IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2020, accepted March 31, 2020, date of publication April 6, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985719

A Privacy-Preserving Authentication, Authorization, and Key Agreement Scheme for Wireless Sensor Networks in 5G-Integrated Internet of Things

SOOYEON SHIN, (Member, IEEE), AND TAEKYOUNG KWON, (Member, IEEE)
Graduate School of Information, Yonsei University, Seoul 03722, South Korea

Corresponding author: Taekyoung Kwon (taekyoung@yonsei.ac.kr)

This work was supported as part of Military Crypto Research Center (UD170109ED) funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

ABSTRACT Wireless sensor networks (WSNs) have played an important role in the Internet of Things (IoT), and the 5G network is being considered as a major candidate for IoT's communication network with the advent of 5G commercialization. The potential of integrating WSNs and 5G in the IoT is expected to allow IoT to penetrate deeply into our daily lives and to provide various services that are convenient, but at the same time, it also brings new security threats. From this aspect, user authentication and key agreement are essential for secure end-to-end communication. As IoT devices, including sensors, collect and process more and more personal information, both anonymous authentication and authorization are also required to protect the privacy and to prevent anyone without privileges from accessing private data. Recently, Adavoudi-Jolfaei *et al.* proposed an anonymous three-factor authentication and access control scheme for real-time applications in WSNs. However, we found that this scheme does not provide sensor-node anonymity and suffers from user collusion and desynchronization attacks. In this paper, we introduce a system architecture by considering the integration of WSNs and 5G for IoT. Based on a cryptanalysis of Adavoudi-Jolfaei *et al.*'s scheme and the system architecture, we propose an elliptic curve cryptography (ECC)-based privacy-preserving authentication, authorization, and key agreement scheme for WSNs in 5G-integrated IoT. We conduct a formal and informal security analysis in order to demonstrate that the proposed scheme withstands various security attacks and guarantees all desired security features, overcoming the drawbacks of Adavoudi-Jolfaei *et al.*'s scheme. Finally, a performance and comparative analysis with the related schemes indicate that the proposed scheme is both efficient and more secure.

INDEX TERMS Three-factor authentication, authorization, key agreement, elliptic curve cryptography, anonymity, untraceability, 5G network, wireless sensor networks, the Internet of Things.

I. INTRODUCTION

The Internet of Things (IoT) is an intelligent technology and service that connects all things including sensors, smartphones, and home appliances to communicate information between people and things based on the Internet. Recently, various IoT applications have made it possible for users, through linkage with smart devices, to access, use, and process information collected from sensors. From this aspect, the smartphone, the most common device that is steadily increasing in performance, plays an important interface role in allowing users to access and control other devices in the IoT through Wi-Fi and cellular networks. 5G is becoming an active candidate for various IoT applications such as the smart home, smart city, smart health, and smart grid [1]. Owing to the commercialization of 5G, one of the cellular networks, IoT has penetrated into everyday life.

The wireless sensor network (WSN) is one of the core elements of the IoT and is responsible for collecting and delivering the physical phenomena and information using a number of heterogeneous and resource-constrained sensors. Therefore, the integration of WSN and 5G will be a

The associate editor coordinating the review of this manuscript and approving it for publication was Ilsun You.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

key driver for successful IoT deployment. With WSNs and 5G-integrated IoT, more sensors and smart devices will surround us and reach deeper into our private lives. This increases connectivity and provides convenient services to us, but at the same time, it increases the number of attack surfaces. To protect IoT devices and their data and provide secure communication, user authentication and key establishment are the most basic security requirements.

IoT devices including sensors have various types of data, and the collected data in several IoT applications are privacy-sensitive. For example, personal health information collected from wearable and implanted medical devices and private information collected from home sensors are privacy-sensitive data. If these types of data are leaked or controlled by malicious people, they may pose serious threats, and these threats may be linked directly to human life. Anonymity with untraceability is a representative technique for stronger privacy preservation. Anonymity hides the identities of participants, including users, from third parties so that they do not know who accesses data at certain points in time. Untraceability disallows an adversary who wants to trace different sessions of a particular user from publicly exchanged messages. The authorization and access control mechanisms grant different access rights according to the data's importance and privacy sensitivity, and verify whether a user has the corresponding privilege to access data. Therefore, authorization and access control mechanisms and privacy-preserving techniques are also essential for securing the IoT.

A. RELATED WORKS

Since Das *et al.* introduced a two-factor user authentication scheme for WSNs [2], a large number of two-factor user authentication schemes using passwords and smart cards have been proposed [3]–[8]. To address the security vulnerabilities associated with two-factor user authentication schemes and to improve their security strength, three-factor authentication, with biometrics as the third factor, has attracted attention from many researchers in recent years [9]–[16].

Park *et al.* proposed a security-enhanced authentication and key agreement scheme to overcome the security weaknesses of Chang *et al.*'s scheme [6] by using biometric information and an elliptic curve cryptogrphy (ECC) [10]. However, Wang *et al.* [17] and Maurya and Sastry [12] revealed that Park *et al.*'s scheme has security flaws. Moon *et al.* showed how an adversary can impersonate a legitimate user or a sensor node, and proposed an improved authentication scheme [18]. Das proposed a novel biometric-based user authentication scheme suitable for WSNs [19]. Unfortunately, in the same year, Maurya *et al.* pointed out that these two schemes including Park *et al.*'s scheme are insecure against various security attacks. Instead, Maurya *et al.* proposed a fuzzy extractor and ECC-based efficient authenticated session key establishment protocol for WSNs and IoT [12].

Amin *et al.* proposed a new secure three-factor authentication scheme that claimed to be secure against all known security attacks [11], but Jiang *et al.* found that Amin *et al.*'s scheme has security drawbacks [13]. Jiang *et al.* then proposed a three-factor authentication and key agreement scheme based on the Rabin cryptosystem for Internet-integrated WSNs. Wadiz *et al.* proposed a secure and lightweight three-factor authenticated key management scheme for the hierarchical IoT network as a special kind of generic IoT network [14]. All the abovementioned schemes have evolved by identifying and solving the security problems in the previous systems. However, they still have security drawbacks and do not support authorization to access control, which is one of the essential security requirements in WSNs for IoT.

Adavoudi-Jolfaei *et al.* [15] pointed out a security vulnerability in Gope and Hwang's [7] two-factor authentication protocol for WSNs. To address this vulnerability, Adavoudi-Jolfaei *et al.* devised an enhanced scheme by employing biometrics with a fuzzy extractor and by providing access control as an additional desired security property for WSNs. They proved their scheme was secure against various attacks. However, as illustrated in Section II-B, we found that Adavoudi-Jolfaei *et al.*'s scheme still has several security flaws. Their scheme does not provide sensor node anonymity, and it is vulnerable to user collusion attacks in which malicious users collude with each other in order to access data that is inaccessible with their own privileges. In addition, it is also vulnerable to desynchronization attacks in which an attacker breaks the synchronization of the secret values that are shared between a server and a user and updated on a per session basis, thereby preventing the server from authenticating a legitimate user's credentials [20].

B. RESEARCH CONTRIBUTIONS

As discussed in Section I-A, the existing schemes for authentication and key establishment for WSNs and IoT still suffer from security attacks and fail to guarantee all desirable security features. In particular, most of them do not support authorization, another desirable security requirement. The contributions of our research to overcoming these drawbacks are as follows:

- We analyze the recent lightweight and anonymous three-factor authentication and access control scheme of Adavoudi-Jolfaei *et al.* [15]. We show that their scheme does not provide sensor node anonymity and is vulnerable to user collusion and desynchronization attacks.
- We introduce a system architecture suitable for WSNs in 5G-integrated IoT. Based on the system architecture, we design an ECC-based privacy-preserving authentication, authorization, and key agreement scheme. The proposed scheme provides three-factor user authentication and overcomes the security weaknesses of the Adavoudi-Jolfaei *et al.*'s scheme. In addition, the proposed scheme not only satisfies various security features, including authorization, but also withstands all known attacks.

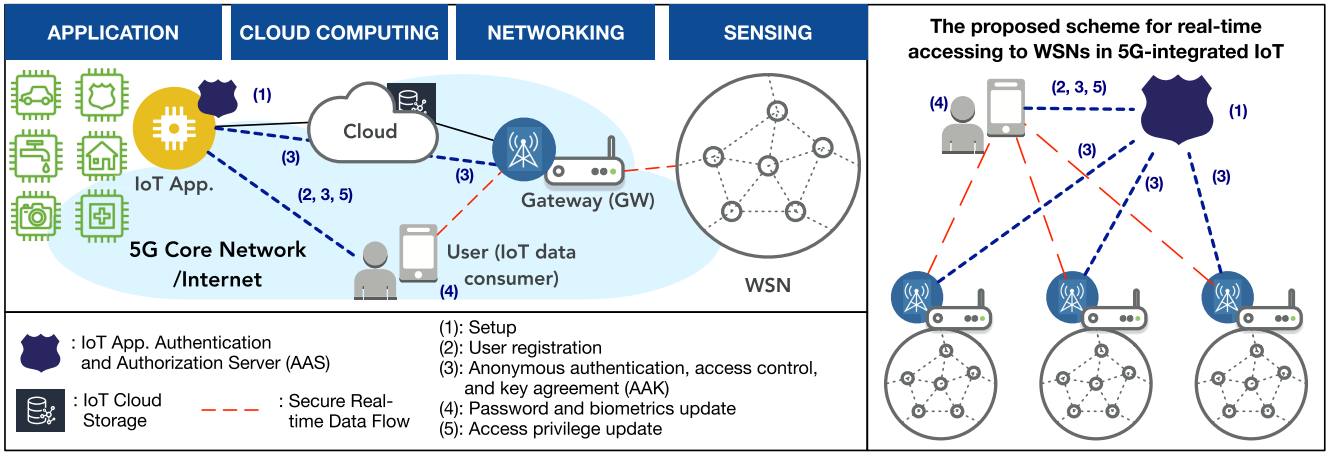

FIGURE 1. System architecture for WSNs in 5G-integrated IoT.

- We formally verify the security of the proposed scheme using both the widely used Burrows-Abadi-Needham (BAN) logic and a robust security verification tool, Automated Validation of Internet Security Protocols (AVISPA). We also informally analyze the security of the proposed scheme to show that it can satisfy the required security features and resist various attacks. We then compare the security of the proposed scheme with those of related schemes in terms of security features.
- Through a performance evaluation, we compare the performance of the proposed scheme with those of related schemes in terms of computation and communication costs.

The remainder of the paper is organized as follows: Section II briefly reviews Adavoudi-Jolfaei *et al.*'s scheme and demonstrates its security weaknesses. Section III describes the details of the proposed scheme. Section IV conducts a formal and informal security analysis of the proposed scheme. Section V presents a performance evaluation of the proposed scheme and compares the performance with related schemes. Section VI concludes the paper.

C. PRELIMINARIES

This section introduces the necessary mathematical preliminaries and system architecture for the proposed scheme.

1) FUZZY EXTRACTOR

In recent years, the fuzzy extractor technique has been a useful tool that is widely accepted for biometric authentication [9], [15], [21]. The fuzzy extractor extracts biometric information as a uniformly random string with an error tolerance limit t from a biometric template and also outputs a public string as auxiliary information. Namely, the fuzzy extractor can output the same random string with the help of the public string even if there is a minor change in the input. The fuzzy extractor consists of two algorithms, as follows:

- $GEN(Bio_i) = (B_i, C_i)$: Given a biometric template Bio_i as the input, this probabilistic algorithm outputs a secret biometric key B_i and a helper string C_i.
- $REP(Bio_i', C_i) = (B_i)$: Given a noisy biometric Bio_i' and a helper string C_i as inputs, this deterministic algorithm reproduces the biometric key B_i.

2) ELLIPTIC CURVE CRYPTOGRAPHY

Let $q > 3$ be a large prime and $E_{a,b}$ denote a group of points of the elliptic curve $E_{a,b} : y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_q$, where $a, b \in \mathbb{F}_q$ satisfy $4a^3 + 27b^2 \neq 0 \bmod q$. Let $G_p = P$ be a cyclic group of prime order p,

- *The Elliptic Curve Discrete Logarithm (ECDL) problem* finds a when given a point $Q \in G_p$, where $a \in \mathbb{Z}_p^*$ and $Q = aP$.
- *The Elliptic Curve Computational Diffie-Hellman (ECCDH) problem* finds abP when given aP and bP, where $a, b \in \mathbb{Z}_p^*$.

3) SYSTEM ARCHITECTURE

Many researchers have proposed different layers of architecture for IoT, including three-layer, four-layer, and five-layer architectures [22]. However, in terms of operations related to WSNs, IoT architecture can be simply expressed as shown in Figure 1, where the information collected from a WSN in the perception layer is delivered to the cloud through the gateway in the network layer, processed and refined at the cloud computing of the support layer, and passed to the application layer.

The IoT is highly heterogeneous because it connects a variety of devices, including existing ones, to devices newly developed for the IoT. As IoT applications are deployed at different locations and evolve over time, it is very likely that the heterogeneous devices developed by different manufacturers will communicate through a variety of communication techniques such as IEEE 802.15.4, ZigBee, Wi-Fi, Bluetooth, and 4G/5G [23]. In addition, for IoT applications, multiple WSNs

TABLE 1. Notations used in Adavoudi-Jolfaei *et al.*'s scheme.

Notation	Description	Notation	Description
U_i	User	N_i	Random number generated by U_i
GW	Gateway node	SK	Session key between U_i and GW_j
GW_j	Sensor node	APM	A set of users' access privilege masks
SC	Smart card	G	A set of users' group IDs
ID_i	Identity of the user	K_{ug}	Shared key between U_i and GW
AID_i	One-time-alias identity of U_i	KEM_{ug}	Shared emergency key between U_i and GW
SID	Shadow identity of the user	K_{gs}	Secret key between GW and GW_j
ID_G	Identity of the gateway	Ts_{ug}	Transaction sequence number
w	Secret key of the gateway	$h(\cdot)$	One-way hash function
SN_{id}	Identity of the sensor node	$\oplus$	XOR operation
PW_i	Password of the user	Bio_i	Biometric of the user
$GEN(Bio_i)$	One part of fuzzy extraction function, output a biometric key B_i, and a helper string C_i		
$REP(Bio_i, C_i)$	One part of fuzzy extraction function, output the biometric key B_i in $GEN(Bio_i)$		

may be deployed in large numbers in various environments. Sensor nodes in WSNs are also heterogeneous, ranging from just various types of sensor motes to sensors embedded in IoT devices. Unlike traditional mobile communication networks and the Internet, WSNs primarily use short-distance communication between objects by constructing wireless networks in an ad hoc manner. Namely, it is difficult to directly connect WSNs and traditional communication networks and the Internet to each other because they lack uniform standardization in communication techniques, and the data from WSNs cannot be transmitted long distance given the limited transmission range of WSNs [24]. Thus, there are limitations to direct communication between heterogeneous sensor nodes and user mobile devices. Instead, they are more likely to communicate through a gateway that acts as a bridge between them.

Figure 1 describes a system architecture for WSNs and 5G-integrated IoT as an extension and generic version of the previously introduced architecture for 5G-integrated WSNs [8]. There are three types of participants: users, the authentication and authorization server (AAS), and gateways (GWs). AAS with IoT application servers and system administrators is responsible for registering users, issuing membership parameters including access rights based on personal credit information, deploying WSNs, and setting up identities and keys for gateways and sensor nodes. AAS also helps with authentication, authorization, and key establishment between a gateway and a user when the user tries to log into the WSN in real time. After registration, authentication, and authorization, through the 5G network or the Internet, a user with a mobile device usually accesses IoT application data in the cloud and directly accesses WSNs through gateways for real-time data acquisition. In general, an IoT gateway plays an important role in IoT applications: facilitating the seamless integration of WSNs and traditional mobile communication networks or the Internet, and managing and controlling WSNs [24]. Likewise, in our system architecture, a gateway usually collects data from the sensor nodes of the WSN and delivers it to the cloud, which serves as a bridge between the user's mobile device and the sensor nodes for real-time data access. Among these, the proposed scheme focuses on user authentication, authorization, and key agreement when accessing WSNs in real time.

II. REVIEW AND CRYPTANALYSIS OF ADAVOUDI-JOLFAEI *et al.*'s SCHEME

In this section, we briefly review Adavoudi-Jolfaei *et al.*'s scheme [15] and show that it has security weaknesses.

A. REVIEW OF ADAVOUDI-JOLFAEI et al.'s SCHEME

Adavoudi-Jolfaei *et al.*'s scheme consists of four phases: registration, anonymous authentication and key exchange, password and biometric update, and dynamic node addition. We describe the first two phases in detail. The last two phases were omitted because they have little relevance to this work. Table 1 lists the notations used in Adavoudi-Jolfaei *et al.*'s scheme.

In both Gope *et al.*'s scheme and Adavoudi-Jolfaei *et al.*'s scheme, the sensor registration phase was missed, and thus we add it briefly according to their papers [7], [15]. Before WSN deployment, GW preloads SN_{id} and K_{gs} into the memory of each GW_j and saves SN_{id} and $K_{gs}^{\#}$ into the database, where $K_{gs}^{\#} = K_{gs} \oplus h(ID_G||w||SN_{id})$. In Adavoudi-Jolfaei *et al.*'s scheme, to provide access control, GW generates a set of access group IDs $G = \{G_1, G_2, \cdots\}$ and a set of access privilege masks $APM = \{APM_1, APM_2 \cdots\}$, where $G_j \in G$ is a 128-bit unique random number used to identity a particular access group, and $APM_j \in APM$ is a 128-bit random number except for the first 16 bits (high order) in which each bit defines a different task or service. `[83246]: [87132]: [00: 07: 9E: 45: F4: A4: ...]` is an example of a user access list [15] that consists of `[user id]: [group id]: [APM]`. If the first bit of the APM is a temperature bit and the corresponding bit is set as 1, then this indicates that all members of this access group can use the temperature parameter. A user can belong to one or more access groups, and multiple users who have similar access privileges can be organized into the same group.

1) REGISTRATION PHASE

In this phase, GW issues a smart card to an intended user via a secure channel. During this phase, depending on the probable user query, GW prepares an access list that defines the user's privilege and consists of ID_i, G_j and user access privilege mask APM_j.

1) U_i sends ID_i and a personal credential to GW.
2) For U_i, GW issues a smart card containing $\{K_{ug}, (SID, KEM_{ug}), Ts_{ug}, G_u, h(\cdot)\}$, where $K_{ug} = h(ID_i||n_g) \oplus ID_G$, $sid_j = h(ID_i||r_j||K_{ug})$, $SID = \{sid_1, sid_2, \cdots\}$, $KEM_{ug} = \{KEM_{ug_1}, KEM_{ug_2}, \cdots\}$, $KEM_{ug_j} = h(ID_i||sid_j||r_j')$, n_g, r_j, and r_j' are random numbers, and Ts_{ug} is a 64-bit random sequence number generated by GW. For U_i, GW finally saves $\langle Ts_{ug}, (SID, KEM_{ug}^{\#}), K_{ug}^{\#}, K_{gs}^{\#}, ID_i^{\#}, G^{\#}, APM^{\#}\rangle$ into the database, where $KEM_{ug}^{\#} = KEM_{ug} \oplus h(ID_G||ID_i||w)$, $K_{ug}^{\#} = K_{ug} \oplus h(ID_G||ID_i||w)$, $ID_i^{\#} = ID_i \oplus h(ID_G||ID_i||w)$, $G_j^{\#} = G_j \oplus h(ID_G||ID_i||w)$, $G^{\#} = \{G_1^{\#}, G_2^{\#}, \cdots\}$, $APM_j^{\#} = APM_j \oplus h(ID_G||ID_i||w)$, and $APM^{\#} = \{APM_1^{\#}, APM_2^{\#}, \cdots\}$.
3) U_i inputs PW_i and Bio_i, then SC stores $\langle K_{ug}^*, f_{ug}^*, (SID^*, KEM_{ug}^*), Ts_{ug}, G^*, C_i, GEN(\cdot), REP(\cdot), h(\cdot)\rangle$ in its memory, where $GEN(Bio_i) = (B_i, C_i)$, $K_{ug}^* = h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$, $KEM_{ug}^* = KEM_{ug} \oplus h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$, $SID^* = SID \oplus h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$, $G^* = G \oplus h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$, $f_u^* = h(h(K_{ug}) \oplus h(ID_i) \oplus h(PW_i) \oplus h(B_i))$.

2) ANONYMOUS AUTHENTICATION AND KEY EXCHANGE PHASE

In both Gope *et al.*'s scheme and Adavoudi-Jolfaei *et al.*'s scheme, to speed up the authentication processes and to prevent any replay attacks, a 64-bit random sequence number, Ts_{ug}, is used as a one-time pseudonym. To provide user anonymity and untraceability, the researchers also employed a set of unlinkable shadow IDs, SID, and a corresponding set of emergency keys, KEM. These values are used during a loss of synchronization of Ts_{ug} between U_i and GW.

1) U_i inputs ID_i, PW_i and biometrics Bio_i, then SC computes $B_i = REP(Bio_i, C_i)$, $K_{ug} = K_{ug}^* \oplus h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$, and $f_u = h(h(K_{ug}) \oplus h(ID_i) \oplus h(PW_i) \oplus h(B_i))$. SC checks $f_u \stackrel{?}{=} f_u^*$. If so, then SC computes $N_x = K_{ug} \oplus N_i$, where N_i is a random number generated by U_i, $G = G^* \oplus h(h(ID_i) \oplus h(PW_i) \oplus h(B_i))$ and $AID_i = h(ID_i||K_{ug}||N_i||Ts_{ug})$. Then, U_i chooses an access group-ID G_j from G. Finally, SC computes $G_j' = G_j \oplus N_i$ and $V_1 = h(AID_i||G_j'||K_{ug}||N_x||SN_{id})$. In case of loss of synchronization, U_i chooses one of the unused pair of (sid_j, KEM_{ug_j}) from (SID^*, KEM_{ug}^*) and assigns sid_j as AID_i and KEM_{ug_j} as K_{ug}. SC sends a request message $\langle AID_i, G_j', N_x, Ts_{ug}$ (if req), $SN_{id}, V_1\rangle$ to GW.
2) GW first checks the validity of Ts_{ug} provided by U_i. If GW cannot find Ts_{ug} in its database, then it terminates the connection. Otherwise, GW selects the tuple related to U_i using Ts_{ug}. GW decodes ID_i and K_{ug}, and checks the validity of V_1. If so, then GW computes $N_i = N_x \oplus K_{ug}$ and $G_j = G_j \oplus N_i$, and checks $AID_u' \stackrel{?}{=} AID_i$, where $AID_u' = h(ID_i||K_{ug}||N_i||Ts_{ug})$. If so, then GW computes $APM_j' = h(K_{gs}) \oplus APM_j$ by finding APM_j related to G_j, generates SK and a timestamp T, and finally sends the message by computing $SK' = h(K_{gs}) \oplus SK$ and $V_2 = h(AID_i||APM_j'||SK'||T||K_{gs})$. In case of loss of synchronization, U_i will resend the request message using $AID_i = sid_j$ and $K_{ug} = KEM_j$ instead of using Ts_{ug}. In this case, GW will check the validity of AID_i by comparing sid_j with the entries in its database. If GW can find it, then GW derives the tuple associated with sid_j and retrieves KEM_j. GW checks the validity of V_1 with these values and sends a message $\langle AID_i, APM_j', SK', T, V_2\rangle$ to GW_j.
3) GW_j first checks the freshness of T and verifies V_2. If so, then GW_j computes $APM_j = APM_j' \oplus h(K_{gs})$ and generates a timestamp T'. GW_j then derives $SK = SK' \oplus h(K_{ug})$ and computes $V_3 = h(SK||K_{gs}||SN_{id}||T')$. Finally, GW_j sends the response message $\langle T', SN_{id}, V_3\rangle$ and updates $K_{gs} = K_{gs_{new}}$, where $K_{gs_{new}} = h(K_{gs}||SN_{id})$.
4) GW first checks the freshness of T', generates a random number m, and computes $Ts_{ug_{new}} = m, Ts = h(K_{ug}||ID_i||N_i) \oplus Ts_{ug_{new}}$, $SK'' = h(K_{ug}||ID_i||N_i) \oplus SK$, and $V_4 = h(SK''||N_i||Ts||K_{ug})$. Finally, GW sends the response message $\langle SK'', V_4, Ts, x$ (if req)$\rangle$ and updates $K_{ug} = K_{ug_{new}}$ and $K_{gs} = K_{gs_{new}}$, where $K_{ug_{new}} = h(K_{ug}||ID_i||Ts_{ug_{new}})$ and $K_{gs_{new}} = h(K_{gs}||SN_{id})$. In the case of loss of synchronization, instead of the above update method, GW randomly generates $K_{ug_{new}}$ and sends $x = K_{ug_{new}} \oplus h(ID_i||KEM_j)$ with other parameters.
5) U_i first checks V_4. If so, then U_i derives $SK = SK'' \oplus h(K_{ug}||ID_i||N_i)$ and updates $Ts_{ug} = Ts_{ug_{new}}$ and $K_{ug} = K_{ug_{new}}$, where $Ts_{ug_{new}} = h(K_{ug}||ID_i||N_i) \oplus Ts$ and $K_{ug_{new}} = h(K_{ug}||ID_i||Ts_{ug_{new}})$. In the case of loss of synchronization, U_i performs a different update to $K_{ug} = K_{ug_{new}}$, where $K_{ug_{new}} = h(ID_i||KEM_j) \oplus x$.

B. SECURITY FLAWS IN ADAVOUDI-JOLFAEI et al.'s SCHEME

In this section, we discuss the cryptanalysis of Adavoudi-Jolfaei *et al.*'s scheme and the observation of several security weaknesses.

1) USER COLLUSION ATTACKS

In access control systems, a user collusion attack is that two or more malicious users with different privileges deceive the system to obtain a service or data with higher privileges. In the systems, if there is no collusion of users possible, this may be a too strong assumption. Rather, users are more likely to try to get more data that requires higher privileges than their

own at low cost [25]. In Adavoudi-Jolfaei *et al.*'s scheme, a user can have multiple access group IDs, and multiple users with similar privileges can share the same access group ID. As users' access group IDs are given to users when they are in the registration phase, the users can exploit other users' group IDs through user collusion to obtain sensor data that requires higher privileges. GW stores the group ID that a user has in the database but does not verify that the group ID presented by the user in the anonymous authentication and key exchange phase is the group to which the user belongs. Therefore, Adavoudi-Jolfaei *et al.*'s scheme is vulnerable to user collusion attacks. In addition, Adavoudi-Jolfaei *et al.*'s scheme does not provide access privilege updates.

2) DESYNCHRONIZATION ATTACK

Both Gope *et al.*'s scheme [7] and Adavoudi-Jolfaei *et al.*'s scheme [15] employ a transaction sequence number Ts_{ug} as a one-time pseudonym to provide user anonymity and untraceability, and to prevent replay attacks. In Gope *et al.*'s scheme, at the end of the anonymous authentication and key exchange phase, this number is updated by incrementing U_i and GW by 1 to speed up the authentication process and to prevent a replay attack. On the other hand, Adavoudi-Jolfaei *et al.* showed that Gope *et al.*'s scheme is vulnerable to a session-key disclosure attack owing to the above simple update method. However, to solve this problem, they used the vulnerable update method of [3], [9] that Gope *et al.* identified. Namely, in [3], [9], for untraceability, the updated temporal identity used for the next session is transmitted from the gateway to a user at the end of the authentication phase. Likewise, in Adavoudi-Jolfaei *et al.*'s scheme, the updated $Ts_{ug_{new}}$ is transmitted to a user. Thus, if the last response message sent from GW is disrupted by an adversary, it will cause a loss of synchronization between the user and GW.

Moreover, both schemes utilize a set of shadow IDs, SID, and the corresponding set of emergency keys, KEM_{ug}, for each user to solve the problem of synchronization loss. However, this causes another desynchronization attack. In the registration phase, if GW cannot find Ts_{ug} of the request message sent from U_i in its database, then GW will terminate the connection. Upon receiving this termination message, U will resend the request message using one of the shadow IDs and an emergency key. In this case, an adversary can exploit this method by arbitrarily changing Ts_{ug} of the request message to break the synchronization between GW and U_i and to exhaust SID and KEM_{ug} shared between them.

In addition, they have not specified how many sid_js and KEM_{ug_j}s each user has in both schemes and how to handle them when they are exhausted. Although the storage of smart cards, users' terminals, and GW is not restrictive compared to the storage of sensor nodes, as the number of sid_js and KEM_{ug_j} and the number of users increase, the storage cost will be exacerbated.

3) NO SENSOR NODE ANONYMITY

Anonymity in WSNs means preventing a third party other than the message sender and receiver from knowing the identity of the two primary parties in communication. This includes the sender anonymity, receiver anonymity, and unlinkability between the sender and receiver [26]. Thus, the anonymity of the sensor node is as important as the user anonymity. In particular, sensor node anonymity means that no adversary can trace different sessions from a special sensor node and launch further attacks (e.g., a sensor node impersonation attack and sensor node capture attack) by hiding the sensor node's identity. If the identity of a sensor node is exposed to the adversary in plain text in the transmitted messages, then the adversary can identify the frequently accessed sensor node by users. This means that the adversary can identify an important sensor node with more data of interest to users, and eventually, that sensor node is likely to be the adversary's preferred target for attack.

In Adavoudi-Jolfaei *et al.*'s scheme, U_i and GW_j send the request message $\langle AID_i, G'_j, N_x, Ts_{ug}$ (if req), $SN_{id}, V_1\rangle$ and response message $\langle T', SN_{id}, V_3\rangle$ to GW via an insecure channel. Clearly, if an adversary intercepts either the request message of U_i or the response message of GW_j, he/she can obtain GW_j's identity SN_{id}. Thus, Adavoudi-Jolfaei *et al.*'s scheme does not ensure sensor node anonymity.

III. OUR PROPOSED SCHEME

In this section, we propose an ECC-based anonymous authentication, authorization and key agreement scheme as an improved version of Adavoudi-Jolfaei *et al.*'s scheme. The proposed scheme remedies security vulnerabilities based on the system architecture in WSNs for 5G-integrated IoT. Figure 1 illustrates the system architecture of the proposed scheme. Our proposed scheme is split into five phases: (1) setup; (2) user registration; (3) authentication, authorization, and key agreement (AAK); (4) password and biometrics update; and (5) access privilege update. ECC is asymmetric key cryptography and provides similar security measures with smaller key sizes in comparison with other non-ECC-based asymmetric key cryptography methods such as RSA [27]. As WSNs are resource-constrained, techniques that are more lightweight, such as symmetric ones (XOR and hash computations, for example), are more appropriate [4]. However, efficient ECC implementations in resource-constrained sensor motes have continued to be proposed [28]–[30], thus increasing the feasibility and practicality of ECC in IoT devices. Moreover, although we employ ECC to address the security weaknesses found in Adavoudi-Jolfaei *et al.*'s scheme, ECC operations are performed by users, authentication and authorization servers, and gateways with fewer resource constraints than sensor nodes. Table 2 lists different and additional notations used in the proposed scheme.

TABLE 2. List of notations used in proposed scheme.

Notation	Description	Notation	Description
AAS	Authentication & authorization server	α	Membership verification secret of AAS
ID_{AAS}	Identity of AAS	β	Access privilege verification secret of AAS for authorization
(y, Q_{AAS})	Private and public keys of AAS	K_j	Secret key between AAS and GW_j
TID_i	Temporal identity of U_i	AL_i	Authorization list for U_i
M_i	Membership of U_i	APG_ℓ	ℓ-th access privilege group index
MD_i	Mobile device of U_i	APR_ℓ	Unique random number of APG_ℓ
GW_j	j-th Gateway	LMA_ℓ^i	Linking value between TID_i and APG_ℓ for U_i
GID_j	Identity of GW_j	$T, \Delta T$	Timestamp and time interval for allowed transmission delay

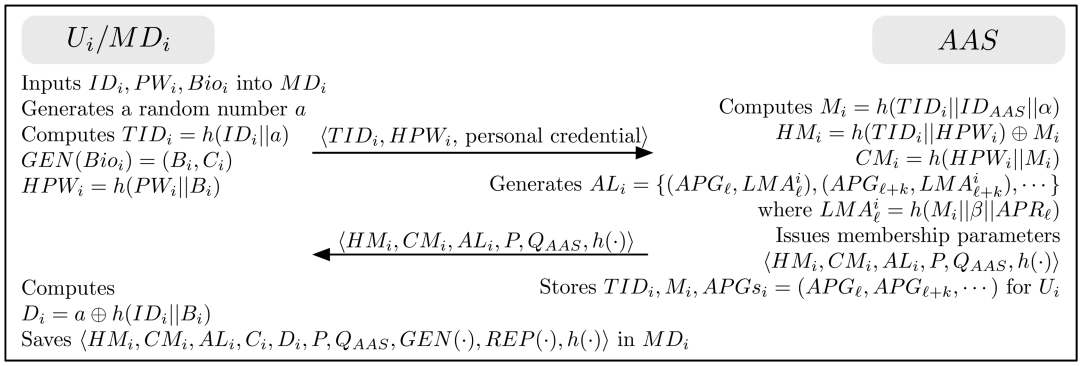

FIGURE 2. User registration phase of our proposed scheme.

A. SYSTEM SETUP PHASE

This phase includes the initialization of the system parameters and gateway and sensor node registration before deployment.

1) *AAS* chooses an elliptic curve E over prime finite field F_q and an additional subgroup G of E, which is generated by P with a large prime order p. *AAS* then generates its private and public key pair $\{y, Q_{AAS}\}$, where $y \in \mathbb{Z}_p^*$ and $Q_{AAS} = yP$. *AAS* also chooses its own identity ID_{AAS}, membership verification secret α, access privilege verification secret β, and secure one-way hash function $h(\cdot)$. *AAS* publishes the system parameters $\{E, G, p, P\}$.
2) According to authorization policies of the system, *AAS* generates indices of access privilege groups, $(APG_1, \cdots, APG_\ell, \cdots, APG_L)$, a unique random number, APR_ℓ, and access privilege mask, APM_ℓ, for each group [15], where L is the number of access privilege groups.
3) For each gateway GW_j, where $1 \le j \le J$ and J is the number of gateways, *AAS* selects a unique identity GID_j and generates a shared secret key K_j between *AAS* and GW_j. According to the WSN deployment plan, *AAS* divides and allocates all sensor nodes into J gateways. For each sensor node SN_n, where $1 \le n \le N$ and N is the total number of sensor nodes, *AAS* selects a unique identity SID_n and generates a shared secret key $K_{GW_jSN_n}$ between GW_j and SN_n. Secure communication through these shared secret keys and data transmission between gateways and sensor nodes are out of the scope of this paper.
4) For each gateway GW_j, *AAS* preloads $\langle ID_{AAS}, Q_{AAS}, GID_j, K_j\rangle$, identities of sensor nodes belonging to the corresponding gateway and secret keys shared with them, into GW_j's memory. For each sensor node SN_n, *AAS* then preloads $\langle GID_j, SID_n, K_{GW_jSN_n}\rangle$ into SN_n's memory.
5) Finally, *AAS* stores all system parameters and generated values for gateways and sensor nodes in its database, and deploys WSNs in the target area.

B. USER REGISTRATION PHASE

The user registration phase begins when a user U_i sends a request message for registration to *AAS* over a secure channel. Figure 2 illustrates the user registration phase. This phase is described below.

1) U_i inputs an identity ID_i, a password PW_i,, and biometrics Bio_i into MD_i. U_i then selects a random number a and computes $TID_i = h(ID_i||a)$, $GEN(Bio_i) = (B_i, C_i)$, and $HPW_i = h(PW_i||B_i)$. Finally, U_i sends a registration request with TID_i, HPW_i, and a personal credential to *AAS* over a secure channel.
2) *AAS* verifies U_i's personal credential and computes a membership value $M_i = h(TID_i||ID_{AAS}||\alpha)$, $HM_i = h(TID_i||HPW_i) \oplus M_i$, and $CM_i = h(HPW_i||M_i)$. *AAS* selects access privilege groups (i.e., ℓ-th and $\ell + k$-th privileges) suitable for the user's privileges and computes linking values between the

membership M_i and access privilege groups such that $LMA_\ell^i = h(M_i||\beta||APR_\ell)$. *AAS* then generates an $AL_i = \{(APG_\ell, LMA_\ell^i), (APG_{\ell+k}, LMA_{\ell+2}^i)\}$ for U_i and sends membership parameters $\langle HM_i, CM_i, AL_i, P, Q_{AAS}, h(\cdot)\rangle$ to U_i over a secure channel. *AAS* finally stores TID_i, M_i and $APGs_i = \{APG_\ell, APG_{\ell+k}, \cdots\}$ in its database.

3) Upon receiving the membership parameters, U_i computes $D_i = a \oplus h(ID_i||B_i)$ and stores $\langle HM_i, CM_i, AL_i, C_i, D_i, P, Q_{AAS}, GEN(\cdot), REP(\cdot), h(\cdot)\rangle$ into its memory.

C. AUTHENTICATION, AUTHORIZATION, AND KEY AGREEMENT (AAK) PHASE

Whenever U_i wants to access the WSN in charge of GW_j, the following steps should be performed with U_i, *AAS*, and GW_j over a public channel. With the help of *AAS*, U_i and GW_j mutually authenticate each other and establish a common session key for future communication. Finally, U_i can obtain the sensory data in real time from the WSN that matches his/her access privileges. Figure 3 illustrates the AAK phase, and this phase is described below.

1) To log into the WSN, U_i inputs an identity ID_i, password PW_i,, and biometrics Bio_i into MD_i. Using the stored values, MD_i computes $B_i = REP(Bio_i, C_i)$, $a = D_i \oplus h(ID_i||B_i)$, $TID_i = h(ID_i||a)$, $HPW_i = h(PW_i||B_i)$, $M_i = HM_i \oplus h(TID_i||HPW_i)$, and $CM_i^* = h(M_i \oplus h(TID_i||PW_i||B_i))$ and checks $CM_i^* \stackrel{?}{=} CM_i$. If this does not hold, then the login request is rejected by MD_i as at least one factor of the identity, password, or biometrics is invalid. Otherwise, MD_i selects GID_j and retrieves a proper access privilege group APG_ℓ and LMA_ℓ^i from AL_i. MD_i then generates a random value $x \in \mathbb{Z}_p^*$ and timestamp T_1. MD_i computes $X_i = xP$, $Y_i = xQ_{AAS}$, $MID_i = TID_i \oplus h(X_i||Y_i)$, $MGW_i = GID_j \oplus h(Y_i||T_1)$, $MAPG_\ell^i = APG_\ell \oplus h(M_i||T_1)$, $MLMA_\ell^i = LMA_\ell^i \oplus h(TID_i||T_1)$, and $V_1 = h(TID_i||GID_n||APG_\ell||M_i\ ||X_i||Y_i||T_1)$. MD_i sends a login request $\langle MID_i, MGW_i, MAPG_\ell^i, MLMA_\ell^i, X_i, V_1, T_1\rangle$ to *AAS*.
2) Upon receiving the login request, *AAS* checks the validity of timestamp T_1. *AAS* computes $Y_i' = yX_i$, $TID_i' = MID_i \oplus h(X_i||Y_i')$ and $M_i' = h(TID_i'||ID_{AAS}||\alpha)$ and retrieves U_i's membership M_i from the database using TID_i'. *AAS* checks whether U_i is a member of *AAS* by verifying $M_i' \stackrel{?}{=} M_i$. If this does not hold, then *AAS* rejects the U_i's login request. Otherwise, *AAS* computes $GID_j' = MGW_i \oplus h(Y_i'||T_1)$, $APG_\ell'^i = MAPG_\ell^i \oplus h(M_i'||T_1)$, and $V_1' = h(TID_i'||GID_j||APG_\ell'^i||M_i'||X_i||Y_i||T_1)$ and checks $V_1' \stackrel{?}{=} V_1$. If this does not hold, then *AAS* terminates the U_i's login request. Otherwise, *AAS* computes $LMA_\ell'^i = MLMA_\ell^i \oplus h(TID_i'||T_1)$ and checks whether the access privilege group APG_ℓ' suggested by U_i matches the data access privileges of the requested WSN in charge of GW_j. If so, then *AAS* retrieves $APGs_i$, unique random number APR_ℓ, and access privilege mask APM_ℓ regarding APG_ℓ'. *AAS* then checks that APG_ℓ' belongs to $APGs_i$, computes $LMA_\ell''^i = h(M_i'||\beta||APR_\ell)$, and checks $LMA_\ell''^i \stackrel{?}{=} LMA_\ell'^i$ to verify that U_i actually has legitimate privileges of access privilege group $APG_\ell'^i$. If this does not hold, then *AAS* regards that U_i does not have legitimate privilege and sends a message that it is inaccessible to the WSN to U_i. Otherwise, *AAS* generates a timestamp T_2 and computes $MID_i^* = h(TID_i'||GID_j'||Y_i')$, $MAPM_\ell = APM_\ell \oplus h(GID_j'||K_j||T_2)$, and $V_2 = h(MID_i^*||GID_j'||APM_\ell||X_i||K_j||T_2)$. *AAS* sends the message $\langle MID_i^*, MAPM_\ell, X_i, V_2, T_2\rangle$ to GW_j.
3) Upon the receiving the message from *AAS*, GW_j checks the validity of timestamp T_2. If so, then GW_j computes $APM_\ell' = MAPM_\ell \oplus h(GID_n||K_j||T_2)$ and $V_2' = h(MID_i^*||GID_j\ ||APM_\ell||X_i||K_j||T_2)$ and checks $V_2' \stackrel{?}{=} V_2$. If this does not hold, then GW_j terminates the session. Otherwise, GW_j generates a random number z and timestamp T_3 and computes $Z_j = zP$, $SK = h(MID_i^*||SK)$, $V_3 = h(ID_{AAS}||GID_j||SK)$, and $V_4 = h(MID_i^*||GID_j||Z_j||V_3||K_j\ ||T_3)$. GW_j finally sends the message $\langle Z_j, V_3, V_4, T_3\rangle$ to *AAS*.
4) Upon receiving the message from GW_j, *AAS* checks the validity of timestamp T_3. If so, *AAS* computes $V_3' = h(MID_i^*||GID_j||Z_j||V_3||K_j||T_3)$ and checks $V_4' \stackrel{?}{=} V_4$. If this does not hold, then *AAS* terminates the session. Otherwise, *AAS* generates a timestamp T_4 and computes $V_5 = h(TID_i'||GID_j||M_i'||Z_j||V_3||Y_i'||T_4)$. *AAS* finally sends a response message $\langle Z_j, V_4, V_5, T_4\rangle$ to U_i.
5) Upon receiving the response message, MD_i checks the validity of timestamp T_4. If so, then MD_i computes $MID_i^* = h(TID_i||GID_j||Y_i)$, $SK = h(MID_i^*||xZ_j)$, and $V_3' = h(ID_{AAS}||GID_j||SK)$ and checks $V_3' \stackrel{?}{=} V_3$. If this does not hold, then the session is terminated. Otherwise, U_i can be confident that SK is shared with the desired GW_j. MD_i, and then computes $V_5' = h(TID_i||GID_j||M_i'||Z_j||V_3||Y_i||T_4)$ and checks $V_5' \stackrel{?}{=} V_5$. If this does not hold, then the session is terminated. Otherwise, *AAS* and GW_j are authenticated by U_i, and U_i shares a session key SK with GW_j.

D. PASSWORD AND BIOMETRIC UPDATE PHASE

This phase allows a user to update his/her own password PW_i and biometrics Bio_i without any interaction with *AAS*. When U_i wants to update PW_i and Bio_i, U_i first inserts his identity ID_i and old password PW_i, and imprints old biometrics Bio_i at MD_i. MD_i computes $B_i = REP(Bio_i, C_i)$, $a = D_i \oplus h(ID_i||B_i)$, $TID_i = h(ID_i||a)$, $HPW_i = h(PW_i||B_i)$, $M_i = HM_i \oplus h(TID_i||HPW_i)$, and $CM_i' = h(HPW_i||M_i)$, and checks $CM_i' \stackrel{?}{=} CM_i$. If this does not hold, then it means at least one of the authentication factors is invalid, and the update is

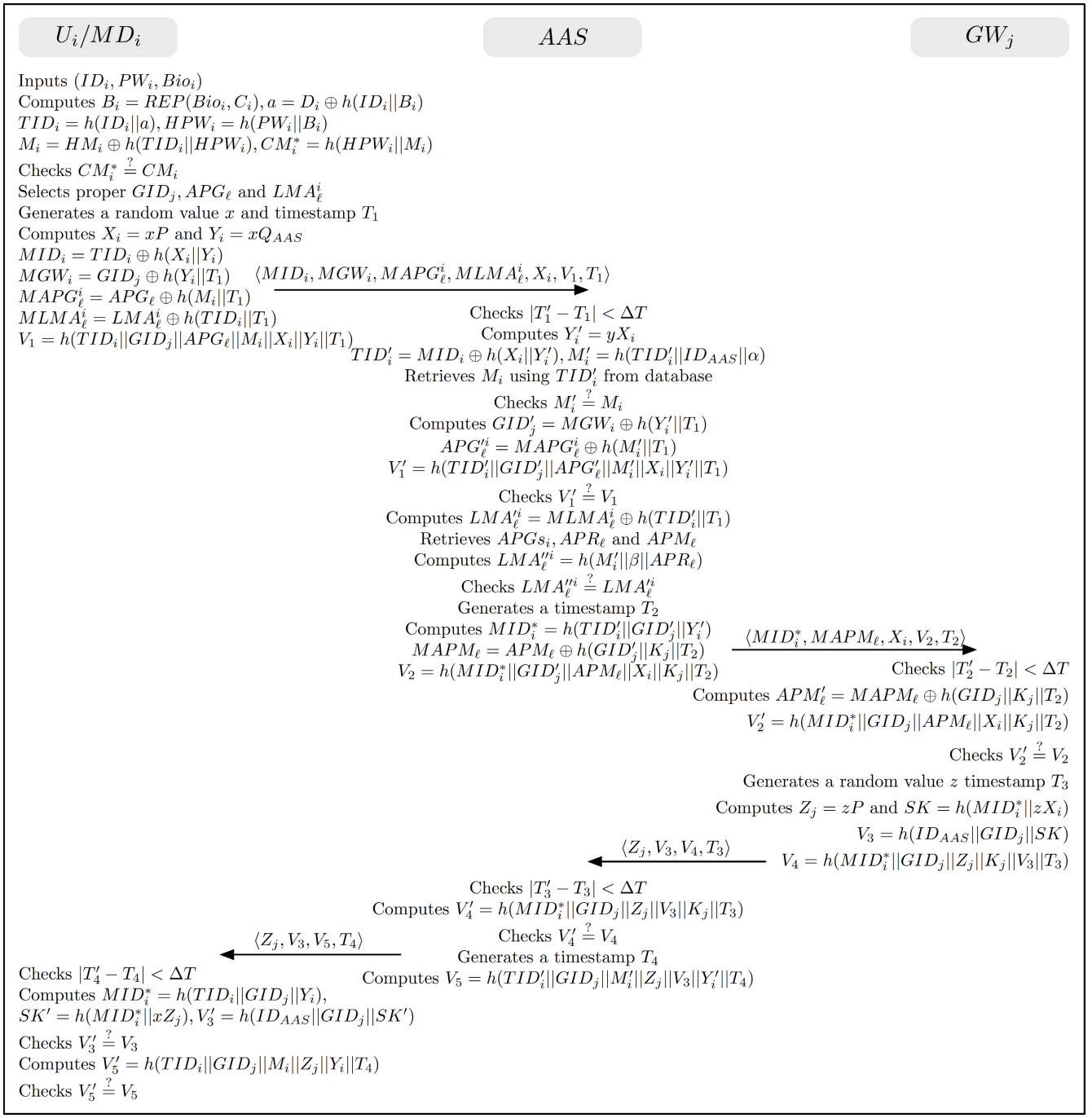

FIGURE 3. Authentication, authorization, and key agreement (AAK) phase of proposed scheme.

canceled. Otherwise, MD_i requires U_i to input a new password PW_i^* and to imprint new biometrics Bio_i^*. MD_i then computes $GEN(Bio_i^*) = (B_i^*, C_i^*), D_i^* = a \oplus h(ID_i||B_i^*), HPW_i^* = h(PW_i^*||B_i^*), HM_i^* = M_i \oplus h(TID_i||HPW_i^*)$, and $CM_i^* = h(HPW_i^*||M_i)$. Finally, MD_i replaces HM_i, CM_i, C_i, and D_i with HM_i^*, CM_i^*, C_i^*, and D_i^*, respectively.

E. ACCESS PRIVILEGE UPDATE PHASE

In most applications, there are often cases where a change in the access privileges given to a user is required owing to a change in policies, change in a user's position, and so on. This phase is performed between U_i and AAS to handle these cases. All messages in this phase are transmitted over a secure channel.

1) AAS sends an update request with TID_i and new access privilege list AL_i' to MD_i to inform U_i that his/her access privileges need to be updated.
2) Upon receiving the update request, MD_i informs U_i. U_i then inputs an identity ID_i, password PW_i, and biometrics Bio_i into MD_i. Using the inputted user information and stored values, MD_i then confirms U_i as in the login phase (step 1 in Section III-C) and replaces the stored AL_i with AL_i'. Finally, MD_i sends a message with the M_i that the access privilege list update is complete.
3) After verifying the membership M_i of U_i, AAS also replaces the stored $APGs_i$ with $APGs_i'$, including new access privilege groups for U_i.

IV. SECURITY ANALYSIS

In this section, we discuss the security of the proposed scheme by considering an informal and formal analysis. Using the widely accepted BAN logic [31], we prove that a session key can be correctly generated between U_i and GW_j. We adopt the AVISPA tool [32], [33] for the formal security verification of the proposed scheme. The informal analysis of the proposed scheme discusses its security features and robustness against relevant and well-known attacks. We finally compare the proposed scheme with other related schemes in terms of security features.

A. AUTHENTICATION PROOF BASED ON BAN LOGIC

We use BAN logic to prove the method in which a session key can be correctly generated between communicating parties during the AAK phase. The basic notations used in BAN logic are as follows:

- $P| \equiv X$: P believes X,
- $P \triangleleft X$: P sees X,
- $P| \sim X$: P said X,
- $P| \Rightarrow X$: P has jurisdiction over X,
- $\#(X)$: X is fresh,
- $P \overset{K}{\leftrightarrow} Q$: K is the shared key between P and Q,
- $\langle X \rangle_K$: X is combined with K, and K is usually a secret,

Generally, the BAN logic provides some rules as follows:

- **Rule 1 (Message meaning rule)** $\frac{P| \equiv P \overset{K}{\leftrightarrow} Q, P \triangleleft \langle X \rangle_K}{P| \equiv Q| \sim X}$: If P believes that the K is shared with Q and P sees X combined with K, then P believes Q said X.
- **Rule 2 (Nonce verification rule)** $\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$: If P believes that X is fresh and P believes that Q said X, then P believes that Q believes X.
- **Rule 3 (Freshness conjuncation rule)** $\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$: If P believes that X is fresh, then P believes that (X, Y) is fresh.
- **Rule 4 (Jurisdiction rule)** $\frac{P| \equiv Q| \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X}$: If P believes that X has jurisdiction over X and P believes that Q believes X, then P also believes X.

In the informal analysis based on BAN logic, the goals of the proposed scheme are defined as

- Goal 1: $U_i| \equiv GW_j| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$
- Goal 2: $U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$
- Goal 3: $GW_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$
- Goal 4: $GW_j| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$.

According to the proof steps in BAN logic, we convert the protocol messages into the idealized format as follows:

- M_1: $U_i \rightarrow AAS$: $\langle TID_i, GID_j, APG_\ell, LMA_\ell^i, X_i, T_1, U_i \overset{Y_i}{\longleftrightarrow} AAS \rangle_{M_i}$
- M_2: $AAS \rightarrow GW_j$: $\langle GID_j, MID_i^*, APM_\ell, X_i, T_2 \rangle_{K_j}$
- M_3: $GW_j \rightarrow AAS$: $\langle GID_j, MID_i^*, Z_j, V_3, T_3 \rangle_{K_j}$
- M_4: $AAS \rightarrow U_i$: $\langle TID_i, GID_j, Z_j, V_3, T_4 \rangle_{M_i}$.

We then define some assumptions as initiative premises as follows:

- P_1: $AAS| \equiv \#(T_1)$
- P_2: $GW_j| \equiv \#(T_2)$
- P_3: $AAS| \equiv \#(T_3)$
- P_4: $U_i| \equiv \#(T_4)$
- P_5: $U_i| \equiv (U_i \overset{M_i}{\longleftrightarrow} AAS)$
- P_6: $AAS| \equiv (U_i \overset{M_i}{\longleftrightarrow} AAS)$
- P_7: $GW_j| \equiv (GW_j \overset{K_j}{\longleftrightarrow} AAS)$
- P_8: $AAS| \equiv (GW_j \overset{K_j}{\longleftrightarrow} AAS)$
- P_9: $U_i| \equiv GW_j| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$
- P_{10}: $GW_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$.

We then prove that the proposed scheme achieves the above goals based on the idealized form of the messages, assumptions, and BAN logic rules as follows:

- From M_1, we get V_1: $AAS \triangleleft \langle TID_i, GID_j, APG_\ell, LMA_\ell^i, X_i, T_1, U_i \overset{Y_i}{\longleftrightarrow} AAS \rangle_{M_i}$.
- Then, according to P_6, V_1, and Rule 1, we get V_2: $AAS| \equiv U_i| \sim \langle TID_i, GID_j, APG_\ell, LMA_\ell^i, X_i, T_1, U_i \overset{Y_i}{\longleftrightarrow} AAS \rangle$.
- According to P_1 and Rule 3, we get V_3: $AAS| \equiv \#\langle TID_i, GID_j, APG_\ell, LMA_\ell^i, X_i, T_1, U_i \overset{Y_i}{\longleftrightarrow} AAS \rangle$.
- According to V_2, V_3, and Rule 2, we get V_4: $AAS| \equiv U_i| \equiv \langle TID_i, GID_j, APG_\ell, LMA_\ell^i, X_i, T_1, U_i \overset{Y_i}{\longleftrightarrow} AAS \rangle$.
- According to M_2, we get V_5: $GW_j \triangleleft \langle GID_j, MID_i^*, APM_\ell, X_i, T_2 \rangle_{K_j}$.
- According to P_7 and Rule 1, we get V_6: $GW_j| \equiv AAS| \sim \langle GID_j, MID_i^*, APM_\ell, X_i, T_2 \rangle$.
- According to P_2 and Rule 3, we get V_7: $GW_j| \equiv \#\langle GID_j, MID_i^*, APM_\ell, X_i, T_2 \rangle$.
- According to V_6, V_7, and Rule 2, we get V_8: $GW_j| \equiv AAS| \equiv \langle GID_j, MID_i^*, APM_\ell, X_i, T_2 \rangle$.
- According to M_3, we get V_9: $AAS \triangleleft \langle GID_j, MID_i^*, Z_j, V_3, T_3 \rangle_{K_j}$.
- According to P_8 and Rule 1, we get V_{10}: $AAS| \equiv | \sim \langle GID_j, MID_i^*, Z_j, V_3, T_3 \rangle$.
- According to P_3 and Rule 3, we get V_{11}: $AAS| \equiv \#\langle GID_j, MID_i^*, Z_j, V_3, T_3 \rangle$.

- According to V_{10}, V_{11}, and Rule 2, we get V_{12}: $AAS| \equiv GW_j| \equiv \langle GID_j, MID_i^*, Z_j, V_3, T_3\rangle$.
- According to M_4, we get V_{13}: $U_i \triangleleft \langle TID_i, GID_j, Z_j, V_3, T_4\rangle_{M_i}$.
- According to P_5 and Rule 1, we get V_{14}: $U_i| \equiv AAS| \sim \langle TID_i, GID_j, Z_j, V_3, T_4\rangle$.
- According to P_4 and Rule 3, we get V_{15}: $U_i| \equiv \#\langle TID_i, GID_j, Z_j, V_3, T_4\rangle$.
- According to V_{14}, V_{15}, and Rule 2, we get V_{16}: $U_i| \equiv AAS| \equiv \langle TID_i, GID_j, Z_j, V_3, T_4\rangle$.
- As $SK = h(MID_i^*||xZ_j)$ and combining V_{12}, V_{16}, we get V_{17}: $U_i| \equiv GW_j| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$ (**Goal 1**).
- As $SK = h(MID_i^*||zX_i)$ and combining V_4, V_8, we get V_{18}: $GW_j| \equiv U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$ (**Goal 3**).
- According to P_9, V_{17} and Rule 4, we get V_{19}: $U_i| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$ (**Goal 2**).
- According to P_{10}, V_{18} and Rule 4, we get V_{20}: $GW_j| \equiv (U_i \overset{SK}{\longleftrightarrow} GW_j)$ (**Goal 4**).

Therefore, the above logic proves that the proposed scheme achieves Goals 1–4 successfully. In other words, the proposed scheme achieves mutual authentication, and the session key SK is securely shared between U_i and GW_j.

B. SECURITY VERIFICATION USING AVISPA

AVISPA is one of the widely accepted tools for semi-automated formal security analysis. AVISPA provides the High-Level Protocol Specification Language (HLPSL), a modular role-based expressive formal language, for specifying protocols and their security properties. The HLPSL specification of the protocols is translated into a lower-level description language using the HLPSL2IF translator [32], [33]. In AVISPA, the intruder is modeled using the Dolev-Yao model, and the output format (OF) is generated by applying one of four back ends: On-the-fly Model-Checker (OFMA), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), or Tree-Automata-based Protocol Analyzer (TA4SP). The output describes precise information about the result and the conditions obtained.

The proposed scheme by AVISPA was simulated to evaluate its security. We first implemented the specifications in the HLPSL language for user U_i, authentication and authorization server AAS, gateway GW_j, session, environment, and goal. Figure 4, 5, and 6 illustrate the roles of U_i, AAS, and GW_j in the HLPSL language, respectively. Figure 7 illustrates the session, environment, and goal roles in the HLPSL language. The current version of HLPSL supports the standard authentication and secrecy goals. Five secrecy goals and four authentications of the proposed scheme are verified in the HLPSL implementation.

We executed the HLPSL specifications using the Security Protocol ANimator for AVISPA (SPAN) [34]. We chose the widely accepted OFMA and CL-AtSe back ends for the execution tests and a bounded number of session model checks. Figure 8 and 9 show the simulation results based on the OFMC and CL-AtSe back ends, respectively. The simulation results show that the proposed scheme is secure against passive and active attacks, such as the man-in-the-middle and replay attacks.

```
role user(Ui, AAS, GWj: agent, SKey1, SKey2: symmetric_key,
H, GEN, REP, EccMul: hash_func, Snd, Rcv: channel(dy))
played_by Ui
def=
    local State: nat, IDi, IDaas, GIDj, PWi, Bioi, Ai, Bi,
        Ci, Di, TIDi, HPWi, Mi, HMi, CMi, APGil, LMAil, P,
        Qaas, Xi, XXi, YYi, ZZj, MIDi2, T1, T4, SKij: text,
        MIDi1, MGWi, MAPGil, MLMAil, V1, V3, V5: message,
    Inc: hash_func
    const user_server, server_user, server_gateway,
        gateway_server, subs1, subs2, subs3, subs4,
        subs5: protocol_id
    init State :=0
    transition
        1. State = 0 /\ Rcv(start) =|>
        State' := 1 /\ IDi' := new() /\ PWi' := new()
           /\ Ai' := new() /\ TIDi' := H(IDi'.Ai')
           /\ Bi' := GEN(Bioi) /\ Ci' := GEN(Bioi)
           /\ HPWi' := H(PWi'.Bi')
           /\ Di' := xor(Ai, H(IDi'.Bi'))
           /\ Snd({TIDi'.HPWi'}_SKey1)
           /\ secret({IDi,PWi,Bioi,Bi}, subs1, {Ui})
        2. State = 1
           /\ Rcv({HMi'.CMi'.APGil'.LMAil'.P.Qaas}_SKey1) =|>
        State' := 2 /\ Xi' := new() /\ T1' := new()
           /\ Bi' := REP(Bioi,Ci) /\ TIDi' := H(IDi.Ai)
           /\ HPWi' := h(PWi.Bi') /\ CMi' := H(HPWi'.Mi')
           /\ Mi' := xor(HMi', H(TIDi'.HPWi'))
           /\ XXi' := EccMul(Xi'.P)
           /\ YYi' := EccMul(Xi'.Qaas)
           /\ MIDi1' := xor(TIDi', H(XXi'.YYi'))
           /\ MGWi' := xor(GIDj, H(YYi'.T1'))
           /\ MAPGil' := xor(APGil, H(Mi'.T1'))
           /\ MLMAil' := xor(LMAil, H(TIDi'.T1'))
           /\ V1' := H(TIDi'.GIDj.APGil.Mi'.XXi'.YYi'.T1')
           /\ Snd(MIDi1'.MGWi'.MAPGil'.MLMAil'.XXi'.V1'.T1')
           /\ secret({TIDi,Mi,YYi}, subs2, {Ui, AAS})
           /\ witness(Ui, AAS, user_server, T1')
        3. State = 2 /\ Rcv(ZZj'.V3'.V5'.T4') =|>
        State' := 3 /\ MIDi2' := H(TIDi.GIDj.YYi)
           /\ SKij' := H(MIDi2'.EccMul(Xi.ZZj'))
           /\ request(AAS, Ui, server_user, T1)
           /\ secret({SKij}, subs3, {Ui, GWj})
end role
```

FIGURE 4. Role specification for user U_i.

C. INFORMAL SECURITY ANALYSIS

In this section, we show that the proposed scheme provides the desired security features and is also secure against well-known attacks.

1) MUTUAL AUTHENTICATION

In steps 2) and 5) of Section III-C, AAS and U_i authenticate each other by verifying the membership M_i and the correctness of V_1 and V_5. As only U_i with the correct password, biometrics, and the issued membership from AAS can compute the correct V_1, AAS can authenticate U_i via V_1. After receiving MID_i during step 1), as only AAS (who knows the corresponding private key y of Q_{AAS}) can compute the one-time share key Y_i between U_i and AAS, we derive TID_i from MID_i and compute the correct V_5. U_i can authenticate AAS via V_4.

In steps 3) and 4) in Section III-C, AAS and GW_j authenticate each other by verifying the correctness of V_2 and V_3. An adversary cannot generate legal $V_2 = h(MID_i^*||GID_j'||APM^{\ell}||X_i||K_j||T_2)$ and $V_3 = h(MID_i^*||GID_j||Z_j||K_j||T_3)$ without knowing their shared secret key K_j.

```
role server(Ui, AAS, GWj: agent, SKey1, SKey2: symmetric_key,
H, GEN, REP, EccMul: hash_func, Snd, Rcv: channel(dy))
played_by AAS
def=
  local State: nat, Alpha, Beta, Y, IDaas, TIDi, HPWi, Mi,
      HMi, CMi, APGil, LMAil, APMil, APRil, P, Qaas, XXi,
      YYi, ZZj, GIDj, Kj, MIDi2, T1, T2, T3, T4: text,
      MIDi1, MGWi, MAPGil, MLMAil, MAPMil,
      V1, V2, V3, V4, V5: message,
    Inc: hash_func
    const user_server, server_user, server_gateway,
      gateway_server, subs1, subs2, subs3, subs4,
      subs5: protocol_id
    init State :=0
    transition
       1. State = 0 /\ Rcv({TIDi'.HPWi'}_SKey1) =|>
       State' := 1 /\ Mi' := H(TIDi'.IDaas.Alpha)
         /\ HMi' := xor(Mi', H(TIDi'.HPWi'))
         /\ CMi' := H(HPWi'.Mi') /\ APGil' := new()
         /\ APRil' := new() /\ APMil' := new()
         /\ LMAil' := H(Mi'.Beta.APRil')
         /\ Snd({HMi'.CMi'.APGil'.LMAil'.P.Qaas}_SKey1)
         /\ GIDj' := new() /\ Kj' := new()
         /\ Snd({GIDj'.Kj'}_SKey2)
         /\ secret({Alpha, Beta, Y}, subs4, {AAS})
         /\ secret({Kj}, subs5, {AAS, GWj})
       2. State = 1
         /\ Rcv(MIDi1'.MGWi'.MAPGil'.MLMAil'.XXi'.V1'.T1') =|>
       State' := 2 /\ YYi' := EccMul(Y.XXi')
         /\ TIDi' := xor(MIDi1', H(XXi'.YYi'))
         /\ Mi' := H(TIDi'.IDaas.Alpha)
         /\ GIDj' := xor(MGWi', H(YYi'.T1'))
         /\ APGil' := xor(MAPGil', H(Mi'.T1'))
         /\ LMAil' := xor(MLMAil', H(TIDi'.T1'))
         /\ T2' := new() /\ MIDi2' := H(TIDi'.GIDj'.YYi')
         /\ MAPMil' := xor(APMil, H(GIDj'.Kj.T2'))
         /\ V2' := H(MIDi2'.GIDj'.APMil.XXi'.Kj.T2')
         /\ Snd(MIDi2'.MAPMil'.XXi'.V2'.T2')
         /\ request(Ui, AAS, user_server, T1')
         /\ witness(AAS, GWj, server_gateway, T2')
       3. State = 2 /\ Rcv(ZZj'.V3'.V4'.T3') =|>
       State' := 3 /\ T4' := new()
         /\ V5' := H(TIDi.GIDj.Mi.ZZj'.V3'.YYi.T4')
         /\ request(GWj, AAS, gateway_server, T3')
         /\ Snd(ZZj'.V3'.V5'.T4')
         /\ witness(AAS, Ui, server_user, T4')
end role
```

FIGURE 5. Role specification for server *AAS*.

```
role gateway(Ui, AAS, GWj:agent, SKey1, SKey2: symmetric_key,
H, GEN, REP, EccMul: hash_func, Snd, Rcv: channel(dy))
played_by GWj
def=
    local State: nat, IDaas, GIDj, Kj, P, APMil, Zj, ZZj,
      XXi, MIDi2, T2, T3, SKij: text,
      MAPMil, V2, V3, V4: message,
    Inc: hash_func
    const user_server, server_user, server_gateway,
      gateway_server, subs1, subs2, subs3, subs4,
      subs5: protocol_id
    init State :=0
    transition
       1. State = 0 /\ Rcv({GIDj'.Kj'}_SKey2) =|>
       State' := 1 /\ T3' := new()
       2. State = 1 /\ Rcv(MIDi2'.MAPMil'.XXi'.V2'.T2') =|>
       State' := 2 /\ APMil' := xor(MAPMil', H(GIDj.Kj.T2'))
         /\ Zj' := new() /\ ZZj' := EccMul(Zj'.P)
         /\ T3' := new() /\ SKij' := H(MIDi2'.EccMul(Zj'.XXi'))
         /\ V3' := H(IDaas.GIDj.SKij)
         /\ V4' := H(MIDi2'.GIDj.ZZj'.Kj.V3'.T3')
         /\ Snd(ZZj'.V3'.V4'.T3')
         /\ request(AAS, GWj, server_gateway, T2')
         /\ witness(GWj, AAS, gateway_server, T3')
end role
```

FIGURE 6. Role specification for gateway GW_j.

U_i and GW_j, and AAS authenticate each other. From the authentication relationship of the three parties, equivalently, U_i and GW_j can authenticate each other through the help of AAS. Therefore, the proposed scheme can achieve mutual authentication.

2) ANONYMITY AND UNTRACEABILITY

In the proposed scheme, the U_i's real identity ID_i is not transmitted during all phases, including the registration phase.

```
role session(Ui, AAS, GWj:agent, SKey1, SKey2: symmetric_key,
H, GEN, REP, EccMul: hash_func)
def=
    local US, UR, SS, SR, VS, VR: channel(dy)
    composition
      user(Ui, AAS, GWj, SKey1, SKey2, H, GEN, REP, EccMul,
      US, UR) /\ server(Ui, AAS, GWj, SKey1, SKey2, H, GEN,
      REP, EccMul, SS, SR) /\ gateway(Ui, AAS, GWj, SKey1,
      SKey2, H, GEN, REP, EccMul, VS, VR)
end role

role environment()
def=
    const ui, aas, gwj: agent, skey1, skey2 : symmetric_key,
      h, gen, rep, eccmul: hash_func,
     midi1, mgwi, mapgil, mlmail, xxi, midi2,
     mapmil, zzj, v1, v2, v3, v4, v5: text,
     user_server, server_user, server_gateway, gateway_server,
     subs1, subs2, subs3, subs4, subs5: protocol_id

    intruder_knowledge = {ui, aas, gwj, h, gen, rep, eccmul,
      midi1, mgwi, mapgil, mlmail, xxi, midi2, mapmil,
      zzj, v1, v2, v3, v4, v5}
    composition
      session(aas, ui, gwj, skey1, skey2, h, gen, rep, eccmul)
      /\ session(ui, aas, gwj, skey1, skey2, h, gen, rep, eccmul)
      /\ session(gwj, ui, aas, skey1, skey2, h, gen, rep, eccmul)
end role

goal
    secrecy_of subs1 secrecy_of subs2 secrecy_of
    subs3 secrecy_of subs4 secrecy_of subs5
    authentication_on user_server
    authentication_on server_user
    authentication_on server_gateway
    authentication_on gateway_server
end goal
environment()
```

FIGURE 7. Role specification for session, environment, and goal.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/myprotocol.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.11s
  visitedNodes: 8 nodes
  depth: 3 plies
```

FIGURE 8. Simulation result based on OFMC.

Thus, even if an adversary eavesdrops on all communication messages, it is not possible to obtain ID_i directly from the messages. Furthermore, the temporal identity TID_i is protected by the random value X_i and one-time shared key Y_i between AAS and U_i during transmission in the AAK phase. Even if an adversary obtains TID_i, it is not possible to derive ID_i from TID_i because ID_i is masked with a, and a is protected by Bio_i, which is only known to U_i.

In addition, for each session, every element of all messages during the AAK phase dynamically changes with random numbers and time stamps. Therefore, any adversary is unable to trace the different sessions of the specific user from the exchanged messages via public channels. Thus, the proposed scheme ensures the user anonymity with untraceability.

Further, the GW_j's identity GID_j is transmitted as masked by $h(Y_i||T_1)$ via a public channel. AAS can obtain it from

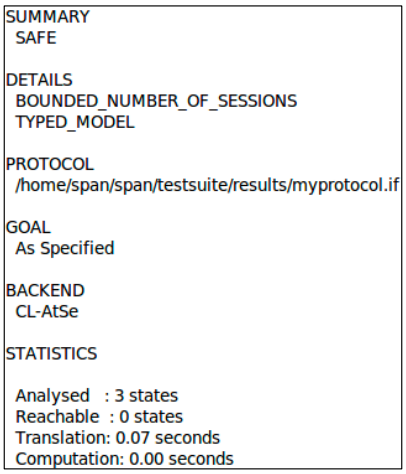

FIGURE 9. Simulation result based on CL-AtSe.

the login request by calculating $Y_i' = yX_i$, $TID_i' = MID_i \oplus h(X_i||Y_i')$, and $GID_j' = MGW_i \oplus h(Y_i'||T_1)$. Therefore, the proposed scheme also guarantees gateway anonymity.

3) SESSION KEY AGREEMENT, KNOWN-KEY SECRECY, AND FORWARD AND BACKWARD SECRECY

During the AAK phase, a session key $SK = h(MID_i^*||xZ_i) = h(MID_i^*||zX_i) = h(MID_i^*||xzP)$ is established between U_i and GW_j for protecting further communication. In the proposed scheme, the session key relies on xP and zP, where both x and z are random numbers that are different in each session. This session key is used only once for a current session and is independent of other sessions' keys owing to its random numbers. Even MID_i^* changes with the one-time shared key Y_i for each session. Thus, even if an adversary obtains SK for the $k-th$ session, he/she cannot compute any of the past and future session keys by using this disclosed SK. Furthermore, U_i can confirm that the correct SK agrees with the intended GW_j by verifying the correctness of V_4. Therefore, the proposed scheme guarantees both the session key agreement and known-key security.

Forward secrecy means that even if the long-term secret including the current session keys and all other long-term secret information is corrupted, then the past sessions are still secure. Backward secrecy is also referred to as future secrecy and guarantees the opposite direction of forward secrecy. In other words, this security property means that even if at some point the long-term secret information is corrupted, future messages can still be secure. As shown above, the session key $SK = h(MID_i^*||xzP)$ is associated with the secret random numbers x and z that are only known to U_i and GW_j, respectively. Even if all long-term secrets of U_i, AAS, and GW_j are compromised by an adversary and he/she obtains MID_i^*, X_i and Z_i by intercepting all messages, the previous and future session keys are still secure because the adversary has to resolve the intractable ECDL problem or ECDH problem in order to obtain them. As a result, the proposed scheme ensures forward secrecy and backward secrecy.

4) RESISTANCE TO MOBILE DEVICE LOSS AND OFFLINE PASSWORD GUESSING ATTACKS

If a mobile device MD_i of U_i is lost or stolen by an adversary, he/she can extract all stored values $\langle HM_i, CM_i, AL_i, C_i, D_i, P, Q_{AAS}, GEN(\cdot), REP(\cdot), h(\cdot)\rangle$ from MD_i through side-channel attacks such as a differential and simple power analysis [35]–[37]. Suppose the adversary obtains ID_i by accident and extracts all stored values from MD_i. If this occurs, then the adversary is likely to attempt to guess PW_i and B_i to obtain the information needed for user impersonation. However, owing to the one-way hash function with collision-resistant property, it is also intractable to guess the two values at the same time.

On the other hand, even if an adversary successfully extracts all information stored on MD_i, B_i is necessary to attempt an offline password guessing attack. However, the adversary cannot derive B_i using only C_i without knowing the U_i's biometric Bio_i. Moreover, even if the adversary guesses B_i correctly, HPW_i is necessary to check whether the guessed PW_i is correct. However, HPW_i is not stored on MD_i, so the adversary cannot obtain it. Therefore, the proposed scheme is resistant to mobile device loss and offline password guessing attacks.

5) RESISTANCE TO PRIVILEGED INSIDER AND STOLEN VERIFIER ATTACKS

U_i does not transmit PW_i in any phases of the proposed scheme. ID_i and PW_i are not used as is, but TID_i and HPW_i masked by the random value a and biometric key B_i, respectively, are used. Namely, only U_i knows ID_i and PW_i, and thus the proposed scheme is resistant to privileged insider attacks.

In a stolen verifier attack, an adversary steals or modifies the verification information (e.g., the plain texts of passwords, hashed passwords, biometric data, or hashed biometric key data) stored in the server's database. However, in the proposed scheme, U_i submits HPW_i masked by B_i instead of PW_i, and AAS maintains only $\langle TID_i, M_i, APGs_i\rangle$, which have no information related to the password or biometric key. Hence, the proposed protocol is resistant to stolen verifier attacks.

6) RESISTANCE TO IMPERSONATION ATTACKS

Assume that an adversary launches a user impersonation attack. The adversary may have the U_i's mobile device MD_i and all stored values in MD_i, and intercepts the messages transmitted in the previous session. For a successful attack, the adversary has to forge the login request with a new timestamp. However, without knowledge of the correct $ID_i, PW_i, B_i,$, and M_i and the possession of MD_i, he/she cannot generate a valid login request. Therefore, the proposed scheme is resistant to user impersonation attacks.

Assume that an adversary with the intercepted messages of the previous session tries to impersonate AAS to deceive

TABLE 3. Security feature comparison of proposed scheme with related three-factor authentication schemes.

Security Feature	Maurya and Sastry [12]	Jiang et al. [13]	Wadiz et al. [14]	Adavoudi-Jolfaei et al. [15]	Proposed Scheme
Mutual authentication	O	O	O	O	O
User anonymity w/o exhaustive search	X	O	X	O	O
Untraceability	X	O	O	O	O
Sensor node anonymity	O	O	O	X	O
Session key security	O	O	O	O	O
Perfect forward secrecy	O	X	X	O	O
Authorization	X	X	X	O	O
Resistant to					
Stolen smart card attack	O	O	O	O	O
Privileged insider attack	O	O	O	O	O
Stolen verifier attack	O	O	O	O	O
User impersonation attack	O	O	O	O	O
GW impersonation attack	O	O	O	O	O
Sensor node impersonation attack	O	O	O	O	O
User collusion attack	N/A	N/A	N/A	X	O
Desynchronization attack	O	O	O	X	O

* O: The scheme can provide the security feature or resist the attack; X: The scheme cannot provide the security feature or resist the attack; N/A: The scheme is not applicable because it does not provide an authorization property.

either U_i or GW_j. For this, the adversary has to generate the message of either step 5) or step 2), respectively. However, the adversary cannot compute the correct $\langle V_4 \rangle$ and $\langle MAPM_\ell, V_2 \rangle$ without knowing the AAS's private key y and the shared key K_j between AAS and GW_j, respectively. Thus, the proposed scheme is resistant to server impersonation attacks.

Assume that an adversary carries out a gateway impersonation attack. The adversary may have the intercepted messages of the previous session. For this attack, the adversary has to forge the message including a new timestamp transmitted from GW_j to AAS. However, without knowing the shared key K_j between AAS and GW_j, the adversary cannot compute V_3. Hence, the proposed scheme is resistant to gateway impersonation attacks.

7) RESISTANCE TO USER COLLUSION ATTACKS

For authorization, in the registration phase, AAS issues $AL_i = \{(APG_\ell, LMA^i_\ell), (APG_{\ell+k}, LMA^i_{\ell+2})\}$ for U_i. In the AAK phase, U_i sends APG_ℓ and LMA^i_ℓ, which are protected by M_i and TID_i, respectively. AAS then verifies in two steps that U_i has the certain access privilege APG_ℓ and that it really is the access privilege granted to U_i. The first step searches for $APGs_i$ stored in the database, and the second step examines LMA^i_ℓ.

Assume that a malicious user U_k obtains TID_i, M_i, and APG_ℓ from U_i, who colludes with U_k to escalate the access privilege. APG_ℓ is a higher privilege than $APG_{\ell-2}$ of U_k. To launch the user collusion attack described in Section II-B.1, U_k has to compute the correct $LMA^k_\ell = h(M_j||\beta||APR_\ell)$. However, no one except AAS knows $APRs$, and the access privilege verification secret β is also only known to AAS. Without these values, malicious users cannot collude each other in order to escalate their access privileges. Therefore, the proposed scheme is resistant to user collusion attacks.

8) RESISTANCE TO DESYNCHRONIZATION ATTACKS

In a desynchronization attack, an adversary breaks the synchronization of values shared between the server (or gateway) and users, making it impossible for users to log in and authenticate. In the proposed scheme, there is no need to update a temporal identity TID_i for untraceability because even if the same TID_i is used in each session, it is protected by a one-time secret value Y_i and transmitted as a different value each time. Thus, the proposed scheme avoids desynchronization attacks.

D. COMPARISON OF SECURITY FEATURES

In terms of security features, we compare the proposed scheme with recent three-factor authentication schemes [12]–[15] designed for IoT, except for Adavoudi-Jolfaei *et al.*'s scheme, which does not take IoT into account. Table 3 summarizes the comparison between the security features. From the results, we can see that the first three schemes do not support authorization. In addition, Maurya and Sastry's scheme and Wadiz *et al.*'s scheme do not provide user anonymity or untraceability. In Maurya and Sastry's scheme, a user's identity is transmitted in plain text over the published channel, and Wadiz *et al.*'s scheme requires exhaustive searching to check whether the login user is the registered user. Jiang *et al.*'s scheme and Wadiz *et al.*'s scheme do not guarantee perfect forward secrecy, so both schemes risk exposing session keys if long-term secret information is compromised by an adversary. As we discussed in Section II-B, Adavoudi-Jolfaei *et al.*'s scheme does not provide sensor node anonymity, and it is insecure against user collusion and desynchronization attacks. However, the proposed scheme not only guarantees basic security requirements including authorization but can also resist most known attacks.

V. PERFORMANCE ANALYSIS

In this section, we summarize the performance of the proposed scheme and compare it with related

TABLE 4. Performance comparison of proposed scheme with related three-factor authentication schemes.

Performance		Maurya and Sastry [12]	Jiang et al. [13]	Wadiz et al. [14]	Adavoudi-Jolfaei et al. [15]	Proposed Scheme
Computation	User	$T_F + 4T_H + 3T_P + 2T_E$	$T_B + T_M + 8T_H$	$T_F + 13T_H + 2T_E$	$T_F + 12T_H$	$T_F + 14T_H + 3T_P$
	GW [AAS]	$T_H + 3T_P + 3T_E$	$T_M + 12T_H$	$5T_H + 4T_E$	$9T_H$	$[T_P + 12T_H]$
	SN [GW]	T_E	$5T_H$	$4T_H + 2T_E$	$3T_H$	$[2T_P + 5T_H]$
	Total cost	≈13.66 ms	≈ 24.42 ms	≈ 25.78 ms	≈ 12.08 ms	≈ 8.66 ms
Communication	User	448 bits	864 bits	736 bits	800 bits	1,158 bits
	GW [AAS]	1,152 bits	992 bits	1,600 bits	992 bits	[1,516 bits]
	SN [GW]	625 bits	352 bits	768 bits	352 bits	[678 bits]
	Total cost	2,225 bits	2,208 bits	3,104 bits	2,144 bits	3,352 bits

* GW: gateway; SN: sensor node, [ ]: different participants and their performance for proposed scheme.

schemes [12]–[15] in terms of the computation and communication costs. Table 4 summarizes the results of the performance comparison. As the proposed scheme employs a system model that is distinct from those of related schemes, for the proposed scheme, the performance of AAS and the gateway instead of that of the gateway and sensor node, respectively, are included and marked with square brackets.

A. COMPUTATION COST

We analyze the computation cost of the proposed scheme and compare it with those of related schemes. We focus on the authentication and key agreement phase and do not consider XOR operations because the execution time is negligible. For a computation cost analysis, we define the execution time for the different cryptographic operations performed in two kinds of devices: a common PC and sensor mote. According to [5], for the user, server, and gateway, we use the execution time (T-series notation) measured in a computer system (Intel T5870 at 2.00 GHz) with the C/C++ library MIRACLE. According to [38]–[40], for the sensor node, we use the execution time (T'-series notation) measured in the MicaZ mote (8-bit ATmega128L microcontroller, 4K bytes of ROM, 512K bytes of EEPROM) with necC, TinySec, and TinyECCK. The execution time for the fuzzy extractor and biohash function is almost the same as that for the ECC point multiplication [41], [42] so that $T_F = T_B \approx T_P$. The execution times for different cryptographic operations are listed in Table 5.

The comparison results imply that the computational costs are largely affected by the type of operations at the sensor node. Despite the use of ECC point multiplication, which is a high-cost operation, the computation cost of the proposed scheme was measured at its lowest because it uses a different system model. However, as mentioned earlier, the system model used in the proposed scheme is more suited and reasonable to a 5G-integrated IoT environment, so the proposed scheme can be said to be efficient while guaranteeing various security features including authorization.

B. COMMUNICATION COST

We analyze only the frequently performed authentication and key agreement phase and measure the communication costs in bits as the lengths of messages sent by each participant. For convenience, as with the previous computation cost analysis, we assume a one-way hash function, symmetric key encryption algorithm, and ECC elliptic curves as SHA-1, AES-128, and ECC sect163rl [43], respectively. In other words, we assume that the length of the hash digest is 160 bits, the block size of the encryption message is 128 bits, and the size of the elliptic curve point is 326 bits. In particular, for encryption messages, the ciphertext length is calculated as a multiple of the block size. The other values such as identities and random numbers except for timestamps, whose length is 32 bits, are often XORed with the hash digest, so we assume their lengths are 160 bits.

TABLE 5. Execution time on common PC and sensor mote for cryptographic operations.

Notation	Operation	Execution time
T_H	one-way hash function (SHA-1)	2.58 μs
T_F	fuzzy extractor	1.226 ms
T_B	biohash function	1.226 ms
T_E	symmetric key enc./dec. (AES-128)	2.049 μs
T_P	ECC point multiplication (sect163rl [43])	1.226 ms
T_M	modular multiplication ($\|n\| = 512$)	2.573 ms
T'_H	one-way hash function (SHA-1)	3.6 ms
T'_E	symmetric key enc./dec. (AES-128)	5.05 ms

The communication cost analysis in Table 5 shows that Adavoudi-Jolfaei *et al.*'s scheme has the lowest total communication cost, and the communication costs of all schemes are lower than that of the proposed scheme. However, this can be justified as the proposed scheme provides better security and additional security features (e.g., authorization) compared with the related schemes.

VI. CONCLUSION

In this paper, we analyzed the three-factor authentication and access control scheme of Adavoudi-Jolfaei *et al.* and showed its security weaknesses. Adavoudi-Jolfaei *et al.*'s scheme does not support sensor node anonymity as strongly as user anonymity in WSNs and IoT. Furthermore, the scheme suffers from user collusion attacks because all users have the same values for access control and the gateway node does not check whether the presented access privilege from the user is indeed the user's privilege. To provide user anonymity and untraceability and to prevent a replay attack, the scheme uses a transaction sequence number as a one-time pseudonym, and

it is updated for every session. However, this value becomes a target of desynchronization attacks.

We then introduced a system architecture suitable for WSNs in 5G-integrated IoT. Based on this architecture, we proposed an ECC-based three-factor authentication, authorization, and key agreement scheme. Through a formal and informal security analysis of the proposed scheme, we showed that our scheme is capable of withstanding all possible attacks, and that it supports various security features. We also evaluated the performance of the proposed scheme. By comparing the security and performance of the proposed scheme with those of related schemes, we demonstrated that the proposed scheme achieves all desired security features without largely worsening the communication costs.

In our future work, we expect to evaluate the performance of the proposed scheme either by simulating it using NS3 or conducting experiments on actual devices (e.g., smartphones and sensor motes) in WSNs for 5G-integrated IoT. Based on the experimental results, we plan to optimize the proposed scheme and improve its performance.

REFERENCES

[1] G. Choudhary, J. Kim, and V. Sharma, "Security of 5G-mobile backhaul networks: A survey," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 9, no. 4, pp. 41–70, Dec. 2018.
[2] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1086–1090, Mar. 2009.
[3] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-Peer Netw. Appl.*, vol. 8, no. 6, pp. 1070–1081, Nov. 2015.
[4] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.
[5] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 428–442, Jul. 2015.
[6] I.-P. Chang, T.-F. Lee, T.-H. Lin, and C.-M. Liu, "Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks," *Sensors*, vol. 15, no. 12, pp. 29841–29854, 2015.
[7] P. Gope and T. Hwang, "A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7124–7132, Nov. 2016.
[8] S. Shin and T. Kwon, "Two-factor authenticated key agreement supporting unlinkability in 5G-integrated wireless sensor networks," *IEEE Access*, vol. 6, pp. 11229–11241, 2018.
[9] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-Peer Netw. Appl.*, vol. 9, no. 1, pp. 223–244, Jan. 2016.
[10] Y. Park and Y. Park, "Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 12, p. 2123, 2016.
[11] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Comput. Netw.*, vol. 101, pp. 42–62, Jun. 2016.
[12] A. Maurya and V. N. Sastry, "Fuzzy extractor and elliptic curve based efficient user authentication protocol for wireless sensor networks and Internet of Things," *Information*, vol. 8, no. 4, p. 136, 2017.
[13] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.
[14] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.
[15] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, and S. F. Aghili, "Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks," *Peer-Peer Netw. Appl.*, vol. 12, no. 1, pp. 43–59, Jan. 2019.
[16] S. Shin and T. Kwon, "A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes," *Sensors*, vol. 19, no. 9, p. 2012, 2019.
[17] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks," *Sensors*, vol. 17, no. 12, p. 2946, 2017.
[18] J. Moon, D. Lee, Y. Lee, and D. Won, "Improving biometric-based authentication schemes with smart card revocation/reissue for wireless sensor networks," *Sensors*, vol. 17, no. 5, p. 940, 2017.
[19] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2933, Jan. 2017.
[20] F. Wu, L. Xu, S. Kumari, and X. Li, "A privacy-preserving and provable user authentication scheme for wireless sensor networks based on Internet of Things security," *J. Ambient Intell. Hum. Comput.*, vol. 8, no. 1, pp. 101–116, Feb. 2017.
[21] A. K. Das and A. Goswami, "A robust anonymous biometric-based remote user authentication scheme using smart cards," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 27, no. 2, pp. 193–210, Apr. 2015.
[22] M. Burhan, R. Rehman, B. Khan, and B.-S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors*, vol. 18, no. 9, p. 2796, Aug. 2018.
[23] C. S. Shih, J. J. Chou, and K. J. Lin, "WuKong: Secure run-time environment and data-driven IoT applications for smart cities and smart buildings," *J. Internet Services Inf. Secur.*, vol. 8, no. 2, pp. 1–17, May 2018.
[24] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IoT gateway: Bridging-Wireless sensor networks into Internet of Things," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput.*, Dec. 2010, pp. 347–352.
[25] D. He, J. Bu, S. Zhu, S. Chan, and C. Chen, "Distributed access control with privacy support in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3472–3481, Oct. 2011.
[26] K. K. Gagneja, "Secure communication scheme for wireless sensor networks to maintain anonymity," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 1142–1147.
[27] A. K. Maurya, V. N. Sastry, and S. K. Udgata, "Cryptanalysis and improvement of ECC—Based security enhanced user authentication protocol for wireless sensor networks," in *Security in Computing and Communications*, J. H. Abawajy, S. Mukherjea, S. M. Thampi, and A. Ruiz-Martínez, Eds. Cham, Switzerland: Springer, 2015, pp. 134–145.
[28] D. Aranha, R. Dahab, J. López, and L. Oliveira, "Efficient implementation of elliptic curve cryptography in wireless sensors," *Adv. Math. Commun.*, vol. 4, no. 2, pp. 169–187, May 2010.
[29] Z. Liu, E. Wenger, and J. Großschädl, "MoTE-ECC: Energy-scalable elliptic curve cryptography for wireless sensor networks," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur. (ACNS)*, in Lecture Notes in Computer Science, vol. 8479. Cham, Switzerland: Springer, 2014, pp. 361–379.
[30] U. Gulen and S. Baktir, "Elliptic-curve cryptography for wireless sensor network nodes without hardware multiplier support," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 4992–5002, Dec. 2016.
[31] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
[32] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA tool for the automated validation of Internet security protocols and applications," in *Computer Aided Verification*. Berlin, Germany: Springer, 2005, pp. 281–285.
[33] *AVISPA Automated Validation of Internet Security Protocols and Applications*. Accessed: Apr. 15, 2019. [Online]. Available: http://www.avispa-project.org/
[34] *SPAN a Security Protocol Animator for AVISPA*. Accessed: Feb. 2, 2019. [Online]. Available: http://people.irisa.fr/Thomas.Genet/span/
[35] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*. Berlin, Germany: Springer, 1999, pp. 388–397.
[36] 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.
[37] H. J. Mahanta, A. K. Azad, and A. K. Khan, "Power analysis attack: A vulnerability to smart card security," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 506–510.

[38] S. C. Seo, D.-G. Han, H. C. Kim, and S. Hong, "TinyECCK: Efficient elliptic curve cryptography implementation over GF(2m) on 8-bit micaz mote," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 5, pp. 1338–1347, May 2008.
[39] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Comput. Netw.*, vol. 54, no. 17, pp. 2967–2978, Dec. 2010.
[40] R. Sankar, X. Le, S. Lee, and D. Wang, "Protection of data confidentiality and patient privacy in medical sensor networks," in *Implantable Sensor Systems for Medical Applications*. Cambridge, U.K.: Woodhead Publishing, 2013, pp. 279–298.
[41] D. He, N. Kumar, J.-H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 30–37, Feb. 2014.
[42] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1983–2001, Feb. 2016.
[43] Certicom Research, Certicom. (Jan. 2010). *Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0*. [Online]. Available: http://www.secg.org/download/aid-784/sec2-v2.pdf

SOOYEON SHIN (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Sejong University, Seoul, South Korea, in 2004, 2006, and 2012, respectively.

From 2012 to 2013, she was a Postdoctoral Researcher with Sejong University. In 2013, she joined Yonsei University, Seoul, to continue her Postdoctoral Research. Her current research interests include cryptographic protocols, network security, usable security, and human–computer interaction.

TAEKYOUNG KWON (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, South Korea, in 1992, 1995, and1999, respectively.

From 1999 to 2000, he was a Postdoctoral Research Fellow with the University of California, Berkeley, CA, USA. From 2001 to 2013, he was a Professor of computer engineering with Sejong University, Seoul. He is currently a Professor of information security with Yonsei University, where he is also the Director of the Information Security Laboratory. His research interests include authentication, cryptographic protocols, network security, software and system security, usable security, artificial intelligence security, and adversarial machine learning.

Dr. Kwon is a member of the Association for Computing Machinery and USENIX. He serves on the Director Board of the Korea Institute of Information Security and Cryptology. He also serves on the Editorial Committee of the Korean Institute of Information Scientists and Engineers.