AUTHENTICATED KEY AGREEMENT PROTOCOL WITH FORWARD SECRECY FOR SECURING CYBER PHYSICAL SYSTEMS

Sung-Woon Lee [1] and Hyunsung Kim [2]

[1] Dept. of Information System and Security, Tongmyong University, Korea
[2] Dept. of Smart Engineering, Kyungil University, Korea

ABSTRACT

As Internet of Things (IoT) technologies become increasingly embedded within cyber-physical systems (CPS), safeguarding communications and protecting data have emerged as significant challenges. Existing authenticated key agreement protocols play a fundamental role in securing these systems, but many fail to provide adequate protection against evolving security threats, particularly in terms of forward secrecy. This paper conducts an in-depth cryptanalysis of a newly introduced authentication protocol, highlighting its failure to provide forward secrecy-a crucial feature that maintains communication confidentiality even in the event that the long-term key is exposed. Through detailed analysis, we identify several vulnerabilities within the protocol, particularly its inability to withstand attacks targeting session key exposure. Building on these findings, we propose a new authenticated key agreement protocol specifically designed to enhance security in CPS environments. Our proposed protocol integrates forward secrecy, ensuring that even if an attacker gains access to a device's long-term key, communications remain protected. The protocol aims to deliver robust security assurances with minimal computational demands, rendering it particularly well-suited for IoT devices with limited resources. Our evaluation assesses its security, efficiency, and scalability, and the results indicate that it outperforms current alternatives. The proposed protocol offers a robust, forward secrecy-enabled solution for securing CPS communications, providing a solid foundation for the future of secure IoT networks in critical applications.

KEYWORDS

Security Protocols, Authentication, Key Agreement, Cyber Physical System, Forward Secrecy

1. INTRODUCTION

The rapid growth of IoT devices has resulted in their extensive adoption within CPS, serving as key components in diverse fields such as smart cities, healthcare, industrial automation, and agriculture [1-4]. CPS relies heavily on secure communication between interconnected devices, as these systems often handle sensitive data related to critical infrastructure and operations. A fundamental approach to securing these systems involves the use of authenticated key agreement (AKA) protocols. These protocols facilitate the mutual authentication of devices and the secure exchange of cryptographic keys, thereby establishing protected communication channels [5-6].

However, many of the existing AKA protocols for CPS, particularly in the context of IoT, do not sufficiently address the need for forward secrecy [7]. A crucial feature in contemporary security systems is the ability to protect past communications even if security measures are breached. This concept, often referred to as forward secrecy, ensures that even if an attacker obtains the long-term secret key, they cannot use it to decipher previously intercepted messages. This feature enhances security by ensuring that each communication session remains isolated and unaffected

DOI: 10.5121/ijcnc.2025.17407

by others. This property is particularly crucial in a CPS environment, where attackers gaining access to long-term key could potentially compromise data exchanges, undermining the entire security framework. Although forward secrecy is essential, numerous commonly used AKA protocols either lack this property altogether or incorporate it in ways that are not practical for devices with limited resources. These shortcomings expose IoT devices and CPS to a range of security risks, including the possibility of session key exposure and attacks targeting historical communication sessions [8-10]. With the increasing scale and complexity of IoT devices within CPS environments, there is a growing demand for authentication protocols that are not only secure and efficient but also support forward secrecy.

Recent research has attempted to incorporate forward secrecy into IoT protocols, but the integration of this property remains limited. For example, some studies have explored adding ephemeral key pairs to the elliptic curve Diffie-Hellman (ECDH) protocol to achieve forward secrecy [11-12]. However, these solutions often introduce significant computational overhead, making them impractical for the lightweight devices commonly used in CPS [13]. Additionally, many forward secrecy-enhanced protocols focus on general-purpose IoT networks, which have different requirements than those found in CPS applications. As a result, while forward secrecy is gaining traction in some areas of IoT security, it is still not widely adopted or effectively implemented in protocols designed for CPS. Moreover, existing work often overlooks the computational efficiency of implementing forward secrecy. Since CPS devices often operate with limited resources, incorporating forward secrecy should not result in substantial increases in computational or energy consumption. While the tradeoff between security and performance is a well-known challenge in cryptography, many proposed solutions to this issue fail to strike an appropriate balance for CPS environments. This results in protocols that are either too complex or too resource-intensive to be practical for real-world deployment in critical infrastructure systems.

In this paper, we first review and perform a detailed cryptanalysis of a recently proposed AKA protocol, focusing on its inability to provide forward secrecy. Our analysis identifies key vulnerabilities, such as susceptibility to attacks that could lead to the exposure of session keys, thus compromising the confidentiality of the communications. Building on these insights, we propose a new AKA protocol designed specifically for CPS environments. Our proposed solution integrates forward secrecy into the protocol while maintaining computational efficiency, making it suitable for resource-constrained IoT devices often found in these systems. We evaluate the protocol's performance in terms of security and efficiency, demonstrating its ability to provide enhanced protection for CPS communications while minimizing overhead.

2. BACKGROUND

This section provides a fundamental understanding of hash function and forward secrecy, along with an exploration of relevant related works in the context of CPS and IoT security.

2.1. Security Primitives

This subsection provides an overview of the hash function and property of forward secrecy. Cryptographic systems often employ a mathematical operation known as a hash function, denoted as $h(\cdot)$ [14]. This function accepts various inputs and produces a fixed-size output, commonly referred to as a digest or hash value. The output produced by these functions uniquely corresponds to the original input data. One of their defining features is their one-way design, which makes reversing the process to retrieve the initial input from the hash value computationally infeasible. This characteristic plays a crucial role in maintaining the integrity and security of hashed data. Such functions are integral to numerous cryptographic protocols-such as digital signatures, message authentication codes, and key agreement schemes-where they provide a distinct data fingerprint to verify integrity. Hash functions are vital for ensuring the security of AKA protocols. When applied to a key exchange process, they are used to verify the integrity of the exchanged messages and prevent tampering. In the context of CPS, hash functions also play a critical role in reducing the computational overhead of security mechanisms, especially in resource-constrained IoT environments. Widely used cryptographic hash functions like secure hash algorithm-256 (SHA-256) are often chosen for their ability to offer strong security assurances while remaining computationally efficient. This balance is essential for the broad implementation of secure protocols in IoT devices.

Forward secrecy is a cryptographic feature this protects the confidentiality of session keys, ensuring they remain secure even if the long-term secret key is exposed at a later date [7]. In traditional key exchange protocols, the long-term private key is often used to derive session keys, which can be vulnerable if these private key is exposed. Forward secrecy prevents this issue by ensuring that session keys are ephemeral and not linked to long-term private key, meaning that compromising a device’s long-term key does not jeopardize the confidentiality of communications. The concept of forward secrecy has gained significant importance in modern cryptography, particularly in the context of IoT and CPS, where devices are deployed in large-scale, potentially hostile environments. In CPS, the exposure of historical communication sessions could have disastrous consequences, especially in industries like agriculture, healthcare, energy, and manufacturing.

2.2. Related Works

Over the years, many cryptographic protocols have been proposed to secure key exchange and authentication in IoT-based CPS [15-24]. Early protocols focused primarily on mutual authentication and secure key exchange, relying on algorithms such as RSA, Diffie-Hellman, and ECDH. However, most of these protocols do not adequately consider forward secrecy, leaving CPS vulnerable to attacks in the event of long-term key compromise.

Mansoor et al. introduced a security protocol for IoT devices using radio frequency identification systems [15]. Their approach leverages symmetric encryption techniques to establish secure communications. The protocol’s core mechanism relies on a shared secret key between the central server and the tag. This key facilitates mutual verification of identities. To enhance security, the protocol incorporates a key renewal process that takes place during each communication session. However, challenges arise concerning the synchronization of the shared key states in the protocol. Braeken proposed a symmetric key-based authentication protocol between servers and sensors [16]. In this protocol, the shared key remains constant, and auxiliary information is used to authenticate identities. The protocol incorporates a mechanism to refresh supplementary data in every session. However, this information is transmitted through unsecured public networks. This approach introduces a potential security weakness. In the event that the shared secret is exposed, it could lead to the compromise of all previous communication sessions. This vulnerability stems from the reliance on a single shared key for securing multiple interactions. Braeken in [17] built upon the work in [16] with a symmetric AKA protocol that ensures unlinkability. In this scheme, user equipment and home networks share a pre-established key for mutual authentication, protecting prior sessions even if the pre-shared key is disclosed by preventing the session key from being computed. Ali et al. introduced a security protocol designed for unmanned aerial vehicles [18]. Their system establishes secret keys shared between users and ground control servers, as well as between servers and the drones themselves. However, this approach shares a common weakness with another research: if the shared keys are exposed, the entire communication channel becomes vulnerable [19]. In a separate research, Das et al. developed a novel authentication method for industrial IoT (IIoT) systems deployed in cloud environments [20]. Their approach incorporates biometric data to enhance user verification while also addressing privacy concerns. Hussain and Chaudhry showed that Das et al.'s scheme does not provide perfect forward secrecy [21]. Srinivas et al. proposed a new elliptic curve cryptography (ECC)-based three-factor user authentication scheme for big data collection in IoT environment [22]. They provided formal security analysis under real-or-random model and automated validation of Internet security protocols and applications (AVISPA) tool. However, Tanveer et al. showed that Srinivas et al.'s protocol is susceptible to session key compromised, user impersonation, parallel session, user anonymity, and device impersonation attacks [23].

3. GHAZO ET AL.'S AKA PROTOCOL CRYPTANALYSIS

This section offers a concise examination of the IIoT authentication protocol developed by Ghazo et al., which is designed for critical infrastructure and manufacturing environments. The analysis presented here particularly emphasizes the protocol's security properties, with a specific focus on evaluating its forward secrecy capabilities. This evaluation aims to assess the protocol's resilience against potential future compromises.

Table 1. Notations.

Notation	Description
TA	Trusted authority/system administrator
C_i	ith user
$IIoT_j$	jth industrial IoT device
ID_i	Identity of C_i
ID_j	Identity of $IIoT_j$
T_i	Timestamp
K_S	Session key
$K_{p/p}{}^{TA}$	Private and public master keys of TA
K_m	Master private key generated by TA
r_i, r_j	Random numbers
$h(\cdot)$	One-way hash function
ΔT	Maximum transmission delay
$\oplus$	Exclusive OR operation
$\|\|$	Concatenation operation
$k_{i\text{-}1}$	Previous session key

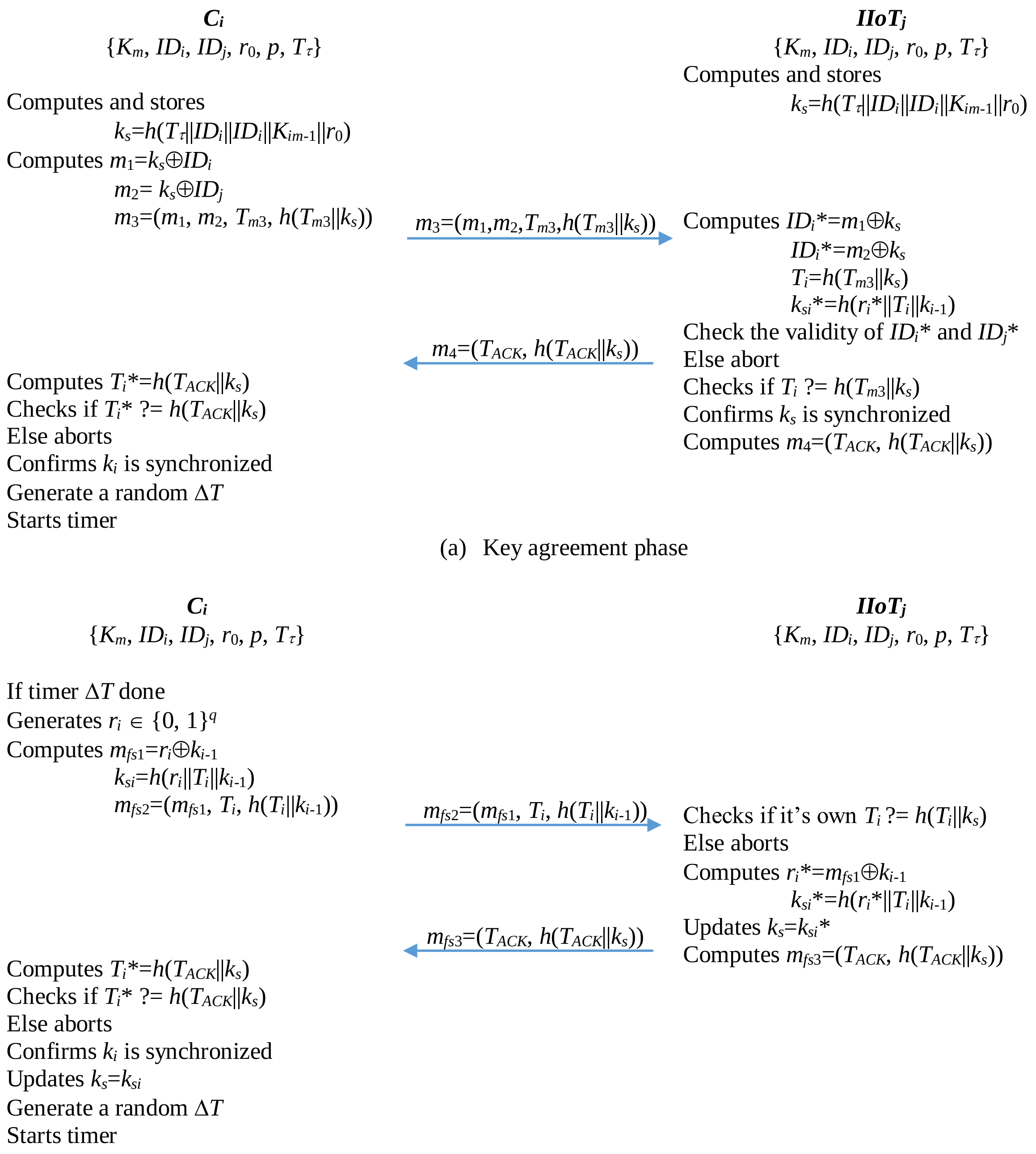
3.1. Ghazo et al.'s AKA Protocol Review

This subsection shows that Ghazo et al.'s protocol has lack of forward secrecy and unlinkability based on Canetti & Krawczyk (CK) adversary model. We will define CK adversary model first and then provide the lack of property concerns in Ghazo et al.'s protocol.

CK adversary model is used to assess the security of AKA protocols within the context of the IoT and CPS [25]. According to this model, the adversary, denoted as $\mathcal{A}$, is assumed to possess the following capabilities:

- $\mathcal{A}$ possesses unrestricted access to and manipulation capabilities over all communication pathways connecting the involved parties. This includes the ability to intercept, eavesdrop on, inject, alter, and delete any messages exchanged over the public channel.
- To evaluate forward secrecy, $\mathcal{A}$ may be granted the ability to compromise legitimate parties and retrieve their long-term secret key.

- $\mathcal{A}$ can exploit side-channel attacks to extract secret information stored in the memory of $IIoT_j$, particularly if one of these devices is stolen or otherwise acquired by $\mathcal{A}$.
- $\mathcal{A}$ could be a legitimate user who has turned malicious, using their position to undermine the security of the system.
- $\mathcal{A}$ has the capability to carry out several known types of attacks, such as impersonation attacks, replay attacks, and attacks involving the extraction of session-specific temporary data.



C_i
$\{K_m, ID_i, ID_j, r_0, p, T_\tau\}$

Computes and stores
$k_s=h(T_\tau||ID_i||ID_i||K_{im-1}||r_0)$
Computes $m_1=k_s\oplus ID_i$
$m_2= k_s\oplus ID_j$
$m_3=(m_1, m_2, T_{m3}, h(T_{m3}||k_s))$

$m_3=(m_1,m_2,T_{m3},h(T_{m3}||k_s))$ →

$IIoT_j$
$\{K_m, ID_i, ID_j, r_0, p, T_\tau\}$
Computes and stores
$k_s=h(T_\tau||ID_i||ID_i||K_{im-1}||r_0)$

Computes $ID_i*=m_1\oplus k_s$
$ID_i*=m_2\oplus k_s$
$T_i=h(T_{m3}||k_s)$
$k_{si}*=h(r_i*||T_i||k_{i-1})$
Check the validity of ID_i* and ID_j*
Else abort
Checks if T_i ?= $h(T_{m3}||k_s)$
Confirms k_s is synchronized
Computes $m_4=(T_{ACK}, h(T_{ACK}||k_s))$

← $m_4=(T_{ACK}, h(T_{ACK}||k_s))$

Computes $T_i*=h(T_{ACK}||k_s)$
Checks if T_i* ?= $h(T_{ACK}||k_s)$
Else aborts
Confirms k_i is synchronized
Generate a random ΔT
Starts timer

(a) Key agreement phase

C_i
$\{K_m, ID_i, ID_j, r_0, p, T_\tau\}$

If timer ΔT done
Generates $r_i \in \{0, 1\}^q$
Computes $m_{fs1}=r_i\oplus k_{i-1}$
$k_{si}=h(r_i||T_i||k_{i-1})$
$m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{i-1}))$

$m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{i-1}))$ →

$IIoT_j$
$\{K_m, ID_i, ID_j, r_0, p, T_\tau\}$

Checks if it's own T_i ?= $h(T_i||k_s)$
Else aborts
Computes $r_i*=m_{fs1}\oplus k_{i-1}$
$k_{si}*=h(r_i*||T_i||k_{i-1})$
Updates $k_s=k_{si}*$
Computes $m_{fs3}=(T_{ACK}, h(T_{ACK}||k_s))$

← $m_{fs3}=(T_{ACK}, h(T_{ACK}||k_s))$

Computes $T_i*=h(T_{ACK}||k_s)$
Checks if T_i* ?= $h(T_{ACK}||k_s)$
Else aborts
Confirms k_i is synchronized
Updates $k_s=k_{si}$
Generate a random ΔT
Starts timer

(b) Forward secrecy phase

Figure 1. Ghazo et al.'s AKA phase

Lack of forward secrecy: If the adversary $\mathcal{A}$ gains access to the long-term key K_m and acquires $\{K_m, ID_i, ID_j, r_0, p, T_\tau\}$ in $IIoT_j$'s memory based on the CK model in Ghazo et al.'s AKA protocol, they can compute all session keys exchanged between C_i and $IIoT_j$ as shown in Fig. 2. This type of attack becomes feasible because Ghazo et al.'s protocol does not uphold the principle of forward secrecy. Following this compromise, $\mathcal{A}$ not only gains access to the previously established session keys but can also determine every necessary information of network entities. Specifically, $\mathcal{A}$ can know the device identity ID_i. It only requires a maximum computational cost of $2T_h$. This is concerning, as it compromises user privacy by exposing device identities. However, this is against the authors' claim of providing forward secrecy to their protocol. Additionally, the absence of forward secrecy exposes the protocol to threats that exploit session-specific temporary data, thereby increasing the risk of security and privacy breaches within the system.

— *$\mathcal{A}$'s capability*:

(1) Acquiring the long-term secret key K_m.

(2) Acquiring $\{ K_m, ID_i, ID_j, r_0, p, T_\tau\}$ in the memory of $IIoT_j$

(3) Eavesdropping every messages from the beginning until the current session, which are composed of $m_3=(m_1, m_2, T_{m3}, h(T_{m3}||k_s))$, $m_4=(T_{ACK}, h(T_{ACK}||k_s))$, $m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{i-1}))$ and $m_{fs3}=(T_{ACK}, h(T_{ACK}||k_s))$

— *The attack result*: Obtaining all session keys k_s.

— *The attack steps of $\mathcal{A}$*:

Step 1. Computes $k_s=h(T_\tau||ID_i||ID_j||K_m||r_0)$ from the first session message of Fig. 2 (a) phase

Step 2. Computes $T_i=h(T_{ACK}||k_s)$ by using T_{ACK} in m_4 from the first session message of Fig. 2 (a) phase

Step 3. Withdraws $r_i^*=m_{fs1}\oplus k_s$ from m_{fs2} from the first session message of Fig 2. (b) phase

Step 4. Computes $k_s=h(r_i^*||T_i||k_s)$

.

.

— *The time overhead of $\mathcal{A}$*: $2T_h$ for the first phase but $1T_h+1T_{XOR}$, where T_h is used to indicate the duration needed to execute the hash function, and T_{XOR} symbolizes the time taken to perform an exclusive OR operation

Figure 2. Attack flow for the forward secrecy in Ghazo et al.'s protocol

Fig. 2 shows the conceptual steps for $\mathcal{A}$'s attack with two steps each for the key agreement phase and the forward secrecy phase. This means that Ghazo et al.'s protocol has lack of forward secrecy.

Lack of unlinkability: Unlinkability is a crucial property that ensures an adversary $\mathcal{A}$ cannot correlate multiple sessions involving the same parties. In Ghazo et al.'s AKA protocol, $\mathcal{A}$ under the CK model can exploit the lack of unlinkability between sessions. If $\mathcal{A}$ has eavesdropping session messages $m_3=(m_1, m_2, T_{m3}, h(T_{m3}||k_s))$, $m_4=(T_{ACK}, h(T_{ACK}||k_s))$, $m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{i-1}))$ and $m_{fs3}=(T_{ACK}, h(T_{ACK}||k_s))$, they can link multiple communication sessions between C_i and $IIoT_j$. This occurs because Ghazo et al.'s protocol does not use a session challenge/response mechanism for the key agreement phase, allowing $\mathcal{A}$ to correlate distinct interactions. The important session message has the two parameters $m_1=k_s\oplus ID_i$ and $m_2= k_s\oplus ID_j$, which does not provide session freshness. The ability to link sessions breaks the unlinkability property, which should ensure that $\mathcal{A}$ cannot correlate two sessions involving the same network entity without explicit access to the session keys. Without this unlinkability, $\mathcal{A}$ can trace the same entity across multiple communication exchanges, thus compromising user privacy.

4. Authenticated Key Agreement Protocol with Forward Secrecy

This section explores the development of a new authentication and key agreement protocol designed specifically for the unique challenges of smart agriculture networks. It highlights the importance of forward secrecy in ensuring the confidentiality and integrity of data exchanged among smart devices, enhancing both the security and reliability of these critical agricultural systems.

4.1. Smart Agriculture Network Environments

In recent years, incorporating CPS into agriculture has driven significant changes, enabling farming methods that are more efficient, sustainable, and guided by data analysis. CPSs combine computational resources with physical processes, enabling real-time monitoring, control, and optimization of agricultural activities. This integration is further enhanced by the IIoT, which connects a network of smart devices, sensors, and machines to gather critical data, providing farmers with actionable insights. The focus of smart agriculture is not only on automating processes but also on empowering users-farmers and agricultural managers-with the tools to make informed decisions. By leveraging IIoT, farmers can monitor soil conditions, crop health, irrigation systems, and weather patterns, ensuring timely interventions that improve productivity and reduce resource waste. This technology allows for the creation of a responsive, dynamic farming environment that adapts to changing conditions in real-time, thereby increasing crop yields and fostering sustainability. As these technologies advance, they hold increasing promise to transform agricultural methods, fostering a more intelligent and interconnected ecosystem that supports both end-users and environmental sustainability.

4.2. New Authenticated Key Agreement Protocol

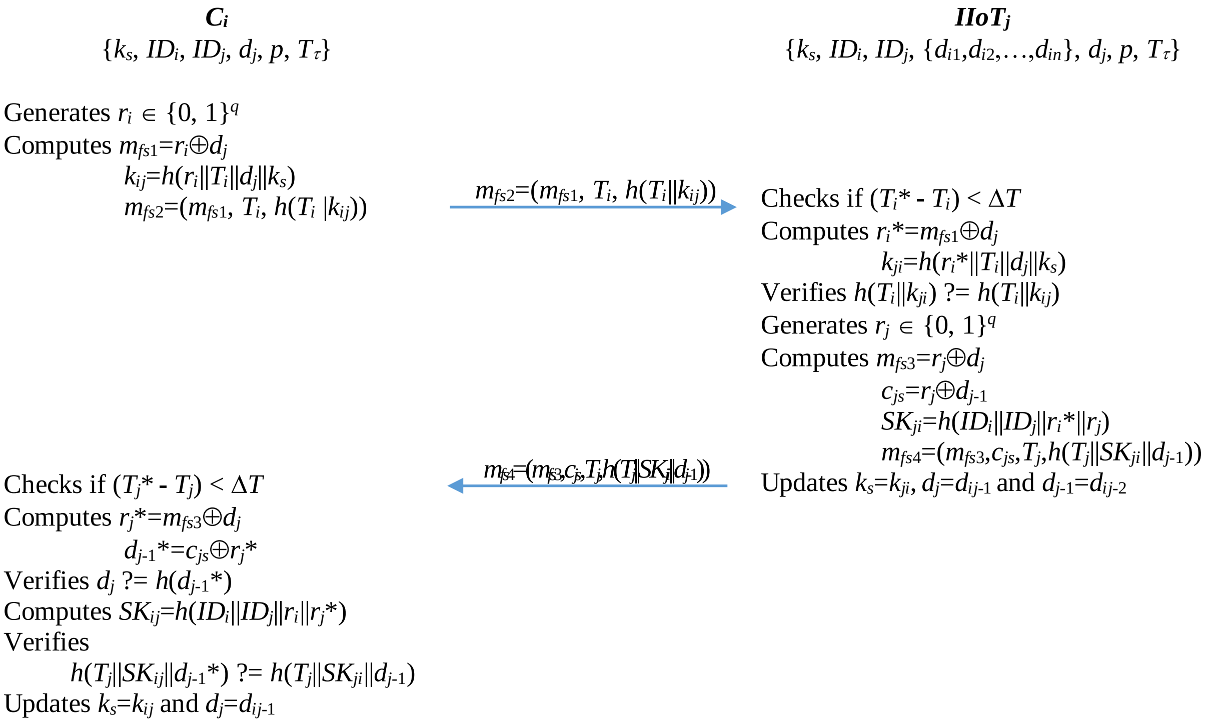
This section proposes a new AKA protocol as a solution for Ghazo et al.'s protocol by adopting the forward secrecy framework proposed by Chiphiko et al. in [13], which uses a hash chain for each session's credential.

Registration: TA performs the registration process for C_i and $IIoT_j$.

R1: For this, TA creates a primary confidential key K_m. TA assigns distinct identifiers to C_i and $IIoT_j$, denoted as ID_i and ID_j, respectively. TA also generates a prime number p.

R2: TA randomly generates two integers r_0 and $r_i \in \{0, 1\}^q$ and computes n hash values as $d_{i1} = h(r_i)$, $d_{i2} = h(d_1)$, …, and $d_{in} = h(d_{in\text{-}1})$. Once TA generates all these parameters, it sends $\{K_m, ID_i, ID_j, r_0, d_j{=}d_{in}, p, T_\tau\}$ and $\{K_m, ID_i, ID_j, r_0, \{d_{i1}, d_{i2}, \dots, d_{in},\} T_\tau\}$ to C_i and $IIoT_j$, respectively, where T_τ is the current timestamp.

R3: Upon receiving the parameters, $IIoT_j$ stores $\{K_m, ID_i, ID_j, r_0, \{d_{i1}, d_{i2}, \dots, d_{in},\} T_\tau\}$ on physical-unclonable-function PUF in [26] and sets counter j to n and stores it in PUF. Both C_i and $IIoT_j$ compute $k_s{=}h(T_\tau||ID_i||ID_i||K_m||r_0)$ and store it in the PUF.



C_i
$\{k_s, ID_i, ID_j, d_j, p, T_\tau\}$

$IIoT_j$
$\{k_s, ID_i, ID_j, \{d_{i1},d_{i2},\ldots,d_{in}\}, d_j, p, T_\tau\}$

Generates $r_i \in \{0, 1\}^q$
Computes $m_{fs1}=r_i \oplus d_j$
$k_{ij}=h(r_i||T_i||d_j||k_s)$
$m_{fs2}=(m_{fs1}, T_i, h(T_i\ |k_{ij}))$

$m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{ij}))$

Checks if $(T_i^* - T_i) < \Delta T$
Computes $r_i^*=m_{fs1}\oplus d_j$
$k_{ji}=h(r_i^*||T_i||d_j||k_s)$
Verifies $h(T_i||k_{ji})$?= $h(T_i||k_{ij})$
Generates $r_j \in \{0, 1\}^q$
Computes $m_{fs3}=r_j\oplus d_j$
$c_{js}=r_j\oplus d_{j-1}$
$SK_{ji}=h(ID_i||ID_j||r_i^*||r_j)$
$m_{fs4}=(m_{fs3},c_{js},T_j,h(T_j||SK_{ji}||d_{j-1}))$
Updates $k_s=k_{ji}$, $d_j=d_{ij-1}$ and $d_{j-1}=d_{ij-2}$

$m_{fs4}=(m_{fs3},c_{js},T_j,h(T_j||SK_{ji}||d_{j-1}))$

Checks if $(T_j^* - T_j) < \Delta T$
Computes $r_j^*=m_{fs3}\oplus d_j$
$d_{j-1}^*=c_{js}\oplus r_j^*$
Verifies d_j ?= $h(d_{j-1}^*)$
Computes $SK_{ij}=h(ID_i||ID_j||r_i||r_j^*)$
Verifies
$h(T_j||SK_{ij}||d_{j-1}^*)$?= $h(T_j||SK_{ji}||d_{j-1})$
Updates $k_s=k_{ij}$ and $d_j=d_{ij-1}$

Figure 3. AKA phase of the proposed protocol

AKA: The AKA phase is carried out between C_i and $IIoT_j$ as shown in Fig. 3.

A1: C_i randomly generates an integer $r_i \in \{0, 1\}^q$ and computes $m_{fs1}=r_i\oplus d_j$ and $k_{si}=h(r_i||T_i||d_j||k_s)$ and sends $m_{fs2}=(m_{fs1}, T_i, h(T_i\ ||k_{si}))$ to $IIoT_j$.

A2: Once $IIoT_j$ receives m_{fs2}, it checks if $(T_i^* - T_i) < \Delta T$ and computes $r_i^*=m_{fs1}\oplus d_j$ and $k_{ji}=h(r_i^*||T_i||d_j||k_s)$. $IIoT_j$ verifies whether $h(T_i||k_{ji})$ matches $h(T_i||k_{ij})$. Only if the validation is successful, $IIoT_j$ generates $r_j \in \{0, 1\}^q$, computes $m_{fs3}=r_j\oplus d_j$, $c_{js}=r_j\oplus d_{j-1}$ and $SK_{ji}=h(ID_i||ID_j||r_i^*||r_j)$, updates $k_s=k_{ji}$, $d_j=d_{ij-1}$ and $d_{j-1}=d_{ij-2}$, and sends $m_{fs4}=(m_{fs3}, c_{js}, T_j, h(T_j||SK_{ji}||d_{j-1}))$ to C_i.

A3: Once C_i receives m_{fs4}, it checks if $(T_j^* - T_j) < \Delta T$ and computes $r_j^*=m_{fs3}\oplus d_j$ and $d_{j-1}^*=c_{js}\oplus r_j^*$. C_i verifies whether d_j matches $h(d_{j-1}^*)$. Only if the validation is successful, C_i computes $SK_{ij}=h(ID_i||ID_j||r_i||r_j^*)$ and checks if $h(T_j||SK_{ij}||d_{j-1}^*)$ matches $h(T_j||SK_{ji}||d_{j-1})$. Only after the successful validation, C_i updates $k_s=k_{ij}$ and $d_j=d_{ij-1}$.

Finally, C_i and $IIoT_j$ can access successfully authenticate each other and agree on the same session key $SK_{ij} = SK_{ji}$ with forward secrecy property.

5. Security Analysis

The proposed protocol undergoes a comprehensive security assessment to verify its strength and ability to withstand diverse threats within IoT environments. This analysis focuses on several critical security properties, including forward secrecy, anonymity, unlinkability, mutual authentication, message integrity and session key agreement. Each of these factors is rigorously analyzed to confirm that the protocol offers robust protection against numerous possible attacks, including impersonation, man-in-the-middle, replay, eavesdropping, data modification, and known session key compromises. By employing advanced cryptographic techniques and protocols, the proposed protocol not only ensures the confidentiality and integrity of communications but also strengthens the trust and privacy of users and devices interacting within IIoT ecosystems. The subsequent sections present a detailed examination of the security characteristics incorporated within the proposed protocol.

Forward secrecy: It is maintained in the proposed protocol by generating new session keys with each session based on session-dependent random numbers. The session key $SK_{ij} = SK_{ji}$ is computed using the formula $SK_{ij} = SK_{ji} = h(ID_i||ID_j||r_i||r_j)$, where r_i and r_j are only withdrawable based on the session-dependent secret key derived from the hash chain. This design ensures that once a session concludes, the previous session key becomes unrecoverable, even if the current session key is compromised. The one-way hash function ensures that attackers cannot recompute past session keys, thus preserving forward secrecy. As a result, the proposed protocol guarantees that previous communication sessions remain secure.

Anonymity: It is a crucial security property for ensuring that users and devices in the IIoT ecosystem are not easily identifiable or tracked across multiple interactions. The proposed protocol integrates robust mechanisms to preserve anonymity by preventing the disclosure of any user or device identifiers during the AKA process. In this framework, identity information remains hidden from unauthorized parties. During the AKA process, the identifiers, ID_i and ID_i are never directly exposed. This makes it difficult for external attackers or adversaries to correlate messages and identify participants in the system. Consequently, the proposed protocol provides a strong layer of protection for user and device anonymity, ensuring that entities can interact securely without revealing their true identities.

Unlinkability: This security feature is crucial for preventing attackers from linking multiple sessions or interactions involving the same parties. The proposed protocol provides strong unlinkability by employing unique session keys and cryptographic processes that prevent the association of multiple sessions with the same user or device. During the AKA process, the session keys are updated dynamically after each interaction, and they are not derived from any persistent or easily guessable information. This dynamic nature of session key generation ensures that even if an attacker intercepts or observes several communication sessions, they cannot link those sessions back to a particular entity. These identifiers are not tied to the users or devices in any persistent manner. As a result, the proposed protocol effectively ensures that adversaries cannot correlate different sessions or trace a user's behavior across multiple interactions, enhancing both privacy and security in IIoT systems.

Mutual authentication: Mutual authentication within the proposed protocol forms a crucial element of its security infrastructure. This sophisticated process ensures that both IIoT devices and users authenticate each other, establishing a trustworthy environment for communication. During AKA, $IIoT_j$ verifies message $m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{ij}))$ by computing values as $m_{fs1}=r_i \oplus d_j$ and $k_{ij}=h(r_i||T_i||d_j||k_s)$. These computations yield distinct combinations with the previously established synched values that serve as the foundation for the verification process. In return, C_i performs a similar authentication by calculating $r_j^*=m_{fs3} \oplus d_j$ and $d_{j\text{-}1}^*=c_{js} \oplus r_j^*$ using message $m_{fs4}=(m_{fs3}, c_{js}, T_j, h(T_j||SK_{ji}||d_{j\text{-}1}))$ and checks for validates the current session's authentication parameter with the previous one by checking d_j matches $h(d_{j\text{-}1}^*)$. This reciprocal authentication not only guarantees secure validation but also strengthens the trust between interacting IIoT entities and users. As a result, the proposed protocol goes beyond mutual authentication, elevating it to a key security feature that ensures a robust, trusted communication environment in IIoT settings.

Message integrity: Ensuring the integrity of transmitted messages is a fundamental principle of secure communication. The proposed protocol prioritizes the protection of data integrity by securing information exchanges between IIoT devices and users. Shared session keys are vital in serving as cryptographic mechanisms that help identify and block any unauthorized alterations to transmitted messages. The system employs hash functions to ensure data remains untampered with during transmission, and any discrepancies trigger a verification failure alert, protecting against potential security threats. This emphasis on maintaining the unaltered state of messages reinforces the proposed protocol's commitment to secure, reliable communication in IIoT environments.

Session key agreement: A key aspect of the proposed protocol's secrecy is the exclusive usage of hash chain values $\{d_{i1}, d_{i2}, \ldots, d_{in}\}$ and d_j sharing between C_i and $IIoT_j$. During the registration phase, TA generates the hash chain values, using the formula $d_{i1} = h(r_i)$, $d_{i2} = h(d_1)$, …, and $d_{in} = h(d_{in\text{-}1})$, which is then used as a session dependent credential between C_i and $IIoT_j$. Concurrently, C_i computes the session key $SK_{ij}{=}h(ID_i||ID_j||r_i||r_j{*})$ independently and verifies its accuracy by checking message $m_{fs4}{=}(m_{fs3}, c_{js}, T_j, h(T_j||SK_{ji}||d_{j\text{-}1}))$. This elaborate and secure process guarantees that both parties agree on the same session key, enhancing the security of the communication. The session key agreement mechanism establishes a secure foundation for interactions, adapting to the dynamic security needs of IIoT ecosystems and countering evolving threats.

Resistance against various attacks: The proposed protocol is thoroughly analyzed for its resilience against various types of attacks, including impersonation, man-in-the-middle, replay, eavesdropping, modification, and known key attacks. [Impersonation attack] If an attacker attempts to impersonate C_i and $IIoT_j$, they must acquire the session dependent hash chain values d_j and $d_{j\text{-}1}$. However, these secrets are securely stored in PUF, preventing unauthorized access. Additionally, the one-wayness of the hash function makes it impossible for attackers to derive the session key from intercepted messages, ensuring robust protection against impersonation. ***Man-in-the-middle-attack***: The proposed protocol prevents man-in-the-middle attacks by ensuring mutual authentication. Even if an attacker intercepts and modifies messages, $m_{fs2}{=}(m_{fs1}, T_i, h(T_i||k_{ij}))$ and $m_{fs4}{=}(m_{fs3}, c_{js}, T_j, h(T_j||SK_{ji}||d_{j\text{-}1}))$, they cannot generate valid messages that pass the verification process, making this type of attack ineffective. ***Replay attack***: By integrating timestamps and random numbers into the exchanged messages, the protocol ensures that messages are fresh. If a message containing an expired timestamp and random number is replayed, the system detects this by verifying the timestamps and promptly rejects any that are no longer valid. ***Eavesdropping***: The proposed protocol effectively counters eavesdropping by securely storing the session key in PUF and using session specific secret credentials based on the hash chain values, which are based on the irreversible hash functions. As a result, even if messages are intercepted, attackers cannot extract meaningful information. ***Modification attack***: Any unauthorized modification of messages is easily detectable, as the recipient can verify the integrity of the message using the session specific hash chain credentials. If a message has been altered, the verification process fails, thus protecting the communication from tampering. ***Known key attack***: The proposed protocol defends against known key attacks by updating the session specific secret at the end of each communication session. Since each new session credential is used independently, attackers cannot use a previously obtained session secret to derive subsequent ones, ensuring continued security.

Our rigorous evaluation of the protocol's security features reveals its robustness and advanced design in safeguarding IIoT communications. The protocol employs a multifaceted strategy to counter various potential security risks. This comprehensive approach ensures that data transmissions within IIoT ecosystems maintain their confidentiality, remain unaltered during transit, and can be verified as originating from legitimate sources. The protocol's design demonstrates a strong commitment to addressing the complex security challenges inherent in

IIoT environments. By incorporating mechanisms such as forward secrecy, anonymity, unlinkability, mutual authentication, message integrity, and session key agreement, the proposed protocol offers a robust solution for securing CPSs.

6. Performance Analysis

IIoT devices are often constrained by limited computational and communicational resources. To evaluate the efficiency of the proposed protocol, it is crucial to examine its requirements for computation and communication. This section compares the proposed protocol with three alternative protocols [20, 22, 24], focusing on key metrics such as computational complexity and communication costs. Table 2 presents the performance evaluation results only focused on the AKA phase. When analyzing communication costs, the AKA phase is emphasized as the most significant component, in contrast to the registration phase, which occurs only once and is performed on a computer, primarily for registering C_i and $IIoT_j$ with the network.

Computational analysis: The computational costs of the AKA phases were evaluated by counting the number of essential operations, such as scalar multiplication, point addition, random number generation, hash function and fuzzy extraction. For this evaluation, we used an Intel Xeon CPU E5-2630, 1 GB of RAM, and Ubuntu 14.04 with the MIRACL library [27-29]. The execution times for operations are 2.3 ms, 0.01 ms, 0.24 ms, 0.03 ms and 2.226 ms for T_m, T_a, T_r, $T_{h,}$ and T_f, respectively. The results of the execution time analysis are presented in Table 2. Note that the bitwise XOR operation was excluded from the timing analysis, as it typically completes in a single cycle. The proposed protocol requires three hash operations more than Ghazo et al.'s protocol to provide forward secrecy.

Table 2. Performance comparison.

Protocol	Computation cost	Communication cost	Number of messages
Das et al. in [20]	$21T_h+1T_f$ (2.856 ms)	$9\|h\|+3\|TS\|$ (2,481 bits)	3
Srinivas et al. in [22]	$9T_m+2T_a+24T_h+1T_f$ (23.666 ms)	$3\|M\|+9\|h\|+3\|TS\|$ (2,961 bits)	3
Ghazo et al. in [24]	$6T_h+2T_r$ (0.66 ms)	$3\|h\|+2\|R\|$ (1,280 bits)	2
Proposed protocol	$9T_h+2T_r$ (0.75 ms)	$3\|h\|+2\|R\|+2\|TS\|$ (1,344 bits)	2

Scalar multiplication (T_m), point addition (T_a), random number generation (T_r), hash function (T_h), fuzzy extraction (T_f)

Communicational analysis: To match the security level of the 1024-bit RSA algorithm, the lengths of various components are 160 bits for scalar multiplication $|M|$ and point addition $|A|$, 256 bits for random number $|R|$, hash function $|h|$ and fuzzy extraction $|F|$, and 32 bits for identifier $|I|$ and timestamp $|TS|$. During the AKA phase of the proposed protocol, two messages are required as $m_{fs2}=(m_{fs1}, T_i, h(T_i||k_{ij}))$ and $m_{fs4}=(m_{fs3} ,c_{js}, T_j, h(T_j||SK_{ji}||d_{j-1}))$. A comparison of these communication costs with other state-of-the-art protocols is shown in Table 2. The proposed protocol demonstrates similar performance in terms of overall communication costs to Ghazo et al.'s protocol.

7. CONCLUSIONS

In this paper, we addressed a critical gap in the security of CPS by analyzing the lack of forward secrecy in existing authentication and key agreement protocols. Through comprehensive cryptanalysis of widely adopted protocols, we demonstrated that while these systems provide secure key exchange and authentication, they fail to protect past communications in the event of a long-term key compromise. This weakness poses a considerable threat in CPS environments, as it is crucial that sensitive information remains protected over time, even in scenarios where an attacker obtains private keys.

To address this issue, we proposed a new authenticated key agreement protocol that integrates forward secrecy as a core feature. Our approach guarantees that, even in the event of a device's long-term key being exposed, the privacy of individual communication sessions is preserved, thereby enhancing the overall security of CPS applications. Furthermore, we designed the protocol to be lightweight and efficient, ensuring its suitability for resource-constrained IIoT devices typically deployed in CPS environments. The evaluation of our proposed protocol shows that it successfully balances the need for strong security with the practical requirements of scalability and computational efficiency. By incorporating forward secrecy into the protocol, we provide a robust solution that enhances the privacy and integrity of CPS communications while minimizing overhead. This work lays the foundation for future advancements in secure IIoT communication protocols, setting a new standard for resilience and security in critical applications. The proposed protocol presents an effective strategy for overcoming the shortcomings of existing methods, ensuring sustained security for CPS as new threats emerge.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Briatore, F. & Braggio, M., (2024) "Resilience and Sustainability plants improvement through Maintenance 4.0: IoT, Digital Twin and CPS framework and implementation roadmap," *IFAC-PapersOnline*, Vol. 58, No. 8, pp. 365-370.

[2] Nandhini, R. S. & Lakshmanan, R., (2022) "A Review of the Integration of Cyber-Physical System and Internet of Things," *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 4, pp. 459-465.

[3] Chui, K. T., Gupta, B. B., Liu, J., Arya, V., Nedjah, N., Almomani, A. & Chaurasia, P., (2023) "A Survey of Internet of Things and Cyber-Physical Systems: Standards, Algorithms, Applications, Security, Challenges, and Future Directions," *Information*, Vol. 14, No. 7, 388.

[4] Xu, H., Yu, W., Griffith D. & Golmie, N., (2018) "A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective," *IEEE Access*, Vol. 6, pp. 78238-78259.

[5] Chehida, S., Rutten, E., Giraud G. & Mocanu, S., (2024) "A model-based approach for self-adaptive security in CPS: Application to smart grids," *Journal of Systems Architecture*, Vol. 150, 103118.

[6] Agarwal, N. & Kumar, R., (2022) "Security Perspective Analysis of Industrial Cyber Physical Systems (I-CPS): A Decade-wide Survey," *ISA Transactions*, Vol. 130, pp. 10-24.

[7] Chiphiko, B. A., Kim, H., Ali, P. & Eneya, L., (2025) "Forward Secrecy Attack on Privacy Preserving Machine Authenticated Key Agreement for Internet of Thing," *Archives of Advanced Engineering Science*, Vol. 3, No. 1, pp. 29-34.

[8] Ju, Y., Yang, M., Chakraborty, C., Liu, L., Pei, Q., Xiao, M. & Yu, K., (2024) "Reliability-Security Tradeoff Analysis in mmWave Ad Hoc-based CPS," *ACM Transactions on Sensor Networks,* Vol. 20, No. 2, pp. 1-23.

[9] Yang, Z., He, J., Tian, Y. & Zhou, J., (2019) "Faster Authenticated Key Agreement with Perfect Forward Secrecy for Industrial Internet-of-Things," *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 10, pp. 6584-6596.

[10] Ge, M., Kumari, S. & Chen, C., (2022) "AuthPFS: A Method to Verify Perfect Forward Secrecy in Authentication Protocols," *Journal of Network Intelligence*, Vol. 7, No. 3, pp. 734-750.

[11] Thesen, E. K., (2024) *Nearly Tight Diffie-Hellman-Based Key Exchange*, Norwegian University of Science and Technology, Bachelor's thesis.

[12] Kapito, B., Nyirenda, M. & Kim, H., (2021) "Privacy-Preserving Machine Authenticated Key Agreement for Internet of Things," *International Journal of Computer Networks & Communications*, Vol. 13, No. 2, pp. 99-120.

[13] Chiphiko, B. A. & Kim, H., (2023) "Machine To Machine Authenticated Key Agreement with Forward Secrecy for Internet of Things," *International Journal of Computer Networks & Communications*, Vol. 15, No. 6, pp. 27-53.

[14] Preneel, B., (1994) "Cryptographic hash functions," *European Transactions on Telecommunications*, Vol. 5, No. 4, pp. 431-448.

[15] Mansoor, K., Ghani, A., Chaudhry, S. A., Shamshirband, S., Ghayyur, S. A. K. & Mosavi, A., (2019) "Securing IoT-Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography," *Sensors*, Vol. 19, No. 21, 4752.

[16] Braeken, A., (2020) "Highly efficient symmetric key based authentication and key agreement protocol using Keccak," *Sensors*, Vol. 20, No. 8, 2160.

[17] Braeken, A., (2020) "Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability, *Computer Networks*, Vol. 181, 107424.

[18] Ali, Z., Chaudhry, S. A., Ramzan, M. S. & Al-Turjman, F., (2020) "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, Vol. 8, pp. 43711-43724.

[19] Ali, Z., Hussain, S., Rehman, R. H. U., Munshi, A., Liaqat, M., Kumar, N. & Chaudhry, S. A., (2020) "ITSSAKE-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments," *IEEE Access*, Vol. 8, pp. 107993-108003.

[20] Das, A. K., Wazid, M., Kumar, N., Vasilakos, A. V. & Rodrigues, J. J. P. C., (2018) "Biometrics-Based Privacy-Preserving User Authentication Scheme for Cloud-Based Industrial Internet of Things Deployment," *IEEE Internet of Things Journal*, Vol. 5, pp. 4900-4913.

[21] Hussain, S. & Chaudhry, S. A., (2019) "Comments on 'biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment'," *IEEE Internet of Things Journal*, Vo. 6, No. 6, pp. 10936–10940.

[22] Srinivas, J., Das, A. K., Wazid, M. & Vasilakos, A. V., (2021) "Designing Secure User Authentication Protocol for Big Data Collection in IoT-Based Intelligent Transportation System," *IEEE Internet of Things Journal*, Vol. 8, pp. 7727-7744.

[23] Tanveer, M., Khan, A. U., Kumar, N. & Hassan, M. M., (2022) "RAMP-IoD: A Robust Authenticated Key Management Protocol for the Internet of Drones," *IEEE Internet of Things Journal*, Vol. 9, No. 2, pp. 1339-1353.

[24] Ghazo, A. T. A., Mallouh, M. A., Alajlouni, A. & Almalkawi, I. T., (2025) "Securing Cyber Physical Systems: Lightweight Industrial Internet of Things Authentication (LI2A) for Critical Infrastructure and Manufacturing," *Applied System Innovation*, Vol. 8, 11.

[25] Quang, D., Martini, B. & Raymond, C. K., (2019) "The role of the adversary model in applied security research," *Computers & Security*, Vol. 81, pp. 156-181.

[26] Ma, H., Wang, C., Xu, G., Cao, Q., Xu, G.& Duan, L., (2023) "Anonymous Authentication Protocol Based on Physical Unclonable Function and Elliptic Curve Cryptography for Smart Grid," *IEEE Systems Journal*, Vol. 17, pp. 6425–6436.

[27] SDK, M. C., (2020) *MIRACL Cryptographic SDK: Multiprecision Integer and Rational Arithmetic Cryptographic Library*, Available online: https://github.com/miracl/MIRACL (accessed on 31 March 2025).

[28] Mlato, S., Gabriel, Y., Chirwa, P. & Kim, H., (2024) "Privacy-Preserving User Authentication Scheme based on Fuzzy Commitment for Multi-Server Environment," *International Journal of Computer Networks & Communications*, Vol. 16, No. 2, pp. 87-106.

[29] Bebin, J. T. & Misbha, D. D., (2025) "Elliptic Curve Cryptography Algorithm with Recurrent Neural Networks for Attack Detection in Industrial IoT," *International Journal of Computer Networks & Communications*, Vol. 17, No. 1, pp. 59-79.

AUTHORS

Sung-Woon Lee is a professor at the Department of Information System and Security, Tongmyong University, Korea. He received the B.S. and M.S. degrees in Computer Science from Chonnam National University, Korea in 1994 and 1996, respectively, and the Ph.D. degree in Computer Engineering from Kyungpook National University, Korea, in 2005. He was with the Korea Information System as a researcher, Korea, from 1996 to 200. His research interests include cryptography, network security and security protocol. He can be contacted at email: staroun@tu.ac.kr.

Hyunsung Kim* received the M.Sc. and Ph.D. degrees in computer engineering from Kyungpook National University, Korea, in 1998 and 2002, respectively. He is a Professor at the School of Computer Science, Kyungil University, Korea from 2012. Furthermore, he is currently a visiting professor at the Department of Mathematical Sciences, Chancellor College, University of Malawi, Malawi from 2015. He also was a visiting researcher at Dublin City University in 2009. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2012 with the Department of Computer Engineering, Kyungil University. His research interests include cryptography, VLSI, authentication technologies, network security, ubiquitous computing security, and security protocol.
(*Corresponding Author)