sensors

Article

Anonymous and Efficient Chaotic Map-Based Authentication Protocol for Industrial Internet of Things

Dake Zeng [1], Akhtar Badshah [2,*], Shanshan Tu [1], Xin Ai [1], Hisham Alasmary [3], Muhammad Waqas [4,*] and Muhammad Taimoor Khan [4]

[1] College of Computer Science, Beijing University of Technology, Beijing 100124, China; pete.zeng@akuhome.com (D.Z.); sstu@bjut.edu.cn (S.T.); aixin1022@gmail.com (X.A.)
[2] Department of Software Engineering, University of Malakand, Dir Lower, Chakdara 18800, Pakistan
[3] Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia; alasmary@kku.edu.sa
[4] School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London SE10 9LS, UK; m.khan@greenwich.ac.uk
* Correspondence: akhtarbadshah@uom.edu.pk (A.B.); engr.waqas2079@gmail.com (M.W.)

Abstract

The exponential growth of Internet infrastructure and the widespread adoption of smart sensing devices have empowered industrial personnel to conduct remote, real-time data analysis within the Industrial Internet of Things (IIoT) framework. However, transmitting this real-time data over public channels raises significant security and privacy concerns. To prevent unauthorized access, user authentication mechanisms are crucial in the IIoT environment. To mitigate security vulnerabilities within IIoT environments, a novel user authentication and key agreement protocol is proposed. The protocol is designed to restrict service access exclusively to authorized users of designated smart sensing devices. By incorporating cryptographic hash functions, chaotic maps, Physical Unclonable Functions (PUFs), and fuzzy extractors, the protocol enhances security and functional integrity. PUFs provide robust protection against tampering and cloning, while fuzzy extractors facilitate secure biometric verification through the integration of smart cards, passwords, and personal biometrics. Moreover, the protocol accommodates dynamic device enrollment, password and biometric updates, and smart card revocation. A rigorous formal security analysis employing the Real-or-Random (ROR) model was conducted to validate session key security. Complementary informal security analysis was performed to assess resistance to a broad spectrum of attacks. Comparative performance evaluations unequivocally demonstrate the protocol's superior efficiency and security in comparison to existing benchmarks.

Keywords: Industrial Internet of Things; security; privacy; authentication; key agreement; biometrics

Check for updates

Academic Editor: Antonio Corradi

Received: 12 November 2025
Revised: 12 December 2025
Accepted: 15 December 2025
Published: 18 December 2025

1. Introduction

The Industrial Internet of Things (IIoT) employs numerous devices for real-time data sensing, transmission, and analysis, improving industrial control, productivity, and product quality while reducing costs and resource consumption. However, as IIoT expands with Industry 5.0, the rapid proliferation and interaction of IoT devices introduce critical security challenges [1–4]. Many IIoT devices incorporate lightweight software and hardware to minimize costs, limiting their ability to support robust security comparable to traditional Internet environments. This limitation creates vulnerabilities, making devices prone to attacks. Furthermore, deployment in remote, unmonitored environments exacerbates risks, exposing critical industrial processes to malicious disruptions [5–7].

In general, a typical IIoT system comprises smart sensing devices, such as temperature and humidity sensors, vibration sensors, and RFID tags, alongside users and gateway nodes, collectively forming a wireless sensor network (WSN) [8–10]. However, the dependence on wireless communication technologies exposes IIoT systems to substantial security vulnerabilities, as malicious adversaries can compromise system security through attacks such as eavesdropping and message tampering [11–13]. Authentication and key agreement mechanisms are regarded as one of the most effective countermeasures against these threats [14–16]. Nevertheless, given the intrinsic resource limitations of sensing devices, authentication protocols deployed in such settings must carefully balance computational efficiency with robust security guarantees.

1.1. Existing Research and Motivation

Recently, numerous research efforts have focused on developing anonymous and lightweight authenticated key agreement protocols, aiming to enhance security, privacy, and efficiency in IIoT environments [17–20]. Turkanović et al. [21] proposed a mutual authentication scheme that relied on a pre-shared cryptographic key between the sensor node and the user. Their scheme employed simple hash and XOR operations to accommodate the resource constraints of WSN. However, a subsequent analysis by Tai et al. [22] demonstrated that the protocol developed by Turkanović et al. fails to provide anonymity and is susceptible to sensor-capture attacks. Chen et al. [23] introduced an authentication and key agreement protocol for industrial control systems. Nevertheless, their proposed solution suffers from high computational and communication overheads, susceptibility to ephemeral secret leakage (ESL) attacks, and a lack of perfect forward secrecy. Shuai et al. [24] proposed an authentication protocol utilizing the Rabin cryptosystem. However, their protocol remains vulnerable to offline guessing, user impersonation, privileged insider, eavesdropping, and stolen smart card attacks. Gong et al. [25] propose a certificateless anonymous mutual authentication scheme ensuring privacy, traceability, and scalability for IoT. The schemes presented in [26,27] rely on a clock synchronization assumption, limiting their applicability in practical IIoT deployments. Zhai et al. [28] proposed a lightweight authentication protocol that combines blockchain with chaotic maps to enable mutual authentication between smart devices and edge gateways in IIoT systems. However, their protocol relies solely on the security of the chaotic-map discrete logarithm problem and fails to provide anonymity and untraceability. Aman et al. [29] proposed a PUF-based device authentication protocol for IoT systems. However, subsequent analyses demonstrated that the protocol is susceptible to replay attacks and non-invasive physical attacks [30], and it does not account for the influence of environmental noise on PUF responses.

Rafique et al. [31] addressed a critical challenge in the IIoT concerning secure data transmission. Their research proposed a multifactor authentication key agreement protocol that balanced robust security with resource limitations. This protocol utilized bitwise XOR, cryptographic hash functions, and symmetric cryptography to create a secure system tailored for resource-constrained environments, ensuring high-level security while enabling remote access to sensing devices. However, [32] identified that Rafique’s protocol is vulnerable to attacks involving the loss of smart cards or devices. Eldefrawy et al. [33] proposed a user authentication method for IIoT systems, focusing on computational and communication efficiency. While this protocol was efficient, it did not provide mutual authentication between users and the smart devices or sensor nodes within the system. Harishma et al. [34] developed a method for securing data transmission in cyber-physical systems with heterogeneous components. Nevertheless, their approach was found vulnerable to the ESL attack under the Canetti and Krawczyk (CK) adversary model [35]. Additionally, the protocol did not support the dynamic incorporation of new IoT smart devices, limiting its practical application. Chen et al. [36] designed a key agreement and user authentication system for IoT environments. Although their protocol was efficient in terms of computational and communication costs, it was vulnerable to insider attacks, node-capturing attacks, and gateway node-bypassing attacks, and lacked untraceability [37–39].

In summary, although several commendable AKA protocols have been proposed for IIoT systems, most remain impractical for deployment on resource-constrained smart sensing devices due to their substantial resource overhead [40–42]. Furthermore, their incomplete security and functional guarantees further diminish their applicability in real-world deployments. Addressing these limitations is critical to enabling the efficient and secure operation of IIoT environments. The comparative summary is given in Table 1.

Table 1. Existing Authentication Protocols: A Comparative Summary.

Protocol	Cryptographic Primitives	Descriptions	Limitations
Turkanović et al. [21]	• Cryptographic hash function	• Lightweight • Dynamic addition of sensor nodes • Free password update	• No PUF-based physical security provided • Strict clock synchronization required • Anonymity not provided • No resistance to sensor capture attacks
Chen et al. [23]	• Cryptographic hash function • Elliptic curve cryptography	• Supporting node revocation • No strict clock synchronization	• No PUF-based physical security provided • Excessive system overhead • Lack of perfect forward secrecy • No resistance to ephemeral secret leakage attacks
Shuai et al. [24]	• Cryptographic hash function • Rabin cryptosystem	• Free password update • Dynamic addition of sensor nodes	• No PUF-based physical security provided • Strict clock synchronization required • No resistance to offline guessing and user impersonation attacks • No resistance to privileged insider and eavesdropping attacks • No resistance to stolen smart-card attacks
Zhai et al. [28]	• Cryptographic hash function • Chaotic Map	• Lightweight • Supporting cross-domain authentication	• No PUF-based physical security provided • Strict clock synchronization required • Lack of anonymity and untraceability
Aman et al. [29]	• Cryptographic hash function • PUF	• Lightweight • Providing PUF-based physical security • No strict clock synchronization	• No resistance to replay and non-invasive physical attacks • The neglect of PUF noise impact
Rafique et al. [31]	• Cryptographic hash function • AES symmetric encryption/decryption	• Lightweight	• No PUF-based physical security provided • No resistance to smart-card and device theft attacks • Strict clock synchronization required
Eldefrawy et al. [33]	• Cryptographic hash function	• Lightweight • Free password update • Dynamic addition/revocation of sensor nodes • No strict clock synchronization	• No PUF-based physical security provided • Lack of mutual authentication
Harishma et al. [34]	• Cryptographic hash function • Identity-based encryption • PUF	• Providing PUF-based physical security • No strict clock synchronization	• No resistance to ephemeral secret leakage attacks
Chen et al. [36]	• Cryptographic hash function • Elliptic curve cryptography	• Lightweight	• No PUF-based physical security provided • No resistance to insider and node-capturing attacks • Strict clock synchronization required • Lack of untraceability

1.2. Contribution

This paper proposes an anonymous and efficient Chaotic Map-based authentication protocol for the IIoT environment that ensures both efficiency and security. The main contributions are as follows:

1. We propose a novel chaotic map-based mutual authentication and session key agreement protocol for the IIoT environment, where independent session keys are generated between users and smart sensing devices to ensure secure communications.
2. We design the proposed protocol by integrating a one-way cryptographic hash function, chaotic map, physical unclonable function (PUF), and fuzzy extractor to enhance security and functional integrity. The PUF component provides robust protection against tampering and cloning attacks on the smart sensing device.

3. We conduct a formal security analysis of the protocol using the real-or-random (ROR) model to rigorously assess and ensure session key security. Additionally, we provide an informal security analysis to demonstrate resistance to a broad spectrum of attacks.
4. A rigorous comparative performance evaluation was conducted to assess the security, functionality, communication overhead, and computational efficiency of the proposed protocol in relation to existing benchmarks. The study clearly demonstrates the proposed protocol's superior efficiency and enhanced security features compared to existing protocols.

1.3. Novelty

The proposed protocol introduces several key innovations that collectively address critical gaps in existing IIoT authentication protocols. First, it pioneers a hybrid security architecture that uniquely integrates authentication based on chaotic-map and lightweight cryptographic hashing with PUF-based device identity verification, achieving both algorithmic robustness and hardware-rooted trust. Unlike existing protocols that rely solely on lightweight cryptography or PUF, the proposed protocol further incorporates a fuzzy extractor to mitigate the impact of environmental disturbances on PUF responses in industrial settings. In addition, the protocol establishes an independent session key directly between the user and the device without involving a central entity, ensuring genuine end-to-end confidentiality for sensitive information. By avoiding the use of timestamps for replay protection, it also eliminates the dependence on strict clock synchronization. Finally, under a unified framework, the proposed protocol achieves the most comprehensive set of security features with exceptionally high efficiency, as demonstrated in its multidimensional comparison with existing protocols in terms of computational, runtime, communication, and storage overhead, an advantage unmatched by prior works. These contributions collectively position the proposed protocol as a holistic authentication framework expressly tailored to the distinct security and performance demands of IIoT environments.

1.4. Paper Organization

The remainder of this paper is organized as follows: Section 2 presents the background, including network and threat models, and foundational concepts. The proposed protocol is detailed in Section 3. A comprehensive security analysis is provided in Section 4 and formal security analysis is presented in Section 5. Section 6 offers a performance comparison with existing state-of-the-art protocols. Finally, the paper is concluded in Section 7.

2. Background

This section provides a comprehensive overview of the authentication model, threat model, and cryptographic foundations.

2.1. System Models

This section delineates the authentication model and threat models employed in the design of the proposed authentication and key agreement protocol.

2.1.1. Authentication Model

As illustrated in Figure 1, the proposed Internet of Things (IoT)-based smart sensing system for industrial monitoring aims to establish intelligent factories through the integration of IoT sensors and a robust authentication framework. This integrated system is designed to optimize supply chain, production, safety, and energy management. To address the challenges of securing real-time data transmission from resource-constrained IoT devices operating in inherently insecure wireless environments, a comprehensive authentication model is essential. This model safeguards data integrity and confidentiality while enabling authorized industrial personnel to securely access and utilize real-time data from smart sensing devices. A trusted registration authority (RA) establishes secure communication by registering all network entities and issuing cryptographic credentials. Authorized users, authenticated by the gateway node, can access the collected data through these gateway nodes, ensuring secure and reliable data transmission. The subsequent sections detail the proposed authentication protocol, which provides a secure and efficient mechanism for user authentication and data protection within the smart manufacturing environment [43–45].

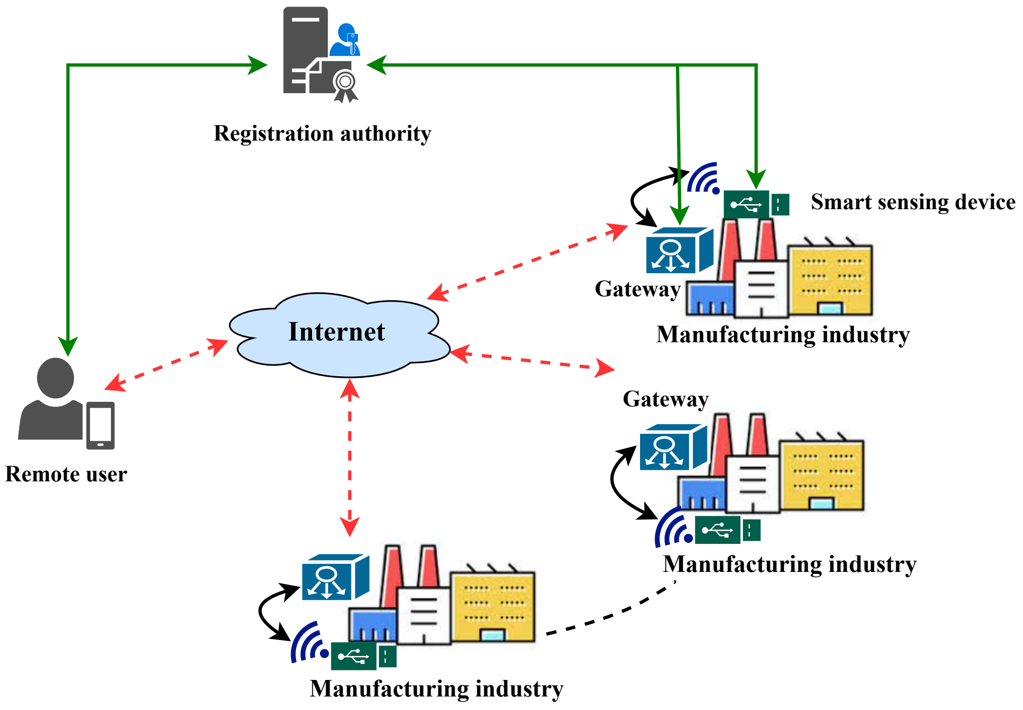

Figure 1. IoT-based industrial monitoring system architecture.

2.1.2. Threat Model

To assess the security of the proposed protocol, this study adopts the widely recognized Dolev-Yao (DY) threat model [46,47]. Within this framework, an adversary $\mathcal{A}$ is endowed with the capability to intercept, modify, delete, or inject spurious messages during communication over an insecure channel. Given the inherently hostile nature of the IIoT environment, IoT sensing devices are susceptible to physical compromise by $\mathcal{A}$, both internal and external. Such breaches can lead to the unauthorized acquisition of sensitive credentials stored within these devices. Furthermore, an $\mathcal{A}$ may physically acquire an authorized user's smart card, thereby enabling sophisticated power analysis attacks to compromise the stored secret credentials. Leveraging these credentials, the $\mathcal{A}$ can potentially extract sensitive user data, including identity, password, and biometric information. This compromised position facilitates a range of attack vectors, such as privileged insider attacks, replay attacks, man-in-the-middle (MitM) attacks, and impersonation attacks. Recently, the CK adversary model [48–50] has emerged as the *de facto* standard for evaluating the security of authenticated key agreement protocols. This model extends the DY adversary's capabilities by granting the attacker the power to compromise secret credentials, session states, and session keys. Consequently, a robust user authentication protocol for the IIoT must safeguard against the exposure of sensitive information, limiting the adversary's ability to compromise additional credentials even if some secrets are compromised. In addition, gateway nodes are protected by hardware security boundaries (for example, HSM/TEE) to safely store $\{ID_{GWN_k}, X_{GWN_k}\}$. However, their databases are only semi-trusted and may be subject to physical capture attacks, allowing $\mathcal{A}$ to extract the hashed credentials stored within them. The symbols used in this paper and respective description is given in Table 2.

Table 2. Symbols and their definitions.

Symbol	**Definition**
GWN_k	kth GWN
X_{GWN_k}	Secret parameter of GWN_k
ID_{GWN_k}	Identity of GWN_k
SD_j	jth smart sensing deivce
C_j	Challenge component for SD_j
DR_j	Response component for SD_j
X_j	Secret parameter bound to SD_j
Dk_j, s_j	Cryptographic key and auxiliary data of SD_j
U_i	ith user
ID_i, PW_i	Identity and password of U_i
B_i	Biometric template of U_i
Bk_i, s_i	Cryptographic key and auxiliary data of U_i
SC_i, SCN_i	Smart card of U_i and its number of U_i
$X_i, Y_i, Auth_i$	Secret parameter bound to U_i
PID_i	Pseudo-identity of U_i
$PUF_j(\cdot)$	Physical unclonable function associated with SD_j
RA	The trusted registration authority
SCN	Smart card number
SK	Session key
TID_i	Temporary identity of U_i
TID_i^o/TID_i^n	The Old/new temporary identitity
X_{RA}	Secret key of RA
$h(\cdot)$	Cryptographic hash function
Gen$(\cdot)$	Fuzzy extractor generation function
Rep$(\cdot)$	Fuzzy extractor reproduction function

2.2. Preliminaries

This section provides a foundational overview of relevant cryptographic primitives, including PUFs, fuzzy extractors, and chaotic maps.

2.2.1. Physical Unclonable Function

The PUF is a cryptographic mechanism that relies on challenge-response pairs (CRPs). The core process of a PUF involves generating a unique response (output) when given a specific challenge (input) [51,52], which can be formalized with $DR_j = PUF_j(C_j)$. Here, C_j denotes the unique challenge presented to different devices (SD_j), while DR_j signifies the corresponding unique response. PUFs function as irreversible mathematical functions, deriving cryptographic secrets from inherent physical variations found in integrated circuits (ICs). These variations, stemming from the manufacturing process, ensure that each IC generates a unique response to specific challenges. PUFs provide a cost-effective solution with minimal computational requirements and energy consumption, making them highly suitable for lightweight and physically secure cryptographic applications. They are advanced circuit primitives used to generate secret keys for cryptographic operations. For example, the Static Random Access Memory (SRAM) PUF is widely adopted due to SRAM's critical role in electronic control units (ECUs). The SRAM PUF exploits random disparities in SRAM threshold voltages to establish a distinctive digital fingerprint for each device. Unlike traditional methods of key storage, the secret key is dynamically regenerated from the SRAM PUF within a secure environment. This approach ensures that even in the event of a memory breach, the secret key remains protected and immune to compromise. Importantly, any attempt to manipulate or tamper with a PUF alters the device's behavior, thereby invalidating the PUF and facilitating tamper detection. However, in real-world noisy environments, factors such as temperature variations, voltage fluctuations, and device aging may cause a PUF to produce different responses even when supplied with the same challenge, which can result in the erroneous rejection of secret parameters generated solely from PUF responses during authentication. To address this important issue, we integrate a fuzzy extractor with the PUF. During the registration phase, the device first generates a response $DR_j = PUF_j(C_j)$. Subsequently, in an authentication session, the device regenerates a new response $DR'_j = PUF_j(C_j)$. When the Hamming distance $DIST(DR_j, DR'_j)$ falls within the tolerable error threshold t, the fuzzy extractor is used to reconcile (DR_j, DR'_j), thereby ensuring stable key derivation.

2.2.2. Fuzzy Extractors

Fuzzy extractors are cryptographic mechanisms designed to derive stable and secure cryptographic keys from inherently noisy data sources, such as biometric measurements or PUFs [53]. By mitigating the impact of noise and variations inherent in these data types, fuzzy extractors enhance the robustness and security of recognition systems. A fuzzy extractor consists of two primary components:

1. **Generation** $(Bk_i, s_i) = Gen(B_i)$: Given a unique biometric or PUF measurement B_i of entity i, the generation algorithm produces a cryptographic key Bk_i and public auxiliary data s_i.
2. **Reproduction** $Bk'_i = \mathrm{Rep}(B'_i, s_i)$: Utilizing the public auxiliary data s_i and a noisy measurement B'_i approximating Bk_i, the reproduction algorithm reconstructs the original key Bk'_i. For sufficiently similar input measurements (typically within a predetermined Hamming distance), Bk'_i will be identical to Bk_i.

2.2.3. Chaotic Map

Chaotic map encryption leverages the intrinsic unpredictability and extreme sensitivity to initial conditions of chaotic systems for cryptographic applications [54–56]. This method employs the enhanced Chebyshev polynomial, defined as $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \mod lp$, where lp is a large prime number, $n \geq 2$, and $x \in (-\infty, \infty)$. Alternatively, $T_n(x) = \cos(n \cdot \arccos(x))$. The semigroup property of Chebyshev polynomials, expressed as $T_n(T_m(x)) = T_m(T_n(x)) = T_{mn}(x) \mod lp$, forms the basis for the encryption process. The security of this encryption scheme relies on the computational intractability of the chaotic map discrete logarithm problem (CMDLP). An adversary's ability to solve this problem within a given time frame is quantified as $Adv_{\mathcal{A}}^{CMDLP}(rt) = Pr[\mathcal{A}(x_1, x_2) = m \mid T_m(x_1) = x_2 \mod lp]$, where $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ and $r \in \mathbb{Z}_p^*$. Some existing studies [57–59] indicate that, $\mathcal{A}$ may be able to solve the CMDLP problem in polynomial time, thereby compromising the security of cryptographic systems based on Chebyshev polynomials. However, this attack capability requires $\mathcal{A}$ to obtain both$T_m(x)$ and x simultaneously. To counteract this, the proposed protocol maintains the encryption of x

during transmission. The protocol is grounded in the computational Diffie-Hellman (CDH) problem adapted to the chaotic map domain. Even with knowledge of $T_m(x)$ and $T_n(x)$, computing $T_m(T_n(x)) = T_n(T_m(x)) \mod lp$ remains computationally infeasible for an adversary, providing the cryptographic strength of the system.

2.3. Design Objectives

The protocol aims to achieve the following design goals:

- **Anonymity:** Protect user privacy by ensuring that their identities cannot be inferred from messages, maintaining anonymity throughout communications.
- **Mutual Authentication:** Verify the authenticity of users, gateways, and smart sensing devices to establish trust within the network.
- **Session Key Agreement:** Securely establish session keys between users and smart sensing devices, independent of the gateway, for protected communication.
- **Unlinkability:** Prevent the correlation of intercepted messages with specific users, preserving message anonymity.
- **Forward Security:** Protect the confidentiality of past communications by ensuring the compromise of current session keys does not affect the security of previous sessions.
- **Resistance to Common Attacks:** Strengthen the protocol's defenses against replay, offline password guessing, and impersonation attacks to enhance overall IIoT network security.

3. The Proposed Protocol

This section presents a secure protocol tailored for IIoT environments to guarantee that only authorized users can access smart sensing devices. The protocol incorporates SHA-256 hashing, XOR operations, and PUFs to safeguard smart sensing devices against physical tampering. Furthermore, to address replay attacks, the protocol avoids reliance on clock synchronization among network entities. The subsequent subsections outline the various phases of the proposed protocol.

3.1. Pre-Deployment Phase

In this phase, the RA is responsible for registering the GWNs and smart sensing devices prior to their deployment. A secure channel is assumed for this one-time setup, as commonly adopted in IIoT protocols, typically ensured through administrator-supervised or trusted in-person registration.

3.1.1. GWN Registration

The RA performs the following operations to register a GWN denoted as GWN_k.

1. Step GR-1: The RA selects a distinct and unique identity, ID_{GWN_k}, and a secret parameter, X_{GWN_k}. These are then transmitted securely to GWN_k through a secure channel.
2. Step GR-2: GWN_k securely stores the received parameters, $\{ID_{GWN_k}, X_{GWN_k}\}$, in its secure database.

3.1.2. Smart Sensing Device Registration

To register a smart sensing device, such as SD_j, the RA follows these steps:

1. Step DR-1: The RA starts the registration process by generating a unique challenge parameter, C_j. This parameter is securely transmitted to SD_j over a secure channel.
2. Step DR-2: Upon receiving C_j from the RA, SD_j uses a $PUF(\cdot)$ to generate the response parameter DR_j. This response is then securely sent back to the RA.
3. Step DR-3: The RA selects an identity ID_j and sends the pair C_j, DR_j to the appropriate gateway node GWN_k through a secure channel.

4. Step DR-4: Upon receiving the $\{ID_j, C_j, DR_j\}$ from the RA, GWN_k generates a random number r_j and computes $X_j = h(ID_j \| X_{GWN_k} \| r_j)$. It then stores the parameters $\{ID_j, r_j, C_j, DR_j\}$ for SD_j in its database and sends X_j back to the RA through a secure channel.
5. Step DR-5: The RA finally transmits the parameters $\{ID_j, X_j\}$ to SD_j, which then stores these parameters in its memory.

3.1.3. User Registration

To ensure secure communication with an SD that has been accessed, U_i must securely register at the RA through the following steps.

1. Step UR-1: User U_i submits their registration request to the RA, along with their identity ID_i, through a secure channel.
2. Step UR-2: The RA then assesses whether the request originates from a legitimate user. It computes $h(ID_i \| X_{RA} \| T_i)$ using a hash function, where X_{RA} is the master secret key of the RA and T_i is the registration timestamp. Based on the computed hash, it consults the database for the corresponding status. If the status indicates that the user is already registered (*status* $= 1$), the registration process ceases. Otherwise, the RA proceeds with registration and picks a unique temporary identity TID_i, a unique pseudo-identity PID_i, and a random number r_i. The user's smart card SC_i is configured with $<TID_i, r_i, PID_i, SCN_i, h(\cdot)>$. The RA stores $<ID_i, SCN_i, T_i, [h(ID_i \| X_{RA} \| T_i), status]>$ in its database and forwards the following parameters to the corresponding gateway node GWN_k as $<TID_i^o = TID_i, TID_i^n = TID_i, PID_i, r_i, SCN_i>$ via a secure channel and dispatches SC_i to U_i.
3. Step UR-3: Upon receiving $<TID_i^o = TID_i, TID_i^n = TID_i, PID_i, r_i, SCN_i>$ from the RA, GWN_k computes $X_{i-g} = (PID_i \| r_i) \oplus h(SCN_i \| X_{GWN_k})$. GWN_k then stores the parameters as $<TID_i^o = TID_i, TID_i^n = TID_i, X_{i-g}, SCN_i>$ in its database.
4. Step UR-4: Upon receipt, U_i inserts SC_i into a card reader, reads the biometric data B_i, and enters their ID_i and password PW_i and picks a random number n_i. The smart card then proceeds to compute the following:

$$(Bk_i, s_i) = \text{Gen}(B_i), X_i = (s_i \| n_i) \oplus h(ID_i \| PW_i)$$
$$Auth_i = h(ID_i \| PW_i \| Bk_i \| n_i)$$
$$Y_i = (PID_i \| r_i) \oplus h(ID_i \| PW_i \| Bk_i)$$

   where $\text{Gen}(\cdot)$ is a fuzzy extractor generator function and $Auth_i$ is an authentication parameter. Finally, the smart card replaces the parameters $<r_i, PID_i>$ with $<X_i, Y_i, Auth_i>$ and stores $<TID_i, X_i, Y_i, Auth_i, SCN_i, h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot)>$.

3.2. Login Phase

A legitimate user, for instance, U_i, must first authenticate themselves using their credentials and smart card. The specific steps are as follows:

1. Step L-1: User U_i inserts their designated smart card and subsequently enters their unique identity (ID_i'), password (PW_i'), and provides biometric data (B_i') via the designated sensor.
2. Step L-2: Based on the provided inputs, the following computations are carried out:

$$(s_i' \parallel n_i') = X_i \oplus h(ID_i' \parallel PW_i'), Bk_i' = \text{Rep}(B_i', s_i')$$
$$Auth_i' = h(ID_i' \parallel PW_i' \parallel Bk_i' \parallel n_i')$$

The $Auth_i'$ is verified against $Auth_i$ as follows: $Auth_i' \overset{?}{=} Auth_i$ If the verification fails, the login procedure is aborted. If the verification succeeds, the sign-in process is considered successful, and the following parameters are computed: $(PID_i\|r_i) = Y_i \oplus h(ID_i'\|PW_i'\|Bk_i')$.

3.3. Authentication and Key Agreement Phase

The following steps are essential for completing this phase, as illustrated in Figure 2.

User U_i	**Gateway node** GWN_k	**Smart sensing device** SD_j
$\{TID_i,\ X_i,\ Y_i,\ Auth_i\}$	$\{TID_i^o/TID_i^n,\ SCN_i,\ X_{i-g}\}$,	$\{ID_j,\ X_j,\ PUF(\cdot),\ h(\cdot)\}$
	$\{ID_j,\ r_j,\ (C_j,\ DR_j)\}$	
	Step A-2:	
Step L-1:	Retrieve SCN_i if either match TID_i^o/TID_i^n	
Input $< ID_i', PW_i', B_i' >$	Compute $(PID_i \Vert r_i) = X_{i-g} \oplus h(SCN_i \Vert X_{GWN_k})$	
Step L-2:	$(ID_j' \Vert N_u') = X_1 \oplus h(r_i \Vert PID_i \Vert TID_i)$	
Compute $(s_i' \Vert n_i') = X_i \oplus h(ID_i' \Vert PW_i')$	$Auth_1' = h(ID_j' \Vert N_u' \Vert r_i \Vert PID_i \Vert TID_i \Vert M_1)$	
$Bk_i' = \text{Rep}(B_i', s_i')$	Check $Auth_1' \overset{?}{=} Auth_1$, if so	
$Auth_i' = h(ID_i' \Vert PW_i' \Vert Bk_i' \Vert n_i')$		Step A-4:
Check if $Auth_i' \overset{?}{=} Auth_i$ sign-in successful		$(N_u' \Vert N_g') = X_2 \oplus h(X_j \Vert ID_j \Vert M_1)$
Compute $(PID_i \Vert r_i) = Y_i \oplus h(ID_i' \Vert PW_i' \Vert Bk_i')$	Step A-3:	$(PID_i' \Vert C_j') = X_3 \oplus h(X_j \Vert ID_j \Vert X_2)$
	Retrieve $r_j, (C_j, DR_j)$ corresponding to ID_j	$Auth_2' = h(N_u' \Vert N_g' \Vert PID_i' \Vert C_j' \Vert M_1)$
Step A-1:	Choose random N_g	Check $Auth_2' \overset{?}{=} Auth_2$, if so
Select random nonce m	Compute $X_j = h(ID_j \Vert X_{GWN_k} \Vert r_j)$	
$M_1 = T_m(PID_i)$	$X_2 = (N_u \Vert N_g) \oplus h(X_j \Vert ID_j \Vert M_1)$	Step A-5:
Choose ID_j, random N_u	$X_3 = (PID_i \Vert C_j) \oplus h(X_j \Vert ID_j \Vert X_2)$	$DR_j = PUF_j(C_j)$
Compute $X_1 = (ID_j \Vert N_u) \oplus h(r_i \Vert PID_i \Vert TID_i)$	$Auth_2 = h(N_u \Vert N_g \Vert PID_i \Vert C_j \Vert M_1)$	$(DK_j, s_j) = \text{Gen}(DR_j)$
$Auth_1 = h(ID_j \Vert N_u \Vert r_i \Vert PID_i \Vert TID_i \Vert M_1)$	$\xrightarrow{Msg_2=<M_1, X_2, X_3, Auth_2>}$	Choose random N_j, n
$\xrightarrow{Msg_1=<TID_i, X_1, M_1, Auth_1>}$		Compute $M_2 = T_n(PID_i)$
	Step A-6:	$psk = T_n(M_1) = T_n(T_m(PID_i))$
	$(N_j' \Vert s_j') = X_4 \oplus h(ID_j \Vert X_j \Vert N_g)$	$SK_{j-i} = h(PID_i \Vert ID_j \Vert psk \Vert N_u \Vert N_j)$
	$DK_j' = \text{Rep}(DR_j, s_j')$	$X_4 = (N_j \Vert s_j) \oplus h(ID_j \Vert X_j \Vert N_g)$
Step A-7:	$Auth_3' = h(N_j' \Vert s_j' \Vert M_2 \Vert ID_j \Vert X_j \Vert N_g \Vert DK_j')$	$Auth_3 = h(N_j \Vert s_j \Vert M_2 \Vert ID_j \Vert X_j \Vert N_g \Vert DK_j)$
$(N_j' \Vert TID_i') = X_5 \oplus h(PID_i \Vert r_i \Vert N_u)$	Check $Auth_3' \overset{?}{=} Auth_3$, if so	$\xleftarrow{Msg_3=<X_4, M_2, Auth_3>}$
$Auth_4' = h(N_j' \Vert TID_i' \Vert M_2 \Vert ID_j \Vert PID_i \Vert r_i \Vert N_u)$	Pick TID_i'	
Check $Auth_4' \overset{?}{=} Auth_4$, if so	Update $TID_i^o = TID_i, TID_i^n = TID_i'$	
$psk = T_m(M_2) = T_m(T_n(PID_i))$	$X_5 = (N_j \Vert TID_i') \oplus h(PID_i \Vert r_i \Vert N_u)$	
$SK_{i-j} = h(PID_i \Vert ID_j \Vert psk \Vert N_u \Vert N_j')$	$Auth_4 = h(N_j \Vert TID_i' \Vert M_2 \Vert ID_j \Vert PID_i \Vert r_i \Vert N_u)$	
Update $TID_i = TID_i'$	$\xleftarrow{Msg_4=<X_5, M_2, Auth_4>}$	

Figure 2. Login and authentication phases.

1. Step A-1: U_i generates a random nonce m and uses it to compute $M_1 = T_m(PID_i)$ as a chaotic-map variable. Then, U_i chooses ID_j and selects random nonce N_u, and calculates $X_1 = (ID_j\|N_u) \oplus h(r_i\|PID_i\|TID_i)$. Additionally, U_i computes $Auth_1 = h(ID_j \| N_u \| r_i \| PID_i \| TID_i \| M_1)$. Afterward, U_i constructs $Msg_1 =< TID_i, X_1, M_1, Auth_1 >$ and sends it to the gateway node GWN_k through an open insecure channel.
2. Step A-2: Upon receiving Msg_1 from U_i, GWN_k searches for SCN_i, which should match either TID_i^o or TID_i^n. The following computations are then performed:

$$
\begin{aligned}
(PID_i\|r_i) &= X_{i-g} \oplus h(SCN_i\|X_{GWN_k}) \\
(ID_j'\|N_u') &= X_1 \oplus h(r_i\|PID_i\|TID_i) \\
Auth_1' &= h(ID_j'\|N_u'\|r_i\|PID_i\|TID_i\|M_1)
\end{aligned}
$$

The calculated $Auth_1'$ is checked for equality with $Auth_1$, i.e., $Auth_1' \overset{?}{=} Auth_1$. If they do not match, the process is terminated. If they match, the procedure continues to the next steps.

3. Step A-3: For ID_j, GWN_k retrieves r_j, (C_j, DR_j) and selects a random nonce N_g. The following computations are then performed:

$$
\begin{aligned}
X_j &= h(ID_j \| X_{GWN_k} \| r_j) \\
X_2 &= (N_u \| N_g) \oplus h(X_j \| ID_j \| M_1) \\
X_3 &= (PID_i \| C_j) \oplus h(X_j \| ID_j \| X_2) \\
Auth_2 &= h(N_u \| N_g \| PID_i \| C_j \| M_1)
\end{aligned}
$$

   Finally, GWN_k constructs Msg_2 as $Msg_2 =< M_1, X_2, X_3, Auth_2 >$ and transmits it to SD_j through an open insecure channel.
4. Step A-4: Upon receiving Msg_2 from GWN_k, SD_j retrieves ID_j and X_j and performs the following:

$$
\begin{aligned}
(N'_u \| N'_g) &= X_2 \oplus h(X_j \| ID_j \| M_1) \\
(PID'_i \| C'_j) &= X_3 \oplus h(X_j \| ID_j \| X_2) \\
Auth'_2 &= h(N'_u \| N'_g \| PID'_i \| C'_j \| M_1)
\end{aligned}
$$

   The resulting $Auth'_2$ is compared with $Auth_2$ to verify if $Auth'_2 \overset{?}{=} Auth_2$. If they do not match, the process is aborted. If they match, the procedure proceeds to the next step.
5. Step A-5: SD_j inputs the challenge C_j into the $PUF(\cdot)$ function to obtain a device-specific but potentially noisy response, $DR_j = PUF_j(C_j)$. Due to the inherent variability of PUFs caused by environmental and hardware factors, SD_j employs a fuzzy extractor to ensure reliable key derivation. Specifically, the fuzzy extractor generates a stable and reproducible secret key DK_j and corresponding helper data s_j, denoted as $(DK_j, s_j) = \text{Gen}(DR_j)$. This process ensures that even if DR_j slightly fluctuates under different conditions, the same DK_j can be reliably recovered using the helper data during key reconstruction. Next, SD_j selects a random integral string n and a random nonce N_j, and computes the following: $M_2 = T_n(PID_i)$, $psk = T_n(M_1) = T_n(T_m(PID_i))$, $SK_{j-i} = h(PID_i \parallel ID_j \parallel psk \parallel N_u \parallel N_j)$, $X_4 = (N_j \parallel s_j) \oplus h(ID_j \parallel X_j \parallel N_g)$, and $Auth_3 = h(N_j \parallel s_j \parallel M_2 \parallel ID_j \parallel X_j \parallel N_g \parallel DK_j)$, where SK_{j-i} is the generated secret session key between SD_j and U_i. Finally, SD_j constructs Msg_3 as $Msg_3 = \langle X_4, M_2, Auth_3 \rangle$ and sends it to GWN_k via an open insecure channel.
6. Step A-6: Upon receiving Msg_3 from SD_j, GWN_k performs the following computations:

$$
\begin{aligned}
(N'_j \| s'_j) &= X_4 \oplus h(ID_j \| X_j \| N_g), DK'_j = \text{Rep}(DR_j, s'_j) \\
Auth'_3 &= h(N'_j \| s'_j \| M_2 \| ID_j \| X_j \| N_g \| DK'_j)
\end{aligned}
$$

   GWN_k accepts the message if $Auth'_3 \overset{?}{=} Auth_3$ holds true. Next, GWN_k picks a unique temporary identity TID'_i and updates the old and new temporary identities as $TID^o_i = TID_i$ and $TID^n_i = TID'_i$. Then, GWN_k computes $X_5 = (N_j \| TID'_i) \oplus h(PID_i \| r_i \| N_u)$ and $Auth_4 = h(N_j \| TID'_i \| M_2 \| ID_j \| PID_i \| r_i \| N_u)$. Finally, GWN_k sends $Msg_4 =< X_5, M_2, Auth_4 >$ to U_i to U_i via an open insecure channel.
7. Step A-7: Upon receiving Msg_4 from GWN_k, U_i computes $(N'_j \| TID'_i) = X_5 \oplus h(PID_i \| r_i \| N_u)$ and $Auth'_4 = h(N'_j \| TID'_i \| M_2 \| ID_j \| PID_i \| r_i \| N_u)$. U_i accepts if $Auth'_4 \overset{?}{=} Auth_4$ holds true. Next, U_i computes $psk = T_m(M_2) = T_m(T_n(PID_i))$, $SK_{UC} = h(PID_i \| r_i \| N_c \| N_u)$ and $SK_{i-j} = h(PID_i \parallel ID_j \parallel psk \parallel N_u \parallel N'_j)$. Finally, U_i stores the session key and updates the temporary identity as $TID_i = TID'_i$.

3.4. Password and Biometrics Update Phase

The proposed protocol allows users to locally update their passwords without needing to interact with the RA. Users can modify both their password (PW_i) and biometric information (B_i) by following these steps:

1. Step PBU-1: The legitimate user U_i inserts their smart card and authenticates themselves with their credentials to update their password and biometrics. The following calculations are then performed using the provided inputs:

$$(s_i' \parallel n_i') = X_i \oplus h(ID_i' \parallel PW_i'), Bk_i' = \text{Rep}(B_i', s_i')$$
$$Auth_i' = h(ID_i' \parallel PW_i' \parallel Bk_i' \parallel n_i')$$

   The $Auth_i'$ is verified against $Auth_i$ as follows: $Auth_i' \stackrel{?}{=} Auth_i$ If the verification fails, the login procedure is aborted. If the verification succeeds, U_i can reset the password and update the biometrics data, and the following parameters are computed:$(PID_i \| r_i) = Y_i \oplus h(ID_i' \| PW_i' \| Bk_i')$.
2. Step PBU-1: Next, U_i inputs new information (PW_i^n and B_i^n), and the card executes the following computations:

$$(Bk_i^n, s_i^n) = \text{Gen}(B_i^n), X_i^n = (s_i^n \| n_i) \oplus h(ID_i \| PW_i^n)$$
$$Auth_i^n = h(ID_i \| PW_i^n \| Bk_i^n \| n_i^n)$$
$$Y_i^n = (PID_i \| r_i) \oplus h(ID_i \| PW_i^n \| Bk_i^n)$$

   Subsequently, the system updates the existing parameters in the smart card $< TID_i, X_i, Y_i, Auth_i, SCN_i, h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot) >$ with the new configurations $< TID_i, X_i^n, Y_i^n, Auth_i^n, SCN_i, h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot) >$.

3.5. Smart Card Revocation

The protocol enables smart card replacement without altering the user's identity. If user U_i loses or has their smart card stolen, they can request a replacement from the RA using the following procedure:

1. Step SCR-1: Initially, U_i securely transmits their identity (ID_i), credentials, and a replacement request to RA. The RA assesses these credentials and the validity of the request. Following confirmation, RA issues a replacement smart card, SCN_i^n, a unique pseudo-identity PID_i^n, and a random number r_i^n. Consequently, the freshly configured smart card SC_i^n is configured with $<TID_i^n, r_i^n, PID_i^n, SCN_i^n, h(\cdot)>$. The RA stores $<ID_i, SCN_i^n, T_i^n, [h(ID_i \| X_{RA}^n \| T_i^n), status]>$ in its database and forwards the following parameters to the corresponding gateway node GWN_k as $<TID_i^o = TID_i^n, TID_i^n = TID_i^n, PID_i^n, r_i^n, SCN_i^n>$ via a secure channel and dispatches SC_i^n to U_i.
2. Step SCR-2: Furthermore, upon receipt of the card, U_i inserts it into a card reader and performs the same steps as outlined in Subsection 3.1.3 *("3. User Registration")*. Finally, the smart card stores $< TID_i^n, X_i^n, Y_i^n, Auth_i^n, SCN_i^n, h(\cdot), \text{Gen}(\cdot), \text{Rep}(\cdot) >$.

3.6. Dynamic Smart Device Addition Phase

This phase is essential for subsequent deployment of new smart sensing devices. To integrate a new device, denoted as SD_j^{new}, the RA performs the following offline steps:

1. Step DA-1: The RA starts the registration process by generating a unique challenge parameter, C_j^{new}. This parameter is securely transmitted to SD_j^{new} over a secure channel.

2. Step DA-2: Upon receiving C_j^{new} from the RA, SD_j^{new} uses a $PUF(\cdot)$ to generate the response parameter DR_j^{new}. This response is then securely sent back to the RA.
3. Step DA-3: The RA selects an identity ID_j^{new} and sends the pair $\{C_j^{new}, DR_j^{new}\}$ to the appropriate gateway node GWN_k through a secure channel.
4. Step DA-4: Upon receiving the $\{ID_j^{new}, C_j^{new}, DR_j^{new}\}$ from the RA, GWN_k generates a random number r_j^{new} and computes $X_j^{new} = h(ID_j^{new} \| X_{GWN_k} \| r_j^{new})$. It then stores the parameters $\{ID_j^{new}, r_j^{new}, C_j^{new}, DR_j^{new}\}$ for SD_j^{new} in its database and sends X_j^{new} back to the RA through a secure channel.
5. Step DA-5: The RA finally transmits the parameters $\{ID_j^{new}, X_j^{new}\}$ to SD_j^{new}, which then stores these parameters in its memory.

Once SD_j^{new} is deployed, the GWN_k/RA will notify all registered users, enabling them to access services from SD_j^{new} if needed.

4. Informal Security Analysis

In this section, security and functional attributes of the proposed protocol are examined through a descriptive analysis.

4.1. Smart Sensing Device Capture Attack

The physical unclonability of PUF technology safeguards smart sensing devices, such as SD_j, against unauthorized access. Any tampering with the PUF-based sensor will significantly alter its response or disable the device entirely. Consequently, extracting sensitive information from the PUF-equipped SD_j becomes extremely challenging for potential attackers.

4.2. Gateway Node Capture Attack

Assume that $\mathcal{A}$ is a privileged insider within the IIoT system who is capable of physically compromising a gateway node and thereby obtaining the secret credentials stored on it, namely $\{TID_i^o/TID_i^n,\ SCN_i,\ X_{i-g}\}$ and $\{ID_j,\ r_j,\ (C_j,\ DR_j)\}$, where $X_{i-g} = (PID_i \| r_i) \oplus h(SCN_i \| X_{GWN_k})$. However, due to the one-way property of the cryptographic hash function, the secret credentials PID_i and r_i, which are bound to U_i, remain protected since X_{GWN_k} is stored within a hardware-isolated region that is inaccessible to $\mathcal{A}$. Moreover, GWN_k does not store the identity ID_i, password PW_i, or biometric template B_i of U_i. Given only $\{TID_i^o/TID_i^n,\ SCN_i\}$ and the hashed credential X_{i-g}, even if $\mathcal{A}$ simultaneously acquires the user's smart card, no valid information can be inferred that would compromise the security of the session key.

4.3. Anonymity and Untraceability

Considering the threat model described in Section 2.1.2, suppose that $\mathcal{A}$ intercepts the transmitted messages Msg_1 to Msg_4 during the login and authentication processes, as illustrated in Figure 2, over an insecure public channel. These intercepted messages contain secret random nonces, which are crucial for maintaining confidentiality. This confidentiality makes it extremely difficult for $\mathcal{A}$ to determine critical identifiers such as the user's identity (ID_i), the user's pseudo-identity (PID_i), or the smart sensing device's identifier (ID_j). This mechanism ensures the anonymity of users and their associated smart sensing devices within the network. Additionally, the use of unique random nonces for each session prevents $\mathcal{A}$ from tracking users across different sessions. Even if $\mathcal{A}$ identifies the temporary identities of U_i from the intercepted messages, the protocol's design requires frequent renewal of these identifiers to new temporary ones (TID_i^n) with each session. This ensures that tracking users and devices over time is not possible. Consequently, the proposed protocol effectively guarantees the anonymity and untraceability of participants, establishing a robust security foundation within the system.

4.4. De-Synchronization Attack

In our protocol, users are assigned unique temporary identities (TID_i) during the registration phase. The gateway node (GWN_k) stores the parameters $< TID_i^o = TID_i, TID_i^n = TID_i, X_{i-g}, SCN_i >$ for each user U_i. To counter de-synchronization attacks, the protocol maintains both the old and new temporary identities in the gateway node's database, ensuring synchronization between U_i and GWN_k. This design ensures the protocol's functionality remains unaffected even if the final confirmation message is delayed or lost, thereby providing robust protection against de-synchronization attacks.

4.5. Replay Attack

The proposed protocol safeguards against replay attacks where an adversary $\mathcal{A}$ intercepts and attempts to reuse previously exchanged messages (Msg_1 to Msg_4), where $Msg_1 =< TID_i, X_1, M_1, Auth_1 >$, $Msg_2 =< M_1, X_2, X_3, Auth_2 >$, $Msg_3 =< X_4, M_2, Auth_3 >$, and $Msg_4 =< X_5, M_2, Auth_4 >$. The strength of the protocol lies in its use of unpredictable, one-time random values (nonces) within each transmitted message. These nonces guarantee that each message is unique to a specific session, rendering them useless in any subsequent attempt by $\mathcal{A}$ to replay them. This effectively prevents replay attacks and ensures the protocol's robustness.

4.6. MitM Attack

Even if an attacker ($\mathcal{A}$) intercepts messages (Msg_1 to Msg_4) intending to tamper with them and impersonate a user, the protocol's design thwarts such attempts. Modifying the initial message ($Msg_1 =< TID_i, X_1, M_1, Auth_1 >$) requires access to secret information like N_u, ID_j, and PID_i, which are beyond $\mathcal{A}$'s reach. Similarly, for messages Msg_2, Msg_3, and Msg_4, the protocol relies on confidential data like PID_i, DR_j, ID_j, and others to generate valid nonces. Without this information, $\mathcal{A}$ cannot forge messages that appear legitimate. This ensures the protocol's resistance to MitM attacks.

4.7. Mutual Authentication

The proposed protocol establishes mutual authentication between user U_i, gateway GWN_k, and sensor SD_j via the subsequent three steps: (1) GWN_k validates U_i by checking SCN_i and $Auth_1$; (2) SD_j authenticates GWN_k directly by verifying $Auth_2$ and indirectly by matching PID_i from the transcripts with U_i; (3) U_i verifies $Auth_4$ to confirm GWN_k directly and also indirectly confirms SD_j, establishing the session key independently of the gateway node.

4.8. Session Key Security

The proposed protocol ensures session key security through a multi-layered approach. An adversary $\mathcal{A}$ cannot bypass local user authentication or access sensitive credentials such as PID_i and r_i stored on the smart card in encrypted form. This protection is due to the requirement of all three factors (password, biometric data, and smart card) for login. Additionally, the session key $SK_{j-i} = h(PID_i \| ID_j \| psk \| N_u \| N_j)$ is derived using a combination of a one-way hash function and unique, secret parameters specific to each user, device, and communication session. These parameters include PID_i, psk, N_j, N_u, and ID_j. The one-way nature of the hash function ensures that even without knowledge of these secret parameters, deriving the session key remains computationally infeasible. This approach safeguards the confidentiality and integrity of communication within the protocol.

4.9. Perfect Forward Secrecy

In the proposed protocol, the session key $SK_{j-i} = h(PID_i \| ID_j \| psk \| N_u \| N_j)$ established between U_i and SD_j is derived through a hybrid mechanism that combines random values, long-term secrets, and chaotic-map–based secrets, thereby ensuring perfect forward secrecy. First, these parameters are unique to each session and each communicating entity, preventing $\mathcal{A}$ from correlating information across different sessions or participants. Second, in every authentication session, the random values used to compute the chaotic-map encryption outputs are randomly selected, and PID_i remains hidden from $\mathcal{A}$. Due to the inherent hardness of the CMDLP, $\mathcal{A}$ can not derive $psk = T_n(T_m(PID_i))$ that satisfies $M_1 = T_m(PID_i)$ and $M_2 = T_n(PID_i)$. Moreover, the random values and long-term secrets that contribute to SK_{j-i} are protected by the cryptographic hash function and thus remain concealed from $\mathcal{A}$. Overall, through this hybrid derivation mechanism that incorporates both long-term and ephemeral secrets, the proposed protocol ensures that the security of SK_{j-i} does not rely on any single cryptographic component, thereby providing perfect forward secrecy.

4.10. ESL Attack

In the proposed protocol, the session key $SK_{j-i} = h(PID_i \| ID_j \| psk \| N_u \| N_j)$ is generated between U_i and SD_j with the help of GWN_k. This key relies on session-specific and entity-specific secret information as well as random parameters. To assess the resilience of the key against ESL attacks, we examine two distinct scenarios:

1. *Scenario 1*: If the attacker $\mathcal{A}$ manages to obtain the temporary secretsTID_i, N_g, N_u, N_j, and psk, the hash function $h(\cdot)$'s collision-resistant properties make it difficult for $\mathcal{A}$ to derive the session key without also having access to the long-term secrets ID_i, PW_i, PID_i, ID_j, X_j, and DK_j.
2. *Scenario 2*: Even if $\mathcal{A}$ gains possession of the long-term secrets ID_i, PW_i, PID_i, ID_j, X_j, and DK_j, the absence of short-term secrets still makes it impractical for $\mathcal{A}$ to compute the session key.

4.11. Stolen Smartcards and Privileged-Insiders Attacks

These scenarios explore the implications of a lost or stolen smartcard, SC_i, belonging to a registered user. The registration process commences with user U_i transmitting ID_j to the RA via a secure channel. Suppose a privileged-insider within the RA, denoted as adversary $\mathcal{A}$, has access to the information $< TID_i, r_i, PID_i, SCN_i >$. Subsequent to registration completion, it is assumed that adversary $\mathcal{A}$ has gained possession of user U_i 's stolen smartcard, denoted as SC_i and then extracted the data $< TID_i, X_i, Y_i, Auth_i, SCN_i >$ from SC_i using power analysis attacks. Extracting or guessing ID_i from the values $(X_i, Y_i, Auth_i)$ without knowing PW_i and B_i of U_i is computationally infeasible, as PW_i and B_i are protected by a collision-resistant hash function $h(\cdot)$. This demonstrates that identity guessing attacks are challenging for $\mathcal{A}$. Similarly, deriving PW_i from $(X_i, Y_i, Auth_i)$ without ID_i and B_i is computationally infeasible due to $h(\cdot)$'s collision resistance. Additionally, it is computationally infeasible for the adversary $\mathcal{A}$ to derive PW_i from the extracted parameters. Thus, the system is robust against offline password guessing attacks, even in the case of stolen smartcards or privileged insider threats.

4.12. Resistance to Modeling Attacks

The proposed protocol prevents modeling attacks by avoiding challenge-response transmission over insecure channels. Using collision-resistant hash function $h(\cdot)$ and XOR to encrypt secret parameters, adversary $\mathcal{A}$ cannot extract useful information from intercepted messages, ensuring security with minimal overhead.

4.13. Impersonation Attacks

To mitigate impersonation attacks, three scenarios are examined: *Case I (User Impersonation Attack):* Suppose an adversary $\mathcal{A}$ intercepts the message $Msg_1 =< TID_i, X_1, M_1, Auth_1 >$. Using this information, $\mathcal{A}$ tries to impersonate the user by crafting a modified message $Msg_1^{\mathcal{A}} =< TID_i^{\mathcal{A}}, X_1^{\mathcal{A}}, M_1^{\mathcal{A}}, Auth_1^{\mathcal{A}} >$ to deceive others. However, without knowledge of PID_i, r_i, and ID_j, $\mathcal{A}$ cannot produce a valid Msg_1, rendering it incapable of establishing a credible communication with GWN_k and thus preventing successful user impersonation.

Case II (Gateway Node Impersonation Attack): Consider the scenario where $\mathcal{A}$ intercepts $Msg_2 =< M_1, X_2, X_3, Auth_2 >$. To impersonate a gateway node, $\mathcal{A}$ would need to fabricate Msg_2 to convince a smart sensing device of its authenticity. However, without knowledge of ID_j and X_j, $\mathcal{A}$ cannot compute X_2, X_3, and $Auth_2$. Therefore, $\mathcal{A}$ cannot create a valid Msg_2 to impersonate a gateway node, and similarly, $\mathcal{A}$ cannot compute Msg_4 to deceive U_i. Thus, $\mathcal{A}$ fails to impersonate a gateway node.

Case III (Smart Sensing Device Impersonation Attack): When $\mathcal{A}$ intercepts the message $Msg_3 =< X_4, M_2, Auth_3 >$ from SD_j to GWN_k, impersonating SD_j requires crafting a credible Msg_3. Nonetheless, lacking crucial information like ID_j, X_j, DK_j, N_j, s_j, N_g, and PID_i, $\mathcal{A}$ fails to generate a legitimate Msg_3. This inability to forge a valid message demonstrates the protocol's effectiveness in thwarting attempts at impersonating the smart sensing device, underscoring its security against such types of attacks.

5. Formal Security Analysis

This section analyzes the security of the proposed protocol using the ROR model, formally proving session key security. Before presenting the session key proof (Theorem 1), we introduce essential ROR model primitives.

Participants. The proposed protocol involves three primary participants: the user U_i, the gateway node GWN_k, and the smart sensing device SD_j. Instances of these participants are represented as $\Omega_U^{t_1}$, $\Omega_{GWN}^{t_2}$, and $\Omega_{SD}^{t_3}$, corresponding to the t_1th instance of U_i, the t_2th instance of GWN_k, and the t_3th instance of SD_j, respectively. Each instance is treated as an oracle, and Ω^t is occasionally used to denote the tth instance of any given entity.

Partnership. Entities Ω^{t_1} and Ω^{t_2} in the protocol are considered partners if they meet the following criteria: (1) Both entities have reached an acceptance state; (2) They share the same session identifier SID; and (3) A session key has been established between them.

Freshness. An entity Ω^t is considered to be fresh if its established session key has not been compromised by the adversary $\mathcal{A}$. An entity Ω^t is deemed fresh if its established session key remains uncompromised by the adversary $\mathcal{A}$.

Adversary. In the ROR model, the adversary $\mathcal{A}$'s capabilities are represented by conducting specific oracle queries: In the ROR model, the capabilities of the adversary $\mathcal{A}$ are represented through specific oracle queries defined in Table 3. It is important to emphasize that, while $\mathcal{A}$ is granted the ability to extract the secret credentials stored in a user's smart card via the $CorruptSmartcard(\Omega^t)$ query, deriving either the long-term or ephemeral secrets from the hashed credentials remains well beyond $\mathcal{A}$'s capability.

Definition 1 (Semantic Security)**.** *In the context of the RoR model, the adversary $\mathcal{A}$ is tasked with distinguishing between the actual session key and a random string during the Test query. The adversary $\mathcal{A}$ attempts to correctly guess the value of b to succeed in this challenge. The advantage of the adversary in compromising the semantic security of the proposed protocol P is denoted as $Adv_{\mathcal{A}}^P$ and is defined by the equation $Adv_{\mathcal{A}}^P = 2 \cdot \Pr[b' = b] - 1$. If there exists a sufficiently small function ϵ such that $Adv_{\mathcal{A}}^P < \epsilon$, then the proposed protocol P is considered to be semantically secure.*

Table 3. Adversary queries in the ROR model.

Query	Functionality
$Execute(\Omega^{t_1}_{U_i}, \Omega^{t_2}_{GWN_k}, \Omega^{t_3}_{SD_j}$	This query allows the adversary $\mathcal{A}$ to passively observe and collect messages exchanged among $\Omega^{t_1}_{U_i}, \Omega^{t_2}_{GWN_k}$, and $\Omega^{t_3}_{SD_j}$, facilitating the evaluation of forward secrecy.
$Send(\Omega^t, msg)$	This query enables the simulation of active attacks, such as impersonation and replay. The adversary $\mathcal{A}$ sends a message msg to Ω^t and receives the corresponding response.
$CorruptSmartcard(\Omega^t)$	Upon executing this query, the adversary $\mathcal{A}$ obtains complete access to the credentials stored on the compromised smart card SC_i belonging to the legitimate registered user U_i.
$Reveal(\Omega^{t_1}_{U_i})$	This query discloses the actual session key if $\Omega^{t_1}_{U_i}$ has been accepted.
$Test(\Omega^t)$	This query returns the real session key when $b = 1$, or a random string of the same length when $b =$ .. If $\mathcal{A}$ can consistently guess the value of b correctly, it signifies a successful compromise of the session key's semantic security.

Definition 2 (Security of PUF)**.** *For two secure functions $PUF(\cdot)_1$ and $PUF(\cdot)_2$, given any inputs c_1 and c_2 in $0, 1^k$, the probability that the Hamming distance $HD(PUF(c_1)_1, PUF(c_2)_2)$ exceeds d is $1 - \epsilon$. Here, d represents the fault tolerance level.*

Definition 3 (CMDLP)**.** *The advantage of an adversary $\mathcal{A}$ in solving the Chaotic Map Discrete Logarithm Problem (CMDLP) within a runtime of rt is considered negligible if it satisfies $Adv^{CMDLP}_{\mathcal{A}}(rt) < \epsilon$, where ϵ is a negligible function.*

Theorem 1. *For the proposed protocol P, the adversary $\mathcal{A}$ operates within polynomial time, aiming to compromise semantic security. Let q_h denote the number of hash queries, q_p the number of PUF queries, and q_s the number of send queries. Additionally, let l represent the length of the biometric data, D be the uniformly distributed password dictionary, and the PUF has a key length of $|PUF|$. Combining the aforementioned definitions, $Adv^{CMDLP}_{A}(rt)$ denotes the advantage of solving the CMDLP problem within time rt. Thus, we have:*

$$Adv^P_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2\left(\frac{q_s}{2^l \cdot |D|} + Adv^{CMDLP}_{\mathcal{A}}(rt)\right).$$

We establish the provable security of the proposed protocol by defining a series of games $Game_i \ (i = 0, 1, 2, \ldots, 5)$. Specifically, in our protocol P, let $\Pr[\text{suc}_i]$ denote the event where the adversary $\mathcal{A}$ successfully guesses the value of b in the *Test* query within the game $Game_i$.

***Game*₀.** This game models an attack scenario where the adversary $\mathcal{A}$ targets the protocol P. At the start, $\mathcal{A}$ is tasked with determining the value of the bit b. The advantage of $\mathcal{A}$ in this context is given by:

$$Adv^P_{\mathcal{A}} = |2\Pr[\text{suc}_0] - 1|, \tag{1}$$

where $\Pr[\text{succ}_0]$ represents the probability of $\mathcal{A}$ succeeding in guessing b.

***Game*₁.** In this game, $\mathcal{A}$ simulates an eavesdropping attack by performing multiple *Execute* queries. Assume $\mathcal{A}$ intercepts all messages Msg_1 through Msg_4 transmitted within the protocol. To determine the session key $SK_{j-i} = h(PID_i \| ID_j \| psk \| N_u \| N_j)$, $\mathcal{A}$ must have access to specific long-term and short-term secret parameters. However, based on the informal security analysis, obtaining these parameters is impractical for $\mathcal{A}$. Therefore, the probability of $\mathcal{A}$ successfully winning $\boldsymbol{Game_1}$ remains unchanged. Consequently, we have $\Pr[\text{suc}_0] = \Pr[\text{suc}_1]$.

$\boldsymbol{Game_2}$**.** The primary distinction between $Game_2$ and its predecessor, $Game_1$, lies in the incorporation of simulated *Send* and *Hash* queries. In $Game_2$, $\mathcal{A}$ performs an active attack, trying to deceive a participant into accepting forged messages. Although the adversary may conduct multiple hash queries on Msg_1, Msg_2, Msg_3, and Msg_4 to identify potential collisions, the inclusion of random nonces, unique identifiers (PID_i and ID_j), and long-term secrets associated with each message renders this a highly improbable event. As a result, the likelihood of the adversary encountering a collision during *Send* queries is negligible. Moreover, applying the birthday paradox reinforces this assertion by indicating that

$$|\Pr[\text{suc}_1] - \Pr[\text{suc}_2]| \leq \frac{q_h^2}{2|Hash|} \tag{2}$$

$\boldsymbol{Game_3}$**.** In $Game_3$, we simulate PUF queries. Applying the definition of secure PUF functions (Definition 2), we derive the following inequality:

$$|\Pr[\text{suc}_2] - \Pr[\text{suc}_3]| \leq \frac{q_p^2}{2|PUF|} \tag{3}$$

$\boldsymbol{Game_4}$**.** The transition from $Game_3$ to $Game_4$ involves the addition of the $CorruptSmartcard(\Omega)$ query. By utilizing this query, the adversary $\mathcal{A}$ will acquire the credentials TID_i, X_i, Y_i, and $Auth_i$. To correctly identify ID_i and PW_i for U_i from $(s_i' \parallel n_i') = X_i \oplus h(ID_i' \parallel PW_i')$ and $Bk_i' = \text{Rep}(B_i', s_i')$, $\mathcal{A}$ must have both the secret credential n_i' and the biometric key Bk_i'. By capping the number of unsuccessful identity/password or biometric verifications, the system upholds the following condition:

$$|\Pr[\text{suc}_3] - \Pr[\text{suc}_4]| \leq \frac{q_s}{2^l \cdot |D|} \tag{4}$$

$\boldsymbol{Game_5}$**.** The final game involves $\mathcal{A}$ endeavoring to compute the shared session key SK_{i-j} between U_i and SD_j by exploiting intercepted communications and simultaneously solving the CMDLP instance (cf. Definition 3). To calculate the session key $SK_{j-i} = h(PID_i \| ID_j \| psk \| N_u \| N_j)$, $\mathcal{A}$ needs ID_j and psk, where $psk = T_n(T_m(PID_i)) = T_m(T_n(PID_i)) = T_{mn}(PID_i) \mod lp$. It is evident that even with knowledge of PID_i, computing $T_n(PID_i) \mod lp$ necessitates access to n. Similarly, determining $T_m(PID_i) \mod lp$ requires knowledge of m, despite possessing PID_i. Consequently, $\mathcal{A}$ must solve the CMDLP within a runtime of at most rt to obtain the session key SK_{i-j}. Therefore,

$$|\Pr[\text{suc}_4] - \Pr[\text{suc}_5]| \leq Adv_{\mathcal{A}}^{CMDLP}(rt) \tag{5}$$

Taking into account all the games described earlier, after all the oracle queries have been executed, the adversary $\mathcal{A}$ does not gain any extra advantage in correctly guessing the bit b in the *Test* query. Consequently, $\Pr[\text{suc}_5] = \frac{1}{2}$. Additionally, based on the calculations, we get:

$$
\begin{aligned}
|\Pr[suc_0] - \Pr[suc_5]| = \left|\Pr[suc_0] - \frac{1}{2}\right| = \left|\Pr[suc_1] - \frac{1}{2}\right| \\
&\leq |\Pr[suc_1] - \Pr[suc_2]| \\
&\quad + |\Pr[suc_2] - \Pr[suc_3]| \\
&\quad + |\Pr[suc_3] - \Pr[suc_4]| \\
&\quad + |\Pr[suc_4] - \Pr[suc_5]| \\
&= \frac{q_h^2}{2|Hash|} + \frac{q_p^2}{2|PUF|} \\
&\quad + \frac{q_s}{2^l \cdot |D|} + Adv_{\mathcal{A}}^{CMDLP}(rt)
\end{aligned} \tag{6}
$$

Combining the results yields:

$$
\begin{aligned}
Adv_{\mathcal{A}}^{P} &= |2\Pr[suc_0] - 1| = 2|\Pr[suc_0] - \frac{1}{2}| \\
&\leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2\left(\frac{q_s}{2^l \cdot |D|} + Adv_{\mathcal{A}}^{CMDLP}(rt)\right)
\end{aligned}
$$

6. Performance Evaluation

In this section, we compare the proposed protocol with five state-of-the-art schemes [60–64], focusing on computational overhead, communication overhead, and security and functionality features.

6.1. Computational Overhead

This section analyzes the computational cost of our proposed protocol compared to existing ones. Core cryptographic operations underpinning user login and authentication are considered. We exclude basic operations like XOR and concatenation from the analysis. For the efficiency evaluation, we implemented all cryptographic primitives involved in the proposed protocol and the baseline protocols using Python 3.9.13, leveraging libraries such as *hashlib*, *pypuf*, and *ecpy*. We then measured their average execution times, including hash function (T_{H}), ECC point multiplication (T_{EM}), PUF (T_{PUF}), fuzzy extractor ($T_{\mathsf{FE}} \approx T_{\mathsf{EM}}$), and chaotic map ($T_{\mathsf{CM}}$). In addition, to support the testing procedures, a Raspberry Pi 4 equipped with 2 GiB of memory and running Raspberry Pi OS (32-bit) was used as the resource-constrained Platform I to simulate smart sensing devices, whereas a 64-bit Windows 10 machine with 8 GiB of memory and an Intel® Core™ i5-8300H CPU @ 2.30GHz was employed as the resource-rich Platform II to emulate gateway nodes and user devices. For each platform and cryptographic primitive, 1000 test runs were conducted. The resulting average execution times in milliseconds are tabulated in Table 4.

Table 4. Average execution time of various primitives.

Operation	**Platform-I**	**Platform-II**
Hash function (T_{H})	0.007 ms	0.001 ms
ECC point multiplication (T_{EM})	6.549 ms	0.846 ms
ECC point addition (T_{EA})	0.273 ms	0.002 ms
Physical unclonable function (T_{PUF})	0.5 µs	–
Fuzzy extractor ($T_{\mathsf{FE}} \approx T_{\mathsf{EM}}$)	6.549 ms	0.846 ms
Chaotic map (T_{CM})	0.102 ms	0.019 ms

Table 5 summarizes the computational overheads of different protocols. Our protocol incurs a computation overhead of approximately $6T_{\mathrm{H}} + T_{\mathrm{PUF}} + T_{\mathrm{FE}} + 2T_{\mathrm{CM}} \approx 6.7955$ on resource-constrained smart sensing devices (Platform-I). This is lower than the overhead of protocols proposed by the benchmark protocols of Hammad et al. [60], Wazid et al. [61], and Sutrala et al. [63]. This indicates better suitability for resource-constrained settings. While the total computation overhead of our protocol (8.5445 ms) exceeds that of some existing solutions Yang et al. [62] and Srinivas et al. [64], it offers superior security and functionality features, as detailed in Table 5. This trade-off between efficiency and security is a crucial consideration for real-world deployments.

Table 5. Computation overheads comparison (ms).

Protocol	User	GWN / Server	Smart Sensing Device	Total Overhead
Hammad et al. [60]	$T_{PUF} + 2T_{EM} + 7T_H \approx 1.699$	$2T_{EM} + 9T_H \approx 1.701$	$T_{PUF} + 2T_{EM} + 5T_H \approx 13.1335$	16.5335
Wazid et al. [61]	$T_{FE} + 4T_{EM} + T_{EA} + 19T_H \approx 4.251$	$5T_{EM} + T_{EA} + T_H \approx 4.233$	$4T_{EM} + T_{EA} + 12T_H \approx 26.553$	35.037
Yang et al. [62]	$10T_H \approx 0.01$	$19T_H \approx 0.019$	$8T_H \approx 0.056$	0.085
Sutrala et al. [63]	$T_{FE} + 5T_{EM} + 2T_{EA} + 16T_H \approx 5.096$	$3T_{EM} + 2T_{EA} + 9T_H \approx 2.551$	$4T_{EM} + T_{EA} + 8T_H \approx 26.525$	34.172
Srinivas et al. [64]	$15T_{\mathrm{H}} + T_{\mathrm{FE}} + 2T_{\mathrm{CM}} \approx 0.899$	$10T_{\mathrm{H}} \approx 0.01$	$6T_{\mathrm{H}} + 2T_{\mathrm{CM}} \approx 0.246$	1.155
Our Proposed	$8T_{\mathrm{H}} + T_{\mathrm{FE}} + 2T_{\mathrm{CM}} \approx 0.892$	$11T_{\mathrm{H}} + T_{\mathrm{FE}} \approx 0.857$	$6T_{\mathrm{H}} + T_{\mathrm{PUF}} + T_{\mathrm{FE}} + 2T_{\mathrm{CM}} \approx 6.7955$	8.5445

6.2. Runtime Comparison

In this subsection, a comprehensive performance evaluation is conducted by implementing the complete workflows of both the proposed protocol and its baseline protocols in Python on a Windows 10 experimental machine equipped with an Intel® Core™ i5-8300H CPU @ 2.30GHz processor and 8 GiB of memory. Each protocol underwent 100 runs, and the average execution time for the integrated login and AKA phase is recorded; comparative results appear in Figure 3. Empirical data indicate that the proposed protocol attains an average runtime of 1013.86 ms, outperforming the protocols of Hammad et al. [60] (1016.38 ms), Sutrala et al. [63] (1024.63 ms), Wazid et al. [61] (1041.15 ms), and Yang et al. [62] (1019.19 ms). Although its overhead is marginally higher than that of the protocol proposed by Srinivas et al. [64], this slight increase is offset by the additional security features and functional enhancements delivered by the proposed protocol (see Table 8).

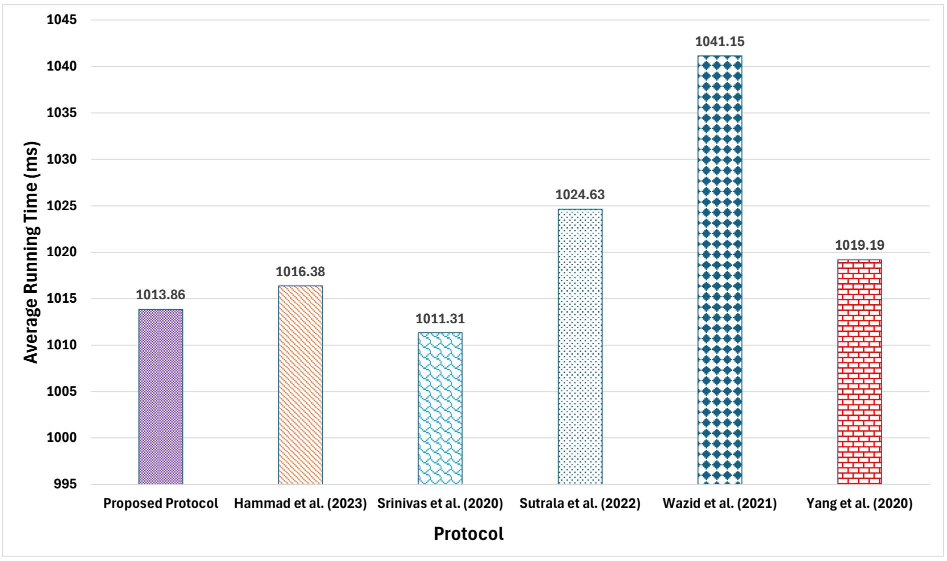

Figure 3. Comparison of computation overheads.

6.3. Communication Overhead

Table 6 presents a comparison of the communication overhead of various protocols, highlighting the number of bits needed for message exchanges. The communication overhead is quantified in bits for each protocol. To ensure a fair evaluation, the following assumptions are made: timestamps are considered to be 32 bits; identities, random numbers, chaotic map-based encryption outputs, and PUF responses are considered to be 128 bits; hash digest outputs are 256 bits; and ECC points are 320 bits. The analysis reveals that the total communication overhead of the proposed protocol is 2944 bits, which is lower than the 3616 bits, 3360 bits, 5376 bits, and 3200 bits required by the benchmark protocols Hammad et al. [60], Wazid et al. [61], Yang et al. [62], and Sutrala et al. [63], respectively. Thus, the proposed protocol is more resource-efficient. Moreover, while the proposed protocol's communication overhead is slightly higher than that of Srinivas et al. [64], this increase is justified by its enhanced and comprehensive security features, as outlined in Table 8.

Table 6. Communication overheads comparison.

Protocol	No. of Messages	Total Overhead (Bits)
Hammad et al. [60]	5	3616
Wazid et al. [61]	3	3360
Yang et al. [62]	6	5376
Sutrala et al. [63]	3	3200
Srinivas et al. [64]	3	1856
Our Proposed	4	2944

6.4. Storage Overhead

Building on the experimental configuration in Section 6.3, Table 7 presents a systematic comparison of the storage overhead incurred by the proposed protocol versus baseline protocols. The focus is on the secret credentials that user devices, gateways/servers, and smart sensing units must store after initialization and registration to support subsequent AKA operations. To illustrate scalability, Table 7 also presents the storage cost of each entity as a function of the number of smart sensing devices, denoted by N. For resource-constrained smart sensing devices, the proposed protocol requires only 384 bits of local storage-substantially less than the 1280 bits of Wazid et al. [61] and the 1984 bits of Sutrala et al. [63], and equal to the requirements of Srinivas et al. [64] and Yang et al. [62]. Given the enhanced security and functionality delivered by our protocol (see Table 8), this minimal storage footprint is justified. From a system-wide perspective, the total storage burden introduced by the proposed protocol grows linearly as $1024N + 2016$ bits-markedly lower than Wazid et al.'s [61] $1792N + 4320$ bits, Yang et al.'s [62] $1152N + 1280$ bits, and Sutrala et al.'s [63] $2624N + 4576$ bits. While slightly higher than the schemes of Hammad et al. [60] and Srinivas et al. [64], the proposed protocol offers stronger security guarantees and richer functionality (refer to Table 8), making this trade-off reasonable and acceptable.

Table 7. Comparison of storage overheads (bits).

Protocol	User	GWN/Server	Smart Sensing Device	Overall Overheads
Hammad et al. [60]	640	$1024 + 128n$	512	$640N + 2176$
Wazid et al. [61]	$256N + 2656$	$256N + 1664$	1280	$1792N + 4320$
Yang et al. [62]	$128N + 640$	$640N + 640$	384	$1152N + 1280$
Sutrala et al. [63]	$256N + 2592$	$384N + 1984$	1984	$2624N + 4576$
Srinivas et al. [64]	$128N + 992$	$384N + 128$	384	$896N + 1120$
Our Proposed	$128N + 1120$	$640N + 896$	384	$1024N + 2016$

Note: N: Number of Smart Sensing Devices.

Table 8. Security and functionality features comparison.

Feature	[60]	[61]	[62]	[63]	[64]	Our
$\mathcal{F}_{\infty}$	✓	×	×	×	×	✓
$\mathcal{F}_{\in}$	✓	✓	×	✓	✓	✓
$\mathcal{F}_{\ni}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\triangle}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\bigtriangledown}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{/}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\shortmid}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\forall}$	×	✓	✓	✓	✓	✓
$\mathcal{F}_{\exists}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\infty\prime}$	×	×	✓	×	×	✓
$\mathcal{F}_{\infty\infty}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\infty\in}$	✓	✓	✓	✓	✓	✓
$\mathcal{F}_{\infty\ni}$	✓	✓	✓	✓	✓	✓

Note: $\mathcal{F}_{\infty}$: Smart sensing device capture attack; $\mathcal{F}_{\in}$: Anonymity; $\mathcal{F}_{\ni}$: Untraceability; $\mathcal{F}_{\triangle}$: De-synchronization attack; $\mathcal{F}_{\bigtriangledown}$: Replay attack; $\mathcal{F}_{:/}$ MitM attack; $\mathcal{F}_{\shortmid}$: Mutual authentication; $\mathcal{F}_{\forall}$: Independent session key establishment; $\mathcal{F}_{\exists}$: Perfect forward secrecy; $\mathcal{F}_{\infty\prime}$: No clock synchronization; $\mathcal{F}_{\infty\infty}$: ESL attack; $\mathcal{F}_{\infty\in}$: Stolen smartcard and privileged-insider attacks; and $\mathcal{F}_{\infty\ni}$: Impersonation attacks; ✓: indicates feature availability; ×: indicates feature unavailability or inapplicability.

6.5. Security and Functionality Features

Table 8 presents a comparison of the proposed protocol against the benchmark protocols: Hammad et al. [60], Wazid et al. [61], Yang et al. [62], Sutrala et al. [63], and Srinivas et al. [64]. The comparison evaluates thirteen security and functionality features: $\mathcal{F}_{\infty}$: Smart sensing device capture attack; $\mathcal{F}_{\in}$: Anonymity; $\mathcal{F}_{\ni}$: Untraceability; $\mathcal{F}_{\triangle}$: De-synchronization attack; $\mathcal{F}_{\bigtriangledown}$: Replay attack; $\mathcal{F}_{/}$: MitM attack; $\mathcal{F}_{\shortmid}$: Mutual authentication; $\mathcal{F}_{\forall}$: Independent session key establishment; $\mathcal{F}_{\exists}$: Perfect forward secrecy; $\mathcal{F}_{\infty\prime}$: No clock synchronization; $\mathcal{F}_{\infty\infty}$: ESL attack; $\mathcal{F}_{\infty\in}$: Smartcard theft and insider threats; and $\mathcal{F}_{\infty\ni}$: Impersonation attacks. In Table 8, a check mark (✓) denotes the presence of a feature, while a cross (×) indicates its absence or inapplicability. The comparison reveals that the proposed protocol is the only one to encompass all the essential and critical security and functionality features. Conversely, the benchmark protocols show shortcomings, either missing certain features or failing to counter specific security threats.

6.6. Critical Discussion

The proposed AKA protocol delivers a balanced solution for real-time IIoT environments by integrating PUFs, fuzzy extractors, and chaotic maps, ensuring strong resistance to attacks and meeting all targeted security and functionality objectives. It maintains moderate computational and communication overheads, making it suitable for resource-constrained devices, with superior runtime performance. Limitations include PUF reliability under environmental variations, computational overhead from fuzzy extractors, and reliance on simulations lacking real-world insights. Trade-offs involve latency, storage overhead, and implementation complexity. Future work targets real-world testbed implementation, IIoT communication stack integration, robust key management, recovery mechanisms, and energy efficiency evaluation under diverse scenarios.

7. Conclusions

To address the threats posed by various known attacks on wireless data communication between users, gateway nodes, and smart sensing devices in Industrial Internet of Things (IIoT) network scenarios, this paper proposes an anonymous and secure authentication and key agreement protocol for IIoT settings. The proposed protocol is based on

PUF, cryptographic hash, XOR, and Chaotic Map, providing strong security features while maintaining resource efficiency, making it more suitable for resource-constrained IIoT environments. Furthermore, a comprehensive comparison with existing protocols demonstrates that the proposed protocol significantly reduces computational and communication overhead while offering superior security features, representing a substantial advancement in the field. Future work includes the exploration of post-quantum cryptographic primitives to ensure long-term resistance against quantum-capable adversaries, and the experimental deployment of the proposed protocol in real-world IIoT testbeds to validate its performance under diverse and dynamic environmental conditions.

Author Contributions: Conceptualization, D.Z., X.A., S.T. and A.B.; Methodology, D.Z., X.A., S.T. and A.B.; Software, D.Z.; Validation, D.Z., X.A., S.T., M.W. and M.T.K.; Formal analysis, D.Z., X.A., S.T., A.B. and M.W.; Investigation, D.Z., X.A., S.T., A.B., M.W. and M.T.K.; Resources, X.A. and H.A.; Data curation, M.T.K.; Writing—original draft, D.Z., X.A. and A.B.; Writing—review & editing, S.T., A.B., H.A. and M.W.; Visualization, S.T., A.B., H.A., M.W. and M.T.K.; Supervision, S.T., A.B., H.A. and M.W.; Project administration, A.B. and H.A.; Funding acquisition, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number RGP.2/637/46.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available from the authors upon reasonable request.

Conflicts of Interest: The authors declared that no conflict of interest exist.

References

1. Kebande, V.R.; Awad, A.I. Industrial Internet of Things ecosystems security and digital forensics: Achievements, open challenges, and future directions. *ACM Comput. Surv.* **2024**, *56*, 131.
2. Awaisi, K.S.; Ye, Q.; Sampalli, S. A survey of industrial AIoT: Opportunities, challenges, and directions. *IEEE Access* **2024**, *12*, 96946–96996.
3. Nallakaruppan, M.K.; Dhanaraj, R.K.; Shukla, S.; Subbaraj, K.; Fuladi, S.; Selvarajan, S.; Alkhayyat, A.; Alturki, N. Reliable secured consumer IIoT framework with multi-layer attack interpretation and prevention. *IEEE Trans. Consum. Electron.* **2025**, *71*, 5036–5043.
4. Sah, D.K.; Vahabi, M.; Fotouhi, H. A comprehensive review on 5G IIoT test-beds. *IEEE Trans. Consum. Electron.* **2025**, *71*, 4139–4163.
5. Khan, W.Z.; Aalsalem, M.Y.; Khan, M.K. Communal acts of IoT consumers: A potential threat to security and privacy. *IEEE Trans. Consum. Electron.* **2019**, *65*, 64–72.
6. Sun, G.; Xu, Z.; Yu, H.; Chang, V. Dynamic network function provisioning to enable network in box for industrial applications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7155–7164.
7. Yu, F.; Wang, X.; Guo, R.; Ying, Z.; Cai, S.; Jin, J. Dynamical analysis, hardware implementation, and image encryption application of new 4D discrete hyperchaotic maps based on parallel and cascade memristors. *Integration* **2025**, *104*, 102475.
8. Sun, G.; Song, L.; Yu, H.; Chang, V.; Du, X.; Guizani, M. V2V routing in a VANET based on the autoregressive integrated moving average model. *IEEE Trans. Veh. Technol.* **2019**, *68*, 908–922.
9. Zhang, M.; Wei, E.; Berry, R.; Huang, J. Age-dependent differential privacy. *IEEE Trans. Inf. Theory* **2024**, *70*, 1300–1319.
10. Luo, T.; Zhou, Y.; He, Z.; Jiang, G.; Xu, H.; Qi, S.; Zhang, Y. StegMamba: Distortion-free immune-cover for multi-image steganography with state space model. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 4576–4591.
11. Han, F.; Yang, P.; Du, H.; Li, X.-Y. Accuth++: Accelerometer-based anti-spoofing voice authentication on wrist-worn wearables. *IEEE Trans. Mob. Comput.* **2024**, *23*, 5571–5588.
12. Hu, J.; Jiang, H.; Liu, D.; Xiao, Z.; Dustdar, S.; Liu, J. A wireless self-service system for library using commodity RFID devices. *IEEE Internet Things J.* **2024**, *11*, 4998–5010.
13. Xu, G.; Lei, L.; Mao, Y.; Li, Z.; Chen, X.; Zhang, K. CBRFL: A framework for committee-based Byzantine-resilient federated learning. *J. Netw. Comput. Appl.* **2025**, *238*, 104165.

14. Yanambaka, V.P.; Mohanty, S.P.; Kougianos, E.; Puthal, D. PMsec: Physical unclonable function-based robust and lightweight authentication in the Internet of medical things. *IEEE Trans. Consum. Electron.* **2019**, *65*, 388–397.
15. Ai, X.; Badshah, A.; Tu, S.; Waqas, M.; Ahmad, I. An improved ultra-lightweight anonymous authenticated key agreement protocol for wearable devices. *IEEE Trans. Mob. Comput.* **2025**, *24*, 4543–4557.
16. Zhou, Z.; Shojafar, M.; Alazab, M.; Abawajy, J.; Li, F. AFED-EF: An energy-efficient VM allocation algorithm for IoT applications in a cloud data center. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 658–669.
17. Chen, Y.; Liang, X.; Zhou, H.; Yang, X.; Wu, L.; Lv, G. GENDN: A geospatially enhanced NDN framework for location-related pub/sub services in NTN-enabled IoT. *IEEE Internet Things J.* **2025**, *12*, 8381–8393.
18. Xu, G.; Xu, S.; Fan, X.; Cao, Y.; Mao, Y.; Xie, Y.; Chen, X. RAT ring: Event-driven publish/subscribe communication protocol for IIoT by report and traceable ring signature. *IEEE Trans. Ind. Inform.* **2025**, *21*, 6670–6678.
19. Xu, Y.; Liu, Y.; Lei, M.; Gao, M.; Fang, Z.; Jiang, C. Joint pseudo-range and Doppler positioning method with LEO satellites' signals of opportunity. *Satell. Nav.* **2025**, *6*, 10.
20. Ding, F.; Liu, Z.; Wang, Y.; Liu, J.; Wei, C.; Nguyen, A.; Wang, N. Intelligent event-triggered lane keeping security control for autonomous vehicle under DoS attacks. *IEEE Trans. Fuzzy Syst.* **2025**, *33*, 3595–3608.
21. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* **2014**, *20*, 96–112.
22. Tai, W.-L.; Chang, Y.-F.; Li, W.-H. An IoT notion-based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks. *J. Inf. Secur. Appl.* **2017**, *34*, 133–141.
23. Chen, Y.; Yin, F.; Hu, S.; Sun, L.; Li, Y.; Xing, B.; Chen, L.; Guo, B. ECC-based authenticated key agreement protocol for industrial control system. *IEEE Internet Things J.* **2022**, *10*, 4688–4697.
24. Shuai, M.; Xiong, L.; Wang, C.; Yu, N. A secure authentication scheme with forward secrecy for industrial Internet of Things using Rabin cryptosystem. *Comput. Commun.* **2020**, *160*, 215–227.
25. Gong, B.; Wu, Y.; Badshah, A.; Waqas, M. Privacy-preserving and traceable certificateless anonymous mutual authentication scheme for IoT. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 7508–7520.
26. Zeng, D.; Badshah, A.; Tu, S.; Waqas, M.; Han, Z. A security-enhanced ultra-lightweight and anonymous user authentication protocol for telehealthcare information systems. *IEEE Trans. Mob. Comput.* **2025**, *24*, 4529–4542.
27. Hammad, M.; Badshah, A.; Almeer, M.A.; Waqas, M.; Song, H.H.; Chen, S.; Han, Z. Lightweight and robust key agreement for securing IIoT-driven flexible manufacturing systems. *IEEE Internet Things J.* **2025**, *12*, 17197–17209.
28. Zhai, Z.; Liu, J.; Liu, X.; Mao, Y.; Zhang, X.; Ma, J.; Jin, C. A lightweight authentication method for Industrial Internet of Things based on blockchain and Chebyshev chaotic maps. *Future Internet* **2025**, *17*, 338.
29. Aman, M.N.; Basheer, M.H.; Sikdar, B. A lightweight protocol for secure data provenance in the Internet of Things using wireless fingerprints. *IEEE Syst. J.* **2021**, *15*, 2948–2958.
30. Modarres, A.M.A.; Sarbishaei, G. An improved lightweight two-factor authentication protocol for IoT applications. *IEEE Trans. Ind. Inform.* **2023**, *19*, 6588–6598.
31. Rafique, F.; Obaidat, M.S.; Mahmood, K.; Ayub, M.F.; Ferzund, J.; Chaudhry, S.A. An efficient and provably secure certificateless protocol for industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8039–8046.
32. Yi, F.; Zhang, L.; Xu, L.; Yang, S.; Lu, Y.; Zhao, D. WSNEAP: An efficient authentication protocol for IIoT-oriented wireless sensor networks. *Sensors* **2022**, *22*, 7413.
33. Eldefrawy, M.H.; Ferrari, N.; Gidlund, M. Dynamic user authentication protocol for industrial IoT without timestamping. In Proceedings of the 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), Sundsvall, Sweden, 27–29 May 2019; pp. 1–7.
34. Harishma, B.; Patranabis, S.; Chatterjee, U.; Mukhopadhyay, D. POSTER: Authenticated key-exchange protocol for heterogeneous CPS. In Proceedings of the 2018 Asia Conference on Computer and Communications Security, Incheon, Republic of Korea, 4–8 June 2018; pp. 849–851.
35. Mutlaq, K.A.-A.; Nyangaresi, V.O.; Omar, M.A.; Abduljabbar, Z.A. Symmetric key based scheme for verification token generation in Internet of Things communication environment. In Proceedings of the EAI International Conference on Applied Cryptography in Computer and Communications, Virtual Event, 14–15 May 2022; pp. 46–64.
36. Chen, Y.; Martínez, J.-F.; Castillejo, P.; López, L. A privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: PriAuth. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 5290579.
37. Patel, C.; Doshi, N. Cryptanalysis of ECC-based key agreement scheme for generic IoT network model. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–7.

38. Chen, P.; Song, Y.; Xia, Y. Adaptively diagnosing system faults in microservice architecture: An autonomous predictive model construction framework. *Future Gener. Comput. Syst.* **2025**, *177*, 108256.
39. Jiang, H.; Ye, L.; Hu, J.; Chen, X.; Chen, S.; Zhang, W.; Yang, K. WarmGait: Thermal array-based gait recognition for privacy-preserving person re-ID. *IEEE Trans. Mob. Comput.* **2025**, 1–14. https://doi.org.10.1109/TMC.2025.3608447.
40. Luo, H.; Li, Q.; Cheng, H.; Li, W.; Sun, W.; Zhao, W.; Liu, Z. A2Tformer: Addressing temporal bias and non-stationarity in transformer-based IoT time series classification. *IEEE Internet Things J.* **2025**, *12*, 42198–42213.
41. Zhang, K.; Wang, H.; Chen, M.; Chen, X.; Liu, L.; Geng, Q.; Zhou, Y. Leveraging machine learning to proactively identify phishing campaigns before they strike. *J. Big Data* **2025**, *12*, 124.
42. Xu, G.; Wang, L.; Chen, S.; Zhu, L.; Guizani, M.; Shi, L. MPAEE: A multi-path adaptive energy-efficient routing scheme for low earth orbit-based industrial Internet of Things. *IEEE Internet Things J.* **2025**, *12*, 34793–34805.
43. Sun, Q.; Jian, X.; Han, C.; Li, Y. An improved opportunistic localization algorithm using LEO signals based on PSODC. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 8512110.
44. Liu, X.; Zhao, L.; Jin, J. A noise-tolerant fuzzy-type zeroing neural network for robust synchronization of chaotic systems. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8218.
45. Shen, X.; Li, L.; Ma, Y.; Xu, S.; Liu, J.; Yang, Z.; Shi, Y. VLCIM: A vision-language cyclic interaction model for industrial defect detection. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2538713 .
46. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
47. Guo, X.; Zhang, J.; Meng, X.; Li, Z.; Wen, X.; Girard, P.; Liang, B.; Yan, A. HALTRAV: Design of a high-performance and area-efficient latch with triple-node-upset recovery and algorithm-based verifications. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *44*, 2367–2377.
48. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 337–351.
49. Yang, J.; Liu, Y.; Wang, W.; Wu, H.; Chen, Z.; Ma, X. PATNAS: A path-based training-free neural architecture search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *47*, 1484–1500.
50. Zhao, Z.; Li, X.; Luan, B.; Jiang, W.; Gao, W.; Neelakandan, S. Secure Internet of Things using a novel Brooks–Iyengar quantum Byzantine agreement-centered blockchain networking model in smart healthcare. *Inf. Sci.* **2023**, *629*, 440–455.
51. Herder, C.; Yu, M.-D.; Koushanfar, F.; Devadas, S. Physical unclonable functions and applications: A tutorial. *Proc. IEEE* **2014**, *102*, 1126–1141.
52. Zhou, Z.; Abawajy, J.; Chowdhury, M.; Hu, Z.; Li, K.; Cheng, H.; Alelaiwi, A.A.; Li, F. Minimizing SLA violation and power consumption in cloud data centers using adaptive energy-aware algorithms. *Future Gener. Comput. Syst.* **2018**, *86*, 836–850.
53. Delvaux, J.; Gu, D.; Verbauwhede, I.; Hiller, M.; Yu, M.-D. Efficient fuzzy extraction of PUF-induced secrets: Theory and applications. In *Cryptographic Hardware and Embedded Systems—CHES 2016*; Gierlichs, B., Poschmann, A.Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 412–431.
54. Lee, T.-F. Efficient three-party authenticated key agreements based on Chebyshev chaotic map-based Diffie–Hellman assumption. *Nonlinear Dyn.* **2015**, *81*, 2071–2078.
55. Wu, Z.Y.; Ismail, M.; Zhang, J.; Zhang, J. Tidal-like concept drift in RIS-covered buildings: When programmable wireless environments meet human behaviors. *IEEE Wireless Commun.* **2025**, 1–8. https://doi.org/10.1109/MWC.2025.3600792.
56. Zhang, K.; Zheng, B.; Xue, J.; Zhou, Y. Explainable and trust-aware AI-driven network slicing framework for 6G IoT using deep learning. *IEEE Internet Things J.* **2025**. https://doi.org/10.1109/JIOT.2025.3619970.
57. Bergamo, P.; D'Arco, P.; De Santis, A.; Kocarev, L. Security of public-key cryptosystems based on Chebyshev polynomials. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2005**, *52*, 1382–1393.
58. Yoshioka, D. Security of public-key cryptosystems based on Chebyshev polynomials over $\mathbb{Z}/p^k\mathbb{Z}$. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2204–2208.
59. Xiao, L.; Zhao, L.; Jin, J. Preset-time convergence fuzzy zeroing neural network for chaotic system synchronization: FPGA validation and secure communication applications. *Sensors* **2025**, *25*, 5394.
60. Hammad, M.; Badshah, A.; Abbas, G.; Alasmary, H.; Waqas, M.; Khan, W.A. A provable secure and efficient authentication framework for smart manufacturing industry. *IEEE Access* **2023**, *11*, 67626–67639.
61. Wazid, M.; Das, A.K.; Kumar, N.; Alazab, M. Designing authenticated key management scheme in 6G-enabled network in a box deployed for industrial applications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7174–7184.
62. Yang, Z.; He, J.; Tian, Y.; Zhou, J. Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6584–6596.

63. Sutrala, A.K.; Obaidat, M.S.; Saha, S.; Das, A.K.; Alazab, M.; Park, Y. Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2316–2330.
64. Srinivas, J.; Das, A.K.; Wazid, M.; Kumar, N. Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1133–1146.