On the Design of Secure and Efficient Three-Factor Authentication Protocol Using Honey List for Wireless Sensor Networks

JOONYOUNG LEE, SUNGJIN YU, MYEONGHYUN KIM, YOUNGHO PARK, (Member, IEEE), AND ASHOK KUMAR DAS, (Senior Member, IEEE)

1School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea
2Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

ABSTRACT

The Internet of Things (IoT) is useful for connecting and collecting variable data of objects through the Internet, which makes it generate useful data for humanity. An indispensable enabler of IoT is the wireless sensor networks (WSNs). Many environments, such as smart healthcare, smart transportation and smart grid, have adopted WSNs. Nonetheless, WSNs remain vulnerable to various attacks because they send and receive data over public channels. Moreover, the performance of IoT enabled sensor devices has limitations, since the sensors are lightweight devices and are resource constrained. To overcome these problems, many security authentication protocols for WSNs have been proposed. However, many researchers have pointed out that preventing smartcard stolen and offline guessing attacks is an important security issue, and guessing identity and password at the same time is still possible. To address these weaknesses, this paper presents a secure and efficient authentication protocol based on three-factor authentication by taking advantage of biometrics. Meanwhile, the proposed protocol uses a honey_list technique to protect against brute force and stolen smartcard attacks. By using the honey_list technique and three factors, the proposed protocol can provide security even if two of the three factors are compromised. Considering the limited performance of the sensors, we propose an efficient protocol using only hash functions excluding the public key based elliptic curve cryptography. For security evaluation of the proposed authentication protocol, we perform informal security analysis, and Real-Or-Random (ROR) model-based and Burrows Abadi Needham (BAN) logic based formal security analysis. We also perform the formal verification using the widely-used Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation software. Besides, compared to previous researches, we demonstrate that our proposed authentication protocol for WSNs systems is more suitable and secure than others.

INDEX TERMS

Authentication, AVISPA, BAN logic, Internet of Things (IoT), ROR model, wireless sensor network, honey list.

I. INTRODUCTION

As the IoT notions has spread in recent years, vast quantities of sensors have been deployed for collecting and exchanging data in various fields related to IoT. An essential technological enabler of IoT is WSNs. WSNs collect user and device data and use these data for various applications such as remote health monitoring for patients, smart grid power usage monitoring, etc.

Figure 1 shows a WSN network model. Generally, WSNs consist of a series of dispersed sensor nodes, plenty distributed users, and one or more gateway nodes which have a powerful performance and play trusted parties. Each set of distributed sensor nodes is located in a specific area. And
a series of sensor nodes collect information data of human, device or environment and then they transmit data to the gateway node through open wireless channels. The gateway can access these data, and analysis of these data can help administrators and automated systems make various functional decisions in real industrial environments. Generally, sensor nodes have limited communication, computing and storage capability. In addition, sensor nodes are easily compromised by attackers and cannot be guaranteed secure, because sensor nodes have limited physical security. Moreover, in WSNs, data are transmitted through open wireless channels and it causes security vulnerabilities that allow data can be captured by malicious attackers. If attackers capture these transmitted data, they can perform variable attacks i.e., man-in-the-middle, replay, privileged insider attacks and identity and password guessing attack and so on. Thus, various protocols have been developed in an attempt to guarantee the security of the transmitted data and the sensor node devices. However, traditional two factor authentication schemes remain vulnerable to guessing attacks according to [1]–[4]. They have been shown that attackers can guess identity and password from identity dictionary space $D_I$ and password dictionary space $D_P$ in real polynomial time. Therefore, in recent years, three-factor based mechanisms that use biometrics of users have been studied. Moreover, the honey_list technique can be used with three-factor to further protect the authentication protocol. Wang and Wang [34], Wang et al. [35] demonstrated that using biometrics and honey_list techniques can be safe, even if two of the three factors are compromised.

Recently, Chen et al. [5] suggested a privacy-preserving authentication protocol for WSNs. However, we demonstrate that the protocol of Chen et al. cannot be safe against stolen smartcard, off-line password and off-line identity guessing and replay attacks. Then, this paper proposes authentication protocol based three-factor utilizing biometrics and honey_list technique for WSNs.

A. MOTIVATION AND CONTRIBUTIONS

In WSNs environments, most authentication protocols are based on two-factor. Thus, they cannot prevent against simultaneously guessing identity and passwords. Furthermore, if users lose their smart cards or attackers steal smart cards, users are vulnerable to password guessing attack. Thus, this paper proposes a three factor authentication protocol to help ensure security of WSNs. The contributions of this paper include:

- This paper discovers that proposed protocol of Chen et al. [5] cannot provide security and is vulnerable to smartcard stolen, identity guessing, password guessing, and replay attacks. And also Chen et al.’s protocol cannot guarantee mutual authentication.
- This paper designs an authentication protocol based on three-factor for WSNs excluding elliptic curve cryptography (ECC), owing to the limited performance capability of sensor nodes. And we adopt the fuzzy-extractor for the biometric awareness. Moreover, we propose authentication protocol using honey_list technique to overcome malicious attacks including smartcard stolen attack and simultaneous guessing attack of identity and password.
- We analyze security using BAN logic, AVISPA software and ROR model for a formal security analysis. We conduct an informal analysis and we show security comparison, computational and communication costs with previous related researches.

B. PAPER ORGANIZATION

We introduce previous interrelated researches in authentication for WSNs in Section II. Section III describes some preliminaries to show necessary backgrounds such as fuzzy extractor, honey_list and related notations. Sections IV and V review the suggested scheme of Chen et al. and analyze its security aspects. Section VI illustrates our proposed protocol for WSNs. Section VII demonstrates the security of the proposed protocol by performing a security analysis. Section VIII compares our efficiency and security features with other previous researches. In the end, we summarize and close the paper in Section IX.

II. RELATED WORKS

Authentication is considered as a primary security service which allows an entity to mutually authenticate with another entity [6]–[20].

Authentication protocols for WSNs have already been researched, and, here, we briefly review works involved in three aspects, i.e., lightweight authentication for WSNs, simultaneous guessing identity and password attack on protocol for WSNs and three-factor based protocol. Owing to the limitations of sensor nodes performance, efficiency communication and computation costs have become an important issue to design authentication protocols for WSNs. For this reason, several lightweight protocols for WSNs have been suggested.

In 2014, Turkanovic et al. [21] suggested key agreement scheme for WSNs. They used masked identities for users and sensors to protect real identities. Unfortunately, Amin and Biswas [22] discovered that their scheme cannot provide security. They discovered that Turkanovic et al.’s protocol doesn’t guarantee safety against smartcard stolen, masquerade and off-line password guessing attacks. Amin and Biswas put forward a novel authentication protocol using
a symmetric key to overcome security vulnerabilities of Turkanovic et al.’s protocol. Nevertheless, Srinivas et al. [23] pointed out that Amin and Biswas’s authentication protocol cannot provide key security and also does not withstand impersonation, stolen smartcard attacks. To resolve these weaknesses, they suggested more efficient user authentication protocol to employing WSNs.

Unfortunately, some researchers have proved that password and smartcard based protocols are not safe against simultaneous guessing of identity and password. In 2016, Maitra et al. [24] proffered an authentication protocol for multiserver environment using a password and a smartcard. Nevertheless, Wang et al. [1] proved that Maitra et al.’s protocol is not safe against off-line guessing attack. They demonstrated that an attacker can conduct attack of simultaneous guessing identity and password through the Zipf’s law [25]. Roy et al. [26] put forward a secure authentication protocol to employing IoT environment. They used a user’s biometric to prevent various attacks. Unfortunately, Park [2] showed Roy et al.’s protocol is insecure against off-line identity guessing attack guessed password at the same time. And also, according to [3], [4], people easily want to choose identities and passwords that are easy to remember for convenience. Both identities and passwords must be taken from a very small dictionary space. Therefore, an attacker can guess identity and password of an user in polynomial time.

To prevent an adversary’s simultaneous identity and password guessing attack, many researchers have suggested using a security three-factor authentication scheme. Biometric keys have several advantages compared with traditional passwords. They are unforgettable and they cannot be lost. Furthermore, they are difficult to fragile and difficult to copy. In 2016, Park and Park [28] discovered that the protocol of Chang et al. [27] cannot provide security such as perfect forward secrecy and password guessing attacks. Moreover Chan et al.’s protocol cannot provide accurate password updates. Thus, Park et al. proposed a three-factor based user authentication protocol for WSNs. They demonstrated that their protocol can provide more secure authentication by utilizing biometrics and elliptic curve cryptosystem. In 2018, Amin et al. [29] suggested a user authentication scheme for medical WSNs. They used a synchronous update mechanism to provide user anonymity. Nevertheless, Li et al. [30] figured out Amin et al.’s protocol cannot provide forward secrecy and also is not safe against denial of service attack. Therefore, they proposed three-factor based with forward secrecy for WMSN with ECC. And they also applied honey_list technique to provide security against device or smartcard stolen and brute-force attacks.

### III. PRELIMINARIES

To improve the readability of this paper, we introduce the preliminary information of this paper: the basis of fuzzy-verifier; honey_list; adversary model; and basic notations adopted in this paper.

#### A. HONEY LIST

Honey Encryption (HE) is an algorithm that can be used to protect data by strongly fooling unauthorized users if an attacker attempts to decrypt plain text using the wrong password or honeyword. When an adversary attempts to decrypt with multiple invalid passwords or honeywords, the HE process generates a fake valid message. HE [31], [32] is based on Distributed Transforming Encoding (DTE). HE manages plain-text space through DTE and includes encryption and decryption. The encryption process takes the space of a plain text message $M$ as input and returns the message $S$ of the $n$-bit string as output. The decryption process makes a conversion that is the value of the seed space $S$ of the $n$-bit string into plain text. DTE encryption and decryption algorithms are as following figure:

In Figure 2, $K$ is a key, $H$ is a hash function, $S$ is a seed, $M$ is a message, $C$ is a cipher-text and $R$ is a random string. $\leftrightarrow$ means uniform random assignment. Let the probability distribution over the message space $M$ be $p_m$. And the message $M$ is over the $M$. If the $M$ gets bigger, the $p_m$ is going to lower. Thus, to assign the corresponding message rate, the DTE process takes a probability distribution theory.

#### FIGURE 2. DTE encryption and decryption algorithms of honey encryption.

In this paper, Honey_list denotes honeywords. Honeywords mean false passwords and honeywords are kinds of honey encryption algorithm. The details of the honeyword generation algorithm are referred to [33]. Among the various methods used to prevent password guessing attack by using the Honey_list during the login phase [33], this paper applies the following method. We allow the login to proceed as usual, but the system tracks the login source. Moreover, the system ends the session when the number of items in the honey_list exceeds the threshold. Wang and Wang [34], Wang et al. [36] demonstrated that simultaneously using a fuzzy-verifier and Honey_list techniques ensures that the system would be safe even if two of the three factors are attacked. In this paper, we use the fuzzy extractor instead of the fuzzy-verifier.

#### B. FUZZY EXTRACTOR

The fuzzy extractor [36] is a technology that uses a user biometric data through data extraction. The data extraction from biometrics normally has difficulty capturing real values due to various noises. To resolve this problem, the fuzzy extractor can help to extract random bit strings evenly without noises. The basic processes of the fuzzy extractor include generation and reproduction. In this paper, $Ge$ denotes the generation process and $Re$ denotes reproduction process.
• $Ge(BIO_i) = \langle R_i, P_i \rangle$. To generate a key information, fuzzy biometric data $BIO_i$ is used as input, public reproduction $P_i$ is a helper string and uniformly random string $R_i$ is secret key data as an output.

• $Re(BIO_i', P_i) = R_i$. To reproduce a secret string $R_i$, the reproduction algorithm is used by the fuzzy extractor. The inputs of reproduction process are $P_i$ and user biometrics $BIO_i$. And the reproduction algorithm reproduces the original secret biometrics $R_i$. For restoring the equal $R_i$, the metric space distance between $BIO_i$ and $BIO_i'$ must be within the allowed specified error tolerance.

C. ADVERSARY MODEL

In the interest of analyze the security of the authentication protocol, it is necessary to first identify attacker’s malicious attacks. We explicitly describe an adversary model consistent with reality by using the widely-accepted “Dolev-Yao threat model” [37] which introduces a simultaneous identity and password guessing attack. We assume capabilities of an adversary as follows.

• The adversary is in full control of transmitted messages through wireless public channels and can learn transmitted messages. Then, the adversary can eliminate, insert, eavesdrop or modify legitimate messages.

• The malicious adversary is able to get or pilfer a valid smartcard, and then the adversary can take out confidential values stored in the smartcard via a power analysis attacks [38], [39].

• The malicious adversary is able to damage some sensor nodes.

• The malicious adversary is able to register as a valid user and conduct a privileged-insider attack for guessing a user’s password [40].

• The malicious adversary is able to get gateway’s secret key when evaluating the system failure. Then, the adversary tries to previous session key.

We assume an adversary can conjecture registered legitimate user’s identity or password. Moreover, we also follow the assumptions in [1]–[4]. We have assumption that the adversary can conjecture identity and password simultaneously. The adversary can choose random identity $ID$ and random password $PW$ from dictionary space of identity $\mathcal{D}_{ID}$ and space of password $\mathcal{D}_{PW}$. The space of identity and password is usually, $|\mathcal{D}_{ID}| < |\mathcal{D}_{PW}| < 10^6$. Therefore, the computational time complexity is very efficient.

D. NOTATIONS

Table 1 describes used the notations in this paper.

IV. REVIEW OF CHEN et al.’s PROTOCOL

We shortly examine the protocol developed by Chen et al., which is composed of the user and sensor’s registration phase, the login and authentication phase and the password change phase. Prior to registration, the gateway forms public parameters $(n, a, b, p, G$, and $h)$ for the ECC and the gateway is published to the whole system. Additionally, the gateway generates a secret key $X_{GWN}$.

A. REGISTRATION PHASE OF USERS AND SENSORS

At Chen et al.’s protocol, they have two registration phase, users and sensors. And the registration phase is through a closed channel.

• User registration: First, a user $U_i$ picks out a unique $ID_i$ and $PW_i$, then $U_i$ randomly generates parameter $r_i$. Then, the user $U_i$ calculates $MP_i = h(r_i || ID_i || PW_i)$ and transmits a composed message $\{ID_i, MP_i\}$ to a gateway $GWN$. After that, $GWN$ calculates $d_i = h(ID_i || X_{GWN})$ and $f_i = d_i \oplus MP_i$. Next, $GWN$ randomly chooses a number $k_i$ and calculates $e_i = h(k_i || X_{GWN})$ and $l_i = e_i \oplus MP_i$. $GWN$ stores values $(f_i, l_i, k_i)$ into a smartcard $SC$ which is issued to the user. At last, $U_i$ stores $\{MP_i, r_i\}$ into the $SC$. Figure 3 describes this phase.

• Sensor registration: A sensor $S_j$ chooses a unique identity $SID_j$ and transmits it to the gateway node $GWN$. After $GWN$ receives $SID_j$, $GWN$ calculates $x_j = h(SID_j || X_{GWN})$ and transmits it to the sensor. $S_j$ keeps $x_j$ in its private memory.

B. LOGIN AND AUTHENTICATION PHASE

When users needs to approach resources of sensor nodes, they have to login and authenticate with a gateway node. Then, the gateway authenticates the sensor nodes. And finally, users and sensors can have a shared session key. The detailed equations are as follows.

Step 1: An user $U_i$ enters $ID_i$, $PW_i$ and a smartcard. The smartcard calculates $MP_i' = h(r_i || ID_i || PW_i)$, $GWN$.


to the gateway node $GWN$. After that, $GWN$ calculates $d_i = h(ID_i || X_{GWN})$ and $f_i = d_i \oplus MP_i$. Next, $GWN$ randomly chooses a number $k_i$ and calculates $e_i = h(k_i || X_{GWN})$ and $l_i = e_i \oplus MP_i$. $GWN$ stores values $(f_i, l_i, k_i)$ into a smartcard $SC$ which is issued to the user. At last, $U_i$ stores $\{MP_i, r_i\}$ into the $SC$. Figure 3 describes this phase.

\begin{table}[h]
  \centering
  \caption{Used notations in this paper.}
  \begin{tabular}{|c|c|}
    \hline
    Notations & Meanings \\
    \hline
    $S_j$, $SID_j$ & $j$th sensor node and its identity \\
    $U_i$, $ID_i$ & $i$th user and his/her identity \\
    GWN & Gateway node \\
    HID_i, PID_j & Hidden identities of $i$th user and $j$th sensor, respectively \\
    $N_i, N_G$ & Random numbers of user and gateway, respectively \\
    $y$ & $GWN$’s long-term secret key \\
    $G$ & The generator of ECC \\
    $X_{GWN}$ & $GWN$’s master key \\
    $SK_{ij}$ & Session key shared by $U_i$, $S_j$ \\
    $h(\cdot)$ & Hash function \\
    $\oplus$ & The conjugation symbol \\
    $\otimes$ & The exclusive-or operator \\
    \hline
  \end{tabular}
\end{table}

\begin{figure}[h]
  \centering
  \caption{User registration phase of Chen et al.’s protocol.}
  \begin{tabular}{c}
    \hline
    Generates random $r_i$ \\
    $MP_i = h(r_i || ID_i || PW_i)$ \rightarrow $d_i = h(ID_i || X_{GWN})$ \\
    \quad $f_i = d_i \oplus MP_i$ \quad chooses random $k_i, e_i = h(k_i || X_{GWN})$ \\
    Stores $MP_i, r_i$ into $SC$ \\
    \hline
  \end{tabular}
\end{figure}
Step 2: After the gateway receives \( < A, k_i, M_1, T_1 > \), the gateway GWN verifies the freshness of the timestamp and calculates \( e_i' = h(k_i||X_{GWN}) \). If they are same values, the gateway chooses a random number \( k_2 \) and then, the gateway calculates \( M_2 = h(A||[ID]||SID)||d_i'||T_1 \) and \( M_1 = e_i(\{ID\}||SID)||M_2 \) and sends a login request message \( < A, k_i, M_1, T_1 > \) to a gateway GWN.

Step 3: The sensor node verifies the freshness of the timestamp and calculates \( e_i' = h(k_i||X_{GWN}) \). If they are same values, the gateway randomly chooses a number \( k_3 \). If they are same values, \( S_j \) calculates \( B = k_2 \cdot G \), \( S_j \) also calculates \( M_4 = h(B||SK_i'||A) \) and \( M_3 = h(x_j||M_3||M_4||B) \), and a shared session key \( SK_{ij} = h(k_2||A) \). Then, it transmits \( < B, M_4, M_5 > \) to GWN.

Step 4: GWN calculates \( M_6 = h(x_j||M_3||M_4||B) \) and verifies whether \( M_6 \neq M_5' \). If they are valid, the gateway randomly chooses a number \( k_3 \), and calculates \( e_{inew} = h(k_j||X_{GWN}) \), \( M_7 = h(e_{inew}||k_j||d_i'||T_1||M_4) \) and \( M_6 = e_{inew}(k_j||M_7) \oplus e_i' \). Then, the gateway sends a message \( < B, M_6 > \) to \( U_i \).

Step 5: \( U_i \) computes \( e_{inew} = h(k_j||X_{GWN}) \), \( M_7 = h(e_{inew}||k_j||d_i'||T_1||M_4) \) and updates smartcard values \( l_i = MP_i \oplus e_{inew} \) and \( k_j = k_3' \).

C. PASSWORD CHANGE PHASE

The user is able to change the \( PW \) within \( k \) times in a period of \( T \) at Chen et al.’s protocol. For using a variable counter, their protocol counts the number of times which is a user incorrectly enter a password. If the user inputs an incorrect password over than \( k \) times, the password will not be allowed to enter. More detailed equations and steps are as follows.

Step 1: A validate user \( U_i \) inserts a smartcard and inputs \( ID_i \) and \( PW_i \).

Step 2: The smartcard checks counter is smaller than \( k \). If it is smaller than \( k \), go Step 4, else, go Step 3.

Step 3: The smartcard checks if \( |TW_{first} - T_{now}| \) is bigger than \( T \). \( TW_{first} \) means the user enters a incorrect password for the first time. If it is bigger than \( T \), go Step 4 and set counter=0. Otherwise, the user is not able to input a password.

Step 4: The smartcard calculates \( h(r_i||ID_i)||PW_i \) and compares with \( MP_i \) stored in the smartcard. If they are same value, the smartcard allows to change password. Otherwise, go to Step 8.

Step 5: Check if counter is larger than 0, set counter = 0.

Step 6: The smartcard calculates \( d_i = f_i \oplus MP_i \) and \( e_i = l_i \oplus MP_i \).

Step 7: The user inputs a new password \( PW_i' \). Then, the smartcard updates \( MP_i' \) to \( MP_i' = h(r_i||ID_i)||PW_i' \) and also updates \( d_i' = d_i \oplus MP_i' \) and \( l_i' = e_i \oplus MP_i' \). Finally, the user completes the password change.

Step 8: Set counter is counter + 1. If counter is 1, go to step 1 and \( TW_{first} \) is set to be now().

V. CRYPTOANALYSIS OF CHEN et al.’S PROTOCOL

We discover security vulnerabilities of Chen et al.’s protocol in this section. They demonstrated that their protocol prevents user anonymity and off-line dictionary attack. Nevertheless, this paper discovers that their protocol is insecure to several attacks as following.

A. SMARTCARD STOLEN ATTACK

Section III-C introduced the adversary model used to obtain values stored in a smartcard. Therefore, an adversary can obtain stored values \( \{MP_i, r_i, f_i, l_i, k_i (= k3)\} \) in a valid user’s smartcard via a stolen smartcard attack.

B. OFF-LINE PASSWORD GUESSING ATTACK

In accordance with references [1]–[4], an adversary can conjecture \( ID_i \) and \( PW_i \) at a same time. From this assumption, the adversary can conjecture a legitimate user’s \( ID_i \) and a \( PW_i \) as following.

Step 1: An adversary randomly selects a identity \( ID^* \) from an identity dictionary space \( D_{ID} \), and picks up a password \( PW^* \) from a password dictionary space \( D_{PW} \). And the adversary obtains smartcard values \( \{MP_i, r_i, f_i, l_i, k_i (= k3)\} \).

Step 2: The adversary calculates \( MP^* = h(r_i||ID^*||PW^*) \) to check the correctness of \( ID^* \) and \( PW^* \).

Step 3: If \( MP^* \) and the stored value \( MP_i \) are the same, the adversary’s guessing result is as successful. Else, the adversary returns to Step 1 and repeats until the adversary correctly guess the ID and password for the user.

\( O(|D_{ID}| \times |D_{PW}| \times T_h) \) is the computational time complexity of this procedure, where \( T_h \) is the hash computation cost. \( |D_{ID}| \) and \( |D_{PW}| \) denote the number of passwords and identities, respectively. According to Zipf’s law [25], \( |D_{ID}| < |D_{PW}| < 10^6 \). Therefore, the off-line guessing attack is very efficient. Thus, the attack can be finished in the real polynomial time.

C. OFF-LINE IDENTITY GUESSING ATTACK

An adversary can conjecture a valid user’s original \( ID_i \) as following steps.
Step 1: An adversary can obtain smartcard values \( MP_i, r_i, f_i, l_i, k_i (= k_3) \) by power analysis. Then, the adversary randomly chooses the identity \( ID^* \) in an identity dictionary space \( D_{ID} \).

Step 2: The adversary calculates \( e_{\text{new}} = MP_i \oplus l_i \) through obtained smartcard values. The adversary computes \( d^* = f_i \oplus MP^i \) and \( M_7 = h(e_{\text{new}}||k_3||d^*||T_1||M_4) \) where \( T_1 \) and \( M_4 \) are obtained through channels. \( e_i' = M_6 \oplus (e_{\text{new}}||k_3||M_7) \) where \( M_6 \) is obtained through channels.

Step 3: The adversary calculates \( M'_2 = h(A||ID^*||SID_j||l'_1||T_1) \) using transmitted values \( SID_j, A, \) and \( T_1 \).

Step 4: The adversary calculates \( M'_1 = e_i' \oplus (ID^*||SID_j||M'_2) \).

Step 5: The adversary compares the computed value \( M'_1 \) with the transmitted value \( M_1 \) to check the correctness of \( ID^* \).

Step 6: If \( M'_1 \) and stored value \( M_1 \) are same, adversary’s guess results as successful. Otherwise, the adversary returns to Steps 1 and repeats until adversary correctly gets ID for the user.

D. USER IMPERSONATION ATTACK
If a malicious adversary can guess a user’s identity according to V-C. The adversary can masquerade the user. The adversary randomly chooses a sensor identity \( SID_j \) and stores values \( A \) and \( T_1 \). Then, the adversary can compute \( M_{2a} = h(A||ID^*||SID_j||l'_1||T_1) \) and also the adversary can compute \( M_{2a} = e_i \oplus (ID^*||SID_j||M_{2a}) \) wherein \( e_i = M_6 \oplus (e_{\text{new}}||k_3||M_7) \) where \( M_6 \) is obtained through channels. Thus, the adversary can impersonate the valid user.

E. REPLAY ATTACK
A malicious adversary attempts to impersonate a valid gateway for obtaining sensitive values of systems. At Chen et al.’s protocol, the adversary is able to generate a legitimate gateway’s message by computed correct values.

Step 1: At a registration phase of sensors, an adversary chooses a sensor identity \( SID_j \). Then, the adversary can obtain a legitimate \( x_j = h(SID_j||X_{GWN}) \).

Step 2: The adversary can compute \( M_3 = h(A||SID_j||x'_j||T_2) \) in a login and authentication phase.

Step 3: Finally, the adversary can generate a legitimate message \( <A, M_3, T_2> \).

In conclusion, the adversary can generate a legitimate message to treat a sensor node.

And also, the adversary can conduct the man-in-the-middle attack. The adversary chooses a random nonce \( k_a \) then the adversary computes \( A_a = k_a \cdot G \).

F. MUTUAL AUTHENTICATION
According to Sections V-C and V-D, an adversary can masquerade a valid user and also can compute a valid login request message. Therefore, Chen et al.’s protocol cannot provide secure mutual authentication.

VI. PROPOSED PROTOCOL
To provide secure wireless IoT service via WSNs, we propose an authentication protocol based on three-factor with the biometrics. And also, our protocol uses “honey_list” and “Fuzzy-extractor” techniques to maintain security even if two of the three factors are damaged by a malicious adversary. Before beginning of the registration phase, a gateway generates a secret key \( X_{GWN} \).

A. REGISTRATION PHASE OF USERS AND SENSORS
To access WSNs service, an user \( U_i \) and a sensor \( S_j \) have to register with gateway. Figures 4 and 5 show the registration phase of users and sensors with detailed equations and steps as following.

- Registration phase of users: An user \( U_i \) selects unique \( ID_i \) and \( PW_i \) and \( U_i \) imprints the biometrics \( BIO_i \). After that \( U_i \) randomly generates a nonce \( r_i \). The adversary chooses a random nonce \( k_i \) and computes \( a_i = h(ID_i||X_{GWN}||k_i) \) wherein \( x_i = h(a_i||PW_i) \). After that, \( GWN \) stores \( ID_i \) with \( k_i \) and \( PW_i \) and stores values \( (b_i, c_i) \) into a smartcard \( SC \). Then, it issues \( SC \) to the user. At last, \( U_i \) calculates \( L_i = h(r_i||PW_i) \) and stores \( [L_i, P_i] \) into the \( SC \). The Figure 4 describes this phase.

- Registration phase of sensors: A sensor \( S_j \) chooses its identity \( SID_j \) and a random nonce \( r_j \). The adversary \( S_j \) computes \( S_1 = SID_j \oplus h(r_j) \) sends \( S_1 \) and \( r_j \) to the gateway node \( GWN \). After \( GWN \) receives registration request message, \( GWN \) computes \( SID'_j = S_1 \oplus h(r_j) \) and \( PID_j = h(SID_j||r_j) \). After that, \( GWN \) generates a random secret key \( y \) and
computes $K_j = h(PID_j || X_{GWN} || y)$ and stores $r_j, PID_j$ in its private memory. Then, $GWN$ sends $K_j$ to the sensor.

Figure 5 describes detailed steps.

**B. LOGIN AND AUTHENTICATION PHASE**

Users have to login and authenticate with the gateway and sensors to access information of sensors. Figure 6 shows the detailed steps of login and authentication phase. We also describe the detailed equations of login and authentication phase.

**Step 1:** User $U_i$ inputs his/her unique identity $ID_i$ and password $PW_i$ and imprints biometric $BIO_i$. Then, $U_i$ calculates $R_i = Re(BIO_i, P_i)$, $r_i = L_i \oplus h(R_i || PW_i)$, $HID_i = h(ID_i || r_i)$ and $HPW_i = h(r_i || ID_i || PW_i)$. $U_i$ extracts $a_i = b_i \oplus HPW_i$ and computes $c_i' = h(a_i || HPW_i)$. Then, $U_i$ computes $c_i = h(a_i || HPW_i)$. $GWN$ checks $c_i' = c_i$ and $a_i$ are equal or not. If they are equal, $U_i$ generates a random number $N_i$ and computes $M_1 = h(a_i || SID_j) \oplus N_i$ and $M_2 = h(a_i || SID_j || N_i)$. Then, $GWN$ sends $M_1, M_2 > $ to a gateway node $GWN$.

**Step 2:** After $GWN$ receives the login request message, $GWN$ retrieves $k_i$ from a database and computes $a_i' = (HID_i || X_{GWN} || k_i)$. Then, $GWN$ checks $M_2 = h(a_i' || SID_j || N_i)$. $GWN$ generates a random number $N_{GWN}$ and computes $K_j = h(ID_j || X_{GWN} || y)$, $M_3 = h(K_j || SID_j || N_{GWN})$, $M_4 = h(K_j || SID_j || N_{GWN})$. $GWN$ checks $M_3 = M_4$. Then, $GWN$ sends $M_3, M_4 > $ to gateway node $GWN$.
If it is not equal, \(a_i'\) inserts into \textit{Honey_list} or suspends the identification when the items in the \textit{Honey_list} exceed a certain threshold. Otherwise, GWN computes \(K_j = h(h(SID_i)||r_j)||X_{GWN}||\gamma\). \(M_5 = h(SID_i)||PID_j||K_j||NG\) and \(M_4 = h(K_j||PID_j||NG)\). Then, GWN sends \(<M_3, M_4>\) to a sensor node \(S_j\).

**Step 3:** \(S_j\) computes \(NG = h(PID_j||K_j)\). \(M_4' = h(PID_j||K_j||NG)\). \(S_j\) checks validation to compare \(M_4\) with \(M_4'\). If they are the same, \(S_j\) randomly generates a nonce \(N_j\) and calculates \(SK_{ij} = h(PID_j||K_j||NG)\) and \(M_5 = h(SK_{ij}||NG)\). Then, \(S_j\) sends \(<M_5>\) to GWN.

**Step 4:** After that, GWN calculates \(SK_{ij} = h(PID_j||K_j||NG)\) and \(M_5' = h(SK_{ij}||NG)\). GWN checks \(M_5 = M_5'\). If it is equal, GWN computes \(HID_{new} = h(NG)||HID_i\). \(a_{new} = h(HID_{new})||X_{GWN}||\gamma\). \(M_6 = (N_j||a_i') \oplus HID_{new}, M_7 = (N_j||a_i') \oplus HID_{new}\). \(M_8 = (N_j||a_i') \oplus SK_{ij} \) and \(M_{gu} = (SK_{ij}||NG)\). Then, GWN sends \(<M_6, M_7, M_8, M_{gu}>\) to \(U_i\). If session key agreement is successful, GWN updates HID\(_i\) to HID\(_{new}\). Otherwise, GWN keeps to store HID\(_i\).

**Step 5:** \(U_i\) computes \(HID_{new}' = M_6 \oplus (N_j||a_i)\), \(a_{new}' = M_7 \oplus (N_j||a_i)\), \(SK_{ij}' = M_8 \oplus (N_j||a_i)\), and \(M_{gu}' = (SK_{ij}||NG)\). \(U_i\) verifies whether \(M_{gu}\) and \(M_{gu}'\) are same value or not. If they are same values, \(U_i\) computes \(b_{new}' = a_{new} \oplus HW_{i}\) and \(c_{new}' = h(a_{new}'||HW_{i})\) and updates \(a_{new}', b_{new}', c_{new}'\) and HID\(_{new}'\). Finally, \(U_i, GWN\) and \(S_j\) authenticate each other and have the same session key.

**C. PASSWORD CHANGE PHASE**

If \(U_i\) wishes to change a password, \(U_i\) conducts the password change phase without the gateway’s assistance. The detailed steps of the password change phase are as following.

**Step 1:** \(U_i\) imprints biometrics \(BIO_i\) and inputs his/her identity and password. And \(U_i\) sends \(ID_i, PW_i\), and \(BIO_i\) to the smartcard.

**Step 2:** The smartcard calculates \(<R_i, P_i> = Ge(BIO_i)\), \(r_i = L_i \oplus h(R_i||PW_i)\) and \(HPW_i = h(r_i||ID_i||PW_i)\) and \(c_i^* = h(a_i||HW_{i})\). Then, smartcard makes a comparison between \(c_i^*\) and \(c_i\) stored value in the smartcard. If they are same values, the smartcard asks the user to supply a new password.

**Step 3:** The user enters a new password \(PW_{new}\) and sends it to the smartcard. Then, smartcard computes \(HPW_{new} = h(r_i||ID_i||PW_{new})\), \(L_{new} = h(R_i||PW_{new}||r_i), b_{new} = a_i \oplus HPW_{new}\) and \(c_{new} = h(a_i||HPW_{new})\). After all computing, the smartcard updates \([L_i', b_i', c_i']\).

**VII. SECURITY ANALYSIS OF THE PROPOSED PROTOCOL**

This section shows that the suggested protocol has security to variable malicious attacks. And also, it shows that our protocol has a secure mutual authentication with key agreement by adopting BAN logic. Besides, we demonstrate that our proposed authentication protocol is secure to guessing attack, man-in-the-middle attack and replay attack employing ROR model and AVISPA.

**A. INFORMAL SECURITY ANALYSIS**

We describe how our protocol achieves security features in this section. And also, we demonstrate that our proposed authentication protocol can ensure safety session key agreement and mutual authentication.

1) **OFF-LINE GUESSING ATTACK**

If a user selects a password which is easy to guess, a malicious adversary is able to conjecture the user’s \(ID_i\) and \(PW_i\) in real polynomial time. Nevertheless, in our authentication protocol, the adversary cannot conjecture user’s \(ID_i\) and \(PW_i\). The adversary can extract values \((b_i, c_i, L_i, P_i)\) stored in a smartcard through the power analysis attack. Then, the adversary can attempt to guess the legitimate user’s \(ID_i\) and \(PW_i\). \(b_i\) and \(c_i\) are masked with \(a_i\) and \(HPW_i\). And also, \(a_i\) is masked with \(X_{GWN}\) and \(k_i\). Therefore, the adversary cannot retrieve user’s identity and password from \(b_i, c_i\). Furthermore, if the adversary attempts to simultaneously guess identity and password, the adversary cannot guess them because of masking with user’s biometric. Meanwhile, the honey_list can prevent to the times in off-line password guessing attack. In conclusion, our authentication protocol is secure to off-line guessing attack.

2) **USER/SENSOR ANONYMITY**

An adversary wants to obtain user’s real identity for performing the tracing attack. In proposed authentication protocol, a true identity \(ID_i\) and \(SID_j\) of user and sensor are encrypted by a random number \(r_i\) and \(r_j\). Meanwhile, \(HID_j\) is updated to \(HID_{new}\) by GWN because \(HID_{new}\) is transmitted through a public channels. Therefore, the adversary cannot know the user’s original \(ID_i\) and sensor’s original identity \(SID_j\).

3) **FORGERY ATTACK**

In our proposed protocol, all transmitted messages are concatenated with the random nonces \(N_i\) and \(N_G\), and the secret parameters \(a_i\) and \(K_j\). The messages are also encapsulated by the one-way collision-resistant cryptographic hash function. It is then impossible to compute correct messages \(M_1\) and \(M_2\) without \(a_i\) on the user side. Moreover, \(a_i\) consists of \(X_{GWN}\) and \(k_i\) which are unknown to the adversary. On the gateway side, \(M_3, M_4, M_5, M_6, M_7\) and \(M_{gu}\) consist of \(a_i, N_i, N_G, PID_j\) and \(K_j\) which are unknown to the adversary. On the sensor side, \(M_5\) is also masked with \(K_j\) and \(N_G\). Therefore, our protocol is secure against forgery attack.

4) **IMPERSONATION ATTACK**

The impersonation attack is a particular case of forgery attack. As an adversary tries to impersonate each entity, the adversary has to compute legitimate messages. In the
proposed protocol, transmitted messages over public channels are encrypted with random secrets $N_i$ and $N_G$. The adversary tries to extract random secrets but the adversary cannot extract them. Meanwhile, $M_2$ is encrypted by $K_j$ and $PID_j$, $K_i$ and $PID_i$, which are masked with random number $r_j$ and secret keys $X_{GWN}$, $y$. In this way, the proposed protocol can be secure to impersonation attack.

5) DESYNCHRONIZATION ATTACK
Assuming a user does not receive the message $< M_6, M_7, M_8, M_{gu} >$ from a gateway because of attacks of adversary or unexpected termination, the adversary can perform the desynchronization attack. However, the adversary cannot perform desynchronization attack because the user checks whether $M_{gu}'$ and $M_{gu}$ are same or not. If it is not same, the session is terminated. Moreover, the gateway does not update $HID_{inew}$ when the session is terminated. In conclusion, the proposed authentication protocol prevents to desynchronization attack.

6) SESSION KEY DISCLOSURE ATTACK
An adversary must know $K_j$ and $N_G$ to compute a valid session key $SK_{ij}$. But, $K_j$ is encrypted with the gateway’s master key $X_{GWN}$, secret key $y$ and random number $r_j$. The adversary cannot extract a random nonce $N_G$. The adversary can also capture the message $M_8$ to compute $SK_{ij}$. However, the adversary does not know the correct random nonce $N_i$. Therefore, we can say that our proposed protocol can resist against session key disclosure attack.

7) TRACE ATTACK
In our proposed protocol, the user’s real identity is hidden by $HID_i$. Moreover, $HID_i$ is updated to $HID_{inew}$ by GWN to protect against adversary’s guessing. And all transmitted messages are changed in all each session because the messages include random numbers are changed in each session. Thus, the proposed protocol resists trace attack.

8) PRIVILEGED-INSIDER ATTACK
We assume that a user is privileged-insider attacker. Then, the privileged-insider attacker knows the registration information $HID_i$, $HPW_i$ of a legitimate $U_i$ over registration phase. Then, the attacker performs the power analysis attack for extracting stores values from a smartcard $\{b_i, c_i, L_i, P_i\}$. However, the attacker cannot guess correctly user’s identity $ID_i$ and password $PW_i$ without having the biometric secret key $R_i$ because of computationally expensive. In concluding, our authentication protocol can prevent privileged-insider attack.

9) SESSION SPECIFIC RANDOM NUMBER LEAKAGE ATTACK
In the proposed protocol, $U_i$ and $GWN$ generate session specific random numbers $N_i$ and $N_G$. Even if $N_i$ and $N_G$ are compromised to the adversary, he/she cannot obtain sensitive information. At the login and authentication phase, $M_1, M_6, M_7$ and $M_8$ are masked with $a_i$. The secret parameter $a_i$ consists of $k_i$ and $X_{GWN}$ which are unknown to the adversary. $M_4$ and $M_5$ are also masked with $K_j, PID_j$ and $SK_{ij}$. The adversary cannot compute $K_j, PID_j$ and $SK_{ij}$ because they consist of $r_j, X_{GWN}$ and $y$. Therefore, our proposed protocol prevents session specific random number leakage attack.

10) STOLEN VERIFIER ATTACK
The adversary can steal a legal registered user’s information from the $GWN$ and $S_j$. However, $HID_i$ is updated to $HID_{inew}$ for every session. Even if $HID_i$ and $k_i$ are compromised to the adversary, he/she cannot obtain entities’ information. This is because the parameters including $HID_i$ are masked with the gateway node’s secret key $X_{GWN}$. If the adversary steals $r_j$ and $PID_j$ through stolen verifier attack, the adversary cannot still compute $K_j$ and $SK_{ij}$ as they are masked with $X_{GWN}$, $y$ and $N_G$. Therefore, the proposed protocol can resist against stolen verifier attack.

11) MAN-IN-THE-MIDDLE ATTACK AND REPLAY ATTACK
We assume that the adversary can learn transmitted messages via open channel. However, the adversary cannot compute a valid login request message as mentioned at Section VII-A4. Moreover, the adversary cannot impersonate a legal registered user because the messages are refreshed in every session with random numbers $N_i$ and $N_G$. In conclusion, our authentication protocol is secure to man-in-the-middle and replay attacks.

12) DENIAL-OF-SERVICE (DoS) ATTACK
The adversary can conduct DoS attack for blocking to user’s access for service. If the adversary intercepts the message $< M_6, M_7, M_8, M_{gu} >$ and replaces with $< M_6, M_7, M_8, M_{gu}' >$, where $M_{gu}' = M_{gu} \oplus N_a$ and $N_a$ is a produced nonce by the adversary. However, our proposed protocol checks whether $M_{gu} \neq M_{gu}'$. Moreover, our proposed protocol can prevent desynchronization attack as Section VII-A5. Therefore, we can say our proposed protocol can prevent DoS attack.

13) KEY AGREEMENT AND MUTUAL AUTHENTICATION
All transmitted messages by each entity are authenticated through verification $M_2 \equiv M_2'$, $M_4 \equiv M_4'$, $M_5 \equiv M_5'$ and $M_{gu} \equiv M_{gu}'$. Moreover, Section VII-A7 shows that all transmitted messages are changed. All entities have authenticated each other, they compute the same session key. Thus, we can say our proposed authentication protocol can achieve secure key agreement and mutual authentication.

B. SECURITY ANALYSIS USING BAN LOGIC
This paper provides the proof which shows that the proposed protocol can provide mutual authentication by performing the BAN logic [41]. We describe basic notations of the BAN logic in the Table 2, and also illustrate logical rules, goals, assumptions and idealized forms. Then, we conduct the BAN logic to confirm the mutual authentication of our proposed protocol.
The Logical rules of the BAN logic are:

1) LOGICAL RULES OF BAN LOGIC
   The logical rules of the BAN logic are:
   
   1. Jurisdiction rule:
      \[ \sigma \models \omega \models S, \quad \sigma \models \omega \models S \]
      \[ \sigma \models S \]
   
   2. Nonce verification rule:
      \[ \sigma \models \#(S), \quad \sigma \models \omega \models \sim S \]
      \[ \sigma \models \omega \models S \]
   
   3. Message meaning rule:
      \[ \sigma \models \sigma \leftrightarrow \omega, \quad \sigma \models \{S\}_K \]
      \[ \sigma \models B \models \sim S \]
   
   4. Belief rule:
      \[ \sigma \models (S, F) \]
      \[ \sigma \models S \]
   
   5. Freshness rule:
      \[ \sigma \models \#(S) \]
      \[ \sigma \models \#(S, F) \]

2) GOALS
   The following goals are presented to demonstrate that the proposed protocol achieves secure mutual authentication:
   
   Goal 1: \( GWN \models U_i \equiv (N_i) \),
   
   Goal 2: \( GWN \models (N_i) \),
   
   Goal 3: \( S_j \models GWN \models (N_G) \),
   
   Goal 4: \( S_j \models (N_G) \),
   
   Goal 5: \( GWN \models S_j \models S_j \stackrel{SK_j}{\leftarrow} GWN \),
   
   Goal 6: \( GWN \models S_j \models \stackrel{SK_j}{GWN} \),
   
   Goal 7: \( U_i \models GWN \models U_i \stackrel{SK_j}{\leftarrow} GWN \),
   
   Goal 8: \( U_i \models U_i \stackrel{SK_j}{GWN} \).

3) IDEALIZED FORMS
   The idealized forms are
   
   \[ M_1 : U_i \rightarrow GWN : (HID_i, SID_j, N_i)_a \]
   \[ M_2 : GWN \rightarrow S_j : (SID_j, PID_j, N_G)_a \]
   \[ M_3 : S_j \rightarrow GWN : (PID_j, N_G, K_j)_{X_{GWN}} \]
   \[ M_4 : GWN \rightarrow U_i : (HID_{inew}, a_{inew}, SK_j)_n \]

4) ASSUMPTIONS
   The following assumptions are generated for the initial state of the proposed protocol to achieve the BAN logic proof:
   
   \[ A_1 : GWN \models (U_i \stackrel{a_i}{\leftarrow} GWN) \]
   \[ A_2 : GWN \models \#(N_i) \]
   \[ A_3 : S_j \models (GWN \stackrel{K_j}{\leftarrow} S_j) \]
   \[ A_4 : S_j \models \#(N_G) \]
   \[ A_5 : GWN \models (S_j \stackrel{X_{GWN}}{\leftarrow} GWN) \]
   \[ A_6 : GWN \models \#(K_j) \]
   \[ A_7 : U_i \models (U_i \stackrel{N_i}{\leftarrow} GWN) \]
   \[ A_8 : U_i \models \#(HID_{inew}) \]
   \[ A_9 : GWN \models U_i \rightarrow (GWN \stackrel{a_i}{\leftarrow} U_i) \]
   \[ A_{10} : S_j \models GWN \models (S_j \stackrel{K_j}{\leftarrow} GWN) \]
   \[ A_{11} : GWN \models S_j \models (S_j \stackrel{SK_j}{\leftarrow} GWN) \]
   \[ A_{12} : U_i \models GWN \models (U_i \stackrel{SK_j}{\leftarrow} GWN) \]

5) PROOF USING BAN LOGIC
   Main proofs using rules and assumptions of the BAN logic are as the following steps:
   
   Step 1: \( S_1 \) can be obtained from \( M_1 \)
   
   \[ S_1 : GWN \leftrightarrow (SID_j, HID_i, N_i)_a \]

   Step 2: For obtaining \( S_2 \), we apply the message meaning rule with \( A_1 \)
   
   \[ S_2 : GWN \models U_i \models \sim (SID_j, HID_i, N_i) \]

   Step 3: For obtaining \( S_3 \), we apply the freshness rule with \( A_2 \)
   
   \[ S_3 : GWN \models \#(SID_j, HID_i, N_i) \]

   Step 4: For obtaining \( S_4 \), we apply the nonce verification rule with \( S_2 \) and \( S_3 \)
   
   \[ S_4 : GWN \models U_i \models (SID_j, HID_i, N_i) \]
Step 5: For obtaining $S_5$, we apply the belief rule

$$S_5: GWN \equiv U_i \equiv (N_i). \quad (\text{Goal 1})$$

Step 6: $S_6$ can be obtained from $M_2$

$$S_6: S_j < (SID_j, PID_j, N_G)K_j.$$  

Step 7: For obtaining $S_7$, we apply the message meaning rule with $A_3$

$$S_7: S_j \equiv GWN \sim (SID_j, PID_j, N_G).$$

Step 8: For obtaining $S_8$, we apply the freshness rule with $A_4$

$$S_8: S_j \equiv \#(SID_j, PID_j, N_G).$$

Step 9: For obtaining $S_9$, we apply the nonce verification rule with $S_7$ and $S_8$

$$S_9: S_j \equiv GWN \equiv (SID_j, PID_j, N_G).$$

Step 10: For obtaining $S_{10}$, we apply the belief rule

$$S_{10}: S_j \equiv GWN \equiv (N_G). \quad (\text{Goal 3})$$

Step 11: $S_{11}$ can be obtained from $M_3$

$$S_{11}: GWN < (PID_j, N_G, K_j)_{xGWN}.$$  

Step 12: For obtaining $S_{12}$, we apply the message meaning rule with $S_{11}$ and $A_5$

$$S_{12}: GWN \equiv S_j \sim (PID_j, N_G, K_j).$$

Step 13: For obtaining $S_{13}$, we apply the freshness rule with $A_6$

$$S_{13}: GWN \equiv \#(PID_j, N_G, K_j).$$

Step 14: For obtaining $S_{14}$, we apply the nonce verification rule with $S_{12}$ and $S_{13}$

$$S_{14}: GWN \equiv S_j \equiv (PID_j, N_G, K_j).$$

Step 15: Since the session key $SK_{ij} = h(PID_j||K_j||N_G)$ from $S_{14}$,

$$S_{15}: GWN \equiv S_j \equiv S_i \rightarrow SK_{ij} \rightarrow GWN. \quad (\text{Goal 5})$$

Step 16: $S_{16}$ can be obtained from $M_4$

$$S_{16}: U_i < (HID_{\text{inew}}, a_{\text{inew}}, SK_{ij})_{xGWN}.$$  

Step 17: For obtaining $S_{17}$, we apply the message meaning rule with $S_{16}$ and $A_7$

$$S_{17}: U_i \equiv GWN \sim (HID_{\text{inew}}, a_{\text{inew}}, SK_{ij})_{xGWN}.$$  

Step 18: For obtaining $S_{18}$, we apply the freshness rule with $S_{17}$ and $A_8$

$$S_{18}: U_i \equiv \#(HID_{\text{inew}}, a_{\text{inew}}, SK_{ij}).$$

Step 19: For obtaining $S_{19}$, we apply the nonce verification rule with $S_{17}$ and $S_{18}$

$$S_{19}: U_i \equiv GWN \equiv (HID_{\text{inew}}, a_{\text{inew}}, SK_{ij}).$$

Step 20: For obtaining $S_{20}$, we apply the belief rule

$$S_{20}: U_i \equiv GWN \equiv (SK_{ij}).$$

Step 21: From $S_{20}$, we can obtain $S_{21}$

$$S_{21}: U_i \equiv GWN \equiv U_i \leftrightarrow SK_{ij} \rightarrow GWN. \quad (\text{Goal 7})$$

Step 22: We apply the jurisdiction rule with $S_5$ and $A_9$ to obtain

$$S_{22}: GWN \equiv (N_i). \quad (\text{Goal 2})$$

Step 23: We apply the jurisdiction rule with $S_{10}$ and $A_{10}$ to obtain

$$S_{23}: S_j \equiv (N_G). \quad (\text{Goal 4})$$

Step 24: We apply the jurisdiction rule with $S_{15}$ and $A_{11}$ to obtain

$$S_{24}: GWN \equiv S_j \leftrightarrow SK_{ij} \rightarrow GWN. \quad (\text{Goal 6})$$

Step 23: We apply the jurisdiction rule with $S_{21}$ and $A_{12}$ to obtain

$$S_{23}: U_i \equiv U_i \leftrightarrow SK_{ij} \rightarrow GWN. \quad (\text{Goal 8})$$

### C. FORMAL SECURITY VERIFICATION USING AVISPA SIMULATION

This section shows that our proposed protocol can be secure to man-in-the-middle and replay attacks by being universally adopted Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool [42], [43]. The AVISPA simulation tool uses High-Level Protocol Specification Language (HLPSL) [44] to check if protocols are secure. The HLPSL inputs to one of four back-end models which are “On-the-Fly Model Checker (OFMC) [45]”, “Constraint Logic-based Attack Searcher (CL-AtSE)” [46], “Tree automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)”, and “SAT-based Model Checker (SATMC)”. This input is converted to a format called “Intermediate Format (IF)”, and output in a format called “Output format (OF)”. The OF shows security analysis results of protocols. We provide similar simulation results as adopted in [47]–[49]. Figs. 7, 8 and 9 each describe role of user, gateway and sensor nodes. And the Figure 10 shows goals and environment of our proposed protocol. Then, according to goals, the results is shown in Fig 11. In CL-AtSe, the translation time has 0.09 seconds. And search time is 7.89 seconds for visiting 1,040 nodes in OFMC analysis. Two of the results all show that the proposed protocol is safe. Therefore, the proposed protocol can be secure to man-in-the-middle and replay attacks.

### D. FORMAL SECURITY ANALYSIS UNDER ROR MODEL

We adopt the ROR model [50] to illustrate the semantic security of our suggested authentication protocol. This section demonstrates that our proposed protocol can achieve the session key security by employing the ROR model. We shortly describe the ROR model and present the proof of the session key security of protocol in Theorem 1. In this model, the proposed protocol has three participants $P_i$, which are user $P_{U_i}$, gateway $P_{GWN}$ and sensor $P_{S_i}$. And each participant has $i$th denotes an instance of an executing participant. We assume that $P_{U_i}^{t_i}$, $P_{GWN}^{t_i}$ and $P_{S_i}^{t_i}$ are instances $i$th of the user, $i$th of the gateway and $i$th of the sensor, respectively. Moreover, we assume that an adversary $A$ can modify, eliminate or insert or learn transmitted messages during the communication. Under the ROR model, the model defines various queries simulating a real attack like Execute, CorruptSC, Reveal, Send and Test queries. The detailed description of queries is as follows.

- **Execute($P_{U_i}^{t_i}$, $P_{GWN}^{t_i}$, $P_{S_i}^{t_i}$):** $A$ performs this query to eavesdrop exchanged messages between wireless
FIGURE 7. Role of user. Communicating entities \(U_i\), \(GWN\) and \(S_j\) over public channels.

- **CorruptSC**: \(A\) can extract all stored sensitive parameters from the smartcard of the user to use the CorruptSC query.

- **Reveal\(P\)**: \(A\) can reveal the session key \(SK_{ij}\) between \(P\) and its partner in the current session.

- **Send\(P\)**: This query corresponds to the security of the session key among with \(U_i\), \(GWN\) and \(S_j\) following the ROR model. Before the game starts, a coin \(c\) without prejudice is flipped. According to the coin result, the following decision is made. Assume that \(A\) executes Test\(P\) and the session key for \(c = 1\) or a random number if \(c = 0\). Otherwise, it returns a null value (\(\perp\)).

Moreover, all communicating participants and \(A\) can access a collision-resistant hash function \(h(\cdot)\) that is modeled as a random oracle, say \(\text{Hash}\).

Wang et al. [25] demonstrated that the chosen passwords by users conform with the Zipf’s law, which differs significantly from uniform distribution. We apply the Zipf’s law for the formal analysis to prove the session key security. We show the detailed Theorem 1 is as in the following.

![FIGURE 7. Role of user.](image)

FIGURE 8. Role of gateway.

**Theorem 1**: We define the advantage probability of an adversary \(A\) running in polynomial time who can break the session key security of the proposed authentication protocol as \(\text{Adv}_A\). Then,

\[
\text{Adv}_A \leq \frac{q_h^2}{|\text{Hash}|} + 2\max\{C' \cdot q_{send}^2, q_{send}^2 / 2r\}
\]

where \(q_h\), \(q_{send}\) and \(|\text{Hash}|\) mean “the amount of Hash queries, the amount of Send queries and the range space of the hash function”, respectively. \(C'\) and \(s'\) mean the Zipf’s parameters, and \(l_q\) is the number of bits in the biometric secret key \(b_i\) of \(U_i\).

**Proof**: We provide the similar proof as adopted in [51]–[53], and we follow this proof. We proof the session key security through a sequence of four games, namely, \(GM_j\), where \(j \in [0, 3]\) wherein an event is defined in which \(A\) is able to accurately conjecture the random bit \(c\) in \(GM_j\), which is defined by \(\text{Succ}_{A,GM_j}\), and its advantage to win the game \(GM_j\) is defined by \(\text{Pr}[\text{Succ}_{A,GM_j}]\). The detailed description of defined four games are as follows.
FIGURE 9. Role of sensor.

- **GM0**: This game is equivalent as the “real attack by A against the proposed protocol” in relation to the game GM0. The randomly selected bit c is at the beginning of the game. Therefore, we get from the semantic security definition,

\[ \text{Adv}_A = |2\Pr[\text{Succ}_A, \text{GM}_0] - 1| \quad (1) \]

- **GM1**: This game is modeled that A can eavesdrop exchanged messages \(< \text{HID}_i, M_1, M_2 >, < M_3, M_4 >, < M_5 > > \) and \(< M_6, M_7, M_8, M_{gu} > \) through an eavesdropping attack. These messages are intercepted by A over the login and authentication phase employing the Execute query. And next, A executes Reveal and Test queries to verify whether the derived session key SKij/SKq between U1, GWN and Sj is a real or random key. In our proposed protocol, we take notice of the session key which is constructed as \( SKij = h(\text{PID}_i || K_j || N_j) \). To derive the session key, A have to need the secret identity PIDi of sensor and also random nonce Nj. And A must calculate the Kj with long term key XGWN and short term secret key y which are unknown to A. In conclusion, we obtain the truth that the A cannot have the GM1’s winning probability. Therefore, games GM0 and GM1 are indistinguishable, we then obtain,

\[ \Pr[\text{Succ}_A, \text{GM}_1] = \Pr[\text{Succ}_A, \text{GM}_0] \quad (2) \]

- **GM2**: In this game, Hash and Send queries are performed to model it calls an “active attack”. The exchanged message \(< \text{HID}_i, M_1, M_2 >, \) the terms \( M_2 \) and \( \text{HID}_i \) are protected by Hash. Likewise, the terms \( M_3, M_4, M_5, M_{gu} \) are protected by hash function. In addition, All terms including \( M_1, M_6, M_7, M_8 \) are constructed the secret credentials and random numbers. Besides, deriving random numbers or secret values from the exchange messages are “computationally infeasible task” because of collision-resistant property. Thus, there are not collision happens if the Hash query is executed. As games GM0 and GM1 are indistinguishable except for the inclusion of the Hash query simulation in GM2. We can obtain the following to adopt the birthday paradox results:

\[ |\Pr[\text{Succ}_A, \text{GM}_2] - \Pr[\text{Succ}_A, \text{GM}_0]| \leq \frac{q^2}{2|\text{Hash}|} \quad (3) \]

- **GM3**: GM3 is the final game which are executed with the CoorruptSC query. According to CoorruptSC query, A can extract stored sensitive values \( \{b_i, c_i, L_i, P_i\} \) by performing the power analysis attack. Here, HPW_i = \( h(r_i || ID_i || PW_i) \), \( L_i = r_i \oplus h(R_i || PW_i) \), \( b_i = a_i \oplus HPW_i \), \( a_i = h(\text{HID}_i || X_{GWN} || k_i) \) and \( c_i = h(a_i || HPW_i) \). Then, to derive the secret values \( r_i \) and \( k_i \) from \( a_i, L_i \) and \( HPW_i \), A have to know the unknowns \( ID_i, PW_i, R_i \) and the

FIGURE 10. Role of session, goal and environment.
gateway’s secret key $X_{GWN}$. Thus, it has computationally infeasible problem for $A$ guessing the password of a legitimate user. Besides, the probability that $A$ guesses the biometric key $R_i$ of $l_R$ bits is roughly $\frac{1}{2^{l_R}}$. Thus, in the absence of a password or biometric guessing attack, the games $GM_2$ and $GM_3$ are the same. In conclusion, by utilizing the Zipf’s law on passwords, we have the next results:

$$\Pr[\text{Succ}_A, GM_3] - \Pr[\text{Succ}_A, GM_2] \leq \max\{C' \cdot q_{send}' \cdot \frac{q_{send}}{2^{l_R}}\} \quad (4)$$

Due to all the games have been run, $A$ must conjecture the exact bit $c$. Consequently, we can obtain below equation:

$$\Pr[\text{Succ}_A, GM_3] = \frac{1}{2}. \quad (5)$$

We can obtain the following result from Eqs. (1) and (2):

$$\frac{1}{2} \text{Adv}_A = |\Pr[\text{Succ}_A, GM_1] - \frac{1}{2}| = |\Pr[\text{Succ}_A, GM_1] - \frac{1}{2}|. \quad (6)$$

Again, Eqs. (5) and (6) give the below equation:

$$\frac{1}{2} \text{Adv}_A = |\Pr[\text{Succ}_A, GM_1] - \Pr[\text{Succ}_A, GM_3]|. \quad (7)$$

We can obtain Eq. (8) by applying the triangular inequality with Eqs. (4), (5) and (7).

$$\frac{1}{2} \text{Adv}_A = |\Pr[\text{Succ}_A, GM_1] - \Pr[\text{Succ}_A, GM_3]| \leq |\Pr[\text{Succ}_A, GM_1] - \Pr[\text{Succ}_A, GM_2]| + |\Pr[\text{Succ}_A, GM_3] - \Pr[\text{Succ}_A, GM_3]| \leq \frac{q_{send}^2}{2[\text{Hash}]} + \max\{C' \cdot q_{send}' \cdot \frac{q_{send}}{2^{l_R}}\} \quad (8)$$

Finally, we can obtain the required result of multiplying both sides of Eq. (8) with a multiple of 2:

$$\text{Adv}_P \leq \frac{q_{send}^2}{[\text{Hash}]} + 2 \max\{C' \cdot q_{send}' \cdot \frac{q_{send}}{2^{l_R}}\}.$$  

Therefore, Theorem 1 is proved. \qed
TABLE 4. Computation and communication cost of login and authentication phase.

| Protocol          | User   | Sensor | Gateway | Total cost | Communication cost |
|-------------------|--------|--------|---------|------------|--------------------|
| Amin and Biswas Case-1 [22] | $7T_h$ | $5T_h$ | $8T_h$ | $20T_h (10.0\, ms)$ | 408 bytes |
| Amin and Biswas Case-2 [22] | $8T_h$ | $5T_h$ | $7T_h$ | $20T_h (10.0\, ms)$ | 540 bytes |
| Amin et al. [29] | $12T_h$ | $18T_h$ | $6T_h$ | $36T_h (18.0\, ms)$ | 404 bytes |
| Chen et al. [5]  | $57h + 2T_{mul}$ | $47h + 2T_{mul}$ | $8T_h$ | $17T_h + 4T_{mul} (260.8\, ms)$ | 380 bytes |
| Ours             | $T_f + 6T_h$ | $4T_h$ | $9T_h$ | $T_f + 19T_h (72.5\, ms)$ | 352 bytes |

VIII. ANALYSIS OF SECURITY AND EFFICIENCY FEATURES

This section discusses security and efficiency aspects of the proposed protocol. We compare the security of our protocol with other related protocols and compare the performance, i.e., computation cost and communication cost with relevant protocols.

A. SECURITY FEATURES COMPARISON

This section compares the security features of our proposed protocol with related schemes [5], [22], [29]. The results of comparison are shown in Table 3. According to Table 3, All previously researches cannot resist the smartcard stolen attack, and also most of researches cannot prevent the desynchronization attack and cannot provide mutual authentication. Therefore, our proposed protocol provides superior security and functionality features according to comparison of results.

B. COMPUTATIONAL AND COMMUNICATION COSTS COMPARISON

We make the computation costs comparison between our proposed protocol and previous related works in this section. Table 4 describes the results of comparing the login and authentication phase. For comparison, we follow the experimental reported results in [54]. We define $T_h$, $T_f$ and $T_{mul}$ as the execution time needed for a hash function, a fuzzy extraction, and an elliptic curve point multiplication, where $T_{mul}$, $T_h$ and $T_f$ are 63.075 ms, 0.5 ms and 63.075 ms, respectively. The exclusive-or (XOR) execution time is not included because it can be ignored in comparison with other operations. Our proposed protocol requires $T_f + 19T_h$ as the total cost. This is higher than Amin and Biswas’s protocol and Amin et al.’s protocol. However, the computational demand for a sensor node is most lightweight than other related works. Also, our proposed protocol allows for a lighter computation than Chen et al.’s protocol. Thus, we can say that our proposed protocol is more efficient than related researches in WSN environment. We also compare the communication overheads with related protocols. For the comparison, we follow the assumption of Chen et al. [5]. Thus, we assume that the timestamp size is 4 bytes and the identity is 8 bytes, a random nonce is 20 bytes and the byte length of a point on the elliptic curve is 48 bytes. Besides, the hash output is 32 bytes. The sum of communicational cost also describes in Table 4. In conclusion, we can say our authentication scheme is more efficient compared to other related previous researched protocols.

IX. CONCLUDING REMARKS

Due to the development of the Internet, the number of objects connected to the IoT is increasing. Therefore, it is necessary to provide a secure service of IoT-enabled WSN that connects sensors of objects. Recently, previous researches and the protocol of Chen et al. are insecure to simultaneous ID and password guessing attacks, and Chen et al.’s protocol is also insecure to replay attack. To resolve these vulnerabilities, this paper provides a more efficient and secure three factor authentication protocol for WSNs using the honey list technique. We show that the proposed protocol is able to provide secure mutual authentication by employing the BAN logic. Moreover, we applied the broadly-accepted ROR model to prove that our protocol could achieve the session key security. Furthermore, we applied AVISPA simulation to show that the proposed protocol could prevent man-in-the-middle and replay attacks. This paper also provided the informal security analysis to demonstrate how the proposed authentication protocol is secure against impersonation, guessing, smartcard stolen, man-in-the-middle, replay, desynchronization and privileged-insider attacks. Furthermore, our protocol can provide mutual authentication and user/sensor anonymity. We also performed a performance analysis to show that our protocol is efficient. In conclusion, the proposed authentication protocol is more secure and efficient for application in practical WSN environment than other related schemes.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers and the associate editor for their valuable feedback on the articles which helped us to improve its quality and presentation.

REFERENCES

[1] C. Wang, G. Xu, and W. Li, “A secure and anonymous two-factor authentication protocol in multicast environment,” Secur. Commun. Netw., vol. 2018, pp. 1–15, Apr. 2018.
[2] Y. Park, “A secure user authentication scheme with biometrics for IoT medical environments,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 11, pp. 607–615, 2018.
[3] F. Wu, L. Xu, S. Kumari, and X. Li, “An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks,” Multimedia Syst., vol. 23, no. 2, pp. 195–205, Mar. 2017.
[4] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurthy, and S.-S. Yeo, “Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks,” Multimedia Syst., vol. 21, no. 1, pp. 49–60, Feb. 2015.
[5] Y. Chen, L. López-Santidrián, J.-F. Martínez, and P. Castillejo, “A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: Lightpriauth,” J. Sensor, vol. 2018, Sep. 2018, Art. no. 7574238.
[46] A VISPA. [Online]. Available: http://www.avispa-project.org/

[47] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, “Secure authentication protocol with novel server mutual verification for multiserver environments,” Int. J. Commun. Syst., vol. 34, no. 3, pp. 1403–1410, Mar. 2018.

[48] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” ACM Trans. Comput. Syst., vol. 8, no. 1, pp. 18–36, 1990.

[49] D. V. Oheimb, “The high-level protocol specification language HLPSL (Lecture Notes in Computer Science), vol. 1666.

[50] A VISPA. (2020). Automated Validation of Internet Security Protocols and Applications. Accessed: Mar. 2020. [Online]. Available: http://www.avispa-project.org/

[51] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” ACM Trans. Comput. Syst., vol. 8, no. 1, pp. 18–36, 1990.

[52] AVISPA. (2020). Automated Validation of Internet Security Protocols and Applications. Accessed: Mar. 2020. [Online]. Available: http://www.avispa-project.org/

[53] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, “Secure authentication protocol for wireless sensor networks in vehicular ad hoc networks,” IEEE Access, vol. 5, pp. 14966–14980, 2017.

[54] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., Interlaken, Switzerland, 2004, pp. 223–250.

[55] D. V. Doele and A. Yao, “On the security of public key protocols,” IEEE Trans. Inf. Theory, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[56] 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.

[57] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 1666.

[58] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” ACM Trans. Comput. Syst., vol. 8, no. 1, pp. 18–36, 1990.

[59] AVISPA. (2020). Automated Validation of Internet Security Protocols and Applications. Accessed: Mar. 2020. [Online]. Available: http://www.avispa-project.org/
M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password-based authenticated key exchange in the three-party setting,” in Public Key Cryptography (Lecture Notes in Computer Science), vol. 3386. Les Diablerets, Switzerland: Springer, 2005, pp. 65–84.

J. Lee, S. Yu, K. Park, Y. Park, and Y. Park, “Secure three-factor authentication protocol for multi-gateway IoT environments,” Sensors, vol. 19, no. 10, p. 2358, May 2019.

K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, “A dynamic privacy-preserving key management protocol for V2G in social Internet of Things,” IEEE Access, vol. 7, pp. 76812–76832, 2019.

J. Lee, S. Yu, K. Park, J. Lee, Y. Park, Y. Park, S. Lee, and B. Chung, “Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment,” Appl. Sci., vol. 10, no. 5, p. 1758, Mar. 2020.

A. Ostad-Sharif, D. Abbasinezhad-Mood, and M. Nikooghadam, “A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications,” J. Med. Syst., vol. 43, no. 1, p. 10, Jan. 2019.

**JOONYOUNG LEE** received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering. His research interests include authentication, the Internet of Things, and information security.

**SUNJIN YU** received the B.S. degree in electronics engineering from Daegu University, in 2017, and the M.S. degree from Kyungpook National University, Daegu, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in electronics engineering. His research interests include authentication, post-quantum cryptography, VANET, blockchain, and information security.

**MYEONGHYUN KIM** received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering. His research interests include authentication, blockchain, the Internet of Things, and information security.

**ASHOK KUMAR DAS** (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, network security, blockchain, security in the Internet of Things (IoT), the Internet of Vehicles (IoV), the Internet of Drones (IoD), smart grids, smart city, cloud/fog computing and industrial wireless sensor networks, and intrusion detection. He has authored over 225 articles in international journals and conferences in the above areas, including over 190 reputed journal articles. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEFENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the IEEE CONSUMER ELECTRONICS MAGAZINE, the IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE ACCESS, the IEEE COMMUNICATIONS MAGAZINE, FUTURE GENERATION COMPUTER SYSTEMS, COMPUTERS AND ELECTRICAL ENGINEERING, COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE, COMPUTER STANDARDS AND INTERFACE, COMPUTER NETWORKS, EXPERT SYSTEMS WITH APPLICATIONS, and JOURNAL OF NETWORK AND COMPUTER APPLICATIONS. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, INTERNATIONAL JOURNAL OF INTERNET TECHNOLOGY AND SECURED TRANSACTIONS (Inderscience), and IET COMMUNICATIONS. He is the Guest Editor of COMPUTERS AND ELECTRICAL ENGINEERING (Elsevier) for the special issue on Internet of Things and the IoV in e-healthcare, ICT EXPRESS (Elsevier) for the special issue on Blockchain Technology and Applications for 5G Enabled IoT, and WIRELESS COMMUNICATIONS AND MOBILE COMPUTING (Wiley/Hindawi) for the special issue on Attacks, Challenges, and New Designs in Security and Privacy for Smart Mobile Devices. He has served as a Program Committee Member in many international conferences. He also served as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN’19), Avila, Spain, in June 2019.

**YOUNGHO PARK** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. His research interests include computer networks, multimedia, and information security.