Lightweight and Robust Key Agreement for Securing IIoT-Driven Flexible Manufacturing Systems

Muhammad Hammad, Akhtar Badshah, Mohammed A. Almeer, Muhammad Waqas, *Senior Member, IEEE*, Houbing Song, *Fellow, IEEE*, Sheng Chen, *Life Fellow, IEEE*, and Zhu Han, *Fellow, IEEE*

***Abstract*—The ever-evolving Internet of Things (IoT) has ushered in a new era of intelligent manufacturing across multiple industries. However, the security and privacy of real-time data transmitted over the public channel of the industrial IoT (IIoT) remain formidable challenges. Existing lightweight protocols often omit one or more critical security features, such as anonymity and untraceability, and are susceptible to threats like desynchronization attacks. Additionally, they struggle to achieve an optimal balance between robust security and performance efficiency. To bridge these gaps, we introduce a new lightweight key agreement security scheme that guarantees secure access to the IIoT-enabled flexible manufacturing system (FMS). The strength of our scheme lies in its utilization of the authenticated encryption with associative data (AEAD) primitive, AEGIS, along with hash functions and physical unclonable functions, which secure the IIoT ecosystem. Additionally, our scheme offers flexibility in the form of the addition of new machines, password updates, and revocation in cases of theft or loss. A comprehensive security analysis demonstrates the efficacy of the proposed scheme in thwarting various attacks. The formal analysis, based on the Real-Or-Random (RoR) model, ensures session key indistinguishability, while the informal analysis highlights its resilience against known attacks. The comparative assessment demonstrate that the proposed scheme consistently outperforms the benchmark schemes across multiple dimensions, including security and functionality features, computational and communication overheads, and runtime efficiency. Specifically, the proposed scheme achieves peak performance enhancements of 77.55%, 44.73%, and 69.6% in computational overhead, runtime overhead, and communication overhead, respectively, underscoring its substantial performance advantages.**

***Index Terms*—Industrial Internet of Things, flexible manufacturing system, physical unclonable functions, user authentication, security.**

This work is partially supported by NSF ECCS-2302469, Toyota, Amazon and Japan Science and Technology Agency (JST), Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) JPMJAP2326.

M. Hammad is with the Faculty of Mechanical Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: muhammadhammad65@gmail.com).

A. Badshah is with the Department of Software Engineering, University of Malakand, Dir Lower 18800, Pakistan (e-mail: akhtarbadshah@uom.edu.pk).

M. A. Almeer is with the Computer Engineering Department, College of Information Technology, University of Bahrain, 32038, Bahrain (e-mail: malmeer@uob.edu.bh).

M. Waqas is with the Centre for Sustainable Cybersecurity, Faculty of Engineering and Science, University of Greenwich, United Kingdom, and also with the School of Engineering, Edith Cowan University, Perth, 6027 WA, Australia (e-mail: engr.waqas2079@gmail.com).

H. Song is with the Department of Information Systems, University of Maryland, Baltimore County (UMBC), Baltimore, MD, USA (e-mail: h.song@ieee.org; songh@umbc.edu).

S. Chen is with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sqc@ecs.soton.ac.uk).

Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701 (e-mail: hanzhu22@gmail.com).

Corresponding Author: Akhtar Badshah (e-mail: akhtarbadshah@uom.edu.pk).

I. Introduction

THE success of industries is heavily dependent on the technology, and Industry 4.0 represents a culmination of multiple advanced technologies aimed at meeting the demands of intelligent automation at a higher level. In particular, manufacturing industries are shifting towards the industry 4.0 approach to reap the full benefits of smart manufacturing [1]. Smart manufacturing, as a vision of Industry 4.0, integrates the physical and digital processes of cyber-physical systems. The introduction of the Internet of Things (IoT) is essential to smart manufacturing. IoT is the interconnection of objects (physical/virtual devices) for sharing information through Internet facilities. A physical object may comprise a cell, phone, machine, sensor, or camera, and the virtual object may consist of an agenda, electronic ticket, wallet or book [2]. There is a need to make the objects smart in IoT to minimize human involvement. Flexible manufacturing systems (FMS) are converted into smart manufacturing systems through the use of IoT [3]. IoT-enabled manufacturing is particularly beneficial to minimize the labor force and enhance productivity. One of the most impressive benefits that IoT-enabled FMS offer is the real-time error capture and automated rework [4]. Alongside the numerous benefits, the IoT-enabled manufacturing industries are facing severe challenges related to security, to implement attack-free smart manufacturing. Traditional security mechanisms are not applicable due to more complex and resource-intensive implementation, which is especially challenging in low-resourced computational IoT devices like the ones present in industrial settings. The Internet engineering task force has delegated the responsibility of designing security measures for resource-constrained IoT systems to the system designers, who are expected to tailor their security schemes to their specific circumstances. This underscores the pressing demand for security schemes that are lightweight yet provide solid protection to IoT devices without reducing their feature richness or performance [5].

Smart manufacturing industries pose a significant challenge in ensuring real-time analysis of systems equipped with smart devices. The security of smart machines is always at risk when accessed by unauthorized users. IoT enabled systems are susceptible to numerous attacks because they operate with resource constrained devices and also lack of robust security measures. As a result of cyber-attacks, whole manufacturing ecosystem affects. These attacks not only undermine the integrity and confidentiality of data exchanged within a system but also resulting in downtime, costs, monetary losses, and damage to the wider supply chain [6], [7]. To address this issue, it is essential to design an environment with real-time data transmission to ensure the security of manufacturing industries, enhancing the accuracy and efficiency of machines and enabling remote monitoring. In particular, transmitting data over the public channel of the Internet makes it vulnerable to attack. Consequently, it is crucial to secure the transmission of confidential data between authorized parties and smart sensing machines to prevent illegal access. This can be accomplished by establishing a confidential key session using a trusted entity such as the master controller node (MCN) in IIoT [8]. The main problem to be dealt with in this study is the vulnerability of IoT-enabled FMS to a variety of cyber-attacks. That's why, an ultralightweight security scheme for IoT-enabled FMS is highly necessary, which would not only ensure securing the system from various cyber-attacks but also sustain its operational efficiency required by real-time manufacturing processes.

Currently, numerous authentication and key agreement schemes have been proposed to meet the security requirements of various IoT scenarios. Turkanović *et al.* [9] presented an authentication scheme designed for a WSN sitting. However, Farash *et al.* [10] discovered its security flaws, leading them to develop an alternative user authentication protocol specifically tailored for IoT deployment. Subsequently, Amin *et al.* [11] analyzed the scheme of [10] and proposed an enhanced authentication scheme to address its security vulnerabilities. Unfortunately, Jiang *et al.* [12] demonstrated that the scheme proposed in [11] also possesses various security loopholes, and they then proposed another improved lightweight authentication scheme for WSN to rectify these vulnerabilities. Rafique *et al.* [13] rectifies a significant issue in the realm of IIoT, which revolves around the secure transmission of data. Their research put forth a multifactor authentication key agreement scheme designed to strike a balance between robust security and the limitations imposed by resource constraints. The proposed scheme employed bitwise XOR, cryptographic hash, and symmetric cryptography to establish a robust system specifically designed for environments with limited resources, ensuring a high level of security. It facilitated remote access to sensing devices while maintaining a high level of security. However, the study [14] found that the scheme of [13] is vulnerable to attacks involving the loss of smart cards/devices. Eldefrawy *et al.* [15] introduced a user authentication method for IIoT systems that emphasizes computational and communication efficiency. Although the proposed scheme demonstrated efficiency, it falls short in terms of establishing mutual authentication between users and smart devices/sensor nodes present in the system. Harishma *et al.* [16] presented a method to secure the transmission of data in cyber-physical systems with heterogeneous components. However, their proposed approach was found to be susceptible to the ephemeral secret leakage (ESL) attack when operating under the Canetti and Krawczyk (CK) adversary model [17]. Moreover, the scheme lacks the capability to incorporate new IoT smart devices dynamically, which may hinder its practical applications. Chen *et al.* [18] devised a key agreement and user authentication system for IoT settings. Although the scheme exhibited efficiency in computational and communication costs, it falls short in terms of security against insider attacks, node-capturing attacks, and gateway node-bypassing attacks as well as lacking the property of untraceability. Masud *et al.* [19] proposed an anonymous authentication protocol for telemedicine systems based solely on hash functions, claiming that their scheme can resist various known attacks. However, Wang *et al.* [20] evaluated the protocol and uncovered significant design flaws, exposing it to risks such as session key leakage, offline password guessing, and traceability issues. Praveen and Pabitha [21] advanced a secure user authentication scheme based on bioacoustics, utilizing the Chinese Remainder Theorem to generate group keys and enhancing protocol security through the integration of fuzzy embedding. However, their scheme is vulnerable to replay attacks and impersonation attacks. Chen *et al.* [22] proposed an authentication protocol for wireless body area networks, validating its security through formal and informal analyses. Nonetheless, this scheme is susceptible to denial-of-service attacks on sensor nodes and fails to achieve system key verification. Pu *et al.* [23] proposed an authentication protocol named LiteAuth; however, its excessive communication overhead makes it unsuitable for resource-constrained IIoT scenarios. Additionally, Hu *et al.* [24] proposed an anonymous authentication and key agreement scheme for advanced metering infrastructure. Although their scheme achieves low performance overhead, it fails to provide untraceability.

In this paper, we present an innovative user ultralightweight authentication scheme designed specifically for FMS environments. Our contributions are summarized as follows:

- We introduce a new user authentication and key agreement scheme for IIoT-based FMS environment. The scheme employs SHA-256 hash function, AEGIS primitive, and PUF to ensure robust security with minimal computational overhead. It guarantees user authenticity, establishes a session key for secure communication between user and smart sensing device, and enhances physical security by preventing unauthorized tampering. To strengthen the security and integrity of the system, we integrate a revocation phase and a password update phase.
- We employ a comprehensive evaluation approach to assess the effectiveness of our scheme in mitigating common types of attacks in IIoT environments. This evaluation encompasses both formal security analysis utilizing the Real-or-Random (ROR) model and informal security verification. The results of our analysis demonstrate that our scheme successfully withstands potential security attacks, thereby highlighting its robust security attributes.
- We conduct an extensive comparative evaluation of the proposed scheme against benchmark schemes to assess its performance across multiple dimensions, including security and functionality features, computational and communication overheads, and runtime efficiency. The results of the comparison demonstrate that our scheme outperforms existing schemes in these aspects, thereby highlighting its overall superiority.

The rest of the paper is structured as follows. Section II presents an introduction to our network and threat models, along with the essential preliminaries. In Section III, we provide a detailed explanation of our proposed scheme. The security assessment of the proposed scheme is discussed in Section IV. Furthermore, in Section V, a comparison between the proposed scheme and other existing schemes is presented. Finally, Section VI concludes the paper.

II. Network, Threat Model and Preliminaries

In this section, we introduce our network and threat models. Moreover, we provide a concise introduction to the relevant foundational concepts that underpin our proposed scheme.

A. Network Model

The network model is illustrated in Fig. 1, which consists of four primary entities:

1) **Users:** To access a smart sensing device, user U_i initiates a request through their device UD_i to the Master Controller Node (MCN) MCN_j. The request is forwarded to the appropriate smart sensing devices for further processing.
2) **Smart Sensing Devices:** These devices are deployed to collect data and monitor various processes, such as manufacturing or environmental conditions. Users can access the real-time data from these devices to make informed decisions and perform necessary actions.
3) **Master Controller Node (MCN):** The MCN is responsible for securely authenticating registered users, storing credentials for both users and smart sensing devices, and facilitating the establishment of secure communication channels (sessions) between users and their designated smart devices. Each MCN is associated with multiple smart sensing devices.
4) **Trusted Registration Authority (TA):** The TA handles the registration process for all network entities (MCNs, smart sensing devices, and users). It securely stores and manages the credentials of all entities and ensures their authenticity during registration.

In this model, smart sensing devices (SD_k) are registered with an MCN (MCN_j), which securely stores their credentials. To access a sensing device, a user (U_i) must first register with MCN_j, which involves storing the user's authentication credentials. During the login and authentication key agreement phase, the user sends a request to MCN_j. Upon verifying the user's authenticity, MCN_j forwards the request to the relevant sensing devices. These devices authenticate the request, generate a shared session key, and send a response back to the user. After authenticating the response, the user generates the same session key. With this shared session key, the user can securely access the data collected by the sensing devices and regulate the monitored processes, ensuring secure and seamless communication.

B. Threat Model

We employs the widely recognized Dolev-Yao (DY) model [25] to secure the proposed system. Within the DY model, adversary $\mathscr{A}$ possesses the ability to read, delete, modify, and send fake messages during communication over an unsecured public channel. Additionally, due to the vulnerabilities inherent in IIoT devices, $\mathscr{A}$ can exploit opportunities to capture IoT sensing machines. Through power analysis attacks, $\mathscr{A}$ can extract secret credentials stored in the memory of these compromised machines. Similarly, if a legitimate user's device or smart card is lost or stolen, $\mathscr{A}$ can gain access to the secret credentials stored within them. Armed with such sensitive information, $\mathscr{A}$ gains the capability to launch a variety of attacks, including replay attacks, privileged-insider attacks, impersonation attacks, and man-in-the-middle attacks. Additionally, the CK-adversary model [26] is considered the standard for authenticated security protocols. The CK model encompasses all the activities discussed within the DY model and includes an additional feature of revealing confidential credentials during sessions, such as session keys and session states. Consequently, the authentication scheme implemented in the proposed system must possess the potential to ensure security by effectively mitigating the effects of attacks, even in the scenarios where confidential credentials are exposed to $\mathscr{A}$ during communication. It should be noted that the MCNs in the proposed IIoT system are operated in a locking mode to safeguard against physical attacks instigated by $\mathscr{A}$. Consequently, the MCNs are regarded as secure within the system.

C. Preliminaries

This subsection provides a brief overview of foundational concepts that underpin our proposed scheme.

1) Physical unclonable function (PUF):

A PUF capitalizes on the distinctive physical attributes of a device to generate an exclusive response, employed for encryption and authentication purposes. Specifically, when a PUF receives multiple inputs (i.e., challenges), even minimal physical differences between devices-such as slight variations in transistors, circuit delays, or manufacturing imperfections-cause the PUF to generate different outputs (responses). Consequently, each device produces a unique set of Challenge-Response Pairs (CRPs). Leveraging these characteristics, a PUF can be defined as the following abstract function:

$$R_i = PUF(C_i) \quad (C_i \in C,\ R_i \in R)$$

In the symbolic representation of a PUF, the challenge set C comprises unique challenges from multiple entities, denoted as C_i where $i = 1, 2, \ldots, n$. Correspondingly, the response set R contains a distinct response R_i for each challenge C_i. The PUF mapping, denoted as $PUF(\cdot)$, precisely maps each challenge directly to its specific response. PUFs offer a cryptographic mechanism that ensures both security and personalized key generation, effectively distinguishing between devices. However, the accuracy of PUF responses may be impacted by environmental noise, introducing a potential risk of compromising sensitive information during critical operations. Recent studies [27] have explored various noise-resistant and stable PUF designs capable of achieving an almost 0% bit error rate, even under challenging conditions such as voltage fluctuations and extreme temperature variations. Thus, in this paper, we assume that smart sensing devices, MCNs, and user devices

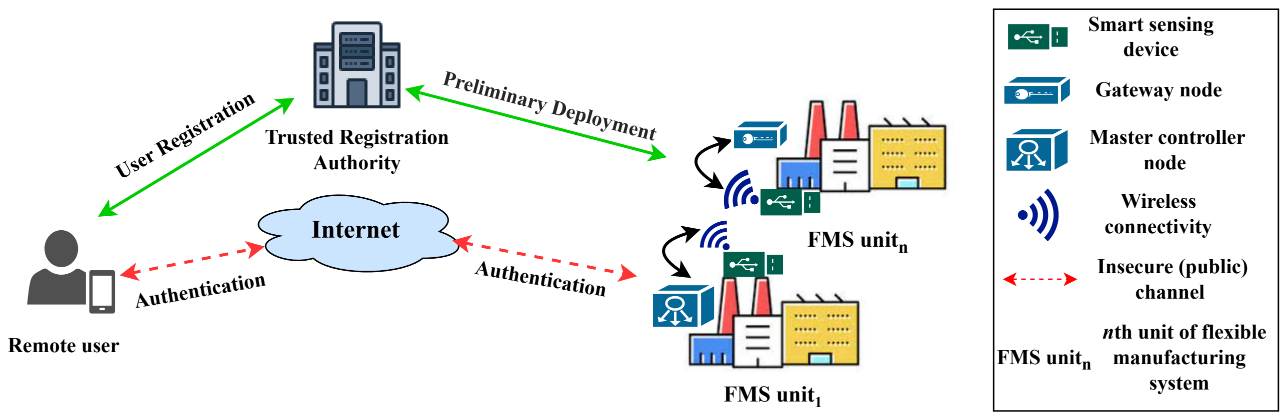


Fig. 1: Network model of flexible manufacturing monitoring system.

are equipped with ideal and noise-resistant PUFs.

2) AEGIS: AEGIS [28] is a cryptographic technique belonging to the category of authenticated encryption with associated data (AEAD). Its design is tailored to suit resource-limited devices as well as high-performance computing applications. Its unique features include its lightweight, robustness, inverse-free and online nature. The encryption process of AEGIS can be symbolically expressed as follows:

$$\{CT_i, MAC_i\} = E_K(IV, AD, PT_i),$$

where CT_i stands for the resulting ciphertext, MAC_i is the authentication tag, IV represents the initialization vector, AD refers to the associated data, K denotes the shared key, and PT_i represents the plaintext to be encrypted. Additionally, the decryption process of AEGIS is described as follows:

$$\{PT_i, \perp\} = D_K(IV, AD, CT_i, MAC_i)$$

Specifically, in the decryption process, AEGIS takes as input the (CT_i, MAC_i) pair generated during encryption, along with (IV, AD, K), and computes a new authentication tag MAC' based on the received (IV, AD, K, CT_i) through the decryption function. It then verifies whether $MAC' = MAC$. If the verification of MAC_i fails, an error $\perp$ is triggered; Otherwise, the plaintext PT_i is retrieved. These features make AEGIS an ideal primitive for our scheme, as it simplifies the authentication scheme, reduces complexity and enhances the overall security of the system.

III. The Proposed Scheme

Table I lists the symbols employed in the design of the proposed scheme. The scheme consists of six phases: registration of MCN and smart sensing device, user registration, authentication and key agreement, password updating, revocation, and deployment of dynamic smart sensing devices.

A. Preliminary Deployment Phase

In this phase, TA plays a crucial role in enrolling MCNs and smart sensing devices before they are deployed.

1) MCN registration: The following operations are performed by TA to register a MCN MCN_j.

Step 1: A distinct challenge parameter C_{MCN_j} is produced by TA and transmitted to MCN_j through a secure channel.

Step 2: MCN_j computes the response parameter, and forwards it to TA via a secure channel. MCN_j computes its unique response parameter as follows: $R_{MCN_j} = PUF(C_{MCN_j})$. Subsequently, MCN_j forwards R_{MCN_j}.

Step 3: TA picks an identity SID_{MCN_j} and a secret parameter SP_{MCN_j}. It then calculates a value X_{MCN_j} by concatenating SID_{MCN_j} and SP_{MCN_j} and XOR-ing the result with the hash of R_{MCN_j} as: $X_{MCN_j} = (SID_{MCN_j} \| SP_{MCN_j}) \oplus h(R_{MCN_j})$. Finally, TA stores $\{X_{MCN_j}, C_{MCN_j}\}$ securely in the memory of MCN_j and deletes $\{X_{MCN_j}, C_{MCN_j}, R_{MCN_j}, SP_{MCN_j}\}$ from its own database to prevent attacks, such as privileged-insider and stolen verifier attacks.

2) Smart sensing device registration: The following steps are carried out by TA to register smart sensing devices SD_k, where $k = 1, 2, \cdots, n$.

Step 1: The TA initiates the process by generating a distinct challenge parameter C_{SD_k}. This parameter is securely transmitted to SD_k.

Step 2: Upon receiving the C_{SD_k} from TA through a secure channel, SD_k employs $PUF(\cdot)$ to calculate the response parameter R_{SD_k}. Subsequently, R_{SD_k} is securely transmitted back to TA.

Step 3: TA selects an identity SID_{SD_k} and a secret parameter SP_{SD_k} for SD_k, and calculates a value X_{SD_k} as $X_{SD_k} = (SID_{SD_k} \| SP_{SD_k}) \oplus h(R_{SD_k})$. $\{X_{SD_k}, C_{SD_k}, PUF(\cdot)\}$ is securely stored in the memory of SD_k.

In addition, TA sends the parameters $\{SID_{SD_k},\ SP_{SD_k}\}$ of SD_k to the associated MCN MCN_j. Upon receiving these parameters, MCN_j uses them to compute $\{CT_{SD_k}, MAC_{SD_k}\} = E_{K_{MCN_j}}(IV, AD, PT)$, where $IV = SID_{MCN_j}$, $K_{MCN_j} = SP_{MCN_j}$, $AD = SID_{MCN_j}$, and $PT = SP_{SD_k}$. Then, MCN_j stores $\{SID_{SD_k}, CT_{SD_k}, MAC_{SD_k}\}$ in its own memory. Finally, TA removes the parameters $\{X_{SD_k}, C_{SD_k}, R_{SD_k}, SP_{SD_k}\}$ from its database to prevent potential attacks, such as stolen verifier and privileged-insider attacks.

B. User Registration Phase

In order to establish a secure communication between user U_i and the deployed SD_k in the flexible manufacturing environment, U_i must register with TA. During the registration

TABLE I: Notations and descriptions

Notation	Description
TA	trusted registration authority
$\mathcal{A}$	Adversary
U_i	ith user
UD_i, MCN_j, SD_k	ith user device, jth MCN, kth smart sensing device
ID_i, PW_i	Identity and password of user U_i
SID_e, SP_e	Pseudonymous identity and secret parameter of the communication entity e
SID^c, SID^p	Current and previous pseudonymous identities
r_i	ith random number utilized in AKA phase
X_j	The jth intermediate result computed during AKA phase
N^a, N^b	Two parts obtained by equally dividing notation N
IV_i, AD_i, PT_i	The ith initialization vector, associated data and plaintext used in the AKA phase
CT_i, MAC_i	The ith ciphertext and its corresponding authentication Tag in AKA phase
T_i	ith timestamps utilized in AKA phase
T_i^*	ith timestamps upon message receipt
(C_e, R_e)	Challenge-response pair of the communication entity e
$h(\cdot)$	collision-resistant cryptographic hash function
$PUF(\cdot)$	physical unclonable function
$E_k(\cdot)/D_k(\cdot)$	AEGIS encryption/decryption using shared secret key k
$\|, \oplus$	Concatenation and bitwise XOR

process, TA assigns secret parameters to U_i for authentication purpose and a list of authorized SDs that U_i can access in real time. During the AKA procedure at MCN_j, U_i is validated. The user registration is conducted offline via a secure channel to preserve data confidentiality and integrity. The user registration process is detailed below.

Step 1: First, U_i selects an identity ID_i and a chosen password PW_i. Next, U_i sends a registration request message $< ID_i >$ to TA via a secure channel.

Step 2: After receiving the registration request, TA selects a secret parameter SP_{UD_i} and generates a list of authorized SDs that U_i can access in real-time, such as SD_k, along with a unique identifier SID_{UD_i}. TA then forwards $\{SID_{UD_i}, SP_{UD_i}, SID_{SD_k}\}$ to both U_i and the associated MCN, such as MCN_j.

Step 3: After receiving $\{SID_{UD_i}, SP_{UD_i}, SID_{SD_k}\}$ from TA, U_i selects two random numbers, rn_1 and rn_2, and computes several values as: $X_{UD_i} = (rn_1 \| rn_2) \oplus h(ID_i \| PW_i)$, $R_i = PUF(PW_i)$, $X_i = h(ID_i \| PW_i \| R_i)$, $K_{UD_i} = X_i^a \oplus X_i^b$, and $\{CT_{UD_i}, MAC_{UD_i}\} = E_{K_{UD_i}}(IV_i, AD_i, PT_i)$, where $IV_i = r_1$, $AD_i = r_2$ and $PT_i = \{SP_{UD_i} \| SID_{SD_k}\}$. U_i then stores $\{SID_{UD_i}, CT_{UD_i}, MAC_{UD_i}, X_{UD_i}\}$ in its own memory.

Step 4: After receiving $\{SID_{UD_i}, SP_{UD_i}, SID_{SD_k}\}$ from TA, MCN_j computes $\{CT_i, MAC_i\} = E_{K_{MCN_j}}(IV_i, AD_i, PT_i)$, where $IV_i = SID_{MCN_j}$, $AD_i = SID_{MCN_j}$, $K_{MCN_j} = SP_{MCN_j}$ and $PT_i = SP_{UD_i} \| SID_{SD_k}$. MCN_j then stores $\{SID^c_{UD_i}, SID^p_{UD_i}, CT_i, MAC_i\}$ in its own memory. Initially, both $SID^c_{UD_i}$ and $SID^p_{UD_i}$ are set to SID_{UD_i}. However, during the execution of the AKA phase, both $SID^c_{UD_i}$ and $SID^p_{UD_i}$ are updated.

C. Login Phase

To access a desired smart sensing device SD_k in the flexible manufacturing environment, a registered user U_i undertakes the following actions to log in.

Step 1: U_i inputs its identity ID_i and password PW_i^l at the registered user device UD_i. UD_i then computes $R_i^l = PUF(PW_i^l)$, $(rn_1 \| rn_2) = X_{UD_i} \oplus h(ID_i \| PW_i^l)$ and $X_i = h(ID_i \| PW_i^l \| R_i^l)$.

Step 2: UD_i extracts the pre-stored (CT_{UD_i}, MAC_{UD_i}) from memory and computes $K_{UD_i} = X_i^a \oplus X_i^b$. It then calculates $\{PT_{UD_i}, \perp\} = D_{K_{UD_i}}(IV_i, AD_i, CT_{UD_i}, MAC_{UD_i})$, where $IV_i = rn_1$ and $AD_i = rn_2$. If the verification of MAC_{UD_i} fails, it indicates that the attempting U_i is an unauthorized entity who failed to provide the correct ID_i and PW_i^l pair, resulting in the inability to decrypt (CT_{UD_i}, MAC_{UD_i}). In such a case, UD_i aborts the login attempt and terminates the session. Otherwise, the login attempt is deemed successful, and the legitimacy of U_i's identity is confirmed. UD_i then retrieves $\{SID_{SD_k}, SP_{UD_i}\}$ from the plaintext PT_{UD_i}.

D. Authenticated Key Agreement Phase

The AKA phase consists of the following steps.

AKA 1: After U_i successfully completes the local login authentication by providing the correct credentials (ID_i, PW_i^l) and passing the verification process detailed in Step 2 of the *Login Phase*, UD_i selects the current timestamp T_1 of size 32 bits and generates two random numbers, r_1 and r_2, each of size 128 bits. Then UD_i calculates IV_1 as the result of XOR operation between SID_{UD_i}, r_2, and T_1, K_a as SP_{UD_i}, AD_1 as SID_{UD_i}, and PT_1 as the concatenation of SID_{SD_k} and r_1. Here, IV_1, K_a, AD_1, PT_1 and SID_{SD_k} are the initialization vector (IV), key, associative data (AD), plaintext and identity of the desired smart sensing device SD_k, respectively. Then, UD_i uses AEGIS to compute ciphertext CT_1 and message authentication code MAC_1 as $\{CT_1, MAC_1\} = E_{K_a}(IV_1, AD_1, PT_1)$. Finally, UD_i constructs message MSG_1 and sends it to MCN_j through a public channel.

AKA 2: MCN_j checks the validity of received timestamp T_1 by verifying if $|T_1 - T_1^*| \overset{?}{\leq} \Delta T$, where T_1^* is the reception time of MSG_1. If this condition is not met, MCN_j halts any further processing. Otherwise, MCN_j extracts the received identity SID_{UD_i} from MSG_1 and verifies the condition ($SID_{UD_i} = SID^c_{UD_i}$ or $SID_{UD_i} = SID^p_{UD_i}$). If the condition is true, MCN_j retrieves the corresponding ciphertext and message authentication code pair $\{CT_i, MAC_i\}$. MCN_j further extracts it own parameters C_{MCN_j} and X_{MCN_j} and then computes $R_{MCN_j} = PUF(C_{MCN_j})$, $(SID_{MCN_j} \| SP_{MCN_j}) = X_{MCN_j} \oplus h(R_{MCN_j})$, $K_i = SP_{MCN_j}$, $AD_2 = SID_{MCN_j}$ and $IV_2 = SID_{MCN_j}$, where K_i, AD_2, and IV_2 are key, AD and IV, respectively. Moreover, by employing AEGIS, MCN_j computes $\{PT_i, \perp\} = D_{K_i}(IV_2, AD_2, CT_i, MAC_i)$. If the verification of MAC_i fails, MCN_j aborts the AKA procedure. Otherwise, MCN_j retrieves $\{SID_{SD_k}, SP_{UD_i}\}$ from plaintext PT_i.

User device UD_i	**Master controller node** MCN_j **/ Gateway** GW_j	**Smart sensing device** SD_k
$\{SID_{UD_i},\ CT_{UD_i},\ MAC_{UD_i},\ X_{UD_i},\ PUF(\cdot)\}$	$\{(SID^c_{UD_i}, SID^p_{UD_i}), CT_i,\ MAC_i\}$, $\{SID_{SD_k}, CT_{SD_k}, MAC_{SD_k}\}$	$\{X_{SD_k}, C_{SD_k}, PUF(\cdot)\}$
LG-1: Enter: $ID_i,\ PW^l_i$; **LG-2:** Compute: $R^l_i = PUF(PW^l_i)$, $(rn_1 \Vert rn_2) = X_{UD_i} \oplus h(ID_i \Vert PW^l_i)$, $X_i = h(ID_i \Vert PW^l_i \Vert R^l_i)$, $K_{UD_i} = X^a_i \oplus X^b_i$, $IV_i = rn_1,\ AD_i = rn_2$, $\{PT_{UD_i}, \perp\} = D_{K_{UD_i}}(IV_i, AD_i, CT_{UD_i}, MAC_{UD_i})$: Error, if verification of MAC_{UD_i} fails, Else, retrieve $PT_{UD_i} = \{SID_{SD_k},\ SP_{UD_i}\}$. **AKA-1:** Pick: r_1, r_2, T_1, Compute: $IV_1 = SID_{UD_i} \oplus r_2 \oplus T_1$, $K_a = SP_{UD_i}, AD_1 = SID_{UD_i}$, $PT_1 = SID_{SD_k} \Vert r_1$, $\{CT_1,\ MAC_1\} = E_{K_a}(IV_1, AD_1, PT_1)$; $\xrightarrow{MSG_1=\{SID_{UD_i},\ CT_1,\ MAC_1,\ r_2,\ T_1\}}$ (Via public channel) **AKA-8:** Check if $\lvert T_3 - T^*_3 \rvert \overset{?}{\leq} \Delta T$, if not, abort Compute: $X_4 = h(SID_{UD_i} \Vert r_1)$, Split: X_4 into X^a_4 and X^b_4, Compute: $IV_7 = X^b_4 \oplus T_3,\ K_f = X^a_4$, $AD_7 = SID_{SD_k}$, $\{PT_7, \perp\} = D_{K_f}(IV_7, AD_7, CT_3,\ MAC_3)$: Error, if verification of MAC_3 fails, Else, retrieve $PT_7 = r_4 \oplus r_5 \oplus SP_{SD_k} \oplus SID_{SD_k}$, Compute: $SK_{U_iD_k} = h(X_4 \Vert PT_7 \Vert T_3)$, Update: $SID_{UD_i} = X^a_4 \oplus X^b_4$.	**AKA-2:** Confirm if $\lvert T_1 - T^*_1 \rvert \overset{?}{\leq} \Delta T$; otherwise, terminate. Retrieve: SID_{UD_i} from MSG_1, Search: SID_{UD_i}, Check if ($SID_{UD_i} = SID^c_{UD_i}$ or $SID_{UD_i} = SID^p_{UD_i}$), If so retrieves $\{CT_i, MAC_i\}$, Compute: $R_{MCN_j} = PUF(C_{MCN_j})$, $(SID_{MCN_j} \Vert SP_{MCN_j}) = X_{MCN_j} \oplus h(R_{MCN_j})$, $K_i = SP_{MCN_j},\ AD_2 = SID_{MCN_j}$, $IV_2 = SID_{MCN_j}$, $\{PT_i, \perp\} = D_{K_i}(IV_2, AD_2, CT_i,\ MAC_i)$: Error, if verification of MAC_i fails, Else, retrieve $PT_i = \{SID_{SD_k}, SP_{UD_i}\}$; **AKA-3:** Compute: $IV_3 = SID_{UD_i} \oplus r_2 \oplus T_1$, $K_b = SP_{UD_i}, AD_3 = SID_{UD_i}$, $\{PT_2, \perp\} = D_{K_b}(IV_3, AD_3, CT_1, MAC_1)$: Error, if verification of MAC_1 fails, Else, retrieve $PT_2 = \{SID_{SD_k} \Vert r_1\}$, Check SID_{SD_k} is in PT_i, if not, abort **AKA-4:** Search: SID_{SD_k} and retrieve $\{CT_{SD_k}, MAC_{SD_k}\}$, $\{PT_3, \perp\} = D_{K_i}(IV_2, AD_2, CT_{SD_k},\ MAC_{SD_k})$: Error, if verification of MAC_{SD_k} fails, Else, retrieve $PT_3 = \{SP_{D_k}\}$; **AKA-5:** Compute: $X_3 = h(SID_{UD_i} \Vert r_1)$, $ID_n = X^a_3 \oplus X^b_3$, Update: $SID^p_{UD_i}$ with SID_{UD_i} and $SID^c_{UD_i}$ with ID_n, Pick: r_3, r_4, T_2, Compute: $IV_4 = T_2 \oplus r_3,\ K_c = SP_{D_k}$, $AD_4 = SID_{SD_k},\ PT_4 = (X_3 \Vert r_4)$, $\{CT_2,\ MAC_2\} = E_{K_c}(IV_4, AD_4, PT_4)$; $\xrightarrow{MSG_2=\{CT_2,\ MAC_2,\ r_3,\ T_2\}}$ (Via public channel)	**AKA-6:** Confirm if $\lvert T_2 - T^*_2 \rvert \overset{?}{\leq} \Delta T$; otherwise, terminate. Retrieve: C_{SD_k}, X_{SD_k}, Compute: $R_{SD_k} = PUF(C_{SD_k})$, $(SID_{SD_k} \Vert SP_{SD_k}) = X_{SD_k} \oplus h(R_{SD_k})$, $IV_5 = T_2 \oplus r_3,\ K_d = SP_{D_k},\ AD_5 = SID_{SD_k}$, $\{PT_5, \perp\} = D_{K_d}(IV_5, AD_5, CT_2,\ MAC_2)$: If MAC_2 verification fails, raise an error; otherwise, retrieve $PT_5 = \{X_3 \Vert r_4\}$; **AKA-7:** Pick: r_5, T_3, Split: X_3 into X^a_3 and X^b_3, Compute: $IV_6 = X^b_3 \oplus T_3,\ K_e = X^a_3,\ AD_6 = SID_{SD_k}$, $PT_6 = r_4 \oplus r_5 \oplus SP_{SD_k} \oplus SID_{SD_k}$, $SK_{D_kU_i} = h(X_3 \Vert PT_6 \Vert T_3)$, $\{CT_3,\ MAC_3\} = E_{K_e}(IV_6, AD_6, PT_6)$; $\xleftarrow{MSG_3=\{CT_3,\ MAC_3,\ T_3\}}$ (to UD_i via public channel)
$SK_{U_iD_k}(= SK_{D_kU_i}) = h((h(SID_{UD_i} \Vert r_1)) \Vert (r_4 \oplus r_5 \oplus SP_{D_k} \oplus SID_{SD_k}) \Vert T_3)$		

Fig. 2: The proposed scheme encompasses procedures for login, authentication, and session key agreement.

AKA 3: MCN_j additionally computes IV_3 as XOR of SID_{UD_i}, r_2, and T_1, sets K_b to SP_{UD_i} and AD_3 to SID_{UD_i}. It then uses these values, along with CT_1 and MAC_1 as well as by employing AEGIS, to compute $\{PT_2, \perp\} = D_{K_b}(IV_3, AD_3, CT_1, MAC_1)$. If the verification of MAC_1 fails, MCN_j aborts the procedure. Otherwise, MCN_j retrieves $SID_{SD_k} \| r_1$ from plaintext PT_i. MCN_j then checks that SID_{SD_k} is in PT_i. If it is not, the process is aborted.

AKA 4: Next, MCN_j searches SID_{SD_k} and retrieves the corresponding ciphertext and message authentication code pair $\{CT_{SD_k}, MAC_{SD_k}\}$. It then computes $\{PT_3, \perp\} = D_{K_i}(IV_2, AD_2, CT_{SD_k},\ MAC_{SD_k})$. If verification of MAC_{SD_k} fails, MCN_j aborts the AKA procedure. Otherwise, MCN_j retrieves $\{SP_{D_k}\}$ from plaintext PT_3.

AKA 5: In order to derive additional parameters, MCN_j performs some computations. First, it computes X_3 by taking the hash of the concatenation of SID_{UD_i} and r_1. Then, X_3 is split into two parts of 128 bits each to obtain X^a_3 and X^b_3. ID_n is derived from X^a_3 and X^b_3 by applying the XOR operation. After computing these values, MCN_j updates $SID^p_{UD_i}$ with the value of SID_{UD_i}, and $SID^c_{UD_i}$ with the value of ID_n. MCN_j then picks two random numbers r_3 and r_4 and current timestamp T_2, and computes $IV_4 = T_2 \oplus r_3$, $K_c = SP_{D_k}$, $AD_4 = SID_{SD_k}$, $PT_4 = (X_3 \| r_4)$, and $\{CT_2, MAC_2\} = E_{K_c}(IV_4, AD_4, PT_4)$. Finally, MCN_j constructs message MSG_2 and transmits it to SD_k via an open channel.

AKA 6: SD_k first verifies the freshness of the received MSG_2. If fresh, SD_k retrieves its own parameters C_{SD_k} and X_{SD_k}. Then it computes $R_{SD_k} = PUF(C_{SD_k})$, $(SID_{SD_k} \| SP_{SD_k}) = X_{SD_k} \oplus h(R_{SD_k})$, $IV_5 = T_2 \oplus r_3$, $K_d = SP_{D_k}$, $AD_5 = SID_{SD_k}$ and $\{PT_5, \perp\} = D_{K_d}(IV_5, AD_5, CT_2, MAC_2)$. If the verification of MAC_2 fails, SD_k aborts the AKA procedure. Otherwise, it retrieves $\{X_3 \| r_4\}$ from plaintext PT_5.

AKA 7: Furthermore, SD_k picks current timestamp T_3 and a random number r_5 and then split X_3 into X^a_3 and X^b_3. SD_k computes $IV_6 = X^b_3 \oplus T_3$, $K_e = X^a_3$, $AD_6 = SID_{SD_k}$, $PT_6 = r_4 \oplus r_5 \oplus SP_{SD_k} \oplus SID_{SD_k}$, $\{CT_3,\ MAC_3\} = E_{K_e}(IV_6, AD_6, PT_6)$, and the session key shared with U_i as $SK_{D_kU_i} = h(X_3 \| PT_6 \| T_3)$. Finally, SD_k constructs a message MSG_3 that includes $\{CT_3,\ MAC_3,\ T_3\}$, and transmits it to UD_i via an open channel.

AKA 8: UD_i verifies the freshness of the received MSG_3. If fresh, UD_i then computes $X_4 = h(SID_{UD_i} \| r_1)$ and then split X_4 into two equal size parts X_4^a and X_4^b each of size 128 bits. Next, UD_i further computes $IV_7 = X_4^b \oplus T_3$, $K_f = X_4^a$, $AD_7 = SID_{SD_k}$, and $\{PT_7, \perp\} = D_{K_f}(IV_7, AD_7, CT_3, MAC_3)$. If verification of MAC_3 fails, UD_i aborts the AKA procedure. Otherwise, UD_i and SD_k successfully established the session key, which is computed as $SK_{U_iD_k} = h(X_4 \| PT_7 \| T_3)$, and the updated SID_{UD_i} is computed as $SID_{UD_i} = X_4^a \oplus X_4^b$.

Fig. 2 summarizes the login and AKA procedure with the associated interactions between the participating parties.

E. Password Update Phase

When U_i needs to update its password, the below steps are required to accomplish this task.

Step 1: First, ID_{U_i} and the current password PW_i^o, must be entered into UD_i to begin the password update process.

Step 2: Second, UD_i computes $R_i^o = PUF(PW_i^o)$, $(rn_1 \| rn_2) = X_{UD_i} \oplus h(ID_i \| PW_i^o)$, and $X_i = h(ID_i \| PW_i^o \| R_i^o)$. UD_i further computes $K_{UD_i} = X_i^a \oplus X_i^b$ and $\{PT_{UD_i}, \perp\} = D_{K_{UD_i}}(IV_i, AD_i, CT_{UD_i}, MAC_{UD_i})$, where $IV_i = rn_1$, $AD_i = rn_2$. If the validation of MAC_{UD_i} does not succeed, the process of updating the password is terminated. Alternatively, if successful, UD_i prompts U_i to input a new password, denoted as PW_i^n, and subsequently recalculates the following parameters again $X_{UD_i}^n = (r_1 \| r_2) \oplus h(ID_i \| PW_i^n)$, $R_i^n = PUF(PW_i^n)$, $X_{ii} = h(ID_i \| PW_i^n \| R_i^n)$, $K_{UD_i}^n = X_{ii}^a \oplus X_{ii}^b$, and $\{CT_{UD_i}^n, MAC_{UD_i}^n\} = E_{K_{UD_i}^n}(IV_i, AD_i, PT_i)$, where $IV_i = r_1$, $AD_i = r_2$, and $PT_i = SP_{UD_i} \| SID_{SD_k}$.

Step 3: Lastly, once the user's password has been successfully updated, UD_i stores the updated parameters $\{SID_{UD_i}$, $CT_{UD_i}^n, MAC_{UD_i}^n, X_{UD_i}^n, PUF(\cdot)\}$ in its own memory.

F. Revocation

In the event that a legitimate user U_i loses their user device UD_i, the TA has the capability to register and issue a new device UD_i^{new} for U_i. To initiate this process, U_i must provide their previous identity ID_{U_i}, along with a physical verification step, such as an ID card (or a similar document), to ensure that the identity is not hijacked by an adversary. The following steps outline the revocation procedure.

Step 1: U_i transmits their previous identity ID_{U_i} to the TA along with the physical ID card (or similar document) to prove their identity. TA conducts a search for ID_{U_i} within its database. Upon finding a matching record, TA proceeds to remove the associated entry linked to ID_{U_i} and prompts U_i to initiate a new registration request.

Step 2: Once U_i receives the message from TA, it generates a new and unique identity represented as $ID_{U_i}^{new}$. U_i securely transmits the registration request message $< ID_{U_i}^{new} >$ to TA. The following steps follow the procedure outlined in Section III-B.

Step 3: U_i keeps $\{SID_{UD_i}^{new}$, $CT_{UD_i}^{new}$, $MAC_{UD_i}^{new}$, $X_{UD_i}^{new}$, $PUF(\cdot)\}$ in UD_i^{new}. TA also forwards the relevant secret credentials to the corresponding MCN as discussed in Section III-B.

IV. Security Analysis

In this section, we examine the security aspects of the proposed authentication scheme. We evaluate the security measures incorporated in our scheme to confirm its effectiveness across various scenarios. The formal security analysis is explained below.

A. Formal Analysis of Security using ROR Model

The ROR model is employed to examine the proposed scheme, showcasing its semantic security and confirming its achievement of the necessary session key security (SK-security) levels. Initially, we present the ROR model of the proposed scheme, followed by an analysis of its SK-security.

Our scheme is evaluated using the ROR model, which assigns the t^{th} instance of an entity Π as Π^t. Specifically, user U_i, MCN MCN_j and smart sensing device SD_k are represented as Π_{U_i}, Π_{MCN_j} and Π_{SD_k}, respectively, and their t_1^{th}, t_2^{th}, and t_3^{th} instances are denoted as $\Pi_{U_i}^{t_1}$, $\Pi_{MCN_j}^{t_2}$ and $\Pi_{SD_k}^{t_3}$ correspondingly. A collision-resistant one-way hash function $h(\cdot)$ and the PUF function $PUF(\cdot)$ are treated as random oracles, publicly accessible to all entities in the ROR model. Additionally, adversary $\mathcal{A}$ is provided with a set of queries to simulate an attack under the ROR model.

- $Execute(\Pi_{U_i}^{t_1}, \Pi_{MCN_j}^{t_2}, \Pi_{SD_k})$: When this query is executed, $\mathcal{A}$ can intercept all communications exchanged between U_i, MCN_j and SD_k. Therefore, this query is regarded as an eavesdropping attack by $\mathcal{A}$ due to the intercepted messages.
- $Reveal(\Pi^t)$: By executing this query, $\mathcal{A}$ can unveil the session key SK generated between $\Pi_{U_i}^{t_1}$ and Π_{SD_k}.
- $Send(\Pi^t, MSG)$: This query enables $\mathcal{A}$ to transmit the message MSG to Π^t and acquire the corresponding response message.
- $CorruptUD(\Pi_{U_i}^{t_1})$: This query enables $\mathcal{A}$ to obtain the confidential parameters that are saved in the stolen user device.
- $CorruptSD(\Pi_{SD_k}^{t_2})$: This query enables $\mathcal{A}$ to obtain the confidential parameters that are saved in the stolen smart sensing device.
- $Test(\Pi^t)$: With this query, $\mathcal{A}$ can request the SK from Π^t, which responds with a randomized outcome determined by the unbiased coin flip result b.

Let's introduce some key definitions that form the basis of our formal analysis:

Definition 1. *Assuming that $\mathcal{A}$ has a polynomial-time complexity of t_p and is making at most $\mathcal{Q}$ queries to an encryption/decryption oracle with a length of $\mathcal{L}_{ED}$, the advantage of $\mathcal{A}$ in the online chosen ciphertext attack (OCCA3) can be expressed as follows:*

$$Adv_{\phi,\mathcal{A}}^{OCCA3}(\mathcal{Q}, \mathcal{L}_{ED}, t_p) \leq Adv_{\phi}^{OPRP-CPA}(\mathcal{Q}, \mathcal{L}_{ED}, t_p) + Adv_{\phi}^{INT-CT}(\mathcal{Q}, \mathcal{L}_{ED}, t_p), \tag{1}$$

where $Adv^{OPRP-CPA}\phi(\mathcal{Q}, \mathcal{L}ED, tp)$ denotes the advantage of $\mathcal{A}$ in an 'online pseudo-random permutation chosen-plaintext' attack, and $Adv^{INT-CT}\phi(\mathcal{Q}, \mathcal{L}_{ED}, t_p)$ is the advantage of $\mathcal{A}$ in maintaining the integrity of the ciphertext.

Definition 2. (Semantic Security): *The security of the secret session key SK established between U_i and SD_k within the ROR model is contingent upon the attacker $\mathcal{A}$'s capability to differentiate between the correct SK and a randomly guessed SK. Let b denote the correct bit and b' represent a bit randomly guessed by $\mathcal{A}$. The success probability of $\mathcal{A}$ is denoted as SU. The advantage of $\mathcal{A}$ in breaching the SK security, which is established during the AKA phase of the proposed scheme $\mathcal{P}$, can be expressed as*

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t_p) = |2 \cdot Prob[SU] - 1|, \tag{2}$$

where $Prob[SU]$ is the probability of $\mathcal{A}$ guessing the correct bit b. The scheme $\mathcal{P}$ is considered secure if $Adv_{\mathcal{A}}^{\mathcal{P}}$ is negligible under the ROR model.

Having established these foundational definitions, we now present the following theorem derived from the AKA phase:

Theorem 1. *Let $\mathcal{A}$ be an attacker attempting to extract the SK established between U_i and SD_k by running against the proposed scheme $\mathcal{P}$ within polynomial time tp. The number of queries made by $\mathcal{A}$, including $Send$, $Hash$, and PUF queries, are denoted as Q_s, Q_h, and $Qpuf$ respectively. The function $h(\cdot)$ has a range space of $|Hash|$, the PUF has a key length of $|PUF|$, and the uniformly distributed password dictionary has a size of $|DT|$. The advantage of $\mathcal{A}$ in compromising the AEGIS scheme is given by $Adv_{\phi,\mathcal{A}}^{OCCA3}(\mathcal{Q}, \mathcal{L}ED, t_p)$ (as defined in (1)). Thus, the advantage of $\mathcal{A}$ in successfully obtaining the SK established between U_i and SD_k can be characterized as follows:*

$$\begin{aligned} Adv_{\mathcal{A}}^{\mathcal{P}}(t_p) \leq & \frac{Q_h^2}{|Hash|} + \frac{Q_{puf}^2}{|PUF|} + \frac{2 \cdot Q_s}{|DT|} \\ & + 2 \cdot Adv_{\phi,\mathcal{A}}^{OCCA3}(\mathcal{Q}, \mathcal{L}_{ED}, t_p). \end{aligned} \tag{3}$$

Proof. The proof involves six games that employ the same queries as those discussed earlier.

Game$_0$: Game$_0$ represents an actual attack conducted by $\mathcal{A}$ against the proposed $\mathcal{P}$ within the realm of the ROR model. The result of Game$_0$ is determined by flipping an unbiased coin, and therefore

$$\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) = |2 \cdot \text{Prob}[SU_0] - 1|. \tag{4}$$

Game$_1$ involves simulating an eavesdropping attack by $\mathcal{A}$, intercepting and monitoring all communication between U_i, MCN_j, and SD_k during the AKA procedure. $\mathcal{A}$ then queries $Execute(\Pi_{U_i}^{t_1}, \Pi_{MCN_j}^{t_2}, \Pi_{SD_k})$, proceeds with $Test$ and $Reveal$ to verify the authenticity of $SK_{U_iD_k}(= SK_{D_kU_i})$. Short-term and long-term secrets are used to calculate SK between U_i and SD_k. $\mathcal{A}$'s computation of SK is demanding, but the probability of $\mathcal{A}$ winning remains the same as in Game$_0$, thus rendering Game$_0$ and Game$_1$ indistinguishable.

$$\text{Prob}[SU_1] = \text{Prob}[SU_0]. \tag{5}$$

Game$_2$: In this scenario, both the $Hash$ and $Send$ queries are employed to simulate an active attack. $\mathcal{A}$ utilizes multiple $Hash$ queries to detect hash collisions. However, due to the inclusion of random numbers and timestamps in every message of $\mathcal{P}$, the occurrence of hash collisions becomes highly unlikely when $\mathcal{A}$ initiates a $Send$ query. Consequently, the birthday paradox leads us to the following conclusion:

$$|\text{Prob}[SU_2] - \text{Prob}[SU_1]| \leq \frac{Q_h^2}{2|Hash|}. \tag{6}$$

Game$_3$: Game$_3$ is an extension of Game$_2$ that simulates $PUF()$ query. Since PUFs in UD_i and SD_k are secure,

$$|\text{Prob}[SU_3] - \text{Prob}[SU_2]| \leq \frac{Q_{puf}^2}{2|PUF|}. \tag{7}$$

Game$_4$: Game$_4$ simulates attacks on lost or stolen UD_i and password guessing. The objective is for $\mathcal{A}$ to retrieve the encrypted secret SP_{UD_i} by successfully determining both ID_i and PW_i within a limited number of guesses and attempts from DT. During the game, $\mathcal{A}$ can utilize the $CorruptUD(\Pi_{U_i}^{t_1})$ query, which allows them to obtain the following information from a stolen or lost UD_i: $\{SID_{UD_i}, CT_{UD_i}, MAC_{UD_i}, X_{UD_i}\}$. The winning condition for $\mathcal{A}$ is to successfully determine both ID_i and PW_i by making informed guesses and attempts from DT. Consequently,

$$|\text{Prob}[SU_4] - \text{Prob}[SU_3]| \leq \frac{Q_s}{|DT|}. \tag{8}$$

Game$_5$: In this game, $\mathcal{A}$ aims to obtain the session keys by carrying out an active attack and using all intercepted messages MSG_1, MSG_2 and MSG_3 from U_i, MCN_j and SD_k, as well as other secret parameters acquired from the previous games. To achieve this, $\mathcal{A}$ must calculate $SK_{U_iD_k}(= SK_{D_kU_i}) = h((h(SID_{UD_i}\|r_1))\|(r_4 \oplus r_5 \oplus SP_{D_k} \oplus SID_{SD_k})\|T_3)$. Note that AEGIS encryption algorithm secures all short-term and long-term secrets and identities utilized to create an SK in $\mathcal{P}$, as explained in **Definition 1**. Therefore, we have

$$|\text{Prob}[SU_5] - \text{Prob}[SU_4]| \leq \text{Adv}_{\phi,\mathcal{A}}^{OCCA3}(\mathcal{Q}, \mathcal{L}_{ED}, t_p). \tag{9}$$

Upon finishing all games, $\mathcal{A}$ executes $Test$ query, and flips a fair coin to evaluate the semantic security of the SK. Therefore, the probability of $\mathcal{A}$ being successful is

$$\text{Prob}[SU_5] = \frac{1}{2}. \tag{10}$$

Now from (4), we obtain:

$$\frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) = \Big|\text{Prob}[SU_0] - \frac{1}{2}\Big|. \tag{11}$$

By utilizing (10) and (11) as well as taking into account equation (5), we can derive the following result:

$$\begin{aligned} \frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) =& |\text{Prob}[SU_0] - \text{Prob}[SU_5]| \\ =& |\text{Prob}[SU_1] - \text{Prob}[SU_5]|. \end{aligned} \tag{12}$$

When the widely recognized triangle inequality is applied to (12), it results in

$$\begin{aligned} \frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) \leq & |\text{Prob}[SU_1] - \text{Prob}[SU_2]| \\ & + |\text{Prob}[SU_2] - \text{Prob}[SU_3]| \\ & + |\text{Prob}[SU_3] - \text{Prob}[SU_4]| \\ & + |\text{Prob}[SU_4] - \text{Prob}[SU_5]|. \end{aligned} \tag{13}$$

Further substituting (6), (7), (8) and (9) into (13) leads to (3). This completes the proof. ■

B. Informal Security Analysis

In this subsection, we conduct a thorough informal security analysis to evaluate the effectiveness of our proposed scheme against potential security threats, which are outlined below.

1) Anonymity and Untraceability: Our AKA scheme ensures anonymity and untraceability by using fresh timestamps and random numbers for message generation, preventing an eavesdropper $\mathcal{A}$ from linking messages across sessions. Each user U_i uses a unique, session-specific pseudonym SID_{UD_i}, updated after each session, to maintain anonymity. This approach also protects against identity guessing attacks by preventing $\mathcal{A}$ from deducing U_i's true identity from transmitted messages.

2) Desynchronization Attack: Our AKA scheme prevents desynchronization attacks by storing both current and previous pseudonyms $(SID^c_{UD_i}, SID^p_{UD_i})$ at the MCN, MCN_j. During the AKA phase, MCN_j updates $SID^c_{UD_i}$ with ID_n and $SID^p_{UD_i}$ with SID_{UD_i}. If $\mathcal{A}$ launches a jamming or packet drop attack, U_i can use the old $SID^p_{UD_i}$ to complete the session, as MCN_j keeps both identities. After a successful AKA session, SID_{UD_i} is updated with ID_n on U_i's side, maintaining anonymity and privacy. Additionally, in light of the potential for a timestamp-based desynchronization attack by $\mathcal{A}$, the proposed scheme mitigates such threats by embedding timestamps in the computation of authentication tags (CT, MAC). Specifically, each message in the protocol is accompanied by a timestamp, and the (CT, MAC) pair is generated with random nonces, timestamps, and unique session identifiers. If an attacker attempts to alter the timestamp in an effort to desynchronize the session states between the communicating parties, the altered message will fail the authentication check due to the mismatch in the computed MAC, which includes the timestamp. As a result, the message will be rejected, and the authentication process will be terminated immediately, ensuring a robust defense against timestamp-based desynchronization attacks.

3) Password Guessing Attacks: Our scheme prevents password guessing attacks by never transmitting user passwords in plaintext or masked form. Even if $\mathcal{A}$ accesses values $\{SID_{UD_i}, CT_{UD_i}, MAC_{UD_i}, X_{UD_i}\}$, guessing the password requires knowing ID_i, which is infeasible. Thus, our scheme is secure against both online and offline password-guessing attacks.

4) Replay Attacks: Our AKA scheme prevents replay attacks by embedding timestamps in messages MSG_1 through MSG_3. If $\mathcal{A}$ replays these messages, the recipient can detect the attack through timestamp verification. This ensures the integrity and confidentiality of communication.

5) Man-in-the-middle Attack: $\mathcal{A}$ may try a man-in-the-middle attack between U_i and MCN_j by manipulating MSG_1. However, this requires knowledge of SP_{UD_i}, SID_{SD_k}, and r_1, making it unlikely to succeed. Even if $\mathcal{A}$ is a registered user U_l, it can't generate valid CT_1 and MAC_1 for U_i. Similarly, intercepting and fabricating MSG_2 without SP_{SD_k}, SID_{SD_k}, and r_4 is impossible. Furthermore, tampering with MSG_3 is prevented due to untampered CT_3 and MAC_3. Thus, our scheme is resilient to such attacks.

6) ESL Attack: Within our scheme, the session key $SK_{D_kU_i}$ is ephemeral, being generated afresh in each iteration of the AKA phase as detailed in Section III-D. SD_k and U_i compute this key using a hash function h with short and long-term secrets. Security analysis in two scenarios:

- Case 1: Even if adversary $\mathcal{A}$ has knowledge of the short-term (ephemeral) keys r_1, r_4 and r_5, it is still unable to compute the session key $SK_{D_kU_i}$ without knowledge of the long-term secrets SP_{D_k} and SID_{SD_k} due to AEGIS primitives and $h(\cdot)$.
- Case 2: Even if $\mathcal{A}$ has complete knowledge of the long-term secrets SP_{D_k} and SID_{SD_k}, it remains computationally infeasible for $\mathcal{A}$ to compute the session key $SK_{D_kU_i}$ without knowledge of the short-term keys r_1, r_4 and r_5 due to AEGIS primitives and $h(\cdot)$, which ensure that the session key cannot be calculated without knowledge of the short-term keys.

The session key $SK_{D_kU_i}$ depends on both ephemeral and long-term secrets, providing forward and backward secrecy. Leakage of $SK_{D_kU_i}$ doesn't affect past or future session keys. Our scheme is resilient against attacks targeting ephemeral secrets leakage.

7) Physical Smart Device Capture Attack: Smart sensing devices are often deployed in hostile environments, and it is possible for $\mathcal{A}$ to physically capture smart device SD_k from a FMS environment. Then $\mathcal{A}$ may attempt to extract secret data from the device's memory, including X_{SD_k}, through physical attacks. However, retrieving the embedded challenge and response pair (CRP) (C_{SD_k}, R_{SD_k}) in the PUF of SD_k requires $\mathcal{A}$ to probe or modify the integrated circuit, which will permanently alter the small physical changes in the circuit and destroy the PUF. Therefore, even if $\mathcal{A}$ manages to obtain X_{SD_k} successfully, it cannot recover the valid CRP. Consequently, our scheme is resilient and immune to captured smart sensing device attacks.

8) Stolen User Device Attack: Assuming that adversary $\mathcal{A}$ has gained unauthorized access to registered user U_i's device UD_i, it is important to note that $\mathcal{A}$ cannot obtain user's sensitive attributes, such as SP_{UD_i} and SID_{SD_k}, without knowledge of user's identity ID_i and password PW_i, as outlined in Section III-B. Furthermore, any tampering attempts made to alter the values of $\{CT_{UD_i}, MAC_{UD_i}, X_{UD_i}\}$ on UD_i will result in validation failure during the login phase, while modifying SID_{UD_i} will result in validation failure during the AKA phase at MCN_j. Therefore, our scheme ensures the protection of registered user's sensitive information even in the event of its device UD_i being stolen.

9) Privileged Insider Attack: In our scheme, even if adversary $\mathcal{A}$ has privileged access to TA and intercepts user registration requests ID_i transmitted securely, accessing registered user device UD_i and extracting stored credentials is fruitless. This is because sensitive credentials are protected by a collision-resistant hash function $h(\cdot)$ and AEGIS primitive, making guessing infeasible for $\mathcal{A}$. Additionally, without prior knowledge of the user's identity ID_i and password PW_i, $\mathcal{A}$ cannot determine sensitive parameters SP_{UD_i} and SID_{SD_k}. Thus, our scheme is resilient against privileged insider attacks.

10) Impersonation Attacks: Suppose that adversary $\mathcal{A}$ attempts to create valid authentication request message on behalf

TABLE II: Comparison of computation overheads (in milliseconds)

Scheme	**User**	**Master Controller Node/Gateway**	**Smart Sensing Device**	**Total Overhead**
Das *et al.* [29]	$14T_{\mathcal{H}} + 1T_{\mathcal{F}} \approx 6.647$	$9T_{\mathcal{H}} \approx 0.009$	$7T_{\mathcal{H}} \approx 0.049$	6.705
Chen *et al.* [18]	$3T_{\mathcal{E}} + 5T_{\mathcal{H}} \approx 19.682$	$1T_{\mathcal{E}} + 7T_{\mathcal{H}} \approx 1.826$	$2T_{\mathcal{E}} + 3T_{\mathcal{H}} \approx 13.119$	34.672
Far *et al.* [30]	$2T_{\mathcal{E}} + 9T_{\mathcal{H}} + T_{\mathcal{F}} \approx 19.71$	$1T_{\mathcal{E}} + 10T_{\mathcal{H}} \approx 1.829$	$5T_{\mathcal{H}} \approx 0.035$	21.574
Yang *et al.* [31]	$10T_{\mathcal{H}} \approx 0.07$	$19T_{\mathcal{H}} \approx 0.019$	$8T_{\mathcal{H}} \approx 0.056$	0.145
Tanveer *et al.* [32]	$4T_{\mathcal{AE}} + 3T_{\mathcal{H}} + T_{\mathcal{F}} \approx 12.618$	$3T_{\mathcal{AE}} + T_{\mathcal{H}} \approx 0.136$	$2T_{\mathcal{AE}} + T_{\mathcal{H}} \approx 3.031$	15.785
Proposed Scheme	$T_{\mathcal{P}} + 3T_{\mathcal{AE}} + 4T_{\mathcal{H}} \approx 4.564$	$T_{\mathcal{P}} + 4T_{\mathcal{AE}} + 2T_{\mathcal{H}} \approx 0.182$	$T_{\mathcal{P}} + 2T_{\mathcal{AE}} + 2T_{\mathcal{H}} \approx 3.038$	7.784

TABLE III: Transmission parameters and their sizes

Transmission Parameter	**Size (bits)**
Random Number	128
Authentication Tag	128
User Identity	128
Hash Output	256
Elliptic Curve Point (ECC)	160
Timestamp	32

of user U_i. In order to accomplish this task, $\mathcal{A}$ needs to choose a value $T_1^{\mathcal{A}}$ as well as two random numbers $r_1^{\mathcal{A}}$ and $r_2^{\mathcal{A}}$. It then computes $IV_1^{\mathcal{A}} = SID_{UD_i}^{\mathcal{A}} \oplus r_2^{\mathcal{A}} \oplus T_1^{\mathcal{A}}$, $K_a = SP_{UD_i}$, $AD_1 = SID_{UD_i}$, $PT_1^{\mathcal{A}} = SID_{SD_k} \| r_1^{\mathcal{A}}$, $\{CT_1^{\mathcal{A}}, MAC_1^{\mathcal{A}}\} = E_{K_a}(IV_1^{\mathcal{A}}, AD_1, PT_1^{\mathcal{A}})$. However, $\mathcal{A}$ will find it difficult to produce a valid AKA message MSG_1, to impersonate U_i in the FMS environment without knowledge of the secret credentials $\{SID_{SD_k}, SP_{UD_i}\}$. The same holds true for the other communicated messages during the AKA process, i.e., MSG_2 and MSG_3. As a result, our scheme is safeguarded against attacks that attempt to impersonate MCN_j, U_i, and SD_k.

V. Comparative Analysis

In this section, we present a comparison with detailed analysis for the proposed scheme and other similar existing state-of-the-art schemes, including Das *et al.* [29], Chen *et al.* [18], Far *et al.* [30], Yang *et al.* [31], and Tanveer *et al.* [32].

A. Comparison of Computation Overheads

This section presents a comparative analysis of the computational overheads of the proposed scheme against existing state-of-the-art schemes. Operations such as XOR and concatenation are excluded from the evaluation due to their negligible computational costs. To ensure a thorough assessment, the basic cryptographic primitives are tested on two distinct hardware platforms. For resource-constrained devices, such as user devices and smart sensing devices, a Raspberry Pi 4 with 2 GiB of memory running Raspberry Pi OS (32-bit) is utilized. For devices with higher computational capabilities, such as master controller nodes, servers, or gateways, a Windows 11 machine with 16 GiB of memory, an Intel® Core™ i5-12500H CPU @ 3 GHz, and a 64-bit operating system is employed. Each cryptographic primitive is executed 1,000 times, and the average execution time is computed on both platforms to ensure reliable results. The average execution times (in milliseconds) for various cryptographic operations are recorded as follows: for hashing operations, $T_{\mathcal{H}}$, AEGIS encryption/decryption, $T_{\mathcal{AE}}$, elliptic curve point multiplication, $T_{\mathcal{E}}$, and fuzzy extractor operations (approximated as $T_{\mathcal{F}} \approx T_{\mathcal{E}}$). On resource-rich computing platforms, the average execution times are: $T_{\mathcal{H}} = 0.001$ ms, $T_{\mathcal{AE}} = 0.045$ ms, and $T_{\mathcal{E}} = 1.819$ ms. On resource-constrained platforms, the corresponding times are: $T_{\mathcal{H}} = 0.007$ ms, $T_{\mathcal{AE}} = 1.512$ ms, and $T_{\mathcal{E}} = 6.549$ ms. For PUF operations, data reported in [33] is referenced, indicating that the execution time for resource-constrained devices is $T_P = 0.4\mu$s, while for resource-rich devices, it is negligible.

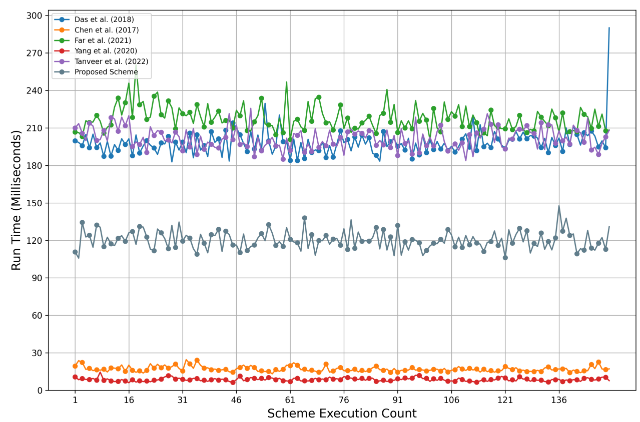


Fig. 3: Comparison of Scheme Runtime

Based on these reported execution times, the computational overhead of our proposed scheme as well as the state-of-the-art schemes are computed. The evaluation results are presented in Table II. The total computational overhead of the proposed scheme is 7.784 ms, which represents an 77.55% improvement over Chen *et al.* [18] (34.672 ms), a 63.92% improvement over Far *et al.* [30] (21.574 ms), and a 50.69% improvement over Tanveer *et al.* [32] (15.785 ms). Although the proposed scheme has a slightly higher overhead compared to Das *et al.* [29] and Yang *et al.* [31], it compensates by offering enhanced security features (see Table V). This tradeoff justifies the marginal increase in computational overhead.

B. Comparison of Scheme Runtime

In this subsection, to rigorously assess the performance of the proposed scheme, we implemented and evaluated its complete execution overhead alongside several state-of-the-art schemes, including those by Das *et al.* [29], Chen *et al.* [18], Far *et al.* [30], Yang *et al.* [31], and Tanveer *et al.* [32], on a designated experimental machine. The experimental setup comprised a system equipped with 16 GB of RAM

TABLE IV: Communication overheads comparison

Scheme	No. of messages	No. of bits
Das *et al.* [29]	3	2400
Chen *et al.* [18]	4	2784
Far *et al.* [30]	4	3200
Yang *et al.* [31]	6	5376
Tanveer *et al.* [32]	3	1632
Proposed scheme	3	1632

and a 12th Gen Intel® Core™ i5-12500 @ 3 GHz processor, operating under Windows 11. Furthermore, a Python-based testing script is executed 100 times to capture the variability and compute the average execution times of the different schemes. Fig. 3 depicts the runtime fluctuations of the proposed scheme in comparison with other benchmark schemes. Based on the experimental data, the average runtimes for the proposed scheme and the benchmark schemes are as follows: our scheme achieved an average runtime of 120.318 ms; Das *et al.* [29] reported 198.486 ms; Chen *et al.* [18] documented 17.08 ms; Far *et al.* [30] measured 217.348 ms; Yang *et al.* [31] recorded 8.699 ms; and Tanveer *et al.* [32] registered 202.058 ms. In comparison to [29], [30], and [32], the proposed protocol demonstrates substantial reductions in overall runtime overhead, achieving decreases of 39.38%, 44.73%, and 40.45%, respectively. Additionally, although the proposed scheme leverages the more efficient AEAD primitive AEGIS, the incorporation of additional secret credential retrieval operations introduces extra runtime overhead, resulting in a slightly higher overall runtime than [18] and [31]. However, considering that our scheme integrates more comprehensive security features (see Table V) and achieves lower communication overhead (see Table IV), this increase is justifiable.

C. Comparison of Communication Overheads

Efficient communication management is a pivotal design goal for AKA schemes. To evaluate the communication efficiency of the proposed scheme, a comparative analysis is conducted against five state-of-the-art AKA schemes, including Das *et al.* [29], Chen *et al.* [18], Far *et al.* [30], Yang *et al.* [31], and Tanveer *et al.* [32]. The comparison results are summarized in Table IV, focusing on the number of messages exchanged during a single AKA cycle as well as the number of bits transmitted. In all schemes considered, the transmitted parameters include random numbers, timestamps, hash outputs, user identities, ECC points, and authentication tags. To ensure a fair comparison, the sizes of the parameters are considered as shown in Table III: random numbers and authentication tags are 128 bits, timestamps are 32 bits, user identities are 128 bits, hash outputs are 256 bits, and ECC points are 160 bits. In the proposed scheme, three messages are exchanged during the AKA process: $MSG_1 = \{SID_{UD_i},\ CT_1,\ MAC_1,\ r_2,\ T_1\}$, $MSG_2 = \{CT_2,\ MAC_2,\ r_3,\ T_2\}$, and $MSG_3 = \{CT_3,\ MAC_3,\ T_3\}$, have sizes of $\{128 + 256 + 128 + 128 + 32\} = 672$ bits, $\{384 + 128 + 128 + 32\} = 672$ bits, and $\{128 + 128 + 32\} = 288$ bits, respectively. Therefore, the total communication overhead sums up to $\{672 + 672 + 288\} = 1,632$ bits, which is the lowest among the compared schemes, as illustrated in Table IV. This is significantly lower compared to the baseline schemes: $2,400$ bits in Das *et al.* [29], $2,784$ bits in Chen *et al.* [18], $3,200$ bits in Far *et al.* [30], and $5,376$ bits in Yang *et al.* [31]. This reduction translates to a 32.0%, 41.4%, 49.0%, and 69.6% improvement, respectively. While the communication overhead in Tanveer *et al.* [32] is identical to that of the proposed scheme, the latter offers enhanced security features (see Table V). These results highlight the efficiency and security balance achieved by the proposed scheme.

TABLE V: Analysis of security and functionality features

Feature	[29]	[18]	[30]	[31]	[32]	Proposed
$\mathcal{FS}_1$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_2$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_3$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_4$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_5$	×	×	✓	✓	✓	✓
$\mathcal{FS}_6$	✓	×	×	×	×	✓
$\mathcal{FS}_7$	×	✓	✓	✓	✓	✓
$\mathcal{FS}_8$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_9$	×	✓	✓	×	✓	✓
$\mathcal{FS}_{10}$	✓	✓	✓	✓	✓	✓
$\mathcal{FS}_{11}$	✓	×	✓	×	✓	✓
$\mathcal{FS}_{12}$	×	×	✓	✓	✓	✓
$\mathcal{FS}_{13}$	✓	✓	×	✓	✓	✓
$\mathcal{FS}_{14}$	✓	✓	✓	✓	✓	✓

Note: '✓' indicates that the feature is available and '×' means that the feature is unavailable.

D. Comparison of Security and Functionality Features

Table V provides a comprehensive comparison of the key security and functionality features ($\mathcal{FS}_1$: "mutual authentication", $\mathcal{FS}_2$: "key agreement", $\mathcal{FS}_3$: "replay attack", $\mathcal{FS}_4$: "impersonation attacks", $\mathcal{FS}_5$: "untraceability", $\mathcal{FS}_6$: "smart sensing device theft attack", $\mathcal{FS}_7$: "user device capture/theft attack", $\mathcal{FS}_8$: "man-in-the-middle attack", $\mathcal{FS}_9$: "anonymity", $\mathcal{FS}_{10}$: "password update attack", $\mathcal{FS}_{11}$: "privileged insider attack", $\mathcal{FS}_{12}$: "ESL attack", $\mathcal{FS}_{13}$: "desynchronization attack", and $\mathcal{FS}_{14}$: "validated via formal model") between our proposed scheme and five state-of-the-art competitors. The analysis unequivocally demonstrates that our scheme outperforms the other five schemes in terms of these features. Thus, our proposed scheme exhibits superior security strength and comprehensive functionality compared to the alternative schemes.

VI. Conclusions

We have presented a new user authentication and key agreement scheme for the flexible manufacturing system based on IIoT. Our proposed scheme has integrated AEGIS primitive, hash function, and PUF to provide strong security with low computational overhead. Specifically, our scheme guarantees user authenticity, establishes an indecipherable communication channel between users and smart sensing devices through a session key, and enhances physical security by preventing tampering. To further enhance the security and integrity of the system, our scheme has included a revocation phase and a password update phase, requiring the registration of legitimate users and smart sensing devices with the MCN. Through our analysis using the ROR model and informal, we have demonstrated the resilience of our scheme against common types of attacks in IIoT-based environments. Furthermore, we have conducted a thorough comparative analysis with existing benchmark schemes, unequivocally demonstrating that our approach surpasses them in terms of security and functionality features, computational and communication overheads, and runtime efficiency. Despite the robust design of our scheme, a few limitations remain, particularly in addressing potential vulnerabilities to denial of service attacks targeting the MCNs, as well as stability challenges related to physically unclonable functions under noisy conditions. Future work will focus on addressing these issues to further improve the system's resilience and scalability.

References

[1] J. Leng, *et al.*, "Blockchain-secured smart manufacturing in Industry 4.0: A survey," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 1, pp. 237–252, Jan. 2021.

[2] G. Aceto, V. Persico and A. Pescapé, "A survey on information and communication technologies for Industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3467–3501, Fourthquarter 2019.

[3] Z. Bi, L. D. Xu and C. Wang, "Internet of Things for enterprise systems of modern manufacturing," *IEEE Trans. Ind. Inform.*, vol. 10, no. 2, pp. 1537–1546, May 2014.

[4] D. Nimawat and B. D. Gidwani, "An overview of industry 4.0 in manufacturing industries", *Int. J. Ind. Syst. Eng.*, vol. 40, no. 4, pp. 415–454, May 2022.

[5] H. Kayan, *et al.*, "Cybersecurity of industrial cyber-physical systems: A review," *ACM Comput. Surv.*, vol. 54, no. 115, pp. 1–35, Sep. 2022.

[6] I. Ahmad *et al.*, "Communications security in Industry X: A survey," *IEEE open j. Commun. Soc.*, vol. 5, pp. 982–1025, Jan. 2024.

[7] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 2985–2996, May 2021.

[8] N. Tuptuk and H. Stephen, "Security of smart manufacturing systems", *J. Manuf. Syst.*, vol. 47, pp. 93–106, Apr. 2018.

[9] M. Turkanović, B. Brumen, and M. Hölbl, "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.*, vol. 20, pp. 96–112, Sep. 2014.

[10] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment", *Ad Hoc Netw.*, vol. 36, no. 1, pp. 152–176, Jan. 2016.

[11] R. Amin, *et al.*, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.

[12] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, Mar. 2017.

[13] F. Rafique, *et al.*, "An efficient and provably secure certificateless protocol for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8039–8046, Nov. 2022.

[14] F. Yi, *et al.*, "WSNEAP: An Efficient authentication protocol for IIoT-oriented wireless sensor networks", *Sensors*, vol. 22, no. 19, Article no. 7413, pp. 1–21, Sep. 2022.

[15] M. H. Eldefrawy, N. Ferrari, and M. Gidlund, "Dynamic user authentication protocol for industrial IoT without timestamping," in *Proc. WFCS*, Sundsvall, Sweden, May 2019.

[16] B. Harishma, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay, "POSTER: Authenticated key-exchange protocol for heterogeneous CPS," in *Proc. ASIACCS*, Incheon, Republic of Korea, Jun. 2018, pp. 849–851.

[17] K. A.-A. Mutlaq, V. O. Nyangaresi, M. A. Omar, and Z. A. Abduljabbar, "Symmetric key based scheme for verification token generation in Internet of Things communication environment," in *Proc. 2nd EAI Int. Conf. Applied Cryptography in Computer and Communications*, May 2022, pp. 46–64.

[18] Y. Chen, J.-F. Martinez, P. Castillejo, and L. López, "A privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: PriAuth," *Wireless Commun. Mobile Comput.*, vol. 2017, Article no. 5290579, pp. 1–17, Dec. 2017.

[19] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2649–2656, Feb. 2022.

[20] S. Wang, X. Zhou, K. Wen, B. Weng, and P. Zeng, "Security analysis of a user authentication scheme for IoT-based healthcare," *IEEE Internet of Things J.*, vol. 10, no. 7, pp. 6527–6530, Apr. 2023.

[21] R. Praveen and P. Pabitha, "A secure lightweight fuzzy embedder-based user authentication scheme for Internet of Medical Things applications," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 7523–7542, Jan. 2023.

[22] C.-M. Chen, Z. Chen, S. Kumari, and M.-C. Lin, "LAP-IoHT: A lightweight authentication protocol for the Internet of Health Things," *Sensors*, vol. 22, no. 14, pp. 5401, 2022.

[23] C. Pu, H. Zerkle, A. Wall, S. Lim, K. -K. R. Choo, and I. Ahmed, "A lightweight and anonymous authentication and key agreement protocol for wireless body area networks," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21136–21146, Nov. 1, 2022.

[24] S. Hu, Y. Chen, Y. Zheng, B. Xing, Y. Li, L. Zhang, and L. Chen, "Provably secure ECC-based authentication and key agreement scheme for advanced metering infrastructure in the smart grid," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5985–5994, Apr. 2023.

[25] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[26] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. EUROCRYPT*, Amsterdam, The Netherlands, Apr. 2002, pp. 337–351.

[27] K.-H. Chuang, *et al.*, "A physically unclonable function using soft oxide breakdown featuring 0% native BER and 51.8 fJ/bit in 40-nm CMOS," *IEEE J. Solid-State Circuits,* vol. 54, no. 10, pp. 2765–2776, Oct. 2019.

[28] H. Wu and B. Preneel, "AEGIS: A fast authenticated encryption algorithm," in *Proc. SAC*, Burnaby, BC, Canada, Aug. 2013, pp. 185–201.

[29] A. K. Das, *et al.*, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[30] H. A. N. Far, *et al.*, "LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT," *Wireless Netw.*, vol. 27, no. 2, pp. 1389–1412, Jan. 2021.

[31] Z. Yang, J. He, Y. Tian, and J. Zhou, "Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6584–6596, Oct. 2020.

[32] M. Tanveer, *et al.*, "REAP-IIoT: Resource-efficient authentication protocol for the industrial Internet of Things," *IEEE Internet of Things J.*, vol. 9, no. 23, pp. 24453–24465, Dec. 2022.

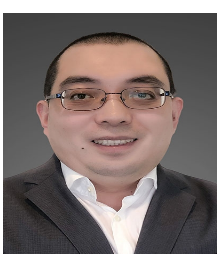
[33] T. Alladi, Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV-UAV communication," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15068–15077, Dec. 2020.

MUHAMMAD HAMMAD completed the Ph.D. degree from the Faculty of Mechanical Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan. He is a distinguished researcher focused on the intersection of smart manufacturing and its security, emphasizing the robustness of Internet of Things (IoT), Industrial Internet of Things (IIoT), and Cyber-Physical Systems. He has contributed to advancing the understanding and implementation of security measures necessary to safeguard next-generation manufacturing technologies. His work delves into developing innovative solutions to protect against cyber-attacks in highly interconnected and automated environments, ensuring the integrity and reliability of modern industrial systems.

AKHTAR BADSHAH received the B.Sc. degree in computer software engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2011, the M.Sc. degree in software engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2017, and the Ph.D. degree in computer engineering from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in 2023. Currently, he is a Lecturer in the Department of Software Engineering at the University of Malakand, Dir Lower, Pakistan. His research interests include information security and privacy, applied cryptography, Internet of Things (IoT) security, blockchain applications and technologies, and digital twins security. He has made notable contributions to his field, with multiple research findings published in renowned journals.

Mohammed A. Almeer is an assistant professor at the Department of Computer Engineering at the University of Bahrain. He holds a Ph.D. in Computer Engineering from the Florida Institute of Technology in Melbourne, USA, which he obtained in 2014. He also received an M.Sc. degree in Network Computing from Monash University in Melbourne, Australia, in 2005. His teaching and research interests encompass computer and network security, wireless sensor networks (WSNs), Internet of Things (IoT), vehicular networking, and network automation.

Muhammad Waqas (Senior Member IEEE) received pursued his PhD degree with the Department of Electronic Engineering at Tsinghua University, Beijing, China, in 2019. From October 2019 to March 2022, he was a Research Associate at the Faculty of Information Technology, Beijing University of Technology, Beijing, China. From April 2022, he has been an Assistant Professor at the Computer Engineering Department, College of Information Technology, University of Bahrain, Bahrain. He is currently a Senior Lecturer at the School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London, UK. He has also been an Adjunct Senior Lecturer at the School of Engineering, Edith Cowan University, Australia, since November 2021. He was also invited as a distinguished speaker at several reputed conferences. His current research interests are in the areas of wireless communications, vehicular networks, cybersecurity and Machine Learning. He is recognised as a Global Talent in the area of Wireless Communications by UK Research and Innovation.



Houbing Herbert Song (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in August 2012. He is currently a Tenured Associate Professor of AI and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab), University of Maryland, Baltimore County, Baltimore, MD, USA. He was a Tenured Associate Professor of Electrical Engineering and Computer Science with Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. His research has been sponsored by federal agencies and industry. He is the author of more than 100 articles and the inventor of 2 patents (US & WO). His research interests include cyber-physical systems/Internet of Things, cybersecurity and privacy, AI/machine learning/big data analytics, edge computing, unmanned aircraft systems, connected vehicle, smart and connected health, and wireless communications and networking. He is a Highly Cited Researcher identified by Clarivate (2021) and a Top 1000 Computer Scientist identified by Research.com.

Sheng Chen (IEEE Life Fellow) received his BEng degree from the East China Petroleum Institute, Dongying, China, in 1982, and his PhD degree from the City University, London, in 1986, both in control engineering. In 2005, he was awarded the higher doctoral degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK. From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with the School of Electronics and Computer Science, the University of Southampton, UK, where he holds the post of Professor in Intelligent Systems and Signal Processing. Dr Chen's research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, evolutionary computation methods and optimization. He has published over 700 research papers. Professor Chen has 19,400+ Web of Science citations with h-index 61 and 37,900+ Google Scholar citations with h-index 82. Dr. Chen is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of Asia-Pacific Artificial Intelligence Association and a Fellow of IET. He is one of the original ISI highly cited researcher in engineering (March 2004).

ZHU HAN (Sâ€™01â€“Mâ€™04-SMâ€™09-Fâ€™14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. Dr. Han's main research targets on the novel game-theory related concepts critical to enabling efficient and distributive use of wireless networks with limited resources. His other research interests include wireless resource allocation and management, wireless communications and networking, quantum computing, data science, smart grid, carbon neutralization, security and privacy. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011,Â the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, IEEE Vehicular Technology Society 2022 Best Land Transportation Paper Award, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015 to 2018 and ACM Distinguished Speaker from 2022 to 2025, AAAS fellow since 2019, and ACM Fellow since 2024. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award (an IEEE Field Award), for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: for contributions to game theory and distributed management of autonomous communication networks.