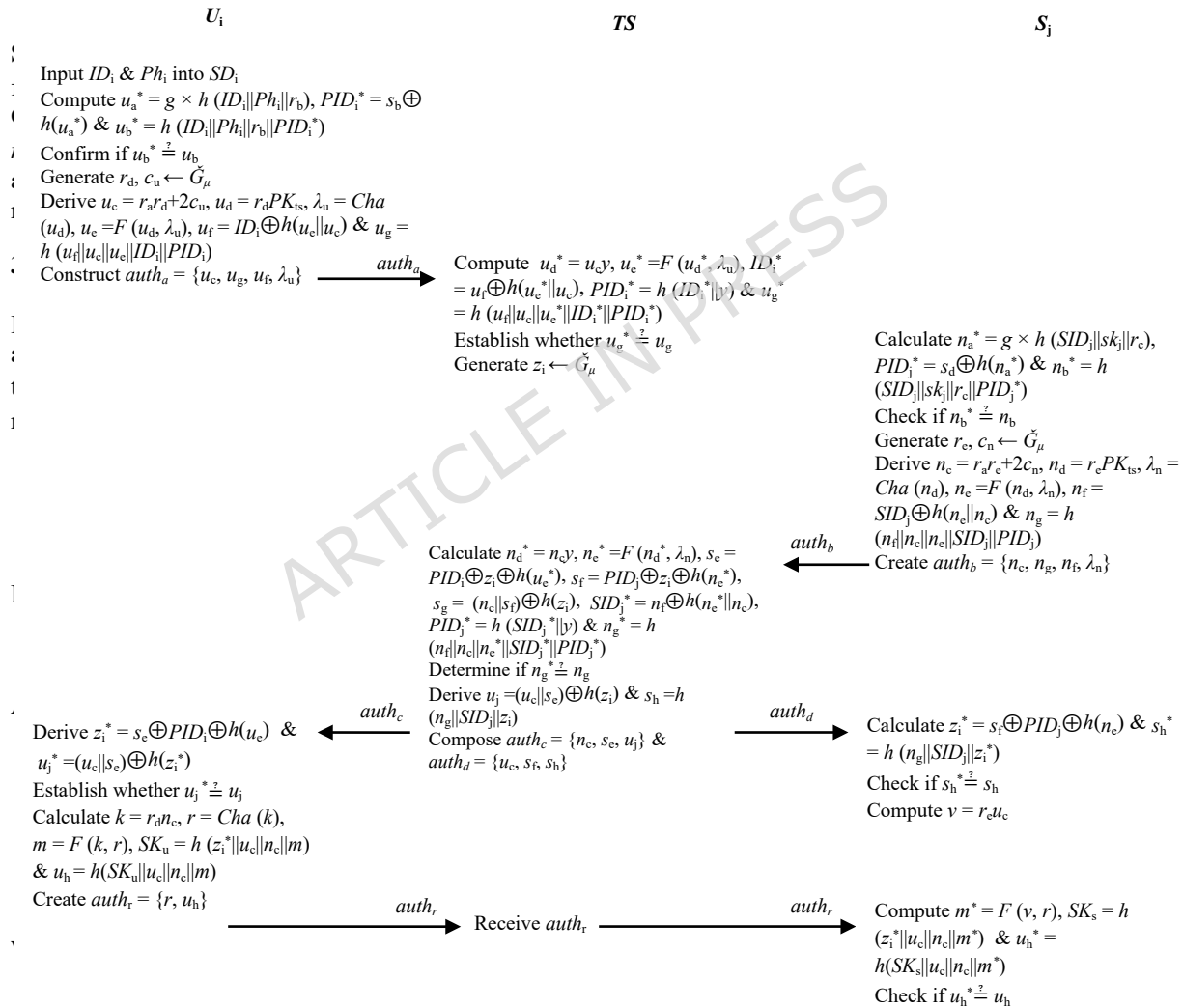
Step 4: Sensor node S_j derives $n_a^* = g \times h(SID_j || sk_j || r_c)$, $PID_j^* = s_d \oplus h(n_a^*)$ and $n_b^* = h(SID_j || sk_j || r_c || PID_j^*)$. This is followed by the confirmation of whether $n_b^* \stackrel{?}{=} n_b$, aborting the session if these values are not equivalent. Otherwise, it generates $r_e, c_n \leftarrow \tilde{G}_\mu$ and derives $n_c = r_a r_e + 2c_n$, $n_d = r_e PK_{ts}$, $\lambda_n = Cha(n_d)$, $n_e = F(n_d, \lambda_n)$, $n_f = SID_j \oplus h(n_e || n_c)$ and

$n_g = h(n_f || n_c || n_e || SID_j || PID_j)$. At the end, it composes authentication message $auth_b = \{n_c, n_g, n_f, \lambda_n\}$ that is forwarded to the trusted server TS across public communication channels.

Step 5: After getting $auth_b$, the TS computes $n_d^* = n_c y$, $n_e^* = F(n_d^*, \lambda_n)$, $s_e = PID_i \oplus z_i \oplus h(u_e^*)$, $s_f = PID_j \oplus z_j \oplus h(n_e^*)$, $s_g = (n_c || s_f) \oplus h(z_i)$, $SID_j^* = n_f \oplus h(n_e^* || n_c)$, $PID_j^* = h(SID_j^* || y)$ and $n_g^* = h(n_f || n_c || n_e^* || SID_j^* || PID_j^*)$. This is followed by the confirmation of whether $n_g^* \stackrel{?}{=} n_g$, halting the session when this check turns false. Otherwise, it calculates $u_j = (u_c || s_e) \oplus h(z_i)$ and $s_h = h(n_g || SID_j || z_i)$. Next, it creates authentication message $auth_c = \{n_c, s_e, u_j\}$ that is transmitted towards the user U_i . Similarly, it constructs authentication message $auth_d = \{u_c, s_f, s_h\}$ which is forwarded to the sensor node S_j across the public internet as shown in Fig.3.

Step 6: Upon obtaining $auth_c$, the user's smart mobile device SD_i derives $z_i^* = s_e \oplus PID_i \oplus h(u_e)$ and $u_j^* = (u_c || s_e) \oplus h(z_i^*)$. Next, it checks if $u_j^* \stackrel{?}{=} u_j$, terminating the session when this condition does not hold. Otherwise, it derives $k = r_d n_c$, $r = Cha(k)$, $m = F(k, r)$, session key $SK_u = h(z_i^* || u_c || n_c || m)$ and $u_h = h(SK_u || u_c || n_c || m)$. At the end, it creates authentication response $auth_r = \{r, u_h\}$ that is forwarded to the sensor node S_j via the trusted server TS .

Fig. 3: Login, authentication and key negotiation



This implies that for a particular factor ω ,

$$\omega(u_d, \lambda_u) = \omega(u_d^*, \lambda_u^*) \quad (6)$$

Based on equation (6), we obtain the following:

$$\begin{aligned} u_g &= h(u_f || u_c || u_e || ID_i || PID_i) \\ &= h(u_f || u_c || u_e^* || ID_i^* || PID_i^*) = u_g^* \quad (7) \end{aligned}$$

In a similar manner, the *TS* verifies sensor node S_j by checking the validity of value n_g . Upon receiving authentication message $auth_b$, the *TS* derives $n_d^* = n_c y$, $n_e^* = F(n_d^*, \lambda_n)$, $s_e = PID_i \oplus z_i \oplus h(u_e^*)$, $s_f = PID_j \oplus z_i \oplus h(n_e^*)$, $s_g = (n_c || s_f) \oplus h(z_i)$, $SID_j^* = n_f \oplus h(n_e^* || n_c)$, $PID_j^* = h(SID_j^* || y)$ and $n_g^* = h(n_f || n_c || n_e^* || SID_j^* || PID_j^*)$. This is followed by the determination of whether $n_g^* \stackrel{?}{=} n_g$, aborting the authentication session if this condition does not hold. To attain valid n_e , values n_d and n_d^* must satisfy the inequality $|n_d - n_d^*| < \frac{q}{8}$.

$$n_d = r_e PK_{ts} = r_e (y r_a + 2d) = r_e \cdot r_a \cdot y + 2d \cdot r_e \quad (8)$$

$$n_d^* = n_c y = y (r_a r_e + 2c_n) = y \cdot r_a r_e + 2c_n \cdot y \quad (9)$$

According to equation (8) and equation (9), the following is obtained.

$$|n_d - n_d^*| = 2|d \cdot r_e - c_n \cdot y| \quad (10)$$

The application of *Lemma 1* and *Lemma 2* to equation (10) yields the following.

$$\begin{aligned} |d \cdot r_e - c_n \cdot y| &\leq |d \cdot r_e| + |c_n \cdot y| \leq \sqrt{a} \cdot ||d|| \cdot ||r_e|| + \sqrt{a} \cdot ||c_n|| \cdot ||\varphi|| \\ &< \sqrt{a} \cdot \sqrt{a} \cdot \mu \cdot \sqrt{a} \cdot \mu + \sqrt{a} \cdot \sqrt{a} \cdot \mu \cdot \sqrt{a} \cdot \mu \\ &= 2 \cdot \sqrt[3]{a \cdot \mu^2} \quad (11) \end{aligned}$$

where $a \ll q$ and $a = b(\sqrt{\log a})$. Therefore,

$$|d \cdot r_e - c_n \cdot y| \leq 2 \cdot \sqrt[3]{a \cdot \mu^2} < \frac{q}{8} \quad (12)$$

The implication is that for a particular factor ω ,

$$\omega(n_d, \lambda_n) = \omega(n_d^*, \lambda_n^*) \quad (13)$$

According to equation (13), the following obtained:

$$\begin{aligned} n_g &= h(n_f || n_c || n_e || SID_j || PID_j) \\ &= h(n_f || n_c || n_e^* || SID_j^* || PID_j^*) = n_g^* \quad (14) \end{aligned}$$

The *TS* then constructs $auth_c = \{n_c, s_e, u_j\}$ that is transmitted towards the user U_i . This is followed by having the user's SD_i verify *TS* by checking whether $u_j \stackrel{?}{=} u_i$, where $u_j = (u_c || s_e) \oplus h(z_i)$. This requires that the user's SD_i derive the correct value for z_i . Since user U_i has access to keying values $\{SID_j, u_e\}$, value z_i can be derived correctly.

4. Security analysis

In this section, we provide formal security analysis of the proposed authentication framework using the Random Oracle Model (ROM). In addition, we carry out informal security analysis to demonstrate that our scheme is secure under some critical attacker capabilities in the eCK model.

4.1 Formal security analysis

In the proposed quantum safe (S_Q) protocol, the three network entities include user U_i , trusted server *TS*, and sensor node S_j . In this sub-section, we deploy the ROM which include various adversarial capabilities and assumptions. In

this model, the oracle queries simulate the attacker capabilities during the actual attacks. These queries depict various interactions between the adversary $\tilde{A}$ and the participants. The different kinds of queries that adversary $\tilde{A}$ can launch include the following:

Execute ($\pi_{U_i}^i, \pi_{S_j}^j, TS^k$): This query models a passive attack which allows attackers to access all the information exchanged between the network entities. Using this query, the adversary solicits the challenger, which then responds with the data exchanged among the participants.

Send ($\pi_{q_s}^i, m$): This query models an active attack through which an adversary $\tilde{A}$ gets access to the exchanged messages. In addition, this query allows the adversary to alter or construct new messages and relay them to the network participants. In response to this query, the receiving network entity generates an appropriate response.

Reveal ($\pi_{q_s}^i$): This query is deployed by the adversary to gauge the security of the proposed protocol against session key extraction attacks from compromised network entities. For instance, $\tilde{A}$ might attempt to get the session keys derived by instances $\{\pi_{U_i}^i, \pi_{S_j}^j, TS^k\}$. The output of this query is either a null $\perp$ value or the session key $\{SK_s, SK_u\}$.

Test ($\pi_{q_s}^i$): Using this query, the attacker determines the semantic security of the proposed S_Q protocol. Provided that the session key is not defined or the reveal query has not been run, it returns null $\perp$ or a random value ϕ . If $\phi=0$, this query returns a random number of same length as session keys $\{SK_s, SK_u\}$, otherwise it returns $\{SK_s, SK_u\}$.

Corrupt ($\pi_{q_s}^i$): This query is utilized by the adversary $\tilde{A}$ to obtain the authentication parameters such as $\{y, Ph_i\}$.

The formal concepts that define the security of the proposed protocol include freshness, accept state and partnering. The details of these concepts are described below.

Accepted state: An instance $\pi_{q_s}^i$ is regarded as being in the accepted state upon receiving the final anticipated message.

Freshness: For an oracle to be regarded as fresh (or is in possession of a fresh session key $SK_s = SK_u$), the following three conditions must be met:

- The *Reveal* (.) query has not yielded instance $\pi_{\tilde{A}}^i$ or its partner.
- Instance $\pi_{\tilde{A}}^i$ is in the accepted state
- Prior to instance $\pi_{\tilde{A}}^i$ being in the accepted state, no oracle has executed the *Corrupt* (.) query.

Partnering: Instances π_X^i and π_Z^i are regarded as being partners if the following three conditions are upheld:

- Both π_X^i and π_Z^i are in accepted state.
- Instances π_X^i and π_Z^i share a common session key SK .
- There exist no other instance apart from π_X^i and π_Z^i that shares this session key SK .

Theorem 1: Let z_a and z_b be Zipf's parameters and q_s, q_e and q_h be the number of *Send* (.), *Execute* (.), and *Hash* (.) queries respectively. In addition, we let $Adv_{\tilde{A}}^{q_s}(t)$ be the advantage of the adversary $\tilde{A}$ in compromising the proposed S_Q protocol with at most q_s, q_e and q_h queries in polynomial time t . Moreover, we let $Adv_{\tilde{A}}^{RLWE}(t)$ be the probability of solving the *RLWE* problem in polynomial time t by adversary $\tilde{A}$.

$$Adv_{\tilde{A}}^{S_Q}(t) \leq \frac{2q_h^2}{q} + \frac{2q_s}{q} + \frac{(q_e+q_s)^2}{q} + (2q_h)Adv_{\tilde{A}}^{RLWE}(t) + \frac{2q_h}{q} + 2(z_a.q_s^{z_b}) \quad (8)$$

Based on equation (8), it is evident that the security of the proposed framework is hinged on the difficulties of solving the *RLWE* problem.

Proof: Let us consider an adversary $\tilde{A}$ and challenger $\hat{C}$. The above equation (8) is proofed by having $\tilde{A}$ and challenger $\hat{C}$ engage in a challenge-response game. This involves adversary $\tilde{A}$ initiating various attacks by asking challenger $\hat{C}$ to carry out some queries. Depending on the adversarial query, $\hat{C}$ institutes some response in accordance with the protocol, which are then forwarded towards $\tilde{A}$. This is followed by the analysis of received responses and attempts to break the semantic security of the session key negotiated in the proposed protocol. We deploy a series of 6 games $Game_0$ to $Game_5$, and denote the event for each game and its probability as $\hat{E}_i$ and $P_r(\hat{E}_i)$ respectively. Therefore, for each game $Game_j$, event $\hat{E}_i$ is defined depending on whether or not the adversary $\tilde{A}$ is successful in breaching the semantic security of our protocol in $Game_j$. In each of these games, while adversary $\tilde{A}$ is executing, event $\hat{E}$ (observable by $\hat{C}$ and independent of $\hat{E}_i$) can take place. Until event $\hat{E}$ takes place, games $Game_i$ and $Game_j$ are identical. As such,

$$|P_r[\hat{E}_{i+1}] - P_r[\hat{E}_i]| \leq P_r[\hat{E}] \quad (9)$$

$Game_0$: This games simulates the actual attack on the semantic security of the proposed S_Q protocol. Under the ROM,

$$Adv_{\tilde{A}}^{S_Q}(t) = |P_r(\hat{E}_0) - \frac{1}{2}| \quad (10)$$

Suppose $Z_i = |P_r(\hat{E}_{i-1}) - P_r(\hat{E}_i)|$. As such, equation (10) can be transformed as follows.

$$\begin{aligned} Adv_{\tilde{A}}^{S_Q}(t) &= |P_r(\hat{E}_0) - \frac{1}{2}| = |P_r(\hat{E}_0) - P_r(\hat{E}_4) + P_r(\hat{E}_4) - \frac{1}{2}| \\ &= |\sum_{i=1}^5 Z_i + P_r(\hat{E}_4) - \frac{1}{2}| \quad (11) \end{aligned}$$

$Game_1$: This games involves the execution of the hash query by adversary $\tilde{A}$. To accomplish this, the challenger $\hat{C}$ maintains an ordered pair (x, y) in list l , where $y = h(x)$. Whenever $\tilde{A}$ launches a hash query for x , $\hat{C}$ searches list l for the pair (x, y) . If this pair exists, challenger $\hat{C}$ returns y to the adversary $\tilde{A}$. Otherwise, it stochastically generates y^* and appends (x, y^*) into l . As such,

$$P_r[\hat{E}_1] = P_r[\hat{E}_0] \quad (12)$$

$Game_2$: In this game, the attacker executes collision attacks on the authentication messages $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$. Based on the birthday paradox, the highest likelihood that the hash oracle can yield the same outcome is at most $\frac{q_h^2}{2q}$. On the other hand, the maximum possibility that two stochastic samples are similar is at most $\frac{(q_e + q_s)^2}{2q}$ and hence,

$$|P_r[\hat{E}_1] - P_r[\hat{E}_2]| \leq (\frac{q_h^2}{2q} + \frac{(q_e + q_s)^2}{2q}) \quad (13)$$

$Game_3$: This game simulates the *Test* (.) query through which adversary may predict bit ϕ , devoid of executing the hash query. As such,

$$|P_r[\hat{E}_2] - P_r[\hat{E}_3]| = \frac{q_h}{2q} \quad (14)$$

$Game_4$: The aim of the attacker in this game is to predict session keys $\{SK_s, SK_u\}$ devoid of invoking the hash query. In the proposed framework, the session keys are derived as $SK_s = h(z_i^* || u_c || n_c || m^*)$ and $SK_u = h(z_i^* || u_c || n_c || m)$. Therefore, the adversary simulates a self-generalizable instance of RLWE problem. Basically, the correct prediction of session keys $\{SK_s, SK_u\}$ involves the successful solving of RLWE problem in polynomial time t .

$$|P_r[\hat{E}_3] - P_r[\hat{E}_4]| \leq q_h \cdot Adv_{\tilde{A}}^{RLWE}(t) + \frac{q_h}{q} \quad (15)$$

Game₅: In this game, the adversary launches the hash query with session keys $SK_s = SK_u$ as inputs. The probability of correctly guessing bit ϕ through the *Test* (.) query is at most $\frac{q_h^2}{2p}$. Devoid of the simulation of the hash query with the correct input, adversary $\tilde{A}$ is unable to distinguish the real session key from any stochastic bit. Therefore,

$$P_r(\hat{E}_5) = \frac{1}{2} \quad (16)$$

To guess the low-entropy user passphrase Ph_i , Zipf's law on passwords is utilized. According to [40], if $q_s = 10^7$ or 10^8 , the adversary's advantage of correctly guessing Ph_i is greater than 0.5. On the other hand, if $q_s \leq 10^6$, the adversary's advantage of carrying out a successful targeted guessing attack is over 0.5. As such, the probability of offline-guessing attack is less than $z_a \cdot q_s^{z_b}$ [40]. As such,

$$|P_r[\hat{E}_4] - P_r[\hat{E}_5]| \leq \frac{q_h^2}{2q} + z_a \cdot q_s^{z_b} \quad (17)$$

By the combination of all inequalities and equations, we obtain equation (18) and hence our scheme is secure under the RLWE postulation.

$$Adv_{\tilde{A}}^{S_Q}(t) \leq \frac{2q_h^2}{q} + \frac{2q_s}{q} + \frac{(q_e + q_s)^2}{q} + (2q_h)Adv_{\tilde{A}}^{RLWE}(t) + \frac{2q_h}{q} + 2(z_a \cdot q_s^{z_b}) \quad (18)$$

4.2 Informal security analysis

In this sub-section, we show that our scheme is secure under some critical the threat assumptions under the Extended Canetti-Krawczyk (eCK) attack model.

- *Robust authentication and key agreement*

In the proposed protocol, all the three network participants mutually verify each other and setup a session key for message enciphering and deciphering. For instance, the smart mobile device SD_i validates user U_i by checking whether $u_b^* \stackrel{?}{=} u_b$; while trusted server TS verifies SD_i by determining if $u_g^* \stackrel{?}{=} u_g$. On the other hand, the TS authenticates sensor node S_j by checking if $n_g^* \stackrel{?}{=} n_g$; while the SD_i validates TS by establishing whether $u_j^* \stackrel{?}{=} u_j$. On its part, the sensor node S_j authenticates TS and U_i by determining if $s_h^* \stackrel{?}{=} s_h$ and $u_h^* \stackrel{?}{=} u_h$, respectively. In all these authentications, the sessions are halted upon verification failure. Otherwise, the SD_i and S_j derive the session keys as $SK_u = h(z_i^* || u_c || n_c || m)$ and $SK_s = h(z_i^* || u_c || n_c || m^*)$ respectively.

- *Man-in-the-Middle and forgery attacks*

The goal of the adversary in this attack is to intercept and manipulate the communication among network participants, making them trust that they are directly communicating with each other. In our scheme, messages $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$ are exchanged over the public internet. Here, $u_c = r_a r_d + 2c_u$, $u_g = h(u_i || u_c || ID_i || PID_i)$, $u_d = r_d PK_{ts}$, $\lambda_u = Cha(u_d)$, $u_e = F(u_d, \lambda_u)$, $u_f = ID_i \oplus h(u_c || u_c)$, $n_c = r_a r_c + 2c_n$, $n_d = r_c PK_{ts}$, $\lambda_n = Cha(n_d)$, $n_e = F(n_d, \lambda_n)$, $n_f = SID_j \oplus h(n_c || n_c)$, $n_g = h(n_i || n_c || n_c || SID_j || PID_j)$, $s_e = PID_i \oplus z_i \oplus h(u_c^*)$, $PID_i = h(ID_i || y)$, $s_f = PID_j \oplus z_i \oplus h(n_c^*)$, $s_g = (n_c || s_f) \oplus h(z_i)$, $u_j = (u_c || s_e) \oplus h(z_i)$, and $s_h = h(n_g || SID_j || z_i)$, $r = Cha(k)$, $m = F(k, r)$, $k = r_d n_c$, $SK_u = h(z_i^* || u_c || n_c || m)$ and $u_h = h(SK_u || u_c || n_c || m)$. Suppose that attacker $\tilde{A}$ wants to forge each of these messages $\{auth_a, auth_b, auth_c, auth_d, auth_r\}$. However, this requires secret values such as $\{SID_j, PID_j, y, r_a, r_d, r_c\}$, which are unavailable to $\tilde{A}$.

- *Confidentiality and integrity protection*

These two security functionalities are critical for the preservation privacy in the sense that confidentiality ensures that message contents are only accessible to authorized parties, while integrity ensures that these contents are not maliciously modified. In the proposed framework, we deploy RLWE problem and hashing functions to preserve these two security features. For instance, parameters u_c and n_c in these messages are protected in a way that the adversary cannot retrieve session specific private keys $\{r_d, r_c\}$ devoid of solving the RLWE problem. On the other hand, values such as $\{u_b, u_f, s_g, s_h\}$ are protected by the one-way hashing functions.

- *Ephemeral secret leakage, key compromise and perfect key secrecy*

These security features ensure that the security of the past and future session keys is preserved even in the face of active compromise of the current session keys SK_u and SK_s . Whereas forward key secrecy preserves the confidentiality of past communications, backward key secrecy upholds the confidentiality of future communication sessions. In the proposed framework, the session keys are derived as $SK_u = h(z_i^* || u_c || n_c || m)$ and $SK_s = h(z_i^* || u_c || n_c || m^*)$. Here, $z_i \leftarrow \check{G}_\mu$, $u_c = r_a r_d + 2c_u$, $c_u \leftarrow \check{G}_\mu$, $n_c = r_a r_e + 2c_n$, $c_n \leftarrow \check{G}_\mu$, $m = F(k, r)$, $k = r_d n_c$, $r = Cha(k)$, $m^* = F(v, r)$, and $v = r_e u_c$. It is evident that our framework incorporates ephemeral secrets/session-specific private keys such as $\{r_d, r_e, z_i, c_u, c_n\}$. In the proposed framework, values $\{y, s_a, s_c\}$, $\{s_b, u_b, r_b, h(.)\}$ and $\{s_d, n_b, r_c, h(.)\}$ are stored in TS , SD_i , and S_j , respectively. On the other hand, the publicly exchanged messages $\{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$ do not contain any of these ephemeral secrets. As such, these session-specific private keys cannot be leaked through eavesdropping and power analysis. Therefore, the only option is leakage of random nonces $\{r_d, r_e\}$ through guessing. However, the derived session keys are still safe due to the fact that these random nonces are regenerated after every authentication phase.

- *Anonymity and untraceability*

In the proposed protocol, messages $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$ are transmitted across the open communication channels. Here, $u_c = r_a r_d + 2c_u$, $u_g = h(u_f || u_c || u_c || ID_i || PID_i)$, $u_d = r_d PK_{ts}$, $\lambda_u = Cha(u_d)$, $u_e = F(u_d, \lambda_u)$, $u_f = ID_i \oplus h(u_c || u_c)$, $n_c = r_a r_e + 2c_n$, $n_d = r_e PK_{ts}$, $\lambda_n = Cha(n_d)$, $n_e = F(n_d, \lambda_n)$, $n_f = SID_j \oplus h(n_c || n_c)$, $n_g = h(n_f || n_c || n_c || SID_j || PID_j)$, $s_e = PID_i \oplus z_i \oplus h(u_c^*)$, $PID_i = h(ID_i || y)$, $s_f = PID_j \oplus z_i \oplus h(n_c^*)$, $s_g = (n_c || s_f) \oplus h(z_i)$, $u_j = (u_c || s_e) \oplus h(z_i)$, $s_h = h(n_g || SID_j || z_i)$, $r = Cha(k)$, $m = F(k, r)$, $k = r_d n_c$, $SK_u = h(z_i^* || u_c || n_c || m)$ and $u_h = h(SK_u || u_c || n_c || m)$. Evidently, none of these messages has plain-text identities $\{ID_i, SID_j\}$. Instead, these identities are encapsulated in parameters $\{u_g, u_f, n_f, n_g, s_e, s_f, s_g, s_h, SK_u\}$. As such, adversaries cannot discern these unique identities from the publicly exchanged messages, preserving anonymity. During the system initialization phase, values $\{q, a, r_a, \check{G}_\mu, PK_{ts}, g, h(.)\}$ are shared to the users and sensor nodes. Evidently, the attacker is unable to recover identities $\{ID_i, SID_j\}$ from any of these public parameters. It is also evident that these messages incorporate one-time values such as $\{r_d, r_e, z_i, c_u, c_n\}$. Therefore, these messages are quite distinct for various communication sessions, upholding untraceability.

- *Impersonation threats*

To effectively masquerade as legitimate user U_i , trusted server TS and sensor node S_j , the attacker must create legitimate authentication messages $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$. However, this requires that adversary $\hat{A}$ has access to secrets such as $\{SID_j, y, ID_i, z_i\}$. At the receiver end, values $\{u_g, n_b, n_g, u_j, s_h, u_h\}$ are re-computed as $u_g^* = h(u_f || u_c || u_c^* || ID_i^* || PID_i^*)$, $n_b^* = h(SID_j || s_k || r_c || PID_j^*)$, $s_h^* = h(n_g || SID_j || z_i^*)$, $u_h^* = h(SK_s || u_c || n_c || m^*)$. This is followed by the verification of whether $u_g^* \stackrel{?}{=} u_g$, $n_b^* \stackrel{?}{=} n_b$, $n_g^* \stackrel{?}{=} n_g$, $u_j^* \stackrel{?}{=} u_j$, $s_h^* \stackrel{?}{=} s_h$, and $u_h^* \stackrel{?}{=} u_h$. As such, any bogus messages created by the adversaries are easily detected and the authentication session halted.

- *Eavesdropping and side-channeling*

In these security threats, we assume that the adversary has intercepted and eavesdropped messages $\{auth_a, auth_b, auth_c, auth_d, auth_r\}$. Here, $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$. In addition, it is assumed that the attacker has utilized power analysis techniques to retrieve values $\{y, s_a, s_c\}$, $\{s_b, u_b, r_b, h(.)\}$ and $\{s_d, n_b, r_c, h(.)\}$ stored in TS , SD_i , and S_j respectively. Here, $y \leftarrow \check{G}_\mu$, $s_a = h(ID_i)$, $s_b = PID_i \oplus h(u_a)$, $s_c = h(SID_j)$, $u_a = g \times h(ID_i || Ph_i || r_b)$, $u_b = h(ID_i || Ph_i || r_b || PID_i)$, $s_d = PID_j \oplus h(n_a)$, and $n_b = h(SID_j || s_k || r_c || PID_j)$. Next, attempts are made to derive session keys $SK_u = h(z_i^* || u_c || n_c || m)$ and $SK_s = h(z_i^* || u_c || n_c || m^*)$. However, these session keys incorporate ephemeral secrets $\{r_d, r_e, z_i, c_u, c_n\}$ which cannot be obtained via eavesdropping and power analysis.

- *Replay attacks*

The goal of the adversary in this attack is to intercept legitimate messages and retransmit them later on, hoping to gain access to the networked systems. To avert these security threats, the proposed framework incorporates session-specific secrets to ensure that each communication session is unique. A typical communication session involves the

exchange of messages $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$. Here, $u_c = r_a r_d + 2c_u$, $u_g = h(u_f || u_c || u_c || ID_i || PID_i)$, $u_d = r_d PK_{ts}$, $\lambda_u = Cha(u_d)$, $u_e = F(u_d, \lambda_u)$, $u_f = ID_i \oplus h(u_c || u_c)$, $n_c = r_a r_e + 2c_n$, $n_d = r_e PK_{ts}$, $\lambda_n = Cha(n_d)$, $n_e = F(n_d, \lambda_n)$, $n_f = SID_j \oplus h(n_c || n_c)$, $n_g = h(n_f || n_c || n_c || SID_j || PID_j)$, $s_e = PID_i \oplus z_i \oplus h(u_c^*)$, $PID_i = h(ID_i || y)$, $s_f = PID_j \oplus z_j \oplus h(n_c^*)$, $s_g = (n_c || s_f) \oplus h(z_i)$, $u_j = (u_c || s_e) \oplus h(z_i)$, $s_h = h(n_g || SID_j || z_i)$, $r = Cha(k)$, $m = F(k, r)$, $k = r_d n_c$, $SK_u = h(z_i^* || u_c || n_c || m)$ and $u_h = h(SK_u || u_c || n_c || m)$. It is evident that these messages include session-specific values such as $\{r_d, r_e, z_i\}$. As such, any replayed message is easily detected at the receiver end.

- *No need for clock synchronization*

In majority of the WSN authentication protocols, timestamps are utilized for replay protection. However, this requires that the communicating systems be clock synchronized. This clock synchronization presents some critical challenges regarding how it should be implemented. To bridge this gap, we deploy random nonces in place of timestamps. This effectively eliminates the need for coordinating time across diverse networks or devices. These random nonces are included in the various security parameters exchanged among the TS , sensor node S_j and user mobile device SD_i .

- *Known key threats*

In these threats, it is assumed that adversary $\tilde{A}$ has access to some of the previously exchanged messages as well as session keys. Using these values, attempts are made to derive the current session keys $SK_s = h(z_i^* || u_c || n_c || m^*)$ and $SK_u = h(z_i^* || u_c || n_c || m)$. However, these session keys are protected by the collision-resistant one-way hashing functions. In addition, this derivation requires random selections for $\{r_d, r_e, z_i, c_u, c_n\}$ be made from the discrete Gaussian distribution $\tilde{G}_\mu$, where $n > 0$ for each session.

- *Unknown key share threats*

In the proposed protocol the sensor node S_j and user mobile device SD_i derive session keys as $SK_s = h(z_i^* || u_c || n_c || m^*)$ and $SK_u = h(z_i^* || u_c || n_c || m)$ respectively. As such, both S_j and SD_i believe that session key $SK_s = SK_u$ is only shared between themselves. This is because these session keys are only derived after successful joint authentication between S_j and SD_i through the trusted server TS . This mutual authentication is accomplished by establishing whether $u_g^* \stackrel{?}{=} u_g$, $n_g^* \stackrel{?}{=} n_g$, $u_j^* \stackrel{?}{=} u_j$, $s_h^* \stackrel{?}{=} s_h$, and $u_h^* = u_h$.

- *No key control*

In the proposed authentication framework, both sensor node S_j and user mobile device SD_i perform stochastic selection of $\{r_d, r_e, z_i, c_u, c_n\}$ from the discrete Gaussian distribution $\tilde{G}_\mu$. These values are then deployed to derive the session keys $SK_s = h(z_i^* || u_c || n_c || m^*)$ and $SK_u = h(z_i^* || u_c || n_c || m)$. Evidently, both S_j and SD_i contribute security tokens deployed to derive these session keys. As such, it is not feasible for a single entity (S_j or SD_i) to derive the same session keys devoid of the knowledge of security tokens from the other party.

- *Session key integrity and session hijacking*

To preserve the integrity of the session key, no any other entity apart from sensor node S_j and user mobile device SD_i should be able to derive it. As already seen, $SK_u = h(z_i^* || u_c || n_c || m)$ and $SK_s = h(z_i^* || u_c || n_c || m^*)$, where $z_i \leftarrow \tilde{G}_\mu$, $u_c = r_a r_d + 2c_u$, $c_u \leftarrow \tilde{G}_\mu$, $n_c = r_a r_e + 2c_n$, $c_n \leftarrow \tilde{G}_\mu$, $m = F(k, r)$, $k = r_d n_c$, $r = Cha(k)$, $m^* = F(v, r)$, and $v = r_e u_c$. It is clear that these session keys incorporate session-specific private keys such as $\{r_d, r_e, z_i, c_u, c_n\}$ selected uniformly from Gaussian distribution $\tilde{G}_\mu$. As such, the proposed authentication framework preserves session key integrity and mitigates session hijacking threats.

- *Privileged insider*

In this attack, we assume that some privileged entity has access to the security parameters $\{y, s_a, s_c\}$ stored at the trusted server TS . Next, $\tilde{A}$ attempts to derive session keys $SK_u = h(z_i^* || u_c || n_c || m)$ and $SK_s = h(z_i^* || u_c || n_c || m^*)$. Evidently, none of the accessed parameters is directly incorporated in the derivation of these session keys. In addition, the attacker lacks session-specific private keys such as $\{r_d, r_e, z_i, c_u, c_n\}$ required for successful derivation of $\{SK_u, SK_s\}$.

DoS

The aim of these attacks is to exhaust the processing resources of the targets to an extent that they are unable to respond to legitimate requests. To prevent these security threats, the proposed authentication framework first verifies the authenticity of the received messages prior to carrying out any further processing. These prior checks include $u_b^* \stackrel{?}{=} u_b$, $u_g^* \stackrel{?}{=} u_g$, $n_g^* \stackrel{?}{=} n_g$, $u_j^* \stackrel{?}{=} u_j$, $s_h^* \stackrel{?}{=} s_h$, and $u_h^{**} \stackrel{?}{=} u_h$. On condition that the outcomes of such checks are negative, the session is halted, preserving the processing resources of the network participants. Although some preliminary computations are executed upon reception of authentication messages, they are limited to derivation of security values required for the stated prior checks. However, upon failure of these checks, no further processing is performed.

5. Performance evaluation

In wireless sensor networks, computation overheads, supported functionalities, energy consumption, communication and memory overheads are frequently deployed during the performance evaluations. As such, these performance indicators are deployed in this section to gauge the performance of the proposed authentication framework.

5.1 Computation overheads

In this sub-section, we deploy the cryptographic operations executed during the login, mutual authentication and key negotiation to derive the computation costs of the developed authentication framework. Here, we denote the time for elliptic curve point multiplication, Gaussian distribution sampling, single scalar multiplication, chaotic map operation, characteristic function, mean cost of single multiplication in Q_q (for one multiplication and addition in Q_q), and hashing as T_{ecm} , T_g , T_m , T_{cm} , T_{cf} , T_{mz} , T_{az} , and T_h respectively. We deployed Raspberry Pi 6 for the SD_i and S_j , and an Intel i7-8565U laptop equipped with 8 GB of RAM and 4.6 Ghz processor for the TS . The implementation was in C/C++ using the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) and Lattice-crypto libraries. Under these platforms, Table 3 gives the runtimes for various crypto-operations.

Table 3: Cryptographic runtimes

Operation	Execution time (μs)	
	SD_i/S_j	TS
T_g	0.6251	0.0752
T_m	0.0073	0.0004
T_{az}	0.0357	0.0048
T_{mz}	0.0029	0.0003
T_{cf}	0.0385	0.0007
T_h	0.1625	0.0128
T_{cm}	1.1374	0.1172
T_{ecm}	3.4122	0.3516

During the login, authentication and key negotiation phase, the sensor node S_j and user's smart mobile device SD_i carry out $5T_h+2T_g+2T_{mz}+T_{az}+T_m+2T_{cf}$ operations each. On the other hand, the trusted server TS executes $10T_h+T_g+2T_{mz}$ operations as shown in Table 4.

Table 4: Computation overheads

Framework	Smart device/sensor node	Server/TA	Total (μs)
Rewal et al., [33]	$12T_h+4T_g+2T_{mz}+2T_{az}+2T_m+2T_{cf} \approx 4.6192$	$6T_h \approx 0.0768$	4.6960
Chaudhary et al., [34]	$13T_h+4T_g+3T_{mz}+2T_{az}+2T_m+2T_{cf} \approx 4.7846$	$7T_h+2T_{cf}+T_{mz} \approx 0.0913$	4.8759
Mishra et al., [35]	$10T_h+4T_g+3T_{mz}+2T_{az}+2T_m+2T_{cf} \approx 4.2971$	$6T_h+T_{mz} \approx 0.0771$	4.3742
Islam & Basu [36]	$16T_h+4T_g+2T_{mz}+2T_{az}+2T_m+2T_{cf} \approx 5.2692$	$8T_h \approx 0.1024$	5.3716
Proposed	$10T_h+4T_g+4T_{mz}+2T_{az}+2T_m+4T_{cf} \approx 4.377$	$10T_h+T_g+2T_{mz} \approx 0.2038$	4.5808

As shown in Fig.4, the authentication protocol developed in [36] incurs the highest overall computation cost of 5.3716 μs . This is followed by the schemes in [34], [33], the proposed framework, and the protocol in [35] with execution durations of 4.8759 μs , 4.696 μs , 4.5808 μs , and 4.3742 μs respectively.

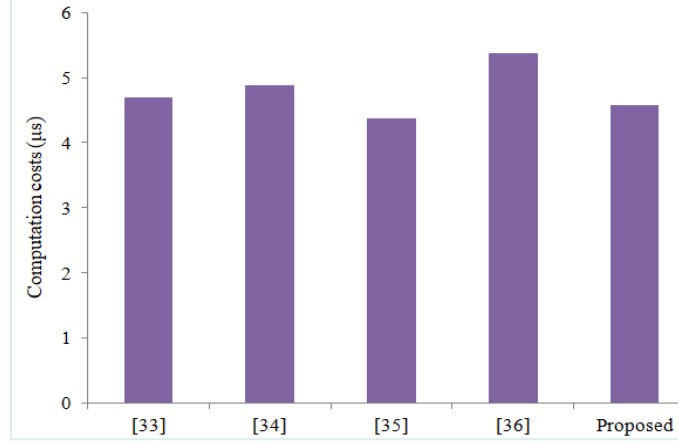


Fig.4: Computation costs comparisons

Although the protocol presented in [35] incurs the lowest computation overheads, it fails to provide security functionalities such as anonymity, untraceability, forward key secrecy and session key integrity. In addition, it has not been analyzed against security threats such as unknown key share, known key side-channeling, ESL, session hijacking, and DoS attacks. Therefore, although the proposed framework incurs slightly higher computation overheads, it supports robust security protection in the wireless sensor networks ecosystem.

5.2 Communication costs

In this sub-section, we deploy the sizes of the messages exchanged during the login, authentication and key agreement phase to derive the communication overhead of our framework. These messages include $auth_a = \{u_c, u_g, u_f, \lambda_u\}$, $auth_b = \{n_c, n_g, n_f, \lambda_n\}$, $auth_c = \{n_c, s_e, u_j\}$, $auth_d = \{u_c, s_f, s_h\}$ and $auth_r = \{r, u_h\}$. Here, we take the size of the passphrase, random nonces and real identities to be 128 bits long, while the timestamp is taken to be 32 bits long. Based on [33]-[36], the output of symmetric encryption/decryption, one-way hashing, chaotic mapping, elliptic curve point and each element from Q_q are 256 bits, 512 bits, 256 bits, 320 bits and 4094 bits respectively. In addition, these benchmark authentication frameworks deploy the following RLWE parameter set: $a = 1024$ bits, the log of standard deviation $\log \mu = 17.01$ for the Gaussian distribution, and $q > 2$. Therefore, $auth_a = \{4094+512+512+256=5374 \text{ bits}\}$, $auth_b = \{4094+512+512+256=5374 \text{ bits}\}$, $auth_c = \{4094+512+512=5118 \text{ bits}\}$, $auth_d = \{4094+512+512=5118 \text{ bits}\}$, and $auth_r = \{256+512=768 \text{ bits}\}$. As such, the cumulative communication overhead is 21752 bits. Table 5 presents the communication costs of the proposed framework, including those of other related schemes.

Table 5: Communication overheads

Framework	No. of messages	Total (bits)
Rewal et al., [33]	4	21504
Chaudhary et al., [34]	4	21248
Mishra et al., [35]	3	16384
Islam & Basu [36]	6	20346
Proposed	5	21752

As evidenced in Fig.5, the proposed authentication technique incurs the highest communication overhead of 21752 bits. This is followed by the protocols in [33], [34], [36], and [35] with communication costs of 21504 bits, 21248 bits, 20346 bits, and 16384 bits respectively.

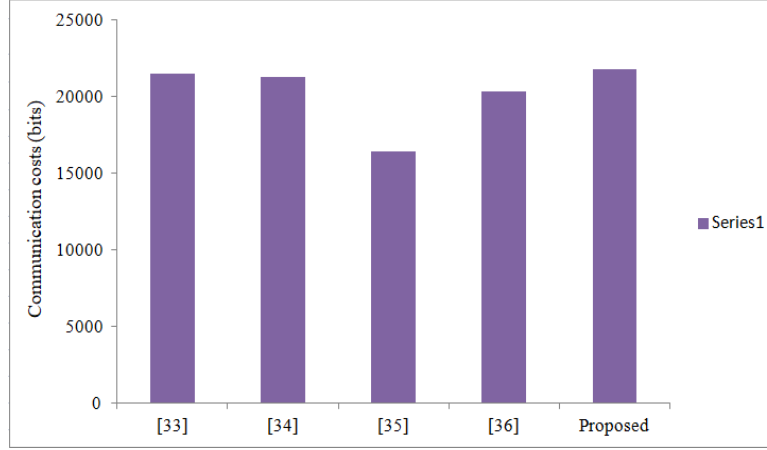


Fig.5: Communication costs comparisons

Although the security architecture developed in [35] is efficient in terms of communication overheads, it has not been evaluated against a wide range of typical WSN security threats such as DoS and ESL. In addition, it cannot offer salient security functionalities such as anonymity and untraceability. Similarly, the authentication schemes developed in [33], [34], [36] have not been evaluated against attacks such as ESL, session hijacking, and DoS. Since the proposed security framework supports these functionalities and resists the said attacks, it offers a good trade-off between security and bandwidth consumption.

5.3 Energy consumption

In the proposed authentication, the Trusted Server (*TS*) has enough computing resources and hence can handle sophisticated cryptographic operations. On the other hand, the sensor node S_j and user's smart mobile device SD_i are resource-constrained. It is therefore required that an authentication protocol be lightweight on these two devices. According to [39], $E = T_{\text{comp}} \times P_{\text{cpu}}$, where E is the energy, T_{comp} is the computation cost, and P_{cpu} is maximum consumed CPU power, which is 10.88 W. Table 6 presents the energy consumption comparisons.

Table 6: Energy consumption

Framework	Energy (J)
Rawal et al., [33]	50.2569×10^{-6}
Chaudhary et al., [34]	52.0564×10^{-6}
Mishra et al., [35]	46.7524×10^{-6}
Islam & Basu [36]	57.3289×10^{-6}
Proposed	47.6218×10^{-6}

As shown in Table 6, the authentication scheme developed in [36] consumes the highest amount of energy of 57.3289×10^{-6} Joules. This is followed by the authentication techniques developed in [34], [33], the proposed authentication framework, and the scheme in [35] with energy consumptions of 52.0564×10^{-6} , 50.2569×10^{-6} , 47.6218×10^{-6} , and 46.7524×10^{-6} , respectively. Although the authentication scheme in [35] consumes the least amount of energy, it does not support security characteristics such as session key integrity, untraceability, anonymity, and forward key secrecy. In addition, it has not been analyzed against security threats such as DoS, ESL, known key side-channeling, session hijacking, and unknown key share. On the other hand, the proposed authentication scheme supports these security features and resists the identified security attacks. As such, our authentication framework offers an ideal balance between energy consumption and security.

5.4 Memory consumption

In wireless sensor networks, it is required that storage overheads be reduced so as to render authentication protocols memory-efficient. This is mostly the case for the sensor nodes S_j and any other storage constrained devices such as smart mobile devices SD_i . Therefore, we present the storage overheads of the developed authentication framework

and compare it with its peers. During the initialization and registration phases, the sensor nodes S_j stores $\{s_d, n_b, r_c, h(.)\}$ while smart mobile devices SD_i store $\{s_b, u_b, r_b, h(.)\}$. Here, $s_d = PID_j \oplus h(n_a)$, $n_b = h(SID_j || sk_j || r_c || PID_j)$, $s_b = PID_i \oplus h(u_a)$ and $u_b = h(ID_i || Ph_i || r_b || PID_i)$. Using the values specified in sub-section 5.2 above, the storage overhead at the sensor nodes S_j and smart mobile devices SD_i are $\{512+512+128+512 = 1664 \text{ bits}\}$ and $\{512+512+128+512 = 1664 \text{ bits}\}$ respectively. As such, the total storage overhead of the proposed authentication framework at the S_j and SD_i is 3328 bits. Table 7 presents the comparative evaluation of the storage overheads of our framework and other related schemes.

Table 7: Storage overheads

Framework	Storage (bits)
Rewal et al., [33]	3072
Chaudhary et al., [34]	2048
Mishra et al., [35]	1984
Islam & Basu [36]	4544
Proposed	3328

As shown in Table 6, the authentication scheme developed in [36] requires the highest storage overhead of 4544 bits. This is followed by the proposed authentication framework with storage cost of 3328 bits, as well as the authentication protocols developed in [33], [34] and [35] with storage overheads of 3072 bits, 2048 bits, and 1984 bits, respectively. Although the schemes [33], [34] and [35] incur relatively low storage overheads, their security evaluation omits critical threats such as Unknown Key share attack (UKS), side-channeling, ESL, session hijacking and DoS. However, the developed authentication protocol mitigates these attacks and hence offers a salient balancing between storage overheads and security.

5.5 Supported functionalities

In this section, we compare and contrast the security and privacy posture of the developed authentication framework against its peers. To accomplish this, we utilize the supported security features and attack resilience of our scheme, which we then contrast against other related approaches as shown in Table 8.

Table 8: Functionalities

Functionality	[33]	[35]	[34]	[36]	[Proposed]
Mutual authentication	√	√	√	√	√
Key agreement	√	√	√	√	√
Anonymity	×	×	√	×	√
No clock synchronization	×	×	√	×	√
Untraceability	√	×	√	×	√
Confidentiality	×	×	×	×	√
No Key Control (NKC)	√	×	×	√	√
Forward key secrecy	√	×	√	√	√
Session key integrity	√	×	×	×	√
Resilience against:					
Unknown Key share attack (UKS)	×	×	×	√	√
Known Key attack (KK)	√	×	×	√	√
Session disclosure	×	√	×	×	√
Side-channeling	×	×	×	×	√
ESL	×	×	×	×	√
Session hijacking	×	×	×	×	√
Key compromise	√	√	×	√	√
Eavesdropping	×	√	×	×	√
Replay	√	√	√	√	√
Forgery	×	√	×	×	√
MitM	√	√	√	√	√
DoS	×	×	×	×	√

Impersonation	√	√	√	√	√
Privileged insider	√	√	×	√	√
	√ Supported; × Not supported or not considered				

As shown in Table 8, the schemes in [33], [35], [34], and [36] support 12, 10, 9 and 11 functionalities respectively. On the other hand, the proposed framework supports all the 23 functionalities. Consequently, the protocol developed in [34] is the most vulnerable and hence not recommended for deployment in the wireless sensor network where some sensitive data is being exchanged. On the other hand, the proposed scheme is the most robust and hence is the most suitable for deployment in the WSN ecosystem. Nevertheless, the proposed authentication executes non-negligible processing prior to performing the pre-checks aimed at mitigating DoS attacks. There is therefore need to explore novel approaches such as rate-limiting and lightweight pre-checks for curing DoS attacks.

6 Conclusion

The data exchanged in wireless sensor networks is vulnerable to security and privacy threats such as forgery, eavesdropping, tampering and replays. As such, numerous security frameworks have been developed to counter these threats. However, most of these current works heavily deploy conventional cryptographic operations such as ECC, bilinear pairings, PUF and biometrics. These techniques have been shown to be either vulnerable to advanced threats such as quantum attacks or inefficient in terms of communication and computation overheads. Therefore, they cannot be readily deployed in WSNs where devices such as sensors have limited resources. To this end, this paper has presented a quantum-safe framework that is shown to resist majority of the conventional wireless sensor network security threats. In addition, we have demonstrated that our approach is secure under all the attacker capabilities in the ROM and eCK model. Future works in this domain will center around further reductions in the yielded communication and computation overheads while retaining the attained security posture. In addition, future works should be focused on the implementation of techniques such as rate-limiting and lightweight pre-checks for addressing DoS attacks. Moreover, innovative techniques for mitigating offline guessing attacks are required, as well as leakage modeling and implementation countermeasures for effectively addressing side-channel attacks.

Declarations

Ethical Approval:

Not applicable.

Competing interests:

The authors declare no competing interests.

Author contributions:

Author contributions; Conceptualization: Z.A.A., Z.A.H., V.O.N., A.H.A., M.A.A., M.A.H., I.Q.A., J.M., A.J.Y.A., D.G.H., M.M.A.; Methodology: Z.A.A., Z.A.H., V.O.N., A.H.A., M.A.A., M.A.H., I.Q.A.; Investigation: J.M., A.J.Y.A., D.G.H., M.M.A.; Validation: J.M., A.J.Y.A., D.G.H., M.M.A.; Formal analysis: Z.A.A., Z.A.H., V.O.N., A.H.A., M.A.A., M.A.H., I.Q.A.; Writing—original draft preparation: Z.A.A., Z.A.H., V.O.N., A.H.A., M.A.A., M.A.H., I.Q.A.; Supervision: Z.A.A., A.H.A.; Writing—review and editing: J.M., A.J.Y.A., D.G.H., M.M.A.; Project Administration: Z.A.A., A.H.A.; Visualization: J.M., A.J.Y.A., D.G.H., M.M.A.; Resources: Z.A.A., Z.A.H., V.O.N., A.H.A., M.A.A., M.A.H., I.Q.A. All authors have read and agreed to the published version of the manuscript. Note: All authors reviewed the manuscript.

Funding:

This research did not receive any funding.

Data availability:

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. Abdulai JD, Adu-Manu KS, Katsriku FA, Engmann F. A modified distance-based energy-aware routing protocol in wireless sensor networks. *J Ambient Intell Human Comput.* 2023;14(8):10195–10217. <https://doi.org/10.1007/s12652-021-03683-y>
2. Tsallis C, Papageorgas P, Piromalis D, Munteanu RA. Industrial wireless networks in Industry 4.0: a systematic review. *J Sens Actuator Netw.* 2026;15(1):7. <https://doi.org/10.3390/jsan15010007>
3. Singh A, Bhardwaj V, Sharma P, Renald CT. Analysis of wireless sensor network in IoT: applications and challenges. *Recent Trends Intell Comput Commun.* 2025:476–481. <https://doi.org/10.1201/9781003593034-75>
4. Sangeethapriya A, Devaraju S, Cynthia A. Advancements in energy optimization and security in wireless sensor networks: a comprehensive survey. In: *From Bits to Qubits: The Quantum Transformation of Computing*. Springer; 2026. p. 449–461. https://doi.org/10.1007/978-3-032-00586-1_20
5. Pajila PB, Robinson YH, Jevin JA. Blockchain-enabled deep learning techniques for providing security in wireless sensor networks. *Peer-to-Peer Netw Appl.* 2026;19(1):34. <https://doi.org/10.1007/s12083-025-02185-4>
6. Nyangaresi VO. Lightweight and privacy-preserving authentication protocol for sensitive data transmission in smart healthcare applications. In: *Digital Twin, Blockchain, and Sensor Networks in the Healthy and Mobile City*. Elsevier; 2025. p. 301–325. <https://doi.org/10.1016/B978-0-443-34174-8.00017-X>
7. Keshmiry A, Hassani S. Digital twin for smart wireless sensor networks. In: *Advances in Fog Computing and the Internet of Things for Smart Healthcare*. Elsevier; 2026. p. 327–342. <https://doi.org/10.1016/B978-0-443-33441-2.00013-6>
8. Kim M, Kim T, Kwon D, Park Y. PUF-based secure authentication protocol for cloud-assisted wireless medical sensor networks. *Electronics.* 2026;15(1):240. <https://doi.org/10.3390/electronics15010240>
9. Ahn J, Kwon D, Park Y. Secure and efficient authentication protocol for underwater wireless sensor network environments using PUF. *Appl Sci.* 2026;16(2):873. <https://doi.org/10.3390/app16020873>
10. Fouad Y, Abbas S, Ahmed N, Ghareeb AN, Selem E. Smart weather aware drone sink for reliable and energy-efficient agricultural wireless sensor networks. *Sci Rep.* 2025. <https://doi.org/10.1038/s41598-025-31821-5>
11. Ogundoyin SO, Kamil IA, Ojedokun IA, Babalola JO, Nyangaresi VO. On the privacy preservation and secure communication in fog-based mobile crowdsensing. *J Parallel Distrib Comput.* 2025:105184. <https://doi.org/10.1016/j.jpdc.2025.105184>
12. Elsmay EF, Mustafa MS, Ahmed ZE, Altahir AA. An examination of AI-integrated wireless sensor networks security: obstacles and prospects. In: *Ensuring Secure Connectivity Through AI-Powered Wireless Systems*. IGI Global; 2026. p. 49–92. <https://doi.org/10.4018/979-8-3373-3576-6.ch002>
13. Kumar MN, Vijayan T, Karthik B. Intrusion detection framework for industrial wireless sensor networks in smart manufacturing. In: *Pioneering AI and Data Technologies for Next-Gen Security, IoT, and Smart Ecosystems*. IGI Global; 2026. p. 197–218. <https://doi.org/10.4018/979-8-3373-4672-4.ch010>
14. Tavakkoli-Moghaddam R, Dolgui A, Nozari H. Energy efficiency in wireless sensor networks. In: *Energy-Efficient Transformative Technologies for Data-Driven Smart Cities*. Elsevier; 2026. p. 21–36. <https://doi.org/10.1016/B978-0-443-27618-7.00011-X>
15. Keshta I. A cloud-assisted key agreement protocol for the E-healthcare system. *PLoS One.* 2025;20(6):e0322313. <https://doi.org/10.1371/journal.pone.0322313>
16. Xie Q, Ding Z, Xie Q. A lightweight and privacy-preserving authentication protocol for healthcare in an IoT environment. *Mathematics.* 2023;11(18):3857. <https://doi.org/10.3390/math11183857>
17. Subramani J, Maria A, Rajasekaran AS, Al-Turjman F. Lightweight privacy and confidentiality preserving anonymous authentication scheme for WBANs. *IEEE Trans Ind Inform.* 2021;18(5):3484–3491. <https://doi.org/10.1109/TII.2021.3097759>

18. Nyangaresi VO, Ahmad M, Maghrabi LA, Althaqafi T. Cost-effective PUF and ECC based authentication protocol for secure internet of drones communication. *IEEE Internet Things J.* 2025. <https://doi.org/10.1109/JIOT.2025.3576313>
19. Shao X, Guo Y, Guo Y. A PUF-based anonymous authentication protocol for wireless medical sensor networks. *Wireless Netw.* 2022;28(8):3753–3770. <https://doi.org/10.1007/s11276-022-03070-1>
20. Wu TY, Wu H, Kumari S, Chen CM. An enhanced three-factor based authentication and key agreement protocol using PUF in IoMT. *Peer-to-Peer Netw Appl.* 2025;18(2):83. <https://doi.org/10.1007/s12083-024-01839-z>
21. Wang X, Wang W, Liu A, Liu W, Zhang Z, Li W. PIA-A secure and efficient identity authentication scheme in telemedicine via the PUF method. *Sci Rep.* 2025;15(1):6846. <https://doi.org/10.1038/s41598-025-89502-2>
22. Yang H, Guo Y, Guo Y. Blockchain-based cloud-fog collaborative smart home authentication scheme. *Comput Netw.* 2024;242:110240. <https://doi.org/10.1016/j.comnet.2024.110240>
23. Kuo WC, Ghaffar Z, Mahmood K, Tariq T, Shamshad S, Das AK. A lightweight authentication scheme for securing patient information in the internet of medical things environment. *IEEE Internet Things J.* 2026;13:1208–1215. <https://doi.org/10.1109/JIOT.2025.3628187>
24. Guo P, Liang W, Xu S. A privacy preserving four-factor authentication protocol for internet of medical things. *Comput Secur.* 2024;137:103632. <https://doi.org/10.1016/j.cose.2023.103632>
25. Arpitha T, Chouhan D, Shreyas J. Anonymous and robust biometric authentication scheme for secure social IoT healthcare applications. *J Eng Appl Sci.* 2024;71(1):8. <https://doi.org/10.1186/s44147-023-00342-1>
26. Masud M, Gaba GS, Choudhary K, Hossain MS, Alhamid MF, Muhammad G. Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet Things J.* 2021;9(4):2649–2656. <https://doi.org/10.1109/JIOT.2021.3080461>
27. Kim K, Ryu J, Lee Y, Won D. An improved lightweight user authentication scheme for the internet of medical things. *Sensors.* 2023;23(3):1122. <https://doi.org/10.3390/s23031122>
28. Shang Y, Chen J, Wang S, Zhang Y, Ma K. A secure and lightweight ECC-based authentication protocol for wireless medical sensors networks. *Sensors.* 2025;25(21):6567. <https://doi.org/10.3390/s25216567>
29. Mishra D, Pursharthi K, Rewal P. Development of quantum-enhanced authenticated key agreement protocol for autonomous vehicles. *Veh Commun.* 2023;44:100688. <https://doi.org/10.1016/j.vehcom.2023.100688>
30. Ahmad A, Jagatheswari S. Quantum safe multi-factor user authentication protocol for cloud assisted medical IoT. *IEEE Access.* 2024;13:3532–3545. <https://doi.org/10.1109/ACCESS.2024.3523530>
31. Dabra V, Bala A, Kumari S. LBA-PAKE: lattice-based anonymous password authenticated key exchange for mobile devices. *IEEE Syst J.* 2021;15(4):5067–5077. <https://doi.org/10.1109/JSYST.2020.3023808>
32. Bahache AN, Chikouche N, Akleyek S. Securing cloud-based healthcare applications with a quantum-resistant authentication and key agreement framework. *Internet Things.* 2024;26:101200. <https://doi.org/10.1016/j.iot.2024.101200>

33. Rewal P, Singh M, Mishra D, Pursharthi K, Mishra A. Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices. *J Inf Secur Appl.* 2023;75:103505. <https://doi.org/10.1016/j.jisa.2023.103505>
34. Chaudhary D, Dadsena PK, Padmavathi A, Hassan MM, Alkhamees BF, Kumar U. Anonymous quantum safe construction of three party authentication and key agreement protocol for mobile devices. *IEEE Access.* 2024;12:74572–74585. <https://doi.org/10.1109/ACCESS.2024.3404232>
35. Mishra D, Singh M, Rewal P, Pursharthi K, Kumar N, Barnawi A, Rathore RS. Quantum-safe secure and authorized communication protocol for internet of drones. *IEEE Trans Veh Technol.* 2023;72(12):16499–16507. <https://doi.org/10.1109/TVT.2023.3292169>
36. Islam SH, Basu S. PB-3PAKA: password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments. *J Inf Secur Appl.* 2021;63:103026. <https://doi.org/10.1016/j.jisa.2021.103026>
37. Ding R, Cheng C, Qin Y. Further analysis and improvements of a lattice-based anonymous PAKE scheme. *IEEE Syst J.* 2022;16(3):5035–5043. <https://doi.org/10.1109/JSYST.2022.3161264>
38. Zhang J, Zhang Z, Ding J, Snook M, Dagdelen O. Authenticated key exchange from ideal lattices. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer; 2015. p. 719–751. https://doi.org/10.1007/978-3-662-46803-6_24
39. Parameswarath RP, Wang C, Sikdar, B. Quantum-Safe Authentication Protocol for IoT-Enabled Transportation Systems. *IEEE Transactions on Industry Applications.* 2025; 62(1):729 – 741. <https://doi.org/10.1109/TIA.2025.3585861>
40. Wang D, Cheng H , Wang P, Huang X, Jian, G. (2017). Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security.* 2017; 12(11): 2776-2791. <https://ieeexplore.ieee.org/abstract/document/7961213/>