1-1-2023

A provable secure and efficient authentication framework for smart manufacturing industry

Muhammad Hammad

Akhtar Badshah

Ghulam Abbas

Hisham Alasmary

Muhammad Waqas
Edith Cowan University

See next page for additional authors

10.1109/ACCESS.2023.3290913
Hammad, M., Badshah, A., Abbas, G., Alasmary, H., Waqas, M., & Khan, W. A. (2023). A provable secure and efficient authentication framework for smart manufacturing industry. IEEE Access, 11, 67626-67639. https://doi.org/10.1109/ACCESS.2023.3290913

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 May 2023, accepted 25 June 2023, date of publication 28 June 2023, date of current version 10 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290913

RESEARCH ARTICLE

A Provable Secure and Efficient Authentication Framework for Smart Manufacturing Industry

MUHAMMAD HAMMAD[1], AKHTAR BADSHAH[2], GHULAM ABBAS[3], (Senior Member, IEEE), HISHAM ALASMARY[4], MUHAMMAD WAQAS[5,6], (Senior Member, IEEE), AND WASIM AHMED KHAN[1]

[1]Faculty of Mechanical Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Swabi 23640, Pakistan
[2]Department of Software Engineering, University of Malakand, Dir Lower 18800, Pakistan
[3]Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Swabi 23640, Pakistan
[4]Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[5]Computer Engineering Department, Faculty of Information Technology, University of Bahrain, Zallaq 32038, Bahrain
[6]School of Engineering, Edith Cowan University, Perth, WA 6027, Australia

Corresponding author: Akhtar Badshah (akhtarbadshah@uom.edu.pk)

This work was supported by the Ministry of Education in Saudi Arabia under Project KKU-IFP2-H-15.

ABSTRACT Smart manufacturing is transforming the manufacturing industry by enhancing productivity and quality, driving growth in the global economy. The Internet of Things (IoT) has played a crucial role in realizing Industry 4.0, where machines can communicate and interact in real-time. Despite these advancements, security remains a major challenge in developing and deploying smart manufacturing. As cyber-attacks become more prevalent, researchers are making security a top priority. Although IoT and Industrial IoT (IIoT) are used to establish smart industries, these systems remain vulnerable to various types of attacks. To address these security issues, numerous authentication methods have been proposed. However, many of these methods are vulnerable to known attacks, such as physical security, privileged-insider, and impersonation attacks, or have high computational and communication costs, making them unsuitable for resource-limited IoT devices. Therefore, in this paper, we present a new approach to mutual authentication between the flexible manufacturing system unit, the user, and the server. It enables secure communication in an IIoT-enabled system, which represents the smart manufacturing industry. This security is achieved through the establishment of session keys. Our proposed scheme demonstrates robust resistance against various security attacks, outperforming existing schemes. Although the communication overhead is slightly higher compared to some benchmark schemes, this trade-off is justified by the significant security advantages it offers. Overall, our scheme strikes a balance, providing superior security and competitive performance.

INDEX TERMS Smart manufacturing, security, authentication, Internet of Things, resource-limited IoT devices, mutual authentication.

I. INTRODUCTION

Technological advancements and digital connectivity in manufacturing open the way for a smart industry or industry 4.0. Smart industry combines different advanced technologies and solutions related to computing, manufacturing, connectivity, virtualization, and data handling. These technologies include Internet of Things (IoT), cyber-physical systems (CPS), artificial intelligence (AI) and machine learning (ML), Big data analytics, cloud computing, advanced robotics and automation, additive manufacturing/3D printing, augmented reality (AR) and virtual reality (VR), digital twin, blockchain, edge computing, and cybersecurity as shown in Fig. 1. Smart industry, through the integration of advanced technologies, plays a vital role in smart manufacturing by enhancing productivity, reducing operational costs, improving quality, and promoting sustainable and efficient production processes [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Ayoub Khan.

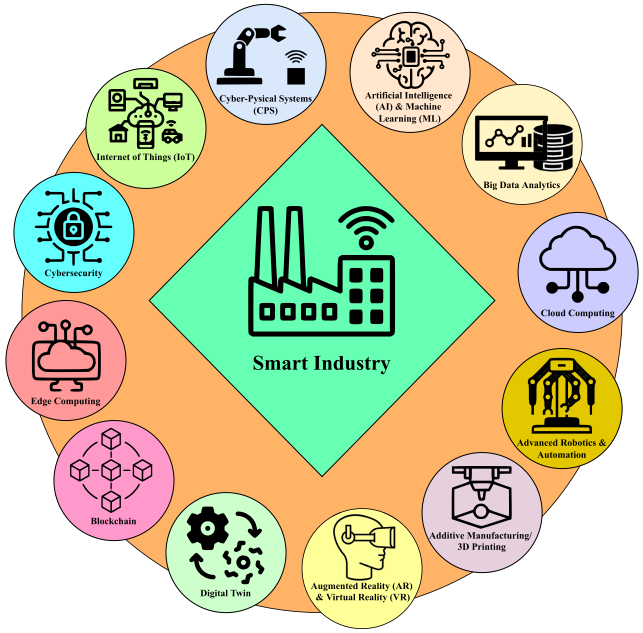


FIGURE 1. Components of smart industry.

Flexible manufacturing system (FMS) considered as the first generation technology that set the foundation of smart industry by introducing flexibility, integration, automation and connectivity into the production process. FMS paved the path for advanced technologies such as AI, IoT, cybersecurity and cloud computing to further increase the efficiency and performance of modern manufacturing systems [2].

An FMS is computer controlled manufacturing system designed to adjust the variations in the volume and type of the product being manufactured. FMS has the ability to produce variety of parts with minimum human intervention. The components of FMS include workstations, material handling system, computer control system, tool management system, automated storage and retrieval system (AS/RS), and sensors and data collection devices. Workstations are equipped with computer numerical control (CNC) machines, robots, and other automated equipment. Based on the number of CNC machine workstations, FMS is further classified as flexible manufacturing cell (FMC) and single machine cell (SMC). SMC has one CNC machine workstation, whereas FMC comprises up to three CNC machines workstations. FMS contains more than three CNC machines workstations. Material handling system is responsible for transporting workpieces, tools and parts within the FMS. Robots, Automated Guided vehicles (AGVs), conveyors and other automated devices are used for this purpose. Computer control system contains the central computer that manages, control and coordinates all the operations within the FMS. It manages the routing, scheduling, and coordination of workpieces and materials within the system. It controls the directs operations at the workstations, material handling system and AS/RS. Tool management system manages the tools used at the required workstation. It is responsible for monitoring and maintaining

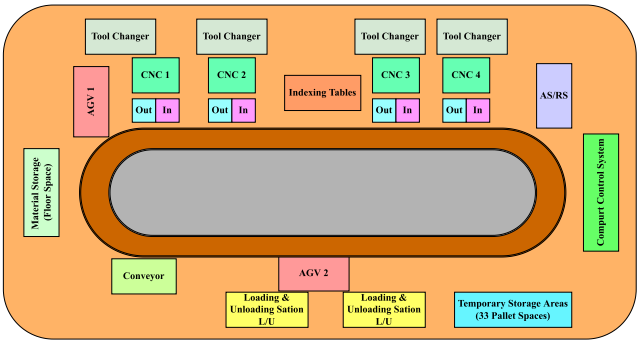


FIGURE 2. Generic model of FMS.

the use and inventory of tools and ensures the availability of right tool when needed. An AS/RS is responsible for providing the automated storage and retrieval of workpieces, tools, and other parts. It may contain shelves, racks, or other automated devices such as shuttles or robots for retrieving materials when required. Sensors and data collection devices are applicable to collect data on different parameters such as machine status, production rates, quality measurements, and other related information. This data is used to monitor, control, and optimize the manufacturing processes [3], [4].

The components of FMS interact with each other through central computer, which controls and coordinates all the operations. The central computer receives data from sensors and data collection devices, monitors the status of workstations and machines, and make decisions on routing, scheduling and other operations to optimize the manufacturing. Based on the instructions from the central computer, the material handling system moves work pieces, tools, and materials among different workstations. Based on the production schedule and other instructions from the central computer, workstations perform the manufacturing operations on the work pieces using the tools provided by the tool management system. Based on the requirements of the manufacturing processes, the AS/RS system provides automated storage and retrieval of materials as needed [5]. The generic model of FMS is shown in Fig. 2.

While FMS offer numerous advantages, they also possess certain drawbacks. The limitations of FMS are as follow [6]:

- High initial cost: Implementation of FMS involves a significant initial investment in terms of purchasing and installing the automated machines, material handling system and the control system.
- Complex integration: An FMS is highly complex system that demands skilled personnel to design, operate, maintain, and software integration.
- Limited flexibility: While FMS is designed to provide maximum possible flexibility, but its flexibility is not unlimited. It can handle variations in product design, process, and production volume, but major changes may require rapid retooling or reprogramming.
- Technological obsolescence: Automation technologies used in FMS are continuously evolving and new

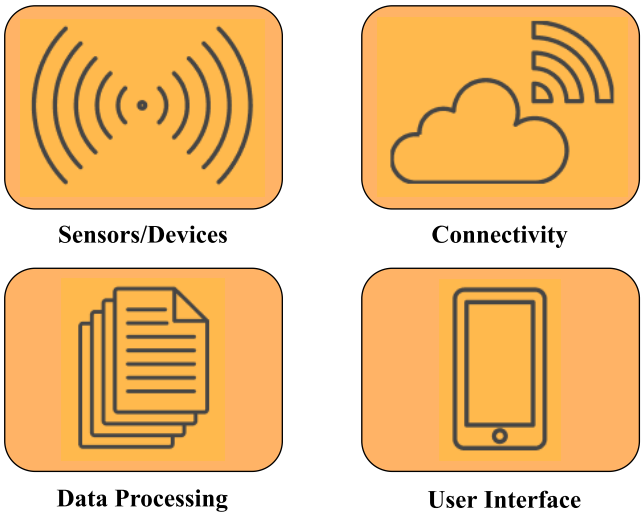


FIGURE 3. Generic model of IIoT system.

advancements are constantly emerging. It can lead to technological obsolescence, where the components of FMS may unsupported or outdated over time. Moreover, if a single machine or the control system in FMS fails, it can render the entire production process. This requires to have robust maintenance program and reliable equipment.

- Long setup time: While FMS aims to minimize the setup times, the initial setup of the system can be lengthy, specifically during programming and configuration of the machines.

FMS can be converted to Industrial IoT (IIoT)-enabled system by assessment and connectivity to make the system more efficient, enabling the smart manufacturing industry [7]. IIoT considered as the second generation of smart industry, building on the foundation laid by FMS to create even more responsive and efficient manufacturing processes. IIoT is the subset of IoT specifically implemented in industrial areas. IIoT refers to the integration of physical assets, machines, and industrial systems with network-connected sensors, data storage, and analytics platform to gather, analyze, and act upon data in real-time. IIoT-enabled systems allow for the collection analysis of significant data in real-time from various sources across the entire industrial ecosystem. This results in better optimization of processes, predictive maintenance, energy efficiency, and decision making [8], [9].

IIoT forms an essential part of smart industry, where the integration of digital and physical worlds creates more intelligent and interconnected manufacturing systems. IIoT-enabled system is better than FMS due to following reasons: remote access, real-time monitoring and control, interconnectivity, scalability and flexibility, cost and energy efficiency, and predictive maintenance [2].

The generic model of IIoT is shown in Fig. 3. It contains the four parts, the description of each part is as follows: Sensors/Devices are the physical devices that help to collect data from the environment. In the connectivity part, the collected data is sent to data processing system. This is accomplished through different connectivity technologies such as Wi-Fi, Bluetooth, cellular network, or wired connections. After reaching the data to its required destination, it needs to be processed in data processing section. This is achieved on local edge computing devices or in the cloud. The processing may involve the inspection of data for anomalies, running predictive algorithms, or aggregating the data for further analysis. After processing the data, the results need to be presented to the end user in the user interface section. This may involve the dashboard displaying key metrics, visualizations or automated reports in a digestible format [10].

Let's consider a use case scenario of remote monitoring of a manufacturing industry's production line containing FMS through IoT. The sensors equipped in the production line CNC machines collect all the required data like temperature, pressure, speed, vibration, acoustic noise, light intensity, product dimensions, weight, flow rate, machine vision inspection, energy consumption, wear and tear, and operational time. Then the data is sent in real-time over a Wi-Fi network to a cloud-based data processing system. After that, the data is processed in real- time to check for anomalies that may indicate a problem. For example, a sudden increase in machine temperature or vibration indicate excessive wear and tear. In case of any potential issues, the manager can remotely shut down the machine to prevent damage or schedule maintenance to sort out the issue.

The design requirements for the IIoT system to mitigate the threats and ensure functionality and efficiency are as follow:

- **Mutual authentication:** In an IIoT environment, entities involved in communication must mutually authenticate each other during the authenticated key agreement (AKA) procedure to verify their legitimacy and the integrity of exchanged messages. Failure to ensure mutual authentication can compromise the security and trustworthiness of the system.
- **Confidentiality:** Maintaining confidentiality of the session key generated through AKA procedure is important to limit access only to communicating entities and preserve the confidentiality of communication.
- **Untraceability:** Ensuring untraceability of the AKA message from potential adversaries' perspective is crucial for maintaining the security and privacy of the system.
- **Anonymity:** Protecting the real identities of communicating entities is essential to maintaining anonymity and preserving the confidentiality and privacy of communication.
- **Resistance to potential security attacks:** Ensuring that the security scheme is resistant to common attacks such as man-in-the-middle (MitM) attacks, replay attacks, ephemeral secret leakage (ESL) attacks, impersonation attacks, and physical attacks. It is necessary to maintain the security and trustworthiness of the system.

- **Minimizing overheads in AKA procedure:** Designing the AKA procedure to minimize communication and computational overheads is crucial in improving the system's performance and scalability, reducing the risk of security vulnerabilities, and ensuring that the system can handle larger amounts of data and tasks without compromising its efficiency.

To ensure the security, privacy, and efficiency of IIoT systems, it is crucial to adhere to the design requirements outlined above. In response, several studies have proposed AKA schemes that meet these requirements while accounting for the unique challenges and specifications of the IIoT environment. In the following section, we provide a review and analysis of some of these proposed AKA schemes, examining their efficacy in fulfilling the design requirements for IIoT systems.

A. RELATED WORK

In recent years, there has been a surge in the development of authentication protocols tailored specifically for IoT environments, with the primary goal of improving both privacy and security. These schemes are designed to ensure secure communication and thwart known attacks. To achieve this, they employ a range of verification methods, including smart cards, passwords, and biometrics, to authenticate user legitimacy. Wireless sensor networks are crucial components in the IIoT and play a pivotal role in smart manufacturing systems. As such, they require robust security mechanisms to protect against malicious actors seeking to disrupt operations, steal sensitive data, or cause other forms of damage.

Choudhary et al. proposed a key exchange model and a lightweight mutual authentication protocol for IIoT users in [11]. Their proposed approach offers several security features, including data integrity, confidentiality, and anonymity, while also safeguarding against common attacks like modification, replay, and man-in-the-middle. However, despite these advantages, the protocol may still be vulnerable to internal security risks within IIoT networks.

Fang et al. [12] thoroughly examines various IIoT control systems and their associated challenges, as well as the security vulnerabilities in detail. The diversity and complexity of the protocols make it difficult for existing efforts to implement a uniform security mechanism for IIoT control systems. However, Fang et al. does not provide a comprehensive framework or secure authentication protocol to address these security threats.

Rafique et al. [13] addressed a crucial challenge in the IIoT: ensuring secure data transmission. To tackle this challenge, they proposed a multifactor AKA scheme that balanced robust security with consideration for resource-constrained environments. The key contribution of their proposed scheme is the use of symmetric cryptography, bitwise XOR operations, and hash functions to create a secure system that was well-suited for resource-constrained environments, enabling legitimate users to access sensing devices remotely without sacrificing security. However, their proposed scheme is vulnerable to smart card/device loss attacks and does not offer user anonymity and untraceability features.

Eldefrawy et al. [14] also put forth a user authentication scheme for IIoT systems. Although their proposed scheme was highly efficient in terms of computational and communication requirements but lacked mutual authentication between users and sensor nodes/smart devices within the system.

Harishma et al. [15] devised a solution aimed at ensuring secure data transmission within heterogeneous CPS. Despite the potential of their proposed scheme, it was found to have vulnerabilities to ESL attacks when operating under the CK-adversary model. Additionally, the scheme does not provide the ability to dynamically add new IoT smart devices, potentially limiting its practicality in real-world applications.

Chen et al. [16] proposed a user AKA scheme for IoT environments. While the scheme demonstrated efficiency in terms of communication and computation requirements, it was discovered to have vulnerabilities to privileged insider attacks and was lacking in untraceability.

In a nutshell, user authentication in IIoT and related environments has become a crucial aspect for ensuring security. Despite the abundance of existing schemes, many are lacking in terms of modern security standards or are simply not feasible for practical use. In this article, we present a groundbreaking solution–a user authentication scheme for smart manufacturing environments that not only eliminates known security vulnerabilities but also offers unparalleled levels of anonymity and untraceability for user, server, and FMS unit. This means that even the most advanced attackers will be unable to compromise the system through various attack strategies or gain insight into network traffic from legitimate users through statistical cryptanalysis. Moreover, our solution provides robust physical security for deployed FMS units and mobile devices, while maintaining comparable computational and communication costs, making it an ideal choice for real-time IIoT applications.

B. RESEARCH CONTRIBUTIONS

This paper presents the following key insights:

- Firstly, to enhance the security of smart manufacturing industry, we propose a scheme that enables mutual authentication and key agreement. The scheme involves the server facilitating mutual authentication between the FMS unit and user, as well as between the user and server. This process establishes session keys between the user and FMS unit, as well as between the user and server.
- Secondly, the proposed scheme uses cryptographic hash functions, elliptic curve cryptography (ECC), and bitwise XOR operations in combination with physical unclonable function (PUF) to protect against physical tampering attacks and address resource constraints in IIoT environments.

TABLE 1. List of notations.

Notation	Description
UR_j, MD_j	jth user and jth mobile device of UR_j
$ID_{UR_j}, PIDS_{UR_j}$	jth user unique identity and pseudo-identity
PSW_{UR_j}	Password of UR_j
MAC	Message authentication code
T_i	ith timestamp
ΔT	Message time delay limit
rn_i	random number
$\|, \oplus$	Concatenation and XOR, respectively
$\mathcal{A}$	Adversary
SK	Session key
S, K	Server and master secret key of S.
(R, C)	(challenge, response) pair
$PUF(\cdot)$	Physical unclonable function
$h(\cdot)$	Cryptographic hash function
$E_q(a, b), P$	Elliptic curve, and its base point
(PB, PR)	Public/private key pair

- Thirdly, the proposed scheme has been formally proven secure under the extended random oracle model (ROR) and demonstrated resistance against various attacks in informal security analysis.
- Finally, after conducting an exhaustive comparison with existing state-of-the-art user authentication schemes, we have determined that our proposed scheme exhibits superior performance.

C. PAPER OUTLINE

The structure of this article is as follows: Section II presents background information crucial for understanding the proposed scheme. Section III outlines the steps involved in the proposed scheme. Section IV evaluates the security of the proposed scheme. Section V compares the proposed scheme with other existing schemes. Lastly, Section VI summarizes the article.

II. BASIC PRELIMINARIES

This section provides the essential background information required to understand the proposed scheme. In addition, Table 1 presents a list of notations used throughout the paper.

A. NETWORK AND THREAT MODELS

1) NETWORK MODEL

In the proposed system model, three crucial components contribute to setting up the IIoT environment, as depicted in Fig. 4. These components are:

1) Users/Mobile terminals: Users can easily access the FMS via Wifi and perform remote operations related to FMS within a specific range.
2) FMS unit: Smart cards/Smart Wifi chips/Sensing devices are embedded in the FMS to allow for easy access through the server.
3) Server: The server serves as a trusted authority in the system and securely registers both users and FMS units offline.

To access the FMS, the user must first be registered with the server. Communication between the FMS unit and the user's mobile terminal occurs via Wi-Fi, which enables real-time access to data. This is particularly important in manufacturing settings where delays can result in production losses. Communication between the mobile terminal and the server takes place over the Internet, which enhances the system's scalability and accessibility. By placing the user in the middle of the architecture, the communication between the user and the FMS unit is secured, preventing unauthorized access to the data. The communication architecture is designed to ensure that communication between the different components is secure and reliable. The server facilitates mutual authentication between the FMS unit and the mobile terminal, as well as between the mobile terminal and the server during the authentication process. Upon successful mutual authentication and key establishment, the user can access data captured by the sensing device/FMS to monitor and regulate the manufacturing process.

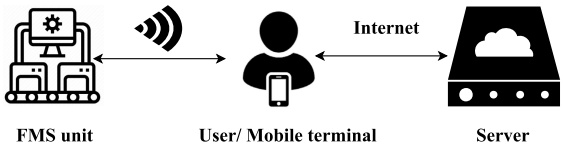


FIGURE 4. IIoT system for remote manufacturing control.

As a practical example, the proposed architecture could be used in a manufacturing setting to monitor and regulate the production process remotely. The use of Wi-Fi for communication between the FMS unit and the user's mobile terminal allows for easy access to the data captured by the sensing device, while the use of the Internet for communication between the mobile terminal and the server provides the ability to control and monitor the manufacturing process from anywhere with Internet access. This demonstrates the rationality of the communication architecture in enabling secure and efficient communication between the different components of the IIoT environment.

2) THREAT MODEL

Our authentication scheme employs the well-established "Dolev-Yao (DY) threat model" [17], which assumes the adversary $\mathcal{A}$ has full control over the communication channels and can perform actions, such as eavesdropping, altering, deleting, or inserting false messages. The end-point entities (FMS units and users) are not considered fully trustworthy.

The "CK-adversary model" [18] is a common way to model key exchange protocols. Within this particular model, the adversary $\mathcal{A}$ has the capability to transmit messages, akin to the DY model, while also having the ability to infiltrate additional information, including session state, private keys, and session keys. Ensuring security requires the AKA protocol to restrict the impact of any leaked sensitive information, like session ephemeral secrets or session keys, on the confidentiality of other credentials used in the communication.

We assume that adversary $\mathcal{A}$ can physically capture some FMS units, extract sensitive information from their memory, and obtain credentials from a lost or stolen user mobile terminal through power analysis attacks [19]. To enhance security, the server is secured with a locking system, making it harder to capture physically than FMS units. The server is considered a trusted entity in the IIoT environment.

In our study, we utilize the assumptions made in Amin et al.'s [20] scheme. These include the usage of dictionary words as passwords and identities by legitimate users in password-based authentication, the ability of adversary $\mathcal{A}$ to individually guess a user's password and identity, but with the verification of both being computationally expensive if proper procedures are in place, and the computational difficulty in polynomial time of guessing high entropy secret keys and nonces.

B. CRYPTOGRAPHICAL BUILDING BLOCKS

1) PHYSICALLY UNCLONABLE FUNCTION

A PUF utilizes the physical characteristics of a device to produce a unique response for authentication and encryption purposes. When presented with a challenge C, the PUF generates a corresponding response R ($R = PUF(C)$) that is specific to the device and can be employed as a secure identifier or key.

Cryptographically, PUFs provide a secure and unique method for generating keys and identifying devices. However, PUF responses may vary slightly due to environmental noise, making them vulnerable to losing sensitive information during critical operations. Recent research has focused on developing stable, noise-resistant PUF designs to maintain a low error rate under harsh conditions [21]. For the purpose of this article, it is assumed that the FMS units and mobile terminals have an ideal, noise-resistant PUF.

2) ELLIPTIC CURVE CRYPTOGRAPHY

ECC employs the mathematical principles of elliptic curves to generate secure keys for encryption and decryption. These curves are defined as a set of points (x, y) that satisfy the equation $y^2 = x^3 + ax + b$, where a and b are constants. In ECC, a private key is represented by a secret scalar value, while a public key is represented by a point on the elliptic curve. The mathematical computational problems of ECC are as follows.

- The elliptic curve discrete logarithm (ECDL) problem is a mathematical challenge that requires finding the private key (scalar value) that corresponds to a known public key on the elliptic curve. The problem is equivalent to finding the integer n that satisfies $P = n \cdot G$, where P is the public key, G is a pre-defined point on the elliptic curve known as the generator point, and n is the private key. ECDL is considered a difficult problem and provides the foundation for the security of ECC.
- Elliptic curve computational Diffie-Hellman (ECCDH) problem is a type of public key encryption method where two parties agree on a shared point on an elliptic curve, and then use it to generate a shared secret that can be used as a key for symmetric encryption. The shared secret is computed as: $S = (n_1 * n_2)G$, where n_1 and n_2 are the private keys of the two parties, and G is the shared point on the elliptic curve.
- Elliptic curve decisional Diffie-Hellman (ECDDH) problem is a type of encryption that involves guessing the shared secret produced by the ECCDH algorithm. The ECDDH problem is equivalent to the problem of given two public keys, $P1 = n_1 \cdot G$ and $P2 = n_2 \cdot G$, and a shared point G, finding the value of the shared secret $S = (n_1 * n_2)G$. Solving the ECDDH problem is considered hard, and provides the basis for the security of the ECCDH encryption scheme.

III. THE PROPOSED SCHEME

Our proposed security solution for smart manufacturing environments aims to restrict access to deployed FMS units to only authorized users. The solution is based on the robust combination of the secure hash algorithm (SHA-256), ECC, and PUF primitives. Time synchronization among all entities in the environment is a crucial requirement for the effective implementation of the proposed scheme. The scheme is divided into five distinct phases, each of which will be thoroughly explained in the following sections.

A. SYSTEM INITIALIZATION PHASE

The trusted authority/server S selects a "non-singular elliptic curve represented as $y^2 = x^3 + ax + b$, (mod q) over the finite field Z_q, where q is a large prime number and the curve satisfies the equation $4a^3 - 27b^2 \neq 0$ (mod q), with the point $\mathcal{O}$ as its infinity or zero point". S then chooses an elliptic curve generation point $P \in E_q(a, b)$ of order n, such that $n \cdot P = \mathcal{O}$. S generates a public-private key pair $\{PR_S, PB_S\}$, where PR_S is an element of Z_q^* and $PB_S = PR_S \cdot P$. S also selects a master key K for itself. The parameters $\{PR_S, PB_S, K, E_q(a, b), P\}$ are stored securely in S's tamper-proof memory, and the parameters $\{PB_S, E_q(a, b), P\}$ are made public in the smart manufacturing environment.

B. FMS UNIT DEPLOYMENT PHASE

During the FMS Unit Deployment Phase (FDP), S registers an FMS_i prior to deploying it in the environment. S carries out these critical steps to successfully deploy the FMS_i in a smart manufacturing setting.

Step FDP1: S selects a challenge parameter C_{FMS_i} and a unique identity ID_{FMS_i}. Next, S computes the pseudo-identity PID_{FMS_i} as $PID_{FMS_i} = h(ID_{FMS_i} \| K)$, where K is the master secret key of S. S only stores and retains the ID_{FMS_i} in its memory and forwards C_{FMS_i}, ID_{FMS_i}, and PID_{FMS_i} through a secure channel to FMS_i.

Step FDP2: Upon receiving the C_{FMS_i}, ID_{FMS_i}, and PID_{FMS_i} parameters, FMS_i computes R_{FMS_i} as $R_{FMS_i} = PUF(C_{FMS_i})$ and X_{FMS_i} as $X_{FMS_i} = PID_{FMS_i} \oplus h(R_{FMS_i} \| ID_{FMS_i})$. FMS_i stores C_{FMS_i}, ID_{FMS_i}, and X_{FMS_i} in its memory.

C. USER REGISTRATION PHASE

The user registration phase (URP) requires the user (UR_j) to register their mobile device (MD_j) with the S in order to access real-time information and communicate with a designated FMS_i. The S will provide secret credentials and a list of approved FMS_i for UR_j to obtain the information and communicate further. The following procedure is followed by the S to complete the URP phase.

Step URP1: UR_j picks an identity ID_{UR_j} and forwards the registration request along with the parameter to the S through a secure private channel.

Step URP2: After obtaining the parameter ID_{UR_j}, S computes the pseudo-identity PID_{UR_j} as $PID_{UR_j} = h(ID_{UR_j} \| K)$, where K is the master secret key of S. Next, S picks challenge parameter C_{MD_j}. The S stores the ID_{UR_j} in its database, and transmits $\{C_{MD_j}, PID_{UR_j}\}$ to UR_j via a secure channel.

Step URP3: After receiving C_{MD_j} and PID_{UR_j}, UR_j selects a password PSW_{UR_j} and then computes $A_{MD_j} = PID_{UR_j} \oplus h(ID_{UR_j} \| PSW_{UR_j})$, $R_{MD_j} = PUF(C_{MD_j})$, and $MAC_{MD_j} = h(PID_{UR_j} \| R_{MD_j})$. Next, store the credentials $\{C_{MD_j}, A_{MD_j}, MAC_{MD_j}, PUF(\cdot)\}$ in its own memory.

D. USER LOGIN PHASE

Step UL1: User UR_j enter his/her identity ID_{UR_j} and password $PSW^l_{UR_j}$ into MD_j.

Step UL2: MD_j retrieves the parameters C_{MD_j} and A_{MD_j} and then calculates $R_{MD_j} = PUF(C_{MD_j})$, $PID_{UR_j} = A_{MD_j} \oplus h(ID_{UR_j} \| PSW^l_{UR_j})$, and $MAC^l_{MD_j} = h(PID_{UR_j} \| R_{MD_j})$. Next, MD_j checks $MAC^l_{MD_j} \stackrel{?}{=} MAC_{MD_j}$, if it holds, UR_j successfully login.

E. AUTHENTICATION AND KEY AGREEMENT PHASE

In the AKA phase, mutual authentication is performed between entities and two session keys are established for secure communication. One key is used to encrypt messages between the FMS unit and mobile device, and the other for messages between mobile device and server. The two keys provide an extra layer of security, preventing unauthorized access or tampering even if one key is compromised. The following is a description of the intricate steps involved in the authentication phase.

AKA1: Once UR_j has completed the login process, MD_j generates a random number rn_1 and the current timestamp T_1. It then creates a message MSG_1 consisting of the parameters $\{rn_1, T_1\}$ and sends it to MD_j via an open channel.

AKA2: Upon receiving the message MSG_1 at time T^*_1, FMS_i checks the timeliness of T_1 by verifying if $|T^*_1 - T_1| \stackrel{?}{\leq} \Delta T$, where ΔT represents the maximum transmission delay. If the condition is met, FMS_i retrieves its own values of C_{FMS_i}, ID_{FMS_i}, and X_{FMS_i} and selects random numbers $rn_2 \in Z^*_q$ and rn_3, and the current timestamp T_2. FMS_i then computes $R_{FMS_i} = PUF(C_{FMS_i})$, $PID_{FMS_i} = X_{FMS_i} \oplus h(R_{FMS_i} \| ID_{FMS_i})$, $SHS_{FS} = rn_2 \cdot PB_S$, $RS_{FMS_i} = rn_2 \cdot P$, $M_1 = (ID_{FMS_i} \| rn_3) \oplus h(SHS_{FS} \| T_2)$, and $M_2 = h(PID_{FMS_i} \| rn_1 \| rn_3 \| SHS_{FS} \| RS_{FMS_i} \| T_2)$, where RS_{FMS_i} is a random public key and SHS_{FS} is a shared secret. Then, FMS_i generates a new message $MSG_2 = \{M_1, M_2, RS_{FMS_i}, T2\}$ and sends it to UR_j through the public channel.

AKA3: After receiving the message MSG_2 at time T^*_2, UR_j checks the timeliness of T_2 by verifying if $|T^*_2 - T_2| \stackrel{?}{\leq} \Delta T$. If the condition is met, UR_j picks a random number $rn_4 \in Z^*_q$, and current timestamp T_3. UR_j then computes $SHS_{US} = rn_4 \cdot PB_S$, $RS_{UR_j} = rn_4 \cdot P$, $M_3 = (ID_{UR_j} \| rn_1) \oplus h(SHS_{US} \| T_3)$, and $M_4 = h(PID_{UR_j} \| rn_1 \| SHS_{US} \| RS_{UR_j} \| T_3)$, where RS_{UR_j} is a random public key and SHS_{US} is a shared secret. Then, UR_j constructs a new message $MSG_3 = \{MSG_2, M_3, M_4, RS_{UR_j}, T_3\}$ and sends it to S through the public channel.

AKA4: S first check the freshness of the received message by verifying the condition $|T^*_3 - T_3| \stackrel{?}{\leq} \Delta T$. S then computes $SHS_{SU} = PR_S \cdot RS_{UR_j}$, $(ID_{UR_j} \| rn_1) = M_3 \oplus h(SHS_{SU} \| T_3)$, and $PID_{UR_j} = h(ID_{UR_j} \| K)$, where PR_S is the private key and K the master secret key of the S. Next, S checks ID_{UR_j} in the revocation list, and if it is found, abort the process, else compute $M^*_4 = h(PID_{UR_j} \| rn_1 \| SHS_{SU} \| RS_{UR_j} \| T_3)$ and then check the condition $M^*_4 \stackrel{?}{=} M_4$. If it fails, abort.

AKA5: S computes $SHS_{SF} = PR_S \cdot RS_{FMS_i}$, $(ID_{FMS_i} \| rn_3) = M_1 \oplus h(SHS_{SF} \| T_2)$, $PID_{FMS_i} = h(ID_{FMS_i} \| K)$, and $M^*_2 = h(PID_{FMS_i} \| rn_1 \| rn_3 \| SHS_{SF} \| RS_{FMS_i} \| T_2)$. S then verifies the condition $M^*_2 \stackrel{?}{=} M_2$. If it fails, abort.

AKA6: S generates the current timestamp T_4, and then computes $SK_{SU} = h(PID_{UR_j} \| SHS_{SU} \| rn_1 \| T_3 \| T_4)$, $SK_{UF} = h(PID_{FMS_i} \| SHS_{SF} \| rn_1 \| rn_3 \| T_2 \| T_3 \| T_4)$, $X_3 = SK_{SU} \oplus SK_{UF}$, and $M_5 = h(X_3 \| SK_{SU} \| T_4)$, where SK_{SU} is the session key between UR_j and S and SK_{UF} is the session key between FMS_i and UR_j. Then, S constructs a message $MSG_4 = \{X_3, M_5, T_4\}$ and sends it to UR_j through the public channel.

AKA7: After receiving the message MSG_4, UR_j verifies the freshness of the received message. UR_j then computes $SK_{US} = h(PID_{UR_j} \| SHS_{US} \| rn_1 \| T_3 \| T_4)$, $SK_{UF} = X_3 \oplus SK_{US}$, and $M^*_5 = h(X_3 \| SK_{US} \| T_4)$, and verifies the condition $M^*_5 \stackrel{?}{=} M_5$, if it holds, store the session keys. UR_j then generates the current timestamp T_5 and computes $M_6 = h(SK_{UF} \| T_5)$. Then UR_j constructs a message $MSG_5 = \{M_6, T_4, T_5\}$ and sends it to FMS_i through the public channel.

AKA8: Upon receiving the message MSG_5, FMS_i verifies the freshness of the received message. FMS_i then computes $SK_{FU} = h(PID_{FMS_i} \| SHS_{FS} \| rn_1 \| rn_3 \| T_2 \| T_3 \| T_4)$ and $M^*_6 = h(SK_{FU} \| T_5)$ and checks the condition $M^*_6 \stackrel{?}{=} M_6$. If it holds, store SK_{FU} as a session key.

Finally, the overall authentication and key agreement procedure is briefed in Fig. 5.

FMS unit FMS_i	**User** UR_j **/ Mobile device** MD_j	**Server** S
	UL1: UR_j enters ID_{UR_j}, $PSW^l_{UR_j}$; **UL2:** Compute: $R_{MD_j} = PUF(C_{MD_j})$, $PID_{UR_j} = A_{MD_j} \oplus h(ID_{UR_j} \| PSW^l_{UR_j})$, $MAC^l_{MD_j} = h(PID_{UR_j} \| R_{MD_j})$; Check $MAC^l_{MD_j} \stackrel{?}{=} MAC_{MD_j}$ if holds, UR_j successfully login.	
	AKA1: Generate: rn_1, T_1; $\xleftarrow{MSG_1=\{rn_1,\ T_1\}}$ (Via public channel)	
AKA2: Verify $\lvert T_1^* - T_1 \rvert \stackrel{?}{\leq} \Delta T$; Pick: rn_2, rn_3, T_2; Retrieve: C_{FMS_i}, ID_{FMS_i}, X_{FMS_i}; Compute: $R_{FMS_i} = PUF(C_{FMS_i})$, $PID_{FMS_i} = X_{FMS_i} \oplus h(R_{FMS_i} \| ID_{FMS_i})$, $SHS_{FS} = rn_2 \cdot PB_S$, $RS_{FMS_i} = rn_2 \cdot P$, $M_1 = (ID_{FMS_i} \| rn_3) \oplus h(SHS_{FS} \| T_2)$, $M_2 = h(PID_{FMS_i} \| rn_1 \| rn_3 \| SHS_{FS} \| RS_{FMS_i} \| T_2)$; $\xrightarrow{MSG_2=\{M_1,\ M_2,\ RS_{FMS_i},\ T2\}}$ (Via public channel)	**AKA3:** Verify $\lvert T_2^* - T_2 \rvert \stackrel{?}{\leq} \Delta T$; Pick: rn_4, T_3; Compute: $SHS_{US} = rn_4 \cdot PB_S$, $RS_{UR_j} = rn_4 \cdot P$, $M_3 = (ID_{UR_j} \| rn_1) \oplus h(SHS_{US} \| T_3)$, $M_4 = h(PID_{UR_j} \| rn_1 \| SHS_{US} \| RS_{UR_j} \| T_3)$; $\xrightarrow{MSG_3=\{MSG_2,\ M_3,\ M_4,\ RS_{UR_j},\ T_3\}}$ (Via public channel)	**AKA4:** Verify $\lvert T_3^* - T_3 \rvert \stackrel{?}{\leq} \Delta T$; Compute: $SHS_{SU} = PR_S \cdot RS_{UR_j}$, $(ID_{UR_j} \| rn_1) = M_3 \oplus h(SHS_{SU} \| T_3)$, $PID_{UR_j} = h(ID_{UR_j} \| K)$; Check ID_{UR_j} in the revocation list, and if available, abort. Compute: $M_4^* = h(PID_{UR_j} \| rn_1 \| SHS_{SU} \| RS_{UR_j} \| T_3)$; Check $M_4^* \stackrel{?}{=} M_4$ If not, abort. **AKA5:** Compute: $SHS_{SF} = PR_S \cdot RS_{FMS_i}$, $(ID_{FMS_i} \| rn_3) = M_1 \oplus h(SHS_{SF} \| T_2)$, $PID_{FMS_i} = h(ID_{FMS_i} \| K)$, $M_2^* = h(PID_{FMS_i} \| rn_1 \| rn_3 \| SHS_{SF} \| RS_{FMS_i} \| T_2)$; Check $M_2^* \stackrel{?}{=} M_2$ If not, abort. **AKA6:** Generate: T_4; Compute: $SK_{SU} = h(PID_{UR_j} \| SHS_{SU} \| rn_1 \| T_3 \| T_4)$, $SK_{UF} = h(PID_{FMS_i} \| SHS_{SF} \| rn_1 \| rn_3 \| T_2 \| T_3 \| T_4)$, $X_3 = SK_{SU} \oplus SK_{UF}$, $M_5 = h(X_3 \| SK_{SU} \| T_4)$; $\xleftarrow{MSG_4=\{X_3,\ M_5,\ T_4\}}$ (Via public channel)
AKA8: Verify $\lvert T_5^* - T_5 \rvert \stackrel{?}{\leq} \Delta T$; Compute: $SK_{FU} = h(PID_{FMS_i} \| SHS_{FS} \| rn_1 \| rn_3 \| T_2 \| T_3 \| T_4)$; $M_6^* = h(SK_{FU} \| T_5)$. Check $M_6^* \stackrel{?}{=} M_6$ If not, abort. Store SK_{FU} as a session key.	**AKA7:** Verify $\lvert T_4^* - T_4 \rvert \stackrel{?}{\leq} \Delta T$; Compute: $SK_{US} = h(PID_{UR_j} \| SHS_{US} \| rn_1 \| T_3 \| T_4)$, $SK_{UF} = X_3 \oplus SK_{US}$, $M_5^* = h(X_3 \| SK_{US} \| T_4)$; Check $M_5^* \stackrel{?}{=} M_5$ If not, abort. Store the session keys; Generate: T_5; Compute: $M_6 = h(SK_{UF} \| T_5)$; $\xleftarrow{MSG_5=\{M_6,\ T_4,\ T_5\}}$ (Via public channel)	

FIGURE 5. The login, authentication, and session key agreement procedures in the proposed scheme.

F. PASSWORD RESET PHASE

The following steps should be taken when it becomes necessary for UR_j to update their password.

Step 1: To initiate the password update process, ID_{UR_j} and the current password, $PSW^o_{UR_j}$, must be entered into MD_j.

Step 2: The next step involves MD_j retrieving the stored parameters $\{C_{MD_j},\ A_{MD_j},\ MAC_{MD_j}\}$ from its memory. MD_j then computes $R_{MD_j} = PUF(C_{MD_j})$, $PID_{UR_j} = A_{MD_j} \oplus h(ID_{UR_j} \| PSW^o_{UR_j})$, and $MAC^l_{MD_j} = h(PID_{UR_j} \| R_{MD_j})$. Next, MD_j checks $MAC^l_{MD_j} \stackrel{?}{=} MAC_{MD_j}$. If it fails, the password update process is aborted. However, if the check is successful, MD_j prompts UR_j to enter a new password $PSW^n_{UR_j}$ and updates the parameter $A^n_{MD_j}$ as $A^n_{MD_j} = PID_{UR_j} \oplus h(ID_{UR_j} \| PSW^o_{UR_j})$.

Step 3: After successfully updating the user's password, MD_j updates the list of credentials in its memory as $\{C_{MD_j},\ A^n_{MD_j},\ MAC_{MD_j},\ PUF(\cdot)\}$.

G. REVOCATION

If a legitimate UR_j loses its mobile device MD_j, the server S can issue and register a new mobile device, MD_j^{new}, to UR_j. To initiate this process, UR_j is required to provide their old identity, ID_{UR_j}. S then takes the following steps to register the new device

Step 1: UR_j picks its previous identity ID_{UR_j} and sends it to S. S computes UR_j's pseudo-identity, PID_{UR_j}, and computed as $PID_{UR_j}! = !h(ID_{UR_j}! \parallel !K)$, where K represents the secret master key of S. Subsequently, S looks up PID_{UR_j} in its database. If a corresponding record is found, S removes the record linked to PID_{UR_j} and requests UR_j to submit a new registration request message.

Step 2: Upon receiving the message from S, UR_j generates a fresh and distinct identity denoted as $ID_{UR_j}^{new}$ and securely sends the registration request message $< ID_{UR_j}^{new} >$ to S. The subsequent steps follow the process outlined in Subsection III-C.

Step 3: UR_j stores $\{C_{MD_j}^{new}, A_{MD_j}^{new}, MAC_{MD_j}^{new}, PUF(\cdot)\}$ in $MD_{UR_j}^{new}$. S keeps the credential $PID_{UR_j}^{new}$ in its database.

IV. SECURITY ANALYSIS

This section evaluates the security of our proposed scheme through both informal and formal (mathematical) analysis methods.

A. INFORMAL SECURITY ANALYSIS

This section demonstrates the robustness of our proposed scheme against potential attacks.

1) IMPERSONATION ATTACK

Suppose adversary $\mathcal{A}$ intercepts the messages exchanged during authentication and key agreement session: $MSG_1 = \{rn_1, T_1\}$, $MSG_2 = \{M_1, M_2, RS_{FMS_i}, T2\}$, $MSG_3 = \{MSG_2, M_3, M_4, RS_{UR_j}, T_3\}$, $MSG_4 = \{X_3, M_5, T_4\}$, and $MSG_5 = \{M_6, T_4, T_5\}$. $\mathcal{A}$ aims to impersonate UR_j by creating valid messages MSG_3 and MSG_5 on behalf of legitimate user or mobile device MD_j. Although $\mathcal{A}$ can create its own random secrets and timestamps, it can't create PID_{UR_j} and can't create a random shared secret and session key as the private key PR_S of S is unknown. Similar restrictions apply to creating valid MSG_2 as the private key PR_S of S and PID_{FMS_i} are unknown to $\mathcal{A}$. Hence, our proposed scheme resists user, FMS unit, and server impersonation attacks.

2) REPLAY ATTACK

Our proposed scheme resists relay attacks because if an adversary $\mathcal{A}$ intercepts messages $MSG_1 \sim MSG_5$ and attempts to replay them later, they will be detected at the recipient's end due to the validation of timestamps included in the messages. The validation of other parameters included in the messages will also fail because timestamps are included in these parameters.

3) MAN-IN-THE-MIDDLE ATTACK

In a scenario where an adversary $\mathcal{A}$ intercepts all transmitted messages MSG_1 to MSG_5 during the authentication and key agreement process and tries to alter them, it becomes clear from the impersonation attack analysis that $\mathcal{A}$ cannot send a genuine message to the recipient S. Hence, the authentication process ends. To alter message MSG_2, $\mathcal{A}$ must have knowledge of ID_{FMS_i}, R_{FMS_i}, K, and PR_S. Similarly, without knowledge of ID_{FMS_i}, R_{FMS_i}, K, ID_{UR_j}, R_{UR_j}, and PR_S, $\mathcal{A}$ cannot generate valid random numbers and timestamps to modify MSG_3, MSG_4, and MSG_5. Thus, our scheme is immune to man-in-the-middle attacks.

4) MUTUAL AUTHENTICATION

The mutual authentication in our proposed scheme occurs as follows: 1) In MSG_3, S verifies if M_4^* matches M_4 to authenticate UR_j; 2) In MSG_3, S then checks if M_2^* equals M_2 to authenticate FMS_i; 3) In MSG_4, UR_j verifies that M_5^* is equal to M_5 to authenticate S; 4) In MSG_5, FMS_i confirms that M_6^* is equal to M_6 to authenticate UR_j.

5) ESL ATTACK

The CK-adversary model assumes that an adversary $\mathcal{A}$ is aware of some ephemeral secrets (e.g., rn_1, rn_2, rn_3, rn_4). The session keys SK_{UF} and SK_{US} include both ephemeral secrets and long-term secrets, such as K, PR_S, and are generated as $SK_{UF} = h(PID_{FMS_i} \parallel SHS_{SF} \parallel rn_1 \parallel rn_3 \parallel T_2 \parallel T_3 \parallel T_4)$ and $SK_{US} = h(PID_{UR_j} \parallel SHS_{US} \parallel rn_1 \parallel T_3 \parallel T_4)$, respectively. It is difficult for the adversary to calculate the session key with just knowledge of rn_1, rn_2, rn_3, rn_4, making the scheme resistant to ESL attacks.

6) STOLEN DEVICE ATTACK

Suppose that the adversary $\mathcal{A}$ has acquired the stolen or lost mobile device MD_j of a legitimate user UR_j. Using PA attacks, $\mathcal{A}$ can extract the stored data $\{A_{MD_j}, MAC_{MD_j}\}$ from MD_j's memory. Despite obtaining this information, $\mathcal{A}$ is unable to extract the encrypted secret credential, i.e., PID_{UR_j}, without accurately guessing ID_{UR_j} and PSW_{UR_j}. As it is computationally infeasible for $\mathcal{A}$ to anticipate both ID_{UR_j} and PSW_{UR_j} accurately, extracting the embedded CRP (C_{MD_j}, R_{MD_j}) from the PUF is also impossible. Likewise, extracting the CRP (C_{FMS_i}, R_{FMS_i}) from the PUF of FMS_i is also impossible. Hence, the proposed scheme is secure against attacks using stolen mobile devices.

7) PRIVILEGED-INSIDER ATTACK

In our proposed scheme, the S holds the sole responsibility for enrolling FMS units and users. To ensure security, none of the registration information is transferred from the registered FMS_i and UR_j to the S. Instead, the FMS_i and UR_j securely obtain the necessary credentials from the S prior to deployment in the smart manufacturing environment. This effectively eliminates the risk of privileged-insider attacks by adversary $\mathcal{A}$.

8) PASSWORD GUESSING ATTACK

If a mobile device is lost or stolen and acquired by $\mathcal{A}$, all the stored parameters, including $\{C_{MD_j}, A_{MD_j}, PUF(\cdot), MAC_{MD_j}\}$, can be retrieved. To obtain the password, $\mathcal{A}$ must compute $PID_{UR_j} = A_{MD_j} \oplus h(ID_{UR_j} \parallel PSW_{UR_j}^l)$ and $MAC_{MD_j}^l =$

$h(PID_{UR_j} \parallel R_{MD_j})$. However, without knowledge of ID_{UR_j}, it is impossible for $\mathcal{A}$ to access the password.

9) ANONYMITY AND UNTRACEABILITY

In the authentication and key agreement process, if $\mathcal{A}$ intercepts messages M_1 to M_5, it cannot determine the user's identity (ID_{UR_j}) or the identity of an FMS_i unit from the messages due to the inclusion of timestamps, random numbers, and secret parameters like K and PR_S. This results in anonymity for both FMS units and users. Additionally, every session has unique random numbers and timestamps, making each message distinct and dynamic, so $\mathcal{A}$ cannot track a user across sessions, resulting in untraceability.

B. SECURITY ANALYSIS THROUGH RANDOM ORACLE MODEL

The ROR is a well-established technique for evaluating the security of session keys in cryptographic systems. In recent times, it has become a widely adopted approach to demonstrate the protection of session keys in authentication protocols.

Before delving into the demonstration of session key security through Theorem 1, it is important to highlight the fundamental components of the ROR.

Participants. We use the symbols $\Omega_{UR}^{t_1}$, $\Omega_{FMS}^{t_2}$, and $\Omega_S^{t_3}$ to represent the t_1^{th}, t_2^{th}, and t_3^{th} of UR_j, FMS_i, and S, respectively. These symbols are also referred to as oracles.

Accepted state. An instance Ω^t is considered to be in an accepted state when it reaches this state upon receipt of the final expected protocol message. The session identification (sid) for Ω^t instance is determined by combining all the messages that were sent and received during the session in the order in which they were communicated.

Partnering. Two instances, Ω^{t_1} and Ω^{t_2}, are considered partners if they meet the following criteria: 1) Both are in a state of acceptance, 2) Both mutually authenticate each other, and 3) Both are reciprocally partnered.

Freshness. An instance $\Omega_{UR_j}^{t_1}$ or $\Omega_{FMS_i}^{t_2}$ is considered fresh if an adversary $\mathcal{A}$ does not possess the session key SK_{ji} through a query known as the Reveal.

We assume that the adversary $\mathcal{A}$ has complete domination over the communication within the network. This gives $\mathcal{A}$ the power to not only eavesdrop on messages, but also manipulate, eliminate, or inject unauthorized messages during the exchange between entities. Moreover, $\mathcal{A}$ has access to the following oracles:

Execute($\Omega_{UR_j}^{t_1}$, $\Omega_{FMS_i}^{t_2}$, $\Omega_S^{t_3}$): By performing this query, $\mathcal{A}$ has the ability to intercept all messages exchanged between UR_j, FMS_i, and S. As a result of this interception, this query is modeled as an eavesdropping attack by $\mathcal{A}$.

Reveal(Ω^t): By performing this query, $\mathcal{A}$ reveals the SKs generated between $\Omega_{UR}^{t_1}$ and $\Omega_{FMS}^{t_3}$ and between $\Omega_{UR}^{t_1}$ and $\Omega_S^{t_2}$.

Send(Ω^t, MSG): With this query, $\mathcal{A}$ can transmit the message MSG to Ω^t and receive the reply message.

CorruptMD($\Omega_{UR}^{t_1}$): By utilizing this query, $\mathcal{A}$ can effectively retrieve the confidential parameters stored in the stolen mobile device.

CorruptFMS($\Omega_{FMS}^{t_2}$): With the aid of this query, $\mathcal{A}$ is able to extract the confidential parameters from the purloined FMS unit.

Test(Ω^t): Through the utilization of this query, $\mathcal{A}$ can transmit a request to Ω^t to obtain the SK. Following this request, Ω^t generates a randomized response based on the outcome of an unbiased coin flip b.

In addition, all participants, including $\mathcal{A}$, are provided access to the collision-resistant hash function $h(\cdot)$ and the PUF function $PUF(\cdot)$. Both functions are used as random oracle mechanisms.

Theorem 1: Suppose $\mathcal{A}$ is an adversary against the proposed scheme $\mathcal{P}$, operating within polynomial time (t_p). The variables Q_s, Q_h, and Q_{puf} indicate the number of Send queries, Hash queries, and PUF queries made by $\mathcal{A}$, respectively. The range space of the $h(\cdot)$ function, key length of PUF, and size of a uniformly distributed password dictionary are denoted by $|Hash|$, $|PUF|$, and $|DT|$. The advantage of $\mathcal{A}$ in compromising the ECDDHP is denoted as $Adv_{\mathcal{A}}^{ECDDHP}(tp)$. We can estimate the advantage of $\mathcal{A}$ in compromising the session key security of our scheme as:

$$Adv_{\mathcal{A}}^{\mathcal{P}}(t_p) \leq \frac{Q_h^2}{|Hash|} + \frac{Q_{puf}^2}{|PUF|} + \frac{2 \cdot Q_s}{|DT|} + 2 \cdot Adv_{\mathcal{A}}^{ECDDHP}(t_p). \quad (1)$$

Proof: The series of games played by $\mathcal{A}$ to compromise the security of $\mathcal{P}$ can be denoted as $Game_k$, where $k = \{0, 1, 2, 3, 4, 5\}$. The variable $Succ_k$ represents the probability of success for $\mathcal{A}$ in winning $Game_k$ within the specified polynomial time t_p. Specifically,

$Game_0^{\mathcal{A}}$: This game represents a simulation of the actual attack by $\mathcal{A}$ against the $\mathcal{P}$. The outcome of the game is determined by flipping an unbiased coin, thus

$$\mathrm{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) = |2 \cdot \mathrm{Prob}[Succ_0] - 1|. \quad (2)$$

$Game_1^{\mathcal{A}}$: In this game, an eavesdropping attack by $\mathcal{A}$ is simulated, where $\mathcal{A}$ listens in on all communication between UR_j, FMS_i, and S during the AKA procedure. $\mathcal{A}$ then runs the query $Execute(\Omega_{UR}^{t_1}, \Omega_{FMS}^{t_2}, \Omega_S^{t_3})$, followed by $Test$ and $Reveal$ to verify the validity of the session keys ($SK_{UR_jF_i} = SK_{F_iUR_j}$ and $SK_{UR_jS} = SK_{SUR_j}$). Note that both long-term and short-term secrets are used to compute the session keys between UR_j and FMS_i and between S and UR_j. It is computationally challenging for $\mathcal{A}$ to compute both keys, and the probability of $\mathcal{A}$ winning $Game_1^{\mathcal{A}}$ remains unchanged from that of $Game_0^{\mathcal{A}}$. Hence, the indistinguishability of $Game_0^{\mathcal{A}}$ and $Game_1^{\mathcal{A}}$ renders:

$$\mathrm{Prob}[Succ_1] = \mathrm{Prob}[Succ_0]. \quad (3)$$

$Game_2^{\mathcal{A}}$: During the adversary's attack, $\mathcal{A}$ executes $Hash$ and $Send$ queries. The goal of these queries is to identify hash collisions and extract information that can be used

to compromise the session key security. While $\mathcal{A}$ performs multiple *Hash* queries to check for collisions, it is unlikely to observe any collisions when running the *Send* query. The reason for this is that each message exchanged in the scheme includes timestamps and random numbers, which significantly reduces the likelihood of a hash collision. As a result, the birthday paradox yields

$$|\text{Prob}[Succ_2] - \text{Prob}[Succ_1]| \leq \frac{Q_h^2}{2|Hash|}. \tag{4}$$

$Game_3^{\mathcal{A}}$: The game extends $Game_2^{\mathcal{A}}$ and simulates the $PUF()$ query. It's important to mention that the PUFs in FMS_i and MD_j are secure, and therefore

$$|\text{Prob}[Succ_3] - \text{Prob}[Succ_2]| \leq \frac{Q_{puf}^2}{2|PUF|}. \tag{5}$$

$Game_4^{\mathcal{A}}$: This game simulates attacks on stolen or lost MD_j and password guessing. By using the $CorruptMD(\Omega_{UR}^{t_1})$ query, $\mathcal{A}$ can obtain $\{C_{MD_j}, A_{MD_j}, PUF(\cdot), MAC_{MD_j}\}$ from a stolen or lost MD_j. The goal for $\mathcal{A}$ is to extract the encrypted secret information, i.e., PID_{UR_j}. To win the game, $\mathcal{A}$ must successfully determine both ID_{UR_j} and PSW_{UR_j} from a limited number of guesses from the DT within a limited number of tries, and therefore

$$|\text{Prob}[Succ_4] - \text{Prob}[Succ_3]| \leq \frac{Q_s}{|DT|}. \tag{6}$$

$Game_5^{\mathcal{A}}$: In this game, the primary objective of $\mathcal{A}$ is to execute an active attack, with the purpose of acquiring the session keys. To achieve this goal, $\mathcal{A}$ makes use of all intercepted messages, which includes MSG_1 through MSG_5 from FMS_i, MD_j, and S, in addition to other confidential parameters obtained from previous games. To accomplish this, $\mathcal{A}$ must compute $SK_{UF} = h(PID_{FMS_i} \parallel SHS_{SF} \parallel rn_1 \parallel rn_3 \parallel T_2 \parallel T_3 \parallel T_4)$ and $SK_{US} = h(PID_{UR_j} \parallel SHS_{US} \parallel rn_1 \parallel T_3 \parallel T_4)$. In simpler terms, $\mathcal{A}$ must successfully solve the ECDDHP to obtain the session keys. It follows that

$$|\text{Prob}[Succ_5] - \text{Prob}[Succ_4]| \leq \text{Adv}_{\mathcal{A}}^{ECDDHP}(t_p). \tag{7}$$

After completing all the games, $\mathcal{A}$ runs a "*Test*" query. A fair coin is then flipped to assess the semantic security of the session keys. Thus,

$$\text{Prob}[Succ_5] = \frac{1}{2}. \tag{8}$$

Thus, from (2) we have

$$\frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) = \left|\text{Prob}[Succ_0] - \frac{1}{2}\right|. \tag{9}$$

Using (8) and (9) as well as noting (3), we obtain

$$\begin{aligned}\frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) &= |\text{Prob}[Succ_0] - \text{Prob}[Succ_5]| \\ &= |\text{Prob}[Succ_1] - \text{Prob}[Succ_5]|. \end{aligned} \tag{10}$$

TABLE 2. Execution time for various primitives [22].

Crytographic operation	Approximated execution time (ms)
T_H: Hash function	0.311
T_E: ECC point multiplication	4.12
T_A: ECC point addition	0.273
$T_F \approx T_E$: Fuzzy extractor	4.12
T_P: physical unclonable function	0.51 μs
T_S/T_D: Symmetric encryption/ decryption	0.413

By employing the well-known triangle inequality to (10), we have

$$\begin{aligned}\frac{1}{2}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) \leq\ & |\text{Prob}[Succ_1] - \text{Prob}[Succ_2]| \\ &+ |\text{Prob}[Succ_2] - \text{Prob}[Succ_3]| \\ &+ |\text{Prob}[Succ_3] - \text{Prob}[Succ_4]| \\ &+ |\text{Prob}[Succ_4] - \text{Prob}[Succ_5]|. \end{aligned} \tag{11}$$

When (4), (5), (6), and (7) into (11), we get

$$\begin{aligned}\text{Adv}_{\mathcal{A}}^{\mathcal{P}}(t_p) \leq\ & \frac{Q_h^2}{|Hash|} + \frac{Q_{puf}^2}{|PUF|} + 2 \cdot \frac{Q_s}{|DT|} \\ &+ 2 \cdot \text{Adv}_{\mathcal{A}}^{ECDDHP}(t_p). \end{aligned} \tag{12}$$

i.e., (1). This completes the proof. ■

V. COMPARATIVE ANALYSIS

In this section, we compare the proposed scheme with similar existing schemes, such as Sutrala et al. [23], Chen et al. [16], He and Zeadally [24], Challa et al. [25], and Wazid et al. [26]. The comparison details are as follows.

A. COMPUTATION OVERHEAD

In this section, we compare the computational overhead of the proposed schemes with related schemes. The comparison is based on the computation time of cryptographic operations provided in Table 2 and uses experimental results reported in [22]. To ensure a fair comparison, we exclude the cryptographic operations involved in the registration phase of both the proposed and related schemes as this is a one-time process. Therefore, we focus only on the cryptographic operations used in the login and authentication phase to calculate the computational overhead of the proposed and related schemes. We analyze the computational overhead of Sutrala et al. [23], Chen et al. [16], He and Zeadally [24], Challa et al. [25], and Wazid et al. [26].

In the proposed scheme, the user performs $T_P + 2T_E + 7T_H$ cryptographic operations, which takes approximately 10.41751 ms to complete. Similarly, the server performs $2T_E + 9T_H$ cryptographic operations, which takes approximately 11.039 ms to complete. Furthermore, $T_P + 2T_E + 5T_H$ cryptographic operations are executed at the end device, which costs approximately 9.79551 ms. Therefore, the aggregated computation overhead of the proposed scheme is 31.25202 ms. We also computed the computation overhead

TABLE 3. Computational overhead comparison.

Scheme	User	Server/Gateway	FMS unit/Smart device	Approximate total execution time (ms)
Sutrala *et al.* [23]	$T_F + 5T_E + 2T_A + 16T_H \approx 30.242$	$3T_E + 2T_A + 9T_H \approx 15.705$	$4T_E + T_A + 8T_H \approx 19.241$	$T_F + 12T_E + 5T_A + 33T_H \approx 65.188$
Chen *et al.* [16]	$3T_E + 5T_H \approx 13.915$	$T_E + 7T_H \approx 6.297$	$2T_E + 3T_H \approx 9.173$	$6T_E + 15T_H \approx 29.385$
He-Zeadally [24]	$3T_E + 2T_S + 2T_H \approx 13.808$	$T_E + 4T_S + T_H \approx 6.083$	$2T_E + 2T_S + T_H \approx 9.377$	$6T_E + 8T_S + 4T_H \approx 29.268$
Challa *et al.* [25]	$5T_E + 5T_H \approx 22.155$	$5T_E + 4T_H \approx 21.844$	$4T_E + 3T_H \approx 17.413$	$14T_E + 12T_H \approx 61.412$
Wazid *et al.* [26]	$T_F + 4T_E + T_A + 19T_H \approx 26.782$	$5T_E + T_A + T_H \approx 21.184$	$4T_E + T_A + 12T_H \approx 20.485$	$T_F + 13T_E + 3T_A + 32T_H \approx 68.451$
Proposed scheme	$T_P + 2T_E + 7T_H \approx 10.41751$	$2T_E + 9T_H \approx 11.039$	$T_P + 2T_E + 5T_H \approx 9.79551$	$2T_P + 6T_E + 21T_H \approx 31.25202$

TABLE 4. Security and functionality features analysis.

Feature	[23]	[16]	[24]	[25]	[26]	Our scheme
$\mathcal{SF}_{\infty}$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_{\in}$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_{\ni}$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_{\triangle}$	✓	✓	✓	×	✓	✓
$\mathcal{SF}_{\triangledown}$	✓	×	✓	×	✓	✓
$\mathcal{SF}_{/}$	×	×	×	×	×	✓
$\mathcal{SF}_{!}$	✓	✓	×	✓	✓	✓
$\mathcal{SF}_{\forall}$	✓	✓	✓	✓	✓	✓
$\mathcal{SF}_{\exists}$	✓	✓	✓	×	✓	✓
$\mathcal{SF}_{\infty\prime}$	✓	✓	✓	×	✓	✓
$\mathcal{SF}_{\infty\infty}$	✓	×	✓	×	✓	✓
$\mathcal{SF}_{\infty\in}$	✓	×	✓	✓	✓	✓
$\mathcal{SF}_{\infty\ni}$	✓	✓	×	✓	×	✓

Note: $\mathcal{SF}_{\infty}$: "mutual authentication", $\mathcal{SF}_{\in}$: "key agreement", $\mathcal{SF}_{\ni}$: "replay attack", $\mathcal{SF}_{\triangle}$: "impersonation attacks", $\mathcal{SF}_{\triangledown}$: "untraceability", $\mathcal{SF}_{/}$:"FMS unit/smart device theft attack", $\mathcal{SF}_{!}$: "mobile device capture/theft attack", $\mathcal{SF}_{\forall}$: "MitM attack", $\mathcal{SF}_{\exists}$: "anonymity", $\mathcal{SF}_{\infty\prime}$: "password update attack", $\mathcal{SF}_{\infty\infty}$: "privileged insider attack", $\mathcal{SF}_{\infty\in}$: "ESL attack", and $\mathcal{SF}_{\infty\ni}$: "validated through formal (mathematical) model".
✓: signifies the feature's availability, ×: denotes the feature's unavailability.

of the related schemes [16], [23], [24], [25], and [26] in the same way, and the results are shown in Table 3. As shown in Table 3, the proposed scheme has less computation overhead than all related schemes [23], [25], and [26] except for schemes [16] and [24].

B. COMMUNICATION OVERHEAD

Assuming bit sizes of 128 for identity and a random nonce, 32 for the current timestamp, 256 for the hash output/digest, and 320 for the elliptic curve point, our proposed scheme requires the transmission of five messages (MSG_1–MSG_5) between a user UR_j, a server S, and an FMS unit (FMS_i). This results in a total communication overhead of 3616 bits for exchanging five messages. The communication overheads of the proposed scheme have been compared with those of other related schemes in Table 5. While the total communication overhead of our proposed scheme is higher than that of the related schemes, it can be justified as our proposed scheme has more security and functionality traits (see Table 4).

C. SECURITY AND FUNCTIONALITY FEATURES COMPARISON

In Table 4, we present a comprehensive comparison of the essential security and functionality features ($\mathcal{SF}_{\infty}$–$\mathcal{SF}_{\infty\ni}$)

TABLE 5. Comparison of communication overheads.

Scheme	No. of messages	No. of bits
Sutrala *et al.* [23]	3	3200
Chen *et al.* [16]	3	2976
He-Zeadally [24]	4	3232
Challa *et al.* [25]	3	2528
Wazid *et al.* [26]	3	3360
Proposed scheme	5	3616

of our proposed scheme and other existing competing schemes. Our analysis clearly indicates that our proposed scheme surpasses the other competing schemes in terms of these features. Therefore, our scheme can be considered as a better choice for users who prioritize robust security and comprehensive functionality.

VI. CONCLUSION

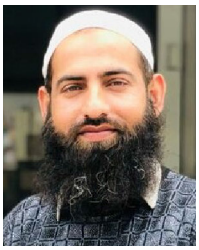
In this paper, we proposed a lightweight authentication scheme that addressed the need for secure communication in the smart manufacturing industry. The scheme established mutual authentication between FMS unit, user, and the server, and generated two secure session keys for communication. Our security analysis, including both formal and informal methods, demonstrated the scheme's effectiveness against various known attacks. Furthermore, our performance analysis showed improved functionality and a higher level of suitability compared to other existing related schemes. These advantages made our proposed scheme a promising solution for secure communication in the smart manufacturing industry. However, we acknowledge that maintaining two independent sessions may increase performance requirements. Additionally, evaluating the proposed scheme in a real-world environment, such as implementing it in a test sub-network, remains a key area of future research for us. Moreover, exploring the development of a more secure and lightweight privacy-preserving authentication scheme specifically for cloud-based IIoT environments represents an intriguing avenue for future research.

REFERENCES

[1] L. A. Estrada-Jimenez, T. Pulikottil, S. Nikghadam-Hojjati, and J. Barata, "Self-organization in smart manufacturing—Background, systematic review, challenges and outlook," *IEEE Access*, vol. 11, pp. 10107–10136, 2023.

[2] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Enabling flexible manufacturing system (FMS) through the applications of industry 4.0 technologies," *Internet Things Cyber-Phys. Syst.*, vol. 2, pp. 49–62, Jan. 2022.
[3] G. Lefranc, "Review of trends in manufacturing systems based on industry 4.0: The opportunities," in *Intelligent Methods Systems and Applications in Computing, Communications and Control*. Cham, Switzerland: Springer, 2022, pp. 182–192.
[4] N. Z. Dlamini, K. Mpofu, I. Daniyan, and B. Ramatsetse, "An overview of the manufacturing systems: A literature survey," in *Smart, Sustainable Manufacturing in an Everchanging World*. Cham, Switzerland: Springer, 2023, pp. 905–927.
[5] T. Jiao and H. Shi, "A customized supervisory control approach for flexible manufacturing systems," *IEEE Access*, vol. 11, pp. 40525–40531, 2023.
[6] V. Dohale, A. Gunasekaran, M. M. Akarte, and P. Verma, "52 years of manufacturing strategy: An evolutionary review of literature (1969–2021)," *Int. J. Prod. Res.*, vol. 60, no. 2, pp. 569–594, Jan. 2022.
[7] C. Cronin, A. Conway, and J. Walsh, "Flexible manufacturing systems using IIoT in the automotive sector," *Proc. Manuf.*, vol. 38, pp. 1652–1659, Jan. 2019.
[8] A. C. B. Monteiro, R. P. Fran, R. Arthur, Y. Iano, A. C. Segatti, G. P. Carnielli, J. C. Pereira, H. A. de Godoy, and E. C. Fernandes, "A look at IIoT: The perspective of IoT technology applied in the industrial field," in *The Industrial Internet of Things (IIoT) Intelligent Analytics for Predictive Maintenance*. 2022, pp. 1–29.
[9] M. Soori, B. Arezoo, and R. Dastres, "Internet of Things for smart factories in industry 4.0, a review," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 192–204, Jan. 2023.
[10] P. M. Rao and B. D. Deebak, "A comprehensive survey on authentication and secure key management in Internet of Things: Challenges, countermeasures, and future directions," *Ad Hoc Netw.*, vol. 146, Jul. 2023, Art. no. 103159.
[11] K. Choudhary, G. S. Gaba, I. Butun, and P. Kumar, "MAKE-IT—A lightweight mutual authentication and key exchange protocol for industrial Internet of Things," *Sensors*, vol. 20, no. 18, p. 5166, Sep. 2020.
[12] L. Fang, H. Zhang, M. Li, C. Ge, L. Liu, and Z. Liu, "A secure and fine-grained scheme for data security in industrial IoT platforms for smart city," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7982–7990, Sep. 2020.
[13] F. Rafique, M. S. Obaidat, K. Mahmood, M. F. Ayub, J. Ferzund, and S. A. Chaudhry, "An efficient and provably secure certificateless protocol for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8039–8046, Nov. 2022.
[14] M. H. Eldefrawy, N. Ferrari, and M. Gidlund, "Dynamic user authentication protocol for industrial IoT without timestamping," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, Sundsvall, Sweden, May 2019, pp. 1–7.
[15] B. Harishma, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay, "POSTER: Authenticated key-exchange protocol for heterogeneous CPS," in *Proc. Asia Conf. Comput. Commun. Secur.*, May 2018, pp. 849–851.
[16] Y. Chen, J.-F. Martínez, 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, pp. 1–17, 2017.
[17] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[18] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. EUROCRYPT*, Amsterdam, The Netherlands, May 2002, pp. 337–351.
[19] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
[20] R. Amin, S. K. H. Islam, N. Kumar, and K.-K.-R. Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 104, pp. 133–144, Feb. 2018.
[21] K. Chuang, E. Bury, R. Degraeve, B. Kaczer, D. Linten, and I. Verbauwhede, "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.
[22] M. Tanveer, A. Alkhayyat, A. U. Khan, N. Kumar, and A. G. Alharbi, "REAP-IIoT: Resource-efficient authentication protocol for the industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24453–24465, Dec. 2022.
[23] A. K. Sutrala, M. S. Obaidat, S. Saha, A. K. Das, M. Alazab, and Y. Park, "Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2316–2330, Mar. 2022.
[24] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 71–77, Jan. 2015.
[25] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.
[26] M. Wazid, A. K. Das, N. Kumar, and M. Alazab, "Designing authenticated key management scheme in 6G-enabled network in a box deployed for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7174–7184, Oct. 2021.

MUHAMMAD HAMMAD received the bachelor's and master's degrees from the University of Engineering and Technology, Lahore. He is currently pursuing the Ph.D. degree with the Faculty of Mechanical Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan. His primary research focuses on security in smart manufacturing systems. His research interests include the security of the Internet of Things (IoT), industrial Internet of Things (IIoT), and cyber-physical 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 with the Department of Software Engineering, University of Malakand, Dir Lower, Pakistan. His research interests include cryptography, network security, and blockchain technology. He has made notable contributions to his field, with multiple research findings published in renowned journals, such as the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and *IT Professional*.

GHULAM ABBAS (Senior Member, IEEE) received the B.S. degree in computer science from the University of Peshawar, Pakistan, in 2003, and the M.S. degree in distributed systems and the Ph.D. degree in computer networks from the University of Liverpool, U.K., in 2005 and 2010, respectively. From 2006 to 2010, he was a Research Associate with Liverpool Hope University, U.K., where he was associated with the Intelligent and Distributed Systems Laboratory. Since 2011, he has been with the Faculty of Computer Sciences and Engineering, Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences and Technology, Pakistan. He is currently an Associate Professor and the Director of the Huawei Network Academy. He is a Co-Founding Member of the Telecommunications and Networking (TeleCoN) Research Laboratory, GIK Institute of Engineering Sciences and Technology. His research interests include computer networks and wireless and mobile communications. He is a fellow of the Institute of Science and Technology, U.K., and the British Computer Society.

HISHAM ALASMARY received the M.Sc. degree in computer science from George Washington University, Washington, DC, USA, in 2016, and the Ph.D. degree from the Department of Computer Science, University of Central Florida, in 2020. He is an Assistant Professor with King Khalid University. His research interests include software security, IoT security and privacy, ML/DL applications in information security, and adversarial machine learning.

MUHAMMAD WAQAS (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan, in 2009 and 2014, respectively, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2019. From 2012 to 2015, he was a Lecturer and a Program Coordinator with the Sarhad University of Science and Information Technology, Peshawar. From October 2019 to September 2021, he was a Postdoctoral Researcher with the Faculty of Information Technology, Beijing University of Technology, Beijing. Currently, he is an Assistant Professor with the Computer Engineering Department, College of Information Technology, University of Bahrain, Bahrain. He is also an Adjunct Senior Lecturer with the School of Engineering, Edith Cowan University, Perth, Australia; and a Visiting Researcher with the Faculty of Information Technology, Beijing University of Technology. He has several research publications in reputed journals and conferences. His current research interests include physical layer security, vehicular networks, mobile edge computing, and the Internet of Things. He received the Best Paper Award at ASSP, in 2021. He is the co-chair, TPC member, and a reviewer of several international conferences and journals.

WASIM AHMED KHAN received the degree in mechanical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, and the Ph.D. degree in operations research from the Department of Mechanical Engineering, University of Shieffield, England, U.K. He has diverse work experience, including working with manufacturing industry, software development for local and overseas clients, and teaching production engineering, business, and computer science students. He is currently a Professor with the Faculty of Mechanical Engineering, GIK Institute of Engineering Sciences and Technology. He is also the Director of the Office of Research, Innovation and Commercialization (ORIC); and a Senior Advisor, an Incubator, and a Coordinator of the GIK Institute Professional Education Program. He is a Life Member of the Pakistan Engineering Council. He is also a fellow of the Institution of Mechanical Engineers (FIMechE), U.K.